Ether Authority

www.EtherAuthority.io
audit@etherauthority.io

SMART CONTRACT

Security Audit Report

Project:	Bass Exchange
Website:	https://bass.exchange
Platform:	Base chain
Language:	Solidity
Date:	September 21st, 2023

Table of contents

Email: audit@EtherAuthority.io

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.

Email: audit@EtherAuthority.io

Introduction

EtherAuthority was contracted by the Bass Exchange team to perform the Security audit of the Bass Exchange 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 September 21st, 2023.

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

- Bass Exchange is a contract that can be divided into multiples, each with unique functionalities:
    - **Asset:** This contract involves presenting an asset in a pool.
    - **CoreV2:** The CoreV2 contract is responsible for handling the mathematical operations of the Wombat protocol.
    - **DynamicPool:** The Dynamic Pool contract is responsible for managing deposits, withdrawals, and swaps, while maintaining a mapping of assets and parameters.
    - **DynamicPoolV2:** The Dynamic Pool V2 contract is responsible for managing deposits, withdrawals, and swaps, while maintaining asset mapping and parameters.
    - **HighCovRatioFeePoolV2:** The HighCovRatioFeePoolV2 contract offers high cov ratio fee protection for managing a pool.
    - **PausableAssets:** The PausableAssetscontract is responsible for managing the assets pause and unpause of the Wombat protocol.
    - **PoolV2:** Pool V2 is responsible for managing deposits, withdrawals, and swaps, while maintaining a mapping of assets and parameters.
    - **MultiRewarderPerSec:** This contract has no knowledge of the LP amount, and Master is responsible for passing the amount into this contract. Supports multiple reward tokens
- There are 15 smart contracts, 5 libraries, 13 interface files which were included in the audit scope. And there were some standard library code such as OpenZepelin,

which were excluded. Because those standard library code is considered as time tested and community audited, so we can safely ignore them.

Audit scope

Name	Code Review and Security Analysis Report for Bass Exchange Smart Contracts
Platform	**Base chain / Solidity**
File 1	ABnbcAsset.sol
File 2	Asset.sol
File 3	BnbxAsset.sol
File 4	DynamicAsset.sol
File 5	StkbnbAsset.sol
File 6	DynamicPool.sol
File 7	DynamicPoolV2.sol
File 8	HighCovRatioFeePool.sol
File 9	HighCovRatioFeePoolV2.sol
File 10	PausableAssets.sol
File 11	Pool.sol
File 12	PoolV2.sol
File 13	MasterWombatV4.sol
File 14	MultiRewarderPerSec.sol
File 15	CoreV2.sol
Github commit hash	39779d604383565cb7ddd6ae9f66965e2de969ac
Audit Date	September 21st, 2023

Claimed Smart Contract Features

Claimed Feature Detail	Our Observation
File 1 ABnbcAsset.sol **Other Specifications:** • Contract presenting an asset in a pool. • The relative price of an asset may change over time.	**YES, This is valid.**
File 2 Asset.sol • Decimals: 18 **The Owner has control over the following functions:** • Set the pool address. • Set the maximum supply value. **Other Specifications:** • Contract presenting an asset in a pool.	**YES, This is valid.**
File 3 BnbxAsset.sol **Other Specifications:** • Contract presenting an asset in a pool. • The relative price of an asset may change over time.	**YES, This is valid.**
File 4 DynamicAsset.sol **Other Specifications:** • Contract presenting an asset in a pool. • The relative price of an asset may change over time.	**YES, This is valid.**
File 5 StkbnbAsset.sol **Other Specifications:** • Contract presenting an asset in a pool. • The relative price of an asset may change over time.	**YES, This is valid.**

File 6 CoreV2.sol **Other Specifications:** • CoreV2 is responsible for handling math operations of the Wombat protocol, with all parameters being signed integers with 18 decimals.	**YES, This is valid.**
File 7 DynamicPool.sol **Other Specifications:** • The Dynamic Pool contract is responsible for managing deposits, withdrawals, and swaps, while maintaining a mapping of assets and parameters.	**YES, This is valid.**
File 8 HighCovRatioFeePool.sol **The Owner has control over the following functions:** • Set the CovRatio fee parameter.	**YES, This is valid.**
File 9 PausableAssets.sol **Other Specifications:** • The PausableAssetscontract is responsible for managing the assets pause and unpause of the Wombat protocol.	**YES, This is valid.**
File 10 Pool.sol **The Owner has control over the following functions:** • Set the dev address. • Set the master wombat address. • Set the pool's amplification factor value. • Set the pool's haircut rate value. • Set the LP Dividend Ratio and retention Ratio values. • Set the fee beneficiary address. **Other Specifications:** • The Pool is responsible for managing deposits, withdrawals, and swaps, while maintaining a mapping of assets and parameters.	**YES, This is valid.**

File 11 MasterWombatV4.sol **The Owner has control over the following functions:** • Pause pool, restricting certain operations. • Unpause pool, enabling certain operations. • Add a new lp to the pool. • Update the given pool's reward. • Update the emission partition value.	**YES, This is valid.**
File 12 MultiRewarderPerSec.sol **The Owner has control over the following functions:** • Set the operator address. • Set the reward token. • Sets the distribution reward rate. • Emergency withdrawal. **Other Specifications:** • This contract has no knowledge of the LP amount, and Master is responsible for passing the amount into this contract. Supports multiple reward tokens.	**YES, This is valid.**
File 13 PoolV2.sol **The Owner has control over the following functions:** • Set the min fee to mint value. • Update assets to pool value. • Transfer Tip Bucket. • Set the pool's amplification factor value. • Set the pool's haircut rate value. • Set the LP Dividend Ratio and retention Ratio values. • Set the fee beneficiary address. **Other Specifications:** • Pool V2 is responsible for managing deposits, withdrawals, and swaps, while maintaining a mapping of assets and	**YES, This is valid.**

parameters.	
File 14 DynamicPoolV2.sol **Other Specifications:** • The Dynamic Pool V2 contract is responsible for managing deposits, withdrawals, and swaps, while maintaining asset mapping and parameters.	**YES, This is valid.**
File 15 HighCovRatioFeePoolV2.sol **The Owner has control over the following functions:** • Set the CovRatio fee parameter. **Other Specifications:** • The HighCovRatioFeePoolV2 contract offers high cov ratio fee protection for managing a pool.	**YES, This is valid.**

Audit Summary

According to the standard audit assessment, Customer`s solidity smart contracts are **“Secured”**. Also, these contracts 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, 2 low and 1 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	Passed
	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	Moderated
	High consumption ‘storage’ storage	Passed
	Assert() misuse	Passed
Business Risk	The maximum limit for mintage not set	Passed
	“Short Address” Attack	Passed
	“Double Spend” Attack	Passed

Overall Audit Result: PASSED

Email: audit@EtherAuthority.io

Code Quality

This audit scope has 15 smart contract files. Smart contracts contain Libraries, Smart contracts, inherits and Interfaces. This is a compact and well written smart contract.

The libraries in Bass Exchange 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 Bass Exchange Protocol.

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

Code parts are well commented on smart contracts.

Documentation

We were given a Bass Exchange smart contract code in the form of a github.com web link. The hash of that code is mentioned above in the table.

As mentioned above, code parts are well commented on. And 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 project URL: https://bass.exchange 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

ABnbcAsset.sol

Functions

Sl.	Functions	Type	Observation	Conclusion
1	constructor	write	Passed	No Issue
2	getRelativePrice	external	Passed	No Issue
3	getRelativePrice	external	Passed	No Issue

DynamicAsset.sol

Functions

Sl.	Functions	Type	Observation	Conclusion
1	constructor	write	Passed	No Issue
2	getRelativePrice	external	Passed	No Issue
3	setPool	external	access only Owner	No Issue
4	setMaxSupply	external	access only Owner	No Issue
5	decimals	read	Passed	No Issue
6	underlyingTokenBalance	external	Passed	No Issue
7	transferUnderlyingToken	external	access only Pool	No Issue
8	mint	external	access only Pool	No Issue
9	burn	external	access only Pool	No Issue
10	addCash	external	access only Pool	No Issue
11	removeCash	external	access only Pool	No Issue
12	addLiability	external	access only Pool	No Issue
13	removeLiability	external	access only Pool	No Issue

Asset.sol

Functions

Sl.	Functions	Type	Observation	Conclusion
1	constructor	write	Passed	No Issue
2	onlyPool	modifier	Passed	No Issue
3	setPool	external	access only Owner	No Issue
4	setMaxSupply	external	The value limit is not set	Refer Audit Findings
5	decimals	read	Passed	No Issue
6	underlyingTokenBalance	external	Passed	No Issue
7	transferUnderlyingToken	external	access only Pool	No Issue
8	mint	external	access only Pool	No Issue
9	burn	external	Unlimited burn	Refer Audit Findings
10	addCash	external	access only Pool	No Issue

11	removeCash	external	access only Pool	No Issue
12	addLiability	external	access only Pool	No Issue
13	removeLiability	external	access only Pool	No Issue
14	onlyOwner	modifier	Passed	No Issue
15	owner	read	Passed	No Issue
16	_checkOwner	internal	Passed	No Issue
17	renounceOwnership	write	access only Owner	No Issue
18	transferOwnership	write	access only Owner	No Issue
19	_transferOwnership	internal	Passed	No Issue
20	name	read	Passed	No Issue
21	symbol	read	Passed	No Issue
22	decimals	read	Passed	No Issue
23	totalSupply	read	Passed	No Issue
24	balanceOf	read	Passed	No Issue
25	transfer	write	Passed	No Issue
26	allowance	read	Passed	No Issue
27	approve	write	Passed	No Issue
28	transferFrom	write	Passed	No Issue
29	_transfer	internal	Passed	No Issue
30	_update	internal	Passed	No Issue
31	_mint	internal	Passed	No Issue
32	_burn	internal	Passed	No Issue
33	_approve	internal	Passed	No Issue
34	_approve	internal	Passed	No Issue
35	_spendAllowance	internal	Passed	No Issue
36	permit	write	Passed	No Issue
37	nonces	read	Passed	No Issue
38	DOMAIN_SEPARATOR	external	Passed	No Issue
39	_useNonce	internal	Passed	No Issue

BnbxAsset.sol

Functions

Sl.	Functions	Type	Observation	Conclusion
1	constructor	write	Passed	No Issue
2	getRelativePrice	external	Passed	No Issue
3	getRelativePrice	external	Passed	No Issue

DynamicAsset.sol

Functions

Sl.	Functions	Type	Observation	Conclusion
1	constructor	write	Passed	No Issue
2	getRelativePrice	external	Passed	No Issue
3	onlyPool	modifier	Passed	No Issue

4	setPool	external	access only Owner	No Issue
5	setMaxSupply	external	access only Owner	No Issue
6	decimals	read	Passed	No Issue
7	underlyingTokenBalance	external	Passed	No Issue
8	transferUnderlyingToken	external	access only Pool	No Issue
9	mint	external	access only Pool	No Issue
10	burn	external	access only Pool	No Issue
11	addCash	external	access only Pool	No Issue
12	removeCash	external	access only Pool	No Issue
13	addLiability	external	access only Pool	No Issue
14	removeLiability	external	access only Pool	No Issue

StkbnbAsset.sol

Functions

Sl.	Functions	Type	Observation	Conclusion
1	constructor	write	Passed	No Issue
2	getRelativePrice	external	Passed	No Issue
3	getRelativePrice	external	Passed	No Issue

CoreV2.sol

Functions

Sl.	Functions	Type	Observation	Conclusion
1	constructor	write	Passed	No Issue
2	_swapQuoteFunc	internal	Passed	No Issue
3	_solveQuad	internal	Passed	No Issue
4	_invariantFunc	internal	Passed	No Issue
5	_coefficientFunc	internal	Passed	No Issue
6	depositRewardImpl	internal	Passed	No Issue
7	withdrawalAmountInEquilImpl	internal	Passed	No Issue
8	exactDepositLiquidityInEquilImpl	internal	Passed	No Issue
9	_targetedCovRatio	internal	Passed	No Issue
10	_equilCovRatio	internal	Passed	No Issue
11	_newEquilCovRatio	internal	Passed	No Issue
12	_newInvariantFunc	internal	Passed	No Issue
13	_haircut	internal	Passed	No Issue

DynamicPoolV2.sol

Functions

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

2	_quoteFactor	internal	Passed	No Issue
3	_globalInvariantFunc	internal	Passed	No Issue
4	initialize	write	Passed	No Issue
5	setCovRatioFeeParam	external	access only Owner	No Issue
6	_highCovRatioFee	internal	Passed	No Issue
7	_quoteFrom	internal	Passed	No Issue
8	_findUpperBound	internal	Passed	No Issue
9	quotePotentialWithdrawFromOtherAsset	external	Passed	No Issue

HighCovRatioFeePoolV2.sol

Functions

Sl.	Functions	Type	Observation	Conclusion
1	constructor	write	Passed	No Issue
2	initialize	write	Passed	No Issue
3	setCovRatioFeeParam	external	access only Owner	No Issue
4	_highCovRatioFee	internal	Passed	No Issue
5	_quoteFrom	internal	Passed	No Issue
6	_findUpperBound	internal	Passed	No Issue
7	quotePotentialWithdrawFromOtherAsset	external	Passed	No Issue
8	_checkLiquidity	internal	Passed	No Issue
9	_checkAddress	internal	Passed	No Issue
10	_checkSameAddress	internal	Passed	No Issue
11	_checkAmount	internal	Passed	No Issue
12	_ensure	internal	Passed	No Issue
13	_onlyDev	internal	Passed	No Issue
14	initialize	write	initializer	No Issue
15	addAsset	external	access only Owner	No Issue
16	removeAsset	external	access only Owner	No Issue
17	setDev	external	access only Owner	No Issue
18	setMasterWombat	external	access only Owner	No Issue
19	setAmpFactor	external	access only Owner	No Issue
20	setHaircutRate	external	access only Owner	No Issue
21	setFee	external	access only Owner	No Issue
22	transferTipBucket	external	access only Owner	No Issue
23	setFeeTo	external	access only Owner	No Issue
24	setMintFeeThreshold	external	access only Owner	No Issue
25	pause	external	Passed	No Issue
26	unpause	external	Passed	No Issue
27	pauseAsset	external	Passed	No Issue
28	unpauseAsset	external	Passed	No Issue
29	fillPool	external	Passed	No Issue
30	getTokens	external	Passed	No Issue
31	_sizeOfAssetList	internal	Passed	No Issue

32	_getAsset	internal	Passed	No Issue
33	_getKeyAtIndex	internal	Passed	No Issue
34	_containsAsset	internal	Passed	No Issue
35	_assetOf	internal	Passed	No Issue
36	addressOfAsset	external	Passed	No Issue
37	_exactDepositToInEquil	internal	Passed	No Issue
38	_deposit	internal	Passed	No Issue
39	deposit	external	Passed	No Issue
40	quotePotentialDeposit	external	Passed	No Issue
41	_withdrawFrom	internal	Passed	No Issue
42	_withdraw	internal	Passed	No Issue
43	withdraw	external	Passed	No Issue
44	withdrawFromOtherAsset	external	Passed	No Issue
45	quotePotentialWithdraw	external	Passed	No Issue
46	_quotePotentialWithdrawFrom OtherAsset	internal	Passed	No Issue
47	quotePotentialWithdrawFromO therAsset	external	Passed	No Issue
48	_quoteFactor	internal	Passed	No Issue
49	_quoteFrom	internal	Passed	No Issue
50	_swap	internal	Passed	No Issue
51	swap	external	Passed	No Issue
52	quotePotentialSwap	read	Passed	No Issue
53	quoteAmountIn	external	Passed	No Issue
54	exchangeRate	external	Passed	No Issue
55	globalEquilCovRatio	external	Passed	No Issue
56	tipBucketBalance	read	Passed	No Issue
57	_globalInvariantFunc	internal	Passed	No Issue
58	_mintFee	internal	Passed	No Issue
59	_mintAllFees	internal	Passed	No Issue
60	mintFee	external	Passed	No Issue

PoolV2.sol

Functions

Sl.	Functions	Type	Observation	Conclusion
1	constructor	write	Passed	No Issue
2	_checkLiquidity	internal	Passed	No Issue
3	_checkAddress	internal	Passed	No Issue
4	_checkSameAddress	internal	Passed	No Issue
5	_checkAmount	internal	Passed	No Issue
6	_ensure	internal	Passed	No Issue
7	_onlyDev	internal	Passed	No Issue
8	initialize	write	initializer	No Issue
9	addAsset	external	access only Owner	No Issue
10	removeAsset	external	access only Owner	No Issue
11	setDev	external	access only Owner	No Issue

12	setMasterWombat	external	access only Owner	No Issue
13	setAmpFactor	external	access only Owner	No Issue
14	setHaircutRate	external	access only Owner	No Issue
15	setFee	external	access only Owner	No Issue
16	transferTipBucket	external	access only Owner	No Issue
17	setFeeTo	external	access only Owner	No Issue
18	setMintFeeThreshold	external	access only Owner	No Issue
19	pause	external	Passed	No Issue
20	unpause	external	Passed	No Issue
21	pauseAsset	external	Passed	No Issue
22	unpauseAsset	external	Passed	No Issue
23	fillPool	external	Passed	No Issue
24	getTokens	external	Passed	No Issue
25	_sizeOfAssetList	internal	Passed	No Issue
26	_getAsset	internal	Passed	No Issue
27	_getKeyAtIndex	internal	Passed	No Issue
28	_containsAsset	internal	Passed	No Issue
29	_assetOf	internal	Passed	No Issue
30	addressOfAsset	external	Passed	No Issue
31	_exactDepositToInEquil	internal	Passed	No Issue
32	_deposit	internal	Passed	No Issue
33	deposit	external	Passed	No Issue
34	quotePotentialDeposit	external	Passed	No Issue
35	_withdrawFrom	internal	Passed	No Issue
36	_withdraw	internal	Passed	No Issue
37	withdraw	external	Passed	No Issue
38	withdrawFromOtherAsset	external	Passed	No Issue
39	quotePotentialWithdraw	external	Passed	No Issue
40	_quotePotentialWithdrawFromOtherAsset	internal	Passed	No Issue
41	quotePotentialWithdrawFromOtherAsset	external	Passed	No Issue
42	_quoteFactor	internal	Passed	No Issue
43	_quoteFrom	internal	Passed	No Issue
44	_swap	internal	Passed	No Issue
45	swap	external	Passed	No Issue
46	quotePotentialSwap	read	Passed	No Issue
47	quoteAmountIn	external	Passed	No Issue
48	exchangeRate	external	Passed	No Issue
49	globalEquilCovRatio	external	Passed	No Issue
50	tipBucketBalance	read	Passed	No Issue
51	_globalInvariantFunc	internal	Passed	No Issue
52	_mintFee	internal	Passed	No Issue
53	_mintAllFees	internal	Passed	No Issue
54	mintFee	external	Passed	No Issue
55	initializer	modifier	Passed	No Issue
56	reinitializer	modifier	Passed	No Issue
57	onlyInitializing	modifier	Passed	No Issue

58	_disableInitializers	internal	Passed	No Issue
59	__Ownable_init	internal	access only Initializing	No Issue
60	__Ownable_init_unchained	internal	access only Initializing	No Issue
61	onlyOwner	modifier	Passed	No Issue
62	owner	read	Passed	No Issue
63	_checkOwner	internal	Passed	No Issue
64	renounceOwnership	write	access only Owner	No Issue
65	transferOwnership	write	access only Owner	No Issue
66	_transferOwnership	internal	Passed	No Issue
67	__ReentrancyGuard_init	internal	access only Initializing	No Issue
68	__ReentrancyGuard_init_unchained	internal	access only Initializing	No Issue
69	nonReentrant	modifier	Passed	No Issue
70	_nonReentrantBefore	write	Passed	No Issue
71	_nonReentrantAfter	write	Passed	No Issue
72	_reentrancyGuardEntered	internal	Passed	No Issue
73	__Pausable_init	internal	access only Initializing	No Issue
74	__Pausable_init_unchained	internal	access only Initializing	No Issue
75	whenNotPaused	modifier	Passed	No Issue
76	whenPaused	modifier	Passed	No Issue
77	paused	read	Passed	No Issue
78	_requireNotPaused	internal	Passed	No Issue
79	_requirePaused	internal	Passed	No Issue
80	_pause	internal	Passed	No Issue
81	_unpause	internal	Passed	No Issue
82	requireAssetNotPaused	internal	Passed	No Issue
83	requireAssetPaused	internal	Passed	No Issue
84	_pauseAsset	internal	Passed	No Issue
85	_unpauseAsset	internal	Passed	No Issue
86	_swapQuoteFunc	internal	Passed	No Issue
87	_solveQuad	internal	Passed	No Issue
88	_invariantFunc	internal	Passed	No Issue
89	_coefficientFunc	internal	Passed	No Issue
90	depositRewardImpl	internal	Passed	No Issue
91	withdrawalAmountInEquilImpl	internal	Passed	No Issue
92	exactDepositLiquidityInEquilImpl	internal	Passed	No Issue
93	_targetedCovRatio	internal	Passed	No Issue
94	_equilCovRatio	internal	Passed	No Issue
95	_newEquilCovRatio	internal	Passed	No Issue
96	_newInvariantFunc	internal	Passed	No Issue
97	_haircut	internal	Passed	No Issue

PausableAssets.sol

Functions

Sl.	Functions	Type	Observation	Conclusion
1	constructor	write	Passed	No Issue
2	requireAssetNotPaused	internal	Passed	No Issue
3	requireAssetPaused	internal	Passed	No Issue
4	_pauseAsset	internal	Passed	No Issue
5	_unpauseAsset	internal	Passed	No Issue

MasterWombatV4.sol

Functions

Sl.	Functions	Type	Observation	Conclusion
1	constructor	write	Passed	No Issue
2	initialize	external	initializer	No Issue
3	pause	external	access only Owner	No Issue
4	unpause	external	access only Owner	No Issue
5	add	external	access only Owner	No Issue
6	setRewarder	external	access only Owner	No Issue
7	massUpdatePools	write	Infinite Loop	Refer Audit Findings
8	updatePool	external	Passed	No Issue
9	_updatePool	write	Passed	No Issue
10	depositFor	external	Passed	No Issue
11	_updateAdjustedAllocPoint	internal	Passed	No Issue
12	poolAdjustFactor	external	Passed	No Issue
13	_assetAdjustFactor	internal	Passed	No Issue
14	_adjustFactor	internal	Passed	No Issue
15	deposit	external	Passed	No Issue
16	multiClaim	external	Infinite Loop	Refer Audit Findings
17	_multiClaim	write	Passed	No Issue
18	withdraw	external	Passed	No Issue
19	_updateUserAmount	internal	Passed	No Issue
20	updateEmissionPartition	external	access only Owner	No Issue
21	rewarderBonusTokenInfo	write	Passed	No Issue
22	boostedPartition	external	Passed	No Issue
23	poolLength	external	Passed	No Issue
24	getAssetPid	external	Passed	No Issue
25	lastTimeRewardApplicable	read	Passed	No Issue
26	calRewardPerUnit	read	Passed	No Issue
27	pendingTokens	external	Passed	No Issue
28	poolInfo	external	Passed	No Issue
29	initializer	modifier	Passed	No Issue

