Ether Authority

www.EtherAuthority.io
audit@etherauthority.io

SMART CONTRACT

Security Audit Report

Project:	10mb Finance
Website:	https://10mb.finance/
Platform:	Cronos
Language:	Solidity
Date:	June 10th, 2022

Table of contents

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 10 MB Finance to perform the Security audit of the 10MB 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 June 10th, 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

10MB Finance Contracts have functions like initialize, earn, stake, withdraw, exit, update, consult, twap, info, mint, redeem, migrate, addPool, removePool, add, set, deposit, withdraw, burn, poolMint, etc. The 10MB Finance contract inherits the IERC20, SafeMath, SafeERC20, ERC20Burnable, Ownable, Math, ERC20, ReentrancyGuard 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 10MB Finance Protocol Smart Contracts**
Platform	**Cronos / Solidity**
File 1	Boardroom.sol
File 1 MD5 Hash	D65AD533CD37C4E505F7A0C40E846A06
File 2	ContractGuard.sol
File 2 MD5 Hash	21C8361B2382D1F1705DDD3BD7B8D0D9
File 3	Oracle.sol
File 3 MD5 Hash	B693D8B00872B37219A55C0B933B5A8C
File 4	Pool.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

File 4 MD5 Hash	8A2A2F5EFA73B5BA37A3B54B8E3461D2
Updated File 4 MD5 Hash	C1D6633D751337297F41BC4C4288E388
File 5	.TaxOffice.sol
File 5 MD5 Hash	0F4FDDE2843E4B660C425491A7E1F561
File 6	TaxOracle.sol
File 6 MD5 Hash	A81A5F2A251D2295F67212271ADDB9A3
File 7	Treasury.sol
File 7 MD5 Hash	938142A6DBDBF8D9FA9B85593A522B67
Updated File 7 MD5 Hash	0B27BBBAF2727610F18A4390584A4888
File 8	_10MBMasterchef.sol
File 8 MD5 Hash	AEF63DB86B56A90C86A0CD400B3FC9E7
Updated File 8 MD5 Hash	423DD8F46DA04963E5B31F0BFC5DD97D
File 9	MintableERC20.sol
File 9 MD5 Hash	D5130643EA95880BC89BF32B499928D3
File 10	_10BOND.sol
File 10 MD5 Hash	A9628540667763DEAB4C73B3FD6FFE06
File 11	_10MB.sol
File 11 MD5 Hash	BF8F409D758A97798F47977E293C8BC1
Updated File 11 MD5 Hash	9D522ABEF30C5A12D9B715DE724EF48E
File 12	_10SHARE.sol
File 12 MD5 Hash	F9B1491D8AD9341CB56B5F7A0BA04054
Updated File 12 MD5 Hash	D18CED807ACFC01098B62A5E6B863441
File 13	Timelock.sol
File 13 MD5 Hash	BF8F409D758A97798F47977E293C8BC1
Updated File 13 MD5 Hash	3CF931F9CF703918D10DC42A909C1835
File 14	ERCCRO.sol
File 14 MD5 Hash	1D97DDDE92AB34B3A32D216ED7FD5450

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 15	.ERCDAI.sol
File 15 MD5 Hash	9E2C18838A7680849C9A5CF02FC32BB1
File 16	ERCMMF.sol
File 16 MD5 Hash	88C33A1D6F6F7EF553012963A49FA020
File 17	.ERCUSDT.sol
File 17 MD5 Hash	427E35D9353FA0C95BBF86A7E0A1C379
File 18	ERCWSMINO.sol
File 18 MD5 Hash	BC51A7D1ABB394C0DC94DC40CD9C193A
File 19	.ShareWrapper.sol
File 19 MD5 Hash	7C8BE2B74A0CBEE3DA04A099B989CAF6
Audit Date	June 10th,2022
Revise Audit Date	June 20th,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 Boardroom.sol ● Lock for 6 epochs (48h) before release withdrawal. ● Lock for 3 epochs (24h) before release claimReward.	**YES, This is valid.**
File 2 ContractGuard.sol ● ContractGuard has functions like: checkSameSenderReentranted, etc.	**YES, This is valid.**
File 3 Oracle.sol ● Oracle has functions like: update, consult, etc.	**YES, This is valid.**
File 4 Pool.sol ● Twap Price Scaling: 98% ● Redemption Delay: 1	**YES, This is valid.**
File 5 TaxOffice.sol ● TaxOffice has functions like: setTaxTiersTwap, setTaxTiersRate,etc.	**YES, This is valid.** **Owner wallet's private key must be handled very securely. Because if that is compromised, then it will create problems.**
File 6 Timelock.sol ● Grace Period: 14 Days ● Minimum Delay: 6 hours ● Maximum Delay: 30 Days	**YES, This is valid.**
File 7 TaxOracle.sol ● TaxOracle has functions like: consult, set10MB, 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

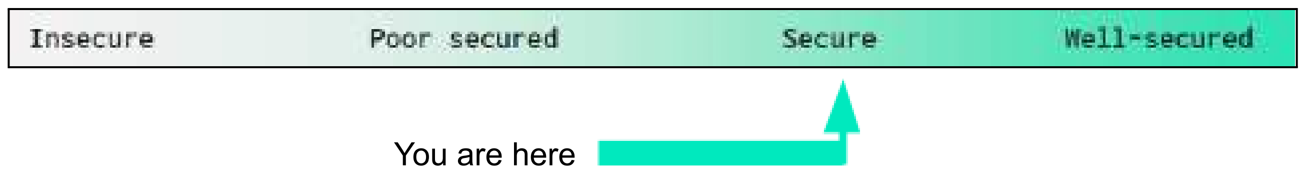
File 8 Treasury.sol ● Period: 8 hours ● Maximum Supply Expansion: 1.44% ● Minimum max of 0.1% supply for expansion ● Boardroom Withdraw Fee: 2% ● Bond supply for depletion floor: 100% ● Seigniorage Expansion Floo: 35% ● Maximum Supply Contraction: 3% ● Maximum Debt Ratio: 35% ● Premium Threshold: 101 ● Premium Percent: 5000 ● Allocate Seigniorage Salary: 0.2 ● Bootstrap Epochs: 24 ● Bootstrap Supply Expansion Percent: 190 ● Ratio Step: 0.25% ● Target Collateral Ratio: 100% ● Effective Collateral Ratio: 100% ● Refresh Cooldown: 1 hour ● Price Target: $0.1 ● Price Band: 500 ● Redemption Fee: 4000 ● Minting Fee: 4000	**YES, This is valid.**
File 9 _10MBMasterchef.sol ● _10MBMasterchef has functions like: set, add, poolLength, etc.	**YES, This is valid.**
File 10 MintableERC20.sol ● MintableERC20 has functions like: mint, burn, etc.	**YES, This is valid.**
File 11 _10BOND.sol ● Name: 10BOND ● Symbol: 10BOND	**YES, This is valid.**
File 12 _10MB.sol	**YES, This is valid.**

● Name: 10MB ● Symbol: 10MB	
File 13 _10SHARE.sol ● Name: 10SHARE ● Symbol: 10SHARE	**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, 1 medium and 4 low and some very low level issues.

All the issues have been resolved / acknowledged in the revised code.

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

Overall Audit Result: PASSED

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 19 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 10MB 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 10MB Finance Protocol.

The 10MB Finance team has not provided unit test scripts, which would have helped to determine the integrity of the code in an automated way.

Some code parts are well commented on smart contracts. We suggest using Ethereum's NatSpec style for the commenting.

Documentation

We were given a 10MB 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 well commented. So it is easy to quickly understand the programming flow as well as complex code logic. Comments are very helpful in understanding the overall architecture of the protocol.

Another source of information was its official website https://10mb.finance/ which provided rich information about the project architecture.

Use of Dependencies

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

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

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

Boardroom.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	checkSameOriginReentranted	internal	Passed	No Issue
8	checkSameSenderReentranted	internal	Passed	No Issue
9	onlyOneBlock	modifier	Passed	No Issue
10	onlyOperator	modifier	Passed	No Issue
11	directorExists	modifier	Passed	No Issue
12	updateReward	modifier	Passed	No Issue
13	notInitialized	modifier	Passed	No Issue
14	initialize	write	Passed	No Issue
15	amIOperator	read	Passed	No Issue
16	setOperator	write	access only Owner	No Issue
17	setLockUp	external	access only Operator	No Issue
18	setReserveFund	external	access only Operator	No Issue
19	setStakeFee	external	access only Operator	No Issue
20	setWithdrawFee	external	access only Operator	No Issue
21	totalSupply	read	Passed	No Issue
22	balanceOf	read	Passed	No Issue
23	latestSnapshotIndex	read	Passed	No Issue
24	getLatestSnapshot	internal	Passed	No Issue
25	getLastSnapshotIndexOf	read	Passed	No Issue
26	getLastSnapshotOf	internal	Passed	No Issue
27	canWithdraw	external	Passed	No Issue
28	canClaimReward	external	Passed	No Issue
29	epoch	external	Passed	No Issue
30	nextEpochPoint	external	Passed	No Issue
31	get10MBPrice	external	Passed	No Issue
32	rewardPerShare	read	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

33	earned	read	Passed	No Issue
34	stake	write	access only One Block	No Issue
35	withdraw	write	access only One Block	No Issue
36	exit	external	Passed	No Issue
37	claimReward	write	Passed	No Issue
38	allocateSeigniorage	external	access only Operator	No Issue
39	governanceRecoverUnsupported	external	access only Operator	No Issue

ContractGuard.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

Oracle.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	Passed	No Issue
12	consult	external	Passed	No Issue
13	twap	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

Pool.sol

Functions

Sl.	Functions	Type	Observation	Conclusion
1	constructor	write	Passed	No Issue
2	nonReentrant	modifier	Passed	No Issue
3	onlyOperator	modifier	Passed	No Issue
4	notMigrated	modifier	Passed	No Issue
5	onlyTreasury	modifier	Passed	No Issue
6	collateral10MBBalance	external	Passed	No Issue
7	info	external	Passed	No Issue
8	getCollateralPrice	read	Passed	No Issue
9	getCollateralToken	external	Passed	No Issue
10	netSupplyMinted	external	Passed	No Issue
11	mint	external	Passed	No Issue
12	redeem	external	Passed	No Issue
13	collectRedemption	external	Passed	No Issue
14	migrate	external	access only Operator	No Issue
15	toggleMinting	external	access only Operator	No Issue
16	toggleRedeeming	external	access only Operator	No Issue
17	setPoolCeiling	external	access only Operator	No Issue
18	setTwapPriceScalingPercentage	external	access only Operator	No Issue
19	setRedemptionDelay	external	access only Operator	No Issue
20	setTreasury	external	access only Operator,	No Issue
21	transferCollateralToTreasury	external	access only Treasury	No Issue
22	transferCollateralToOperator	external	access only Operator	No Issue

TaxOffice.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

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

6	_transferOperator	internal	Passed	No Issue
7	setTaxTiersTwap	write	access only Owner	No Issue
8	setTaxTiersRate	write	access only Operator	No Issue
9	enableAutoCalculateTax	write	access only Operator	No Issue
10	disableAutoCalculateTax	write	access only Operator	No Issue
11	setTaxRate	write	access only Operator	No Issue
12	setBurnThreshold	write	access only Operator	No Issue
13	setTaxCollectorAddress	write	access only Operator	No Issue
14	excludeAddressFromTax	external	access only Operator	No Issue
15	_excludeAddressFromTax	write	Passed	No Issue
16	includeAddressInTax	external	access only Operator	No Issue
17	_includeAddressInTax	write	Passed	No Issue
18	taxRate	external	Passed	No Issue
19	addLiquidityTaxFree	external	Passed	No Issue
20	addLiquidityETHTaxFree	external	Passed	No Issue
21	setTaxable10MBOracle	external	access only Operator	No Issue
22	transferTaxOffice	external	access only Operator	No Issue
23	taxFreeTransferFrom	external	Passed	No Issue
24	setTaxExclusionForAddres s	external	access only Operator	No Issue
25	_approveTokenIfNeeded	write	Passed	No Issue

Timelock.sol

Functions

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

TaxOracle.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	consult	external	Passed	No Issue
8	set10MB	external	access only Owner	No Issue
9	setUsdt	external	access only Owner	No Issue
10	setPair	external	access only Owner	No Issue

Treasury.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
9	notInitialized	modifier	Passed	No Issue
10	isInitialized	read	Passed	No Issue
11	nextEpochPoint	read	Passed	No Issue
12	info	external	Passed	No Issue
13	globalCollateralValue	read	Passed	No Issue
14	globalIronSupply	read	Passed	No Issue
15	calcEffectiveCollateralRatio	read	Passed	No Issue
16	refreshCollateralRatio	external	Passed	No Issue
17	calcCollateralBalance	read	Passed	No Issue
18	get10SHAREPrice	read	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

19	get10MBPrice	read	Passed	No Issue
20	get10MBUpdatedPrice	read	Passed	No Issue
21	getReserve	read	Passed	No Issue
22	getBurnable10MBLeft	read	Passed	No Issue
23	getRedeemableBonds	read	Passed	No Issue
24	getBondDiscountRate	read	Passed	No Issue
25	getBondPremiumRate	read	Passed	No Issue
26	initialize	write	access only Operator	No Issue
27	setOperator	external	access only Operator	No Issue
28	setBoardroom	write	Passed	No Issue
29	setBoardroomWithdrawFee	external	access only Operator	No Issue
30	setBoardroomStakeFee	external	access only Operator	No Issue
31	set10MBOracle	external	access only Operator	No Issue
32	set10MBPriceCeiling	external	access only Operator	No Issue
33	setMinMaxSupplyExpansionPercent	external	access only Operator	No Issue
34	setMaxSupplyExpansionPercent	external	access only Operator	No Issue
35	setBondDepletionFloorPercent	external	access only Operator	No Issue
36	setMaxSupplyContractionPercent	external	access only Operator	No Issue
37	setMaxDebtRatioPercent	external	access only Operator	No Issue
38	setBootstrap	external	access only Operator	No Issue
39	setExtraFunds	external	access only Operator	No Issue
40	setAllocateSeigniorageSalary	external	access only Operator	No Issue
41	setMaxDiscountRate	external	access only Operator	No Issue
42	setMaxPremiumRate	external	access only Operator	No Issue
43	setDiscountPercent	external	access only Operator	No Issue
44	setPremiumThreshold	external	access only Operator	No Issue
45	setPremiumPercent	external	access only Operator	No Issue
46	setMintingFactorForPayingDebt	external	access only Operator	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

47	set10MBSupplyTarget	external	access only Operator	No Issue
48	addPool	write	access only Operator	No Issue
49	removePool	write	access only Operator	No Issue
50	_update10MBPrice	internal	Passed	No Issue
51	_update10SHAREPrice	internal	Passed	No Issue
52	get10MBCirculatingSupply	read	Passed	No Issue
53	buyBonds	external	access only One Block	No Issue
54	redeemBonds	external	access only One Block	No Issue
55	_sendToBoardroom	internal	Passed	No Issue
56	_calculateMaxSupplyExpansionPercent	internal	Passed	No Issue
57	get10MBExpansionRate	read	Passed	No Issue
58	get10MBExpansionAmount	external	Passed	No Issue
59	allocateSeigniorage	external	access only One Block	No Issue
60	treasuryUpdates	external	Passed	No Issue
61	governanceRecoverUnsupported	external	access only Operator	No Issue
62	boardroomSetOperator	external	access only Operator	No Issue
63	boardroomSetReserveFund	external	access only Operator	No Issue
64	boardroomSetLockUp	external	access only Operator	No Issue
65	boardroomAllocateSeigniorage	external	access only Operator	No Issue
66	boardroomGovernanceRecoverUnsupported	external	access only Operator	No Issue
67	hasPool	external	Passed	No Issue
68	setRedemptionFee	write	access only Operator	No Issue
69	setMintingFee	write	access only Operator	No Issue
70	setRatioStep	write	access only Operator	No Issue
71	setPriceTarget	write	access only Operator	No Issue
72	setRefreshCooldown	write	access only Operator	No Issue
73	setPriceBand	external	access only Operator	No Issue
74	toggleCollateralRatio	write	access only Operator	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

75	toggleEffectiveCollateralRatio	write	access only Operator	No Issue
76	executeTransaction	write	access only Operator	No Issue

_10MBMasterchef.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	onlyWhitelisted	modifier	Passed	No Issue
8	isWhitelist	read	Passed	No Issue
9	setWhitelist	external	access only Owner	No Issue
10	disableWhitelist	external	access only Owner	No Issue
11	onERC721Received	external	Passed	No Issue
12	nonDuplicated	modifier	Passed	No Issue
13	nonContract	modifier	Passed	No Issue
14	getNftIdBoosters	read	Passed	No Issue
15	getBoostRate10MB	read	Passed	No Issue
16	getBoostRate10SHARE	read	Passed	No Issue
17	getBoost10MB	read	Passed	No Issue
18	getBoost10SHARE	read	Passed	No Issue
19	getSlots	read	Passed	No Issue
20	getTokenIds	read	Passed	No Issue
21	poolLength	external	Passed	No Issue
22	getMultiplier	write	Passed	No Issue
23	pending10MB	external	Passed	No Issue
24	pending10SHARE	external	Passed	No Issue
25	add	write	access only Owner	No Issue
26	set	write	access only Owner	No Issue
27	depositNFT	write	Passed	No Issue
28	withdrawNFT	write	Passed	No Issue
29	massUpdatePools	write	Passed	No Issue
30	updatePool	write	Passed	No Issue
31	deposit	write	Passed	No Issue
32	withdraw	write	Passed	No Issue
33	emergencyWithdraw	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

34	safe10MBTransfer	internal	Passed	No Issue
35	safe10SHARETransfer	internal	Passed	No Issue
36	update10MBEmissionRate	write	access only Owner	No Issue
37	update10SHAREEmission Rate	write	access only Owner	No Issue
38	setNftBaseBoostRate	write	access only Owner	No Issue
39	setNftSpecificBoost	write	access only Owner	No Issue
40	setNftIdSpecificBoostRang e	write	access only Owner	No Issue
41	removeNftIdBoostRangeB yId	write	access only Owner	No Issue
42	setNftWhitelist	write	access only Owner	No Issue
43	setReserveFund	write	access only Owner	No Issue
44	flipWhitelistAll	write	access only Owner	No Issue
45	harvestAllRewards	write	Passed	No Issue

MintableERC20.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	name	read	Passed	No Issue
8	symbol	read	Passed	No Issue
9	decimals	read	Passed	No Issue
10	totalSupply	read	Passed	No Issue
11	balanceOf	read	Passed	No Issue
12	transfer	write	Passed	No Issue
13	allowance	read	Passed	No Issue
14	approve	write	Passed	No Issue
15	transferFrom	write	Passed	No Issue
16	increaseAllowance	write	Passed	No Issue
17	decreaseAllowance	write	Passed	No Issue
18	_transfer	internal	Passed	No Issue
19	_mint	internal	Passed	No Issue
20	_burn	internal	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

21	_approve	internal	Passed	No Issue
22	_spendAllowance	internal	Passed	No Issue
23	_beforeTokenTransfer	internal	Passed	No Issue
24	_afterTokenTransfer	internal	Passed	No Issue
25	mint	external	access only Owner	No Issue
26	burn	external	access only Owner	No Issue
27	decimals	read	Passed	No Issue

_10BOND.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	onlyOperator	modifier	Passed	No Issue
10	burn	write	Unlimited Burning	Refer audit findings
11	mint	write	Unlimited Minting	Refer audit findings
12	burnFrom	write	access only Operator	No Issue
13	setOperator	write	access only Owner	No Issue
14	amIOperator	read	Passed	No Issue

_10MB.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

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

7	burn	write	Passed	No Issue
8	burnFrom	write	Passed	No Issue
9	onlyPools	modifier	Passed	No Issue
10	onlyOperator	modifier	Passed	No Issue
11	onlyTaxOffice	modifier	Passed	No Issue
12	onlyOperatorOrTaxOffice	modifier	Passed	No Issue
13	getTaxTiersTwapsCount	read	Passed	No Issue
14	getTaxTiersRatesCount	read	Passed	No Issue
15	isAddressExcluded	read	Passed	No Issue
16	setTaxTiersTwap	write	Passed	No Issue
17	setTaxTiersRate	write	Passed	No Issue
18	setBurnThreshold	write	access only Tax Office	No Issue
19	_get10MBPrice	internal	Passed	No Issue
20	_updateTaxRate	internal	Passed	No Issue
21	enableAutoCalculateTax	write	access only Tax Office	No Issue
22	disableAutoCalculateTax	write	access only Tax Office	No Issue
23	setOperator	write	access only Owner	No Issue
24	set10MBOracle	write	access only Operator Or Tax Office	No Issue
25	setTaxOffice	write	access only Operator Or Tax Office	No Issue
26	setTaxCollectorAddress	write	access only Tax Office	No Issue
27	setTaxRate	write	access only Tax Office	No Issue
28	excludeAddress	write	access only Operator Or Tax Office	No Issue
29	includeAddress	write	access only Operator Or Tax Office	No Issue
30	mint	write	Unlimited Minting	Refer audit findings
31	burnFrom	write	access only Operator	No Issue
32	poolBurnFrom	external	access only Pools	No Issue
33	poolMint	external	Unlimited Minting	Refer audit findings
34	transferFrom	write	Passed	No Issue
35	_transferWithTax	internal	Passed	No Issue
36	amIOperator	read	Passed	No Issue
37	setTreasuryAddress	write	Passed	No Issue

_10SHARE.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	onlyPools	modifier	Passed	No Issue
10	onlyOperator	modifier	Passed	No Issue
11	setDaoFund	external	access only Operator	No Issue
12	setEquityFund	external	access only Operator	No Issue
13	setDevFund	external	access only Operator	No Issue
14	unclaimedDaoFund	read	Passed	No Issue
15	unclaimedDevFund	read	Passed	No Issue
16	unclaimedEquityFund	read	Passed	No Issue
17	claimRewards	external	Passed	No Issue
18	setOperator	write	access only Owner	No Issue
19	setFundMultiplier	external	access only Operator	No Issue
20	mint	write	Unlimited Minting	Refer audit findings
21	burn	write	Passed	No Issue
22	poolMint	external	Unlimited Minting	Refer audit findings
23	poolBurnFrom	external	access only Pools	No Issue
24	governanceRecoverUnsupported	external	Passed	No Issue
25	amIOperator	read	Passed	No Issue
26	setTreasuryAddress	write	access only Operator	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

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

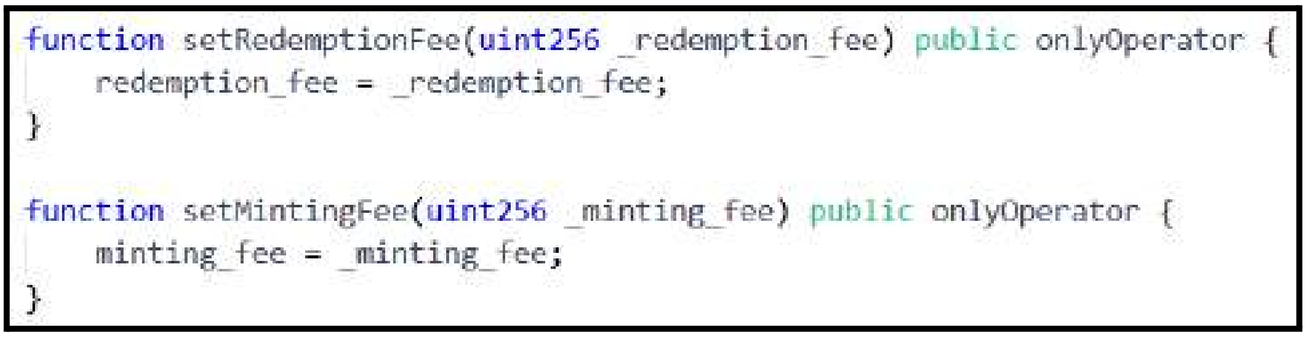
No Critical severity vulnerabilities were found.

