Ether Authority

www.EtherAuthority.io
audit@etherauthority.io

SMART CONTRACT

Security Audit Report

Project: Scrub-Finance Protocol
Platform: Cronos Blockchain
Language: Solidity
Date: April 18th, 2022

Table of contents

Introduction 4
Project Background 4
Audit Scope 5
Claimed Smart Contract Features 6
Audit Summary 8
Technical Quick Stats 9
Code Quality 10
Documentation 10
Use of Dependencies 10
AS-IS overview 11
Severity Definitions 20
Audit Findings 21
Conclusion 29
Our Methodology 30
Disclaimers 32
Appendix

- Code Flow Diagram 33
- Slither Results Log 43
- Solidity static analysis 51
- Solhint Linter 65

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

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.

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

Introduction

EtherAuthority was contracted by Scrub-Finance to perform the Security audit of the Scrub-Finance Protocol smart contracts code. The audit has been performed using manual analysis as well as using automated software tools. This report presents all the findings regarding the audit performed on April 18th, 2022.

The purpose of this audit was to address the following:

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

Project Background

The Scrub-Finance Contracts have functions like burn, mint, stake, setTaxTiersRate, setTaxTiersTwapenableAutoCalculateTax, setBurnTax, buyBonds, setOraclesetLockUp, withdraw,claimReward, setScrub, setBearOracle, setEpoch, twap, OpenTrade, exit, claimReward, etc. The Scrub-Finance contract inherits the SafeMath, ERC20Burnable, Math, SafeERC20, Address, IERC20, Ownable standard smart contracts from the OpenZeppelin library. These OpenZeppelin contracts are considered community-audited and time-tested, and hence are not part of the audit scope.

Audit scope

Name	**Code Review and Security Analysis Report for Scrub-Finance Protocol Smart Contracts**
Platform	**Cronos / Solidity**
File 1	BBond.sol
File 1 MD5 Hash	F9FFE0D5221DD35ECA5F77EAB915AF98
File 2	Bear.sol
File 2 MD5 Hash	AE82BF4B25B372858EBE7A5EEFA089B7
File 3	BearScrub.sol
File 3 MD5 Hash	E60D0930A85D33D2B430C1A248A4874E

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

File 4	BearTreasury.sol
File 4 MD5 Hash	B741E67511B83B73DEBB628414919D59
File 5	LionOracle.sol
File 5 MD5 Hash	FB0A6BD007CFC9DC0940D6B6224BB989
File 6	RewardManager.sol
File 6 MD5 Hash	9F0DD1DC0D5A87FFB9C43E537FA20C94
File 7	Tiger.sol
File 7 MD5 Hash	3535D3002619F1468C5C1D16DA51A8CC
File 8	UserVault.sol
File 8 MD5 Hash	763125AEC6DD5FCAB51AC9C6D29B8570
File 9	SaleBatch.sol
File 9 MD5 Hash	6AA52F732489F818F984A354E0D0273C
File 10	Zap.sol
File 10 MD5 Hash	9E3C51F160113FBD5516D5765DCDD902
Audit Date	April 18th,2022

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

Claimed Smart Contract Features

Claimed Feature Detail	Our Observation
File 1 BBond.sol • Name: Bear Bonds • Symbol: BBOND • Decimals: 18	**YES, This is valid.**
File 2 Bear.sol • Name: BEAR • Symbol: BEAR • Decimals: 18 • Initial distribution for the MMF launchpad: 0.4 Million BEAR • Get Tax Tiers Rates Count: 14 • Get Tax Tiers Twaps Count: 14 • Burn Threshold: 1,100 • Total Supply: 400,001 BEAR	**YES, This is valid.**
File 3 BearScrub.sol • Withdraw Lockup Epochs: 6 epochs • Reward Lockup Epochs: 3 epochs	**YES, This is valid.**
File 4 BearTreasury.sol • Maximum Supply Expansion: 4% • Bond supply for depletion floor: 100% • At least 35% of expansion is reserved for scrub. • Maximum Supply for contraction: 35% • Premium Threshold: 110 • Premium Percentage: 70% • Period: 8 hours	**YES, This is valid.** **Owner authorized wallet can set some percentage value and we suggest handling the private key of that wallet securely.**
File 5 LionOracle.sol • LionOracle has functions like: update, consult, twap.	**YES, This is valid.**

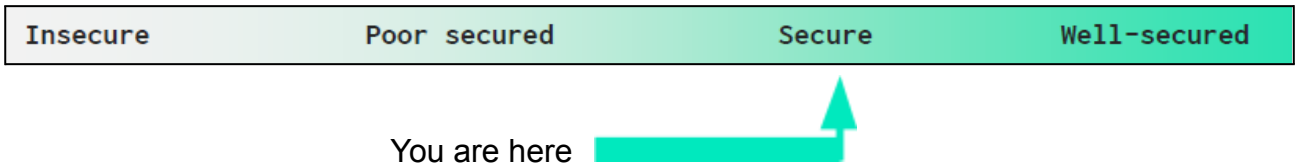
This is a private and confidential document. No part of this document should be disclosed to third party without prior written permission of EtherAuthority.
Email: audit@EtherAuthority.io

File 6 RewardManager.sol • RewardManager has functions like: earned, claimRewards, stake, exit.	**YES, This is valid.**
File 7 Tiger.sol • Name: Tiger • Symbol: TIGER • Decimals: 18 • Initial distribution for the MMF launchpad: 0.4 Million TIGER • Get Tax Tiers Rates Count: 14 • Get Tax Tiers Twaps Count: 14 • Burn Threshold: 1,100 • Total Supply: 400,001 TIGER	**YES, This is valid.**
File 8 UserVault.sol • UserVault has functions like: withdrawAll, stake, withdraw, claimReward, exit.	**YES, This is valid.**
File 9 SaleBatch.sol • SaleBatch has functions like: togglePaused, configureVotingToken, setRaisingAmount, etc.	**YES, This is valid.**
File 10 Zap.sol • Zap has functions like: zapInToken, estimateZapInToken, zapIn, estimateZapIn,etc.	**YES, This is valid.**

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

Audit Summary

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



We used various tools like Slither, Solhint and Remix IDE. At the same time this finding is based on critical analysis of the manual audit.
All issues found during automated analysis were manually reviewed and applicable vulnerabilities are presented in the Audit overview section. General overview is presented in AS-IS section and all identified issues can be found in the Audit overview section.

We found 0 critical, 0 high, 0 medium and 4 low and some very low level issues.

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

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

Technical Quick Stats

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

Overall Audit Result: **PASSED**

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

Code Quality

This audit scope has 10 smart contract files. Smart contracts contain Libraries, Smart contracts, inherits and Interfaces. This is a compact and well written smart contract.

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

The Scrub-Finance 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 **not** well commented on smart contracts.

Documentation

We were given a Scrub-Finance Protocol smart contract code in the form of a file. The hash of that code is mentioned above in the table.

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

Use of Dependencies

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

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

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

AS-IS overview

BBond.sol

Functions

Sl.	Functions	Type	Observation	Conclusion
1	constructor	write	Passed	No Issue
2	mint	write	Unlimited mint	Refer Audit Findings
3	burn	write	Passed	No Issue
4	burnFrom	write	access only Operator	No Issue
5	burn	write	Passed	No Issue
6	burnFrom	write	Passed	No Issue
7	operator	read	Passed	No Issue
8	onlyOperator	modifier	Passed	No Issue
9	isOperator	read	Passed	No Issue
10	transferOperator	write	access only Owner	No Issue
11	_transferOperator	internal	Passed	No Issue

Bear.sol

Functions

Sl.	Functions	Type	Observation	Conclusion
1	constructor	write	Passed	No Issue
2	burn	write	Passed	No Issue
3	burnFrom	write	Passed	No Issue
4	operator	read	Passed	No Issue
5	onlyOperator	modifier	Passed	No Issue
6	isOperator	read	Passed	No Issue
7	transferOperator	write	access only Owner	No Issue
8	_transferOperator	write	Passed	No Issue
9	onlyTaxOffice	modifier	Passed	No Issue
10	onlyOperatorOrTaxOffice	modifier	Passed	No Issue
11	getTaxTiersTwapsCount	read	Passed	No Issue
12	getTaxTiersRatesCount	read	Passed	No Issue
13	isAddressExcluded	read	Passed	No Issue
14	setTaxTiersTwap	write	access only Tax Office	No Issue
15	setTaxTiersRate	write	access only Tax Office	No Issue
16	setBurnThreshold	write	access only Tax Office	No Issue
17	_getLionPrice	internal	Passed	No Issue
18	_updateTaxRate	internal	Passed	No Issue
19	enableAutoCalculateTax	write	Passed	No Issue

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

20	disableAutoCalculateTax	write	access only Tax Office	No Issue
21	setOracle	write	access only Operator Or Tax Office	No Issue
22	setTaxOffice	write	access only Operator	No Issue
23	setTaxCollectorAddress	write	access only Tax Office	No Issue
24	setTaxRate	write	Critical operation lacks event log	Refer Audit Findings
25	setBurnTax	write	access only Tax Office	No Issue
26	excludeAddress	write	access only Operator Or Tax Office	No Issue
27	includeAddress	write	access only Operator Or Tax Office	No Issue
28	OpenTrade	external	Critical operation lacks event log	Refer Audit Findings
29	includeToWhitelist	write	access only Operator Or Tax Office	No Issue
30	excludeFromWhitlist	write	access only Operator Or Tax Office	No Issue
31	mint	write	Unlimited mint	Refer Audit Findings
32	burn	write	Passed	No Issue
33	burnFrom	write	access only Operator	No Issue
34	transferFrom	write	Passed	No Issue
35	_transferWithTax	internal	Passed	No Issue
36	_transfer	internal	Passed	No Issue
37	governanceRecoverUnsupported	external	Function input parameters lack of check	Refer Audit Findings

BearScrub.sol

Functions

Sl.	Functions	Type	Observation	Conclusion
1	constructor	write	Passed	No Issue
2	totalSupply	read	Passed	No Issue
3	balanceOf	read	Passed	No Issue
4	stake	write	Passed	No Issue
5	withdraw	write	Passed	No Issue
6	checkSameOriginReentranted	internal	Passed	No Issue
7	checkSameSenderReentranted	internal	Passed	No Issue
8	onlyOneBlock	modifier	Passed	No Issue
9	onlyOperator	modifier	Passed	No Issue
10	onlyRewardManager	modifier	Passed	No Issue

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

10	masonExists	modifier	Passed	No Issue
11	updateReward	modifier	Passed	No Issue
12	notInitialized	modifier	Passed	No Issue
13	initialize	write	Passed	No Issue
14	setRewardManager	write	access only Operator	No Issue
15	setOperator	external	Function input parameters lack of check	Refer Audit Findings
16	setLockUp	external	access only Operator	No Issue
17	latestSnapshotIndex	read	Passed	No Issue
18	getLatestSnapshot	internal	Passed	No Issue
19	getLastSnapshotIndexOf	read	Passed	No Issue
20	getLastSnapshotOf	internal	Passed	No Issue
21	canWithdraw	external	Passed	No Issue
22	canClaimReward	external	Passed	No Issue
23	epoch	external	Passed	No Issue
24	nextEpochPoint	external	Passed	No Issue
25	getBearPrice	external	Passed	No Issue
26	rewardPerShare	read	Passed	No Issue
27	earned	read	Passed	No Issue
28	stake	write	access only One Block	No Issue
29	withdraw	write	access only One Block	No Issue
30	exit	write	access only Reward Manager	No Issue
31	claimReward	write	access only Reward Manager	No Issue
32	allocateSeigniorage	external	access only Operator	No Issue
33	governanceRecoverUnsupported	external	Function input parameters lack of check	Refer Audit Findings

BearTreasury.sol

Functions

Sl.	Functions	Type	Observation	Conclusion
1	constructor	write	Passed	No Issue
2	checkSameOriginReentranted	internal	Passed	No Issue
3	checkSameSenderReentranted	internal	Passed	No Issue
4	onlyOneBlock	modifier	Passed	No Issue
5	onlyOperator	modifier	Passed	No Issue
6	checkCondition	modifier	Passed	No Issue
7	checkEpoch	modifier	Passed	No Issue
8	checkOperator	modifier	Passed	No Issue

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

9	notInitialized	modifier	Passed	No Issue
10	isInitialized	read	Passed	No Issue
11	nextEpochPoint	read	Passed	No Issue
12	getBearPrice	read	Passed	No Issue
13	getBearUpdatedPrice	read	Passed	No Issue
14	getReserve	read	Passed	No Issue
15	getBurnableBearLeft	read	Passed	No Issue
16	getRedeemableBonds	read	Passed	No Issue
17	getBondDiscountRate	read	Passed	No Issue
18	getBondPremiumRate	read	Passed	No Issue
19	initialize	write	Passed	No Issue
20	setOperator	external	access only Operator	No Issue
21	setScrub	external	access only Operator	No Issue
22	setBearOracle	external	access only Operator	No Issue
23	setBearPriceCeiling	external	access only Operator	No Issue
24	setMaxSupplyExpansion Percents	external	access only Operator	No Issue
25	setSupplyTiersEntry	external	access only Operator	No Issue
26	setMaxExpansionTiersEn try	external	access only Operator	No Issue
27	setBondDepletionFloorPe rcent	external	access only Operator	No Issue
28	setMaxSupplyContraction Percent	external	access only Operator	No Issue
29	setMaxDebtRatioPercent	external	access only Operator	No Issue
30	setBootstrap	external	access only Operator	No Issue
31	setExtraFunds	external	access only Operator	No Issue
32	setMaxDiscountRate	external	access only Operator	No Issue
33	setMaxPremiumRate	external	access only Operator	No Issue
34	setDiscountPercent	external	access only Operator	No Issue
35	setPremiumThreshold	external	access only Operator	No Issue
36	setPremiumPercent	external	access only Operator	No Issue
37	setMintingFactorForPayin gDebt	external	access only Operator	No Issue
38	_updateBearPrice	internal	Passed	No Issue
39	getBearCirculatingSupply	read	Infinite loop	Refer Audit Findings
40	buyBonds	external	access only One Block	No Issue
41	redeemBonds	external	access only One Block	No Issue
42	_sendToScrub	internal	Passed	No Issue
43	_calculateMaxSupplyExp ansionPercent	internal	Passed	No Issue
44	allocateSeigniorage	external	access only One Block	No Issue
45	excludeFromTotalSupply	external	Infinite loop	Refer Audit Findings

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

46	includeToTotalSupply	external	access only Operator	No Issue
47	governanceRecoverUnsupported	external	access only Operator	No Issue
48	scrubSetOperator	external	access only Operator	No Issue
49	scrubSetLockUp	external	access only Operator	No Issue
50	scrubAllocateSeigniorage	external	access only Operator	No Issue
51	scrubGovernanceRecoverUnsupported	external	access only Operator	No Issue

LionOracle.sol

Functions

Sl.	Functions	Type	Observation	Conclusion
1	constructor	write	Passed	No Issue
2	checkStartTime	modifier	Passed	No Issue
3	checkEpoch	modifier	Passed	No Issue
4	getCurrentEpoch	read	Passed	No Issue
5	getPeriod	read	Passed	No Issue
6	getStartTime	read	Passed	No Issue
7	getLastEpochTime	read	Passed	No Issue
8	nextEpochPoint	read	Passed	No Issue
9	setPeriod	external	access only Operator	No Issue
10	setEpoch	external	access only Operator	No Issue
11	update	external	checkEpoch	No Issue
12	consult	external	Passed	No Issue
13	twap	external	Passed	No Issue

RewardManager.sol

Functions

Sl.	Functions	Type	Observation	Conclusion
1	constructor	write	Passed	No Issue
2	checkSameOriginReentranted	internal	Passed	No Issue
3	checkSameSenderReentranted	internal	Passed	No Issue
4	onlyOneBlock	modifier	Passed	No Issue
5	earned	read	Passed	No Issue
6	claimRewards	write	Passed	No Issue
7	stake	write	access only One Block	No Issue
8	exit	external	Passed	No Issue

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

Tiger.sol

Functions

Sl.	Functions	Type	Observation	Conclusion
1	constructor	write	Passed	No Issue
2	operator	read	Passed	No Issue
3	onlyOperator	modifier	Passed	No Issue
4	isOperator	read	Passed	No Issue
5	transferOperator	write	access only Owner	No Issue
6	_transferOperator	internal	Passed	No Issue
7	burn	write	Passed	No Issue
8	burnFrom	write	Passed	No Issue
9	onlyTaxOffice	modifier	Passed	No Issue
10	onlyOperatorOrTaxOffice	modifier	Passed	No Issue
11	getTaxTiersTwapsCount	read	Passed	No Issue
12	getTaxTiersRatesCount	read	Passed	No Issue
13	isAddressExcluded	read	Passed	No Issue
14	setTaxTiersTwap	write	access only Tax Office	No Issue
15	setTaxTiersRate	write	access only Tax Office	No Issue
16	setBurnThreshold	write	access only Tax Office	No Issue
17	_getLionPrice	internal	Passed	No Issue
18	_updateTaxRate	internal	Passed	No Issue
19	enableAutoCalculateTax	write	access only Tax Office	No Issue
20	disableAutoCalculateTax	write	access only Tax Office	No Issue
21	setOracle	write	access only Operator Or Tax Office	No Issue
22	setTaxOffice	write	access only Operator Or Tax Office	No Issue
23	setTaxCollectorAddress	write	access only Tax Office	No Issue
24	setTaxRate	write	access only Tax Office	No Issue
25	setBurnTax	write	access only Tax Office	No Issue
26	excludeAddress	write	access only Operator Or Tax Office	No Issue
27	includeAddress	write	access only Operator Or Tax Office	No Issue
28	OpenTrade	external	access only Operator Or Tax Office	No Issue
29	includeToWhitelist	write	access only Operator Or Tax Office	No Issue

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

30	excludeFromWhitlist	write	access only Operator Or Tax Office	No Issue
31	mint	write	access only Operator	No Issue
32	burn	write	Passed	No Issue
33	burnFrom	write	access only Operator	No Issue
34	transferFrom	write	Passed	No Issue
35	_transferWithTax	internal	Passed	No Issue
36	_transfer	internal	Passed	No Issue
37	governanceRecoverUnsupported	external	Function input parameters lack of check	Refer Audit Findings