30	reinitializer	modifier	Passed	No Issue
31	onlyInitializing	modifier	Passed	No Issue
32	_disableInitializers	internal	Passed	No Issue
33	__Ownable_init	internal	access only Initializing	No Issue
34	__Ownable_init_unchained	internal	access only Initializing	No Issue
35	onlyOwner	modifier	Passed	No Issue
36	owner	read	Passed	No Issue
37	_checkOwner	internal	Passed	No Issue
38	renounceOwnership	write	access only Owner	No Issue
39	transferOwnership	write	access only Owner	No Issue
40	_transferOwnership	internal	Passed	No Issue
41	__ReentrancyGuard_init	internal	access only Initializing	No Issue
42	__ReentrancyGuard_init_unchained	internal	access only Initializing	No Issue
43	nonReentrant	modifier	Passed	No Issue
44	_nonReentrantBefore	write	Passed	No Issue
45	_nonReentrantAfter	write	Passed	No Issue
46	_reentrancyGuardEntered	internal	Passed	No Issue
47	__Pausable_init	internal	access only Initializing	No Issue
48	__Pausable_init_unchained	internal	access only Initializing	No Issue
49	whenNotPaused	modifier	Passed	No Issue
50	whenPaused	modifier	Passed	No Issue
51	paused	read	Passed	No Issue
52	_requireNotPaused	internal	Passed	No Issue
53	_requirePaused	internal	Passed	No Issue
54	_pause	internal	Passed	No Issue
55	_unpause	internal	Passed	No Issue

MultiRewarderPerSec.sol

Functions

Sl.	Functions	Type	Observation	Conclusion
1	constructor	write	Passed	No Issue
2	onlyMaster	modifier	Passed	No Issue
3	onlyOperatorOrOwner	modifier	Passed	No Issue
4	receive	external	Passed	No Issue
5	setOperator	external	access only Owner	No Issue
6	addRewardToken	external	access only Owner	No Issue

7	updateReward	write	Passed	No Issue
8	_updateReward	internal	Passed	No Issue
9	_updateReward	internal	Passed	No Issue
10	setRewardRate	external	access only Operator Or Owner	No Issue
11	onReward	external	access only Master	No Issue
12	_onReward	internal	Passed	No Issue
13	rewardLength	external	Passed	No Issue
14	_rewardLength	internal	Passed	No Issue
15	pendingTokens	external	Passed	No Issue
16	_pendingTokens	internal	Passed	No Issue
17	_getTotalShare	internal	Passed	No Issue
18	_rewardTokens	internal	Passed	No Issue
19	rewardTokens	external	Passed	No Issue
20	emergencyWithdraw	external	access only Owner	No Issue
21	emergencyTokenWithdraw	write	access only Owner	No Issue
22	balances	external	Passed	No Issue
23	toUint128	internal	Passed	No Issue
24	max	internal	Passed	No Issue
25	onlyOwner	modifier	Passed	No Issue
26	owner	read	Passed	No Issue
27	_checkOwner	internal	Passed	No Issue
28	renounceOwnership	write	access only Owner	No Issue
29	transferOwnership	write	access only Owner	No Issue
30	_transferOwnership	internal	Passed	No Issue
31	nonReentrant	modifier	Passed	No Issue
32	_nonReentrantBefore	write	Passed	No Issue
33	_nonReentrantAfter	write	Passed	No Issue
34	_reentrancyGuardEntered	internal	Passed	No Issue

DynamicPool.sol

Functions

Sl.	Functions	Type	Observation	Conclusion
1	constructor	write	Passed	No Issue
2	_quoteFactor	internal	Passed	No Issue
3	_globalInvariantFunc	internal	Passed	No Issue
4	_checkLiquidity	internal	Passed	No Issue
5	_checkAddress	internal	Passed	No Issue
6	_checkSameAddress	internal	Passed	No Issue
7	_checkAmount	internal	Passed	No Issue

8	_ensure	internal	Passed	No Issue
9	_onlyDev	internal	Passed	No Issue
10	initialize	write	initializer	No Issue
11	pause	external	Passed	No Issue
12	unpause	external	Passed	No Issue
13	pauseAsset	external	Passed	No Issue
14	unpauseAsset	external	Passed	No Issue
15	setDev	external	access only Owner	No Issue
16	setMasterWombat	external	access only Owner	No Issue
17	setAmpFactor	external	access only Owner	No Issue
18	setHaircutRate	external	access only Owner	No Issue
19	setFee	external	access only Owner	No Issue
20	setFeeTo	external	access only Owner	No Issue
21	setMintFeeThreshold	external	access only Owner	No Issue
22	addAsset	external	access only Owner	No Issue
23	removeAsset	external	access only Owner	No Issue
24	getTokens	external	Passed	No Issue
25	_sizeOfAssetList	internal	Passed	No Issue
26	_getAsset	internal	Passed	No Issue
27	_getKeyAtIndex	internal	Passed	No Issue
28	_containsAsset	internal	Passed	No Issue
29	_assetOf	internal	Passed	No Issue
30	addressOfAsset	external	Passed	No Issue
31	_exactDepositToInEquil	internal	Passed	No Issue
32	_deposit	internal	Passed	No Issue
33	deposit	external	Passed	No Issue
34	quotePotentialDeposit	external	Passed	No Issue
35	_withdrawFrom	internal	Passed	No Issue
36	_withdraw	internal	Passed	No Issue
37	withdraw	external	Passed	No Issue
38	withdrawFromOtherAsset	external	Passed	No Issue
39	quotePotentialWithdraw	external	Passed	No Issue
40	_quotePotentialWithdrawFromOtherAsset	internal	Passed	No Issue
41	quotePotentialWithdrawFromOtherAsset	external	Passed	No Issue
42	_quoteFactor	internal	Passed	No Issue
43	_quoteFrom	internal	Passed	No Issue
44	_swap	internal	Passed	No Issue

45	swap	external	Passed	No Issue
46	quotePotentialSwap	read	Passed	No Issue
47	quoteAmountIn	external	Passed	No Issue
48	exchangeRate	external	Passed	No Issue
49	globalEquilCovRatio	external	Passed	No Issue
50	tipBucketBalance	read	Passed	No Issue
51	fillPool	external	Passed	No Issue
52	transferTipBucket	external	access only Owner	No Issue
53	_globalInvariantFunc	internal	Passed	No Issue
54	_mintFee	internal	Passed	No Issue
55	mintAllFee	internal	Passed	No Issue
56	mintFee	external	Passed	No Issue

Pool.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	__Ownable_init	internal	access only Initializing	No Issue
7	__Ownable_init_unchained	internal	access only Initializing	No Issue
8	onlyOwner	modifier	Passed	No Issue
9	owner	read	Passed	No Issue
10	_checkOwner	internal	Passed	No Issue
11	renounceOwnership	write	access only Owner	No Issue
12	transferOwnership	write	access only Owner	No Issue
13	_transferOwnership	internal	Passed	No Issue
14	__ReentrancyGuard_init	internal	access only Initializing	No Issue
15	__ReentrancyGuard_init_unchained	internal	access only Initializing	No Issue
16	nonReentrant	modifier	Passed	No Issue
17	_nonReentrantBefore	write	Passed	No Issue
18	_nonReentrantAfter	write	Passed	No Issue
19	_reentrancyGuardEntered	internal	Passed	No Issue
20	__Pausable_init	internal	access only Initializing	No Issue
21	__Pausable_init_unchained	internal	access only Initializing	No Issue

22	whenNotPaused	modifier	Passed	No Issue
23	whenPaused	modifier	Passed	No Issue
24	paused	read	Passed	No Issue
25	_requireNotPaused	internal	Passed	No Issue
26	_requirePaused	internal	Passed	No Issue
27	_pause	internal	Passed	No Issue
28	_unpause	internal	Passed	No Issue
29	requireAssetNotPaused	internal	Passed	No Issue
30	requireAssetPaused	internal	Passed	No Issue
31	_pauseAsset	internal	Passed	No Issue
32	_unpauseAsset	internal	Passed	No Issue
33	_swapQuoteFunc	internal	Passed	No Issue
34	_solveQuad	internal	Passed	No Issue
35	_invariantFunc	internal	Passed	No Issue
36	_coefficientFunc	internal	Passed	No Issue
37	depositRewardImpl	internal	Passed	No Issue
38	withdrawalAmountInEquilImpl	internal	Passed	No Issue
39	exactDepositLiquidityInEquilImpl	internal	Passed	No Issue
40	_targetedCovRatio	internal	Passed	No Issue
41	_equilCovRatio	internal	Passed	No Issue
42	_newEquilCovRatio	internal	Passed	No Issue
43	_newInvariantFunc	internal	Passed	No Issue
44	_haircut	internal	Passed	No Issue
45	_checkLiquidity	internal	Passed	No Issue
46	_checkAddress	internal	Passed	No Issue
47	_checkSameAddress	internal	Passed	No Issue
48	_checkAmount	internal	Passed	No Issue
49	_ensure	internal	Passed	No Issue
50	_onlyDev	internal	Passed	No Issue
51	initialize	write	initializer	No Issue
52	pause	external	Passed	No Issue
53	unpause	external	Passed	No Issue
54	pauseAsset	external	Passed	No Issue
55	unpauseAsset	external	Passed	No Issue
56	setDev	external	access only Owner	No Issue
57	setMasterWombat	external	access only Owner	No Issue
58	setAmpFactor	external	access only Owner	No Issue
59	setHaircutRate	external	access only Owner	No Issue
60	setFee	external	access only Owner	No Issue
61	setFeeTo	external	access only Owner	No Issue
62	setMintFeeThreshold	external	The value limit is not set	Refer Audit Findings

63	addAsset	external	access only Owner	No Issue
64	removeAsset	external	access only Owner	No Issue
65	getTokens	external	Passed	No Issue
66	_sizeOfAssetList	internal	Passed	No Issue
67	_getAsset	internal	Passed	No Issue
68	_getKeyAtIndex	internal	Passed	No Issue
69	_containsAsset	internal	Passed	No Issue
70	_assetOf	internal	Passed	No Issue
71	addressOfAsset	external	Passed	No Issue
72	_exactDepositToInEquil	internal	Passed	No Issue
73	_deposit	internal	Passed	No Issue
74	deposit	external	Function input parameters lack of check	Refer Audit Findings
75	quotePotentialDeposit	external	Passed	No Issue
76	_withdrawFrom	internal	Passed	No Issue
77	_withdraw	internal	Passed	No Issue
78	withdraw	external	Function input parameters lack of check	Refer Audit Findings
79	withdrawFromOtherAsset	external	Passed	No Issue
80	quotePotentialWithdraw	external	Passed	No Issue
81	_quotePotentialWithdrawFromOtherAsset	internal	Passed	No Issue
82	quotePotentialWithdrawFromOtherAsset	external	Passed	No Issue
83	_quoteFactor	internal	Passed	No Issue
84	_quoteFrom	internal	Passed	No Issue
85	_swap	internal	Passed	No Issue
86	swap	external	Passed	No Issue
87	quotePotentialSwap	read	Passed	No Issue
88	quoteAmountIn	external	Passed	No Issue
89	exchangeRate	external	Passed	No Issue
90	globalEquilCovRatio	external	Passed	No Issue
91	tipBucketBalance	read	Passed	No Issue
92	fillPool	external	Passed	No Issue
93	transferTipBucket	external	access only Owner	No Issue
94	_globalInvariantFunc	internal	Passed	No Issue
95	_mintFee	internal	Passed	No Issue
96	mintAllFee	internal	Passed	No Issue
97	mintFee	external	Passed	No Issue

HighCovRatioFeePool.sol

Functions

Sl.	Functions	Type	Observation	Conclusion
1	constructor	write	Passed	No Issue
2	initialize	write	Passed	No Issue
3	setCovRatioFeeParam	external	ow	No Issue
4	_highCovRatioFee	internal	Passed	No Issue
5	_quoteFrom	internal	Passed	No Issue
6	_findUpperBound	internal	Passed	No Issue
7	quotePotentialWithdrawFromOtherAsset	external	Passed	No Issue
8	_checkLiquidity	internal	Passed	No Issue
9	_checkAddress	internal	Passed	No Issue
10	_checkSameAddress	internal	Passed	No Issue
11	_checkAmount	internal	Passed	No Issue
12	_ensure	internal	Passed	No Issue
13	_onlyDev	internal	Passed	No Issue
14	initialize	write	initializer	No Issue
15	addAsset	external	access only Owner	No Issue
16	removeAsset	external	access only Owner	No Issue
17	setDev	external	access only Owner	No Issue
18	setMasterWombat	external	access only Owner	No Issue
19	setAmpFactor	external	access only Owner	No Issue
20	setHaircutRate	external	access only Owner	No Issue
21	setFee	external	access only Owner	No Issue
22	transferTipBucket	external	access only Owner	No Issue
23	setFeeTo	external	access only Owner	No Issue
24	setMintFeeThreshold	external	access only Owner	No Issue
25	pause	external	Passed	No Issue
26	unpause	external	Passed	No Issue
27	pauseAsset	external	Passed	No Issue
28	unpauseAsset	external	Passed	No Issue
29	fillPool	external	Passed	No Issue
30	getTokens	external	Passed	No Issue
31	_sizeOfAssetList	internal	Passed	No Issue
32	_getAsset	internal	Passed	No Issue
33	_getKeyAtIndex	internal	Passed	No Issue
34	_containsAsset	internal	Passed	No Issue
35	_assetOf	internal	Passed	No Issue
36	addressOfAsset	external	Passed	No Issue

37	_exactDepositToInEquil	internal	Passed	No Issue
38	_deposit	internal	Passed	No Issue
39	deposit	external	Passed	No Issue
40	quotePotentialDeposit	external	Passed	No Issue
41	_withdrawFrom	internal	Passed	No Issue
42	_withdraw	internal	Passed	No Issue
43	withdraw	external	Passed	No Issue
44	withdrawFromOtherAsset	external	Passed	No Issue
45	quotePotentialWithdraw	external	Passed	No Issue
46	_quotePotentialWithdrawFromOtherAsset	internal	Passed	No Issue
47	quotePotentialWithdrawFromOtherAsset	external	Passed	No Issue
48	_quoteFactor	internal	Passed	No Issue
49	_quoteFrom	internal	Passed	No Issue
50	_swap	internal	Passed	No Issue
51	swap	external	Passed	No Issue
52	quotePotentialSwap	read	Passed	No Issue
53	quoteAmountIn	external	Passed	No Issue
54	exchangeRate	external	Passed	No Issue
55	globalEquilCovRatio	external	Passed	No Issue
56	tipBucketBalance	read	Passed	No Issue
57	_globalInvariantFunc	internal	Passed	No Issue
58	_mintFee	internal	Passed	No Issue
59	_mintAllFees	internal	Passed	No Issue
60	mintFee	external	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.

Email: audit@EtherAuthority.io

Audit Findings

Critical Severity

No critical severity vulnerabilities were found in the contract code.

High Severity

No high severity vulnerabilities were found in the contract code.

Medium

No medium severity vulnerabilities were found in the contract code.

Low

(1) The value limit is not set:
In the listed functions, variable values can be set by any number. An explicit range limit should be set for each variable according to its use in calculations.

Pool.sol

- setMintFeeThreshold

Asset.sol

- setMaxSupply

Resolution: Consider adding an explicit value to the values.

(2) Function input parameters lack of check: **Pool.sol**
Variable validation is not performed in below functions:

- withdraw = token
- deposit = token

Resolution: We suggest putting validation: integer type variables should not be empty and greater than 0, and address type variables should not be address(0).

Very Low / Informational / Best practices:

(1) Infinite Loop: **MasterWombatV4.sol**
The massUpdatePools, _multiClaim for loop _pids.length must have some limit set to save gas.

Resolution: The upper bound should have a certain limit for loops.

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:

Asset.sol

- setPool: The pool address can be set by the owner.
- setMaxSupply: The maximum supply value can be set by the owner.
- transferUnderlyingToken:Transfers of the ERC20 underlying token from this contract to another account can only be called by Pool.
- mint: Mint ERC20 Asset LP Token: expect pool coordinates for other state updates. can only be called by Pool.
- burn:Burn ERC20 Asset LP Token: expect pool coordinates for other state updates. Can only be called by Pool.
- addCash: Add cash; assumes the underlying ERC20 token has been transferred in. Can only be called by Pool.
- removeCash:Deducts cash, except actual ERC20 got transferred out. Can only be called by Pool.
- addLiability: Adds deposit or dividend; expect LP underlying token minted in case of deposit. Can only be called by Pool.
- removeLiability: Removes deposit and dividend earned, except LP underlying token burned. Can only be called by Pool.

HighCovRatioFeePool.sol

- setCovRatioFeeParam: CovRatio fee parameter value can be set by the owner.

HighCovRatioFeePoolV2.sol

- setCovRatioFeeParam: CovRatio fee parameter value can be set by the owner.

Pool.sol

- setDev: The dev address can be set by the owner.
- setMasterWombat: The Master Wombat address can be set by the owner.
- setAmpFactor: The pool's amplification factor value can be set by the owner.
- setHaircutRate: The pool's haircut rate value can be set by the owner.
- setFee: LP Dividend ratio and retention The ratio value can be set by the owner.
- setFeeTo: The fee beneficiary address can be set by the owner.
- setMintFeeThreshold: The minimum fee for the mint value can be set by the owner.
- addAsset: Adds assets to the pool value, which can be set by the owner.
- removeAsset: Removes an asset from the asset structure by the owner.
- transferTipBucket: Transfer Tip Bucket by the owner.
- pause: Pause pool by the dev.
- unpause: Unpause pool by the developer.
- pauseAsset: Pause asset, restricting deposit and swap operations by the developer.
- unpauseAsset: Unpause asset, restricting deposit and swap operations by the developer.
- fillPool: Move funds from the tip bucket to the pool by the developer.

PoolV2.sol

- addAsset: Adds assets to the pool value, which can be set by the owner.
- removeAsset: Removes an asset from the asset structure by the owner.
- setDev: The dev address can be set by the owner.
- setMasterWombat: The Master Wombat address can be set by the owner.
- setAmpFactor: The pool's amplification factor value can be set by the owner.
- setHaircutRate: The pool's haircut rate value can be set by the owner.
- setFee: LP Dividend ratio and retention The ratio value can be set by the owner.
- transferTipBucket: Transfer Tip Bucket by the owner.
- setFeeTo: The fee beneficiary address can be set by the owner.

- setMintFeeThreshold: The minimum fee for the mint value can be set by the owner.
- pause: Pause pool by the dev.
- unpause: Unpause pool by the developer.
- pauseAsset: Pause asset, restricting deposit and swap operations by the developer.
- unpauseAsset: Unpause asset, restricting deposit and swap operations by the developer.
- fillPool: Move funds from the tip bucket to the pool by the developer.

WombatRouter.sol

- approveSpendingByPool: Approve spending of router tokens by pool by the owner.

MasterWombatV4.sol

- pause: Pause pool, restricting certain operations by the owner.
- unpause: Unpause pool, enabling certain operations by the owner.
- add: Add a new LP to the pool by the owner.
- setRewarder: Update the given pool's reward by the owner.
- updateEmissionPartition: The emission partition value can be updated by the owner.

MultiRewarderPerSec.sol

- setOperator: The operator address can be set by the owner.
- addRewardToken: A reward token can be set by the owner.
- setRewardRate: Sets the distribution reward rate set by the owner or operator.
- onReward: onReward by the Master owner.
- emergencyWithdraw: Emergency withdrawal by the owner.
- emergencyTokenWithdraw: Emergency token withdrawal by the owner.

OwnableUpgradeable.sol

- renounceOwnership: Deleting ownership will leave the contract without an owner, removing any owner-only functionality.
- transferOwnership: Current owner can transfer ownership of the contract to a new account.

Ownable.sol

- renounceOwnership: Deleting ownership will leave the contract without an owner, removing any owner-only functionality.
- transferOwnership: Current owner can transfer ownership of the contract to a new account.

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

Conclusion

We were given a contract code in the form of github.com web link. And we have used all possible tests based on given objects as files. We had observed 2 low and 1 informational severity issues in the smart contracts. but those are not critical ones. **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.

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

The 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.

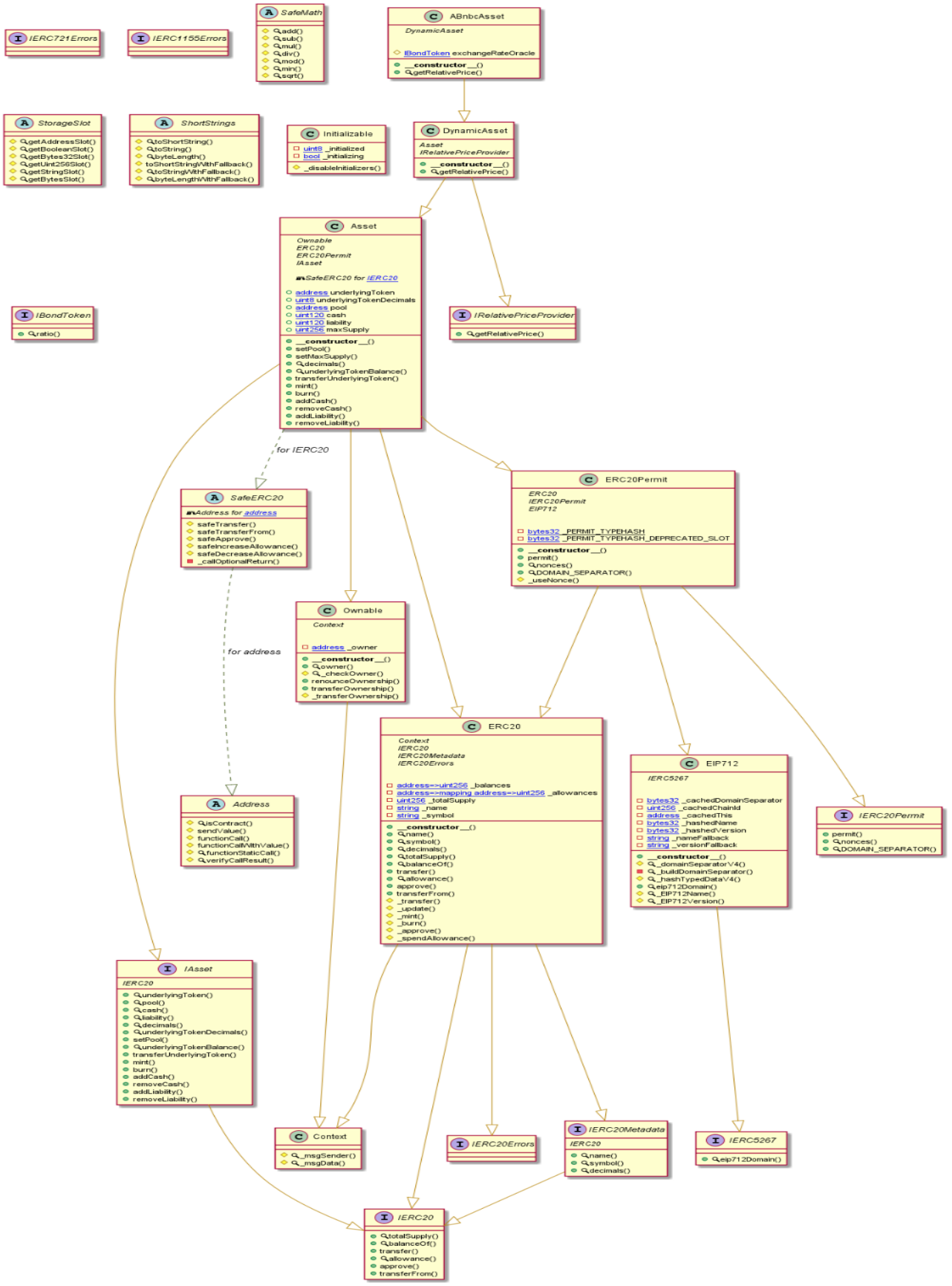
Technical Disclaimer

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

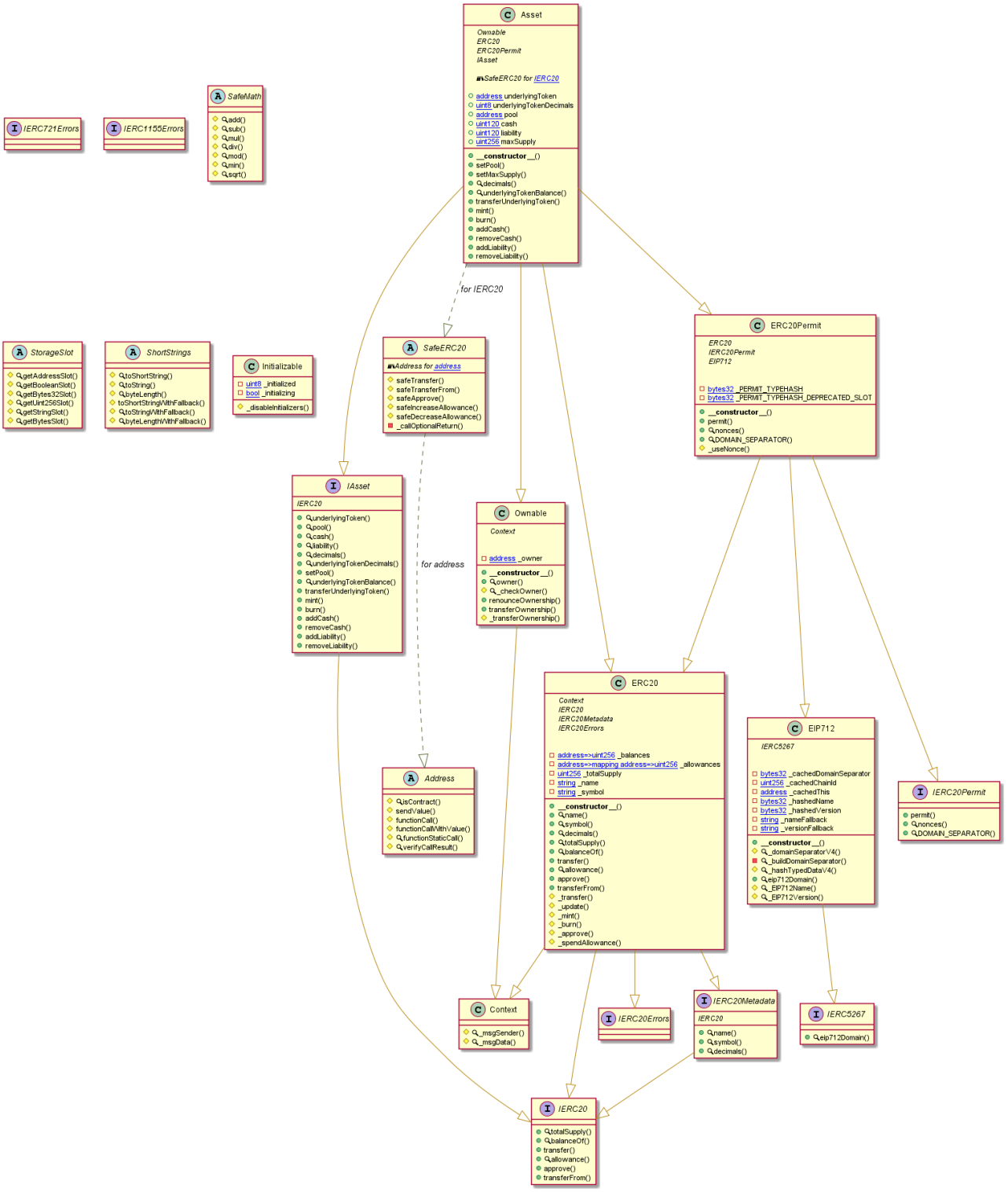
Code Flow Diagram - Bass Exchange Protocol