High Severity

No High severity vulnerabilities were found.

Medium

(1) Fee limit is not set: **Treasury.sol**



```
function setRedemptionFee(uint256 _redemption_fee) public onlyOperator {
    redemption_fee = _redemption_fee;
}

function setMintingFee(uint256 _minting_fee) public onlyOperator {
    minting_fee = _minting_fee;
}
```

Operators can set the individual fees to any variable. This might deter investors as they could be wary that these fees might one day be set to 100% to force transfers to go to the contract owner.

Resolution: Consider adding an explicit cap to the total fee on every fee adjustment function.

Status: Fixed

Low

(1) Critical operation lacks event log

Missing event log for:

_10SHARE.sol

- claimRewards

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

_10MB.sol

- setTaxTiersTwap
- setTaxTiersRate

Resolution: Write an event log for listed events.

Status: Fixed

(2) Function input parameters lack of check:

Variable validation is not performed in below functions:

_10SHARE.sol

- setTreasuryAddress = _treasury
- governanceRecoverUnsupported = _token , to

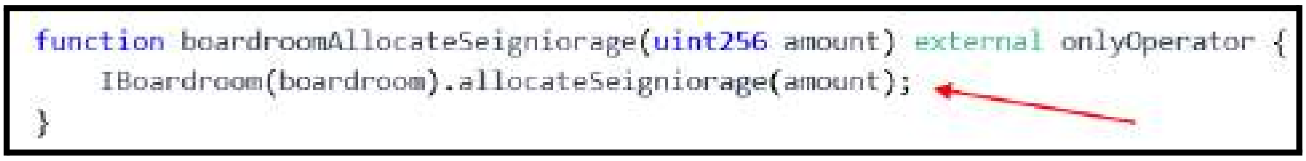
_10MB.sol

- setTaxTiersRate = _value
- setTreasuryAddress = _treasury

Resolution: We advise to put validation : int type variables should not be empty and > 0 & address type variables should not be address(0).

Status: Fixed

(3) Insufficient allowance: **Treasury.sol**



```
function boardroomAllocateSeigniorage(uint256 amount) external onlyOperator {
    IBoardroom(boardroom).allocateSeigniorage(amount);
}
```

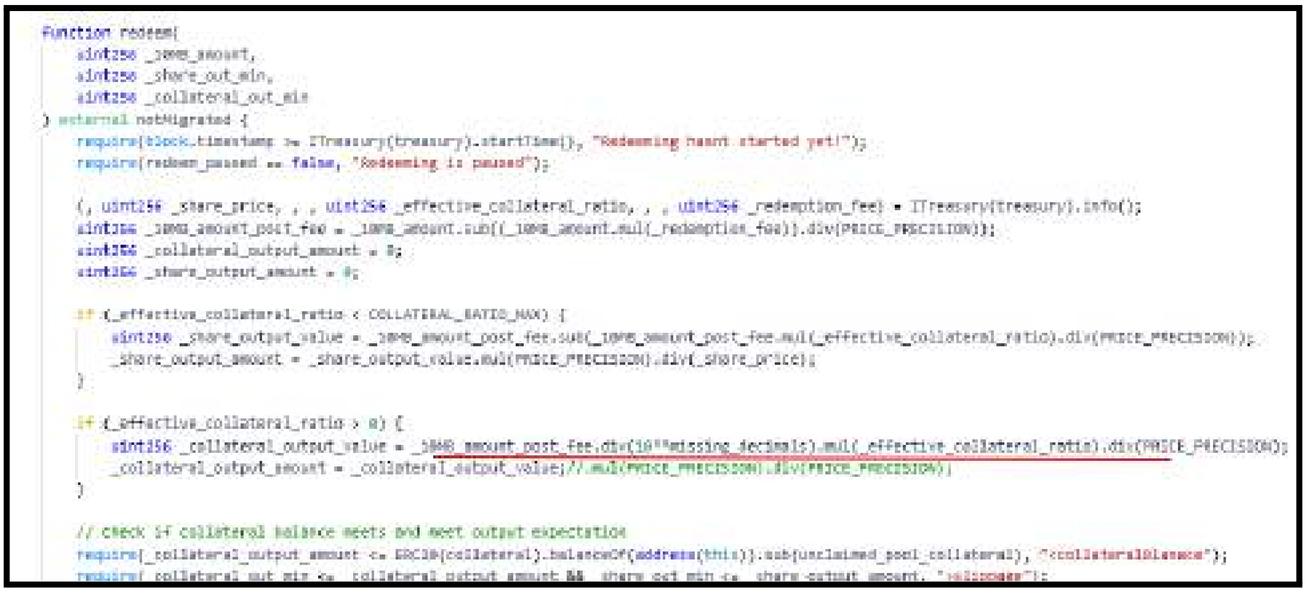
SafeApproval is missing in boardroomAllocateSeigniorage function, which throws insufficient allowance error. Treasury's boardroomAllocateSeigniorage function calls boardroom's allocateSeigniorage function. Here treasury becomes msg.sender which requires approval.

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

Resolution: We suggest adding the below line in the boardroomAllocateSeigniorage function just before allocateSeigniorage call. IERC20(_10MB).safeApprove(boardroom, amount);

Status: Fixed

(4) Division before multiplication: **Pool.sol**



```
function redeem(
    uint256 _10MB_amount,
    uint256 _share_out_min,
    uint256 _collateral_out_min
) external notMigrated {
    require(block.timestamp >= ITreasury(treasury).startTime(), "Redeeming hasnt started yet!");
    require(redeem_paused == false, "Redeeming is paused");

    (, uint256 _share_price, , , uint256 _effective_collateral_ratio, , , uint256 _redemption_fee) = ITreasury(treasury).info();
    uint256 _10MB_amount_post_fee = _10MB_amount.sub((_10MB_amount.mul(_redemption_fee)).div(PRICE_PRECISION));
    uint256 _collateral_output_amount = 0;
    uint256 _share_output_amount = 0;

    if (_effective_collateral_ratio < COLLATERAL_RATIO_MAX) {
        uint256 _share_output_value = _10MB_amount_post_fee.sub(_10MB_amount_post_fee.mul(_effective_collateral_ratio).div(PRICE_PRECISION));
        _share_output_amount = _share_output_value.mul(PRICE_PRECISION).div(_share_price);
    }

    if (_effective_collateral_ratio > 0) {
        uint256 _collateral_output_value = _10MB_amount_post_fee.div(10**missing_decimals).mul(_effective_collateral_ratio).div(PRICE_PRECISION);
        _collateral_output_amount = _collateral_output_value;//.mul(PRICE_PRECISION).div(PRICE_PRECISION);
    }

    // check if collateral balance meets and meet output expectation
    require(_collateral_output_amount <= ERC20(collateral).balanceOf(address(this)).sub(unclaimed_pool_collateral), "<collateralBalance");
    require(_collateral_out_min <= _collateral_output_amount && _share_out_min <= _share_output_amount, ">slippage");
```

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

Resolution: Consider ordering multiplication before division.

Status: Fixed

Very Low / Informational / Best practices:

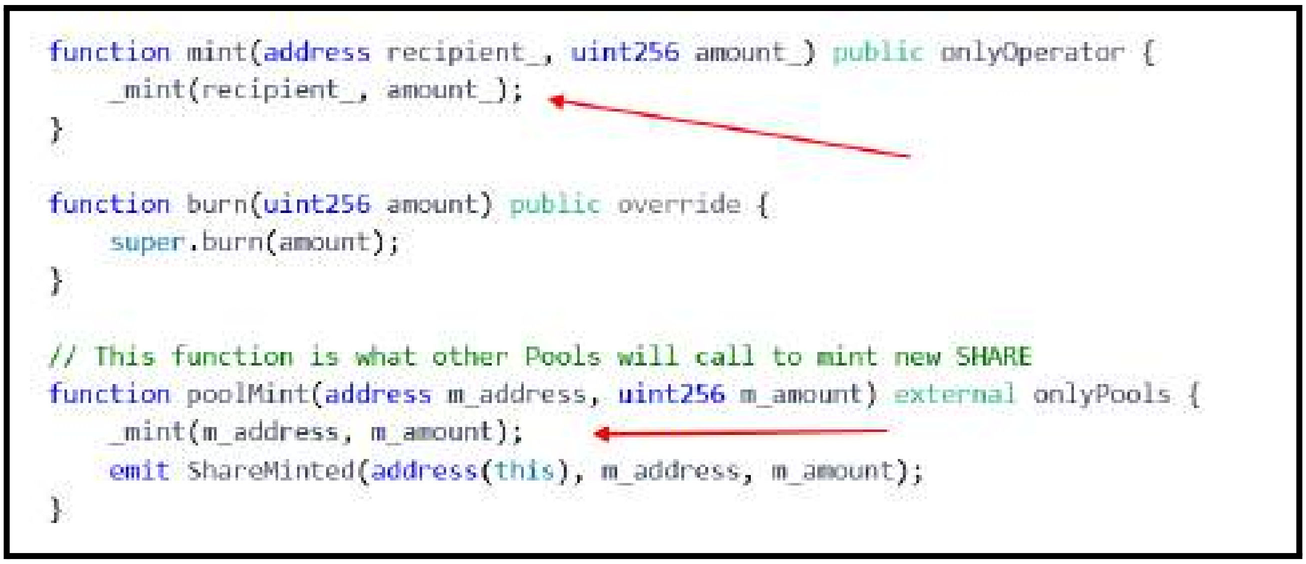
(1) Unlimited Minting:

_10BOND.sol

Operators can mint unlimited tokens.

_10Share.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

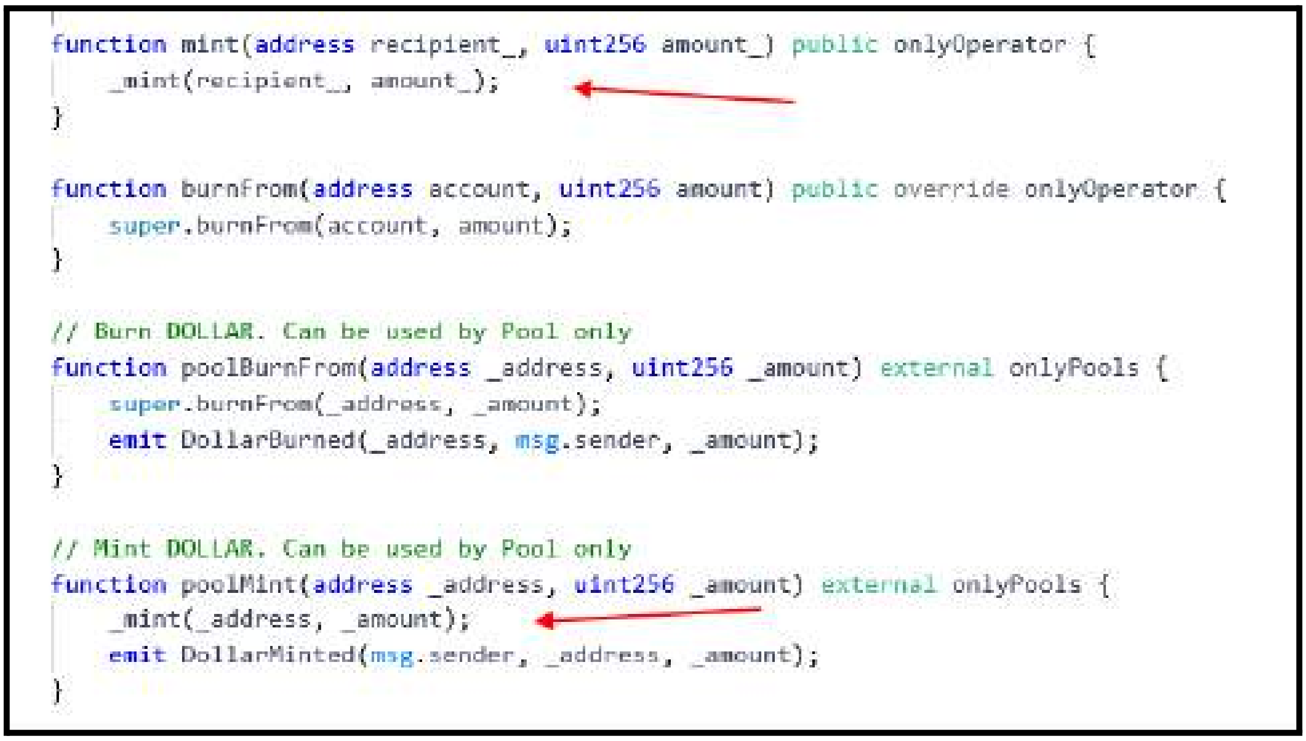


```
function mint(address recipient_, uint256 amount_) public onlyOperator {
    _mint(recipient_, amount_);
}

function burn(uint256 amount) public override {
    super.burn(amount);
}

// This function is what other Pools will call to mint new SHARE
function poolMint(address m_address, uint256 m_amount) external onlyPools {
    _mint(m_address, m_amount);
    emit ShareMinted(address(this), m_address, m_amount);
}
```

_10MB.sol



```
function mint(address recipient_, uint256 amount_) public onlyOperator {
    _mint(recipient_, amount_);
}

function burnFrom(address account, uint256 amount) public override onlyOperator {
    super.burnFrom(account, amount);
}

// Burn DOLLAR. Can be used by Pool only
function poolBurnFrom(address _address, uint256 _amount) external onlyPools {
    super.burnFrom(_address, _amount);
    emit DollarBurned(_address, msg.sender, _amount);
}

// Mint DOLLAR. Can be used by Pool only
function poolMint(address _address, uint256 _amount) external onlyPools {
    _mint(_address, _amount);
    emit DollarMinted(msg.sender, _address, _amount);
}
```

Operators & pools can mint unlimited tokens.

Resolution: We suggest putting a minting limit.

Status: Acknowledged

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) Initialize function: **Treasury.sol**

Initialize function is public.If it is not initialized first by the contract owner then anyone can initialize it.

Resolution: The owner should make sure to initialize the function before it is executed by others. Or make it accessible to onlyOwner.

Status: Fixed

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: _10BOND Operator mints basis bonds to a recipient.
- burnFrom: _10BOND Operator can burn a token from the address.
- setOperator: _10BOND owner can set a new operator address.
- setOperator: _10MB owner can set a new operator address.
- set10MBOracle: _10MB owner can set a new 10MB oracle address.
- setTaxOffice: _10MB Operator can set a new tax office address.
- excludeAddress: _10MB Operator can exclude account.
- includeAddress: _10MB Operator can include account.
- mint: _10MB Operator mints 10MB to a recipient.
- burnFrom: _10MB Operator can burn a token from the address.
- setTreasuryAddress: _10MB Operator can set a new treasury address.
- setDaoFund: _10SHARE Operator can set a new Dao fund address.
- setEquityFund: _10SHARE Operator can set a new equity fund address.
- setDevFund: _10SHARE Operator can set a new dev fund address.
- setOperator: _10SHARE Owner can set a new operator address.
- setFundMultiplier: _10SHARE Owner can set a new fund multiplier address.
- mint: _10SHARE Operator mints 10SHARE to a recipient.
- governanceRecoverUnsupported: _10SHARE Operator can governance recover unsupported.

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

- setTreasuryAddress: _10SHARE Operator can set treasury address.
- add: _10MBMasterChef owner can add a new lp to the pool.
- set: _10MBMasterChef owner can update the given pool's USDT allocation point and deposit fee.
- depositNFT: _10MBMasterChef owner can deposit NFTs.
- withdrawNFT: _10MBMasterChef owner can withdraw NFTs.
- deposit: _10MBMasterChef owner can deposit LP tokens to MasterChef for USDT allocation.
- withdraw: _10MBMasterChef owner can withdraw LP tokens from MasterChef.
- update10MBEmissionRate: _10MBMasterChef owner can update 10mb emission rate.
- update10SHAREEmissionRate: _10MBMasterChef owner can update 10share emission rate.
- setNftBaseBoostRate: _10MBMasterChef owner can set NFT boost rate.
- setNftSpecificBoost: _10MBMasterChef owner can set NFT specific boost rate.
- setNftIdSpecificBoostRange: _10MBMasterChef owner can set NFT ID specific boost range.
- removeNftIdBoostRangeById: _10MBMasterChef owner can remove NFT Id Boost Range by id.
- setNftWhitelist: _10MBMasterChef owner can set NFT Whitelist address.
- setReserveFund: _10MBMasterChef owner can set Reserve fund address.
- flipWhitelistAll: _10MBMasterChef owner can flip whitelist all addresses.
- mint: MintableERC20 owner can mint a token to a recipient.
- burn: MintableERC20 owner can burn tokens from address.
- setOperator: Boardroom owner can set operator address.
- setLockUp: Boardroom Operator can withdraw Lockup Epochs value, reward Lockup Epochs value.
- setReserveFund: Boardroom Operator can set reserve fund address.
- setStakeFee: Boardroom Operator can set stake fee.
- setWithdrawFee: Boardroom Operator can set withdrawal fee.
- allocateSeigniorage: Boardroom Operators can allocate seigniorage value.
- governanceRecoverUnsupported: Boardroom Operators can governance recover unsupported value.
- migrate: Pool Operators can move collateral to a new pool 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

- toggleMinting: Pool Operators can toggle minting.
- toggleRedeeming: Pool Operators can toggle redeeming.
- setPoolCeiling: Pool .Operators can set pool ceiling value.
- setTwapPriceScalingPercentage: Pool Operators can set twap price scaling percentage value.
- setRedemptionDelay: Pool Operators can set redemption delay.
- setTreasury: Pool Operators can set treasury addresses.
- transferCollateralToTreasury: Pool owner can transfer collateral to Treasury to execute strategies.
- transferCollateralToOperator: Pool operator can transfer collateral to Treasury to execute strategies.
- setTaxTiersTwap: TaxOffice operator can set tax tiers twap index and value.
- setTaxTiersRate: TaxOffice operator can set tax tiers rate.
- enableAutoCalculateTax: TaxOffice operator can enable auto calculate tax.
- disableAutoCalculateTax: TaxOffice operator can disable auto calculate tax.
- setTaxRate: TaxOffice operator can set tax rate.
- setBurnThreshold: TaxOffice operator can set burn threshold value.
- setTaxCollectorAddress: TaxOffice operator can set tax collector address.
- excludeAddressFromTax: TaxOffice operator can exclude address from tax.
- includeAddressInTax: TaxOffice operator can include address in tax.
- setTaxable10MBOracle: TaxOffice operator can set taxable 10mb oracle address.
- transferTaxOffice: TaxOffice operator can transfer tax office address.
- setTaxExclusionForAddress: TaxOffice operator can set tax exclusion for address.
- set10MB: TaxOracle owner can set 10mb address.
- setUsdt: TaxOracle owner can set USDT address.
- setPair: TaxOracle owner can set Pair address.
- setOperator: Treasury Operator can set operator address.
- setBoardroom: Treasury Operator can set boardroom address.
- setBoardroomWithdrawFee: Treasury Operator can set boardroom withdrawal fee.
- setBoardroomStakeFee: Treasury Operator can set boardroom stake fee.
- set10MBOracle: Treasury Operator can set 10MB oracle address.
- set10MBPriceCeiling: Treasury Operator can set 10MB price ceiling value.
- setMinMaxSupplyExpansionPercent: Treasury Operator can set minimum and maximum supply expansion percentage.

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

- setMaxSupplyExpansionPercent: Treasury Operator can set maximum supply expansion percentage.
- setBondDepletionFloorPercent: Treasury Operator can set bond depletion floor percentage.
- setMaxSupplyContractionPercent: Treasury Operator can set maximum supply contraction percentage.
- setMaxDebtRatioPercent: Treasury Operator can set maximum debt ratio percentage.
- setBootstrap: Treasury Operator can set bootstrap values.
- setExtraFunds: Treasury Operator can set extra funds values.
- setAllocateSeigniorageSalary: Treasury Operator can set allocation seigniorage salary.
- setMaxDiscountRate: Treasury Operator can set maximum discount rate.
- setMaxPremiumRate: Treasury Operator can set maximum premium rate.
- setDiscountPercent: Treasury Operator can set discount percentage.
- setPremiumThreshold: Treasury Operator can set premium threshold.
- setPremiumPercent: Treasury Operator can set premium percentage.
- setMintingFactorForPayingDebt: Treasury Operator can set minting factor for paying debt.
- set10MBSupplyTarget: Treasury Operator can set 10MB supply target value.
- addPool: Treasury Operator can add a new Pool.
- removePool: : Treasury Operator can remove Pool.
- governanceRecoverUnsupported: Treasury Operator can governance recover unsupported.
- boardroomSetOperator: Treasury operator can set boardroom operator address.
- boardroomSetReserveFund: Treasury operator can set boardroom reserve fund address.
- boardroomSetLockUp: Treasury operator can set boardroom lockup value.
- boardroomAllocateSeigniorage:Treasury operator can allocate boardroom seigniorage value.
- boardroomGovernanceRecoverUnsupported: Treasury operator can recover boardroom governance unsupported value.
- setRedemptionFee: Treasury operator can set redemption fee.
- setMintingFee: Treasury operator can set minting fee.

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

- setRatioStep: Treasury operator can set ratio steps.
- setPriceTarget: Treasury operator can set price target values.
- setRefreshCooldown: Treasury operator can set refresh cooldown value.
- setPriceBand: Treasury operator can set price band value.
- toggleCollateralRatio: Treasury operator can toggle collateral ratio.
- toggleEffectiveCollateralRatio: Treasury operator can toggle effective collateral ratio.
- executeTransaction: Treasury operator can execute transactions.