UserVault.sol

Functions

Sl.	Functions	Type	Observation	Conclusion
1	constructor	write	Passed	No Issue
2	checkSameOriginReentranted	internal	Passed	No Issue
3	checkSameSenderReentranted	internal	Passed	No Issue
4	onlyOneBlock	modifier	Passed	No Issue
5	onlyManager	modifier	Passed	No Issue
6	withdrawAll	write	Passed	No Issue
7	stake	external	access only Manager	No Issue
8	withdraw	external	access only Manager	No Issue
9	claimReward	external	access only Manager	No Issue
10	exit	external	access only Manager	No Issue

SaleBatch.sol

Functions

Sl.	Functions	Type	Observation	Conclusion
1	constructor	write	Passed	No Issue
2	owner	read	Passed	No Issue
3	onlyOwner	modifier	Passed	No Issue
4	renounceOwnership	write	access only Owner	No Issue
5	transferOwnership	write	access only Owner	No Issue
6	_transferOwnership	internal	Passed	No Issue
7	nonReentrant	modifier	Passed	No Issue
8	configureVotingToken	write	access only Owner	No Issue
9	setRaisingAmount	write	access only Owner	No Issue
10	togglePaused	write	access only Owner	No Issue
11	finalize	write	Critical operation lacks event log	Refer Audit Findings
12	getAddressListLength	external	Passed	No Issue

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

13	getParams	external	Passed	No Issue
14	getVotingParams	external	Passed	No Issue
15	_deposit	write	Passed	No Issue
16	deposit	write	Passed	No Issue
17	onTokenTransfer	write	Passed	No Issue
18	harvestRefund	write	Passed	No Issue
19	harvestTokens	write	Passed	No Issue
20	harvestAll	write	Passed	No Issue
21	getUserAllocation	read	Passed	No Issue
22	getOfferingAmount	read	Passed	No Issue
23	getRefundingAmount	read	Passed	No Issue
24	withdrawToken	write	Owner can drain all tokens	Refer Audit Findings
25	_transferFrom	write	Passed	No Issue

Zap.sol

Functions

Sl.	Functions	Type	Observation	Conclusion
1	constructor	write	Passed	No Issue
2	owner	read	Passed	No Issue
3	onlyOwner	modifier	Passed	No Issue
4	renounceOwnership	write	access only Owner	No Issue
5	transferOwnership	write	access only Owner	No Issue
6	zapInToken	external	Passed	No Issue
7	estimateZapInToken	read	Passed	No Issue
8	receive	external	Passed	No Issue
9	zapIn	external	Function input parameters lack of check	Refer Audit Findings
10	estimateZapIn	read	Passed	No Issue
11	zapAcross	external	Function input parameters lack of check	Refer Audit Findings
12	zapOut	external	Function input parameters lack of check	Refer Audit Findings
13	zapOutToken	external	Function input parameters lack of check	Refer Audit Findings
14	swapToken	external	Function input parameters lack of check	Refer Audit Findings
15	swapToNative	external	Function input parameters lack of check	Refer Audit Findings
16	_approveTokenIfNeeded	write	Passed	No Issue

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

17	_swapTokenToLP	write	Passed	No Issue
18	_swapNativeToLP	write	Passed	No Issue
19	_swapHalfNativeAndProvide	write	Passed	No Issue
20	_swapNativeToEqualTokensAndProvide	write	Passed	No Issue
21	_swapNativeForToken	write	Passed	No Issue
22	_swapTokenForNative	write	Passed	No Issue
23	_swap	write	Passed	No Issue
24	_estimateSwap	read	Passed	No Issue
25	setTokenBridgeForRouter	external	access only Owner	No Issue
26	withdraw	external	Owner can drain all tokens	Refer Audit Findings
27	setUseNativeRouter	external	Can not update router	Refer Audit Findings
28	setFee	external	Function input parameters lack of check	Refer Audit Findings

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

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.

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

Audit Findings

Critical Severity

No Critical severity vulnerabilities were found.

High Severity

No High severity vulnerabilities were found.

Medium

No Medium severity vulnerabilities were found.

Low

(1) Function input parameters lack of check:

Variable validation is not performed in below functions:

Bear.sol

- governanceRecoverUnsupported = to

BearScrub.sol

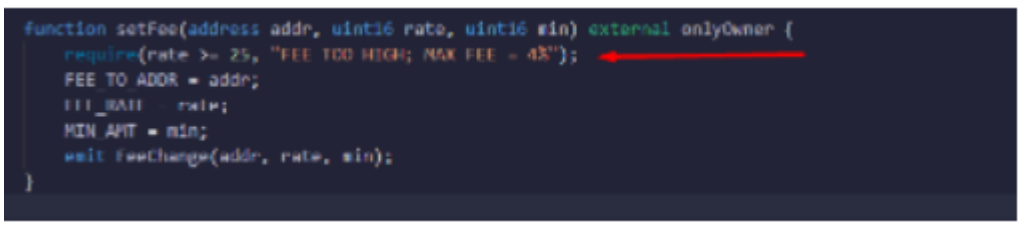
- setOperator = _operator
- governanceRecoverUnsupported = to

Tiger.sol

- governanceRecoverUnsupported = to

Zap.sol

- zapIn = _to , routerAddr , _recipient
- zapAcross = _from , _toRouter , _recipient
- zapOut = _from , routerAddr , _recipient
- zapOutToken = _from , _to, routerAddr ,_recipient
- swapToken = _from , _to , routerAddr, _recipient
- swapToNative = _from , routerAddr , _recipient
- settee



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

- In the setFee function, the required condition and error message are conflicting. Error message says "FEE TOO HIGH; MAX FEE = 4%" and condition rate must be >=25.

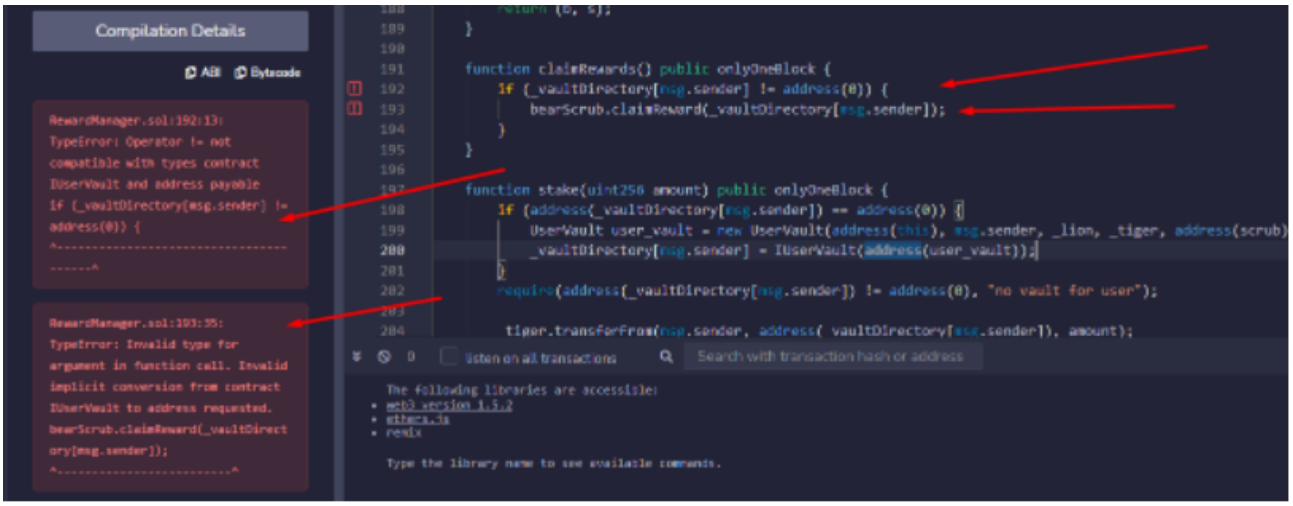
Resolution: We advise using validation like integer type variables should be > 0 and address type variables should not be address(0). Fees must have a maximum limit.

(2) Infinite loop possibility: **BearTreasury.sol**
In below functions ,for loops do not have excludedFromTotalSupply length limit , which costs more gas:

- getBearCirculatingSupply
- excludeFromTotalSupply

Resolution: Upper limit should have a certain limit in for loops.

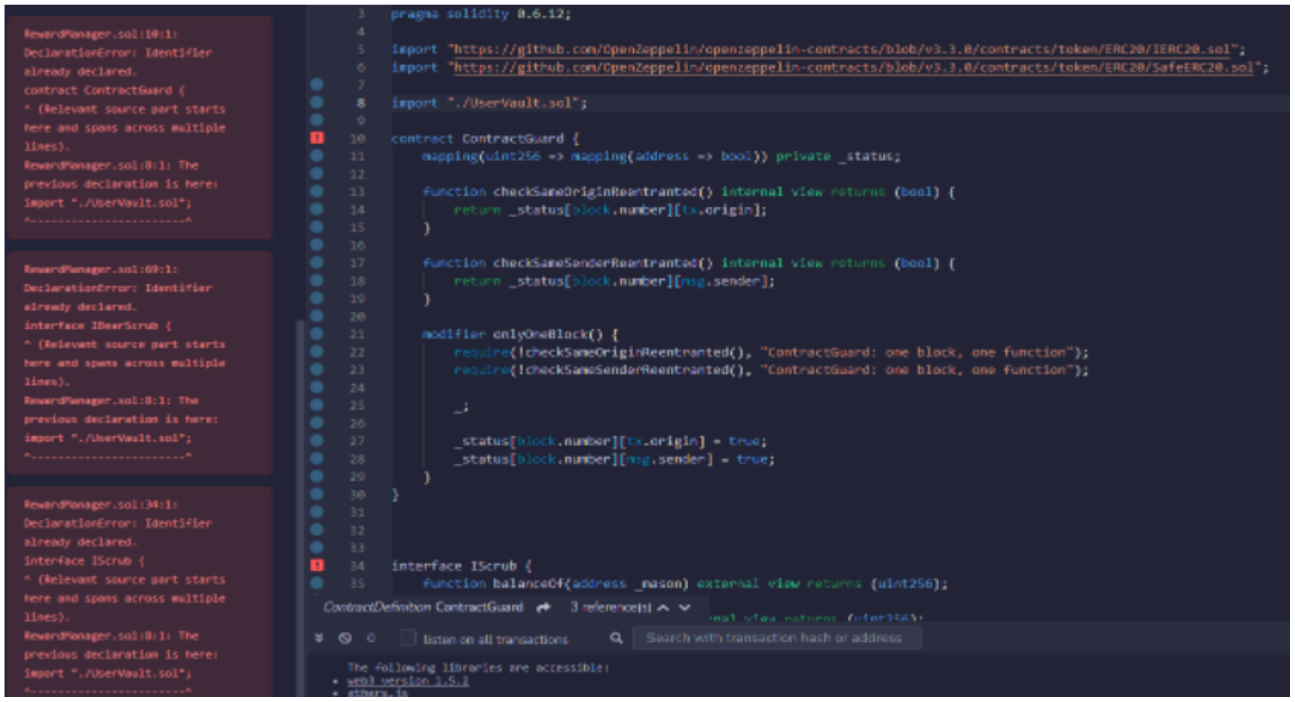
(3) Compile error: **RewardManager.sol**



Operator != not compatible with types contract IUserVault and address payable.

Resolution: We suggest replacing _vaultDirectory[msg.sender] with address(_vaultDirectory[msg.sender]).
Status: Fixed

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



DeclarationError: Identifier already declared. UserVault also has the same Files imported as RewardManager.

Resolution: We suggest not to import the UserVault.sol file. Instead add UserVault contract code to RewardManager.sol.

(4) Critical operation lacks event log: **SaleBatch.sol**
Missing event log for : finalize.

Resolution: Write an event log for listed events.

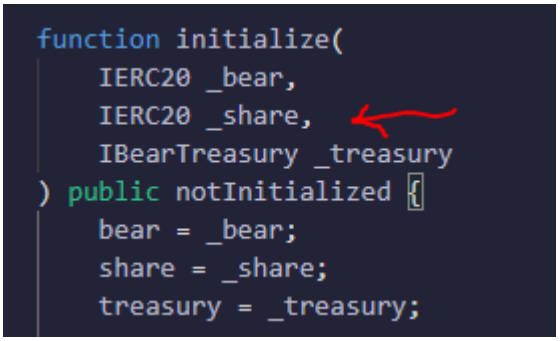
Very Low / Informational / Best practices:

(1) Unlimited mint: **Bbond.sol, Bear.sol, Tiger.sol**
Operators can mint unlimited tokens.

Resolution: We suggest putting a mint limit.

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

(2) Unused variable: **BearScrub.sol**



```
function initialize(
    IERC20 _bear,
    IERC20 _share,
    IBearTreasury _treasury
) public notInitialized {
    bear = _bear;
    share = _share;
    treasury = _treasury;
```

The share variable is not declared and used anywhere in the contract.

Resolution: We suggest removing unused variables.

(3) Immutable variables:

In below variables values are set in initialize() function and will be unchanged.

BearTreasury.sol

- bearPriceOne , startTime, bear , bbond

BearScrub.sol

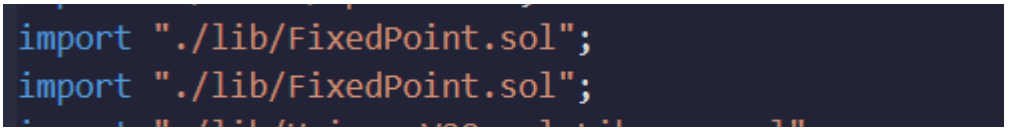
- bear, share, treasury

SaleBatch.sol

- startTime , endTime , offeringAmount ,offeringToken ,paymentToken ,perUserCap

Resolution: We suggest setting all these variables as immutable.

(4) Same file imported twice: **LionOracle.sol**



```
import "./lib/FixedPoint.sol";
import "./lib/FixedPoint.sol";
```

Multiple imports for the same file "fixedPoint.sol".

Resolution: We suggest making a single import for "FixedPoint.sol".

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

(5) Owner can drain all tokens:

Zap.sol

```
function withdraw(address token) external onlyOwner {
    if (token == address(0)) {
        payable(owner()).transfer(address(this).balance);
        return;
    }

    IERC20(token).transfer(owner(), IERC20(token).balanceOf(address(this)));
}
```

SaleBatch.sol

```
function withdrawToken(address token, uint amount) public onlyOwner {
    IERC20(token).safeTransfer(msg.sender, amount);
}
```

The function withdraw() will allow the owner to withdraw all the ERC20 tokens. This would create trust issues in the users.

Resolution: If these are desired features, then please ignore this point.

(6) Can not update router: **Zap.sol**

```
function setUseNativeRouter(address router) external onlyOwner {
    useNativeRouter[router] = true;
}
```

The owner can update the router that generates liquidity to an address or contract of choice (including the zero address). This contract could be a malicious contract that simply keeps the tokens sent to it and thus drains all the fees. Additionally, this contract could be used to revert sell transactions turning the token into a honeypot.

Resolution: Consider removing this function. If this is not possible, consider using an Owner account that is behind a significantly long time lock so investors can reasonably see this change coming and inspect the new router. Also consider requiring the router address to be non-zero.

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

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:

- mint: The BBond Operator can mint an amount from the recipient address.
- burnFrom: The BBond Operator can burn an amount from an account.
- setOracle: The Bear Operator can update oracle addresses.
- setTaxOffice: The Bear Operator can update the tax office address.
- excludeAddress: The Bear Operator can exclude address.
- includeAddress: The Bear Operator can include address.
- OpenTrade: The Bear Operator Or TaxOffice can open trade status true.
- includeToWhitelist: The Bear Operator Or TaxOffice can include address in whitelist.
- excludeFromWhitlist: The Bear Operator Or TaxOffice can exclude address in whitelist.
- mint: The Bear Operator can mint LION to a recipient address.
- burnFrom: The Bear Operator can burn a LION amount from the address.
- governanceRecoverUnsupported: The Bear Operator can governance recover unsupported amounts.
- setRewardManager: BearScrub Operator can update reward manager address.
- setOperator: BearScrub Operator can update Operator address.
- setLockUp: BearScrub Operator can withdraw lockup Epochs and reward lockup Epochs value.
- stake: BearScrub Reward Manager can create a new stake.
- withdraw: BearScrub Reward Manager can withdraw the amount.
- exit: BearScrub Reward Manager can exit from address.
- claimReward: BearScrub Reward Manager can claim reward address.
- allocateSeigniorage: BearScrub Reward Manager can allocate seigniorage amount.
- governanceRecoverUnsupported: BearScrub Reward Manager can governance recover unsupported amounts.
- setOperator: BearTreasury Operator can update operator address.
- setScrub: BearTreasury Operator can update scrub address.

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

- setBearOracle: BearTreasury Operator can update bear oracle address.
- setBearPriceCeiling: BearTreasury Operator can update bear price ceiling value.
- setMaxSupplyExpansionPercents: BearTreasury Operator can update maximum supply expansion percentage.
- setSupplyTiersEntry: BearTreasury Operator can update supply tiers entry.
- setMaxExpansionTiersEntry: BearTreasury Operator can update maximum expansion tiers entry.
- setBondDepletionFloorPercent: BearTreasury Operator can update bond depletion floor percentage.
- setMaxSupplyContractionPercent: BearTreasury Operator can update maximum supply contraction percentage.
- setMaxDebtRatioPercent: BearTreasury Operator can update maximum debt ratio percentage.
- setBootstrap: BearTreasury Operator can update Bootstrap value.
- setExtraFunds: BearTreasury Operator can update extra funds value.
- setMaxDiscountRate: BearTreasury Operator can set maximum discount rate.
- setMaxPremiumRate: BearTreasury Operator can set maximum premium rate.
- setDiscountPercent: BearTreasury Operator can set discount percentage.
- setPremiumThreshold: BearTreasury Operator can set premium threshold value.
- setPremiumPercent: BearTreasury Operator can set premium percentage.
- setMintingFactorForPayingDebt: BearTreasury Operator can set minting factor for paying debt value.
- excludeFromTotalSupply: BearTreasury Operator can exclude account from total supply.
- includeToTotalSupply: BearTreasury Operator can include index from total supply.
- governanceRecoverUnsupported: BearTreasury Operator can governance recover unsupported.
- mint: Tiger Operator mints LION to a recipient.
- burnFrom: Tiger Operator burn amount from account.
- governanceRecoverUnsupported: Tiger Operator can governance recover unsupported.
- exit: UserVault manager can exit.
- claimReward: UserVault manager can claim reward.
- withdraw: UserVault manager can withdraw all amounts.