ABnbcAsset Diagram

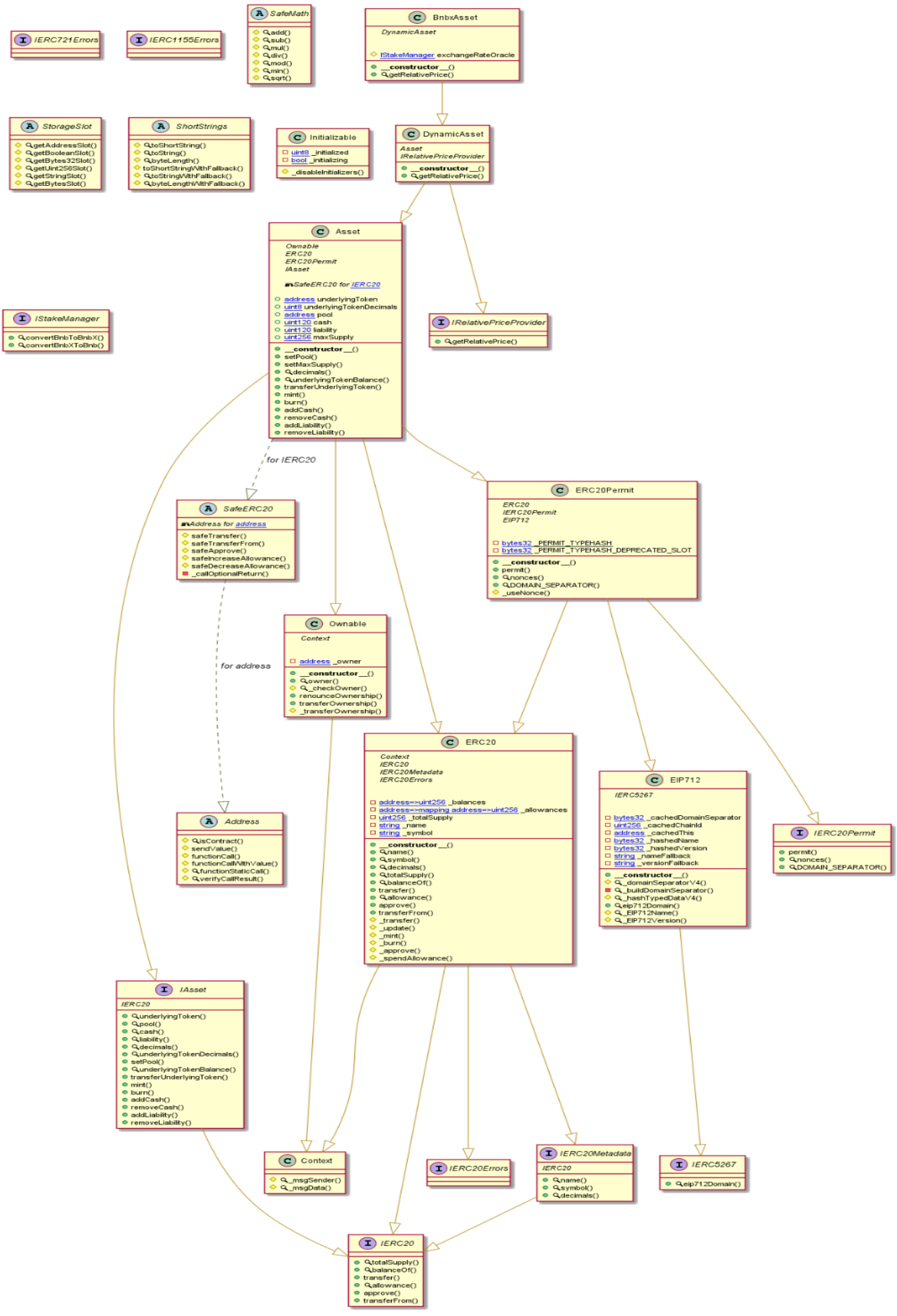


Email: audit@EtherAuthority.io

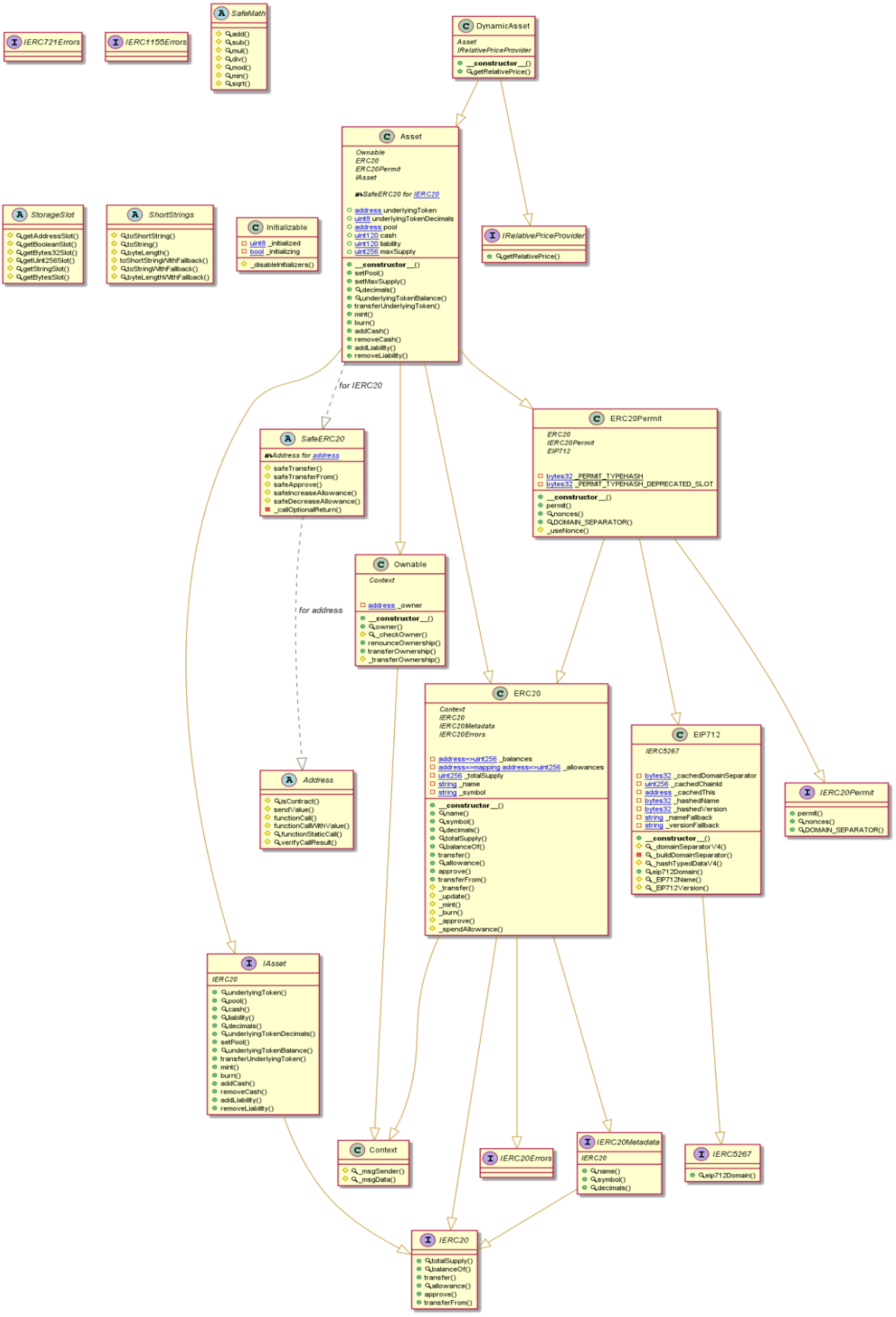
Asset Diagram



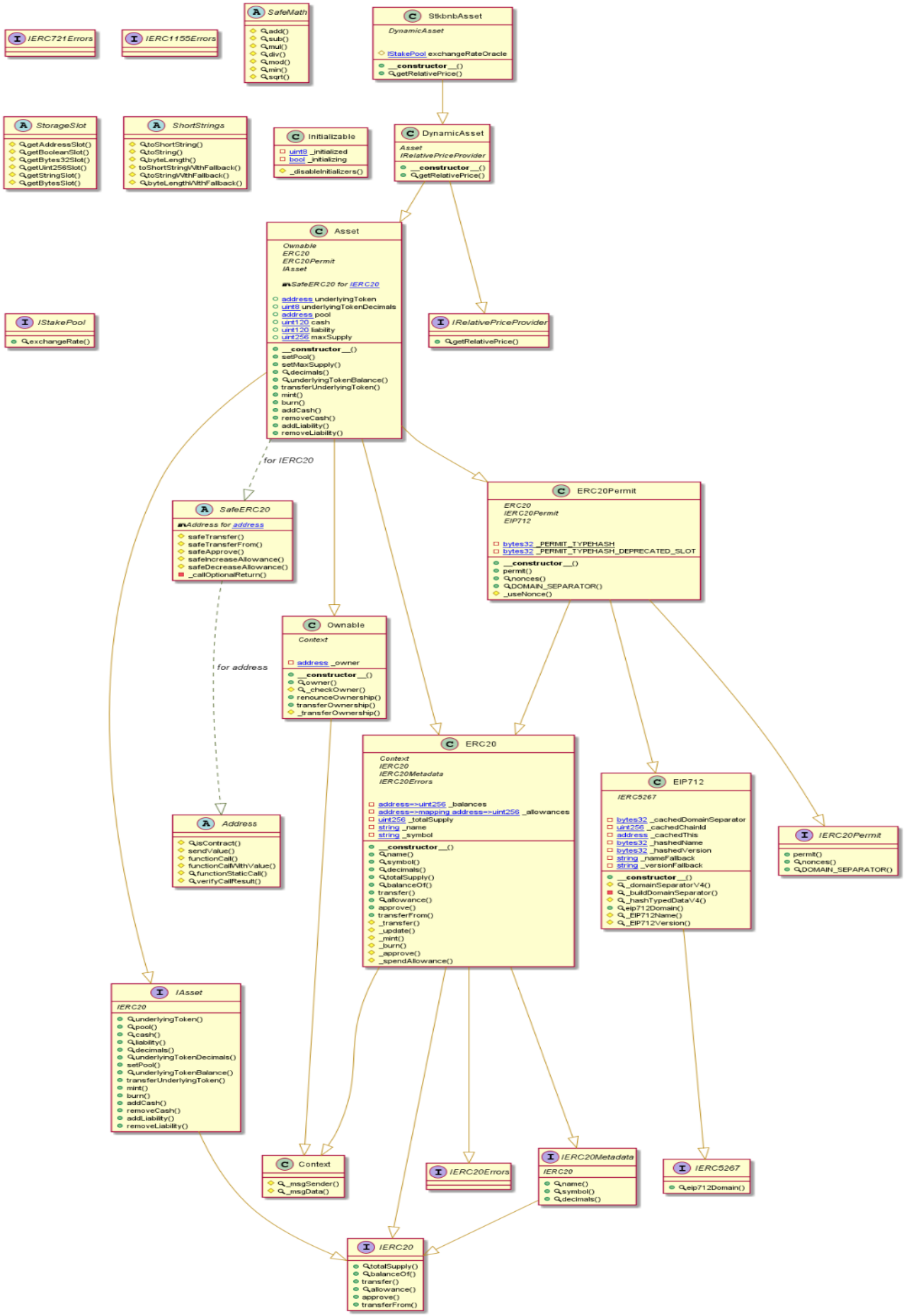
BnbxAsset Diagram



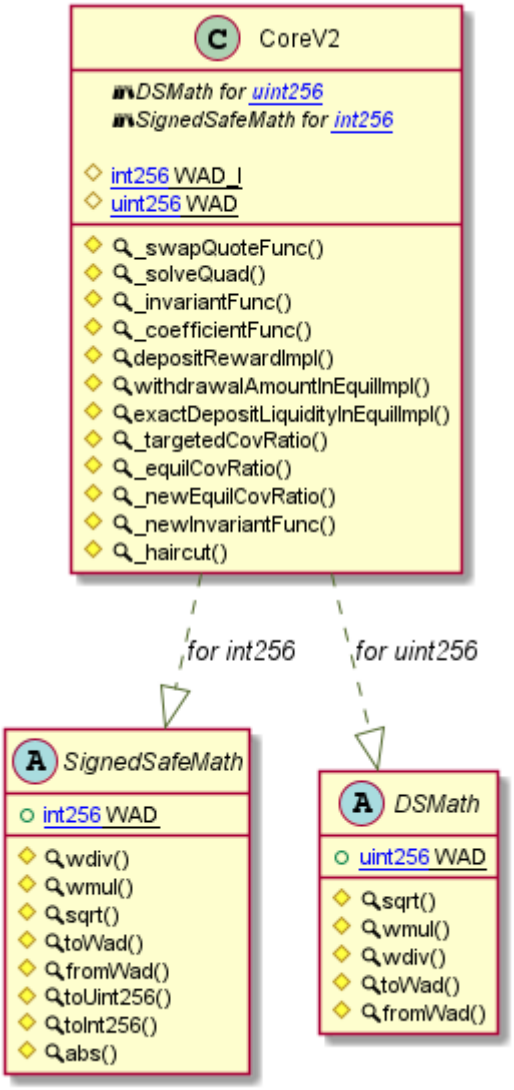
DynamicAsset Diagram



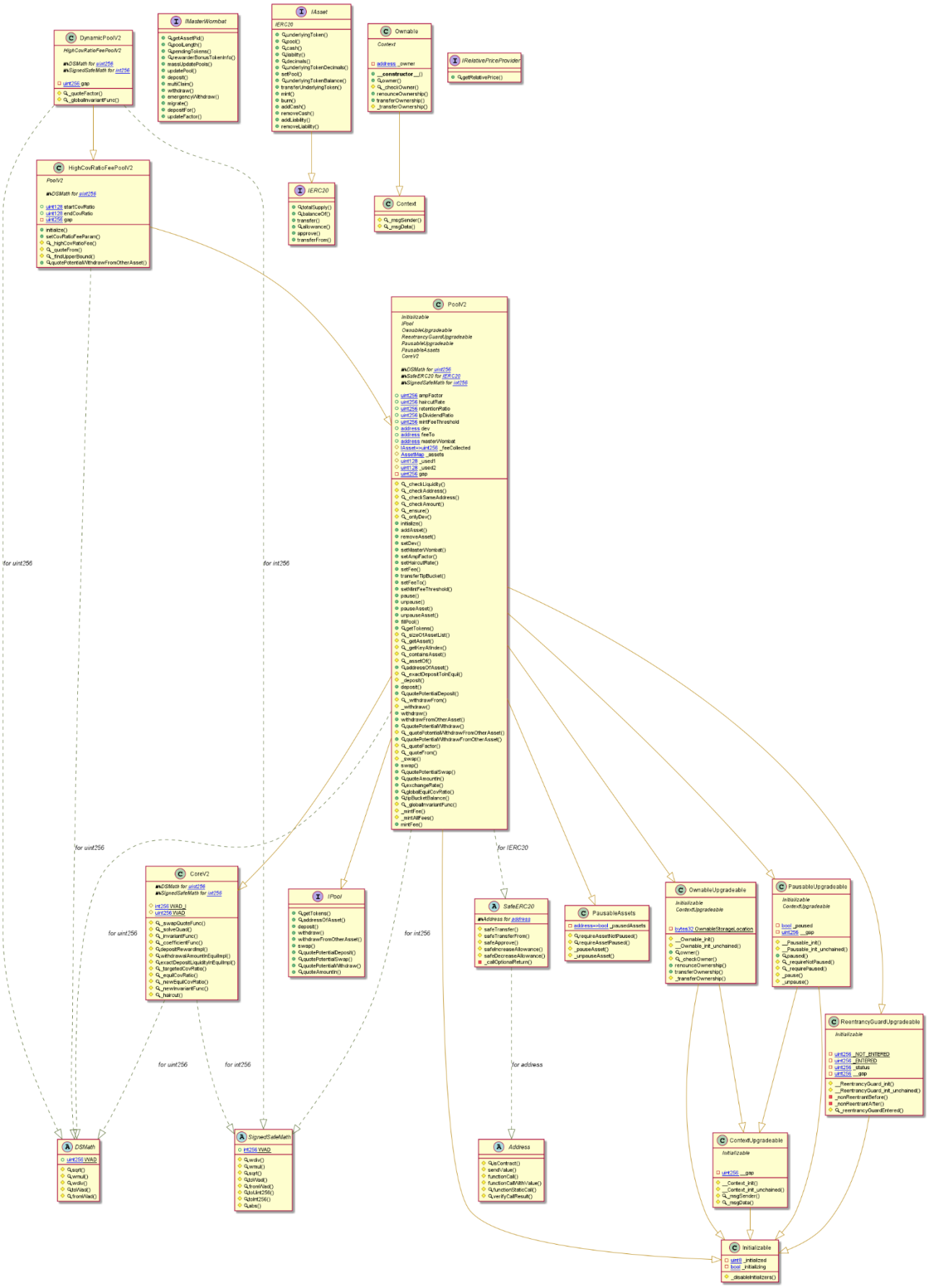
StkbnbAsset Diagram



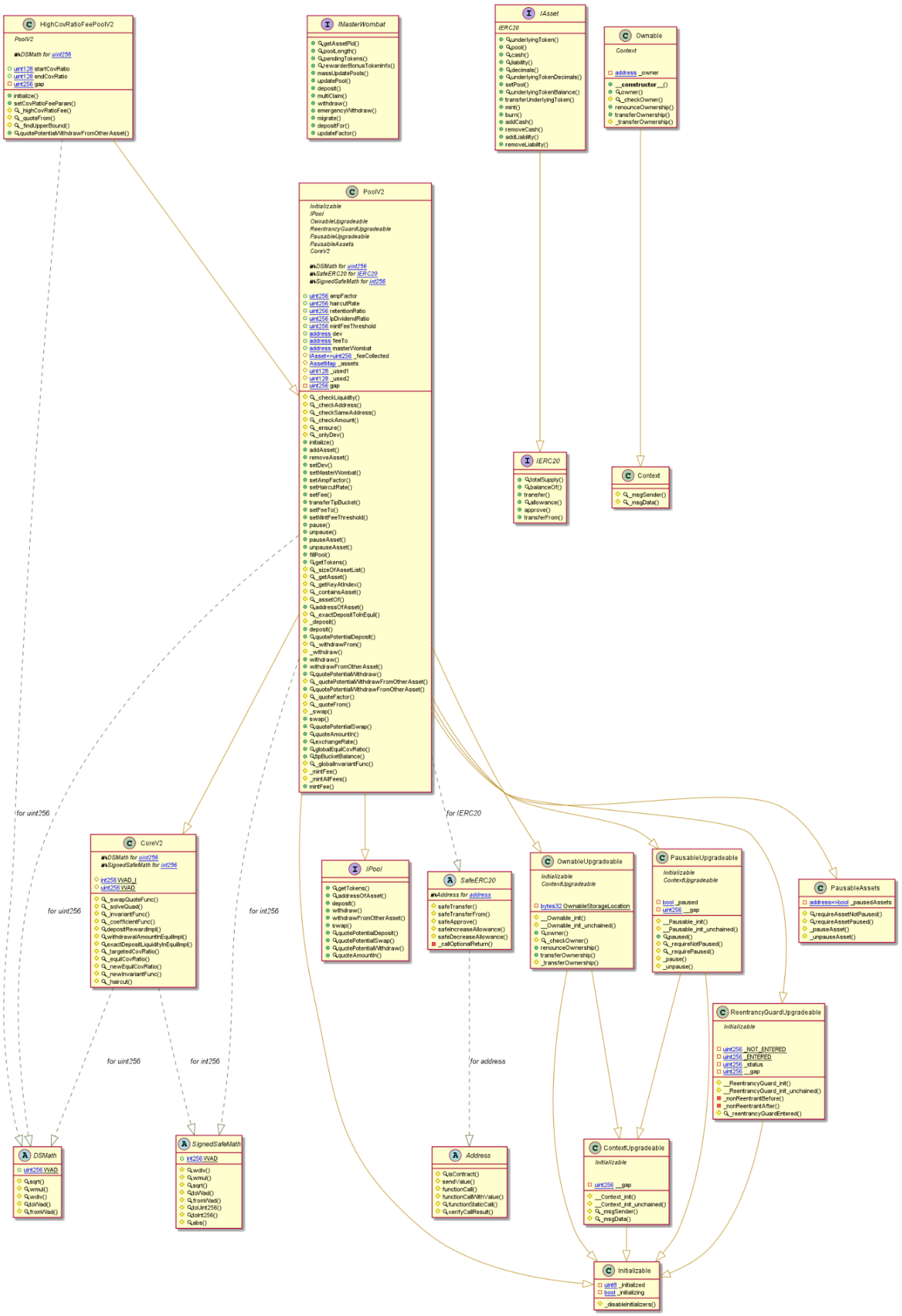
CoreV2 Diagram



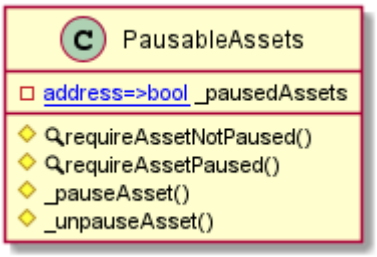
DynamicPoolV2 Diagram



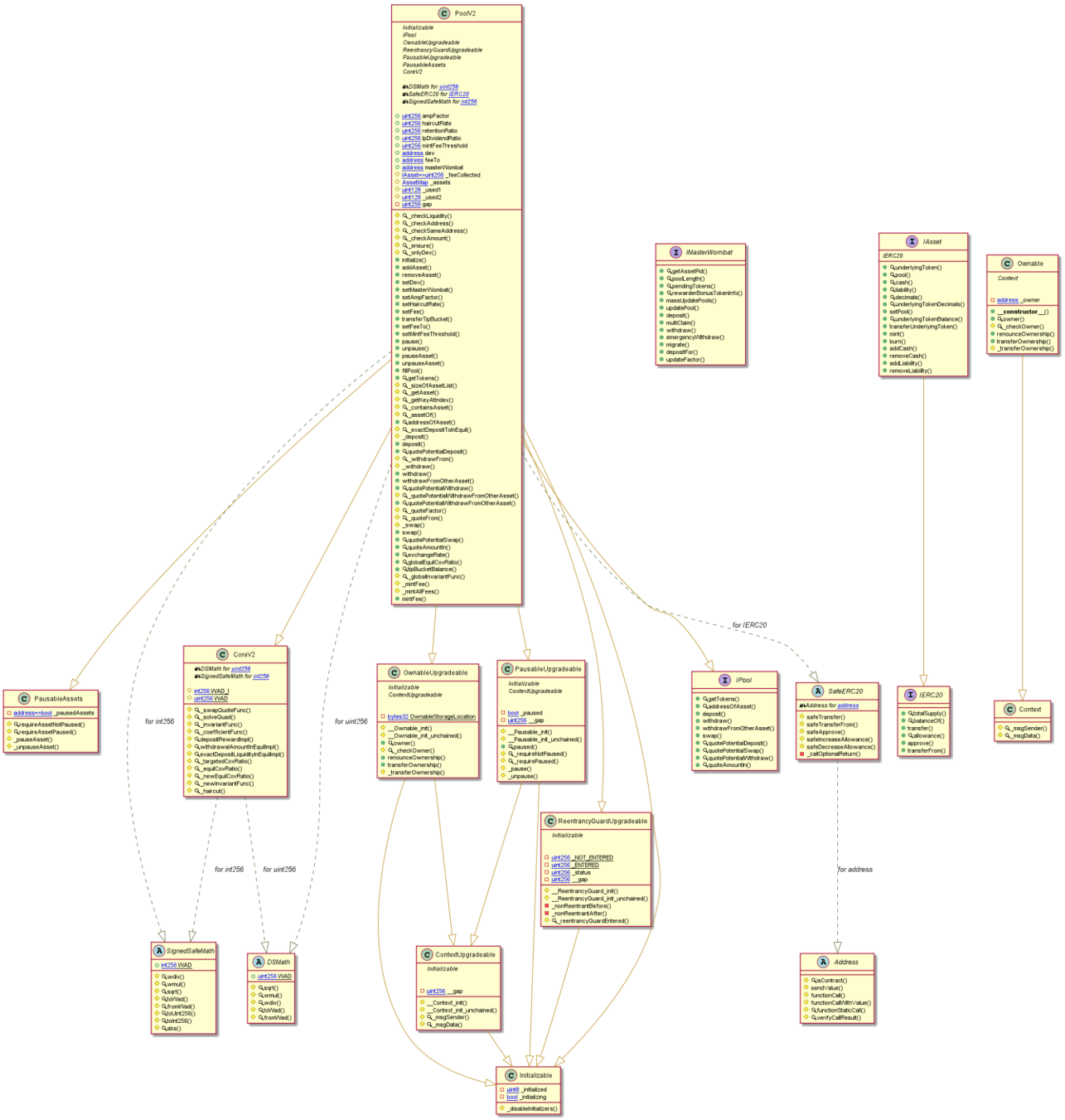
HighCovRatioFeePoolV2 Diagram



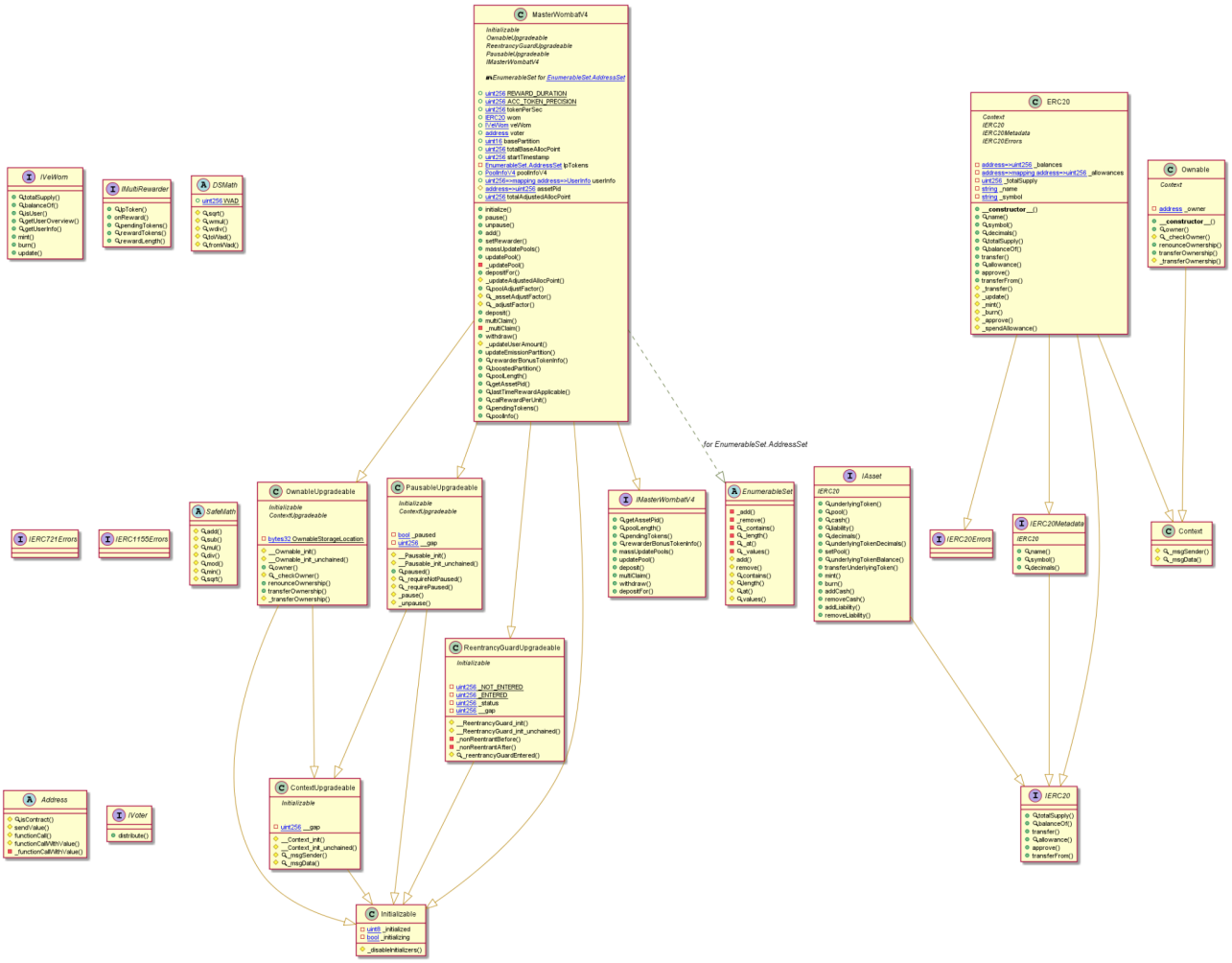
PausableAssets Diagram



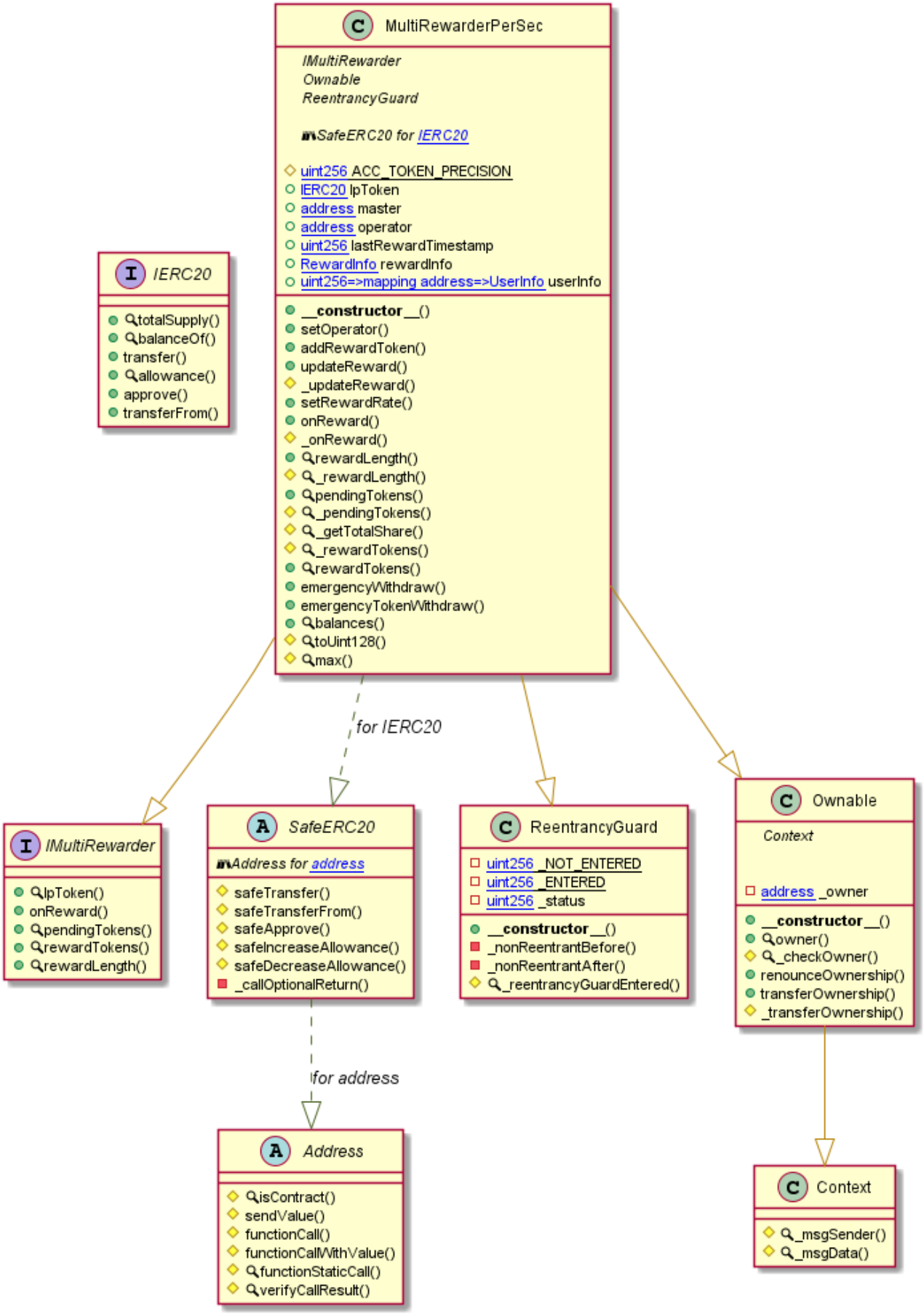
PoolV2 Diagram