To make the smart contract 100% decentralized, we suggest renouncing ownership in the 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 a file. And we have used all possible tests based on given objects as files. We have observed some issues in the smart contracts and those are fixed/ acknowledged in the revised code. **So, 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

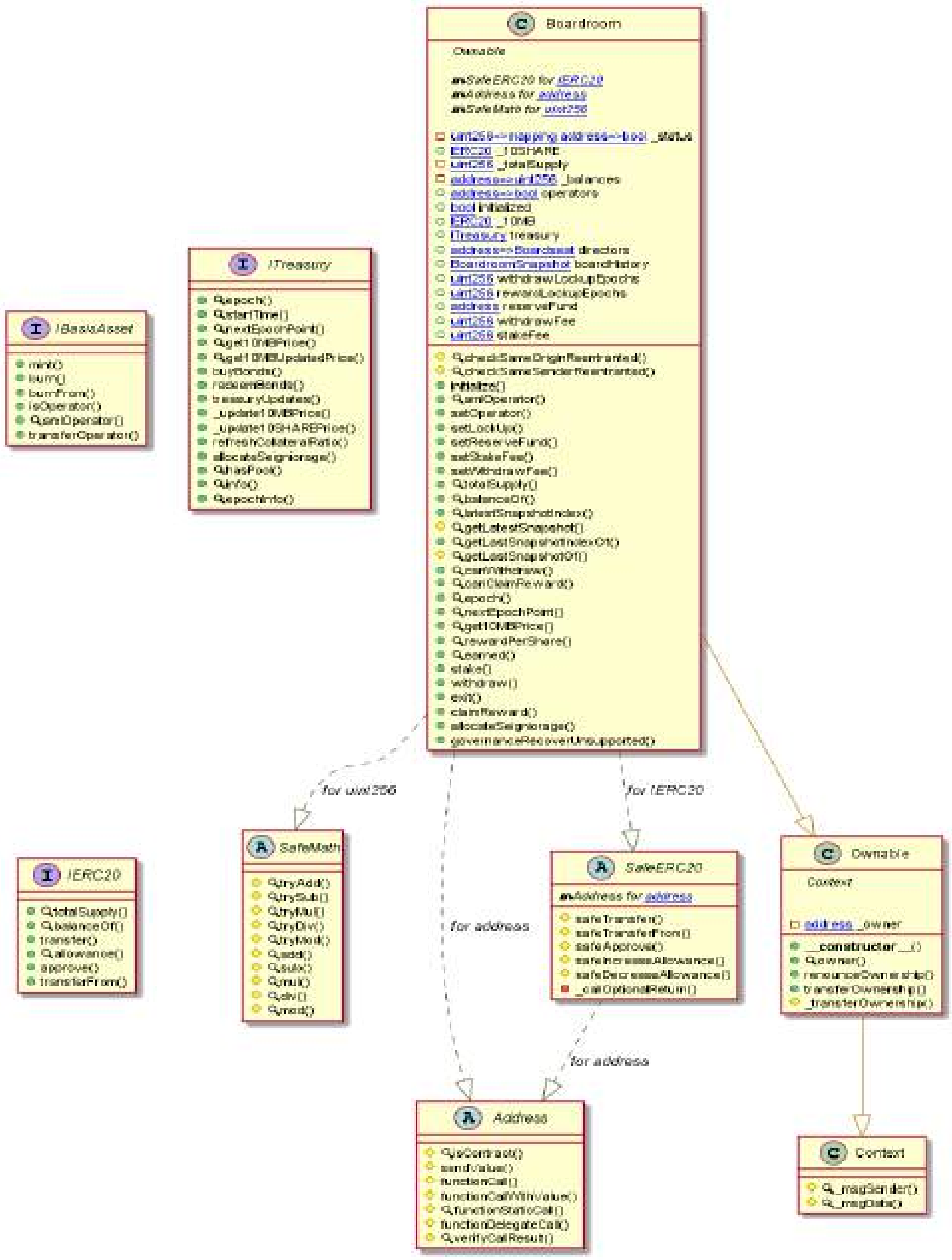
Smart contracts are deployed and executed on the blockchain platform. The platform, its programming language, and other software related to the smart contract can have their own vulnerabilities that can lead to hacks. Thus, the audit can’t guarantee explicit security of the audited smart contracts.

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

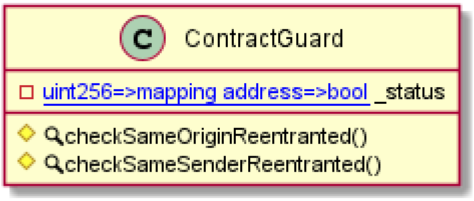
Code Flow Diagram - 10MB Finance Protocol

Boardroom Diagram

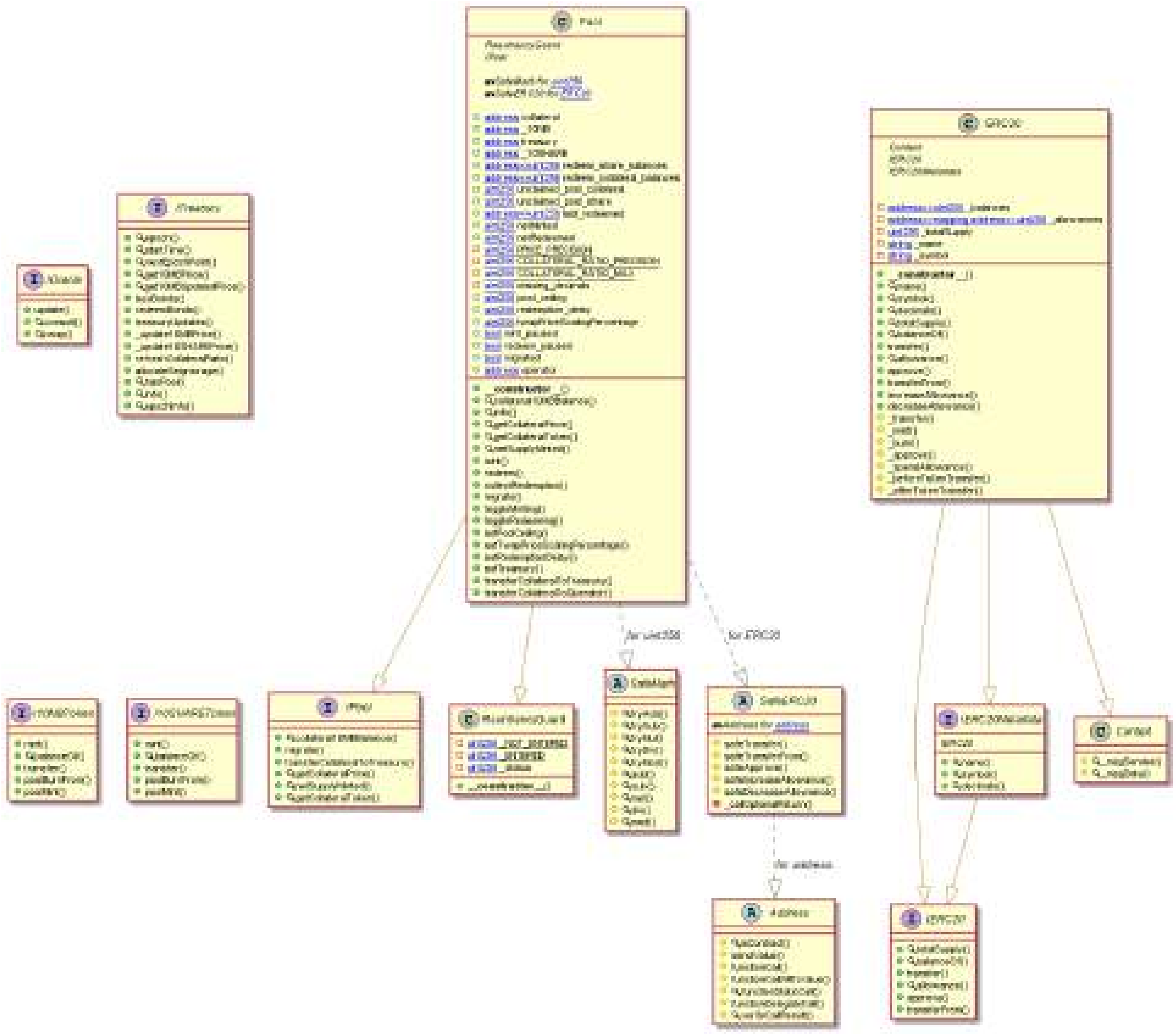


This is a private and confidential document. No part of this document should be disclosed to third party without prior written permission of EtherAuthority.
Email: audit@EtherAuthority.io

ContractGuard Diagram

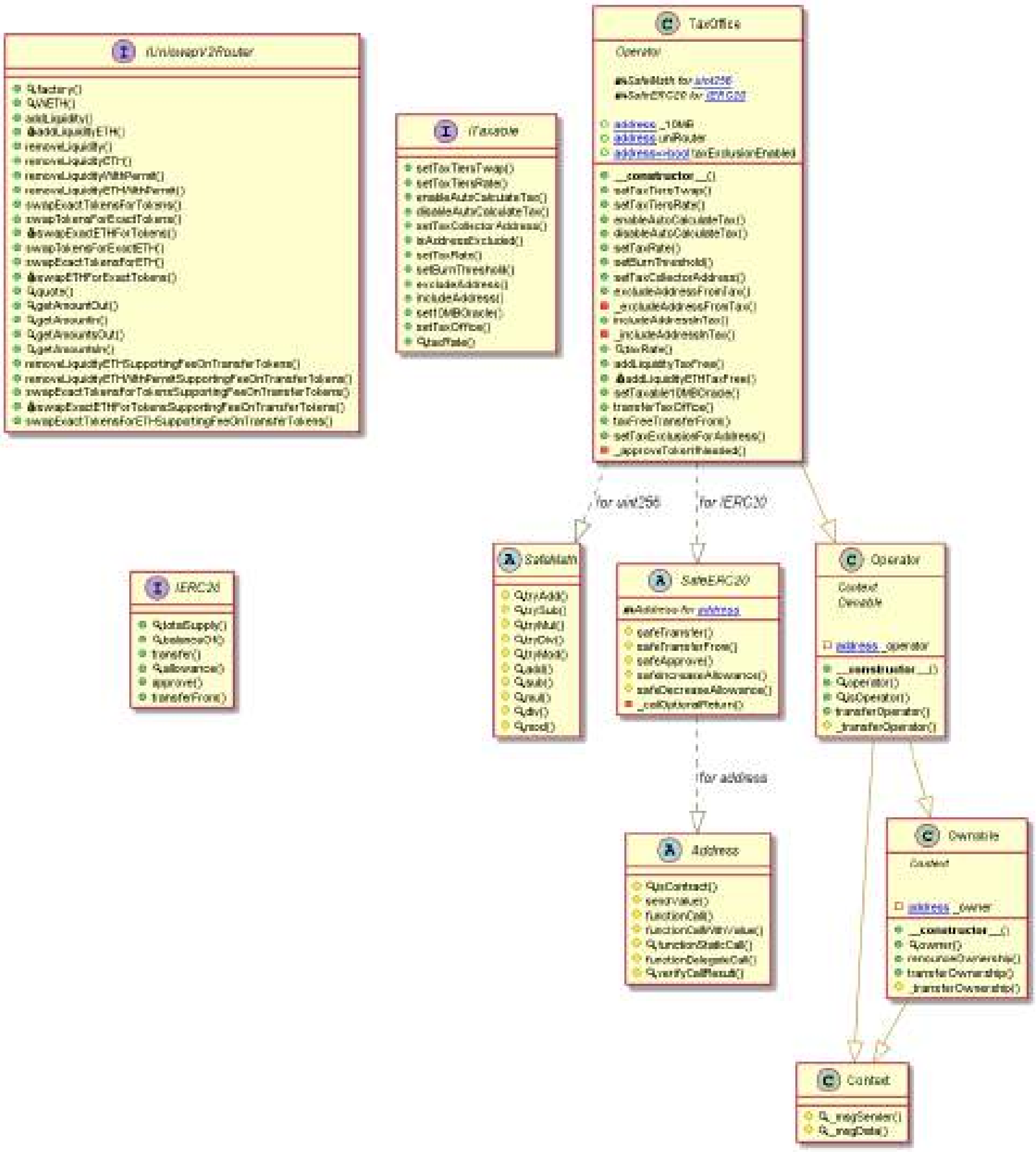

ContractGuard
uint256=>mapping address=>bool _status
checkSameOriginReentranted()
checkSameSenderReentranted()

Pool Diagram



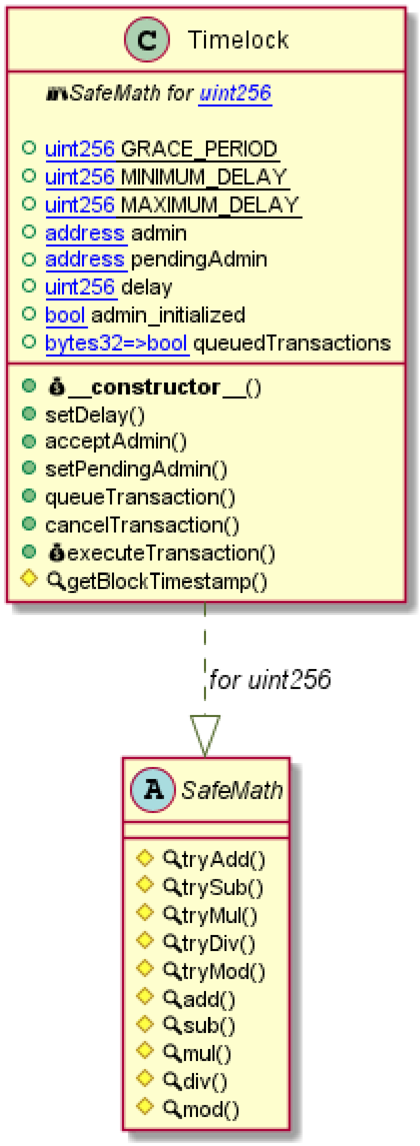
This is a private and confidential document. No part of this document should be disclosed to third party without prior written permission of EtherAuthority.
Email: audit@EtherAuthority.io

TaxOffice Diagram



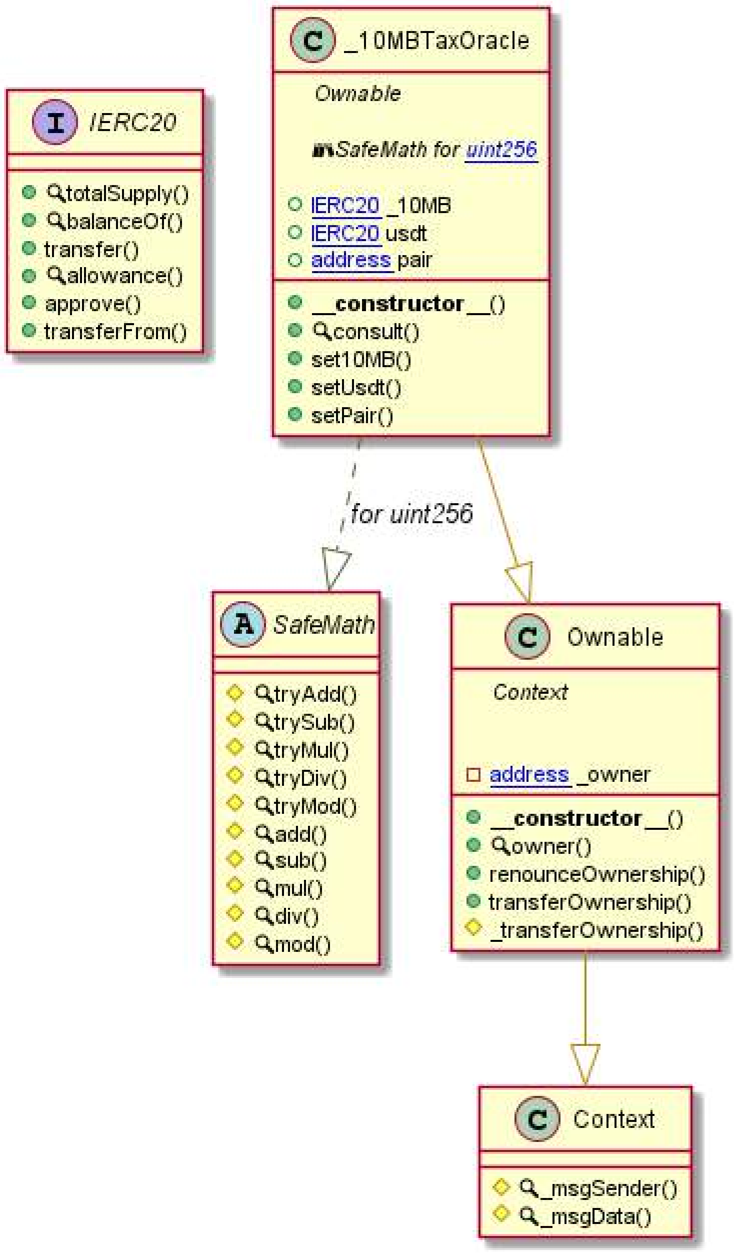
This is a private and confidential document. No part of this document should be disclosed to third party without prior written permission of EtherAuthority.
Email: audit@EtherAuthority.io

Timelock Diagram

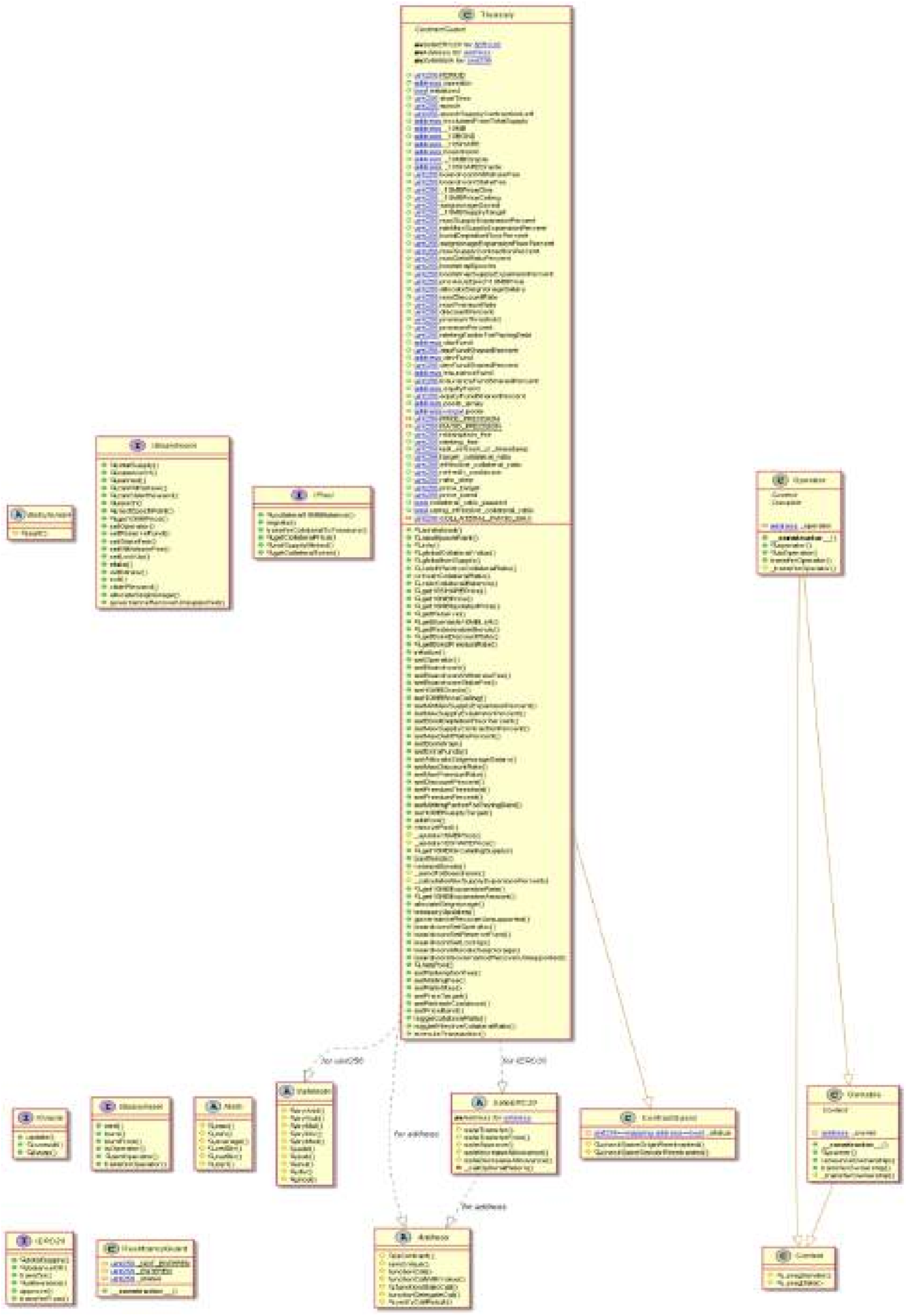


This is a private and confidential document. No part of this document should be disclosed to third party without prior written permission of EtherAuthority.
Email: audit@EtherAuthority.io