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

- stake: UserVault manager can stake amounts.
- setFee: Zap owner can set fees.
- setUseNativeRouter: Zap owner can set user native router address.
- withdraw: Zap owner can withdraw amount.
- setTokenBridgeForRouter: Zap owner can set token bridge router address.
- withdrawToken: SaleBatch owner can withdraw all tokens.
- setRaisingAmount: SaleBatch owner can set raising amount.
- configureVotingToken: SaleBatch owner can configure voting token.

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

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

Conclusion

We were given a contract code in the form of files. And we have used all possible tests based on given objects as files. We had observed some issues in the smart contracts, but they were resolved in the revised smart contract code. **So, the smart contracts are ready for the mainnet deployment**.

Since possible test cases can be unlimited for such smart contracts protocol, we provide no such guarantee of future outcomes. We have used all the latest static tools and manual observations to cover maximum possible test cases to scan everything.

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

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

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

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

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.

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

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.

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

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

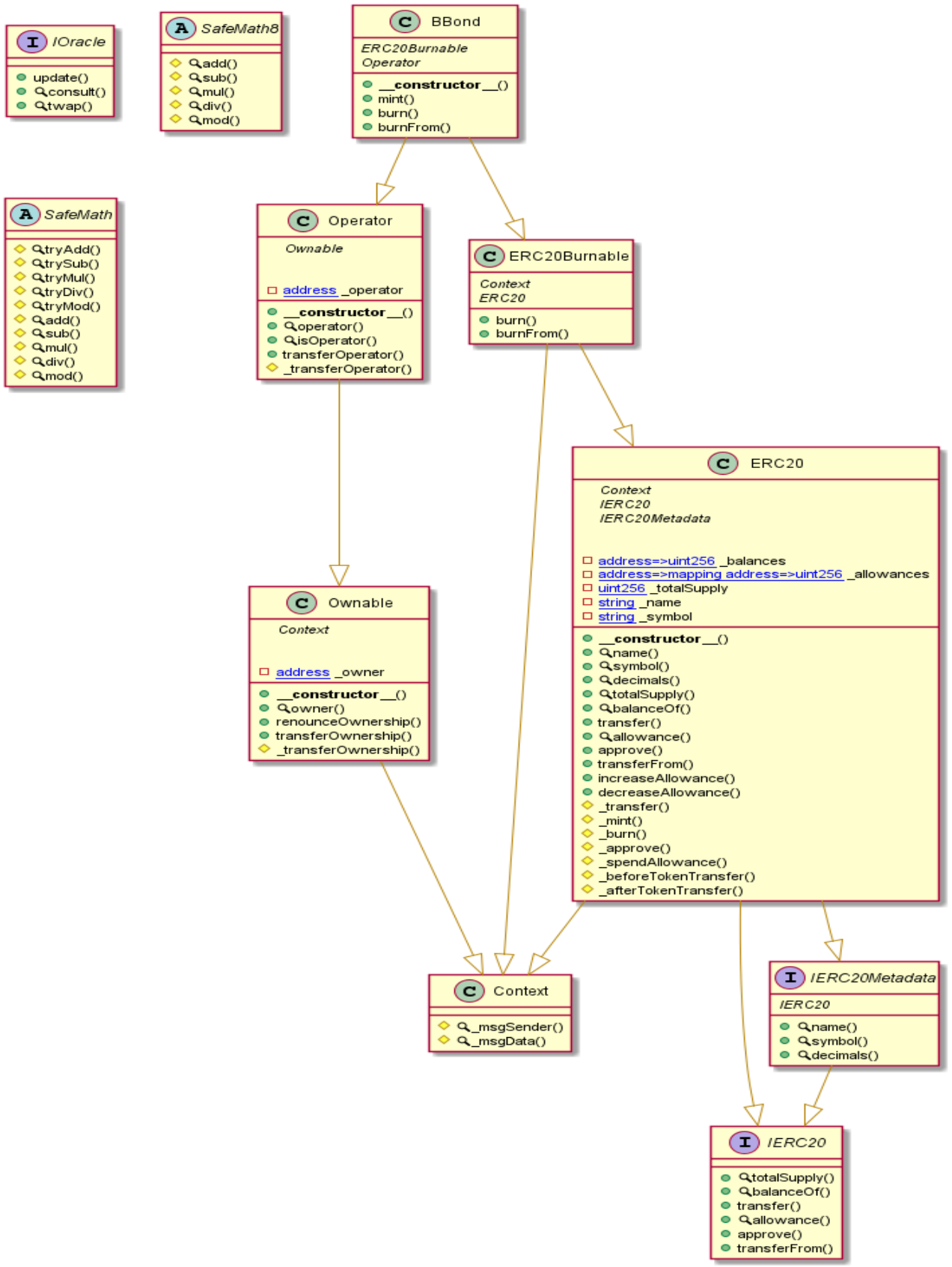
Smart contracts are deployed and executed on the blockchain platform. The platform, its programming language, and other software related to the smart contract can have their own vulnerabilities that can lead to hacks. Thus, the audit can't guarantee explicit security of the audited smart contracts.

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

Appendix

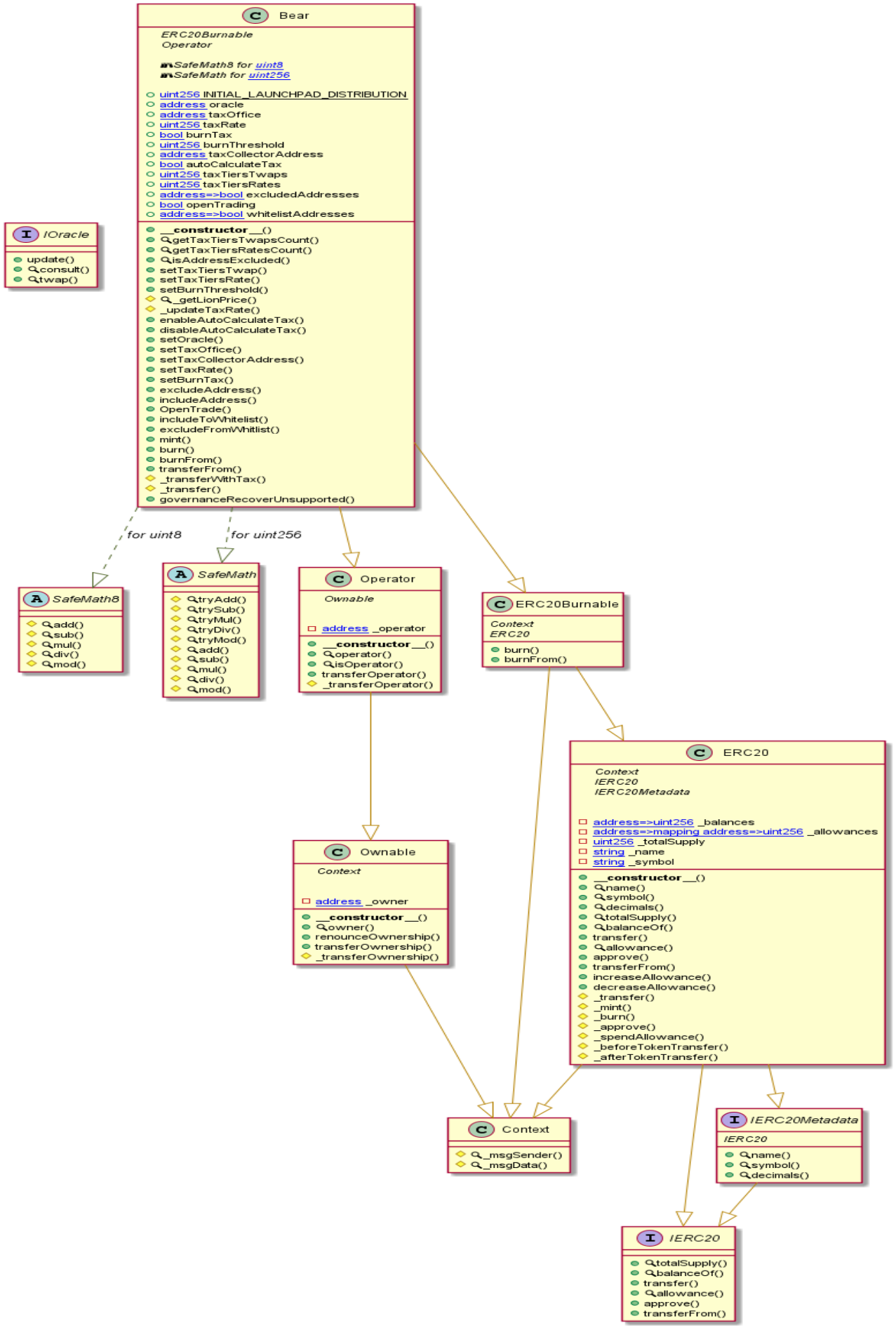
Code Flow Diagram - Scrub-Finance Protocol