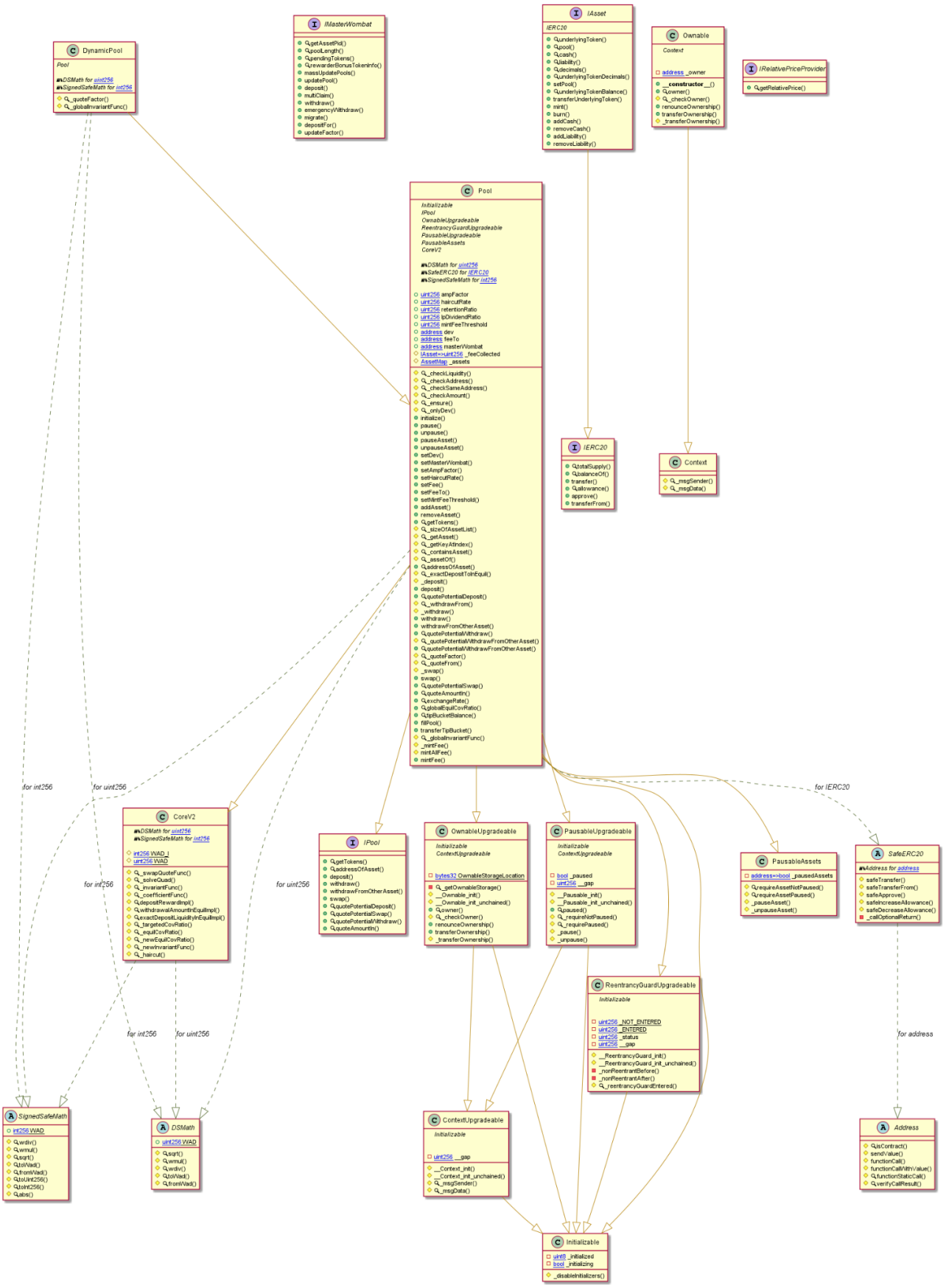
MasterWombatV4 Diagram



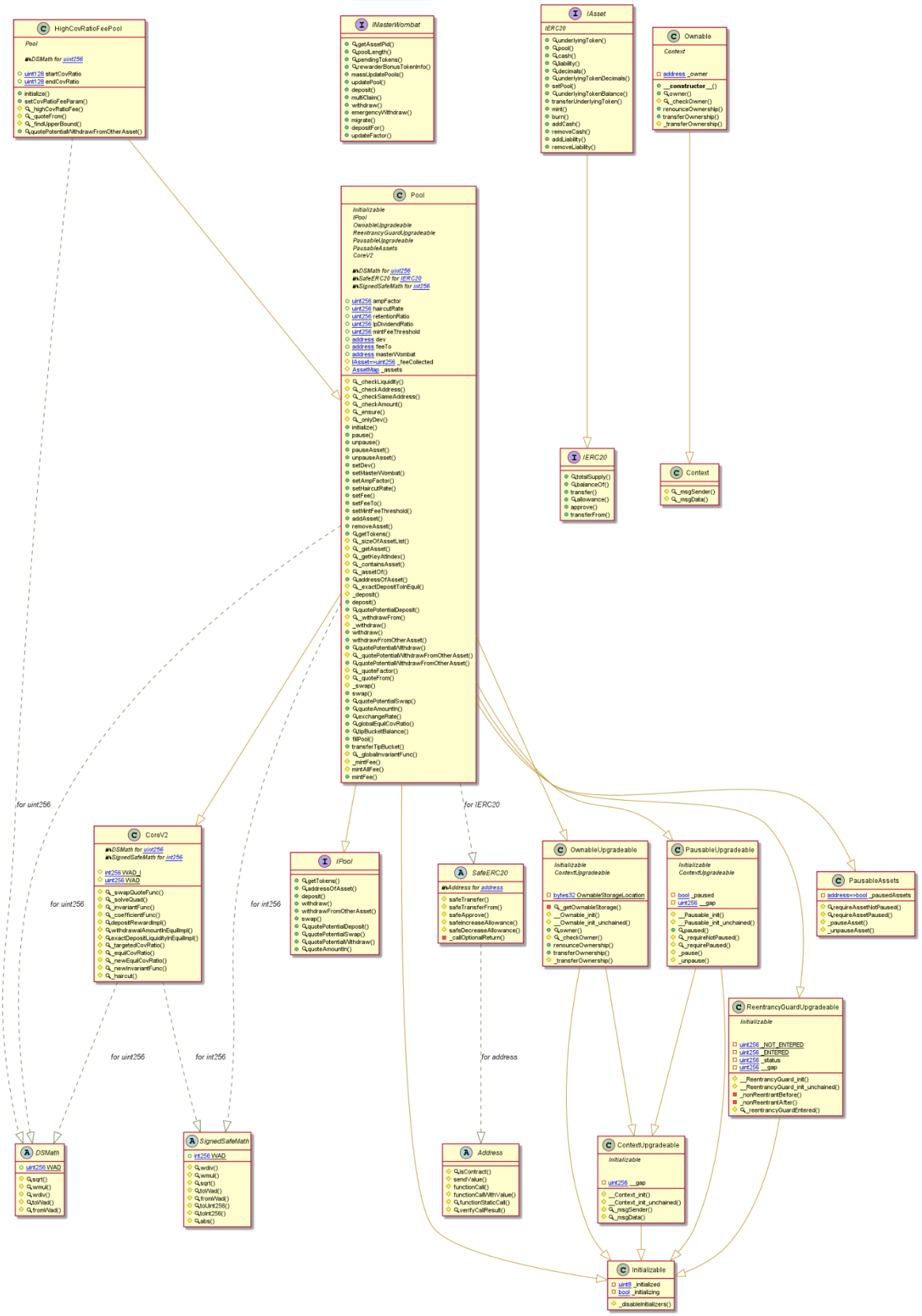
MultiRewarderPerSec Diagram



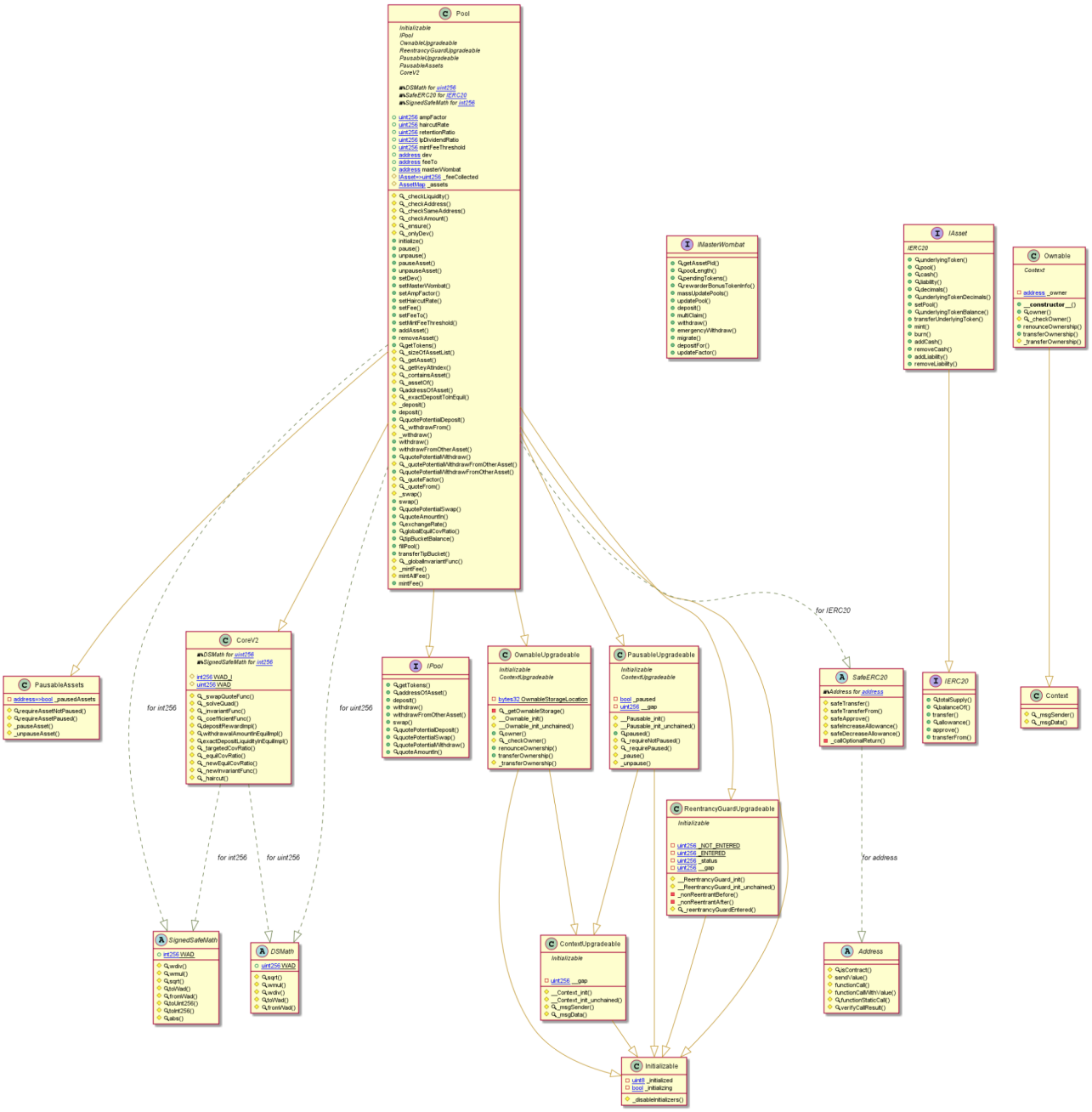
DynamicPool Diagram



HighCovRatioFeePool Diagram



Pool Diagram



Slither Results Log

Slither is a Solidity static analysis framework that uses vulnerability detectors, displays contract details, and provides an API for writing custom analyses. It helps developers identify vulnerabilities, improve code comprehension, and prototype custom analyses quickly. The analysis includes a report with warnings and errors, allowing developers to quickly prototype and fix issues.