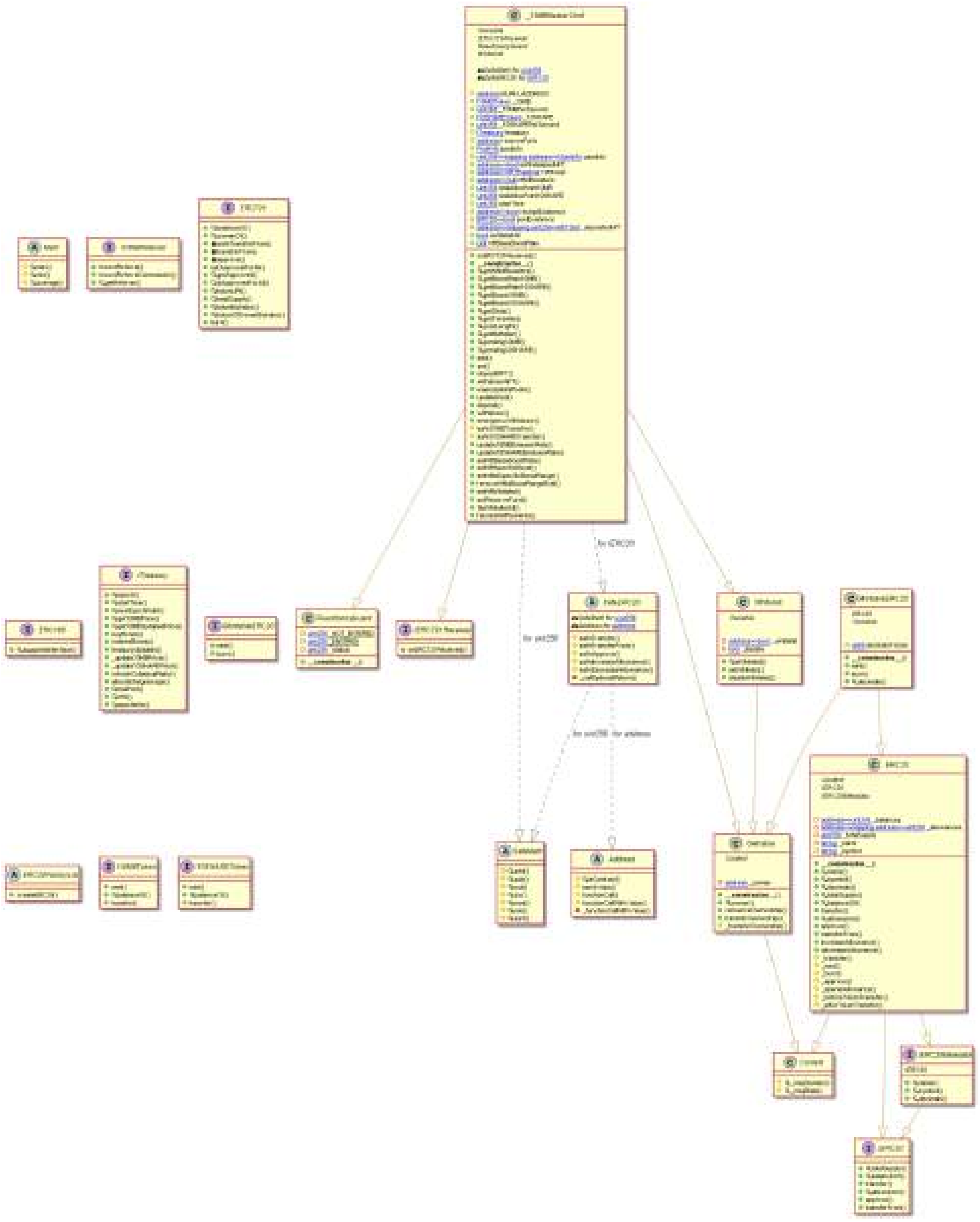
TaxOracle Diagram



Treasury Diagram

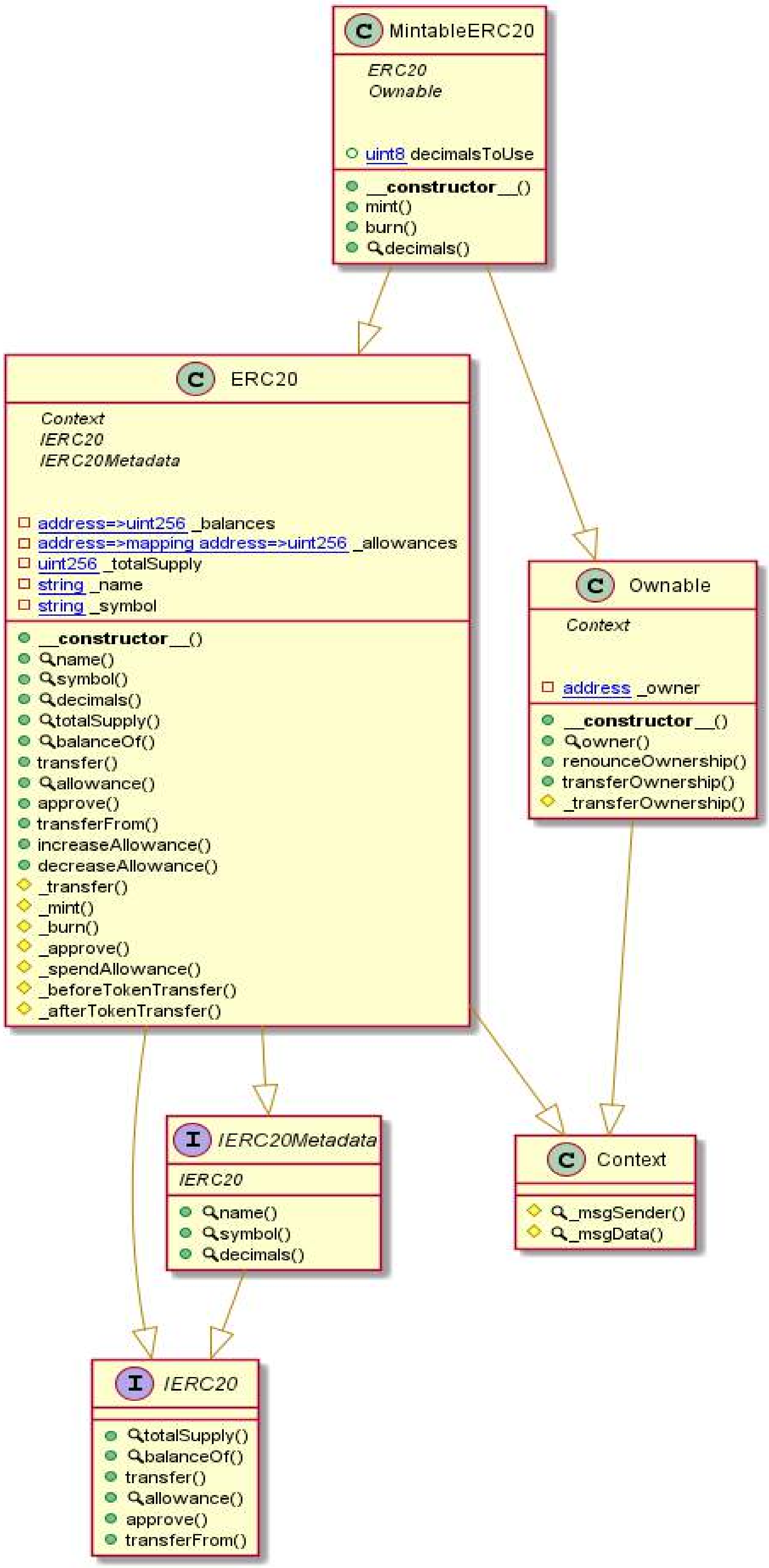


_10MBMasterchef Diagram



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

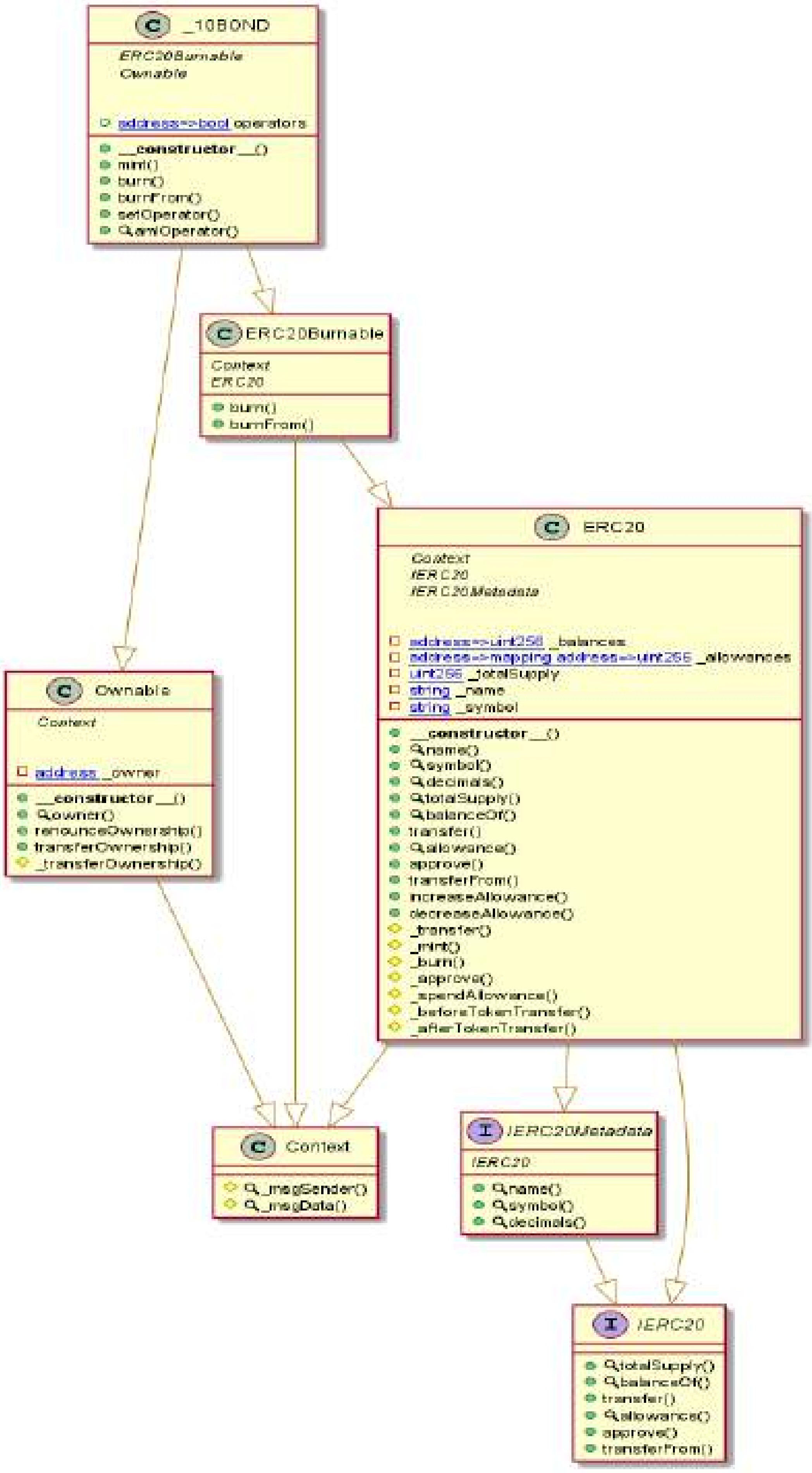
MintableERC20 Diagram



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

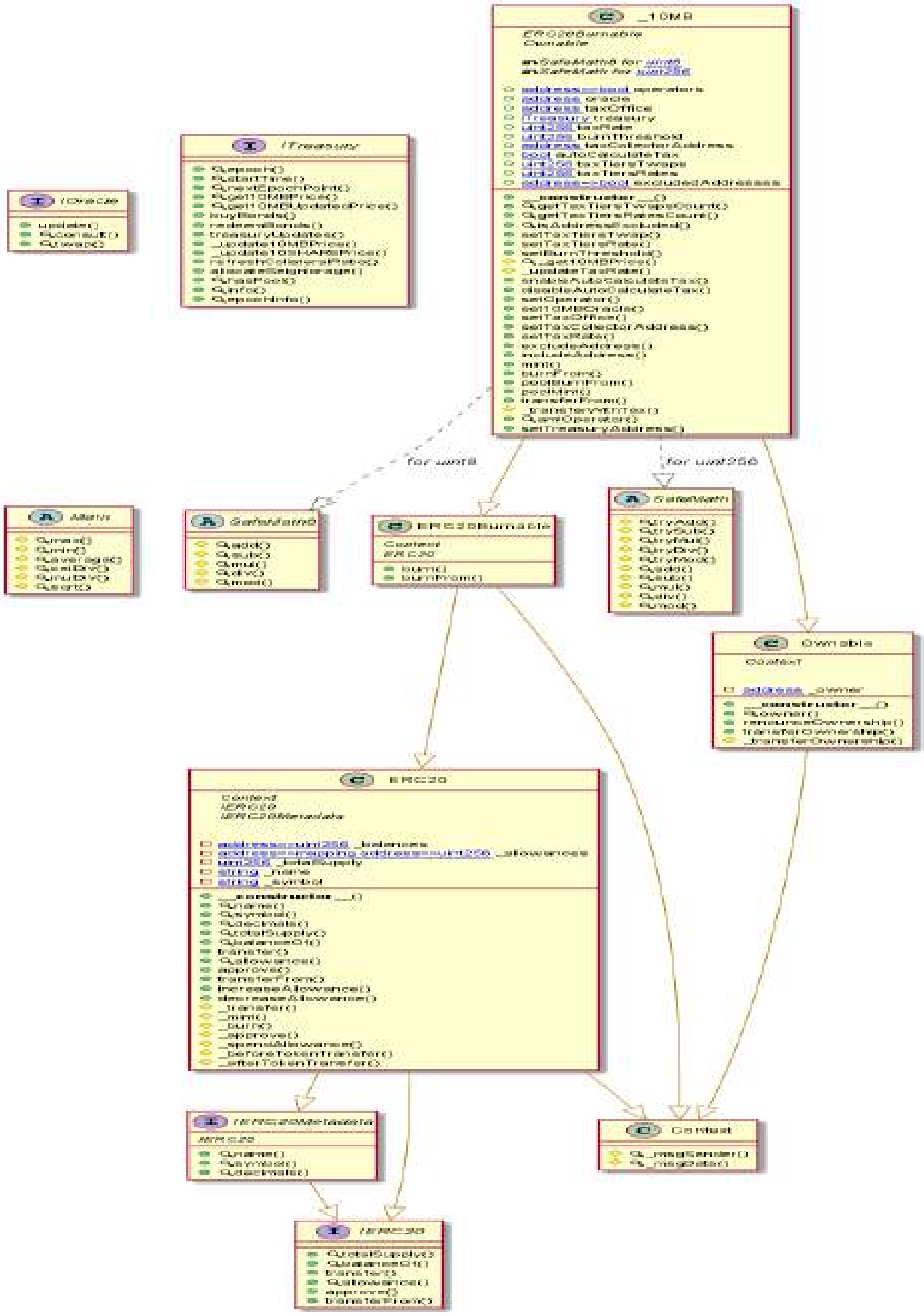
_10BOND Diagram



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

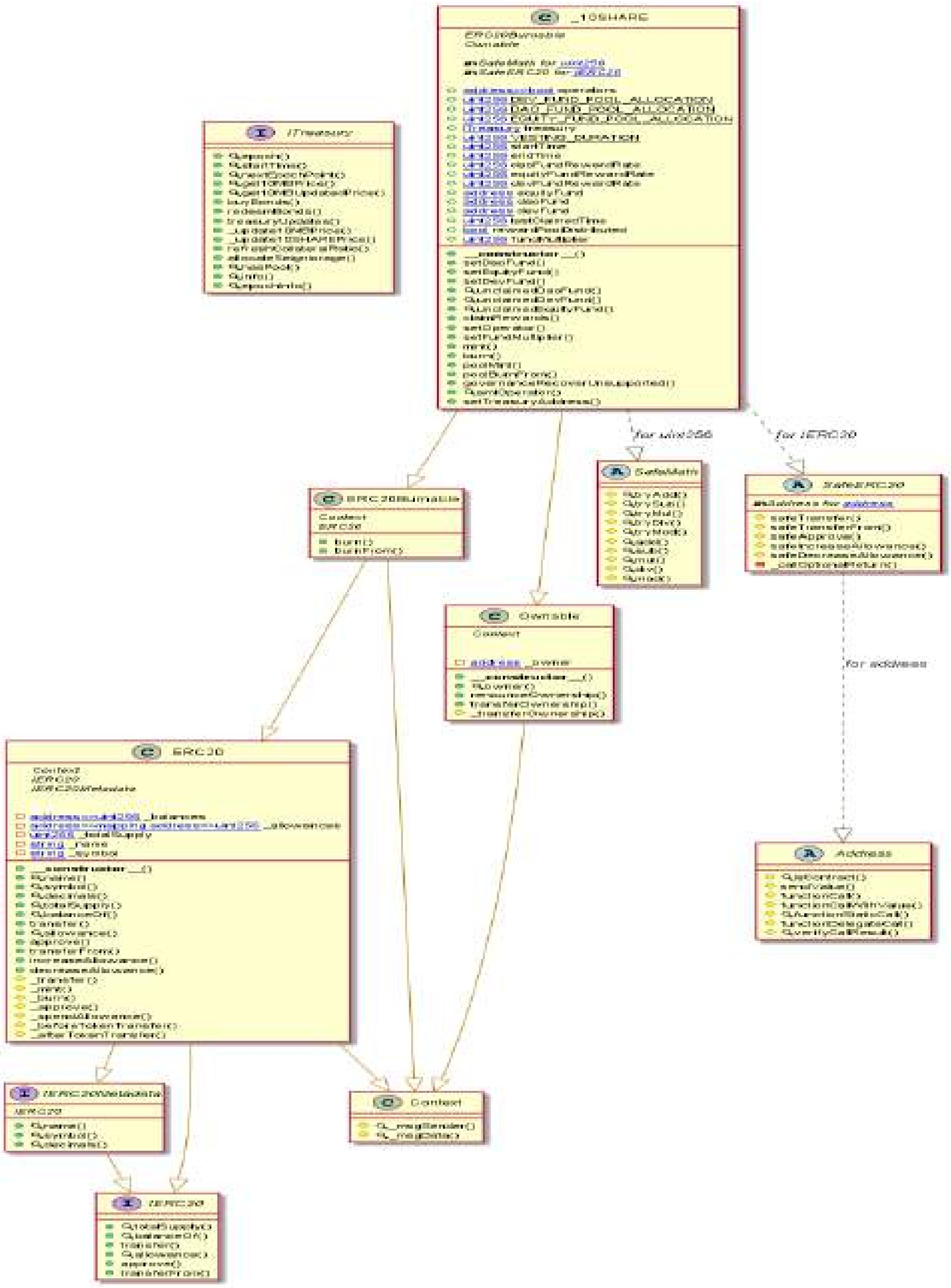
_10MB Diagram



_

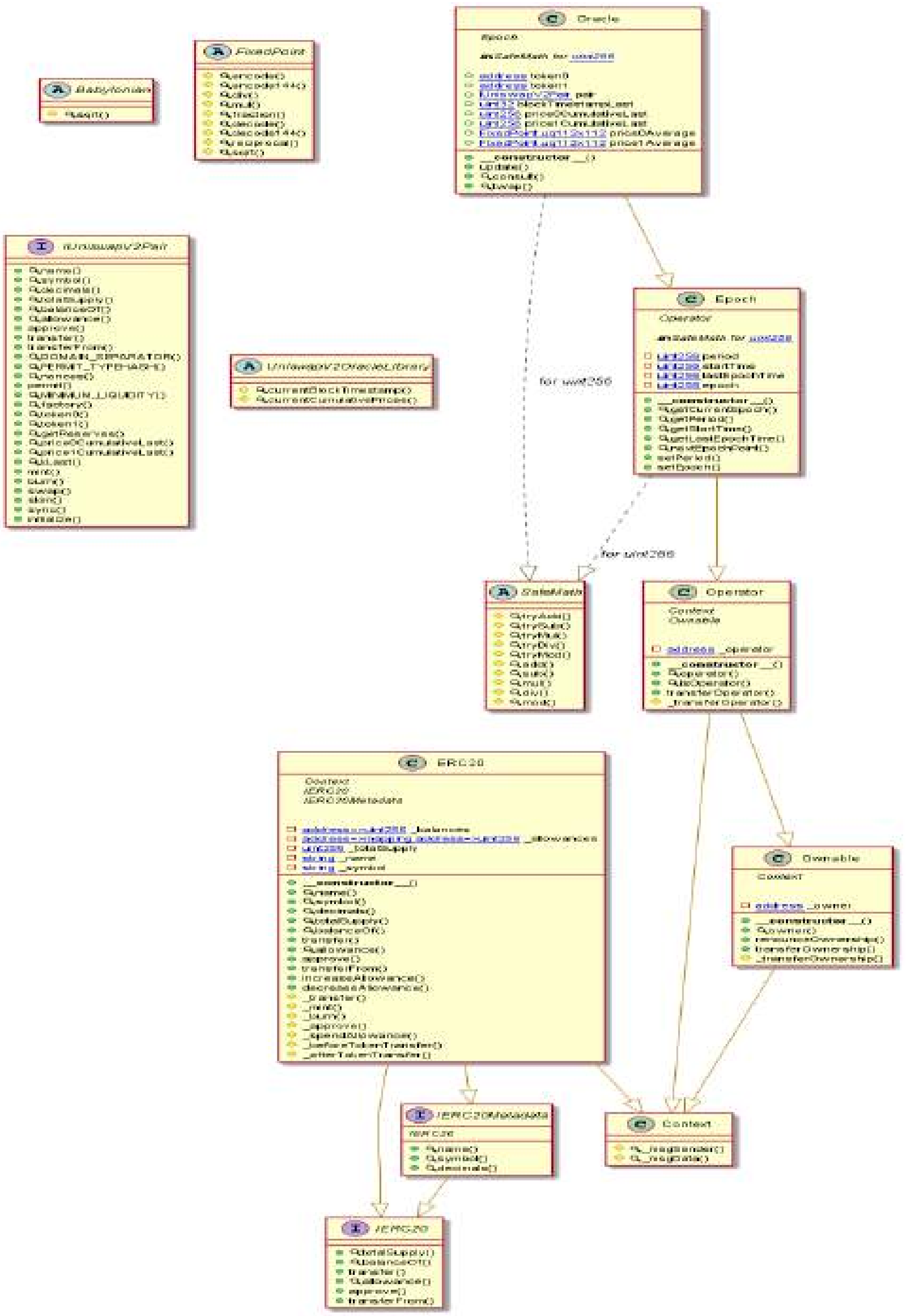
This is a private and confidential document. No part of this document should be disclosed to third party without prior written permission of EtherAuthority.
Email: audit@EtherAuthority.io

10SHARE Diagram



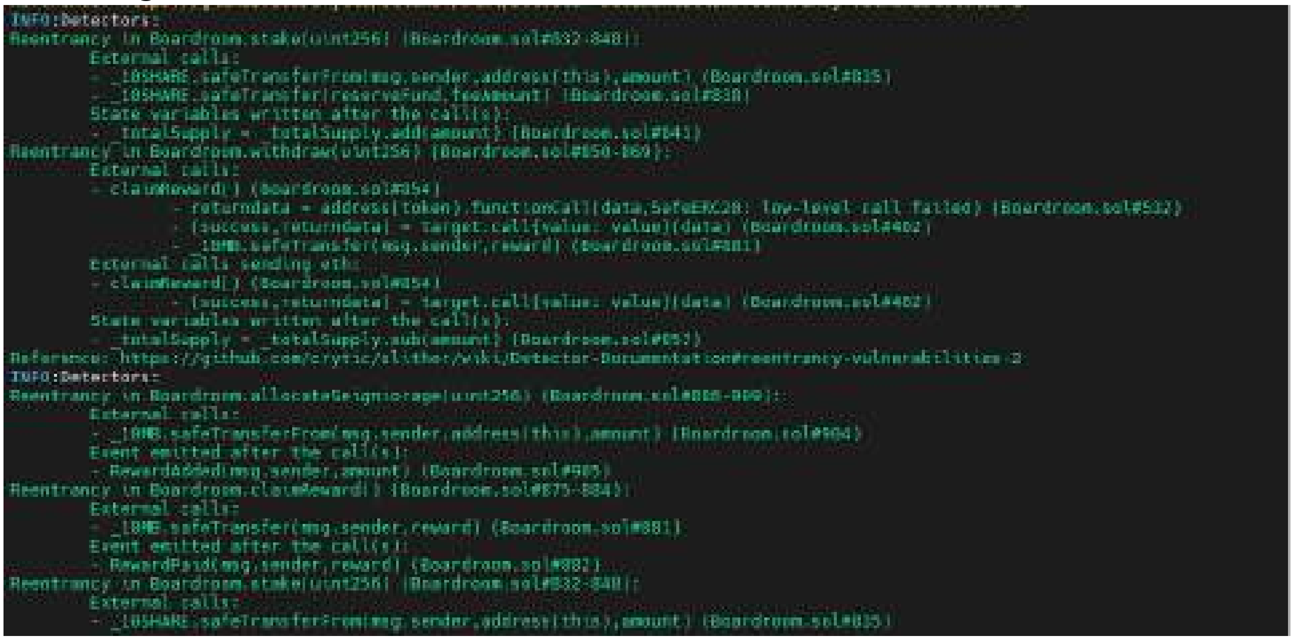
This is a private and confidential document. No part of this document should be disclosed to third party without prior written permission of EtherAuthority.
Email: audit@EtherAuthority.io