BBond Diagram



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

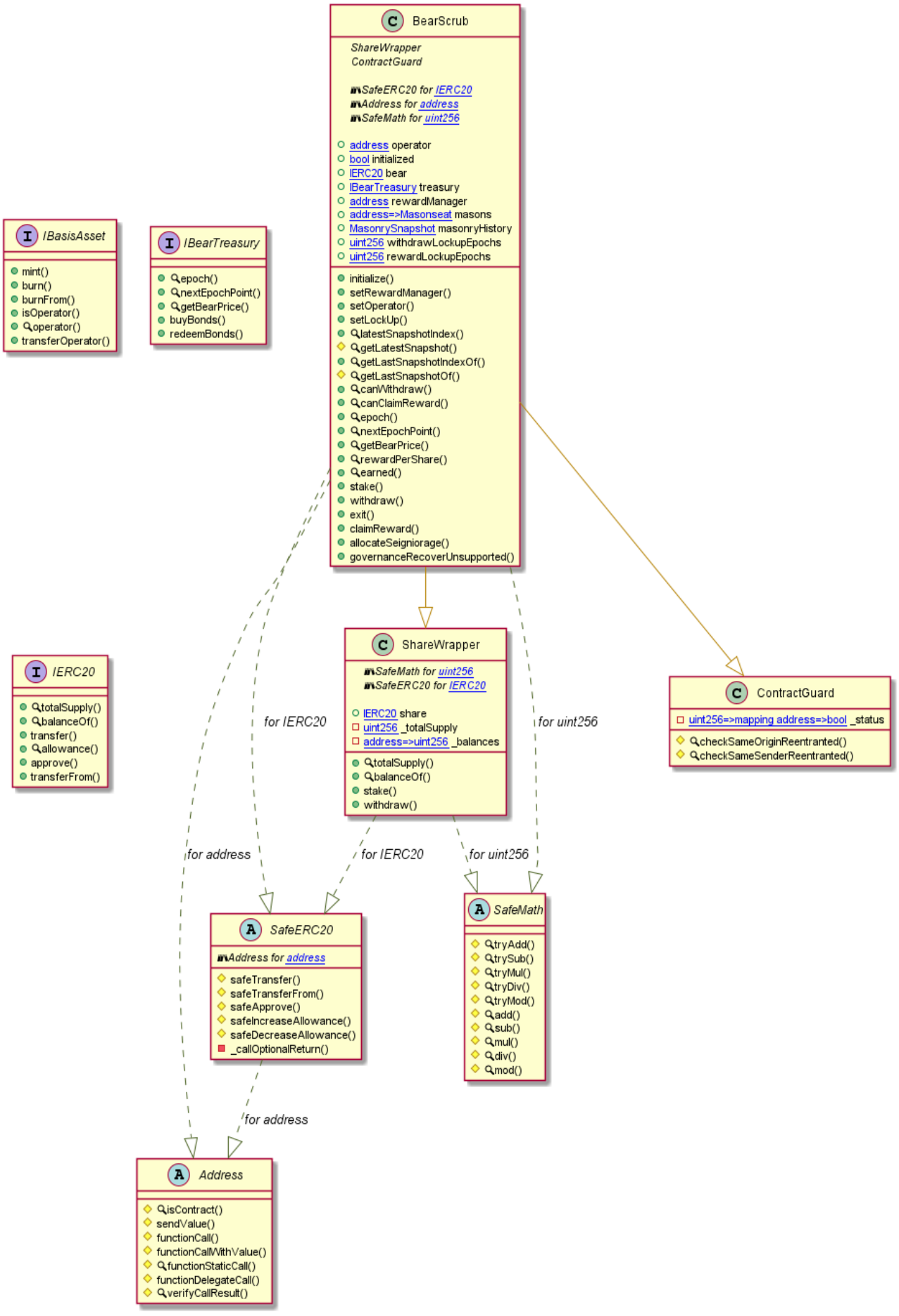
Bear Diagram



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

Email: audit@EtherAuthority.io

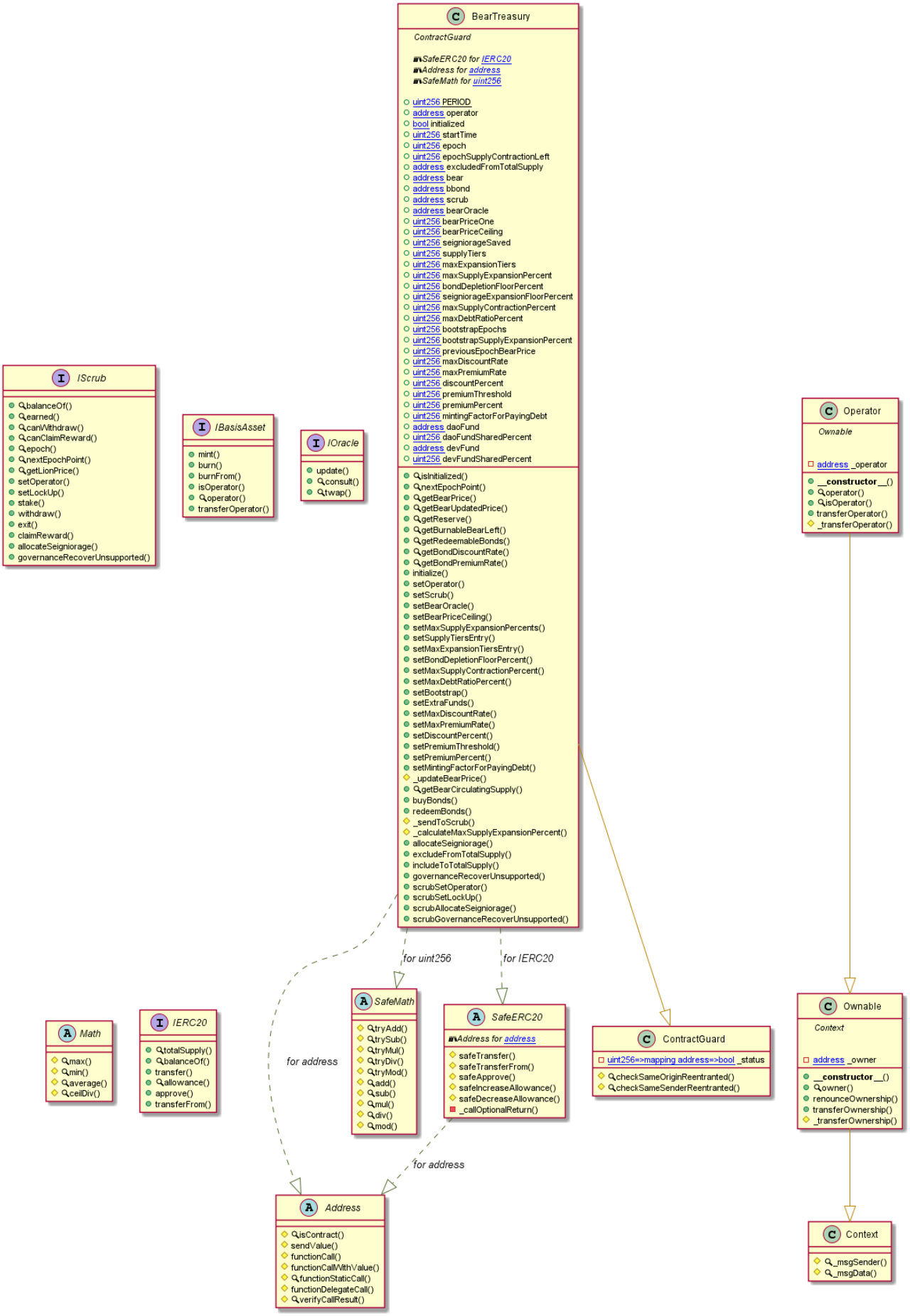
BearScrub Diagram



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

Email: audit@EtherAuthority.io

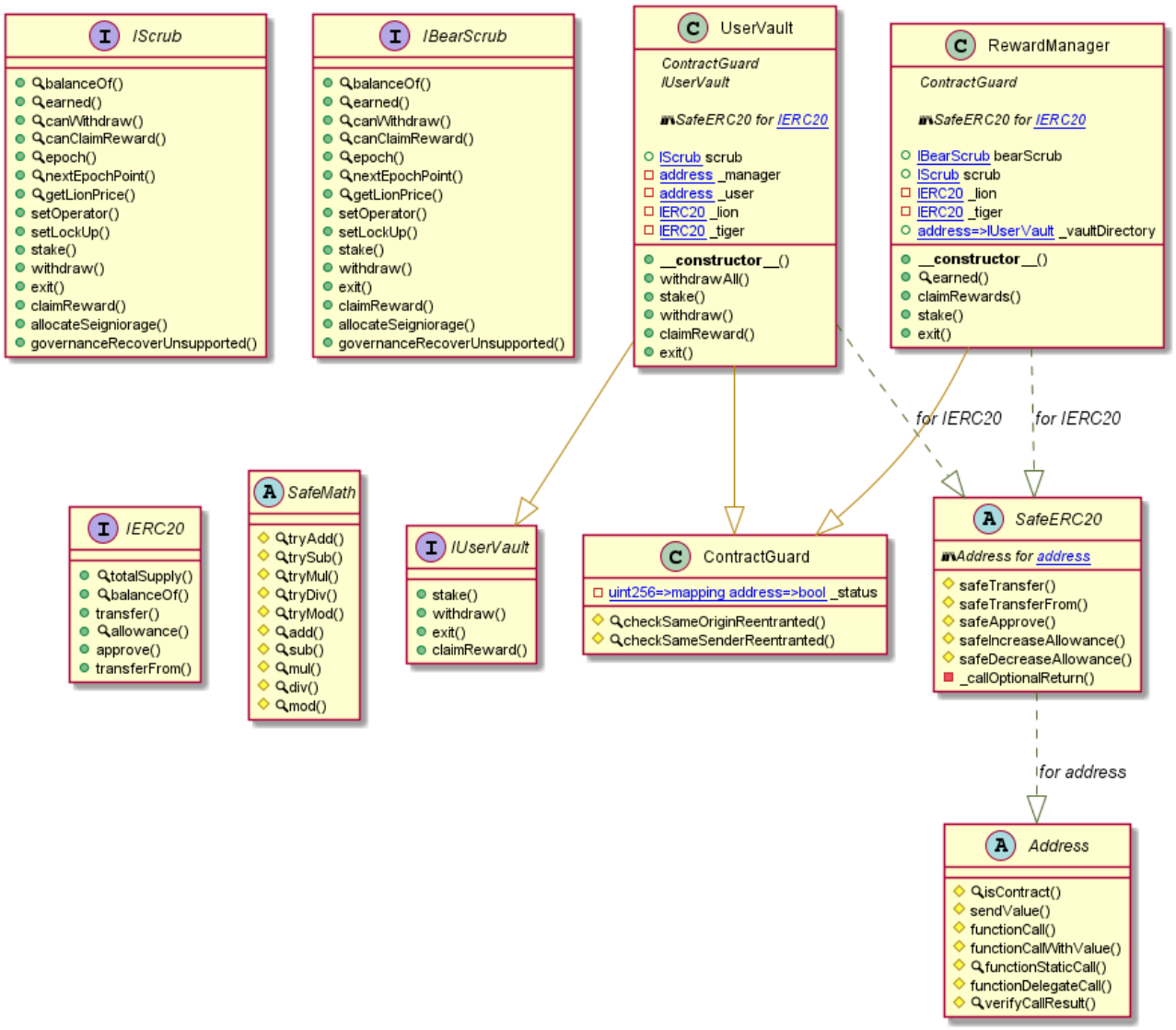
BearTreasury Diagram



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

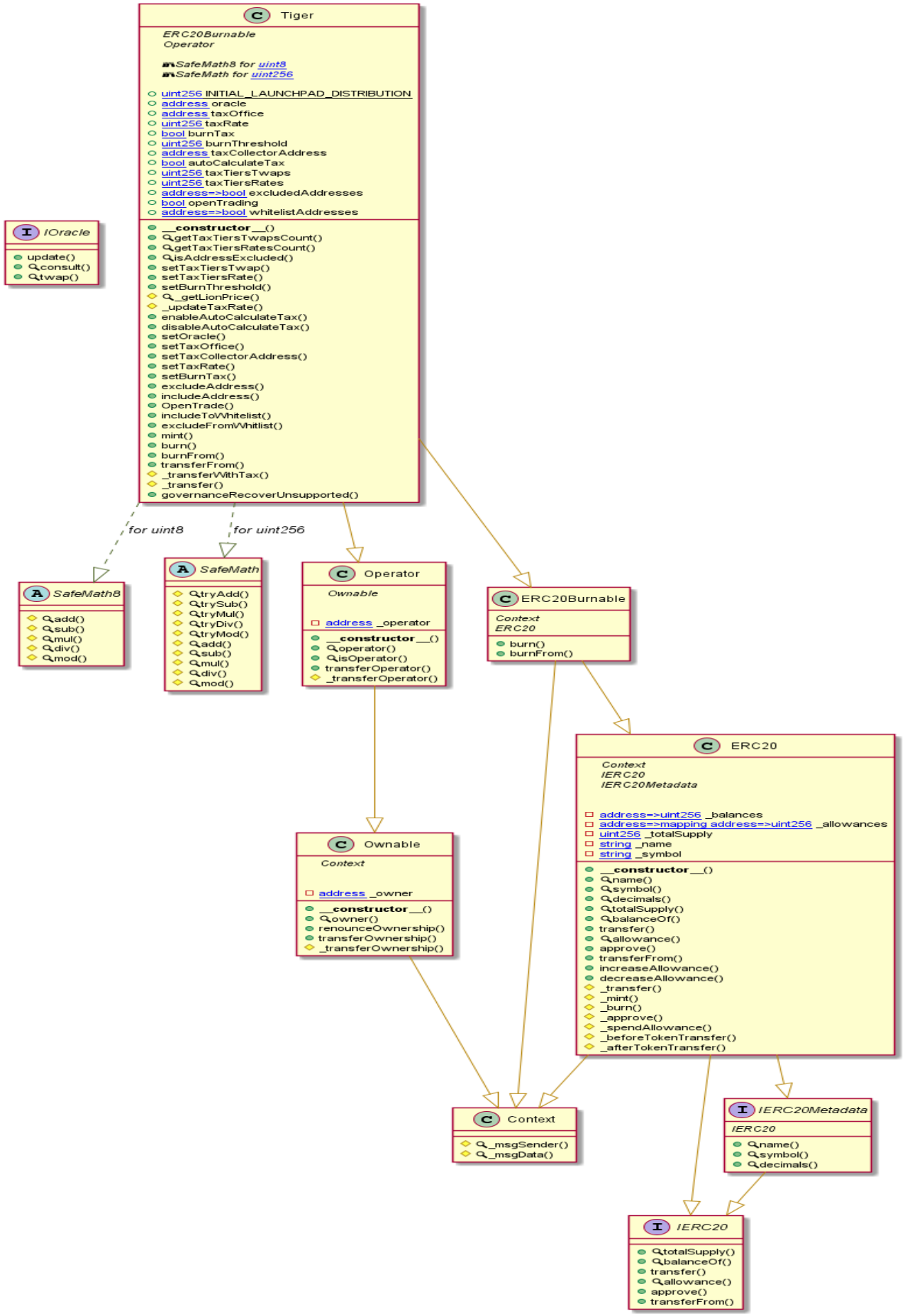
Email: audit@EtherAuthority.io

RewardManager Diagram



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

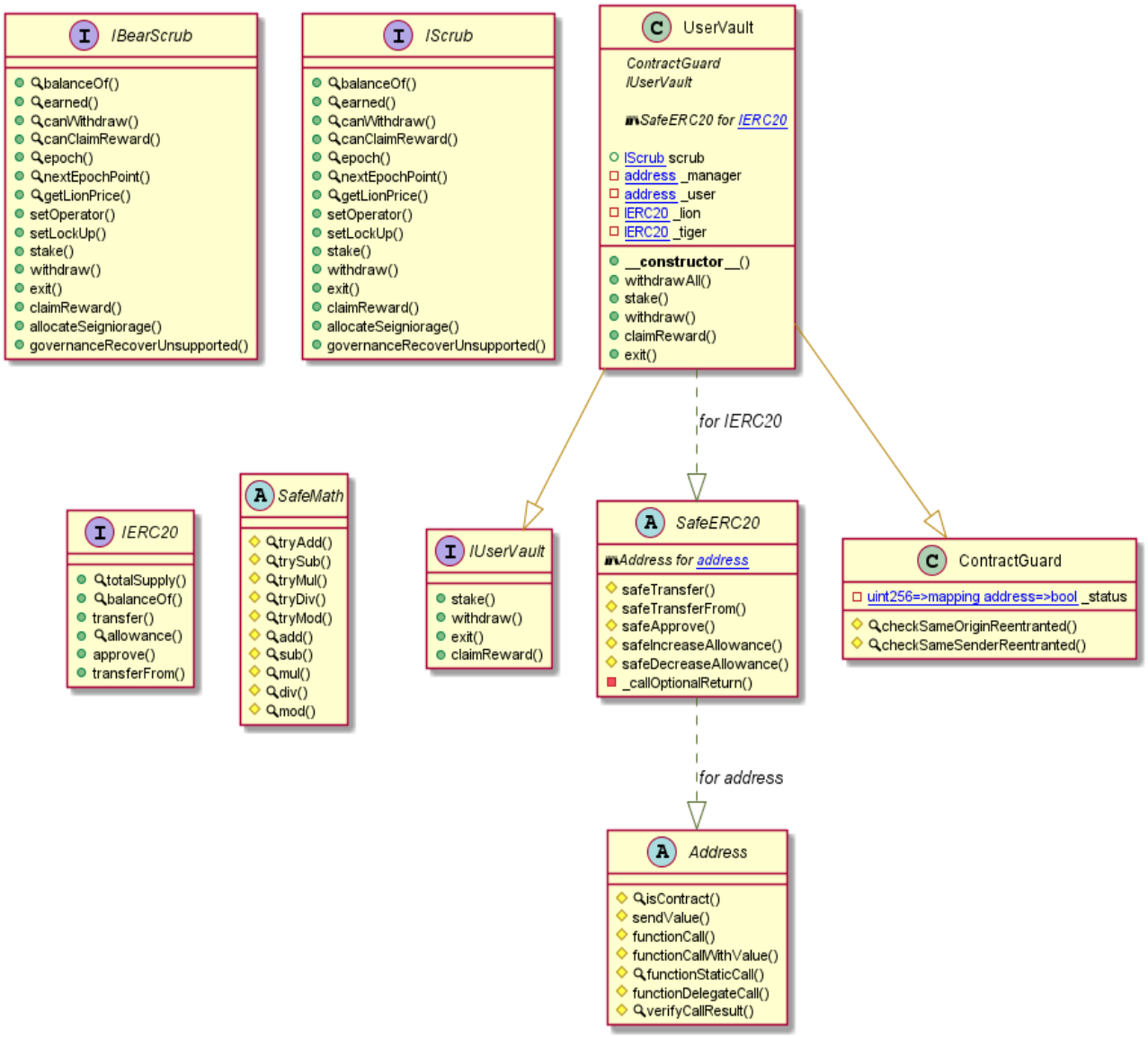
Tiger Diagram



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

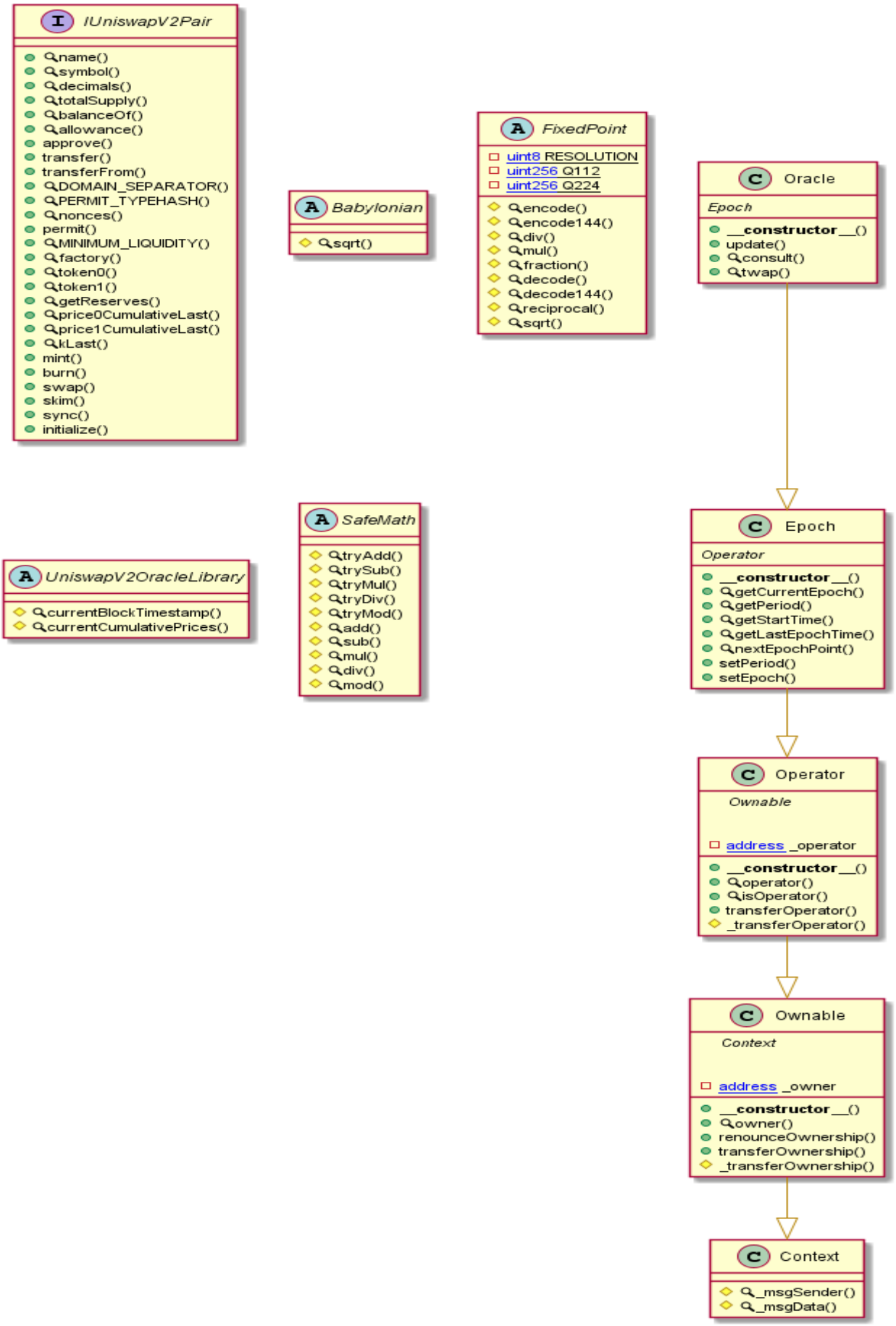
Email: audit@EtherAuthority.io

UserVault Diagram



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

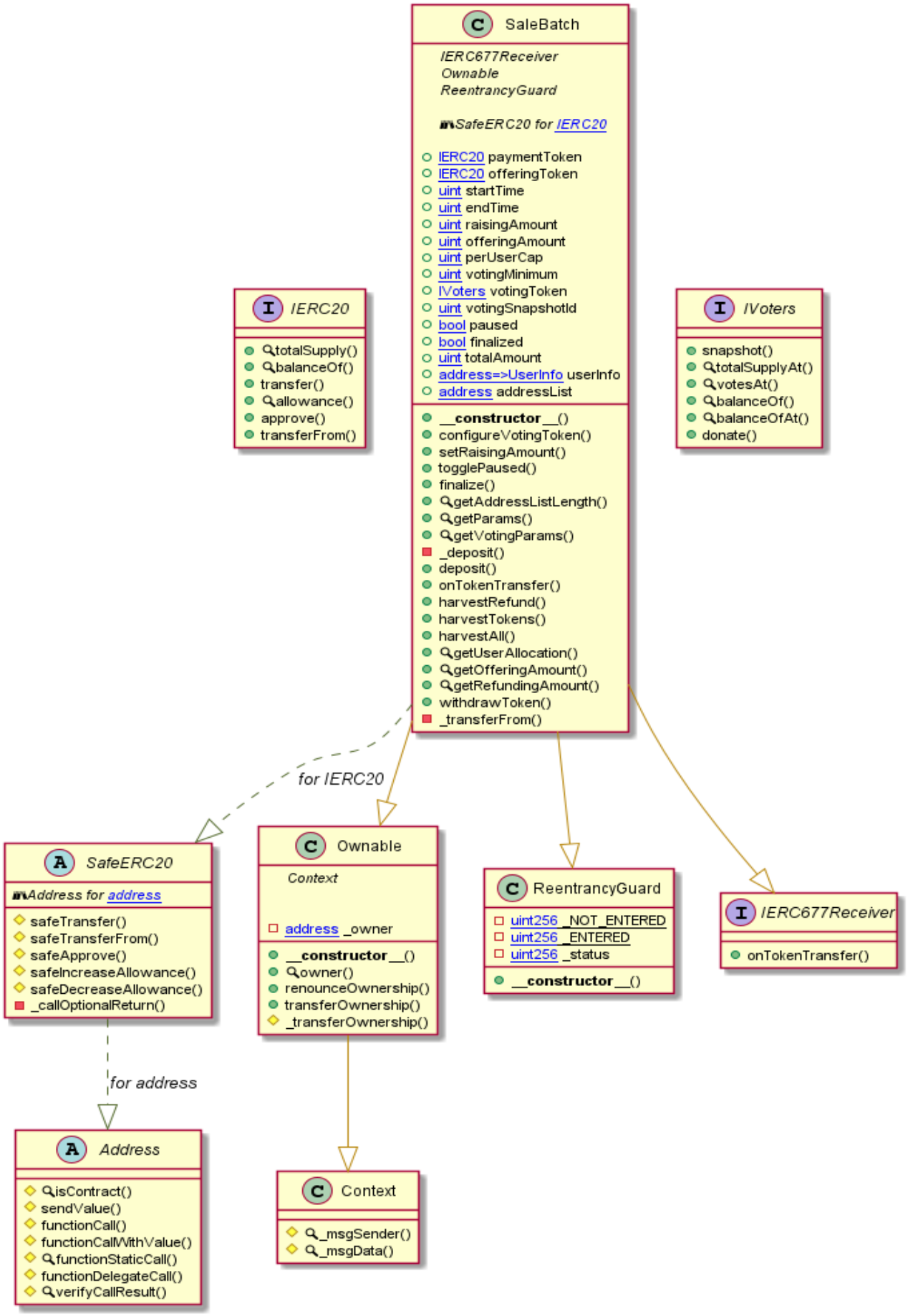
LionOracle Diagram



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

Email: audit@EtherAuthority.io

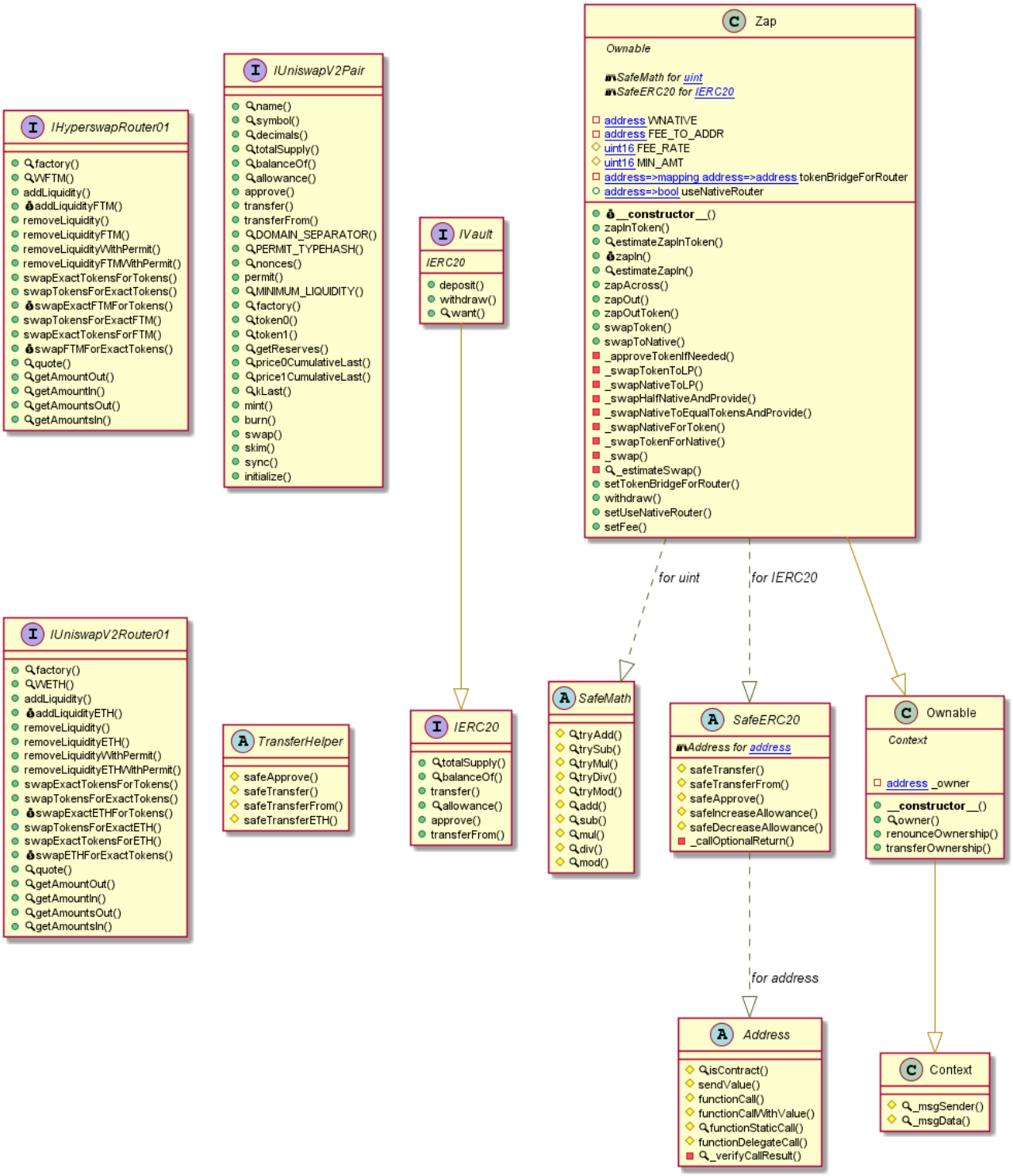
SaleBatch Diagram



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

Email: audit@EtherAuthority.io

Zap Diagram



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

Email: audit@EtherAuthority.io

Slither Results Log

Slither log >> BBond.sol

```
INFO:Detectors:
Context._msgData() (BBond.sol#464-466) is never used and should be removed
SafeMath.add(uint256,uint256) (BBond.sol#322-324) is never used and should be removed
SafeMath.div(uint256,uint256) (BBond.sol#364-366) is never used and should be removed
SafeMath.div(uint256,uint256,string) (BBond.sol#420-429) is never used and should be removed
SafeMath.mod(uint256,uint256) (BBond.sol#380-382) is never used and should be removed
SafeMath.mod(uint256,uint256,string) (BBond.sol#446-455) is never used and should be removed
SafeMath.mul(uint256,uint256) (BBond.sol#350-352) is never used and should be removed
SafeMath.sub(uint256,uint256) (BBond.sol#336-338) is never used and should be removed
SafeMath.sub(uint256,uint256,string) (BBond.sol#397-406) is never used and should be removed
SafeMath.tryAdd(uint256,uint256) (BBond.sol#251-257) is never used and should be removed
SafeMath.tryDiv(uint256,uint256) (BBond.sol#293-298) is never used and should be removed
SafeMath.tryMod(uint256,uint256) (BBond.sol#305-310) is never used and should be removed
SafeMath.tryMul(uint256,uint256) (BBond.sol#276-286) is never used and should be removed
SafeMath.trySub(uint256,uint256) (BBond.sol#264-269) is never used and should be removed
SafeMath8.add(uint8,uint8) (BBond.sol#22-27) is never used and should be removed
SafeMath8.div(uint8,uint8) (BBond.sol#96-98) is never used and should be removed
SafeMath8.div(uint8,uint8,string) (BBond.sol#112-118) is never used and should be removed
SafeMath8.mod(uint8,uint8) (BBond.sol#132-134) is never used and should be removed
SafeMath8.mod(uint8,uint8,string) (BBond.sol#148-151) is never used and should be removed
SafeMath8.mul(uint8,uint8) (BBond.sol#70-82) is never used and should be removed
SafeMath8.sub(uint8,uint8) (BBond.sol#39-41) is never used and should be removed
SafeMath8.sub(uint8,uint8,string) (BBond.sol#53-58) is never used and should be removed
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#dead-code
INFO:Detectors:
renounceOwnership() should be declared external:
        - Ownable.renounceOwnership() (BBond.sol#502-504)
transferOwnership(address) should be declared external:
        - Ownable.transferOwnership(address) (BBond.sol#510-513)
operator() should be declared external:
        - Operator.operator() (BBond.sol#536-538)
isOperator() should be declared external:
        - Operator.isOperator() (BBond.sol#545-547)
transferOperator(address) should be declared external:
        - Operator.transferOperator(address) (BBond.sol#549-551)
```

```
name() should be declared external:
        - ERC20.name() (BBond.sol#587-589)
symbol() should be declared external:
        - ERC20.symbol() (BBond.sol#595-597)
decimals() should be declared external:
        - ERC20.decimals() (BBond.sol#612-614)
totalSupply() should be declared external:
        - ERC20.totalSupply() (BBond.sol#619-621)
transfer(address,uint256) should be declared external:
        - ERC20.transfer(address,uint256) (BBond.sol#638-642)
approve(address,uint256) should be declared external:
        - ERC20.approve(address,uint256) (BBond.sol#661-665)
transferFrom(address,address,uint256) should be declared external:
        - ERC20.transferFrom(address,address,uint256) (BBond.sol#683-692)
increaseAllowance(address,uint256) should be declared external:
        - ERC20.increaseAllowance(address,uint256) (BBond.sol#706-710)
decreaseAllowance(address,uint256) should be declared external:
        - ERC20.decreaseAllowance(address,uint256) (BBond.sol#726-734)
mint(address,uint256) should be declared external:
        - BBond.mint(address,uint256) (BBond.sol#948-954)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#public-function-that-could-be-declared-external
INFO:Slither:BBond.sol analyzed (11 contracts with 75 detectors), 37 result(s) found
INFO:Slither:Use https://crytic.io/ to get access to additional detectors and Github integration
```

Slither log >> Bear.sol

```
INFO:Detectors:
Bear.setBurnThreshold(uint256) (Bear.sol#1035-1037) should emit an event for:
        - burnThreshold = _burnThreshold (Bear.sol#1036)
Bear.setTaxRate(uint256) (Bear.sol#1083-1087) should emit an event for:
        - taxRate = _taxRate (Bear.sol#1086)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#missing-events-arithmetic
INFO:Detectors:
Variable 'Bear._getLionPrice()._price (Bear.sol#1040)' in Bear._getLionPrice() (Bear.sol#1039-1045) potentially used before
eclaration: uint256(_price) (Bear.sol#1041)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#pre-declaration-usage-of-local-variables
INFO:Detectors:
Bear._updateTaxRate(uint256) (Bear.sol#1047-1057) has costly operations inside a loop:
        - taxRate = taxTiersRates[tierId] (Bear.sol#1052)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#costly-operations-inside-a-loop
INFO:Detectors:
Context._msgData() (Bear.sol#465-467) is never used and should be removed
SafeMath.add(uint256,uint256) (Bear.sol#323-325) is never used and should be removed
SafeMath.div(uint256,uint256,string) (Bear.sol#421-430) is never used and should be removed
SafeMath.mod(uint256,uint256) (Bear.sol#381-383) is never used and should be removed
SafeMath.mod(uint256,uint256,string) (Bear.sol#447-456) is never used and should be removed
SafeMath.tryAdd(uint256,uint256) (Bear.sol#252-258) is never used and should be removed
SafeMath.tryDiv(uint256,uint256) (Bear.sol#294-299) is never used and should be removed
SafeMath.tryMod(uint256,uint256) (Bear.sol#306-311) is never used and should be removed