We did the analysis of the project altogether. Below are the results.

Slither log >> ABnbcAsset.sol

```
ERC20Permit.constructor(string).name (ABnbcAsset.sol#1110) shadows:
        - ERC20.name() (ABnbcAsset.sol#445-447) (function)
        - IERC20Metadata.name() (ABnbcAsset.sol#408) (function)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#local-variable-shadowing

ERC20Permit.permit(address,address,uint256,uint256,uint8,bytes32,bytes32) (ABnbcAsset.sol#1115-1134) uses timestamp for compar
isons
        Dangerous comparisons:
        - require(bool,string)(block.timestamp <= deadline,ERC20Permit: expired deadline) (ABnbcAsset.sol#1124)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#block-timestamp

Address.verifyCallResult(bool,bytes,string) (ABnbcAsset.sol#130-148) uses assembly
        - INLINE ASM (ABnbcAsset.sol#140-143)
StorageSlot.getAddressSlot(bytes32) (ABnbcAsset.sol#731-736) uses assembly
        - INLINE ASM (ABnbcAsset.sol#733-735)
StorageSlot.getBooleanSlot(bytes32) (ABnbcAsset.sol#741-746) uses assembly
        - INLINE ASM (ABnbcAsset.sol#743-745)
StorageSlot.getBytes32Slot(bytes32) (ABnbcAsset.sol#751-756) uses assembly
        - INLINE ASM (ABnbcAsset.sol#753-755)
StorageSlot.getUint256Slot(bytes32) (ABnbcAsset.sol#761-766) uses assembly
        - INLINE ASM (ABnbcAsset.sol#763-765)
StorageSlot.getStringSlot(bytes32) (ABnbcAsset.sol#771-776) uses assembly
        - INLINE ASM (ABnbcAsset.sol#773-775)
StorageSlot.getStringSlot(string) (ABnbcAsset.sol#781-786) uses assembly
        - INLINE ASM (ABnbcAsset.sol#783-785)
StorageSlot.getBytesSlot(bytes32) (ABnbcAsset.sol#791-796) uses assembly
        - INLINE ASM (ABnbcAsset.sol#793-795)
StorageSlot.getBytesSlot(bytes) (ABnbcAsset.sol#801-806) uses assembly
        - INLINE ASM (ABnbcAsset.sol#803-805)
ShortStrings.toString(ShortStrings.ShortString) (ABnbcAsset.sol#830-840) uses assembly
        - INLINE ASM (ABnbcAsset.sol#835-838)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#assembly-usage
```

```
Pragma version^0.8.21 (ABnbcAsset.sol#2) necessitates a version too recent to be trusted. Consider deploying with 0.6.12/0.7.6
/0.8.16
solc-0.8.21 is not recommended for deployment
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#incorrect-versions-of-solidity

Low level call in Address.sendValue(address,uint256) (ABnbcAsset.sol#72-77):
        - (success) = recipient.call{value: amount}() (ABnbcAsset.sol#75)
Low level call in Address.functionCallWithValue(address,bytes,uint256,string) (ABnbcAsset.sol#99-110):
        - (success,returndata) = target.call{value: value}(data) (ABnbcAsset.sol#108)
Low level call in Address.functionStaticCall(address,bytes,string) (ABnbcAsset.sol#116-125):
        - (success,returndata) = target.staticcall(data) (ABnbcAsset.sol#123)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#low-level-calls

Function EIP712._EIP712Name() (ABnbcAsset.sol#1023-1025) is not in mixedCase
Function EIP712._EIP712Version() (ABnbcAsset.sol#1034-1036) is not in mixedCase
Function IERC20Permit.DOMAIN_SEPARATOR() (ABnbcAsset.sol#1085) is not in mixedCase
Function ERC20Permit.DOMAIN_SEPARATOR() (ABnbcAsset.sol#1147-1149) is not in mixedCase
Variable ERC20Permit._PERMIT_TYPEHASH_DEPRECATED_SLOT (ABnbcAsset.sol#1103) is not in mixedCase
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#conformance-to-solidity-naming-conventions

Redundant expression "this (ABnbcAsset.sol#328)" inContext (ABnbcAsset.sol#322-331)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#redundant-statements

ShortStrings.slitherConstructorConstantVariables() (ABnbcAsset.sol#808-889) uses literals with too many digits:
        - FALLBACK_SENTINEL = 0x00000000000000000000000000000000000000000000000000000000000000FF (ABnbcAsset.sol#811)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#too-many-digits

ABnbcAsset.exchangeRateOracle (ABnbcAsset.sol#1398) should be immutable
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#state-variables-that-could-be-declared-immutable

The function Asset.mint(address,uint256) (ABnbcAsset.sol#1307-1313) reads require(bool,string)(amount + this.totalSupply() <=
maxSupply,Wombat: MAX_SUPPLY_REACHED) (ABnbcAsset.sol#1310) with `this` which adds an extra STATICCALL.
Reference: https://github.com/crytic/slither/wiki/Vulnerabilities-Description#public-variable-read-in-external-context
ABnbcAsset.sol analyzed (24 contracts with 84 detectors), 61 result(s) found
```

Slither log >> Asset.sol

```
ERC20Permit.constructor(string).name (Asset.sol#1107) shadows:
        - ERC20.name() (Asset.sol#440-442) (function)
        - IERC20Metadata.name() (Asset.sol#403) (function)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#local-variable-shadowing

ERC20Permit.permit(address,address,uint256,uint256,uint8,bytes32,bytes32) (Asset.sol#1112-1131) uses timestamp for comparisons
        Dangerous comparisons:
        - require(bool,string)(block.timestamp <= deadline,ERC20Permit: expired deadline) (Asset.sol#1121)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#block-timestamp

Address.verifyCallResult(bool,bytes,string) (Asset.sol#130-148) uses assembly
        - INLINE ASM (Asset.sol#140-143)
StorageSlot.getAddressSlot(bytes32) (Asset.sol#728-733) uses assembly
        - INLINE ASM (Asset.sol#730-732)
StorageSlot.getBooleanSlot(bytes32) (Asset.sol#738-743) uses assembly
        - INLINE ASM (Asset.sol#740-742)
StorageSlot.getBytes32Slot(bytes32) (Asset.sol#748-753) uses assembly
        - INLINE ASM (Asset.sol#750-752)
StorageSlot.getUint256Slot(bytes32) (Asset.sol#758-763) uses assembly
        - INLINE ASM (Asset.sol#760-762)
StorageSlot.getStringSlot(bytes32) (Asset.sol#768-773) uses assembly
        - INLINE ASM (Asset.sol#770-772)
StorageSlot.getStringSlot(string) (Asset.sol#778-783) uses assembly
        - INLINE ASM (Asset.sol#780-782)
StorageSlot.getBytesSlot(bytes32) (Asset.sol#788-793) uses assembly
        - INLINE ASM (Asset.sol#790-792)
StorageSlot.getBytesSlot(bytes) (Asset.sol#798-803) uses assembly
        - INLINE ASM (Asset.sol#800-802)
ShortStrings.toString(ShortStrings.ShortString) (Asset.sol#827-837) uses assembly
        - INLINE ASM (Asset.sol#832-835)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#assembly-usage
```

```
Pragma version^0.8.21 (Asset.sol#2) necessitates a version too recent to be trusted. Consider deploying with 0.6.12/0.7.6/0.8.
16
solc-0.8.21 is not recommended for deployment
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#incorrect-versions-of-solidity

Low level call in Address.sendValue(address,uint256) (Asset.sol#72-77):
        - (success) = recipient.call{value: amount}() (Asset.sol#75)
Low level call in Address.functionCallWithValue(address,bytes,uint256,string) (Asset.sol#99-110):
        - (success,returndata) = target.call{value: value}(data) (Asset.sol#108)
Low level call in Address.functionStaticCall(address,bytes,string) (Asset.sol#116-125):
        - (success,returndata) = target.staticcall(data) (Asset.sol#123)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#low-level-calls

Function EIP712._EIP712Name() (Asset.sol#1020-1022) is not in mixedCase
Function EIP712._EIP712Version() (Asset.sol#1031-1033) is not in mixedCase
Function IERC20Permit.DOMAIN_SEPARATOR() (Asset.sol#1082) is not in mixedCase
Function ERC20Permit.DOMAIN_SEPARATOR() (Asset.sol#1144-1146) is not in mixedCase
Variable ERC20Permit._PERMIT_TYPEHASH_DEPRECATED_SLOT (Asset.sol#1100) is not in mixedCase
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#conformance-to-solidity-naming-conventions

Redundant expression "this (Asset.sol#322)" inContext (Asset.sol#316-325)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#redundant-statements

ShortStrings.slitherConstructorConstantVariables() (Asset.sol#805-886) uses literals with too many digits:
        - FALLBACK_SENTINEL = 0x00000000000000000000000000000000000000000000000000000000000000FF (Asset.sol#808)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#too-many-digits

The function Asset.mint(address,uint256) (Asset.sol#1304-1310) reads require(bool,string)(amount + this.totalSupply() <= maxSu
pply,Wombat: MAX_SUPPLY_REACHED) (Asset.sol#1307) with `this` which adds an extra STATICCALL.
Reference: https://github.com/crytic/slither/wiki/Vulnerabilities-Description#public-variable-read-in-external-context
Asset.sol analyzed (20 contracts with 84 detectors), 60 result(s) found
```

Slither log >> BnbxAsset.sol

```
ERC20Permit.constructor(string).name (BnbxAsset.sol#1108) shadows:
        - ERC20.name() (BnbxAsset.sol#442-444) (function)
        - IERC20Metadata.name() (BnbxAsset.sol#405) (function)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#local-variable-shadowing

ERC20Permit.permit(address,address,uint256,uint256,uint8,bytes32,bytes32) (BnbxAsset.sol#1113-1132) uses timestamp for compari
sons
        Dangerous comparisons:
        - require(bool,string)(block.timestamp <= deadline,ERC20Permit: expired deadline) (BnbxAsset.sol#1122)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#block-timestamp

Address.verifyCallResult(bool,bytes,string) (BnbxAsset.sol#130-148) uses assembly
        - INLINE ASM (BnbxAsset.sol#140-143)
StorageSlot.getAddressSlot(bytes32) (BnbxAsset.sol#730-735) uses assembly
        - INLINE ASM (BnbxAsset.sol#732-734)
StorageSlot.getBooleanSlot(bytes32) (BnbxAsset.sol#740-745) uses assembly
        - INLINE ASM (BnbxAsset.sol#742-744)
StorageSlot.getBytes32Slot(bytes32) (BnbxAsset.sol#750-755) uses assembly
        - INLINE ASM (BnbxAsset.sol#752-754)
StorageSlot.getUint256Slot(bytes32) (BnbxAsset.sol#760-765) uses assembly
        - INLINE ASM (BnbxAsset.sol#762-764)
StorageSlot.getStringSlot(bytes32) (BnbxAsset.sol#770-775) uses assembly
        - INLINE ASM (BnbxAsset.sol#772-774)
StorageSlot.getStringSlot(string) (BnbxAsset.sol#780-785) uses assembly
        - INLINE ASM (BnbxAsset.sol#782-784)
StorageSlot.getBytesSlot(bytes32) (BnbxAsset.sol#790-795) uses assembly
        - INLINE ASM (BnbxAsset.sol#792-794)
StorageSlot.getBytesSlot(bytes) (BnbxAsset.sol#800-805) uses assembly
        - INLINE ASM (BnbxAsset.sol#802-804)
ShortStrings.toString(ShortStrings.ShortString) (BnbxAsset.sol#829-839) uses assembly
        - INLINE ASM (BnbxAsset.sol#834-837)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#assembly-usage
```

```
Pragma version^0.8.21 (BnbxAsset.sol#2) necessitates a version too recent to be trusted. Consider deploying with 0.6.12/0.7.6/
0.8.16
solc-0.8.21 is not recommended for deployment
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#incorrect-versions-of-solidity

Low level call in Address.sendValue(address,uint256) (BnbxAsset.sol#72-77):
        - (success) = recipient.call{value: amount}() (BnbxAsset.sol#75)
Low level call in Address.functionCallWithValue(address,bytes,uint256,string) (BnbxAsset.sol#99-110):
        - (success,returndata) = target.call{value: value}(data) (BnbxAsset.sol#108)
Low level call in Address.functionStaticCall(address,bytes,string) (BnbxAsset.sol#116-125):
        - (success,returndata) = target.staticcall(data) (BnbxAsset.sol#123)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#low-level-calls

Function EIP712._EIP712Name() (BnbxAsset.sol#1022-1024) is not in mixedCase
Function EIP712._EIP712Version() (BnbxAsset.sol#1033-1035) is not in mixedCase
Function IERC20Permit.DOMAIN_SEPARATOR() (BnbxAsset.sol#1084) is not in mixedCase
Function ERC20Permit.DOMAIN_SEPARATOR() (BnbxAsset.sol#1145-1147) is not in mixedCase
Variable ERC20Permit._PERMIT_TYPEHASH_DEPRECATED_SLOT (BnbxAsset.sol#1101) is not in mixedCase
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#conformance-to-solidity-naming-conventions

Redundant expression "this (BnbxAsset.sol#324)" inContext (BnbxAsset.sol#318-327)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#redundant-statements

ShortStrings.slitherConstructorConstantVariables() (BnbxAsset.sol#807-888) uses literals with too many digits:
        - FALLBACK_SENTINEL = 0x00000000000000000000000000000000000000000000000000000000000000FF (BnbxAsset.sol#810)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#too-many-digits

BnbxAsset.exchangeRateOracle (BnbxAsset.sol#1399) should be immutable
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#state-variables-that-could-be-declared-immutable

The function Asset.mint(address,uint256) (BnbxAsset.sol#1306-1312) reads require(bool,string)(amount + this.totalSupply() <= m
axSupply,Wombat: MAX_SUPPLY_REACHED) (BnbxAsset.sol#1309) with `this` which adds an extra STATICCALL.
Reference: https://github.com/crytic/slither/wiki/Vulnerabilities-Description#public-variable-read-in-external-context
BnbxAsset.sol analyzed (24 contracts with 84 detectors), 61 result(s) found
```

Slither log >> DynamicAsset.sol

```
ERC20Permit.constructor(string).name (DynamicAsset.sol#1105) shadows:
        - ERC20.name() (DynamicAsset.sol#438-440) (function)
        - IERC20Metadata.name() (DynamicAsset.sol#401) (function)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#local-variable-shadowing

ERC20Permit.permit(address,address,uint256,uint256,uint8,bytes32,bytes32) (DynamicAsset.sol#1110-1129) uses timestamp for comp
arisons
        Dangerous comparisons:
        - require(bool,string)(block.timestamp <= deadline,ERC20Permit: expired deadline) (DynamicAsset.sol#1119)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#block-timestamp

Address.verifyCallResult(bool,bytes,string) (DynamicAsset.sol#130-148) uses assembly
        - INLINE ASM (DynamicAsset.sol#140-143)
StorageSlot.getAddressSlot(bytes32) (DynamicAsset.sol#726-731) uses assembly
        - INLINE ASM (DynamicAsset.sol#728-730)
StorageSlot.getBooleanSlot(bytes32) (DynamicAsset.sol#736-741) uses assembly
        - INLINE ASM (DynamicAsset.sol#738-740)
StorageSlot.getBytes32Slot(bytes32) (DynamicAsset.sol#746-751) uses assembly
        - INLINE ASM (DynamicAsset.sol#748-750)
StorageSlot.getUint256Slot(bytes32) (DynamicAsset.sol#756-761) uses assembly
        - INLINE ASM (DynamicAsset.sol#758-760)
StorageSlot.getStringSlot(bytes32) (DynamicAsset.sol#766-771) uses assembly
        - INLINE ASM (DynamicAsset.sol#768-770)
StorageSlot.getStringSlot(string) (DynamicAsset.sol#776-781) uses assembly
        - INLINE ASM (DynamicAsset.sol#778-780)
StorageSlot.getBytesSlot(bytes32) (DynamicAsset.sol#786-791) uses assembly
        - INLINE ASM (DynamicAsset.sol#788-790)
StorageSlot.getBytesSlot(bytes) (DynamicAsset.sol#796-801) uses assembly
        - INLINE ASM (DynamicAsset.sol#798-800)
ShortStrings.toString(ShortStrings.ShortString) (DynamicAsset.sol#825-835) uses assembly
        - INLINE ASM (DynamicAsset.sol#830-833)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#assembly-usage
```

```
Pragma version^0.8.21 (DynamicAsset.sol#2) necessitates a version too recent to be trusted. Consider deploying with 0.6.12/0.7
.6/0.8.16
solc-0.8.21 is not recommended for deployment
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#incorrect-versions-of-solidity

Low level call in Address.sendValue(address,uint256) (DynamicAsset.sol#72-77):
        - (success) = recipient.call{value: amount}() (DynamicAsset.sol#75)
Low level call in Address.functionCallWithValue(address,bytes,uint256,string) (DynamicAsset.sol#99-110):
        - (success,returndata) = target.call{value: value}(data) (DynamicAsset.sol#108)
Low level call in Address.functionStaticCall(address,bytes,string) (DynamicAsset.sol#116-125):
        - (success,returndata) = target.staticcall(data) (DynamicAsset.sol#123)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#low-level-calls

Function EIP712._EIP712Name() (DynamicAsset.sol#1018-1020) is not in mixedCase
Function EIP712._EIP712Version() (DynamicAsset.sol#1029-1031) is not in mixedCase
Function IERC20Permit.DOMAIN_SEPARATOR() (DynamicAsset.sol#1080) is not in mixedCase
Function ERC20Permit.DOMAIN_SEPARATOR() (DynamicAsset.sol#1142-1144) is not in mixedCase
Variable ERC20Permit._PERMIT_TYPEHASH_DEPRECATED_SLOT (DynamicAsset.sol#1098) is not in mixedCase
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#conformance-to-solidity-naming-conventions

Redundant expression "this (DynamicAsset.sol#322)" inContext (DynamicAsset.sol#316-325)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#redundant-statements

ShortStrings.slitherConstructorConstantVariables() (DynamicAsset.sol#803-884) uses literals with too many digits:
        - FALLBACK_SENTINEL = 0x00000000000000000000000000000000000000000000000000000000000000FF (DynamicAsset.sol#806)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#too-many-digits

The function Asset.mint(address,uint256) (DynamicAsset.sol#1302-1308) reads require(bool,string)(amount + this.totalSupply() <
= maxSupply,Wombat: MAX_SUPPLY_REACHED) (DynamicAsset.sol#1305) with `this` which adds an extra STATICCALL.
Reference: https://github.com/crytic/slither/wiki/Vulnerabilities-Description#public-variable-read-in-external-context
DynamicAsset.sol analyzed (22 contracts with 84 detectors), 60 result(s) found
```

Slither log >> StkbnbAsset.sol

```
ERC20Permit.constructor(string).name (StkbnbAsset.sol#1107) shadows:
        - ERC20.name() (StkbnbAsset.sol#441-443) (function)
        - IERC20Metadata.name() (StkbnbAsset.sol#404) (function)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#local-variable-shadowing

ERC20Permit.permit(address,address,uint256,uint256,uint8,bytes32,bytes32) (StkbnbAsset.sol#1112-1131) uses timestamp for compa
risons
        Dangerous comparisons:
        - require(bool,string)(block.timestamp <= deadline,ERC20Permit: expired deadline) (StkbnbAsset.sol#1121)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#block-timestamp

Address.verifyCallResult(bool,bytes,string) (StkbnbAsset.sol#130-148) uses assembly
        - INLINE ASM (StkbnbAsset.sol#140-143)
StorageSlot.getAddressSlot(bytes32) (StkbnbAsset.sol#729-734) uses assembly
        - INLINE ASM (StkbnbAsset.sol#731-733)
StorageSlot.getBooleanSlot(bytes32) (StkbnbAsset.sol#739-744) uses assembly
        - INLINE ASM (StkbnbAsset.sol#741-743)
StorageSlot.getBytes32Slot(bytes32) (StkbnbAsset.sol#749-754) uses assembly
        - INLINE ASM (StkbnbAsset.sol#751-753)
StorageSlot.getUint256Slot(bytes32) (StkbnbAsset.sol#759-764) uses assembly
        - INLINE ASM (StkbnbAsset.sol#761-763)
StorageSlot.getStringSlot(bytes32) (StkbnbAsset.sol#769-774) uses assembly
        - INLINE ASM (StkbnbAsset.sol#771-773)
StorageSlot.getStringSlot(string) (StkbnbAsset.sol#779-784) uses assembly
        - INLINE ASM (StkbnbAsset.sol#781-783)
StorageSlot.getBytesSlot(bytes32) (StkbnbAsset.sol#789-794) uses assembly
        - INLINE ASM (StkbnbAsset.sol#791-793)
StorageSlot.getBytesSlot(bytes) (StkbnbAsset.sol#799-804) uses assembly
        - INLINE ASM (StkbnbAsset.sol#801-803)
ShortStrings.toString(ShortStrings.ShortString) (StkbnbAsset.sol#828-838) uses assembly
        - INLINE ASM (StkbnbAsset.sol#833-836)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#assembly-usage
```


```
ERC20Permit.constructor(string).name (StkbnbAsset.sol#1107) shadows:
        - ERC20.name() (StkbnbAsset.sol#441-443) (function)
        - IERC20Metadata.name() (StkbnbAsset.sol#404) (function)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#local-variable-shadowing

ERC20Permit.permit(address,address,uint256,uint256,uint8,bytes32,bytes32) (StkbnbAsset.sol#1112-1131) uses timestamp for compa
risons
        Dangerous comparisons:
        - require(bool,string)(block.timestamp <= deadline,ERC20Permit: expired deadline) (StkbnbAsset.sol#1121)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#block-timestamp

Address.verifyCallResult(bool,bytes,string) (StkbnbAsset.sol#130-148) uses assembly
        - INLINE ASM (StkbnbAsset.sol#140-143)
StorageSlot.getAddressSlot(bytes32) (StkbnbAsset.sol#729-734) uses assembly
        - INLINE ASM (StkbnbAsset.sol#731-733)
StorageSlot.getBooleanSlot(bytes32) (StkbnbAsset.sol#739-744) uses assembly
        - INLINE ASM (StkbnbAsset.sol#741-743)
StorageSlot.getBytes32Slot(bytes32) (StkbnbAsset.sol#749-754) uses assembly
        - INLINE ASM (StkbnbAsset.sol#751-753)
StorageSlot.getUint256Slot(bytes32) (StkbnbAsset.sol#759-764) uses assembly
        - INLINE ASM (StkbnbAsset.sol#761-763)
StorageSlot.getStringSlot(bytes32) (StkbnbAsset.sol#769-774) uses assembly
        - INLINE ASM (StkbnbAsset.sol#771-773)
StorageSlot.getStringSlot(string) (StkbnbAsset.sol#779-784) uses assembly
        - INLINE ASM (StkbnbAsset.sol#781-783)
StorageSlot.getBytesSlot(bytes32) (StkbnbAsset.sol#789-794) uses assembly
        - INLINE ASM (StkbnbAsset.sol#791-793)
StorageSlot.getBytesSlot(bytes) (StkbnbAsset.sol#799-804) uses assembly
        - INLINE ASM (StkbnbAsset.sol#801-803)
ShortStrings.toString(ShortStrings.ShortString) (StkbnbAsset.sol#828-838) uses assembly
        - INLINE ASM (StkbnbAsset.sol#833-836)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#assembly-usage
```