Oracle Diagram

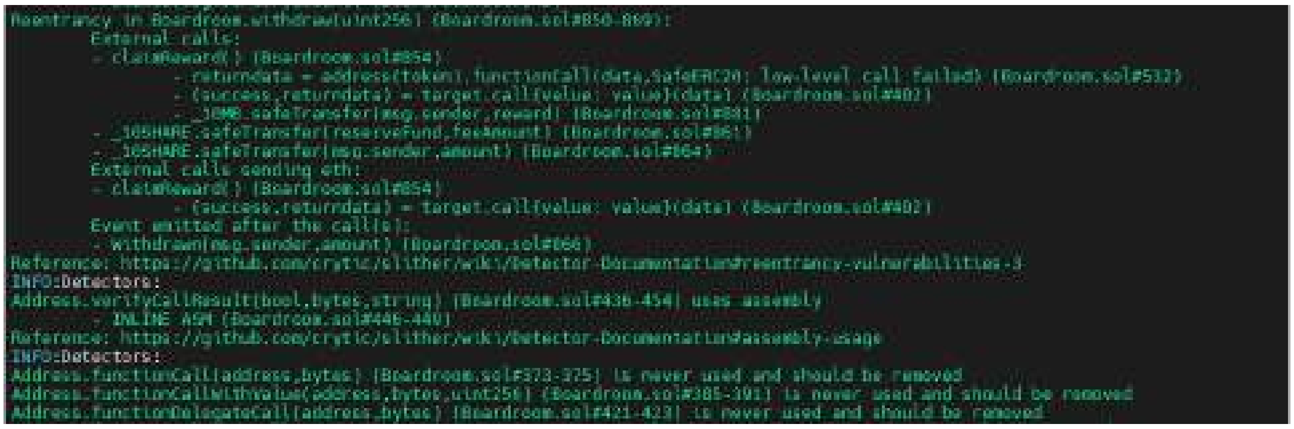


Slither Results Log

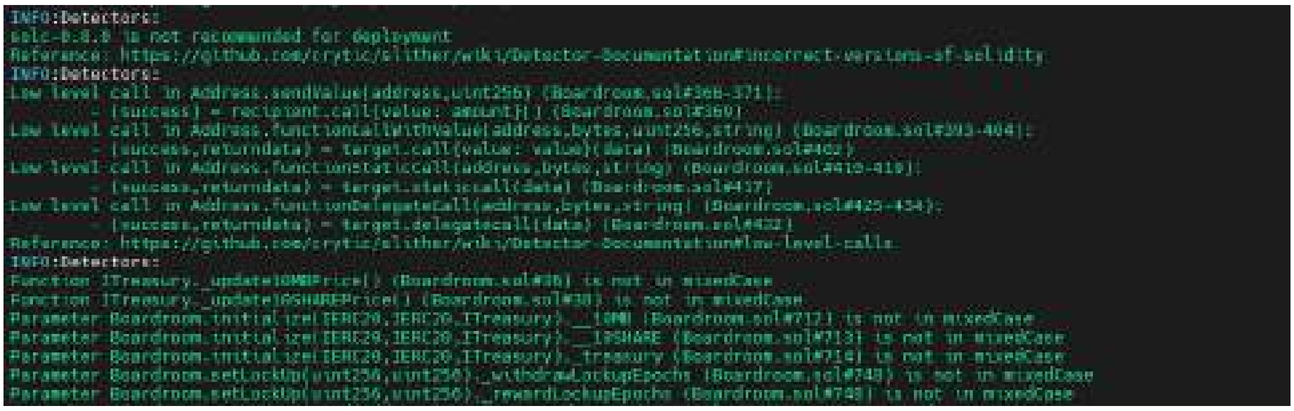
Slither log >> Boardroom.sol



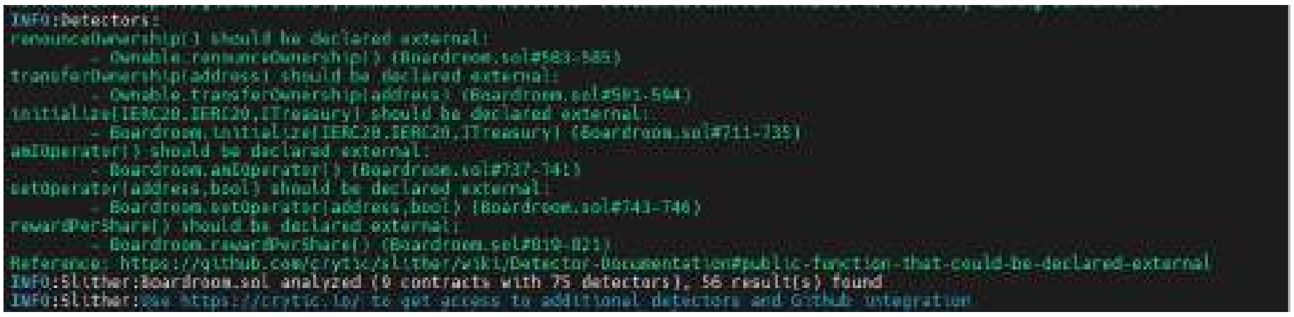
```
INFO:Detectors:
Reentrancy in Boardroom.stake(uint256) (Boardroom.sol#832-848):
        External calls:
        - _10SHARE.safeTransferFrom(msg.sender,address(this),amount) (Boardroom.sol#835)
        - _10SHARE.safeTransfer(reserveFund,feeAmount) (Boardroom.sol#838)
        State variables written after the call(s):
        - _totalSupply = _totalSupply.add(amount) (Boardroom.sol#841)
Reentrancy in Boardroom.withdraw(uint256) (Boardroom.sol#850-869):
        External calls:
        - claimReward() (Boardroom.sol#854)
                - returndata = address(token).functionCall(data,SafeERC20: low-level call failed) (Boardroom.sol#532)
                - (success,returndata) = target.call{value: value}(data) (Boardroom.sol#402)
                - _10MB.safeTransfer(msg.sender,reward) (Boardroom.sol#881)
        External calls sending eth:
        - claimReward() (Boardroom.sol#854)
                - (success,returndata) = target.call{value: value}(data) (Boardroom.sol#402)
        State variables written after the call(s):
        - _totalSupply = _totalSupply.sub(amount) (Boardroom.sol#857)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#reentrancy-vulnerabilities-2
INFO:Detectors:
Reentrancy in Boardroom.allocateSeigniorage(uint256) (Boardroom.sol#888-909):
        External calls:
        - _10MB.safeTransferFrom(msg.sender,address(this),amount) (Boardroom.sol#904)
        Event emitted after the call(s):
        - RewardAdded(msg.sender,amount) (Boardroom.sol#905)
Reentrancy in Boardroom.claimReward() (Boardroom.sol#875-884):
        External calls:
        - _10MB.safeTransfer(msg.sender,reward) (Boardroom.sol#881)
        Event emitted after the call(s):
        - RewardPaid(msg.sender,reward) (Boardroom.sol#882)
Reentrancy in Boardroom.stake(uint256) (Boardroom.sol#832-848):
        External calls:
        - _10SHARE.safeTransferFrom(msg.sender,address(this),amount) (Boardroom.sol#835)
```



```
Reentrancy in Boardroom.withdraw(uint256) (Boardroom.sol#850-869):
        External calls:
        - claimReward() (Boardroom.sol#854)
                - returndata = address(token).functionCall(data,SafeERC20: low-level call failed) (Boardroom.sol#532)
                - (success,returndata) = target.call{value: value}(data) (Boardroom.sol#402)
                - _10MB.safeTransfer(msg.sender,reward) (Boardroom.sol#881)
        - _10SHARE.safeTransfer(reserveFund,feeAmount) (Boardroom.sol#861)
        - _10SHARE.safeTransfer(msg.sender,amount) (Boardroom.sol#864)
        External calls sending eth:
        - claimReward() (Boardroom.sol#854)
                - (success,returndata) = target.call{value: value}(data) (Boardroom.sol#402)
        Event emitted after the call(s):
        - Withdrawn(msg.sender,amount) (Boardroom.sol#866)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#reentrancy-vulnerabilities-3
INFO:Detectors:
Address.verifyCallResult(bool,bytes,string) (Boardroom.sol#436-454) uses assembly
        - INLINE ASM (Boardroom.sol#446-449)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#assembly-usage
INFO:Detectors:
Address.functionCall(address,bytes) (Boardroom.sol#373-375) is never used and should be removed
Address.functionCallWithValue(address,bytes,uint256) (Boardroom.sol#385-391) is never used and should be removed
Address.functionDelegateCall(address,bytes) (Boardroom.sol#421-423) is never used and should be removed
```



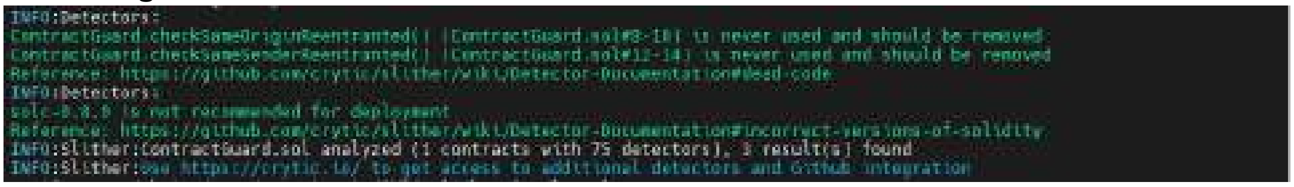
```
INFO:Detectors:
solc-0.8.0 is not recommended for deployment
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#incorrect-versions-of-solidity
INFO:Detectors:
Low level call in Address.sendValue(address,uint256) (Boardroom.sol#366-371):
        - (success) = recipient.call{value: amount}() (Boardroom.sol#369)
Low level call in Address.functionCallWithValue(address,bytes,uint256,string) (Boardroom.sol#393-404):
        - (success,returndata) = target.call{value: value}(data) (Boardroom.sol#402)
Low level call in Address.functionStaticCall(address,bytes,string) (Boardroom.sol#410-419):
        - (success,returndata) = target.staticcall(data) (Boardroom.sol#417)
Low level call in Address.functionDelegateCall(address,bytes,string) (Boardroom.sol#425-434):
        - (success,returndata) = target.delegatecall(data) (Boardroom.sol#432)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#low-level-calls
INFO:Detectors:
Function ITreasury._update10MBPrice() (Boardroom.sol#36) is not in mixedCase
Function ITreasury._update10SHAREPrice() (Boardroom.sol#38) is not in mixedCase
Parameter Boardroom.initialize(IERC20,IERC20,ITreasury).__10MB (Boardroom.sol#712) is not in mixedCase
Parameter Boardroom.initialize(IERC20,IERC20,ITreasury).__10SHARE (Boardroom.sol#713) is not in mixedCase
Parameter Boardroom.initialize(IERC20,IERC20,ITreasury)._treasury (Boardroom.sol#714) is not in mixedCase
Parameter Boardroom.setLockUp(uint256,uint256)._withdrawLockupEpochs (Boardroom.sol#748) is not in mixedCase
Parameter Boardroom.setLockUp(uint256,uint256)._rewardLockupEpochs (Boardroom.sol#748) is not in mixedCase
```



```
INFO:Detectors:
renounceOwnership() should be declared external:
        - Ownable.renounceOwnership() (Boardroom.sol#583-585)
transferOwnership(address) should be declared external:
        - Ownable.transferOwnership(address) (Boardroom.sol#591-594)
initialize(IERC20,IERC20,ITreasury) should be declared external:
        - Boardroom.initialize(IERC20,IERC20,ITreasury) (Boardroom.sol#711-735)
amIOperator() should be declared external:
        - Boardroom.amIOperator() (Boardroom.sol#737-741)
setOperator(address,bool) should be declared external:
        - Boardroom.setOperator(address,bool) (Boardroom.sol#743-746)
rewardPerShare() should be declared external:
        - Boardroom.rewardPerShare() (Boardroom.sol#819-821)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#public-function-that-could-be-declared-external
INFO:Slither:Boardroom.sol analyzed (9 contracts with 75 detectors), 56 result(s) found
INFO:Slither:Use https://crytic.io/ to get access to additional detectors and Github integration
```

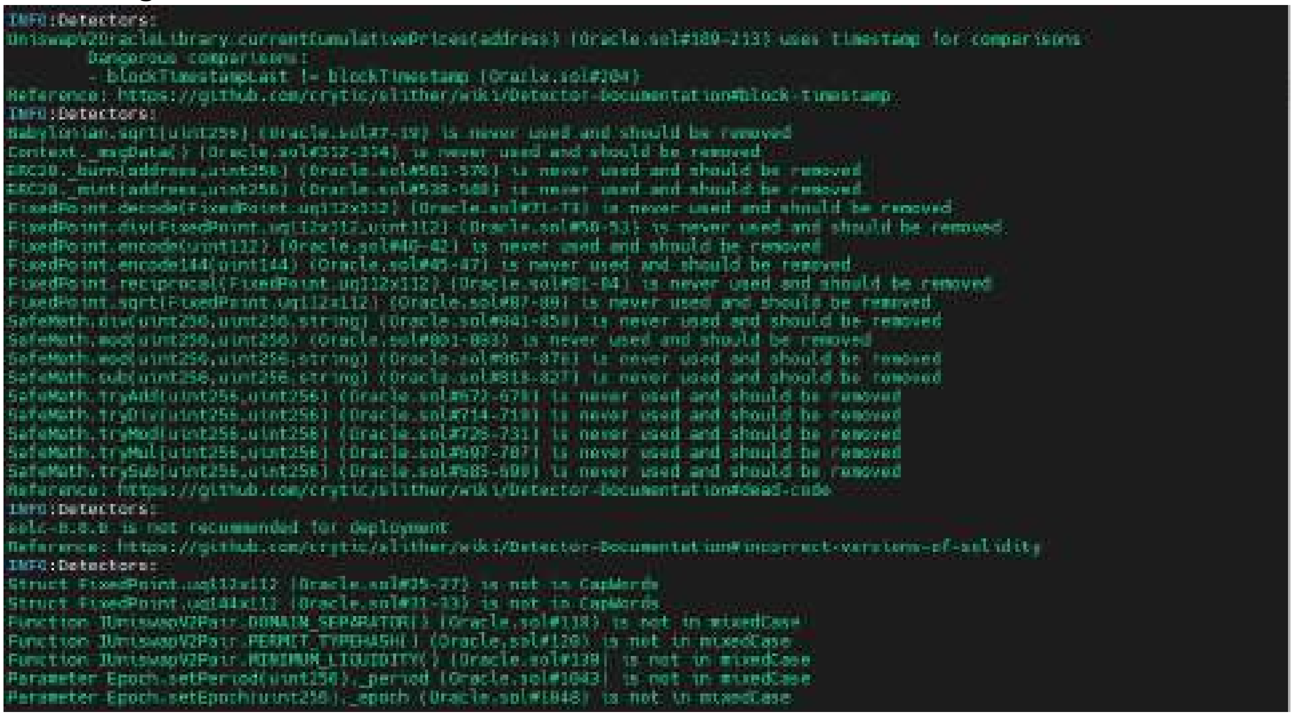
This is a private and confidential document. No part of this document should be disclosed to third party without prior written permission of EtherAuthority.
Email: audit@EtherAuthority.io

Slither log >> ContractGuard.sol

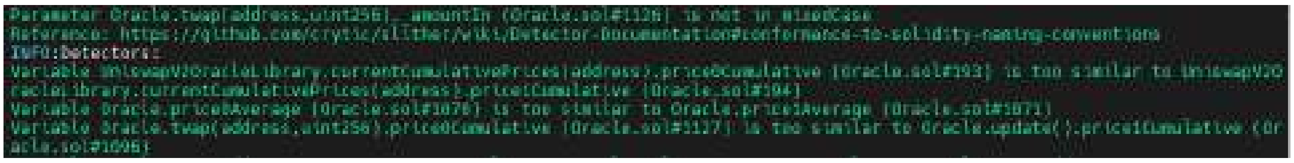


```
INFO:Detectors:
ContractGuard.checkSameOriginReentranted() (ContractGuard.sol#8-10) is never used and should be removed
ContractGuard.checkSameSenderReentranted() (ContractGuard.sol#12-14) is never used and should be removed
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#dead-code
INFO:Detectors:
solc-0.8.0 is not recommended for deployment
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#incorrect-versions-of-solidity
INFO:Slither:ContractGuard.sol analyzed (1 contracts with 75 detectors), 3 result(s) found
INFO:Slither:Use https://crytic.io/ to get access to additional detectors and Github integration
```

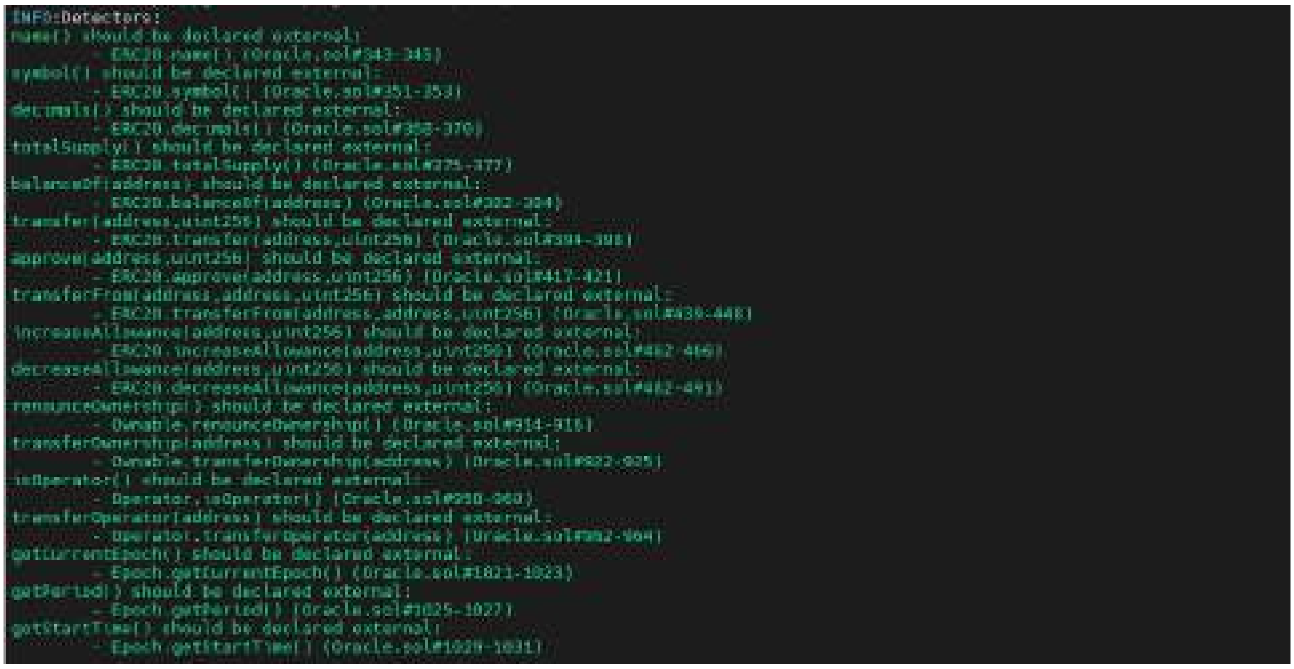
Slither log >> Oracle.sol



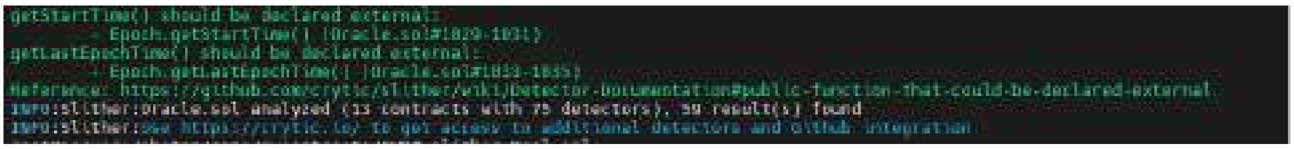
```
INFO:Detectors:
UniswapV2OracleLibrary.currentCumulativePrices(address) (Oracle.sol#189-213) uses timestamp for comparisons
        Dangerous comparisons:
        - blockTimestampLast != blockTimestamp (Oracle.sol#204)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#block-timestamp
INFO:Detectors:
Babylonian.sqrt(uint256) (Oracle.sol#7-19) is never used and should be removed
Context._msgData() (Oracle.sol#312-314) is never used and should be removed
ERC20._burn(address,uint256) (Oracle.sol#561-578) is never used and should be removed
ERC20._mint(address,uint256) (Oracle.sol#538-588) is never used and should be removed
FixedPoint.decode(FixedPoint.uq112x112) (Oracle.sol#71-73) is never used and should be removed
FixedPoint.div(FixedPoint.uq112x112,uint112) (Oracle.sol#50-53) is never used and should be removed
FixedPoint.encode(uint112) (Oracle.sol#40-42) is never used and should be removed
FixedPoint.encode144(uint144) (Oracle.sol#45-47) is never used and should be removed
FixedPoint.reciprocal(FixedPoint.uq112x112) (Oracle.sol#81-84) is never used and should be removed
FixedPoint.sqrt(FixedPoint.uq112x112) (Oracle.sol#87-89) is never used and should be removed
SafeMath.div(uint256,uint256,string) (Oracle.sol#841-850) is never used and should be removed
SafeMath.mod(uint256,uint256) (Oracle.sol#801-803) is never used and should be removed
SafeMath.mod(uint256,uint256,string) (Oracle.sol#867-876) is never used and should be removed
SafeMath.sub(uint256,uint256,string) (Oracle.sol#818-827) is never used and should be removed
SafeMath.tryAdd(uint256,uint256) (Oracle.sol#672-678) is never used and should be removed
SafeMath.tryDiv(uint256,uint256) (Oracle.sol#714-719) is never used and should be removed
SafeMath.tryMod(uint256,uint256) (Oracle.sol#726-731) is never used and should be removed
SafeMath.tryMul(uint256,uint256) (Oracle.sol#697-707) is never used and should be removed
SafeMath.trySub(uint256,uint256) (Oracle.sol#685-690) is never used and should be removed
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#dead-code
INFO:Detectors:
solc-0.8.0 is not recommended for deployment
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#incorrect-versions-of-solidity
INFO:Detectors:
Struct FixedPoint.uq112x112 (Oracle.sol#25-27) is not in CapWords
Struct FixedPoint.uq144x112 (Oracle.sol#31-33) is not in CapWords
Function IUniswapV2Pair.DOMAIN_SEPARATOR() (Oracle.sol#118) is not in mixedCase
Function IUniswapV2Pair.PERMIT_TYPEHASH() (Oracle.sol#120) is not in mixedCase
Function IUniswapV2Pair.MINIMUM_LIQUIDITY() (Oracle.sol#138) is not in mixedCase
Parameter Epoch.setPeriod(uint256)._period (Oracle.sol#1043) is not in mixedCase
Parameter Epoch.setEpoch(uint256)._epoch (Oracle.sol#1048) is not in mixedCase
```



```
Parameter Oracle.twap(address,uint256)._amountIn (Oracle.sol#1126) is not in mixedCase
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#conformance-to-solidity-naming-conventions
INFO:Detectors:
Variable UniswapV2OracleLibrary.currentCumulativePrices(address).price0Cumulative (Oracle.sol#193) is too similar to UniswapV2O
racleLibrary.currentCumulativePrices(address).price1Cumulative (Oracle.sol#194)
Variable Oracle.price0Average (Oracle.sol#1070) is too similar to Oracle.price1Average (Oracle.sol#1071)
Variable Oracle.twap(address,uint256).price0Cumulative (Oracle.sol#1127) is too similar to Oracle.update().price1Cumulative (Or
acle.sol#1096)
```