SafeMath.tryMul(uint256,uint256) (Bear.sol#277-287) is never used and should be removed
SafeMath.trySub(uint256,uint256) (Bear.sol#265-270) is never used and should be removed
SafeMath8.add(uint8,uint8) (Bear.sol#23-28) is never used and should be removed
SafeMath8.div(uint8,uint8) (Bear.sol#97-99) is never used and should be removed
SafeMath8.div(uint8,uint8,string) (Bear.sol#113-119) is never used and should be removed
SafeMath8.mod(uint8,uint8) (Bear.sol#133-135) is never used and should be removed
SafeMath8.mod(uint8,uint8,string) (Bear.sol#149-152) is never used and should be removed
SafeMath8.mul(uint8,uint8) (Bear.sol#71-83) is never used and should be removed
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#dead-code
```

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

```
INFO:Detectors:
Parameter Bear.isAddressExcluded(address)._address (Bear.sol#1011) is not in mixedCase
Parameter Bear.setTaxTiersTwap(uint8,uint256)._index (Bear.sol#1015) is not in mixedCase
Parameter Bear.setTaxTiersTwap(uint8,uint256)._value (Bear.sol#1015) is not in mixedCase
Parameter Bear.setTaxTiersRate(uint8,uint256)._index (Bear.sol#1028) is not in mixedCase
Parameter Bear.setTaxTiersRate(uint8,uint256)._value (Bear.sol#1028) is not in mixedCase
Parameter Bear.setBurnThreshold(uint256)._burnThreshold (Bear.sol#1035) is not in mixedCase
Parameter Bear.setOracle(address)._oracle (Bear.sol#1067) is not in mixedCase
Parameter Bear.setTaxOffice(address)._taxOffice (Bear.sol#1072) is not in mixedCase
Parameter Bear.setTaxCollectorAddress(address)._taxCollectorAddress (Bear.sol#1078) is not in mixedCase
Parameter Bear.setTaxRate(uint256)._taxRate (Bear.sol#1083) is not in mixedCase
Parameter Bear.setBurnTax(bool)._burnTax (Bear.sol#1089) is not in mixedCase
Parameter Bear.excludeAddress(address)._address (Bear.sol#1093) is not in mixedCase
Parameter Bear.includeAddress(address)._address (Bear.sol#1099) is not in mixedCase
Function Bear.OpenTrade() (Bear.sol#1105-1108) is not in mixedCase
Parameter Bear.includeToWhitelist(address)._address (Bear.sol#1110) is not in mixedCase
Parameter Bear.excludeFromWhitlist(address)._address (Bear.sol#1116) is not in mixedCase
Parameter Bear.governanceRecoverUnsupported(IERC20,uint256,address)._token (Bear.sol#1199) is not in mixedCase
Parameter Bear.governanceRecoverUnsupported(IERC20,uint256,address)._amount (Bear.sol#1200) is not in mixedCase
Parameter Bear.governanceRecoverUnsupported(IERC20,uint256,address)._to (Bear.sol#1201) is not in mixedCase
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#conformance-to-solidity-naming-conventions
INFO:Detectors:
Bear.slitherConstructorConstantVariables() (Bear.sol#938-1205) uses literals with too many digits:
        - INITIAL_LAUNCHPAD_DISTRIBUTION = 40000000000000000000000 (Bear.sol#943)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#too-many-digits
INFO:Detectors:
renounceOwnership() should be declared external:
        - Ownable.renounceOwnership() (Bear.sol#503-505)
transferOwnership(address) should be declared external:
        - Ownable.transferOwnership(address) (Bear.sol#511-514)
operator() should be declared external:
        - Operator.operator() (Bear.sol#537-539)
transferOperator(address) should be declared external:
        - Operator.transferOperator(address) (Bear.sol#550-552)
name() should be declared external:
        - ERC20.name() (Bear.sol#588-590)
symbol() should be declared external:
```

```
symbol() should be declared external:
        - ERC20.symbol() (Bear.sol#596-598)
decimals() should be declared external:
        - ERC20.decimals() (Bear.sol#613-615)
totalSupply() should be declared external:
        - ERC20.totalSupply() (Bear.sol#620-622)
transfer(address,uint256) should be declared external:
        - ERC20.transfer(address,uint256) (Bear.sol#639-643)
approve(address,uint256) should be declared external:
        - ERC20.approve(address,uint256) (Bear.sol#662-666)
transferFrom(address,address,uint256) should be declared external:
        - Bear.transferFrom(address,address,uint256) (Bear.sol#1144-1168)
        - ERC20.transferFrom(address,address,uint256) (Bear.sol#684-693)
increaseAllowance(address,uint256) should be declared external:
        - ERC20.increaseAllowance(address,uint256) (Bear.sol#707-711)
decreaseAllowance(address,uint256) should be declared external:
        - ERC20.decreaseAllowance(address,uint256) (Bear.sol#727-736)
isAddressExcluded(address) should be declared external:
        - Bear.isAddressExcluded(address) (Bear.sol#1011-1013)
setTaxTiersTwap(uint8,uint256) should be declared external:
        - Bear.setTaxTiersTwap(uint8,uint256) (Bear.sol#1015-1026)
setTaxTiersRate(uint8,uint256) should be declared external:
        - Bear.setTaxTiersRate(uint8,uint256) (Bear.sol#1028-1033)
setBurnThreshold(uint256) should be declared external:
        - Bear.setBurnThreshold(uint256) (Bear.sol#1035-1037)
enableAutoCalculateTax() should be declared external:
        - Bear.enableAutoCalculateTax() (Bear.sol#1059-1061)
disableAutoCalculateTax() should be declared external:
        - Bear.disableAutoCalculateTax() (Bear.sol#1063-1065)
setOracle(address) should be declared external:
        - Bear.setOracle(address) (Bear.sol#1067-1070)
setTaxOffice(address) should be declared external:
        - Bear.setTaxOffice(address) (Bear.sol#1072-1076)
setTaxCollectorAddress(address) should be declared external:
        - Bear.setTaxCollectorAddress(address) (Bear.sol#1078-1081)
```

```
setTaxRate(uint256) should be declared external:
        - Bear.setTaxRate(uint256) (Bear.sol#1083-1087)
setBurnTax(bool) should be declared external:
        - Bear.setBurnTax(bool) (Bear.sol#1089-1091)
includeAddress(address) should be declared external:
        - Bear.includeAddress(address) (Bear.sol#1099-1103)
includeToWhitelist(address) should be declared external:
        - Bear.includeToWhitelist(address) (Bear.sol#1110-1114)
excludeFromWhitlist(address) should be declared external:
        - Bear.excludeFromWhitlist(address) (Bear.sol#1116-1120)
mint(address,uint256) should be declared external:
        - Bear.mint(address,uint256) (Bear.sol#1128-1134)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#public-function-that-could-be-declared-external
INFO:Slither:Bear.sol analyzed (11 contracts with 75 detectors), 74 result(s) found
INFO:Slither:Use https://crytic.io/ to get access to additional detectors and Github integration
```

Slither log >> BearScrub.sol

```
INFO:Detectors:
BearScrub.setRewardManager(address) (BearScrub.sol#772-774) should emit an event for:
        - rewardManager = _rewardManager (BearScrub.sol#773)
BearScrub.setOperator(address) (BearScrub.sol#776-778) should emit an event for:
        - operator = _operator (BearScrub.sol#777)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#missing-events-access-control
```