Slither log >> CoreV2.sol

```
CoreV2.withdrawalAmountInEquilImpl(int256,int256,int256,int256) (CoreV2.sol#256-268) performs a multiplication on the result o
f a division:
        - beta = (rho + delta_i.wmul(WAD_I - A)) / 2 (CoreV2.sol#265)
        - A_i_ = beta + (beta * beta + A.wmul(L_i_ * L_i_)).sqrt(beta) (CoreV2.sol#266)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#divide-before-multiply

CoreV2._coefficientFunc(int256,int256,int256,int256,int256) (CoreV2.sol#226-228) is never used and should be removed
CoreV2._equilCovRatio(int256,int256,int256) (CoreV2.sol#315-318) is never used and should be removed
CoreV2._haircut(uint256,uint256) (CoreV2.sol#335-337) is never used and should be removed
CoreV2._invariantFunc(int256,int256,int256,int256,int256) (CoreV2.sol#210-214) is never used and should be removed
CoreV2._newEquilCovRatio(int256,int256,int256) (CoreV2.sol#320-322) is never used and should be removed
CoreV2._newInvariantFunc(int256,int256,int256,int256) (CoreV2.sol#324-326) is never used and should be removed
CoreV2._solveQuad(int256,int256) (CoreV2.sol#196-198) is never used and should be removed
CoreV2._swapQuoteFunc(int256,int256,int256,int256,int256,int256) (CoreV2.sol#165-187) is never used and should be removed
CoreV2._targetedCovRatio(int256,int256,int256,int256,int256,int256) (CoreV2.sol#297-313) is never used and should be removed
CoreV2.depositRewardImpl(int256,int256,int256,int256,int256,int256) (CoreV2.sol#233-251) is never used and should be removed
CoreV2.exactDepositLiquidityInEquilImpl(int256,int256,int256,int256) (CoreV2.sol#273-295) is never used and should be removed
CoreV2.withdrawalAmountInEquilImpl(int256,int256,int256,int256) (CoreV2.sol#256-268) is never used and should be removed
DSMath.fromWad(uint256,uint8) (CoreV2.sol#137-144) is never used and should be removed
DSMath.sqrt(uint256) (CoreV2.sol#104-115) is never used and should be removed
DSMath.toWad(uint256,uint8) (CoreV2.sol#127-134) is never used and should be removed
DSMath.wdiv(uint256,uint256) (CoreV2.sol#122-124) is never used and should be removed
DSMath.wmul(uint256,uint256) (CoreV2.sol#118-120) is never used and should be removed
SignedSafeMath.abs(int256) (CoreV2.sol#83-89) is never used and should be removed
SignedSafeMath.fromWad(int256,uint8) (CoreV2.sol#64-71) is never used and should be removed
SignedSafeMath.sqrt(int256) (CoreV2.sol#20-31) is never used and should be removed
SignedSafeMath.sqrt(int256,int256) (CoreV2.sol#34-51) is never used and should be removed
```

```
SignedSafeMath.toInt256(uint256) (CoreV2.sol#78-81) is never used and should be removed
SignedSafeMath.toUint256(int256) (CoreV2.sol#73-76) is never used and should be removed
SignedSafeMath.toWad(int256,uint8) (CoreV2.sol#54-61) is never used and should be removed
SignedSafeMath.wdiv(int256,int256) (CoreV2.sol#10-12) is never used and should be removed
SignedSafeMath.wmul(int256,int256) (CoreV2.sol#15-17) is never used and should be removed
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#dead-code

Pragma version^0.8.21 (CoreV2.sol#2) necessitates a version too recent to be trusted. Consider deploying with 0.6.12/0.7.6/0.8
.16
solc-0.8.21 is not recommended for deployment
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#incorrect-versions-of-solidity

Parameter CoreV2.depositRewardImpl(int256,int256,int256,int256,int256,int256).D (CoreV2.sol#234) is not in mixedCase
Parameter CoreV2.depositRewardImpl(int256,int256,int256,int256,int256,int256).SL (CoreV2.sol#235) is not in mixedCase
Parameter CoreV2.depositRewardImpl(int256,int256,int256,int256,int256,int256).delta_i (CoreV2.sol#236) is not in mixedCase
Parameter CoreV2.depositRewardImpl(int256,int256,int256,int256,int256,int256).A_i (CoreV2.sol#237) is not in mixedCase
Parameter CoreV2.depositRewardImpl(int256,int256,int256,int256,int256,int256).L_i (CoreV2.sol#238) is not in mixedCase
Parameter CoreV2.depositRewardImpl(int256,int256,int256,int256,int256,int256).A (CoreV2.sol#239) is not in mixedCase
Parameter CoreV2.withdrawalAmountInEquilImpl(int256,int256,int256,int256).delta_i (CoreV2.sol#257) is not in mixedCase
Parameter CoreV2.withdrawalAmountInEquilImpl(int256,int256,int256,int256).A_i (CoreV2.sol#258) is not in mixedCase
Parameter CoreV2.withdrawalAmountInEquilImpl(int256,int256,int256,int256).L_i (CoreV2.sol#259) is not in mixedCase
Parameter CoreV2.withdrawalAmountInEquilImpl(int256,int256,int256,int256).A (CoreV2.sol#260) is not in mixedCase
Parameter CoreV2.exactDepositLiquidityInEquilImpl(int256,int256,int256,int256).D_i (CoreV2.sol#274) is not in mixedCase
Parameter CoreV2.exactDepositLiquidityInEquilImpl(int256,int256,int256,int256).A_i (CoreV2.sol#275) is not in mixedCase
Parameter CoreV2.exactDepositLiquidityInEquilImpl(int256,int256,int256,int256).L_i (CoreV2.sol#276) is not in mixedCase
Parameter CoreV2.exactDepositLiquidityInEquilImpl(int256,int256,int256,int256).A (CoreV2.sol#277) is not in mixedCase
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#conformance-to-solidity-naming-conventions

CoreV2.WAD (CoreV2.sol#151) is never used in CoreV2 (CoreV2.sol#147-338)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#unused-state-variable
CoreV2.sol analyzed (3 contracts with 84 detectors), 44 result(s) found
```

Slither log >> PausableAssets.sol

```
PausableAssets._pauseAsset(address) (PausableAssets.sol#52-56) is never used and should be removed
PausableAssets._unpauseAsset(address) (PausableAssets.sol#65-69) is never used and should be removed
PausableAssets.requireAssetNotPaused(address) (PausableAssets.sol#30-32) is never used and should be removed
PausableAssets.requireAssetPaused(address) (PausableAssets.sol#41-43) is never used and should be removed
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#dead-code

Pragma version^0.8.21 (PausableAssets.sol#2) necessitates a version too recent to be trusted. Consider deploying with 0.6.12/0
.7.6/0.8.16
solc-0.8.21 is not recommended for deployment
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#incorrect-versions-of-solidity
PausableAssets.sol analyzed (1 contracts with 84 detectors), 6 result(s) found
```

Slither log >> MultiRewarderPerSec.sol

```
IMultiRewarder.lpToken().lpToken (MultiRewarderPerSec.sol#29) shadows:
        - IMultiRewarder.lpToken() (MultiRewarderPerSec.sol#29) (function)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#local-variable-shadowing

MultiRewarderPerSec.setOperator(address) (MultiRewarderPerSec.sol#414-416) should emit an event for:
        - operator = _operator (MultiRewarderPerSec.sol#415)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#missing-events-access-control

MultiRewarderPerSec.setOperator(address)._operator (MultiRewarderPerSec.sol#414) lacks a zero-check on :
                - operator = _operator (MultiRewarderPerSec.sol#415)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#missing-zero-address-validation

MultiRewarderPerSec.emergencyTokenWithdraw(address) (MultiRewarderPerSec.sol#603-612) has external calls inside a loop: (succe
ss) = msg.sender.call{value: address(this).balance}() (MultiRewarderPerSec.sol#607)
MultiRewarderPerSec.emergencyTokenWithdraw(address) (MultiRewarderPerSec.sol#603-612) has external calls inside a loop: IERC20
(token).safeTransfer(msg.sender,IERC20(token).balanceOf(address(this))) (MultiRewarderPerSec.sol#610)
Address.functionCallWithValue(address,bytes,uint256,string) (MultiRewarderPerSec.sol#74-85) has external calls inside a loop:
(success,returndata) = target.call{value: value}(data) (MultiRewarderPerSec.sol#83)
MultiRewarderPerSec.balances() (MultiRewarderPerSec.sol#615-628) has external calls inside a loop: balances_[i] = pool.rewardT
oken.balanceOf(address(this)) (MultiRewarderPerSec.sol#625)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation/#calls-inside-a-loop

Variable 'MultiRewarderPerSec._onReward(address,uint256).success (MultiRewarderPerSec.sol#500)' in MultiRewarderPerSec._onRewa
rd(address,uint256) (MultiRewarderPerSec.sol#481-529) potentially used before declaration: (success) = _user.call{value: pendi
ng}() (MultiRewarderPerSec.sol#505)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#pre-declaration-usage-of-local-variables
```

```
MultiRewarderPerSec.constructor(address,IERC20,uint256,IERC20,uint96) (MultiRewarderPerSec.sol#387-411) uses timestamp for com
parisons
        Dangerous comparisons:
        - require(bool)(_startTimestamp >= block.timestamp) (MultiRewarderPerSec.sol#394)
MultiRewarderPerSec._updateReward(uint256) (MultiRewarderPerSec.sol#441-454) uses timestamp for comparisons
        Dangerous comparisons:
        - block.timestamp > lastRewardTimestamp && totalShare > 0 (MultiRewarderPerSec.sol#442)
        - i < length (MultiRewarderPerSec.sol#444)
MultiRewarderPerSec._onReward(address,uint256) (MultiRewarderPerSec.sol#481-529) uses timestamp for comparisons
        Dangerous comparisons:
        - i < length (MultiRewarderPerSec.sol#484)
MultiRewarderPerSec._pendingTokens(address) (MultiRewarderPerSec.sol#547-570) uses timestamp for comparisons
        Dangerous comparisons:
        - i < length (MultiRewarderPerSec.sol#551)
        - block.timestamp > lastRewardTimestamp && totalShare > 0 (MultiRewarderPerSec.sol#558)
MultiRewarderPerSec._rewardTokens() (MultiRewarderPerSec.sol#577-584) uses timestamp for comparisons
        Dangerous comparisons:
        - i < length (MultiRewarderPerSec.sol#580)
MultiRewarderPerSec.balances() (MultiRewarderPerSec.sol#615-628) uses timestamp for comparisons
        Dangerous comparisons:
        - i < length (MultiRewarderPerSec.sol#619)
MultiRewarderPerSec.toUint128(uint256) (MultiRewarderPerSec.sol#630-633) uses timestamp for comparisons
        Dangerous comparisons:
        - val > type()(uint128).max (MultiRewarderPerSec.sol#631)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#block-timestamp

Address.verifyCallResult(bool,bytes,string) (MultiRewarderPerSec.sol#105-123) uses assembly
        - INLINE ASM (MultiRewarderPerSec.sol#115-118)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#assembly-usage
```

```
Address.functionCall(address,bytes) (MultiRewarderPerSec.sol#54-56) is never used and should be removed
Address.functionCallWithValue(address,bytes,uint256) (MultiRewarderPerSec.sol#66-72) is never used and should be removed
Address.functionStaticCall(address,bytes) (MultiRewarderPerSec.sol#87-89) is never used and should be removed
Address.functionStaticCall(address,bytes,string) (MultiRewarderPerSec.sol#91-100) is never used and should be removed
Address.sendValue(address,uint256) (MultiRewarderPerSec.sol#47-52) is never used and should be removed
Context._msgData() (MultiRewarderPerSec.sol#254-256) is never used and should be removed
ReentrancyGuard._reentrancyGuardEntered() (MultiRewarderPerSec.sol#243-245) is never used and should be removed
SafeERC20.safeApprove(IERC20,address,uint256) (MultiRewarderPerSec.sol#147-157) is never used and should be removed
SafeERC20.safeDecreaseAllowance(IERC20,address,uint256) (MultiRewarderPerSec.sol#168-179) is never used and should be removed
SafeERC20.safeIncreaseAllowance(IERC20,address,uint256) (MultiRewarderPerSec.sol#159-166) is never used and should be removed
SafeERC20.safeTransferFrom(IERC20,address,address,uint256) (MultiRewarderPerSec.sol#138-145) is never used and should be remov
ed
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#dead-code

Pragma version^0.8.21 (MultiRewarderPerSec.sol#2) necessitates a version too recent to be trusted. Consider deploying with 0.6
.12/0.7.6/0.8.16
solc-0.8.21 is not recommended for deployment
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#incorrect-versions-of-solidity

Low level call in Address.sendValue(address,uint256) (MultiRewarderPerSec.sol#47-52):
        - (success) = recipient.call{value: amount}() (MultiRewarderPerSec.sol#50)
Low level call in Address.functionCallWithValue(address,bytes,uint256,string) (MultiRewarderPerSec.sol#74-85):
        - (success,returndata) = target.call{value: value}(data) (MultiRewarderPerSec.sol#83)
Low level call in Address.functionStaticCall(address,bytes,string) (MultiRewarderPerSec.sol#91-100):
        - (success,returndata) = target.staticcall(data) (MultiRewarderPerSec.sol#98)
Low level call in MultiRewarderPerSec._onReward(address,uint256) (MultiRewarderPerSec.sol#481-529):
        - (success) = _user.call{value: tokenBalance}() (MultiRewarderPerSec.sol#500)
        - (success) = _user.call{value: pending}() (MultiRewarderPerSec.sol#505)
Low level call in MultiRewarderPerSec.emergencyTokenWithdraw(address) (MultiRewarderPerSec.sol#603-612):
        - (success) = msg.sender.call{value: address(this).balance}() (MultiRewarderPerSec.sol#607)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#low-level-calls
```

```
Parameter MultiRewarderPerSec.setOperator(address)._operator (MultiRewarderPerSec.sol#414) is not in mixedCase
Parameter MultiRewarderPerSec.addRewardToken(IERC20,uint96)._rewardToken (MultiRewarderPerSec.sol#418) is not in mixedCase
Parameter MultiRewarderPerSec.addRewardToken(IERC20,uint96)._tokenPerSec (MultiRewarderPerSec.sol#418) is not in mixedCase
Parameter MultiRewarderPerSec.setRewardRate(uint256,uint96)._tokenId (MultiRewarderPerSec.sol#458) is not in mixedCase
Parameter MultiRewarderPerSec.setRewardRate(uint256,uint96)._tokenPerSec (MultiRewarderPerSec.sol#458) is not in mixedCase
Parameter MultiRewarderPerSec.onReward(address,uint256)._user (MultiRewarderPerSec.sol#474) is not in mixedCase
Parameter MultiRewarderPerSec.onReward(address,uint256)._lpAmount (MultiRewarderPerSec.sol#475) is not in mixedCase
Parameter MultiRewarderPerSec.pendingTokens(address)._user (MultiRewarderPerSec.sol#543) is not in mixedCase
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#conformance-to-solidity-naming-conventions
MultiRewarderPerSec.sol analyzed (8 contracts with 84 detectors), 51 result(s) found
```

Slither log >> PoolV2.sol

```
PoolV2.setDev(address).dev_ (PoolV2.sol#1005) lacks a zero-check on :
                - dev = dev_ (PoolV2.sol#1007)
PoolV2.setMasterWombat(address).masterWombat_ (PoolV2.sol#1011) lacks a zero-check on :
                - masterWombat = masterWombat_ (PoolV2.sol#1013)
PoolV2.setFeeTo(address).feeTo_ (PoolV2.sol#1047) lacks a zero-check on :
                - feeTo = feeTo_ (PoolV2.sol#1049)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#missing-zero-address-validation

Reentrancy in PoolV2.fillPool(address,uint256) (PoolV2.sol#1078-1090):
        External calls:
        - asset.addCash(amount) (PoolV2.sol#1088)
        Event emitted after the call(s):
        - FillPool(token,amount) (PoolV2.sol#1089)
Reentrancy in PoolV2.setFee(uint256,uint256) (PoolV2.sol#1027-1034):
        External calls:
        - _mintAllFees() (PoolV2.sol#1030)
                - asset.transferUnderlyingToken(feeTo,dividend.fromWad(asset.underlyingTokenDecimals())) (PoolV2.sol#1532)
                - asset.addLiability(liabilityToMint) (PoolV2.sol#1539)
                - asset.addCash(lpDividend) (PoolV2.sol#1540)
        Event emitted after the call(s):
        - SetFee(lpDividendRatio_,retentionRatio_) (PoolV2.sol#1033)
Reentrancy in PoolV2.transferTipBucket(address,uint256,address) (PoolV2.sol#1036-1045):
        External calls:
        - asset.transferUnderlyingToken(to,amount.fromWad(asset.underlyingTokenDecimals())) (PoolV2.sol#1043)
        Event emitted after the call(s):
        - TransferTipBucket(token,amount,to) (PoolV2.sol#1044)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#reentrancy-vulnerabilities-3

Address.verifyCallResult(bool,bytes,string) (PoolV2.sol#618-636) uses assembly
        - INLINE ASM (PoolV2.sol#628-631)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#assembly-usage
```

```
Pragma version^0.8.21 (PoolV2.sol#2) necessitates a version too recent to be trusted. Consider deploying with 0.6.12/0.7.6/0.8
.16
solc-0.8.21 is not recommended for deployment
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#incorrect-versions-of-solidity

Low level call in Address.sendValue(address,uint256) (PoolV2.sol#560-565):
        - (success) = recipient.call{value: amount}() (PoolV2.sol#563)
Low level call in Address.functionCallWithValue(address,bytes,uint256,string) (PoolV2.sol#587-598):
        - (success,returndata) = target.call{value: value}(data) (PoolV2.sol#596)
Low level call in Address.functionStaticCall(address,bytes,string) (PoolV2.sol#604-613):
        - (success,returndata) = target.staticcall(data) (PoolV2.sol#611)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#low-level-calls

Parameter CoreV2.depositRewardImpl(int256,int256,int256,int256,int256,int256).D (PoolV2.sol#206) is not in mixedCase
Parameter CoreV2.depositRewardImpl(int256,int256,int256,int256,int256,int256).SL (PoolV2.sol#207) is not in mixedCase
Parameter CoreV2.depositRewardImpl(int256,int256,int256,int256,int256,int256).delta_i (PoolV2.sol#208) is not in mixedCase
Parameter CoreV2.depositRewardImpl(int256,int256,int256,int256,int256,int256).A_i (PoolV2.sol#209) is not in mixedCase
Parameter CoreV2.depositRewardImpl(int256,int256,int256,int256,int256,int256).L_i (PoolV2.sol#210) is not in mixedCase
Parameter CoreV2.depositRewardImpl(int256,int256,int256,int256,int256,int256).A (PoolV2.sol#211) is not in mixedCase
Parameter CoreV2.withdrawalAmountInEquilImpl(int256,int256,int256,int256).delta_i (PoolV2.sol#225) is not in mixedCase
Parameter CoreV2.withdrawalAmountInEquilImpl(int256,int256,int256,int256).A_i (PoolV2.sol#226) is not in mixedCase
Parameter CoreV2.withdrawalAmountInEquilImpl(int256,int256,int256,int256).L_i (PoolV2.sol#227) is not in mixedCase
```

```
Function ReentrancyGuardUpgradeable.__ReentrancyGuard_init() (PoolV2.sol#785-787) is not in mixedCase
Function ReentrancyGuardUpgradeable.__ReentrancyGuard_init_unchained() (PoolV2.sol#789-791) is not in mixedCase
Variable ReentrancyGuardUpgradeable.__gap (PoolV2.sol#813) is not in mixedCase
Function PausableUpgradeable.__Pausable_init() (PoolV2.sol#824-826) is not in mixedCase
Function PausableUpgradeable.__Pausable_init_unchained() (PoolV2.sol#828-830) is not in mixedCase
Variable PausableUpgradeable.__gap (PoolV2.sol#864) is not in mixedCase
Variable PoolV2._feeCollected (PoolV2.sol#904) is not in mixedCase
Variable PoolV2._assets (PoolV2.sol#906) is not in mixedCase
Variable PoolV2._used1 (PoolV2.sol#908) is not in mixedCase
Variable PoolV2._used2 (PoolV2.sol#909) is not in mixedCase
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#conformance-to-solidity-naming-conventions

Redundant expression "this (PoolV2.sol#409)" inContext (PoolV2.sol#404-412)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#redundant-statements

PoolV2._used1 (PoolV2.sol#908) is never used in PoolV2 (PoolV2.sol#868-1557)
PoolV2._used2 (PoolV2.sol#909) is never used in PoolV2 (PoolV2.sol#868-1557)
PoolV2.gap (PoolV2.sol#910) is never used in PoolV2 (PoolV2.sol#868-1557)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#unused-state-variable

PoolV2._used1 (PoolV2.sol#908) should be constant
PoolV2._used2 (PoolV2.sol#909) should be constant
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#state-variables-that-could-be-declared-constant
PoolV2.sol analyzed (18 contracts with 84 detectors), 87 result(s) found
```