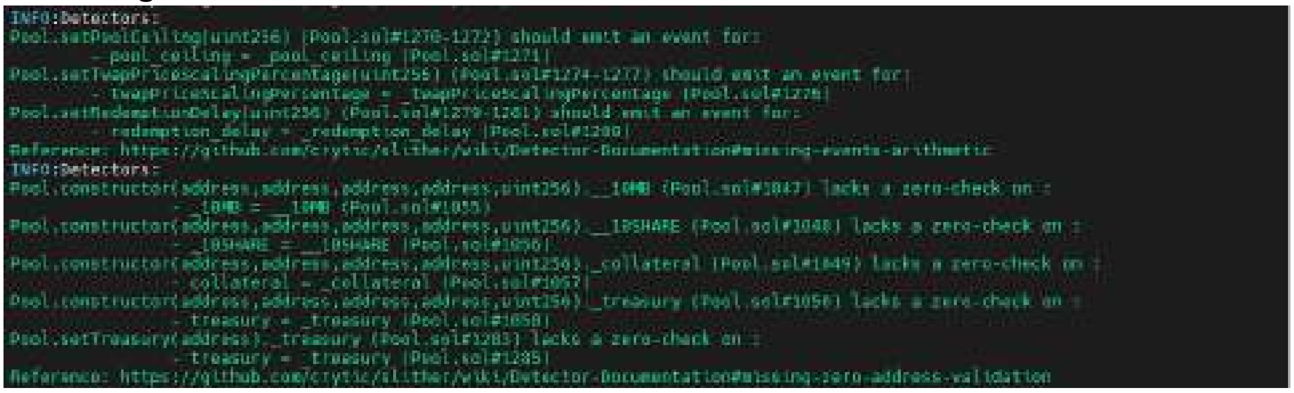
```
INFO:Detectors:
name() should be declared external:
        - ERC20.name() (Oracle.sol#343-345)
symbol() should be declared external:
        - ERC20.symbol() (Oracle.sol#351-353)
decimals() should be declared external:
        - ERC20.decimals() (Oracle.sol#368-370)
totalSupply() should be declared external:
        - ERC20.totalSupply() (Oracle.sol#375-377)
balanceOf(address) should be declared external:
        - ERC20.balanceOf(address) (Oracle.sol#382-384)
transfer(address,uint256) should be declared external:
        - ERC20.transfer(address,uint256) (Oracle.sol#394-398)
approve(address,uint256) should be declared external:
        - ERC20.approve(address,uint256) (Oracle.sol#417-421)
transferFrom(address,address,uint256) should be declared external:
        - ERC20.transferFrom(address,address,uint256) (Oracle.sol#439-448)
increaseAllowance(address,uint256) should be declared external:
        - ERC20.increaseAllowance(address,uint256) (Oracle.sol#462-466)
decreaseAllowance(address,uint256) should be declared external:
        - ERC20.decreaseAllowance(address,uint256) (Oracle.sol#482-491)
renounceOwnership() should be declared external:
        - Ownable.renounceOwnership() (Oracle.sol#914-916)
transferOwnership(address) should be declared external:
        - Ownable.transferOwnership(address) (Oracle.sol#922-925)
isOperator() should be declared external:
        - Operator.isOperator() (Oracle.sol#958-960)
transferOperator(address) should be declared external:
        - Operator.transferOperator(address) (Oracle.sol#962-964)
getCurrentEpoch() should be declared external:
        - Epoch.getCurrentEpoch() (Oracle.sol#1021-1023)
getPeriod() should be declared external:
        - Epoch.getPeriod() (Oracle.sol#1025-1027)
getStartTime() should be declared external:
        - Epoch.getStartTime() (Oracle.sol#1029-1031)
```



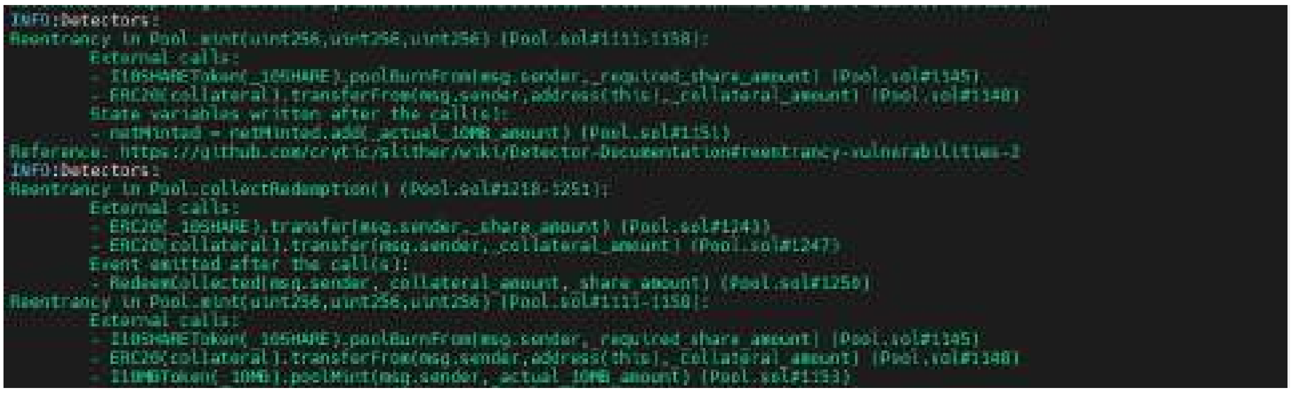
```
getStartTime() should be declared external:
        - Epoch.getStartTime() (Oracle.sol#1029-1031)
getLastEpochTime() should be declared external:
        - Epoch.getLastEpochTime() (Oracle.sol#1033-1035)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#public-function-that-could-be-declared-external
INFO:Slither:Oracle.sol analyzed (13 contracts with 75 detectors), 59 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.

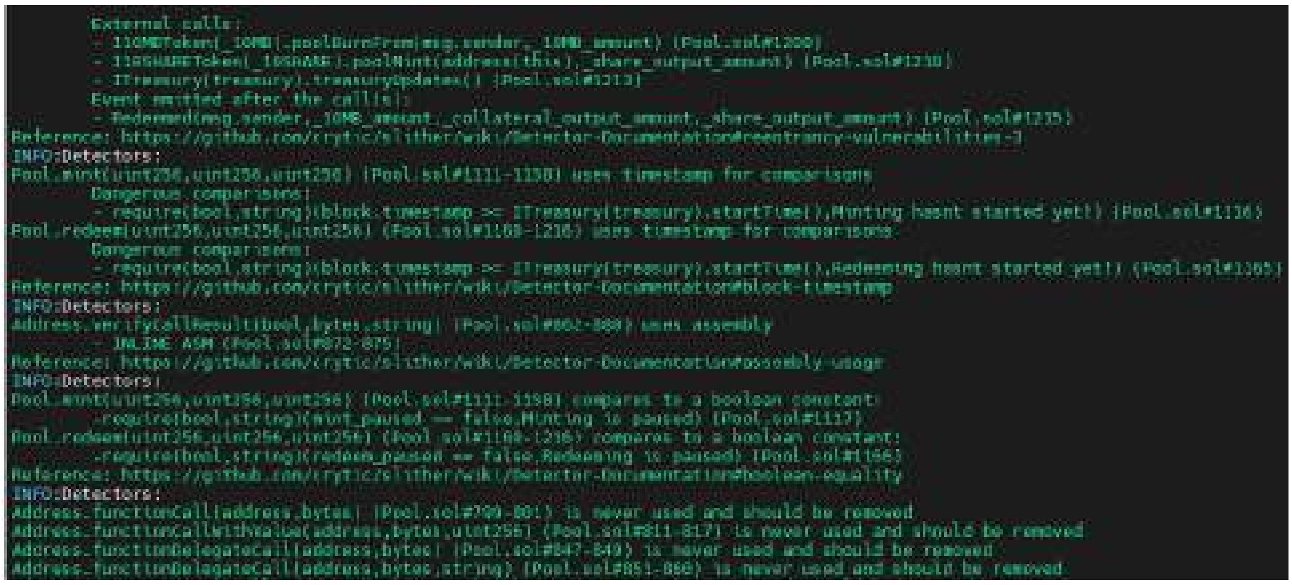
Slither log >> Pool.sol



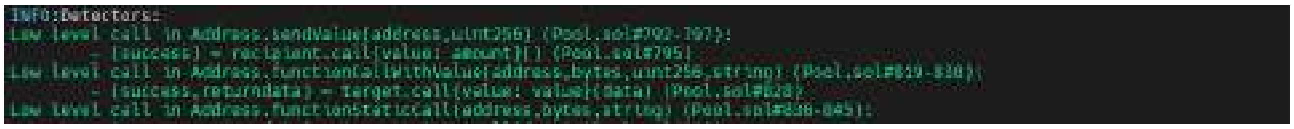
```
INFO:Detectors:
Pool.setPoolCeiling(uint256) (Pool.sol#1270-1272) should emit an event for:
        - pool_ceiling = _pool_ceiling (Pool.sol#1271)
Pool.setTwapPriceScalingPercentage(uint256) (Pool.sol#1274-1277) should emit an event for:
        - twapPriceScalingPercentage = _twapPriceScalingPercentage (Pool.sol#1276)
Pool.setRedemptionDelay(uint256) (Pool.sol#1279-1281) should emit an event for:
        - redemption_delay = _redemption_delay (Pool.sol#1280)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#missing-events-arithmetic
INFO:Detectors:
Pool.constructor(address,address,address,address,uint256).__10MB (Pool.sol#1047) lacks a zero-check on :
                - _10MB = __10MB (Pool.sol#1055)
Pool.constructor(address,address,address,address,uint256).__10SHARE (Pool.sol#1048) lacks a zero-check on :
                - _10SHARE = __10SHARE (Pool.sol#1056)
Pool.constructor(address,address,address,address,uint256)._collateral (Pool.sol#1049) lacks a zero-check on :
                - collateral = _collateral (Pool.sol#1057)
Pool.constructor(address,address,address,address,uint256)._treasury (Pool.sol#1050) lacks a zero-check on :
                - treasury = _treasury (Pool.sol#1058)
Pool.setTreasury(address)._treasury (Pool.sol#1283) lacks a zero-check on :
                - treasury = _treasury (Pool.sol#1285)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#missing-zero-address-validation
```



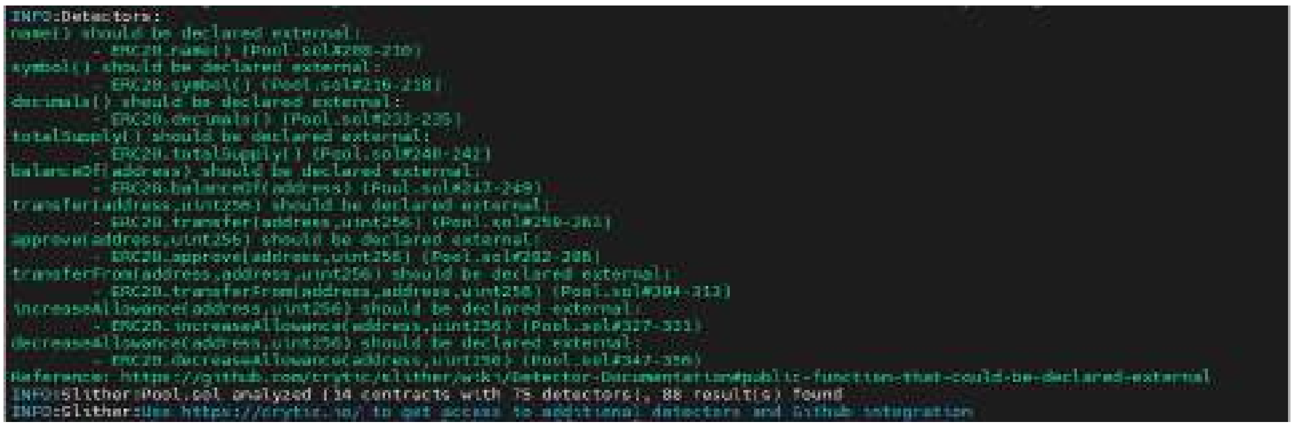
```
INFO:Detectors:
Reentrancy in Pool.mint(uint256,uint256,uint256) (Pool.sol#1111-1158):
        External calls:
        - I10SHAREToken(_10SHARE).poolBurnFrom(msg.sender,_required_share_amount) (Pool.sol#1145)
        - ERC20(collateral).transferFrom(msg.sender,address(this),_collateral_amount) (Pool.sol#1148)
        State variables written after the call(s):
        - netMinted = netMinted.add(_actual_10MB_amount) (Pool.sol#1151)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#reentrancy-vulnerabilities-2
INFO:Detectors:
Reentrancy in Pool.collectRedemption() (Pool.sol#1218-1251):
        External calls:
        - ERC20(_10SHARE).transfer(msg.sender,_share_amount) (Pool.sol#1243)
        - ERC20(collateral).transfer(msg.sender,_collateral_amount) (Pool.sol#1247)
        Event emitted after the call(s):
        - RedeemCollected(msg.sender,_collateral_amount,_share_amount) (Pool.sol#1250)
Reentrancy in Pool.mint(uint256,uint256,uint256) (Pool.sol#1111-1158):
        External calls:
        - I10SHAREToken(_10SHARE).poolBurnFrom(msg.sender,_required_share_amount) (Pool.sol#1145)
        - ERC20(collateral).transferFrom(msg.sender,address(this),_collateral_amount) (Pool.sol#1148)
        - I10MBToken(_10MB).poolMint(msg.sender,_actual_10MB_amount) (Pool.sol#1153)
```



```
        External calls:
        - I10MBToken(_10MB).poolBurnFrom(msg.sender,_10MB_amount) (Pool.sol#1200)
        - I10SHAREToken(_10SHARE).poolMint(address(this),_share_output_amount) (Pool.sol#1210)
        - ITreasury(treasury).treasuryUpdates() (Pool.sol#1213)
        Event emitted after the call(s):
        - Redeemed(msg.sender,_10MB_amount,_collateral_output_amount,_share_output_amount) (Pool.sol#1215)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#reentrancy-vulnerabilities-3
INFO:Detectors:
Pool.mint(uint256,uint256,uint256) (Pool.sol#1111-1158) uses timestamp for comparisons
        Dangerous comparisons:
        - require(bool,string)(block.timestamp >= ITreasury(treasury).startTime(),Minting hasnt started yet!) (Pool.sol#1116)
Pool.redeem(uint256,uint256,uint256) (Pool.sol#1160-1216) uses timestamp for comparisons
        Dangerous comparisons:
        - require(bool,string)(block.timestamp >= ITreasury(treasury).startTime(),Redeeming hasnt started yet!) (Pool.sol#1165)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#block-timestamp
INFO:Detectors:
Address.verifyCallResult(bool,bytes,string) (Pool.sol#862-880) uses assembly
        - INLINE ASM (Pool.sol#872-875)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#assembly-usage
INFO:Detectors:
Pool.mint(uint256,uint256,uint256) (Pool.sol#1111-1158) compares to a boolean constant:
        -require(bool,string)(mint_paused == false,Minting is paused) (Pool.sol#1117)
Pool.redeem(uint256,uint256,uint256) (Pool.sol#1160-1216) compares to a boolean constant:
        -require(bool,string)(redeem_paused == false,Redeeming is paused) (Pool.sol#1166)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#boolean-equality
INFO:Detectors:
Address.functionCall(address,bytes) (Pool.sol#799-801) is never used and should be removed
Address.functionCallWithValue(address,bytes,uint256) (Pool.sol#811-817) is never used and should be removed
Address.functionDelegateCall(address,bytes) (Pool.sol#847-849) is never used and should be removed
Address.functionDelegateCall(address,bytes,string) (Pool.sol#851-860) is never used and should be removed
```

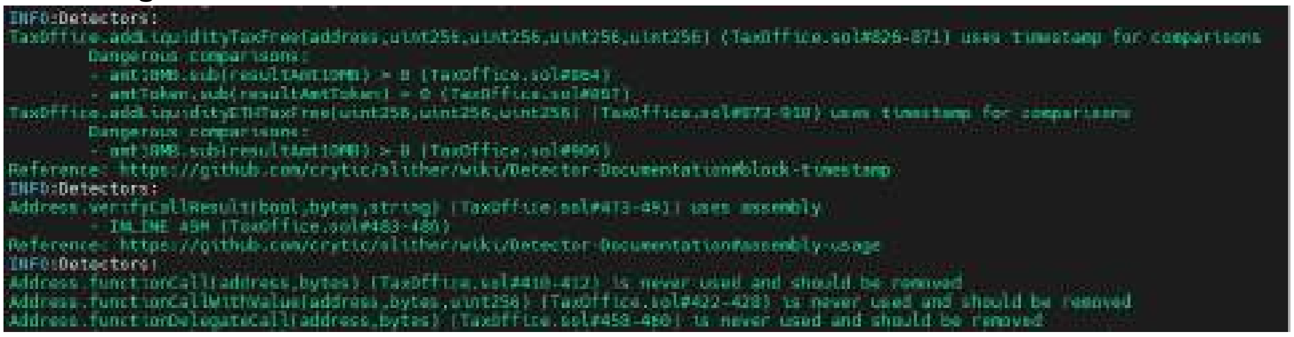


```
INFO:Detectors:
Low level call in Address.sendValue(address,uint256) (Pool.sol#792-797):
        - (success) = recipient.call{value: amount}() (Pool.sol#795)
Low level call in Address.functionCallWithValue(address,bytes,uint256,string) (Pool.sol#819-830):
        - (success,returndata) = target.call{value: value}(data) (Pool.sol#828)
Low level call in Address.functionStaticCall(address,bytes,string) (Pool.sol#838-845):
```

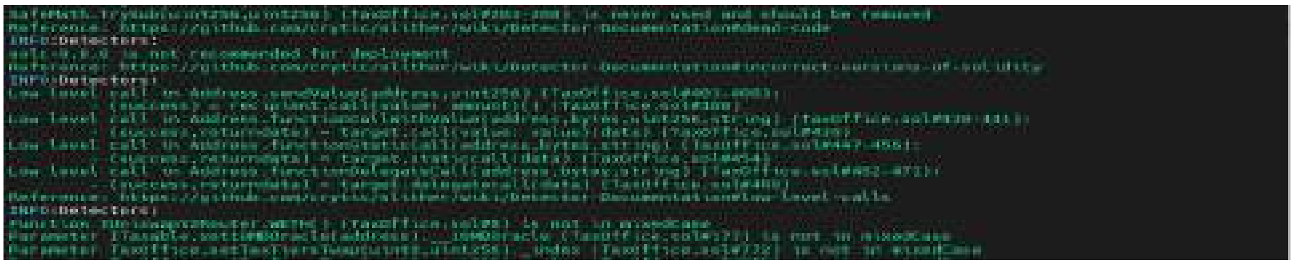


```
INFO:Detectors:
name() should be declared external:
        - ERC20.name() (Pool.sol#208-210)
symbol() should be declared external:
        - ERC20.symbol() (Pool.sol#216-218)
decimals() should be declared external:
        - ERC20.decimals() (Pool.sol#233-235)
totalSupply() should be declared external:
        - ERC20.totalSupply() (Pool.sol#240-242)
balanceOf(address) should be declared external:
        - ERC20.balanceOf(address) (Pool.sol#247-249)
transfer(address,uint256) should be declared external:
        - ERC20.transfer(address,uint256) (Pool.sol#259-262)
approve(address,uint256) should be declared external:
        - ERC20.approve(address,uint256) (Pool.sol#282-288)
transferFrom(address,address,uint256) should be declared external:
        - ERC20.transferFrom(address,address,uint256) (Pool.sol#304-313)
increaseAllowance(address,uint256) should be declared external:
        - ERC20.increaseAllowance(address,uint256) (Pool.sol#327-331)
decreaseAllowance(address,uint256) should be declared external:
        - ERC20.decreaseAllowance(address,uint256) (Pool.sol#347-356)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#public-function-that-could-be-declared-external
INFO:Slither:Pool.sol analyzed (14 contracts with 75 detectors), 88 result(s) found
INFO:Slither:Use https://crytic.io/ to get access to additional detectors and Github integration
```

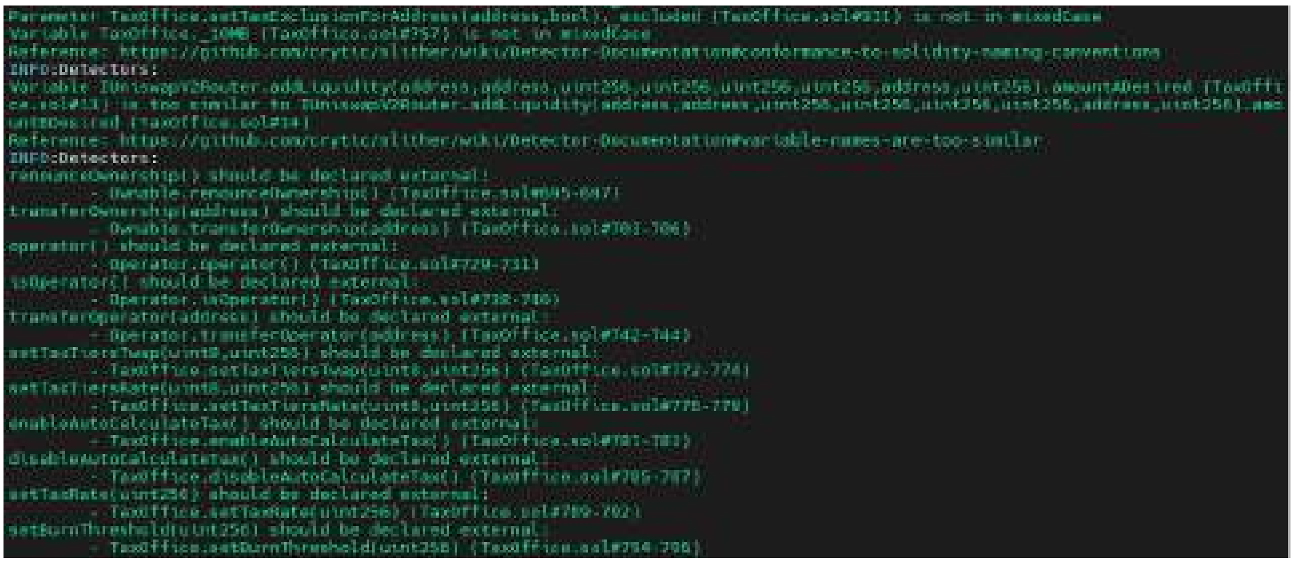
Slither log >> TaxOffice.sol



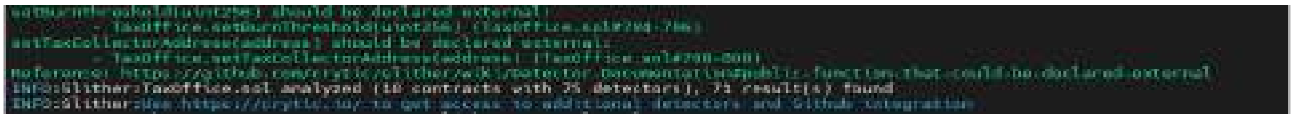
```
INFO:Detectors:
TaxOffice.addLiquidityTaxFree(address,uint256,uint256,uint256,uint256) (TaxOffice.sol#826-871) uses timestamp for comparisons
        Dangerous comparisons:
        - amt10MB.sub(resultAmt10MB) > 0 (TaxOffice.sol#864)
        - amtToken.sub(resultAmtToken) > 0 (TaxOffice.sol#867)
TaxOffice.addLiquidityETHTaxFree(uint256,uint256,uint256) (TaxOffice.sol#873-910) uses timestamp for comparisons
        Dangerous comparisons:
        - amt10MB.sub(resultAmt10MB) > 0 (TaxOffice.sol#906)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#block-timestamp
INFO:Detectors:
Address.verifyCallResult(bool,bytes,string) (TaxOffice.sol#473-491) uses assembly
        - INLINE ASM (TaxOffice.sol#483-486)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#assembly-usage
INFO:Detectors:
Address.functionCall(address,bytes) (TaxOffice.sol#410-412) is never used and should be removed
Address.functionCallWithValue(address,bytes,uint256) (TaxOffice.sol#422-428) is never used and should be removed
Address.functionDelegateCall(address,bytes) (TaxOffice.sol#458-460) is never used and should be removed
```