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

```
INFO:Detectors:
BearScrub.setLockUp(uint256,uint256) (BearScrub.sol#780-784) should emit an event for:
        - withdrawLockupEpochs = _withdrawLockupEpochs (BearScrub.sol#782)
        - rewardLockupEpochs = _rewardLockupEpochs (BearScrub.sol#783)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#missing-events-arithmetic
INFO:Detectors:
BearScrub.setRewardManager(address)._rewardManager (BearScrub.sol#772) lacks a zero-check on :
                - rewardManager = _rewardManager (BearScrub.sol#773)
BearScrub.setOperator(address)._operator (BearScrub.sol#776) lacks a zero-check on :
                - operator = _operator (BearScrub.sol#777)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#missing-zero-address-validation
INFO:Detectors:
Reentrancy in BearScrub.allocateSeigniorage(uint256) (BearScrub.sol#871-891):
        External calls:
        - bear.safeTransferFrom(msg.sender,address(this),amount) (BearScrub.sol#876)
        State variables written after the call(s):
        - masonryHistory.push(newSnapshot) (BearScrub.sol#888)
Reentrancy in BearScrub.withdraw(address,uint256) (BearScrub.sol#848-854):
        External calls:
        - claimReward(from_) (BearScrub.sol#851)
                - returndata = address(token).functionCall(data,SafeERC20: low-level call failed) (BearScrub.sol#609)
                - (success,returndata) = target.call{value: value}(data) (BearScrub.sol#160)
                - bear.safeTransfer(from_,reward) (BearScrub.sol#866)
        External calls sending eth:
        - claimReward(from_) (BearScrub.sol#851)
                - (success,returndata) = target.call{value: value}(data) (BearScrub.sol#160)
        State variables written after the call(s):
        - super.withdraw(from_,amount) (BearScrub.sol#852)
                - _totalSupply = _totalSupply.sub(amount) (BearScrub.sol#645)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#reentrancy-vulnerabilities-2
INFO:Detectors:
Reentrancy in BearScrub.allocateSeigniorage(uint256) (BearScrub.sol#871-891):
        External calls:
        - bear.safeTransferFrom(msg.sender,address(this),amount) (BearScrub.sol#876)
        Event emitted after the call(s):
        - RewardAdded(msg.sender,amount) (BearScrub.sol#890)
Reentrancy in BearScrub.claimReward(address) (BearScrub.sol#860-869):
```

```
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#reentrancy-vulnerabilities-3
INFO:Detectors:
Address.verifyCallResult(bool,bytes,string) (BearScrub.sol#224-244) uses assembly
        - INLINE ASM (BearScrub.sol#236-239)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#assembly-usage
INFO:Detectors:
Address.functionCall(address,bytes) (BearScrub.sol#108-110) is never used and should be removed
Address.functionCallWithValue(address,bytes,uint256) (BearScrub.sol#137-143) is never used and should be removed
Address.functionDelegateCall(address,bytes) (BearScrub.sol#197-199) is never used and should be removed
Address.functionDelegateCall(address,bytes,string) (BearScrub.sol#207-216) is never used and should be removed
Address.functionStaticCall(address,bytes) (BearScrub.sol#170-172) is never used and should be removed
Address.functionStaticCall(address,bytes,string) (BearScrub.sol#180-189) is never used and should be removed
Address.sendValue(address,uint256) (BearScrub.sol#83-88) is never used and should be removed
SafeERC20.safeApprove(IERC20,address,uint256) (BearScrub.sol#561-574) is never used and should be removed
SafeERC20.safeDecreaseAllowance(IERC20,address,uint256) (BearScrub.sol#585-596) is never used and should be removed
SafeERC20.safeIncreaseAllowance(IERC20,address,uint256) (BearScrub.sol#576-583) is never used and should be removed
SafeMath.div(uint256,uint256,string) (BearScrub.sol#496-505) is never used and should be removed
SafeMath.mod(uint256,uint256) (BearScrub.sol#456-458) is never used and should be removed
SafeMath.mod(uint256,uint256,string) (BearScrub.sol#522-531) is never used and should be removed
SafeMath.sub(uint256,uint256,string) (BearScrub.sol#473-482) is never used and should be removed
SafeMath.tryAdd(uint256,uint256) (BearScrub.sol#327-333) is never used and should be removed
SafeMath.tryDiv(uint256,uint256) (BearScrub.sol#369-374) is never used and should be removed
SafeMath.tryMod(uint256,uint256) (BearScrub.sol#381-386) is never used and should be removed
SafeMath.tryMul(uint256,uint256) (BearScrub.sol#352-362) is never used and should be removed
SafeMath.trySub(uint256,uint256) (BearScrub.sol#340-345) is never used and should be removed
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#dead-code
INFO:Detectors:
Low level call in Address.sendValue(address,uint256) (BearScrub.sol#83-88):
        - (success) = recipient.call{value: amount}() (BearScrub.sol#86)
Low level call in Address.functionCallWithValue(address,bytes,uint256,string) (BearScrub.sol#151-162):
        - (success,returndata) = target.call{value: value}(data) (BearScrub.sol#160)
Low level call in Address.functionStaticCall(address,bytes,string) (BearScrub.sol#180-189):
        - (success,returndata) = target.staticcall(data) (BearScrub.sol#187)
Low level call in Address.functionDelegateCall(address,bytes,string) (BearScrub.sol#207-216):
        - (success,returndata) = target.delegatecall(data) (BearScrub.sol#214)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#low-level-calls
```

```
INFO:Detectors:
Parameter BearScrub.initialize(IERC20,IERC20,IBearTreasury)._bear (BearScrub.sol#753) is not in mixedCase
Parameter BearScrub.initialize(IERC20,IERC20,IBearTreasury)._share (BearScrub.sol#754) is not in mixedCase
Parameter BearScrub.initialize(IERC20,IERC20,IBearTreasury)._treasury (BearScrub.sol#755) is not in mixedCase
Parameter BearScrub.setRewardManager(address)._rewardManager (BearScrub.sol#772) is not in mixedCase
Parameter BearScrub.setOperator(address)._operator (BearScrub.sol#776) is not in mixedCase
Parameter BearScrub.setLockUp(uint256,uint256)._withdrawLockupEpochs (BearScrub.sol#780) is not in mixedCase
Parameter BearScrub.setLockUp(uint256,uint256)._rewardLockupEpochs (BearScrub.sol#780) is not in mixedCase
Parameter BearScrub.governanceRecoverUnsupported(IERC20,uint256,address)._token (BearScrub.sol#893) is not in mixedCase
Parameter BearScrub.governanceRecoverUnsupported(IERC20,uint256,address)._amount (BearScrub.sol#893) is not in mixedCase
Parameter BearScrub.governanceRecoverUnsupported(IERC20,uint256,address)._to (BearScrub.sol#893) is not in mixedCase
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#conformance-to-solidity-naming-conventions
INFO:Detectors:
initialize(IERC20,IERC20,IBearTreasury) should be declared external:
        - BearScrub.initialize(IERC20,IERC20,IBearTreasury) (BearScrub.sol#752-770)
rewardPerShare() should be declared external:
        - BearScrub.rewardPerShare() (BearScrub.sol#828-830)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#public-function-that-could-be-declared-external
INFO:Slither:BearScrub.sol analyzed (9 contracts with 75 detectors), 47 result(s) found
INFO:Slither:Use https://crytic.io/ to get access to additional detectors and Github integration
```

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

Slither log >> BearTreasury.sol

```
INFO:Detectors:
BearTreasury.setOperator(address) (BearTreasury.sol#1058-1060) should emit an event for:
        - operator = _operator (BearTreasury.sol#1059)
BearTreasury.setScrub(address) (BearTreasury.sol#1062-1064) should emit an event for:
        - scrub = _scrub (BearTreasury.sol#1063)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#missing-events-access-control
INFO:Detectors:
BearTreasury.setBearPriceCeiling(uint256) (BearTreasury.sol#1070-1073) should emit an event for:
        - bearPriceCeiling = _bearPriceCeiling (BearTreasury.sol#1072)
BearTreasury.setMaxSupplyExpansionPercents(uint256) (BearTreasury.sol#1075-1078) should emit an event for:
        - maxSupplyExpansionPercent = _maxSupplyExpansionPercent (BearTreasury.sol#1077)
BearTreasury.setBondDepletionFloorPercent(uint256) (BearTreasury.sol#1101-1104) should emit an event for:
        - bondDepletionFloorPercent = _bondDepletionFloorPercent (BearTreasury.sol#1103)
BearTreasury.setMaxDebtRatioPercent(uint256) (BearTreasury.sol#1111-1114) should emit an event for:
        - maxDebtRatioPercent = _maxDebtRatioPercent (BearTreasury.sol#1113)
BearTreasury.setBootstrap(uint256,uint256) (BearTreasury.sol#1116-1121) should emit an event for:
        - bootstrapEpochs = _bootstrapEpochs (BearTreasury.sol#1119)
        - bootstrapSupplyExpansionPercent = _bootstrapSupplyExpansionPercent (BearTreasury.sol#1120)
BearTreasury.setExtraFunds(address,uint256,address,uint256) (BearTreasury.sol#1123-1137) should emit an event for:
        - daoFundSharedPercent = _daoFundSharedPercent (BearTreasury.sol#1134)
        - devFundSharedPercent = _devFundSharedPercent (BearTreasury.sol#1136)
BearTreasury.setMaxDiscountRate(uint256) (BearTreasury.sol#1139-1141) should emit an event for:
        - maxDiscountRate = _maxDiscountRate (BearTreasury.sol#1140)
```

```
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#missing-events-arithmetic
INFO:Detectors:
BearTreasury.initialize(address,address,address,address,uint256)._bear (BearTreasury.sol#1019) lacks a zero-check on :
                - bear = _bear (BearTreasury.sol#1026)
BearTreasury.initialize(address,address,address,address,uint256)._bbond (BearTreasury.sol#1020) lacks a zero-check on :
                - bbond = _bbond (BearTreasury.sol#1027)
BearTreasury.initialize(address,address,address,address,uint256)._bearOracle (BearTreasury.sol#1022) lacks a zero-check on :
                - bearOracle = _bearOracle (BearTreasury.sol#1029)
BearTreasury.initialize(address,address,address,address,uint256)._scrub (BearTreasury.sol#1023) lacks a zero-check on :
                - scrub = _scrub (BearTreasury.sol#1030)
BearTreasury.setOperator(address)._operator (BearTreasury.sol#1058) lacks a zero-check on :
                - operator = _operator (BearTreasury.sol#1059)
BearTreasury.setScrub(address)._scrub (BearTreasury.sol#1062) lacks a zero-check on :
                - scrub = _scrub (BearTreasury.sol#1063)
BearTreasury.setBearOracle(address)._bearOracle (BearTreasury.sol#1066) lacks a zero-check on :
                - bearOracle = _bearOracle (BearTreasury.sol#1067)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#missing-zero-address-validation
INFO:Detectors:
BearTreasury.getBearCirculatingSupply() (BearTreasury.sol#1174-1182) has external calls inside a loop: balanceExcluded = bal
nceExcluded.add(bearErc20.balanceOf(excludedFromTotalSupply[entryId])) (BearTreasury.sol#1179)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation/#calls-inside-a-loop
INFO:Detectors:
Variable 'BearTreasury.getBearPrice().price (BearTreasury.sol#936)' in BearTreasury.getBearPrice() (BearTreasury.sol#935-941
 potentially used before declaration: uint256(price) (BearTreasury.sol#937)
Variable 'BearTreasury.getBearUpdatedPrice().price (BearTreasury.sol#944)' in BearTreasury.getBearUpdatedPrice() (BearTreasu
y.sol#943-949) potentially used before declaration: uint256(price) (BearTreasury.sol#945)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#pre-declaration-usage-of-local-variables
```

```
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#reentrancy-vulnerabilities-3
INFO:Detectors:
Address.verifyCallResult(bool,bytes,string) (BearTreasury.sol#289-309) uses assembly
        - INLINE ASM (BearTreasury.sol#301-304)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#assembly-usage
INFO:Detectors:
BearTreasury._calculateMaxSupplyExpansionPercent(uint256) (BearTreasury.sol#1264-1272) has costly operations inside a loop:
        - maxSupplyExpansionPercent = maxExpansionTiers[tierId] (BearTreasury.sol#1267)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#costly-operations-inside-a-loop
INFO:Detectors:
Address.functionCall(address,bytes) (BearTreasury.sol#173-175) is never used and should be removed
Address.functionCallWithValue(address,bytes,uint256) (BearTreasury.sol#202-208) is never used and should be removed
Address.functionDelegateCall(address,bytes) (BearTreasury.sol#262-264) is never used and should be removed
Address.functionDelegateCall(address,bytes,string) (BearTreasury.sol#272-281) is never used and should be removed
Address.functionStaticCall(address,bytes) (BearTreasury.sol#235-237) is never used and should be removed
Address.functionStaticCall(address,bytes,string) (BearTreasury.sol#245-254) is never used and should be removed
Address.sendValue(address,uint256) (BearTreasury.sol#148-153) is never used and should be removed
Context._msgData() (BearTreasury.sol#688-690) is never used and should be removed
Math.average(uint256,uint256) (BearTreasury.sol#81-84) is never used and should be removed
Math.ceilDiv(uint256,uint256) (BearTreasury.sol#92-95) is never used and should be removed
Math.max(uint256,uint256) (BearTreasury.sol#66-68) is never used and should be removed
SafeERC20.safeDecreaseAllowance(IERC20,address,uint256) (BearTreasury.sol#650-661) is never used and should be removed
```

```
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#dead-code
INFO:Detectors:
Low level call in Address.sendValue(address,uint256) (BearTreasury.sol#148-153):
        - (success) = recipient.call{value: amount}() (BearTreasury.sol#151)
Low level call in Address.functionCallWithValue(address,bytes,uint256,string) (BearTreasury.sol#216-227):
        - (success,returndata) = target.call{value: value}(data) (BearTreasury.sol#225)
Low level call in Address.functionStaticCall(address,bytes,string) (BearTreasury.sol#245-254):
        - (success,returndata) = target.staticcall(data) (BearTreasury.sol#252)
Low level call in Address.functionDelegateCall(address,bytes,string) (BearTreasury.sol#272-281):
        - (success,returndata) = target.delegatecall(data) (BearTreasury.sol#279)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#low-level-calls
INFO:Detectors:
Parameter BearTreasury.initialize(address,address,address,address,uint256)._bear (BearTreasury.sol#1019) is not in mixedCase
Parameter BearTreasury.initialize(address,address,address,address,uint256)._bbond (BearTreasury.sol#1020) is not in mixedCas
Parameter BearTreasury.initialize(address,address,address,address,uint256)._bearOracle (BearTreasury.sol#1022) is not in mix
dCase
Parameter BearTreasury.initialize(address,address,address,address,uint256)._scrub (BearTreasury.sol#1023) is not in mixedCas
Parameter BearTreasury.initialize(address,address,address,address,uint256)._startTime (BearTreasury.sol#1024) is not in mixe
Case
Parameter BearTreasury.setOperator(address)._operator (BearTreasury.sol#1058) is not in mixedCase
Parameter BearTreasury.setScrub(address)._scrub (BearTreasury.sol#1062) is not in mixedCase
Parameter BearTreasury.setBearOracle(address)._bearOracle (BearTreasury.sol#1066) is not in mixedCase
Parameter BearTreasury.setBearPriceCeiling(uint256)._bearPriceCeiling (BearTreasury.sol#1070) is not in mixedCase
Parameter BearTreasury.setMaxSupplyExpansionPercents(uint256)._maxSupplyExpansionPercent (BearTreasury.sol#1075) is not in m
xedCase
Parameter BearTreasury.setSupplyTiersEntry(uint8,uint256)._index (BearTreasury.sol#1080) is not in mixedCase
Parameter BearTreasury.setSupplyTiersEntry(uint8,uint256)._value (BearTreasury.sol#1080) is not in mixedCase
```

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

```
INFO:Detectors:
BearTreasury.initialize(address,address,address,address,uint256) (BearTreasury.sol#1018-1056) uses literals with too many di
its:
        - supplyTiers = (0,2000000000000000000000000,3000000000000000000000000,6000000000000000000000000,1200000000000000000
000000,1800000000000000000000000,2400000000000000000000000,6000000000000000000000000) (BearTreasury.sol#1037)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#too-many-digits
INFO:Detectors:
renounceOwnership() should be declared external:
        - Ownable.renounceOwnership() (BearTreasury.sol#726-728)
transferOwnership(address) should be declared external:
        - Ownable.transferOwnership(address) (BearTreasury.sol#734-737)
operator() should be declared external:
        - Operator.operator() (BearTreasury.sol#760-762)
isOperator() should be declared external:
        - Operator.isOperator() (BearTreasury.sol#769-771)
transferOperator(address) should be declared external:
        - Operator.transferOperator(address) (BearTreasury.sol#773-775)
isInitialized() should be declared external:
        - BearTreasury.isInitialized() (BearTreasury.sol#925-927)
getBearUpdatedPrice() should be declared external:
        - BearTreasury.getBearUpdatedPrice() (BearTreasury.sol#943-949)
getReserve() should be declared external:
        - BearTreasury.getReserve() (BearTreasury.sol#952-954)
getBurnableBearLeft() should be declared external:
        - BearTreasury.getBurnableBearLeft() (BearTreasury.sol#956-968)
getRedeemableBonds() should be declared external:
        - BearTreasury.getRedeemableBonds() (BearTreasury.sol#970-979)
initialize(address,address,address,address,uint256) should be declared external:
        - BearTreasury.initialize(address,address,address,address,uint256) (BearTreasury.sol#1018-1056)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#public-function-that-could-be-declared-external
INFO:Slither:BearTreasury.sol analyzed (13 contracts with 75 detectors), 131 result(s) found
INFO:Slither:Use https://crytic.io/ to get access to additional detectors and Github integration
```

Slither log >> LionOracle.sol

```
INFO:Detectors:
UniswapV2OracleLibrary.currentCumulativePrices(address) (LionOracle.sol#196-220) uses timestamp for comparisons
        Dangerous comparisons:
        - blockTimestampLast != blockTimestamp (LionOracle.sol#211)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#block-timestamp
INFO:Detectors:
Babylonian.sqrt(uint256) (LionOracle.sol#98-110) is never used and should be removed
Context._msgData() (LionOracle.sol#445-447) is never used and should be removed
FixedPoint.decode(FixedPoint.uq112x112) (LionOracle.sol#164-166) is never used and should be removed
FixedPoint.div(FixedPoint.uq112x112,uint112) (LionOracle.sol#143-146) is never used and should be removed
FixedPoint.encode(uint112) (LionOracle.sol#133-135) is never used and should be removed
FixedPoint.encode144(uint144) (LionOracle.sol#138-140) is never used and should be removed
FixedPoint.reciprocal(FixedPoint.uq112x112) (LionOracle.sol#174-177) is never used and should be removed
FixedPoint.sqrt(FixedPoint.uq112x112) (LionOracle.sol#180-182) is never used and should be removed
SafeMath.div(uint256,uint256) (LionOracle.sol#345-347) is never used and should be removed
SafeMath.div(uint256,uint256,string) (LionOracle.sol#401-410) is never used and should be removed
SafeMath.mod(uint256,uint256) (LionOracle.sol#361-363) is never used and should be removed
SafeMath.mod(uint256,uint256,string) (LionOracle.sol#427-436) is never used and should be removed
SafeMath.mul(uint256,uint256) (LionOracle.sol#331-333) is never used and should be removed
SafeMath.sub(uint256,uint256,string) (LionOracle.sol#378-387) is never used and should be removed
SafeMath.tryAdd(uint256,uint256) (LionOracle.sol#232-238) is never used and should be removed
SafeMath.tryDiv(uint256,uint256) (LionOracle.sol#274-279) is never used and should be removed
SafeMath.tryMod(uint256,uint256) (LionOracle.sol#286-291) is never used and should be removed
SafeMath.tryMul(uint256,uint256) (LionOracle.sol#257-267) is never used and should be removed
SafeMath.trySub(uint256,uint256) (LionOracle.sol#245-250) is never used and should be removed
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#dead-code
INFO:Detectors:
Function IUniswapV2Pair.DOMAIN_SEPARATOR() (LionOracle.sol#35) is not in mixedCase
Function IUniswapV2Pair.PERMIT_TYPEHASH() (LionOracle.sol#37) is not in mixedCase
Function IUniswapV2Pair.MINIMUM_LIQUIDITY() (LionOracle.sol#56) is not in mixedCase
Struct FixedPoint.uq112x112 (LionOracle.sol#118-120) is not in CapWords
Struct FixedPoint.uq144x112 (LionOracle.sol#124-126) is not in CapWords
Parameter Epoch.setPeriod(uint256)._period (LionOracle.sol#612) is not in mixedCase
Parameter Epoch.setEpoch(uint256)._epoch (LionOracle.sol#617) is not in mixedCase
Parameter Oracle.consult(address,uint256)._token (LionOracle.sol#684) is not in mixedCase
Parameter Oracle.consult(address,uint256)._amountIn (LionOracle.sol#684) is not in mixedCase
```

```
Variable Oracle.twap(address,uint256).price0Cumulative (LionOracle.sol#694) is too similar to Oracle.update().price1Cumulati
e (LionOracle.sol#663)
Variable Oracle.twap(address,uint256).price0Cumulative (LionOracle.sol#694) is too similar to Oracle.twap(address,uint256).p
ice1Cumulative (LionOracle.sol#694)
Variable Oracle.update().price0Cumulative (LionOracle.sol#663) is too similar to Oracle.twap(address,uint256).price1Cumulati
e (LionOracle.sol#694)
Variable Oracle.price0CumulativeLast (LionOracle.sol#636) is too similar to Oracle.price1CumulativeLast (LionOracle.sol#637)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#variable-names-are-too-similar
INFO:Detectors:
renounceOwnership() should be declared external:
        - Ownable.renounceOwnership() (LionOracle.sol#483-485)
transferOwnership(address) should be declared external:
        - Ownable.transferOwnership(address) (LionOracle.sol#491-494)
isOperator() should be declared external:
        - Operator.isOperator() (LionOracle.sol#526-528)
transferOperator(address) should be declared external:
        - Operator.transferOperator(address) (LionOracle.sol#530-532)
getCurrentEpoch() should be declared external:
        - Epoch.getCurrentEpoch() (LionOracle.sol#590-592)
getPeriod() should be declared external:
        - Epoch.getPeriod() (LionOracle.sol#594-596)
getStartTime() should be declared external:
        - Epoch.getStartTime() (LionOracle.sol#598-600)
getLastEpochTime() should be declared external:
        - Epoch.getLastEpochTime() (LionOracle.sol#602-604)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#public-function-that-could-be-declared-external
INFO:Slither:LionOracle.sol analyzed (10 contracts with 75 detectors), 48 result(s) found
INFO:Slither:Use https://crytic.io/ to get access to additional detectors and Github integration
```