Slither log >> MasterWombatV4.sol

```
IMultiRewarder.lpToken().lpToken (MasterWombatV4.sol#106) shadows:
        - IMultiRewarder.lpToken() (MasterWombatV4.sol#106) (function)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#local-variable-shadowing

MasterWombatV4.initialize(IERC20,IVeWom,address,uint16,uint256,uint256)._voter (MasterWombatV4.sol#981) lacks a zero-check on
:
                - voter = _voter (MasterWombatV4.sol#991)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#missing-zero-address-validation

MasterWombatV4.calRewardPerUnit(uint256) (MasterWombatV4.sol#1282-1302) has external calls inside a loop: lpSupply = pool.lpTo
ken.balanceOf(address(this)) (MasterWombatV4.sol#1284)
MasterWombatV4._updatePool(uint256) (MasterWombatV4.sol#1067-1080) has external calls inside a loop: IVoter(voter).distribute(
address(pool.lpToken)) (MasterWombatV4.sol#1077)
MasterWombatV4.rewarderBonusTokenInfo(uint256) (MasterWombatV4.sol#1245-1264) has external calls inside a loop: bonusTokenSymb
ols[i] = IERC20Metadata(address(bonusTokenAddresses[i])).symbol() (MasterWombatV4.sol#1261)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation/#calls-inside-a-loop

Reentrancy in MasterWombatV4.add(uint256,IERC20,IMultiRewarder) (MasterWombatV4.sol#1004-1038):
        External calls:
        - massUpdatePools() (MasterWombatV4.sol#1011)
                - IVoter(voter).distribute(address(pool.lpToken)) (MasterWombatV4.sol#1077)
        State variables written after the call(s):
        - assetPid[address(_lpToken)] = poolInfoV4.length (MasterWombatV4.sol#1034)
        - totalBaseAllocPoint += _baseAllocPoint (MasterWombatV4.sol#1016)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#reentrancy-vulnerabilities-2

Reentrancy in MasterWombatV4.add(uint256,IERC20,IMultiRewarder) (MasterWombatV4.sol#1004-1038):
        External calls:
        - massUpdatePools() (MasterWombatV4.sol#1011)
                - IVoter(voter).distribute(address(pool.lpToken)) (MasterWombatV4.sol#1077)
        Event emitted after the call(s):
        - Add(poolInfoV4.length - 1,_lpToken,_rewarder) (MasterWombatV4.sol#1037)
Reentrancy in MasterWombatV4.updateEmissionPartition(uint16) (MasterWombatV4.sol#1237-1242):
        External calls:
        - massUpdatePools() (MasterWombatV4.sol#1239)
```


```
IMultiRewarder.lpToken().lpToken (MasterWombatV4.sol#106) shadows:
        - IMultiRewarder.lpToken() (MasterWombatV4.sol#106) (function)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#local-variable-shadowing

MasterWombatV4.initialize(IERC20,IVeWom,address,uint16,uint256,uint256)._voter (MasterWombatV4.sol#981) lacks a zero-check on
:
                - voter = _voter (MasterWombatV4.sol#991)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#missing-zero-address-validation

MasterWombatV4.calRewardPerUnit(uint256) (MasterWombatV4.sol#1282-1302) has external calls inside a loop: lpSupply = pool.lpTo
ken.balanceOf(address(this)) (MasterWombatV4.sol#1284)
MasterWombatV4._updatePool(uint256) (MasterWombatV4.sol#1067-1080) has external calls inside a loop: IVoter(voter).distribute(
address(pool.lpToken)) (MasterWombatV4.sol#1077)
MasterWombatV4.rewarderBonusTokenInfo(uint256) (MasterWombatV4.sol#1245-1264) has external calls inside a loop: bonusTokenSymb
ols[i] = IERC20Metadata(address(bonusTokenAddresses[i])).symbol() (MasterWombatV4.sol#1261)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation/#calls-inside-a-loop

Reentrancy in MasterWombatV4.add(uint256,IERC20,IMultiRewarder) (MasterWombatV4.sol#1004-1038):
        External calls:
        - massUpdatePools() (MasterWombatV4.sol#1011)
                - IVoter(voter).distribute(address(pool.lpToken)) (MasterWombatV4.sol#1077)
        State variables written after the call(s):
        - assetPid[address(_lpToken)] = poolInfoV4.length (MasterWombatV4.sol#1034)
        - totalBaseAllocPoint += _baseAllocPoint (MasterWombatV4.sol#1016)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#reentrancy-vulnerabilities-2

Reentrancy in MasterWombatV4.add(uint256,IERC20,IMultiRewarder) (MasterWombatV4.sol#1004-1038):
        External calls:
        - massUpdatePools() (MasterWombatV4.sol#1011)
                - IVoter(voter).distribute(address(pool.lpToken)) (MasterWombatV4.sol#1077)
        Event emitted after the call(s):
        - Add(poolInfoV4.length - 1,_lpToken,_rewarder) (MasterWombatV4.sol#1037)
Reentrancy in MasterWombatV4.updateEmissionPartition(uint16) (MasterWombatV4.sol#1237-1242):
        External calls:
        - massUpdatePools() (MasterWombatV4.sol#1239)
```

```
Pragma version^0.8.21 (MasterWombatV4.sol#2) necessitates a version too recent to be trusted. Consider deploying with 0.6.12/0
.7.6/0.8.16
solc-0.8.21 is not recommended for deployment
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#incorrect-versions-of-solidity

Low level call in Address.sendValue(address,uint256) (MasterWombatV4.sol#556-561):
        - (success) = recipient.call{value: amount}() (MasterWombatV4.sol#559)
Low level call in Address._functionCallWithValue(address,bytes,uint256,string) (MasterWombatV4.sol#593-615):
        - (success,returndata) = target.call{value: weiValue}(data) (MasterWombatV4.sol#601)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#low-level-calls
```

```
Parameter MasterWombatV4.setRewarder(uint256,uint256,IMultiRewarder)._rewarder (MasterWombatV4.sol#1040) is not in mixedCase
Parameter MasterWombatV4.updatePool(uint256)._pid (MasterWombatV4.sol#1063) is not in mixedCase
Parameter MasterWombatV4.depositFor(uint256,uint256,address)._pid (MasterWombatV4.sol#1082) is not in mixedCase
Parameter MasterWombatV4.depositFor(uint256,uint256,address)._amount (MasterWombatV4.sol#1082) is not in mixedCase
Parameter MasterWombatV4.depositFor(uint256,uint256,address)._user (MasterWombatV4.sol#1082) is not in mixedCase
Parameter MasterWombatV4.deposit(uint256,uint256)._pid (MasterWombatV4.sol#1119) is not in mixedCase
Parameter MasterWombatV4.deposit(uint256,uint256)._amount (MasterWombatV4.sol#1120) is not in mixedCase
Parameter MasterWombatV4.multiClaim(uint256[])._pids (MasterWombatV4.sol#1134) is not in mixedCase
Parameter MasterWombatV4.withdraw(uint256,uint256)._pid (MasterWombatV4.sol#1184) is not in mixedCase
Parameter MasterWombatV4.withdraw(uint256,uint256)._amount (MasterWombatV4.sol#1185) is not in mixedCase
Parameter MasterWombatV4.updateEmissionPartition(uint16)._basePartition (MasterWombatV4.sol#1237) is not in mixedCase
Parameter MasterWombatV4.rewarderBonusTokenInfo(uint256)._pid (MasterWombatV4.sol#1246) is not in mixedCase
Parameter MasterWombatV4.calRewardPerUnit(uint256)._pid (MasterWombatV4.sol#1282) is not in mixedCase
Parameter MasterWombatV4.pendingTokens(uint256,address)._pid (MasterWombatV4.sol#1305) is not in mixedCase
Parameter MasterWombatV4.pendingTokens(uint256,address)._user (MasterWombatV4.sol#1306) is not in mixedCase
Parameter MasterWombatV4.poolInfo(uint256)._pid (MasterWombatV4.sol#1335) is not in mixedCase
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#conformance-to-solidity-naming-conventions

Redundant expression "this (MasterWombatV4.sol#161)" inContext (MasterWombatV4.sol#156-164)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#redundant-statements

ERC20._name (MasterWombatV4.sol#632) should be immutable
ERC20._symbol (MasterWombatV4.sol#633) should be immutable
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#state-variables-that-could-be-declared-immutable
MasterWombatV4.sol analyzed (23 contracts with 84 detectors), 139 result(s) found
```

Slither log >> DynamicPoolV2.sol

```
PoolV2.setDev(address).dev_ (DynamicPoolV2.sol#1007) lacks a zero-check on :
                - dev = dev_ (DynamicPoolV2.sol#1009)
PoolV2.setMasterWombat(address).masterWombat_ (DynamicPoolV2.sol#1013) lacks a zero-check on :
                - masterWombat = masterWombat_ (DynamicPoolV2.sol#1015)
PoolV2.setFeeTo(address).feeTo_ (DynamicPoolV2.sol#1049) lacks a zero-check on :
                - feeTo = feeTo_ (DynamicPoolV2.sol#1051)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#missing-zero-address-validation

Reentrancy in PoolV2.fillPool(address,uint256) (DynamicPoolV2.sol#1080-1092):
        External calls:
        - asset.addCash(amount) (DynamicPoolV2.sol#1090)
        Event emitted after the call(s):
        - FillPool(token,amount) (DynamicPoolV2.sol#1091)
Reentrancy in PoolV2.setFee(uint256,uint256) (DynamicPoolV2.sol#1029-1036):
        External calls:
        - _mintAllFees() (DynamicPoolV2.sol#1032)
                - asset.transferUnderlyingToken(feeTo,dividend.fromWad(asset.underlyingTokenDecimals())) (DynamicPoolV2.sol#15
34)
                - asset.addLiability(liabilityToMint) (DynamicPoolV2.sol#1541)
                - asset.addCash(lpDividend) (DynamicPoolV2.sol#1542)
        Event emitted after the call(s):
        - SetFee(lpDividendRatio_,retentionRatio_) (DynamicPoolV2.sol#1035)
Reentrancy in PoolV2.transferTipBucket(address,uint256,address) (DynamicPoolV2.sol#1038-1047):
        External calls:
        - asset.transferUnderlyingToken(to,amount.fromWad(asset.underlyingTokenDecimals())) (DynamicPoolV2.sol#1045)
        Event emitted after the call(s):
        - TransferTipBucket(token,amount,to) (DynamicPoolV2.sol#1046)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#reentrancy-vulnerabilities-3

Address.verifyCallResult(bool,bytes,string) (DynamicPoolV2.sol#621-639) uses assembly
        - INLINE ASM (DynamicPoolV2.sol#631-634)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#assembly-usage
```

```
Pragma version^0.8.21 (DynamicPoolV2.sol#2) necessitates a version too recent to be trusted. Consider deploying with 0.6.12/0.
7.6/0.8.16
solc-0.8.21 is not recommended for deployment
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#incorrect-versions-of-solidity

Low level call in Address.sendValue(address,uint256) (DynamicPoolV2.sol#563-568):
        - (success) = recipient.call{value: amount}() (DynamicPoolV2.sol#566)
Low level call in Address.functionCallWithValue(address,bytes,uint256,string) (DynamicPoolV2.sol#590-601):
        - (success,returndata) = target.call{value: value}(data) (DynamicPoolV2.sol#599)
Low level call in Address.functionStaticCall(address,bytes,string) (DynamicPoolV2.sol#607-616):
        - (success,returndata) = target.staticcall(data) (DynamicPoolV2.sol#614)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#low-level-calls

Parameter CoreV2.depositRewardImpl(int256,int256,int256,int256,int256,int256).D (DynamicPoolV2.sol#205) is not in mixedCase
Parameter CoreV2.depositRewardImpl(int256,int256,int256,int256,int256,int256).SL (DynamicPoolV2.sol#206) is not in mixedCase
Parameter CoreV2.depositRewardImpl(int256,int256,int256,int256,int256,int256).delta_i (DynamicPoolV2.sol#207) is not in mixedC
ase
Parameter CoreV2.depositRewardImpl(int256,int256,int256,int256,int256,int256).A_i (DynamicPoolV2.sol#208) is not in mixedCase
Parameter CoreV2.depositRewardImpl(int256,int256,int256,int256,int256,int256).L_i (DynamicPoolV2.sol#209) is not in mixedCase
Parameter CoreV2.depositRewardImpl(int256,int256,int256,int256,int256,int256).A (DynamicPoolV2.sol#210) is not in mixedCase
Parameter CoreV2.withdrawalAmountInEquilImpl(int256,int256,int256,int256).delta_i (DynamicPoolV2.sol#224) is not in mixedCase
Parameter CoreV2.withdrawalAmountInEquilImpl(int256,int256,int256,int256).A_i (DynamicPoolV2.sol#225) is not in mixedCase
Parameter CoreV2.withdrawalAmountInEquilImpl(int256,int256,int256,int256).L_i (DynamicPoolV2.sol#226) is not in mixedCase
Parameter CoreV2.withdrawalAmountInEquilImpl(int256,int256,int256,int256).A (DynamicPoolV2.sol#227) is not in mixedCase
Parameter CoreV2.exactDepositLiquidityInEquilImpl(int256,int256,int256,int256).D_i (DynamicPoolV2.sol#238) is not in mixedCase
Parameter CoreV2.exactDepositLiquidityInEquilImpl(int256,int256,int256,int256).A_i (DynamicPoolV2.sol#239) is not in mixedCase
```

```
Function OwnableUpgradeable.__Ownable_init(address) (DynamicPoolV2.sol#736-738) is not in mixedCase
Function OwnableUpgradeable.__Ownable_init_unchained(address) (DynamicPoolV2.sol#740-744) is not in mixedCase
Constant OwnableUpgradeable.OwnableStorageLocation (DynamicPoolV2.sol#729) is not in UPPER_CASE_WITH_UNDERSCORES
Function ReentrancyGuardUpgradeable.__ReentrancyGuard_init() (DynamicPoolV2.sol#787-789) is not in mixedCase
Function ReentrancyGuardUpgradeable.__ReentrancyGuard_init_unchained() (DynamicPoolV2.sol#791-793) is not in mixedCase
Variable ReentrancyGuardUpgradeable.__gap (DynamicPoolV2.sol#815) is not in mixedCase
Function PausableUpgradeable.__Pausable_init() (DynamicPoolV2.sol#826-828) is not in mixedCase
Function PausableUpgradeable.__Pausable_init_unchained() (DynamicPoolV2.sol#830-832) is not in mixedCase
Variable PausableUpgradeable.__gap (DynamicPoolV2.sol#866) is not in mixedCase
Variable PoolV2._feeCollected (DynamicPoolV2.sol#906) is not in mixedCase
Variable PoolV2._assets (DynamicPoolV2.sol#908) is not in mixedCase
Variable PoolV2._used1 (DynamicPoolV2.sol#910) is not in mixedCase
Variable PoolV2._used2 (DynamicPoolV2.sol#911) is not in mixedCase
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#conformance-to-solidity-naming-conventions

Redundant expression "this (DynamicPoolV2.sol#409)" inContext (DynamicPoolV2.sol#404-412)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#redundant-statements

PoolV2._used1 (DynamicPoolV2.sol#910) is never used in DynamicPoolV2 (DynamicPoolV2.sol#1698-1730)
PoolV2._used2 (DynamicPoolV2.sol#911) is never used in DynamicPoolV2 (DynamicPoolV2.sol#1698-1730)
DynamicPoolV2.gap (DynamicPoolV2.sol#1702) is never used in DynamicPoolV2 (DynamicPoolV2.sol#1698-1730)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#unused-state-variable

PoolV2._used1 (DynamicPoolV2.sol#910) should be constant
PoolV2._used2 (DynamicPoolV2.sol#911) should be constant
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#state-variables-that-could-be-declared-constant
DynamicPoolV2.sol analyzed (21 contracts with 84 detectors), 93 result(s) found
```

Slither log >> HighCovRatioFeePoolV2.sol

```
HighCovRatioFeePoolV2.setCovRatioFeeParam(uint128,uint128) (HighCovRatioFeePoolV2.sol#1603-1608) should emit an event for:
        - startCovRatio = startCovRatio_ (HighCovRatioFeePoolV2.sol#1606)
        - endCovRatio = endCovRatio_ (HighCovRatioFeePoolV2.sol#1607)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#missing-events-arithmetic

PoolV2.setDev(address).dev_ (HighCovRatioFeePoolV2.sol#1019) lacks a zero-check on :
                - dev = dev_ (HighCovRatioFeePoolV2.sol#1021)
PoolV2.setMasterWombat(address).masterWombat_ (HighCovRatioFeePoolV2.sol#1025) lacks a zero-check on :
                - masterWombat = masterWombat_ (HighCovRatioFeePoolV2.sol#1027)
PoolV2.setFeeTo(address).feeTo_ (HighCovRatioFeePoolV2.sol#1065) lacks a zero-check on :
                - feeTo = feeTo_ (HighCovRatioFeePoolV2.sol#1067)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#missing-zero-address-validation

Reentrancy in PoolV2.fillPool(address,uint256) (HighCovRatioFeePoolV2.sol#1096-1109):
        External calls:
        - asset.addCash(amount) (HighCovRatioFeePoolV2.sol#1107)
        Event emitted after the call(s):
        - FillPool(token,amount) (HighCovRatioFeePoolV2.sol#1108)
Reentrancy in PoolV2.setFee(uint256,uint256) (HighCovRatioFeePoolV2.sol#1043-1051):
        External calls:
        - _mintAllFees() (HighCovRatioFeePoolV2.sol#1047)
                - asset.transferUnderlyingToken(feeTo,dividend.fromWad(asset.underlyingTokenDecimals())) (HighCovRatioFeePoolV
2.sol#1558)
                - asset.addLiability(liabilityToMint) (HighCovRatioFeePoolV2.sol#1565)
                - asset.addCash(lpDividend) (HighCovRatioFeePoolV2.sol#1566)
        Event emitted after the call(s):
        - SetFee(lpDividendRatio_,retentionRatio_) (HighCovRatioFeePoolV2.sol#1050)
Reentrancy in PoolV2.transferTipBucket(address,uint256,address) (HighCovRatioFeePoolV2.sol#1053-1063):
        External calls:
        - asset.transferUnderlyingToken(to,amount.fromWad(asset.underlyingTokenDecimals())) (HighCovRatioFeePoolV2.sol#1061)
        Event emitted after the call(s):
        - TransferTipBucket(token,amount,to) (HighCovRatioFeePoolV2.sol#1062)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#reentrancy-vulnerabilities-3
```

```
Address.verifyCallResult(bool,bytes,string) (HighCovRatioFeePoolV2.sol#626-644) uses assembly
        - INLINE ASM (HighCovRatioFeePoolV2.sol#636-639)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#assembly-usage

Address.functionCall(address,bytes) (HighCovRatioFeePoolV2.sol#575-577) is never used and should be removed
Address.functionCallWithValue(address,bytes,uint256) (HighCovRatioFeePoolV2.sol#587-593) is never used and should be removed
Address.functionStaticCall(address,bytes) (HighCovRatioFeePoolV2.sol#608-610) is never used and should be removed
Address.functionStaticCall(address,bytes,string) (HighCovRatioFeePoolV2.sol#612-621) is never used and should be removed
Address.sendValue(address,uint256) (HighCovRatioFeePoolV2.sol#568-573) is never used and should be removed
```

```
Pragma version^0.8.21 (HighCovRatioFeePoolV2.sol#2) necessitates a version too recent to be trusted. Consider deploying with 0
.6.12/0.7.6/0.8.16
solc-0.8.21 is not recommended for deployment
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#incorrect-versions-of-solidity

Low level call in Address.sendValue(address,uint256) (HighCovRatioFeePoolV2.sol#568-573):
        - (success) = recipient.call{value: amount}() (HighCovRatioFeePoolV2.sol#571)
Low level call in Address.functionCallWithValue(address,bytes,uint256,string) (HighCovRatioFeePoolV2.sol#595-606):
        - (success,returndata) = target.call{value: value}(data) (HighCovRatioFeePoolV2.sol#604)
Low level call in Address.functionStaticCall(address,bytes,string) (HighCovRatioFeePoolV2.sol#612-621):
        - (success,returndata) = target.staticcall(data) (HighCovRatioFeePoolV2.sol#619)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#low-level-calls
```

```
Redundant expression "this (HighCovRatioFeePoolV2.sol#413)" inContext (HighCovRatioFeePoolV2.sol#407-416)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#redundant-statements

PoolV2._used1 (HighCovRatioFeePoolV2.sol#915) is never used in HighCovRatioFeePoolV2 (HighCovRatioFeePoolV2.sol#1588-1716)
PoolV2._used2 (HighCovRatioFeePoolV2.sol#916) is never used in HighCovRatioFeePoolV2 (HighCovRatioFeePoolV2.sol#1588-1716)
HighCovRatioFeePoolV2.gap (HighCovRatioFeePoolV2.sol#1594) is never used in HighCovRatioFeePoolV2 (HighCovRatioFeePoolV2.sol#1
588-1716)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#unused-state-variable

PoolV2._used1 (HighCovRatioFeePoolV2.sol#915) should be constant
PoolV2._used2 (HighCovRatioFeePoolV2.sol#916) should be constant
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#state-variables-that-could-be-declared-constant
HighCovRatioFeePoolV2.sol analyzed (19 contracts with 84 detectors), 90 result(s) found
```

Slither log >> DynamicPool.sol

```
Pool.setDev(address).dev_ (DynamicPool.sol#1315) lacks a zero-check on :
                - dev = dev_ (DynamicPool.sol#1317)
Pool.setMasterWombat(address).masterWombat_ (DynamicPool.sol#1321) lacks a zero-check on :
                - masterWombat = masterWombat_ (DynamicPool.sol#1323)
Pool.setFeeTo(address).feeTo_ (DynamicPool.sol#1360) lacks a zero-check on :
                - feeTo = feeTo_ (DynamicPool.sol#1362)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#missing-zero-address-validation

Reentrancy in Pool.fillPool(address,uint256) (DynamicPool.sol#2013-2027):
        External calls:
        - asset.addCash(amount) (DynamicPool.sol#2025)
        Event emitted after the call(s):
        - FillPool(token,amount) (DynamicPool.sol#2026)
Reentrancy in Pool.setFee(uint256,uint256) (DynamicPool.sol#1347-1353):
        External calls:
        - mintAllFee() (DynamicPool.sol#1349)
                - asset.transferUnderlyingToken(feeTo,dividend.fromWad(asset.underlyingTokenDecimals())) (DynamicPool.sol#2080
)
                - asset.addLiability(liabilityToMint) (DynamicPool.sol#2090)
                - asset.addCash(lpDividend) (DynamicPool.sol#2091)
        Event emitted after the call(s):
        - SetFee(lpDividendRatio_,retentionRatio_) (DynamicPool.sol#1352)
Reentrancy in Pool.transferTipBucket(address,uint256,address) (DynamicPool.sol#2030-2041):
        External calls:
        - asset.transferUnderlyingToken(to,amount.fromWad(asset.underlyingTokenDecimals())) (DynamicPool.sol#2039)
        Event emitted after the call(s):
        - TransferTipBucket(token,amount,to) (DynamicPool.sol#2040)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#reentrancy-vulnerabilities-3

Address.verifyCallResult(bool,bytes,string) (DynamicPool.sol#763-781) uses assembly
        - INLINE ASM (DynamicPool.sol#773-776)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#assembly-usage
```

```
Pragma version^0.8.21 (DynamicPool.sol#2) necessitates a version too recent to be trusted. Consider deploying with 0.6.12/0.7.
6/0.8.16
solc-0.8.21 is not recommended for deployment
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#incorrect-versions-of-solidity

Low level call in Address.sendValue(address,uint256) (DynamicPool.sol#705-710):
        - (success) = recipient.call{value: amount}() (DynamicPool.sol#708)
Low level call in Address.functionCallWithValue(address,bytes,uint256,string) (DynamicPool.sol#732-743):
        - (success,returndata) = target.call{value: value}(data) (DynamicPool.sol#741)
Low level call in Address.functionStaticCall(address,bytes,string) (DynamicPool.sol#749-758):
        - (success,returndata) = target.staticcall(data) (DynamicPool.sol#756)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#low-level-calls
```

```
Function OwnableUpgradeable.__Ownable_init_unchained(address) (DynamicPool.sol#886-891) is not in mixedCase
Constant OwnableUpgradeable.OwnableStorageLocation (DynamicPool.sol#873) is not in UPPER_CASE_WITH_UNDERSCORES
Function ReentrancyGuardUpgradeable.__ReentrancyGuard_init() (DynamicPool.sol#983-985) is not in mixedCase
Function ReentrancyGuardUpgradeable.__ReentrancyGuard_init_unchained() (DynamicPool.sol#987-989) is not in mixedCase
Variable ReentrancyGuardUpgradeable.__gap (DynamicPool.sol#1031) is not in mixedCase
Function PausableUpgradeable.__Pausable_init() (DynamicPool.sol#1051-1053) is not in mixedCase
Function PausableUpgradeable.__Pausable_init_unchained() (DynamicPool.sol#1055-1057) is not in mixedCase
Variable PausableUpgradeable.__gap (DynamicPool.sol#1133) is not in mixedCase
Variable Pool._feeCollected (DynamicPool.sol#1194) is not in mixedCase
Variable Pool._assets (DynamicPool.sol#1197) is not in mixedCase
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#conformance-to-solidity-naming-conventions

Redundant expression "this (DynamicPool.sol#588)" inContext (DynamicPool.sol#582-591)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#redundant-statements
DynamicPool.sol analyzed (20 contracts with 84 detectors), 80 result(s) found
```