```
SafeMath.trySub(uint256,uint256) (TaxOffice.sol#[illegible]) is never used and should be removed
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#dead-code
INFO:Detectors:
solc-0.8.0 is not recommended for deployment
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#incorrect-versions-of-solidity
INFO:Detectors:
Low level call in Address.sendValue(address,uint256) (TaxOffice.sol#[illegible]):
        - (success) = recipient.call{value: amount}() (TaxOffice.sol#[illegible])
Low level call in Address.functionCallWithValue(address,bytes,uint256,string) (TaxOffice.sol#[illegible]):
        - (success,returndata) = target.call{value: value}(data) (TaxOffice.sol#[illegible])
Low level call in Address.functionStaticCall(address,bytes,string) (TaxOffice.sol#447-456):
        - (success,returndata) = target.staticcall(data) (TaxOffice.sol#454)
Low level call in Address.functionDelegateCall(address,bytes,string) (TaxOffice.sol#[illegible]):
        - (success,returndata) = target.delegatecall(data) (TaxOffice.sol#[illegible])
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#low-level-calls
INFO:Detectors:
Function IUniswapV2Router.WETH() (TaxOffice.sol#8) is not in mixedCase
Parameter ITaxable.setTaxOracle(address).__10MBOracle (TaxOffice.sol#[illegible]) is not in mixedCase
Parameter TaxOffice.setTaxTiersTwap(uint8,uint256)._index (TaxOffice.sol#772) is not in mixedCase
```

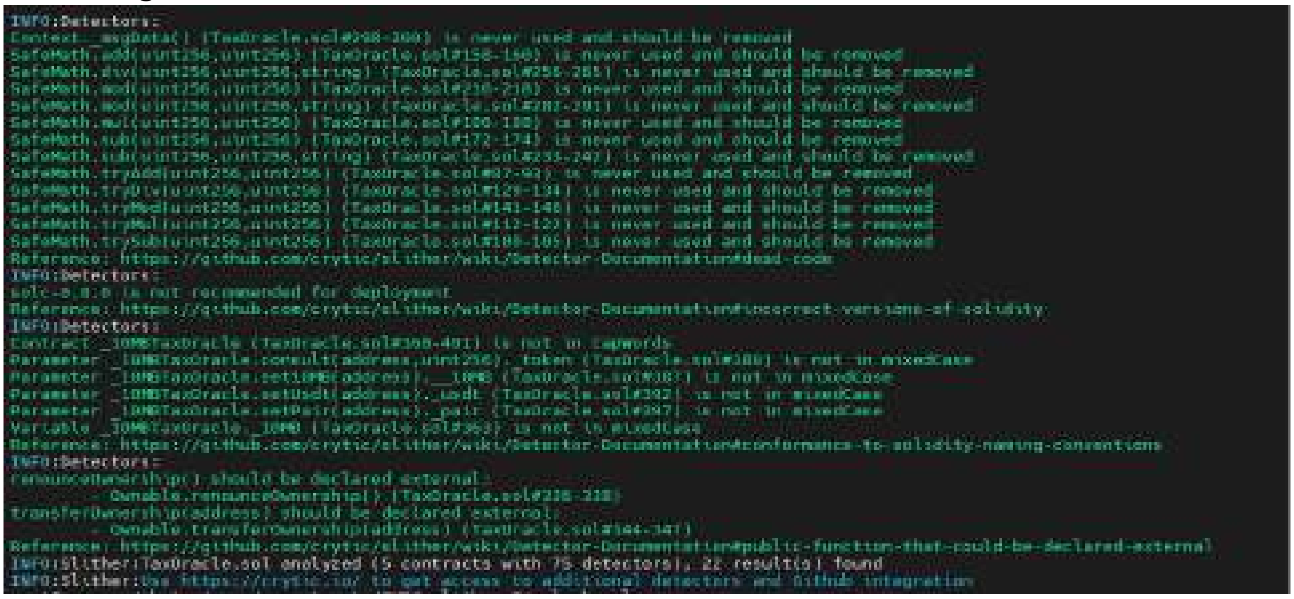


```
Parameter TaxOffice.setTaxExclusionForAddress(address,bool)._excluded (TaxOffice.sol#831) is not in mixedCase
Variable TaxOffice._10MB (TaxOffice.sol#757) is not in mixedCase
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#conformance-to-solidity-naming-conventions
INFO:Detectors:
Variable IUniswapV2Router.addLiquidity(address,address,uint256,uint256,uint256,uint256,address,uint256).amountADesired (TaxOffi
ce.sol#13) is too similar to IUniswapV2Router.addLiquidity(address,address,uint256,uint256,uint256,uint256,address,uint256).amo
untBDesired (TaxOffice.sol#14)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#variable-names-are-too-similar
INFO:Detectors:
renounceOwnership() should be declared external:
        - Ownable.renounceOwnership() (TaxOffice.sol#695-697)
transferOwnership(address) should be declared external:
        - Ownable.transferOwnership(address) (TaxOffice.sol#703-706)
operator() should be declared external:
        - Operator.operator() (TaxOffice.sol#729-731)
isOperator() should be declared external:
        - Operator.isOperator() (TaxOffice.sol#738-740)
transferOperator(address) should be declared external:
        - Operator.transferOperator(address) (TaxOffice.sol#742-744)
setTaxTiersTwap(uint8,uint256) should be declared external:
        - TaxOffice.setTaxTiersTwap(uint8,uint256) (TaxOffice.sol#772-774)
setTaxTiersRate(uint8,uint256) should be declared external:
        - TaxOffice.setTaxTiersRate(uint8,uint256) (TaxOffice.sol#776-778)
enableAutoCalculateTax() should be declared external:
        - TaxOffice.enableAutoCalculateTax() (TaxOffice.sol#781-783)
disableAutoCalculateTax() should be declared external:
        - TaxOffice.disableAutoCalculateTax() (TaxOffice.sol#785-787)
setTaxRate(uint256) should be declared external:
        - TaxOffice.setTaxRate(uint256) (TaxOffice.sol#789-792)
setBurnThreshold(uint256) should be declared external:
        - TaxOffice.setBurnThreshold(uint256) (TaxOffice.sol#794-796)
```



```
setBurnThreshold(uint256) should be declared external:
        - TaxOffice.setBurnThreshold(uint256) (TaxOffice.sol#794-796)
setTaxCollectorAddress(address) should be declared external:
        - TaxOffice.setTaxCollectorAddress(address) (TaxOffice.sol#798-800)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#public-function-that-could-be-declared-external
INFO:Slither:TaxOffice.sol analyzed (10 contracts with 75 detectors), 71 result(s) found
INFO:Slither:Use https://crytic.io/ to get access to additional detectors and Github integration
```

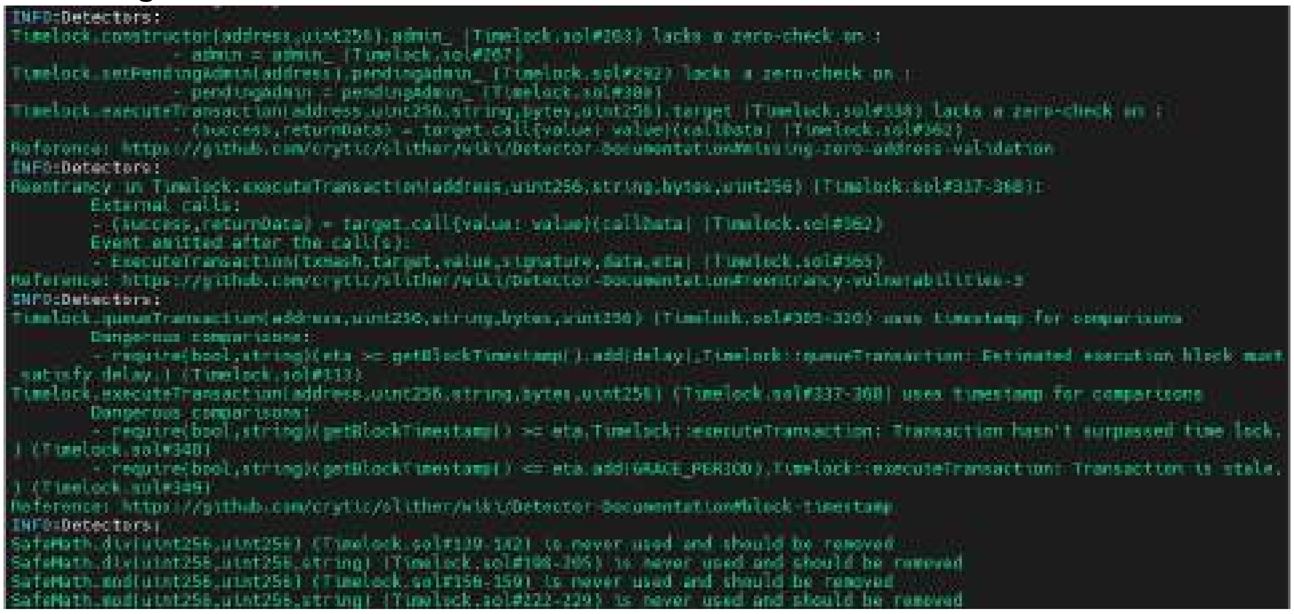
Slither log >> TaxOracle.sol



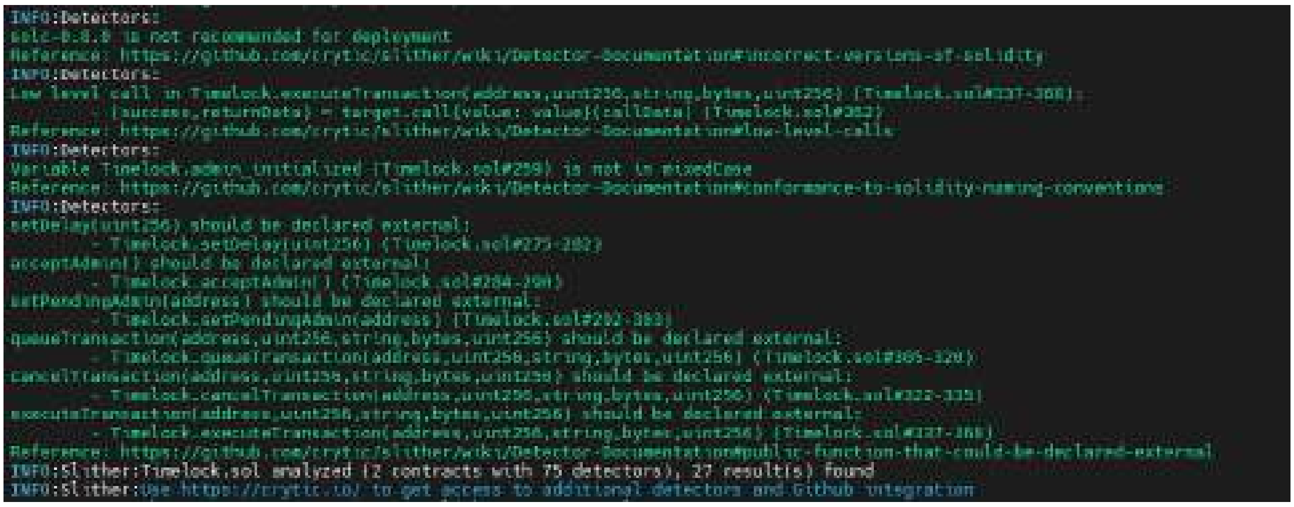
```
INFO:Detectors:
Context._msgData() (TaxOracle.sol#298-300) is never used and should be removed
SafeMath.add(uint256,uint256) (TaxOracle.sol#158-160) is never used and should be removed
SafeMath.div(uint256,uint256,string) (TaxOracle.sol#256-265) is never used and should be removed
SafeMath.mod(uint256,uint256) (TaxOracle.sol#216-218) is never used and should be removed
SafeMath.mod(uint256,uint256,string) (TaxOracle.sol#282-291) is never used and should be removed
SafeMath.mul(uint256,uint256) (TaxOracle.sol#186-188) is never used and should be removed
SafeMath.sub(uint256,uint256) (TaxOracle.sol#172-174) is never used and should be removed
SafeMath.sub(uint256,uint256,string) (TaxOracle.sol#233-242) is never used and should be removed
SafeMath.tryAdd(uint256,uint256) (TaxOracle.sol#87-93) is never used and should be removed
SafeMath.tryDiv(uint256,uint256) (TaxOracle.sol#129-134) is never used and should be removed
SafeMath.tryMod(uint256,uint256) (TaxOracle.sol#141-146) is never used and should be removed
SafeMath.tryMul(uint256,uint256) (TaxOracle.sol#112-122) is never used and should be removed
SafeMath.trySub(uint256,uint256) (TaxOracle.sol#100-105) is never used and should be removed
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#dead-code
INFO:Detectors:
solc-0.8.0 is not recommended for deployment
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#incorrect-versions-of-solidity
INFO:Detectors:
Contract _10MBTaxOracle (TaxOracle.sol#358-401) is not in CapWords
Parameter _10MBTaxOracle.consult(address,uint256)._token (TaxOracle.sol#380) is not in mixedCase
Parameter _10MBTaxOracle.set10MB(address).__10MB (TaxOracle.sol#387) is not in mixedCase
Parameter _10MBTaxOracle.setUsdt(address)._usdt (TaxOracle.sol#392) is not in mixedCase
Parameter _10MBTaxOracle.setPair(address)._pair (TaxOracle.sol#397) is not in mixedCase
Variable _10MBTaxOracle._10MB (TaxOracle.sol#363) is not in mixedCase
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#conformance-to-solidity-naming-conventions
INFO:Detectors:
renounceOwnership() should be declared external:
        - Ownable.renounceOwnership() (TaxOracle.sol#226-228)
transferOwnership(address) should be declared external:
        - Ownable.transferOwnership(address) (TaxOracle.sol#344-347)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#public-function-that-could-be-declared-external
INFO:Slither:TaxOracle.sol analyzed (5 contracts with 75 detectors), 22 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.

Slither log >> Timelock.sol

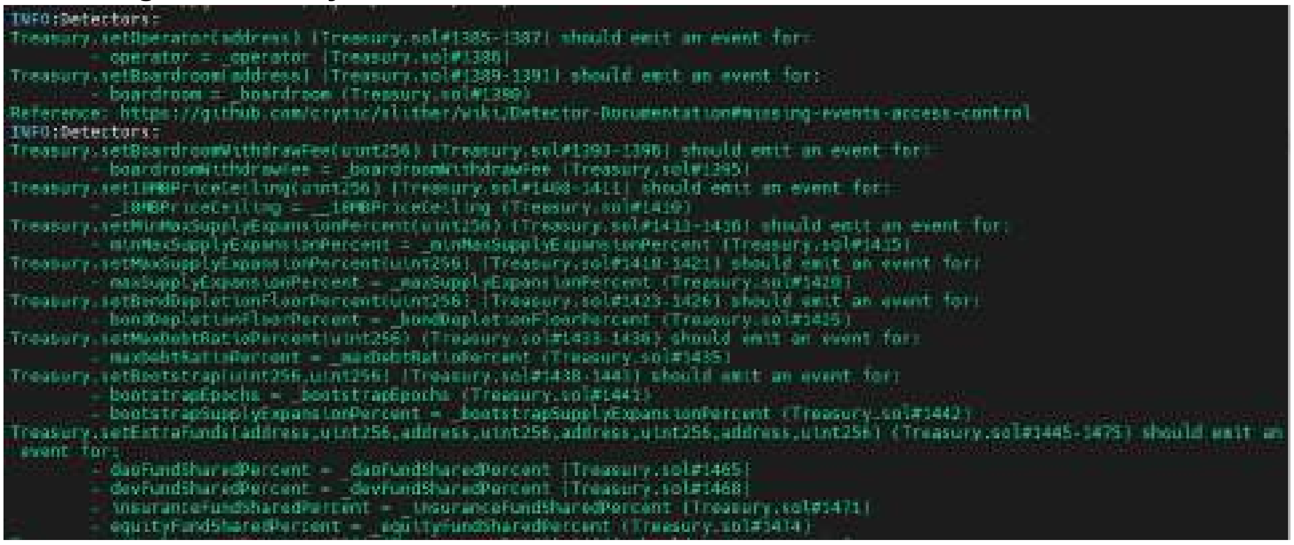


```
INFO:Detectors:
Timelock.constructor(address,uint256).admin_ (Timelock.sol#263) lacks a zero-check on :
                - admin = admin_ (Timelock.sol#267)
Timelock.setPendingAdmin(address).pendingAdmin_ (Timelock.sol#292) lacks a zero-check on :
                - pendingAdmin = pendingAdmin_ (Timelock.sol#300)
Timelock.executeTransaction(address,uint256,string,bytes,uint256).target (Timelock.sol#338) lacks a zero-check on :
                - (success,returnData) = target.call{value: value}(callData) (Timelock.sol#362)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#missing-zero-address-validation
INFO:Detectors:
Reentrancy in Timelock.executeTransaction(address,uint256,string,bytes,uint256) (Timelock.sol#337-368):
        External calls:
        - (success,returnData) = target.call{value: value}(callData) (Timelock.sol#362)
        Event emitted after the call(s):
        - ExecuteTransaction(txHash,target,value,signature,data,eta) (Timelock.sol#365)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#reentrancy-vulnerabilities-3
INFO:Detectors:
Timelock.queueTransaction(address,uint256,string,bytes,uint256) (Timelock.sol#305-320) uses timestamp for comparisons
        Dangerous comparisons:
        - require(bool,string)(eta >= getBlockTimestamp().add(delay),Timelock::queueTransaction: Estimated execution block must
 satisfy delay.) (Timelock.sol#313)
Timelock.executeTransaction(address,uint256,string,bytes,uint256) (Timelock.sol#337-368) uses timestamp for comparisons
        Dangerous comparisons:
        - require(bool,string)(getBlockTimestamp() >= eta,Timelock::executeTransaction: Transaction hasn't surpassed time lock.
) (Timelock.sol#348)
        - require(bool,string)(getBlockTimestamp() <= eta.add(GRACE_PERIOD),Timelock::executeTransaction: Transaction is stale.
) (Timelock.sol#349)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#block-timestamp
INFO:Detectors:
SafeMath.div(uint256,uint256) (Timelock.sol#139-142) is never used and should be removed
SafeMath.div(uint256,uint256,string) (Timelock.sol#198-205) is never used and should be removed
SafeMath.mod(uint256,uint256) (Timelock.sol#156-159) is never used and should be removed
SafeMath.mod(uint256,uint256,string) (Timelock.sol#222-229) is never used and should be removed
```

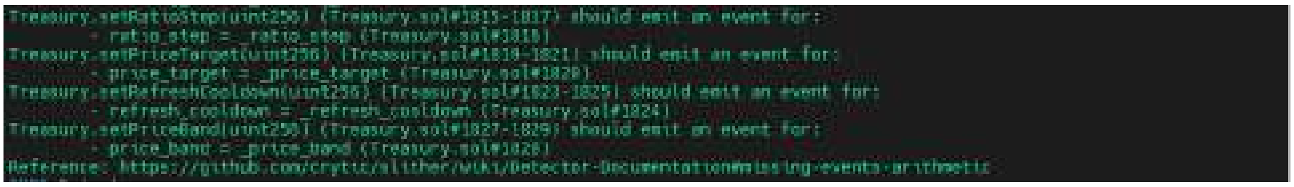


```
INFO:Detectors:
solc-0.8.0 is not recommended for deployment
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#incorrect-versions-of-solidity
INFO:Detectors:
Low level call in Timelock.executeTransaction(address,uint256,string,bytes,uint256) (Timelock.sol#337-368):
        - (success,returnData) = target.call{value: value}(callData) (Timelock.sol#362)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#low-level-calls
INFO:Detectors:
Variable Timelock.admin_initialized (Timelock.sol#259) is not in mixedCase
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#conformance-to-solidity-naming-conventions
INFO:Detectors:
setDelay(uint256) should be declared external:
        - Timelock.setDelay(uint256) (Timelock.sol#275-282)
acceptAdmin() should be declared external:
        - Timelock.acceptAdmin() (Timelock.sol#284-290)
setPendingAdmin(address) should be declared external:
        - Timelock.setPendingAdmin(address) (Timelock.sol#292-303)
queueTransaction(address,uint256,string,bytes,uint256) should be declared external:
        - Timelock.queueTransaction(address,uint256,string,bytes,uint256) (Timelock.sol#305-320)
cancelTransaction(address,uint256,string,bytes,uint256) should be declared external:
        - Timelock.cancelTransaction(address,uint256,string,bytes,uint256) (Timelock.sol#322-335)
executeTransaction(address,uint256,string,bytes,uint256) should be declared external:
        - Timelock.executeTransaction(address,uint256,string,bytes,uint256) (Timelock.sol#337-368)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#public-function-that-could-be-declared-external
INFO:Slither:Timelock.sol analyzed (2 contracts with 75 detectors), 27 result(s) found
INFO:Slither:Use https://crytic.io/ to get access to additional detectors and Github integration
```