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

Slither log >> RewardManager.sol

```
INFO:Detectors:
UserVault.constructor(address,address,IERC20,IERC20,address).manager_ (RewardManager.sol#701) lacks a zero-check on :
                - _manager = manager_ (RewardManager.sol#702)
UserVault.constructor(address,address,IERC20,IERC20,address).user_ (RewardManager.sol#701) lacks a zero-check on :
                - _user = user_ (RewardManager.sol#703)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#missing-zero-address-validation
INFO:Detectors:
Address.verifyCallResult(bool,bytes,string) (RewardManager.sol#267-287) uses assembly
        - INLINE ASM (RewardManager.sol#279-282)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#assembly-usage
INFO:Detectors:
Address.functionCall(address,bytes) (RewardManager.sol#151-153) is never used and should be removed
Address.functionCallWithValue(address,bytes,uint256) (RewardManager.sol#180-186) is never used and should be removed
Address.functionDelegateCall(address,bytes) (RewardManager.sol#240-242) is never used and should be removed
Address.functionDelegateCall(address,bytes,string) (RewardManager.sol#250-259) is never used and should be removed
Address.functionStaticCall(address,bytes) (RewardManager.sol#213-215) is never used and should be removed
Address.functionStaticCall(address,bytes,string) (RewardManager.sol#223-232) is never used and should be removed
Address.sendValue(address,uint256) (RewardManager.sol#126-131) is never used and should be removed
SafeERC20.safeApprove(IERC20,address,uint256) (RewardManager.sol#604-617) is never used and should be removed
```

```
SafeMath.sub(uint256,uint256,string) (RewardManager.sol#516-525) is never used and should be removed
SafeMath.tryAdd(uint256,uint256) (RewardManager.sol#370-376) is never used and should be removed
SafeMath.tryDiv(uint256,uint256) (RewardManager.sol#412-417) is never used and should be removed
SafeMath.tryMod(uint256,uint256) (RewardManager.sol#424-429) is never used and should be removed
SafeMath.tryMul(uint256,uint256) (RewardManager.sol#395-405) is never used and should be removed
SafeMath.trySub(uint256,uint256) (RewardManager.sol#383-388) is never used and should be removed
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#dead-code
INFO:Detectors:
Low level call in Address.sendValue(address,uint256) (RewardManager.sol#126-131):
        - (success) = recipient.call{value: amount}() (RewardManager.sol#129)
Low level call in Address.functionCallWithValue(address,bytes,uint256,string) (RewardManager.sol#194-205):
        - (success,returndata) = target.call{value: value}(data) (RewardManager.sol#203)
Low level call in Address.functionStaticCall(address,bytes,string) (RewardManager.sol#223-232):
        - (success,returndata) = target.staticcall(data) (RewardManager.sol#230)
Low level call in Address.functionDelegateCall(address,bytes,string) (RewardManager.sol#250-259):
        - (success,returndata) = target.delegatecall(data) (RewardManager.sol#257)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#low-level-calls
INFO:Detectors:
Parameter UserVault.stake(uint256)._amount (RewardManager.sol#717) is not in mixedCase
Parameter UserVault.withdraw(uint256)._amount (RewardManager.sol#721) is not in mixedCase
Variable RewardManager._vaultDirectory (RewardManager.sol#745) is not in mixedCase
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#conformance-to-solidity-naming-conventions
INFO:Detectors:
earned(address) should be declared external:
        - RewardManager.earned(address) (RewardManager.sol#759-764)
claimRewards() should be declared external:
        - RewardManager.claimRewards() (RewardManager.sol#766-767)
stake(uint256) should be declared external:
        - RewardManager.stake(uint256) (RewardManager.sol#769-780)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#public-function-that-could-be-declared-external
INFO:Slither:RewardManager.sol analyzed (10 contracts with 75 detectors), 41 result(s) found
INFO:Slither:Use https://crytic.io/ to get access to additional detectors and Github integration
```

Slither log >> Tiger.sol

```
INFO:Detectors:
Tiger.setBurnThreshold(uint256) (Tiger.sol#1038-1040) should emit an event for:
        - burnThreshold = _burnThreshold (Tiger.sol#1039)
Tiger.setTaxRate(uint256) (Tiger.sol#1086-1090) should emit an event for:
        - taxRate = _taxRate (Tiger.sol#1089)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#missing-events-arithmetic
INFO:Detectors:
Variable 'Tiger._getLionPrice()._price (Tiger.sol#1043)' in Tiger._getLionPrice() (Tiger.sol#1042-1048) potentially used bef
re declaration: uint256(_price) (Tiger.sol#1044)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#pre-declaration-usage-of-local-variables
INFO:Detectors:
Tiger._updateTaxRate(uint256) (Tiger.sol#1050-1060) has costly operations inside a loop:
        - taxRate = taxTiersRates[tierId] (Tiger.sol#1055)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#costly-operations-inside-a-loop
INFO:Detectors:
Context._msgData() (Tiger.sol#467-469) is never used and should be removed
SafeMath.add(uint256,uint256) (Tiger.sol#325-327) is never used and should be removed
SafeMath.div(uint256,uint256,string) (Tiger.sol#423-432) is never used and should be removed
SafeMath.mod(uint256,uint256) (Tiger.sol#383-385) is never used and should be removed
SafeMath.mod(uint256,uint256,string) (Tiger.sol#449-458) is never used and should be removed
SafeMath.tryAdd(uint256,uint256) (Tiger.sol#254-260) is never used and should be removed
SafeMath.tryDiv(uint256,uint256) (Tiger.sol#296-301) is never used and should be removed
SafeMath.tryMod(uint256,uint256) (Tiger.sol#308-313) is never used and should be removed
SafeMath.tryMul(uint256,uint256) (Tiger.sol#279-289) is never used and should be removed
SafeMath.trySub(uint256,uint256) (Tiger.sol#267-272) is never used and should be removed
SafeMath8.add(uint8,uint8) (Tiger.sol#22-27) is never used and should be removed
SafeMath8.div(uint8,uint8) (Tiger.sol#96-98) is never used and should be removed
SafeMath8.div(uint8,uint8,string) (Tiger.sol#112-118) is never used and should be removed
SafeMath8.mod(uint8,uint8) (Tiger.sol#132-134) is never used and should be removed
SafeMath8.mod(uint8,uint8,string) (Tiger.sol#148-151) is never used and should be removed
SafeMath8.mul(uint8,uint8) (Tiger.sol#70-82) is never used and should be removed
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#dead-code
INFO:Detectors:
Parameter Tiger.isAddressExcluded(address)._address (Tiger.sol#1014) is not in mixedCase
Parameter Tiger.setTaxTiersTwap(uint8,uint256)._index (Tiger.sol#1018) is not in mixedCase
Parameter Tiger.setTaxTiersTwap(uint8,uint256)._value (Tiger.sol#1018) is not in mixedCase
Parameter Tiger.setTaxTiersRate(uint8,uint256)._index (Tiger.sol#1031) is not in mixedCase
```

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

```
Parameter Tiger.governanceRecoverUnsupported(IERC20,uint256,address)._to (Tiger.sol#1204) is not in mixedCase
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#conformance-to-solidity-naming-conventions
INFO:Detectors:
Tiger.slitherConstructorConstantVariables() (Tiger.sol#941-1208) uses literals with too many digits:
        - INITIAL_LAUNCHPAD_DISTRIBUTION = 40000000000000000000000 (Tiger.sol#946)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#too-many-digits
INFO:Detectors:
renounceOwnership() should be declared external:
        - Ownable.renounceOwnership() (Tiger.sol#505-507)
transferOwnership(address) should be declared external:
        - Ownable.transferOwnership(address) (Tiger.sol#513-516)
operator() should be declared external:
        - Operator.operator() (Tiger.sol#539-541)
transferOperator(address) should be declared external:
        - Operator.transferOperator(address) (Tiger.sol#552-554)
name() should be declared external:
        - ERC20.name() (Tiger.sol#590-592)
symbol() should be declared external:
        - ERC20.symbol() (Tiger.sol#598-600)
decimals() should be declared external:
        - ERC20.decimals() (Tiger.sol#615-617)
totalSupply() should be declared external:
        - ERC20.totalSupply() (Tiger.sol#622-624)
transfer(address,uint256) should be declared external:
        - ERC20.transfer(address,uint256) (Tiger.sol#641-645)
approve(address,uint256) should be declared external:
        - ERC20.approve(address,uint256) (Tiger.sol#664-668)
transferFrom(address,address,uint256) should be declared external:
        - ERC20.transferFrom(address,address,uint256) (Tiger.sol#686-695)
        - Tiger.transferFrom(address,address,uint256) (Tiger.sol#1147-1171)
increaseAllowance(address,uint256) should be declared external:
        - ERC20.increaseAllowance(address,uint256) (Tiger.sol#709-713)
decreaseAllowance(address,uint256) should be declared external:
        - ERC20.decreaseAllowance(address,uint256) (Tiger.sol#729-738)
isAddressExcluded(address) should be declared external:
        - Tiger.isAddressExcluded(address) (Tiger.sol#1014-1016)
```

```
isAddressExcluded(address) should be declared external:
        - Tiger.isAddressExcluded(address) (Tiger.sol#1014-1016)
setTaxTiersTwap(uint8,uint256) should be declared external:
        - Tiger.setTaxTiersTwap(uint8,uint256) (Tiger.sol#1018-1029)
setTaxTiersRate(uint8,uint256) should be declared external:
        - Tiger.setTaxTiersRate(uint8,uint256) (Tiger.sol#1031-1036)
setBurnThreshold(uint256) should be declared external:
        - Tiger.setBurnThreshold(uint256) (Tiger.sol#1038-1040)
enableAutoCalculateTax() should be declared external:
        - Tiger.enableAutoCalculateTax() (Tiger.sol#1062-1064)
disableAutoCalculateTax() should be declared external:
        - Tiger.disableAutoCalculateTax() (Tiger.sol#1066-1068)
setOracle(address) should be declared external:
        - Tiger.setOracle(address) (Tiger.sol#1070-1073)
setTaxOffice(address) should be declared external:
        - Tiger.setTaxOffice(address) (Tiger.sol#1075-1079)
setTaxCollectorAddress(address) should be declared external:
        - Tiger.setTaxCollectorAddress(address) (Tiger.sol#1081-1084)
setTaxRate(uint256) should be declared external:
        - Tiger.setTaxRate(uint256) (Tiger.sol#1086-1090)
setBurnTax(bool) should be declared external:
        - Tiger.setBurnTax(bool) (Tiger.sol#1092-1094)
includeAddress(address) should be declared external:
        - Tiger.includeAddress(address) (Tiger.sol#1102-1106)
includeToWhitelist(address) should be declared external:
        - Tiger.includeToWhitelist(address) (Tiger.sol#1113-1117)
excludeFromWhitlist(address) should be declared external:
        - Tiger.excludeFromWhitlist(address) (Tiger.sol#1119-1123)
mint(address,uint256) should be declared external:
        - Tiger.mint(address,uint256) (Tiger.sol#1131-1137)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#public-function-that-could-be-declared-external
INFO:Slither:Tiger.sol analyzed (11 contracts with 75 detectors), 74 result(s) found
INFO:Slither:Use https://crytic.io/ to get access to additional detectors and Github integration
```

Slither log >> UserVault.sol

```
INFO:Detectors:
UserVault.constructor(address,address,IERC20,IERC20,address).manager_ (UserVault.sol#697) lacks a zero-check on :
                - _manager = manager_ (UserVault.sol#698)
UserVault.constructor(address,address,IERC20,IERC20,address).user_ (UserVault.sol#697) lacks a zero-check on :
                - _user = user_ (UserVault.sol#699)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#missing-zero-address-validation
INFO:Detectors:
Address.verifyCallResult(bool,bytes,string) (UserVault.sol#267-287) uses assembly
        - INLINE ASM (UserVault.sol#279-282)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#assembly-usage
INFO:Detectors:
Address.functionCall(address,bytes) (UserVault.sol#151-153) is never used and should be removed
Address.functionCallWithValue(address,bytes,uint256) (UserVault.sol#180-186) is never used and should be removed
Address.functionDelegateCall(address,bytes) (UserVault.sol#240-242) is never used and should be removed
Address.functionDelegateCall(address,bytes,string) (UserVault.sol#250-259) is never used and should be removed
Address.functionStaticCall(address,bytes) (UserVault.sol#213-215) is never used and should be removed
Address.functionStaticCall(address,bytes,string) (UserVault.sol#223-232) is never used and should be removed
Address.sendValue(address,uint256) (UserVault.sol#126-131) is never used and should be removed
ContractGuard.checkSameOriginReentranted() (UserVault.sol#662-664) is never used and should be removed
ContractGuard.checkSameSenderReentranted() (UserVault.sol#666-668) is never used and should be removed
SafeERC20.safeApprove(IERC20,address,uint256) (UserVault.sol#604-617) is never used and should be removed
SafeERC20.safeDecreaseAllowance(IERC20,address,uint256) (UserVault.sol#628-639) is never used and should be removed
SafeERC20.safeIncreaseAllowance(IERC20,address,uint256) (UserVault.sol#619-626) is never used and should be removed
SafeERC20.safeTransferFrom(IERC20,address,address,uint256) (UserVault.sol#588-595) is never used and should be removed
SafeMath.add(uint256,uint256) (UserVault.sol#441-443) is never used and should be removed
SafeMath.div(uint256,uint256) (UserVault.sol#483-485) is never used and should be removed
SafeMath.div(uint256,uint256,string) (UserVault.sol#539-548) is never used and should be removed
SafeMath.mod(uint256,uint256) (UserVault.sol#499-501) is never used and should be removed
```

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

```
SafeERC20.safeDecreaseAllowance(IERC20,address,uint256) (UserVault.sol#628-639) is never used and should be removed
SafeERC20.safeIncreaseAllowance(IERC20,address,uint256) (UserVault.sol#619-626) is never used and should be removed
SafeERC20.safeTransferFrom(IERC20,address,address,uint256) (UserVault.sol#588-595) is never used and should be removed
SafeMath.add(uint256,uint256) (UserVault.sol#441-443) is never used and should be removed
SafeMath.div(uint256,uint256) (UserVault.sol#483-485) is never used and should be removed
SafeMath.div(uint256,uint256,string) (UserVault.sol#539-548) is never used and should be removed
SafeMath.mod(uint256,uint256) (UserVault.sol#499-501) is never used and should be removed
SafeMath.mod(uint256,uint256,string) (UserVault.sol#565-574) is never used and should be removed
SafeMath.mul(uint256,uint256) (UserVault.sol#469-471) is never used and should be removed
SafeMath.sub(uint256,uint256) (UserVault.sol#455-457) is never used and should be removed
SafeMath.sub(uint256,uint256,string) (UserVault.sol#516-525) is never used and should be removed
SafeMath.tryAdd(uint256,uint256) (UserVault.sol#370-376) is never used and should be removed
SafeMath.tryDiv(uint256,uint256) (UserVault.sol#412-417) is never used and should be removed
SafeMath.tryMod(uint256,uint256) (UserVault.sol#424-429) is never used and should be removed
SafeMath.tryMul(uint256,uint256) (UserVault.sol#395-405) is never used and should be removed
SafeMath.trySub(uint256,uint256) (UserVault.sol#383-388) is never used and should be removed
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#dead-code
INFO:Detectors:
Low level call in Address.sendValue(address,uint256) (UserVault.sol#126-131):
        - (success) = recipient.call{value: amount}() (UserVault.sol#129)
Low level call in Address.functionCallWithValue(address,bytes,uint256,string) (UserVault.sol#194-205):
        - (success,returndata) = target.call{value: value}(data) (UserVault.sol#203)
Low level call in Address.functionStaticCall(address,bytes,string) (UserVault.sol#223-232):
        - (success,returndata) = target.staticcall(data) (UserVault.sol#230)
Low level call in Address.functionDelegateCall(address,bytes,string) (UserVault.sol#250-259):
        - (success,returndata) = target.delegatecall(data) (UserVault.sol#257)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#low-level-calls
INFO:Detectors:
Parameter UserVault.stake(uint256)._amount (UserVault.sol#713) is not in mixedCase
Parameter UserVault.withdraw(uint256)._amount (UserVault.sol#717) is not in mixedCase
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#conformance-to-solidity-naming-conventions
INFO:Slither:UserVault.sol analyzed (9 contracts with 75 detectors), 37 result(s) found
INFO:Slither:Use https://crytic.io/ to get access to additional detectors and Github integration
```

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

Solidity Static Analysis

BBond.sol

Gas & Economy

Gas costs:

Gas requirement of function BBond.transferOwnership is infinite: If the gas requirement of a function is higher than the block gas limit, it cannot be executed. Please avoid loops in your functions or actions that modify large areas of storage (this includes clearing or copying arrays in storage)
Pos: 510:4:

Gas costs:

Gas requirement of function BBond.burnFrom is infinite: If the gas requirement of a function is higher than the block gas limit, it cannot be executed. Please avoid loops in your functions or actions that modify large areas of storage (this includes clearing or copying arrays in storage)
Pos: 960:4:

Miscellaneous

Constant/View/Pure functions:

SafeMath8.sub(uint8,uint8) : Is constant but potentially should not be. Note: Modifiers are currently not considered by this static analysis.
more
Pos: 39:4:

Constant/View/Pure functions:

BBond.burnFrom(address,uint256) : Potentially should be constant/view/pure but is not. Note: Modifiers are currently not considered by this static analysis.
more
Pos: 960:4:

Similar variable names:

BBond.burnFrom(address,uint256) : Variables have very similar names "account" and "amount". Note: Modifiers are currently not considered by this static analysis.
Pos: 961:32:

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: 861:12:

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

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: 427:19:

Bear.sol

Gas & Economy

Gas costs:

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

Gas costs:

Gas requirement of function Bear.transferFrom 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: 1144:4:

Gas costs:

Gas requirement of function ERC20.transferFrom 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: 1144:4:

Miscellaneous

Constant/View/Pure functions:

SafeMath8.sub(uint8,uint8) : Is constant but potentially should not be. Note: Modifiers are currently not considered by this static analysis.
more
Pos: 40:4:

Constant/View/Pure functions:

Bear.burnFrom(address,uint256) : Potentially should be constant/view/pure but is not. Note: Modifiers are currently not considered by this static analysis.
more
Pos: 1140:4:

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

Similar variable names:

Bear.burnFrom(address,uint256) : Variables have very similar names "account" and "amount". Note: Modifiers are currently not considered by this static analysis.
Pos: 1141:32:

No return:

Bear.setBurnThreshold(uint256): Defines a return type but never explicitly returns a value.
Pos: 1035:4:

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: 1084:8:

BearScrub.sol

Security

Transaction origin:

Use of tx.origin: "tx.origin" is useful only in very exceptional cases. If you use it for authentication, you usually want to replace it by "msg.sender", because otherwise any contract you call can act on your behalf.
more
Pos: 655:37:

Check-effects-interaction:

Potential violation of Checks-Effects-Interaction pattern in BearScrub.stake(address,uint256): Could potentially lead to re-entrancy vulnerability. Note: Modifiers are currently not considered by this static analysis.
more
Pos: 841:4:

Gas & Economy

Gas costs:

Gas requirement of function BearScrub.stake 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: 634:4:

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

Gas costs:

Gas requirement of function BearScrub.canWithdraw 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: 806:4:

Gas costs:

Gas requirement of function BearScrub.allocateSeigniorage 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: 871:4:

Miscellaneous

Constant/View/Pure functions:

Address.functionStaticCall(address,bytes) : Is constant but potentially should not be. Note: Modifiers are currently not considered by this static analysis.
more
Pos: 170:4:

Constant/View/Pure functions:

BearScrub.governanceRecoverUnsupported(contract IERC20,uint256,address) : Potentially should be constant/view/pure but is not. Note: Modifiers are currently not considered by this static analysis.
more
Pos: 893:4:

Similar variable names:

BearScrub.earned(address) : Variables have very similar names "mason" and "masons". Note: Modifiers are currently not considered by this static analysis.
Pos: 836:83:

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: 863:12:

Guard conditions:

Use "assert(x)" if you never ever want x to be false, not in any circumstance (apart from a bug in your code). Use "require(x)" if x can be false, due to e.g. invalid input or a failing external component.
more
Pos: 872:8:

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

BearTreasury.sol

Security

Transaction origin:

Use of tx.origin: "tx.origin" is useful only in very exceptional cases. If you use it for authentication, you usually want to replace it by "msg.sender", because otherwise any contract you call can act on your behalf.
more
Pos: 788:37:

Check-effects-interaction:

Potential violation of Checks-Effects-Interaction pattern in BearTreasury.redeemBonds(uint256,uint256): Could potentially lead to re-entrancy vulnerability. Note: Modifiers are currently not considered by this static analysis.
more
Pos: 1213:4:

Block timestamp:

Use of "now": "now" does not mean current time. "now" is an alias for "block.timestamp". "block.timestamp" can be influenced by miners to a certain degree, be careful.
more
Pos: 1253:31:

Gas & Economy

Gas costs:

Gas requirement of function Operator.transferOwnership 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: 734:4:

Gas costs:

Gas requirement of function BearTreasury.excludeFromTotalSupply 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: 1316:4:

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: 1317:8:

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

Miscellaneous

Constant/View/Pure functions:

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

more

Pos: 235:4:

Constant/View/Pure functions:

BearTreasury.governanceRecoverUnsupported(contract IERC20,uint256,address) : Potentially should be constant/view/pure but is not. Note: Modifiers are currently not considered by this static analysis.

more

Pos: 1331:4:

Similar variable names:

BearTreasury.setExtraFunds(address,uint256,address,uint256) : Variables have very similar names "_daoFund" and "_devFund". Note: Modifiers are currently not considered by this static analysis.

Pos: 1131:16:

Similar variable names:

BearTreasury._sendToScrub(uint256) : Variables have very similar names "daoFund" and "devFund". Note: Modifiers are currently not considered by this static analysis.

Pos: 1252:34:

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: 1117:8:

LionOracle.sol

Security

Check-effects-interaction:

INTERNAL ERROR in module Check-effects-interaction: IdentNode is undefined

Pos: not available

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: 583:20:

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

Email: audit@EtherAuthority.io

Gas & Economy

Gas costs:

Gas requirement of function Epoch.transferOwnership 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: 491:4:

Gas costs:

Gas requirement of function Oracle.consult 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: 684:4:

Gas costs:

Gas requirement of function Oracle.twap 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: 693:4:

ERC

ERC20:

ERC20 contract's "decimals" function should have "uint8" as return type
more
Pos: 17:4:

Miscellaneous

Constant/View/Pure functions:

UniswapV2OracleLibrary.currentCumulativePrices(address) : Is constant but potentially should not be. Note: Modifiers are currently not considered by this static analysis.
more
Pos: 196:4:

Constant/View/Pure functions:

Oracle.twap(address,uint256) : Is constant but potentially should not be. Note: Modifiers are currently not considered by this static analysis.
more
Pos: 693:4:

Similar variable names:

Oracle.twap(address,uint256) : Variables have very similar names "price0Cumulative" and "price1Cumulative". Note: Modifiers are currently not considered by this static analysis.
Pos: 699:55:

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

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: 688:12:

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: 697:54:

RewardManager.sol

Security

Transaction origin:

Use of tx.origin: "tx.origin" is useful only in very exceptional cases. If you use it for authentication, you usually want to replace it by "msg.sender", because otherwise any contract you call can act on your behalf.
more
Pos: 664:37:

Check-effects-interaction:

Potential violation of Checks-Effects-Interaction pattern in UserVault.exit(): Could potentially lead to re-entrancy vulnerability. Note: Modifiers are currently not considered by this static analysis.
more
Pos: 731:4:

Gas & Economy

Gas costs:

Gas requirement of function UserVault.withdrawAll 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: 712:4:

Gas costs:

Gas requirement of function UserVault.exit 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: 782:4:

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

Miscellaneous

Constant/View/Pure functions:

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

more

Pos: 213:4:

Constant/View/Pure functions:

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

more

Pos: 766:4:

Similar variable names:

RewardManager.(address,address,address,address) : Variables have very similar names "_tiger" and "tiger_". Note: Modifiers are currently not considered by this static analysis.

Pos: 754:24:

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: 774:8:

Tiger.sol

Gas & Economy

Gas costs:

Gas requirement of function Operator.transferOwnership 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: 513:4:

Gas costs:

Gas requirement of function Tiger.includeToWhitelist 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: 1113:4:

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

Email: audit@EtherAuthority.io

Miscellaneous

Constant/View/Pure functions:

SafeMath8.sub(uint8,uint8) : Is constant but potentially should not be. Note: Modifiers are currently not considered by this static analysis.
more
Pos: 39:4:

Constant/View/Pure functions:

Tiger._transfer(address,address,uint256) : Potentially should be constant/view/pure but is not. Note: Modifiers are currently not considered by this static analysis.
more
Pos: 1196:4:

Similar variable names:

Tiger.burnFrom(address,uint256) : Variables have very similar names "account" and "amount". Note: Modifiers are currently not considered by this static analysis.
Pos: 1144:32:

No return:

Tiger.setBurnThreshold(uint256): Defines a return type but never explicitly returns a value.
Pos: 1038:4:

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: 1097:8:

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: 430:19:

UserVault.sol

Security

Transaction origin:

Use of tx.origin: "tx.origin" is useful only in very exceptional cases. If you use it for authentication, you usually want to replace it by "msg.sender", because otherwise any contract you call can act on your behalf.
more
Pos: 663:37:

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

Check-effects-interaction:

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

more

Pos: 717:4:

Gas & Economy

Gas costs:

Gas requirement of function UserVault.withdrawAll 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: 708:4:

Gas costs:

Gas requirement of function UserVault.exit 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: 727:4:

Miscellaneous

Constant/View/Pure functions:

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

more

Pos: 647:4:

Similar variable names:

UserVault.(address,address,contract IERC20,contract IERC20,address) : Variables have very similar names "_tiger" and "tiger_". Note: Modifiers are currently not considered by this static analysis.

Pos: 704:8:

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: 693:8:

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

Email: audit@EtherAuthority.io

SaleBatch.sol

Security

Check-effects-interaction:

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

more

Pos: 200:4:

Block timestamp:

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

more

Pos: 662:16:

Gas & Economy

Gas costs:

Gas requirement of function SaleBatch.transferOwnership 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: 431:4:

Miscellaneous

Constant/View/Pure functions:

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

more

Pos: 219:4:

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: 664:8:

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: 700:25:

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

Email: audit@EtherAuthority.io

Zap.sol

Security

Check-effects-interaction:

INTERNAL ERROR in module Check-effects-interaction: IdentNode is undefined
Pos: not available

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: 1258:92:

Gas & Economy

Gas costs:

Gas requirement of function Zap.zapInToken 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: 961:4:

ERC

ERC20:

ERC20 contract's "decimals" function should have "uint8" as return type
more
Pos: 121:4:

Miscellaneous

Similar variable names:

Zap.zapOut(address,uint256,address,address) : Variables have very similar names "token" and "token1". Note: Modifiers are currently not considered by this static analysis.
Pos: 1045:33:

No return:

IUniswapV2Router01.swapExactTokensForETH(uint256,uint256,address[],address,uint256): Defines a return type but never explicitly returns a value.
Pos: 247:4:

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: 839:12:

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: 716:19:

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

Solhint Linter

BBond.sol

```
BBond.sol:2:1: Error: Compiler version 0.6.12 does not satisfy the r
semver requirement
BBond.sol:886:24: Error: Code contains empty blocks
BBond.sol:906:24: Error: Code contains empty blocks
BBond.sol:940:55: Error: Code contains empty blocks
```

Bear.sol

```
Bear.sol:2:1: Error: Compiler version 0.6.12 does not satisfy the r
semver requirement
Bear.sol:888:24: Error: Code contains empty blocks
Bear.sol:908:24: Error: Code contains empty blocks
Bear.sol:1105:5: Error: Function name must be in mixedCase
```

BearScrub.sol

```
BearScrub.sol:2:1: Error: Compiler version 0.6.12 does not satisfy
the r semver requirement
BearScrub.sol:59:71: Error: Code contains empty blocks
BearScrub.sol:86:28: Error: Avoid using low level calls.
BearScrub.sol:160:51: Error: Avoid using low level calls.
BearScrub.sol:214:51: Error: Avoid using low level calls.
BearScrub.sol:236:17: Error: Avoid using inline assembly. It is
acceptable only in rare cases
BearScrub.sol:655:38: Error: Avoid to use tx.origin
BearScrub.sol:668:31: Error: Avoid to use tx.origin
BearScrub.sol:856:52: Error: Visibility modifier must be first in
list of modifiers
BearScrub.sol:860:59: Error: Visibility modifier must be first in
list of modifiers
```

BearTreasury.sol

```
BearTreasury.sol:2:1: Error: Compiler version 0.6.12 does not satisfy
the r semver requirement
BearTreasury.sol:124:71: Error: Code contains empty blocks
BearTreasury.sol:151:28: Error: Avoid using low level calls.
BearTreasury.sol:225:51: Error: Avoid using low level calls.
BearTreasury.sol:279:51: Error: Avoid using low level calls.
BearTreasury.sol:301:17: Error: Avoid using inline assembly. It is
acceptable only in rare cases
```

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

```
BearTreasury.sol:788:38: Error: Avoid to use tx.origin
BearTreasury.sol:801:31: Error: Avoid to use tx.origin
BearTreasury.sol:806:1: Error: Contract has 33 states declarations
but allowed no more than 15
BearTreasury.sol:891:17: Error: Avoid to make time-based decisions in
your business logic
BearTreasury.sol:897:17: Error: Avoid to make time-based decisions in
your business logic
BearTreasury.sol:1171:42: Error: Code contains empty blocks
BearTreasury.sol:1171:51: Error: Code contains empty blocks
BearTreasury.sol:1246:32: Error: Avoid to make time-based decisions
in your business logic
BearTreasury.sol:1251:13: Error: Possible reentrancy vulnerabilities.
Avoid state changes after transfer.
BearTreasury.sol:1253:32: Error: Avoid to make time-based decisions
in your business logic
BearTreasury.sol:1261:26: Error: Avoid to make time-based decisions
in your business logic
BearTreasury.sol:1310:41: Error: Avoid to make time-based decisions
in your business logic
```

LionOracle.sol

```
LionOracle.sol:2:1: Error: Compiler version 0.6.12 does not satisfy
the r semver requirement
LionOracle.sol:35:5: Error: Function name must be in mixedCase
LionOracle.sol:37:5: Error: Function name must be in mixedCase
LionOracle.sol:56:5: Error: Function name must be in mixedCase
LionOracle.sol:118:5: Error: Contract name must be in CamelCase
LionOracle.sol:124:5: Error: Contract name must be in CamelCase
LionOracle.sol:192:23: Error: Avoid to make time-based decisions in
your business logic
LionOracle.sol:566:17: Error: Avoid to make time-based decisions in
your business logic
LionOracle.sol:566:47: Error: Use double quotes for string literals
LionOracle.sol:573:13: Error: Avoid to make time-based decisions in
your business logic
LionOracle.sol:574:47: Error: Use double quotes for string literals
LionOracle.sol:583:21: Error: Avoid to make time-based decisions in
your business logic
LionOracle.sol:613:60: Error: Use double quotes for string literals
```

RewardManager.sol

```
RewardManager.sol:2:1: Error: Compiler version 0.6.12 does not
satisfy the r semver requirement
RewardManager.sol:102:71: Error: Code contains empty blocks
RewardManager.sol:129:28: Error: Avoid using low level calls.
RewardManager.sol:203:51: Error: Avoid using low level calls.
RewardManager.sol:257:51: Error: Avoid using low level calls.
RewardManager.sol:279:17: Error: Avoid using inline assembly. It is
```

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

```
acceptable only in rare cases
RewardManager.sol:664:38: Error: Avoid to use tx.origin
RewardManager.sol:677:31: Error: Avoid to use tx.origin
RewardManager.sol:766:49: Error: Code contains empty blocks
RewardManager.sol:771:13: Error: Variable name must be in mixedCase
```

Tiger.sol

```
Tiger.sol:2:1: Error: Compiler version 0.6.12 does not satisfy the r
semver requirement
Tiger.sol:890:24: Error: Code contains empty blocks
Tiger.sol:910:24: Error: Code contains empty blocks
Tiger.sol:1108:5: Error: Function name must be in mixedCase
```

UserVault.sol

```
UserVault.sol:2:1: Error: Compiler version 0.6.12 does not satisfy
the r semver requirement
UserVault.sol:102:71: Error: Code contains empty blocks
UserVault.sol:129:28: Error: Avoid using low level calls.
UserVault.sol:203:51: Error: Avoid using low level calls.
UserVault.sol:257:51: Error: Avoid using low level calls.
UserVault.sol:279:17: Error: Avoid using inline assembly. It is
acceptable only in rare cases
UserVault.sol:663:38: Error: Avoid to use tx.origin
UserVault.sol:676:31: Error: Avoid to use tx.origin
```

SaleBatch.sol

```
SaleBatch.sol:2:1: Error: Compiler version 0.6.12 does not satisfy
the r semver requirement
SaleBatch.sol:108:71: Error: Code contains empty blocks
SaleBatch.sol:135:28: Error: Avoid using low level calls.
SaleBatch.sol:209:51: Error: Avoid using low level calls.
SaleBatch.sol:263:51: Error: Avoid using low level calls.
SaleBatch.sol:285:17: Error: Avoid using inline assembly. It is
acceptable only in rare cases
SaleBatch.sol:572:30: Error: Avoid to make time-based decisions in
your business logic
SaleBatch.sol:589:15: Error: Avoid to make time-based decisions in
your business logic
SaleBatch.sol:598:42: Error: Avoid to make time-based decisions in
your business logic
SaleBatch.sol:616:17: Error: Avoid to make time-based decisions in
your business logic
SaleBatch.sol:616:49: Error: Avoid to make time-based decisions in
your business logic
SaleBatch.sol:642:57: Error: Variable "_data" is unused
SaleBatch.sol:649:17: Error: Avoid to make time-based decisions in
```

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

```
your business logic
SaleBatch.sol:662:17: Error: Avoid to make time-based decisions in
your business logic
```

Zap.sol

```
Zap.sol:545:18: Error: Parse error: missing ';' at '{'
Zap.sol:558:18: Error: Parse error: missing ';' at '{'
Zap.sol:570:18: Error: Parse error: missing ';' at '{'
Zap.sol:587:18: Error: Parse error: missing ';' at '{'
Zap.sol:599:18: Error: Parse error: missing ';' at '{'
Zap.sol:691:18: Error: Parse error: missing ';' at '{'
Zap.sol:714:18: Error: Parse error: missing ';' at '{'
Zap.sol:736:18: Error: Parse error: missing ';' at '{'
Zap.sol:817:18: Error: Parse error: missing ';' at '{'
```

Software analysis result:

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

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

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

Email: audit@EtherAuthority.io