Slither log >> HighCovRatioFeePool.sol

```
HighCovRatioFeePool.setCovRatioFeeParam(uint128,uint128) (HighCovRatioFeePool.sol#2129-2134) should emit an event for:
        - startCovRatio = startCovRatio_ (HighCovRatioFeePool.sol#2132)
        - endCovRatio = endCovRatio_ (HighCovRatioFeePool.sol#2133)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#missing-events-arithmetic

Pool.setDev(address).dev_ (HighCovRatioFeePool.sol#1316) lacks a zero-check on :
                - dev = dev_ (HighCovRatioFeePool.sol#1318)
Pool.setMasterWombat(address).masterWombat_ (HighCovRatioFeePool.sol#1322) lacks a zero-check on :
                - masterWombat = masterWombat_ (HighCovRatioFeePool.sol#1324)
Pool.setFeeTo(address).feeTo_ (HighCovRatioFeePool.sol#1361) lacks a zero-check on :
                - feeTo = feeTo_ (HighCovRatioFeePool.sol#1363)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#missing-zero-address-validation

Reentrancy in Pool.fillPool(address,uint256) (HighCovRatioFeePool.sol#2014-2028):
        External calls:
        - asset.addCash(amount) (HighCovRatioFeePool.sol#2026)
        Event emitted after the call(s):
        - FillPool(token,amount) (HighCovRatioFeePool.sol#2027)
Reentrancy in Pool.setFee(uint256,uint256) (HighCovRatioFeePool.sol#1348-1354):
        External calls:
        - mintAllFee() (HighCovRatioFeePool.sol#1350)
                - asset.transferUnderlyingToken(feeTo,dividend.fromWad(asset.underlyingTokenDecimals())) (HighCovRatioFeePool.
sol#2081)
                - asset.addLiability(liabilityToMint) (HighCovRatioFeePool.sol#2091)
                - asset.addCash(lpDividend) (HighCovRatioFeePool.sol#2092)
        Event emitted after the call(s):
        - SetFee(lpDividendRatio_,retentionRatio_) (HighCovRatioFeePool.sol#1353)
Reentrancy in Pool.transferTipBucket(address,uint256,address) (HighCovRatioFeePool.sol#2031-2042):
        External calls:
        - asset.transferUnderlyingToken(to,amount.fromWad(asset.underlyingTokenDecimals())) (HighCovRatioFeePool.sol#2040)
        Event emitted after the call(s):
        - TransferTipBucket(token,amount,to) (HighCovRatioFeePool.sol#2041)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#reentrancy-vulnerabilities-3
```

```
Address.verifyCallResult(bool,bytes,string) (HighCovRatioFeePool.sol#766-784) uses assembly
        - INLINE ASM (HighCovRatioFeePool.sol#776-779)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#assembly-usage
```

```
Pragma version^0.8.5 (HighCovRatioFeePool.sol#2) allows old versions
solc-0.8.21 is not recommended for deployment
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#incorrect-versions-of-solidity

Low level call in Address.sendValue(address,uint256) (HighCovRatioFeePool.sol#708-713):
        - (success) = recipient.call{value: amount}() (HighCovRatioFeePool.sol#711)
Low level call in Address.functionCallWithValue(address,bytes,uint256,string) (HighCovRatioFeePool.sol#735-746):
        - (success,returndata) = target.call{value: value}(data) (HighCovRatioFeePool.sol#744)
Low level call in Address.functionStaticCall(address,bytes,string) (HighCovRatioFeePool.sol#752-761):
        - (success,returndata) = target.staticcall(data) (HighCovRatioFeePool.sol#759)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#low-level-calls
```

```
Constant OwnableUpgradeable.OwnableStorageLocation (HighCovRatioFeePool.sol#874) is not in UPPER_CASE_WITH_UNDERSCORES
Function ReentrancyGuardUpgradeable.__ReentrancyGuard_init() (HighCovRatioFeePool.sol#984-986) is not in mixedCase
Function ReentrancyGuardUpgradeable.__ReentrancyGuard_init_unchained() (HighCovRatioFeePool.sol#988-990) is not in mixedCase
Variable ReentrancyGuardUpgradeable.__gap (HighCovRatioFeePool.sol#1032) is not in mixedCase
Function PausableUpgradeable.__Pausable_init() (HighCovRatioFeePool.sol#1052-1054) is not in mixedCase
Function PausableUpgradeable.__Pausable_init_unchained() (HighCovRatioFeePool.sol#1056-1058) is not in mixedCase
Variable PausableUpgradeable.__gap (HighCovRatioFeePool.sol#1134) is not in mixedCase
Variable Pool._feeCollected (HighCovRatioFeePool.sol#1195) is not in mixedCase
Variable Pool._assets (HighCovRatioFeePool.sol#1198) is not in mixedCase
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#conformance-to-solidity-naming-conventions

Redundant expression "this (HighCovRatioFeePool.sol#591)" inContext (HighCovRatioFeePool.sol#585-594)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#redundant-statements
HighCovRatioFeePool.sol analyzed (19 contracts with 84 detectors), 80 result(s) found
```

Slither log >> Pool.sol

```
Pool.setDev(address).dev_ (Pool.sol#1311) lacks a zero-check on :
                - dev = dev_ (Pool.sol#1313)
Pool.setMasterWombat(address).masterWombat_ (Pool.sol#1317) lacks a zero-check on :
                - masterWombat = masterWombat_ (Pool.sol#1319)
Pool.setFeeTo(address).feeTo_ (Pool.sol#1356) lacks a zero-check on :
                - feeTo = feeTo_ (Pool.sol#1358)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#missing-zero-address-validation

Reentrancy in Pool.fillPool(address,uint256) (Pool.sol#2009-2023):
        External calls:
        - asset.addCash(amount) (Pool.sol#2021)
        Event emitted after the call(s):
        - FillPool(token,amount) (Pool.sol#2022)
Reentrancy in Pool.setFee(uint256,uint256) (Pool.sol#1343-1349):
        External calls:
        - mintAllFee() (Pool.sol#1345)
                - asset.transferUnderlyingToken(feeTo,dividend.fromWad(asset.underlyingTokenDecimals())) (Pool.sol#2076)
                - asset.addLiability(liabilityToMint) (Pool.sol#2086)
                - asset.addCash(lpDividend) (Pool.sol#2087)
        Event emitted after the call(s):
        - SetFee(lpDividendRatio_,retentionRatio_) (Pool.sol#1348)
Reentrancy in Pool.transferTipBucket(address,uint256,address) (Pool.sol#2026-2037):
        External calls:
        - asset.transferUnderlyingToken(to,amount.fromWad(asset.underlyingTokenDecimals())) (Pool.sol#2035)
        Event emitted after the call(s):
        - TransferTipBucket(token,amount,to) (Pool.sol#2036)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#reentrancy-vulnerabilities-3

Address.verifyCallResult(bool,bytes,string) (Pool.sol#767-785) uses assembly
        - INLINE ASM (Pool.sol#777-780)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#assembly-usage
```

```
Pragma version^0.8.5 (Pool.sol#2) allows old versions
solc-0.8.21 is not recommended for deployment
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#incorrect-versions-of-solidity

Low level call in Address.sendValue(address,uint256) (Pool.sol#709-714):
        - (success) = recipient.call{value: amount}() (Pool.sol#712)
Low level call in Address.functionCallWithValue(address,bytes,uint256,string) (Pool.sol#736-747):
        - (success,returndata) = target.call{value: value}(data) (Pool.sol#745)
Low level call in Address.functionStaticCall(address,bytes,string) (Pool.sol#753-762):
        - (success,returndata) = target.staticcall(data) (Pool.sol#760)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#low-level-calls
```

```
Parameter CoreV2.withdrawalAmountInEquilImpl(int256,int256,int256,int256).A (Pool.sol#328) is not in mixedCase
Parameter CoreV2.exactDepositLiquidityInEquilImpl(int256,int256,int256,int256).D_i (Pool.sol#342) is not in mixedCase
Parameter CoreV2.exactDepositLiquidityInEquilImpl(int256,int256,int256,int256).A_i (Pool.sol#343) is not in mixedCase
Parameter CoreV2.exactDepositLiquidityInEquilImpl(int256,int256,int256,int256).L_i (Pool.sol#344) is not in mixedCase
Parameter CoreV2.exactDepositLiquidityInEquilImpl(int256,int256,int256,int256).A (Pool.sol#345) is not in mixedCase
Function ContextUpgradeable.__Context_init() (Pool.sol#854-855) is not in mixedCase
Function ContextUpgradeable.__Context_init_unchained() (Pool.sol#857-858) is not in mixedCase
Variable ContextUpgradeable.__gap (Pool.sol#867) is not in mixedCase
Function OwnableUpgradeable.__Ownable_init(address) (Pool.sol#885-887) is not in mixedCase
Function OwnableUpgradeable.__Ownable_init_unchained(address) (Pool.sol#889-894) is not in mixedCase
Constant OwnableUpgradeable.OwnableStorageLocation (Pool.sol#876) is not in UPPER_CASE_WITH_UNDERSCORES
Function ReentrancyGuardUpgradeable.__ReentrancyGuard_init() (Pool.sol#986-988) is not in mixedCase
Function ReentrancyGuardUpgradeable.__ReentrancyGuard_init_unchained() (Pool.sol#990-992) is not in mixedCase
Variable ReentrancyGuardUpgradeable.__gap (Pool.sol#1034) is not in mixedCase
Function PausableUpgradeable.__Pausable_init() (Pool.sol#1054-1056) is not in mixedCase
Function PausableUpgradeable.__Pausable_init_unchained() (Pool.sol#1058-1060) is not in mixedCase
Variable PausableUpgradeable.__gap (Pool.sol#1136) is not in mixedCase
Variable Pool._feeCollected (Pool.sol#1190) is not in mixedCase
Variable Pool._assets (Pool.sol#1193) is not in mixedCase
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#conformance-to-solidity-naming-conventions

Redundant expression "this (Pool.sol#590)" inContext (Pool.sol#584-593)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#redundant-statements
Pool.sol analyzed (18 contracts with 84 detectors), 78 result(s) found
```

Solidity Static Analysis

ABnbcAsset.sol

Gas costs:

Gas requirement of function ABnbcAsset.underlyingToken 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: 22:4:

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: 78:8:

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: 33:15:

Asset.sol

Gas costs:

Gas requirement of function Asset.setPool 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: 77:4:

Gas costs:

Gas requirement of function Asset.burn 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: 136:4:

This on local calls:

Use of "this" for local functions: Never use "this" to call functions in the same contract, it only consumes more gas than normal local calls.
more
Pos: 126:29:

Similar variable names:

Asset.setPool(address) : Variables have very similar names "pool" and "pool_".
Note: Modifiers are currently not considered by this static analysis.
Pos: 79:27:

No return:

IAsset.pool(): Defines a return type but never explicitly returns a value.
Pos: 11:4:

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: 155:8:

BnbxAsset.sol

Gas costs:

Gas requirement of function BnbxAsset.getRelativePrice 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: 23:4:

This on local calls:

Use of "this" for local functions: Never use "this" to call functions in the same contract, it only consumes more gas than normal local calls.
more
Pos: 126:29:

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: 78:8:

DynamicAsset.sol

Gas costs:

Gas requirement of function DynamicAsset.removeLiability 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: 174:4:

This on local calls:

Use of "this" for local functions: Never use "this" to call functions in the same contract, it only consumes more gas than normal local calls.
more
Pos: 126:29:

Gas costs:

Gas requirement of function DynamicAsset.removeLiability 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: 174:4:

This on local calls:

Use of "this" for local functions: Never use "this" to call functions in the same contract, it only consumes more gas than normal local calls.
more
Pos: 126:29:

StkbnbAsset.sol

Gas costs:

Gas requirement of function DynamicAsset.removeLiability 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: 174:4:

This on local calls:

Use of "this" for local functions: Never use "this" to call functions in the same contract, it only consumes more gas than normal local calls.
more
Pos: 126:29:

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: 39:15:

CoreV2.sol

Constant/View/Pure functions:

CoreV2._swapQuoteFunc(int256,int256,int256,int256,int256,int256) : Is constant but potentially should not be.

more

Pos: 31:4:

Similar variable names:

CoreV2._swapQuoteFunc(int256,int256,int256,int256,int256,int256) : Variables have very similar names "WAD" and "Ay".

Pos: 43:24:

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: 85:8:

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: 43:36:

DynamicPoolV2.sol

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: 151:23:

Gas costs:

Gas requirement of function DynamicPoolV2.globalEquilCovRatio 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: 922:4:

Similar variable names:

DynamicPoolV2._globalInvariantFunc() : Variables have very similar names "A_i" and "L_i". Note: Modifiers are currently not considered by this static analysis.
Pos: 51:16:

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: 85:8:

Result not used:

A binary operation yields a value that is not used further. This is often caused by confusing assignment (=) and comparison (==).
Pos: 342:12:

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: 213:8:

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: 151:23:

Gas costs:

Gas requirement of function HighCovRatioFeePoolV2.initialize 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: 23:4:

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: 85:8:

Result not used:

A binary operation yields a value that is not used further. This is often caused by confusing assignment (=) and comparison (==).
Pos: 342:12:

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: 43:36:

PausableAssets.sol

Constant/View/Pure functions:

PausableAssets.requireAssetNotPaused(address) : Is constant but potentially should not be.
more
Pos: 32:4:

Constant/View/Pure functions:

PausableAssets.requireAssetPaused(address) : Is constant but potentially should not be.
more
Pos: 43:4:

PoolV2.sol

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: 151:23:

Gas costs:

Gas requirement of function PoolV2.unpauseAsset 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: 329:4:

Similar variable names:

PoolV2.addAsset(address,address) : Variables have very similar names "_assets" and "asset". Note: Modifiers are currently not considered by this static analysis.
Pos: 192:8:

Result not used:

A binary operation yields a value that is not used further. This is often caused by confusing assignment (=) and comparison (==).
Pos: 342:12:

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: 206:8:

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: 72:20:

MasterWombatV4.sol

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: 513:29:

Gas costs:

Gas requirement of function MasterWombatV4.massUpdatePools 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: 220:4:

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: 346:8:

Similar variable names:

MasterWombatV4.add(uint256,contract IERC20,contract IMultiRewarder) : Variables have very similar names "lpTokens" and "_lpToken". Note: Modifiers are currently not considered by this static analysis.

Pos: 163:43:

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: 163:8:

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: 27:24:

MultiRewarderPerSec.sol

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: 27:24:

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: 245:38:

Low level calls:

Use of "call": should be avoided whenever possible. It can lead to unexpected behavior if return value is not handled properly. Please use Direct Calls via specifying the called contract's interface.
more
Pos: 191:43:

Gas costs:

Gas requirement of function MultiRewarderPerSec.updateReward 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:

Constant/View/Pure functions:

MultiRewarderPerSec.pendingTokens(address) : Is constant but potentially should not be. Note: Modifiers are currently not considered by this static analysis.
more
Pos: 229:4:

No return:

IMultiRewarder.rewardLength(): Defines a return type but never explicitly returns a value.
Pos: 15:4:

Similar variable names:

MultiRewarderPerSec._onReward(address,uint256) : Variables have very similar names "reward" and "rewards". Note: Modifiers are currently not considered by this static analysis.
Pos: 213:55:

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: 66:8:

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: 294:12:

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: 135:53:

DynamicPool.sol

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: 144:23:

Gas costs:

Gas requirement of function DynamicPool.quotePotentialWithdraw 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: 618:4:

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: 80:8:

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: 293:8:

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: 72:20:

HighCovRatioFeePool.sol

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: 144:23:

Gas costs:

Gas requirement of function HighCovRatioFeePool.quotePotentialWithdrawFromOtherAsset 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: 135:4:

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: 43:53:

Pool.sol

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: 144:23:

Gas costs:

Gas requirement of function Pool.mintFee 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: 1000:4:

Similar variable names:

Pool.setFee(uint256,uint256) : Variables have very similar names "lpDividendRatio" and "lpDividendRatio_". Note: Modifiers are currently not considered by this static analysis.
Pos: 242:8:

Solhint Linter

ABnbcAsset.sol

```
Compiler version ^0.8.5 does not satisfy the ^0.5.8 semver
requirement
Pos: 1:1
global import of path IRelativePriceProvider.sol is not allowed.
Specify names to import individually or bind all exports of the
module into a name (import "path" as Name)
Pos: 1:3
Use double quotes for string literals
Pos: 8:4
Explicitly mark visibility of state
Pos: 5:17
Explicitly mark visibility in function (Set ignoreConstructors to
true if using solidity >=0.7.0)
Pos: 5:19
```

Asset.sol

```
Compiler version ^0.8.5 does not satisfy the ^0.5.8 semver
requirement
Pos: 1:1
Use double quotes for string literals
Pos: 8:3
global import of path @openzeppelin/contracts/access/Ownable.sol is
not allowed. Specify names to import individually or bind all exports
of the module into a name (import "path" as Name)
Pos: 1:4
Explicitly mark visibility in function (Set ignoreConstructors to
true if using solidity >=0.7.0)
Pos: 5:62
Error message for require is too long
Pos: 9:77
Use double quotes for string literals
Pos: 33:154
Use double quotes for string literals
Pos: 38:174
```

BnbxAsset.sol

```
Compiler version ^0.8.5 does not satisfy the ^0.5.8 semver
requirement
Pos: 1:1
global import of path DynamicAsset.sol is not allowed. Specify names
to import individually or bind all exports of the module into a name
(import "path" as Name)
```

```
Pos: 1:4
Use double quotes for string literals
Pos: 8:4
Explicitly mark visibility of state
Pos: 5:19
Explicitly mark visibility in function (Set ignoreConstructors to
true if using solidity >=0.7.0)
Pos: 5:21
```

DynamicAsset.sol

```
Compiler version ^0.8.5 does not satisfy the ^0.5.8 semver
requirement
Pos: 1:1
global import of path IRelativePriceProvider.sol is not allowed.
Specify names to import individually or bind all exports of the
module into a name (import "path" as Name)
Pos: 1:3
Use double quotes for string literals
Pos: 8:3
Explicitly mark visibility in function (Set ignoreConstructors to
true if using solidity >=0.7.0)
Pos: 5:13
Code contains empty blocks
Pos: 47:17
```

StkbnbAsset.sol

```
Compiler version ^0.8.5 does not satisfy the ^0.5.8 semver
requirement
Pos: 1:1
global import of path IRelativePriceProvider.sol is not allowed.
Specify names to import individually or bind all exports of the
module into a name (import "path" as Name)
Pos: 1:3
Use double quotes for string literals
Pos: 8:3
Explicitly mark visibility of state
Pos: 5:22
Explicitly mark visibility in function (Set ignoreConstructors to
true if using solidity >=0.7.0)
Pos: 5:24
```

CoreV2.sol

```
Compiler version ^0.8.5 does not satisfy the ^0.5.8 semver
requirement
Pos: 1:1
global import of path DSMath.sol is not allowed. Specify names to
```

```
import individually or bind all exports of the module into a name
(import "path" as Name)
Pos: 1:3
Use double quotes for string literals
Pos: 8:3
global import of path SignedSafeMath.sol is not allowed. Specify
names to import individually or bind all exports of the module into a
name (import "path" as Name)
Pos: 1:4
Use double quotes for string literals
Pos: 8:4
Avoid to use letters 'I', 'l', 'O' as identifiers
Pos: 9:158
Variable name must be in mixedCase
Pos: 39:180
Variable name must be in mixedCase
Pos: 50:180
```

DynamicPoolV2.sol

```
Compiler version ^0.8.6 does not satisfy the ^0.5.8 semver
requirement
Pos: 1:1
global import of path DSMath.sol is not allowed. Specify names to
import individually or bind all exports of the module into a name
(import "path" as Name)
Pos: 1:3
Variable name must be in mixedCase
Pos: 13:47
Variable name must be in mixedCase
Pos: 13:55
```

HighCovRatioFeePoolV2.sol

```
Compiler version ^0.8.5 does not satisfy the ^0.5.8 semver
requirement
Pos: 1:1
global import of path DSMath.sol is not allowed. Specify names to
import individually or bind all exports of the module into a name
(import "path" as Name)
Pos: 1:3
Use double quotes for string literals
Pos: 8:4
Variable "fromAsset" is unused
Pos: 9:116
Variable "toAsset" is unused
Pos: 9:117
```

PausableAssets.sol

```
Compiler version ^0.8.5 does not satisfy the ^0.5.8 semver
requirement
Pos: 1:1
```

PoolV2.sol

```
Compiler version ^0.8.5 does not satisfy the ^0.5.8 semver
requirement
Pos: 1:1
global import of path
@openzeppelin/contracts-upgradeable/security/PausableUpgradeable.sol
is not allowed. Specify names to import individually or bind all
exports of the module into a name (import "path" as Name)
Pos: 1:6
Code contains empty blocks
Pos: 27:493
Variable "minimumLiquidity" is unused
Pos: 9:480
Code contains empty blocks
Pos: 74:520
Variable "minimumAmount" is unused
Pos: 9:626
Variable "minimumToAmount" is unused
Pos: 9:834
Variable name must be in mixedCase
Pos: 13:943
Variable name must be in mixedCase
Pos: 13:951
Code contains empty blocks
Pos: 31:971
```

MasterWombatV4.sol

```
Compiler version ^0.8.5 does not satisfy the ^0.5.8 semver
requirement
Pos: 1:1
global import of path
@openzeppelin/contracts-upgradeable/security/ReentrancyGuardUpgradeab
le.sol is not allowed. Specify names to import individually or bind
all exports of the module into a name (import "path" as Name)
Pos: 1:4
Use double quotes for string literals
Pos: 8:4
global import of path IVeWom.sol is not allowed. Specify names to
import individually or bind all exports of the module into a name
(import "path" as Name)
Pos: 1:12
Use double quotes for string literals
```

```
Pos: 8:12Error message for require is too long
Pos: 9:201
Use double quotes for string literals
Pos: 13:203
Provide an error message for require
Pos: 9:454
Use double quotes for string literals
Pos: 40:480
Avoid making time-based decisions in your business logic
Pos: 16:502
Variable "_periodFinish" is unused
Pos: 39:501
Avoid making time-based decisions in your business logic
Pos: 30:512
```

MultiRewarderPerSec.sol

```
Compiler version ^0.8.5 does not satisfy the ^0.5.8 semver
requirement
Pos: 1:1
global import of path @openzeppelin/contracts/utils/Address.sol is
not allowed. Specify names to import individually or bind all exports
of the module into a name (import "path" as Name)
Pos: 1:3
Avoid making time-based decisions in your business logic
Pos: 17:243
Avoid making time-based decisions in your business logic
Pos: 39:244
Use double quotes for string literals
Pos: 78:292
Use double quotes for string literals
Pos: 30:293
```

DynamicPool.sol

```
Compiler version ^0.8.6 does not satisfy the ^0.5.8 semver
requirement
Pos: 1:1
global import of path DSMath.sol is not allowed. Specify names to
import individually or bind all exports of the module into a name
(import "path" as Name)
Pos: 1:3
Use double quotes for string literals
Pos: 8:3
global import of path Pool.sol is not allowed. Specify names to
import individually or bind all exports of the module into a name
(import "path" as Name)
Pos: 1:5
Variable name must be in mixedCase
Pos: 13:42
Variable name must be in mixedCase
Pos: 13:50
```

HighCovRatioFeePool.sol

```
Compiler version ^0.8.5 does not satisfy the ^0.5.8 semver
requirement
Pos: 1:1
global import of path DSMath.sol is not allowed. Specify names to
import individually or bind all exports of the module into a name
(import "path" as Name)
Pos: 1:3
Use double quotes for string literals
Pos: 8:3
global import of path Pool.sol is not allowed. Specify names to
import individually or bind all exports of the module into a name
(import "path" as Name)
Pos: 1:4
Use double quotes for string literals
Pos: 8:4
```

Pool.sol

```
Compiler version ^0.8.5 does not satisfy the ^0.5.8 semver
requirement
Pos: 1:1
global import of path
@openzeppelin/contracts-upgradeable/access/OwnableUpgradeable.sol is
not allowed. Specify names to import individually or bind all exports
of the module into a name (import "path" as Name)
Pos: 1:3
Use double quotes for string literals
Pos: 8:3
global import of path
@openzeppelin/contracts-upgradeable/proxy/utils/Initializable.sol is
not allowed. Specify names to import individually or bind all exports
of the module into a name (import "path" as Name)
Pos: 1:4
Variable name must be in mixedCase
Pos: 13:941
Variable name must be in mixedCase
Pos: 13:949
```

Software analysis result:

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

Email: audit@EtherAuthority.io