Slither log >> Treasury.sol

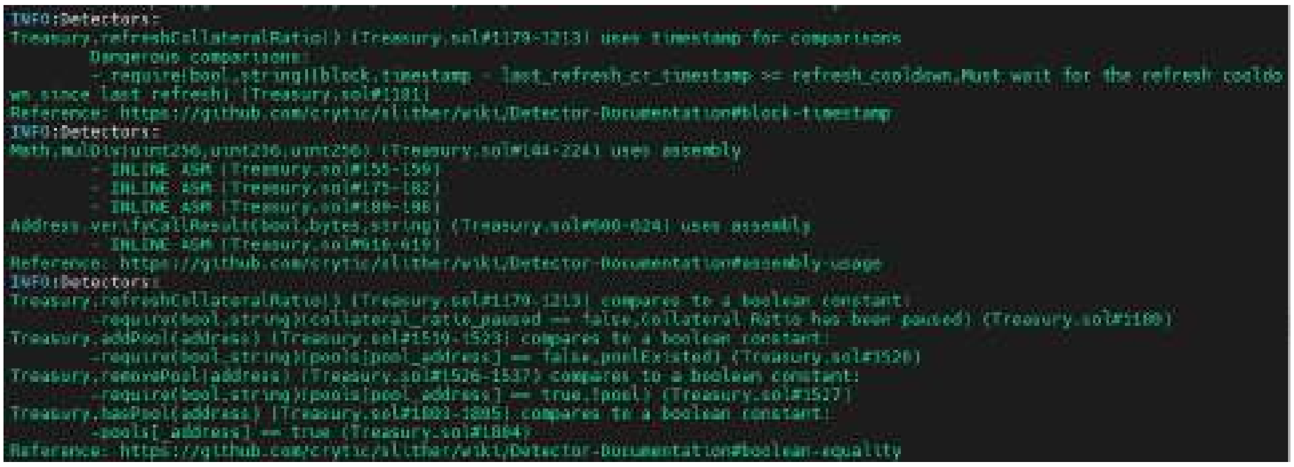


```
INFO:Detectors:
Treasury.setOperator(address) (Treasury.sol#1385-1387) should emit an event for:
        - operator = _operator (Treasury.sol#1386)
Treasury.setBoardroom(address) (Treasury.sol#1389-1391) should emit an event for:
        - boardroom = _boardroom (Treasury.sol#1390)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#missing-events-access-control
INFO:Detectors:
Treasury.setBoardroomWithdrawFee(uint256) (Treasury.sol#1393-1396) should emit an event for:
        - boardroomWithdrawFee = _boardroomWithdrawFee (Treasury.sol#1395)
Treasury.set10MBPriceCeiling(uint256) (Treasury.sol#1408-1411) should emit an event for:
        - _10MBPriceCeiling = __10MBPriceCeiling (Treasury.sol#1410)
Treasury.setMinMaxSupplyExpansionPercent(uint256) (Treasury.sol#1413-1416) should emit an event for:
        - minMaxSupplyExpansionPercent = _minMaxSupplyExpansionPercent (Treasury.sol#1415)
Treasury.setMaxSupplyExpansionPercent(uint256) (Treasury.sol#1418-1421) should emit an event for:
        - maxSupplyExpansionPercent = _maxSupplyExpansionPercent (Treasury.sol#1420)
Treasury.setBondDepletionFloorPercent(uint256) (Treasury.sol#1423-1426) should emit an event for:
        - bondDepletionFloorPercent = _bondDepletionFloorPercent (Treasury.sol#1425)
Treasury.setMaxDebtRatioPercent(uint256) (Treasury.sol#1433-1436) should emit an event for:
        - maxDebtRatioPercent = _maxDebtRatioPercent (Treasury.sol#1435)
Treasury.setBootstrap(uint256,uint256) (Treasury.sol#1438-1443) should emit an event for:
        - bootstrapEpochs = _bootstrapEpochs (Treasury.sol#1441)
        - bootstrapSupplyExpansionPercent = _bootstrapSupplyExpansionPercent (Treasury.sol#1442)
Treasury.setExtraFunds(address,uint256,address,uint256,address,uint256,address,uint256) (Treasury.sol#1445-1475) should emit an
 event for:
        - daoFundSharedPercent = _daoFundSharedPercent (Treasury.sol#1465)
        - devFundSharedPercent = _devFundSharedPercent (Treasury.sol#1468)
        - insuranceFundSharedPercent = _insuranceFundSharedPercent (Treasury.sol#1471)
        - equityFundSharedPercent = _equityFundSharedPercent (Treasury.sol#1474)
```

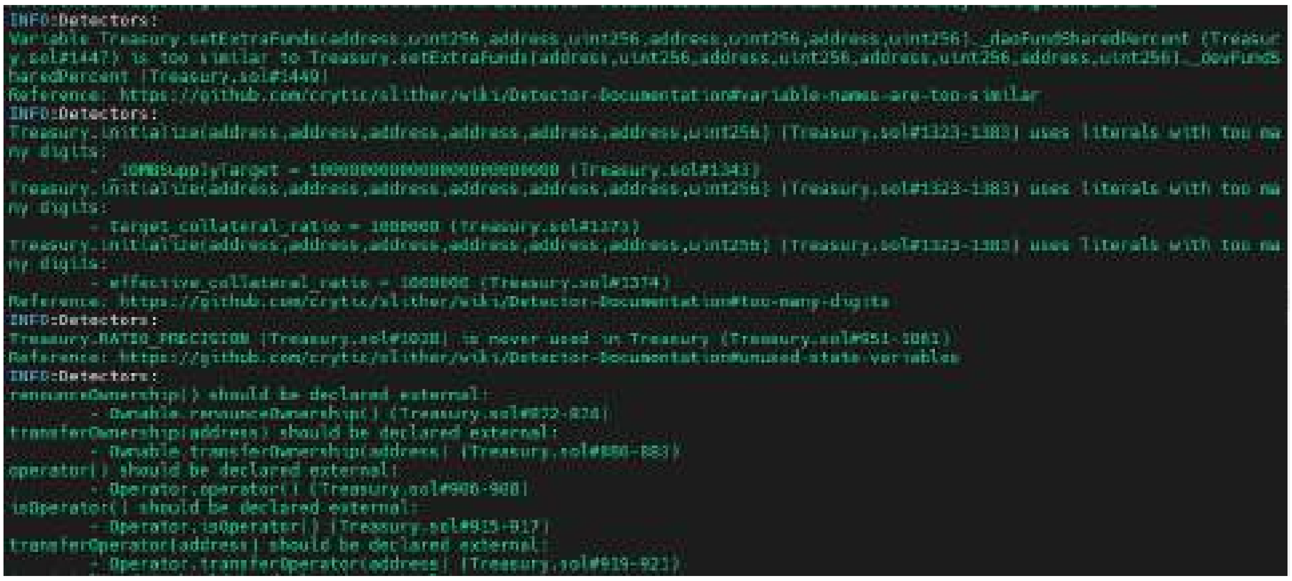


```
Treasury.setRatioStep(uint256) (Treasury.sol#1815-1817) should emit an event for:
        - ratio_step = _ratio_step (Treasury.sol#1816)
Treasury.setPriceTarget(uint256) (Treasury.sol#1819-1821) should emit an event for:
        - price_target = _price_target (Treasury.sol#1820)
Treasury.setRefreshCooldown(uint256) (Treasury.sol#1823-1825) should emit an event for:
        - refresh_cooldown = _refresh_cooldown (Treasury.sol#1824)
Treasury.setPriceBand(uint256) (Treasury.sol#1827-1829) should emit an event for:
        - price_band = _price_band (Treasury.sol#1828)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#missing-events-arithmetic
```

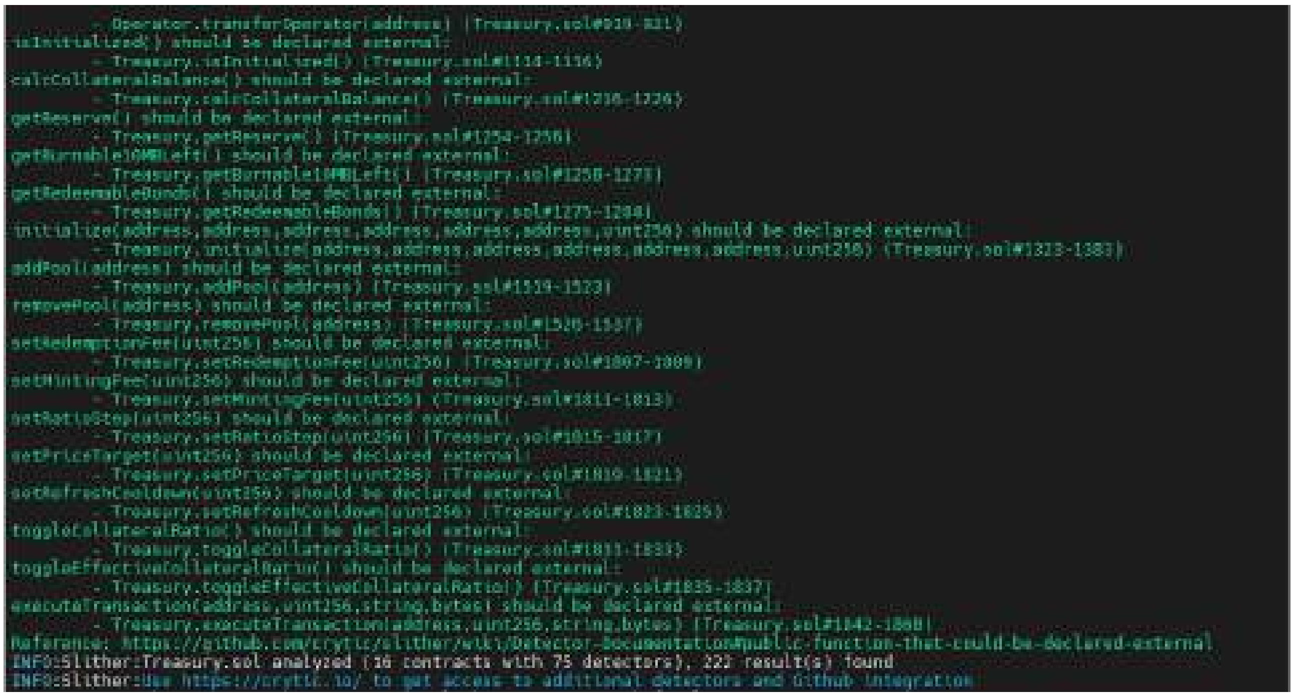
This is a private and confidential document. No part of this document should
be disclosed to third party without prior written permission of EtherAuthority.
Email: audit@EtherAuthority.io



```
INFO:Detectors:
Treasury.refreshCollateralRatio() (Treasury.sol#1179-1213) uses timestamp for comparisons
        Dangerous comparisons:
        - require(bool,string)(block.timestamp - last_refresh_cr_timestamp >= refresh_cooldown,Must wait for the refresh cooldo
wn since last refresh) (Treasury.sol#1181)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#block-timestamp
INFO:Detectors:
Math.mulDiv(uint256,uint256,uint256) (Treasury.sol#144-224) uses assembly
        - INLINE ASM (Treasury.sol#155-159)
        - INLINE ASM (Treasury.sol#175-182)
        - INLINE ASM (Treasury.sol#188-196)
Address.verifyCallResult(bool,bytes,string) (Treasury.sol#600-624) uses assembly
        - INLINE ASM (Treasury.sol#616-619)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#assembly-usage
INFO:Detectors:
Treasury.refreshCollateralRatio() (Treasury.sol#1179-1213) compares to a boolean constant:
        -require(bool,string)(collateral_ratio_paused == false,Collateral Ratio has been paused) (Treasury.sol#1180)
Treasury.addPool(address) (Treasury.sol#1519-1523) compares to a boolean constant:
        -require(bool,string)(pools[pool_address] == false,poolExisted) (Treasury.sol#1520)
Treasury.removePool(address) (Treasury.sol#1526-1537) compares to a boolean constant:
        -require(bool,string)(pools[pool_address] == true,!pool) (Treasury.sol#1527)
Treasury.hasPool(address) (Treasury.sol#1803-1805) compares to a boolean constant:
        -pools[_address] == true (Treasury.sol#1804)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#boolean-equality
```

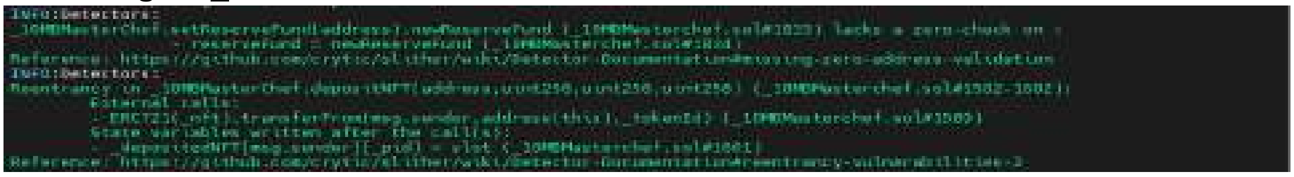


```
INFO:Detectors:
Variable Treasury.setExtraFunds(address,uint256,address,uint256,address,uint256,address,uint256)._daoFundSharedPercent (Treasur
y.sol#1447) is too similar to Treasury.setExtraFunds(address,uint256,address,uint256,address,uint256,address,uint256)._devFundS
haredPercent (Treasury.sol#1449)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#variable-names-are-too-similar
INFO:Detectors:
Treasury.initialize(address,address,address,address,address,address,uint256) (Treasury.sol#1323-1383) uses literals with too ma
ny digits:
        - _10MBSupplyTarget = 1000000000000000000000000 (Treasury.sol#1343)
Treasury.initialize(address,address,address,address,address,address,uint256) (Treasury.sol#1323-1383) uses literals with too ma
ny digits:
        - target_collateral_ratio = 1000000 (Treasury.sol#1373)
Treasury.initialize(address,address,address,address,address,address,uint256) (Treasury.sol#1323-1383) uses literals with too ma
ny digits:
        - effective_collateral_ratio = 1000000 (Treasury.sol#1374)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#too-many-digits
INFO:Detectors:
Treasury.RATIO_PRECISION (Treasury.sol#1028) is never used in Treasury (Treasury.sol#951-1861)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#unused-state-variables
INFO:Detectors:
renounceOwnership() should be declared external:
        - Ownable.renounceOwnership() (Treasury.sol#872-874)
transferOwnership(address) should be declared external:
        - Ownable.transferOwnership(address) (Treasury.sol#880-883)
operator() should be declared external:
        - Operator.operator() (Treasury.sol#906-908)
isOperator() should be declared external:
        - Operator.isOperator() (Treasury.sol#915-917)
transferOperator(address) should be declared external:
        - Operator.transferOperator(address) (Treasury.sol#919-921)
```



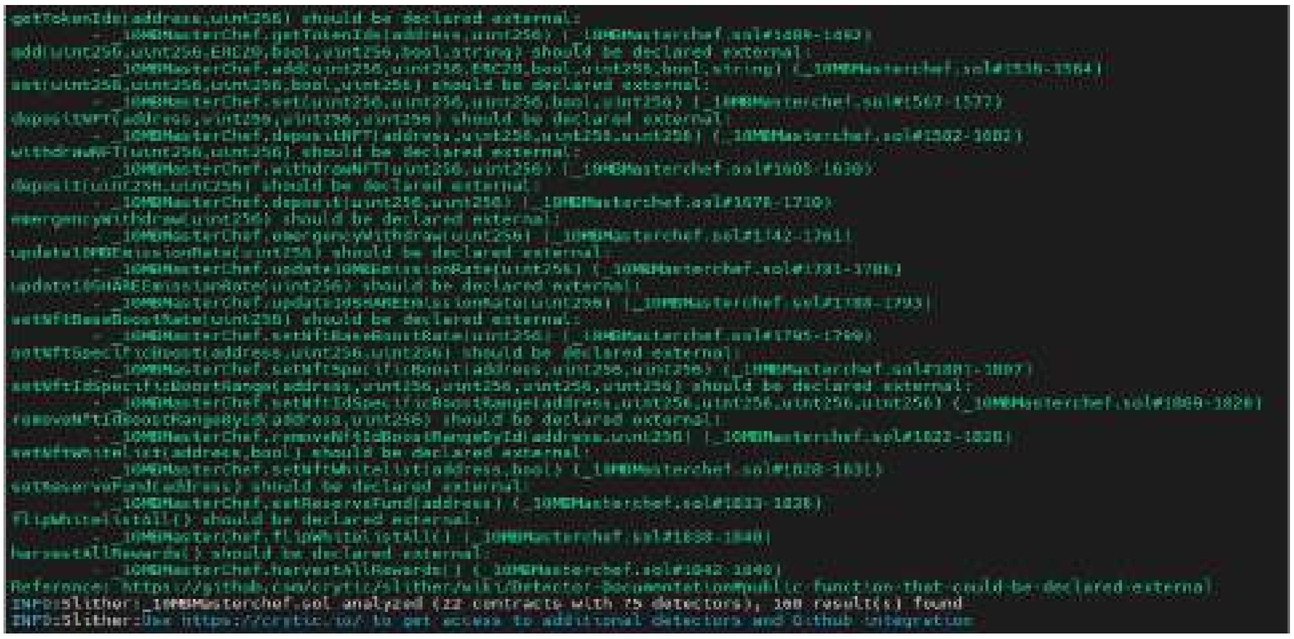
```
        - Operator.transferOperator(address) (Treasury.sol#919-921)
isInitialized() should be declared external:
        - Treasury.isInitialized() (Treasury.sol#1114-1116)
calcCollateralBalance() should be declared external:
        - Treasury.calcCollateralBalance() (Treasury.sol#1216-1226)
getReserve() should be declared external:
        - Treasury.getReserve() (Treasury.sol#1254-1256)
getBurnable10MBLeft() should be declared external:
        - Treasury.getBurnable10MBLeft() (Treasury.sol#1258-1273)
getRedeemableBonds() should be declared external:
        - Treasury.getRedeemableBonds() (Treasury.sol#1275-1284)
initialize(address,address,address,address,address,address,uint256) should be declared external:
        - Treasury.initialize(address,address,address,address,address,address,uint256) (Treasury.sol#1323-1383)
addPool(address) should be declared external:
        - Treasury.addPool(address) (Treasury.sol#1519-1523)
removePool(address) should be declared external:
        - Treasury.removePool(address) (Treasury.sol#1526-1537)
setRedemptionFee(uint256) should be declared external:
        - Treasury.setRedemptionFee(uint256) (Treasury.sol#1807-1809)
setMintingFee(uint256) should be declared external:
        - Treasury.setMintingFee(uint256) (Treasury.sol#1811-1813)
setRatioStep(uint256) should be declared external:
        - Treasury.setRatioStep(uint256) (Treasury.sol#1815-1817)
setPriceTarget(uint256) should be declared external:
        - Treasury.setPriceTarget(uint256) (Treasury.sol#1819-1821)
setRefreshCooldown(uint256) should be declared external:
        - Treasury.setRefreshCooldown(uint256) (Treasury.sol#1823-1825)
toggleCollateralRatio() should be declared external:
        - Treasury.toggleCollateralRatio() (Treasury.sol#1831-1833)
toggleEffectiveCollateralRatio() should be declared external:
        - Treasury.toggleEffectiveCollateralRatio() (Treasury.sol#1835-1837)
executeTransaction(address,uint256,string,bytes) should be declared external:
        - Treasury.executeTransaction(address,uint256,string,bytes) (Treasury.sol#1842-1860)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#public-function-that-could-be-declared-external
INFO:Slither:Treasury.sol analyzed (16 contracts with 75 detectors), 222 result(s) found
INFO:Slither:Use https://crytic.io/ to get access to additional detectors and Github integration
```

Slither log >> _10MBMasterchef.sol

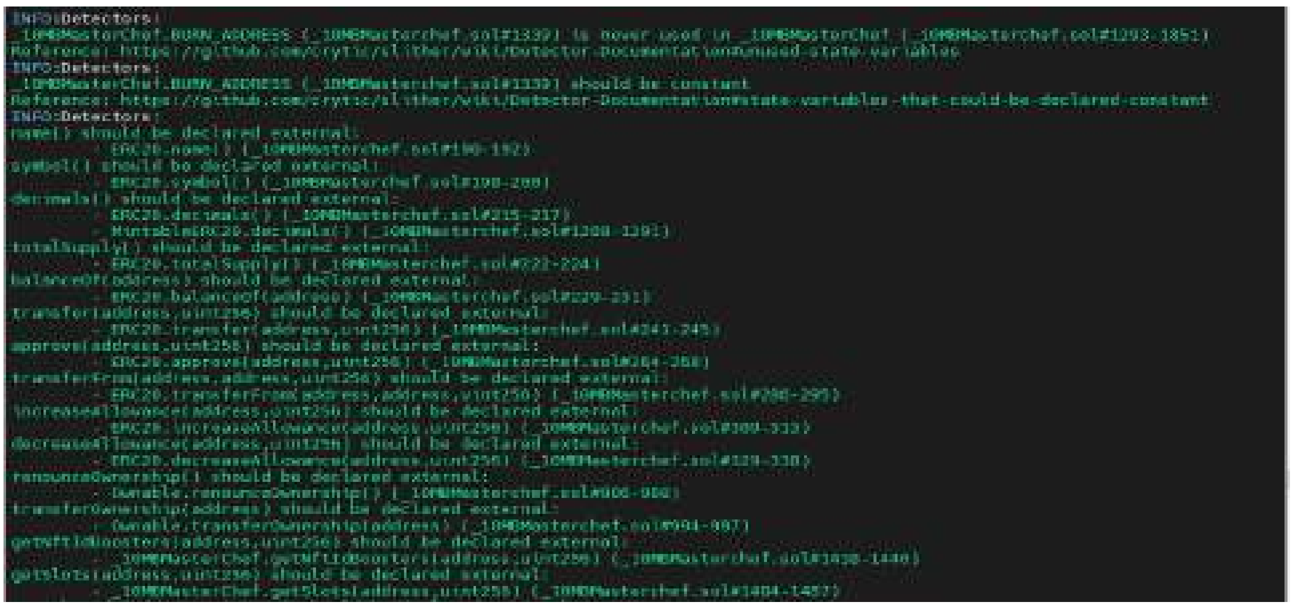


```
INFO:Detectors:
_10MBMasterChef.setReserveFund(address).newReserveFund (_10MBMasterchef.sol#1023) lacks a zero-check on :
                - reserveFund = newReserveFund (_10MBMasterchef.sol#1024)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#missing-zero-address-validation
INFO:Detectors:
Reentrancy in _10MBMasterChef.depositNFT(address,uint256,uint256,uint256) (_10MBMasterchef.sol#1582-1602):
        External calls:
        - ERC721(_nft).transferFrom(msg.sender,address(this),_tokenId) (_10MBMasterchef.sol#1589)
        State variables written after the call(s):
        - _depositedNFT[msg.sender][_pid] = slot (_10MBMasterchef.sol#1601)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#reentrancy-vulnerabilities-2
```

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

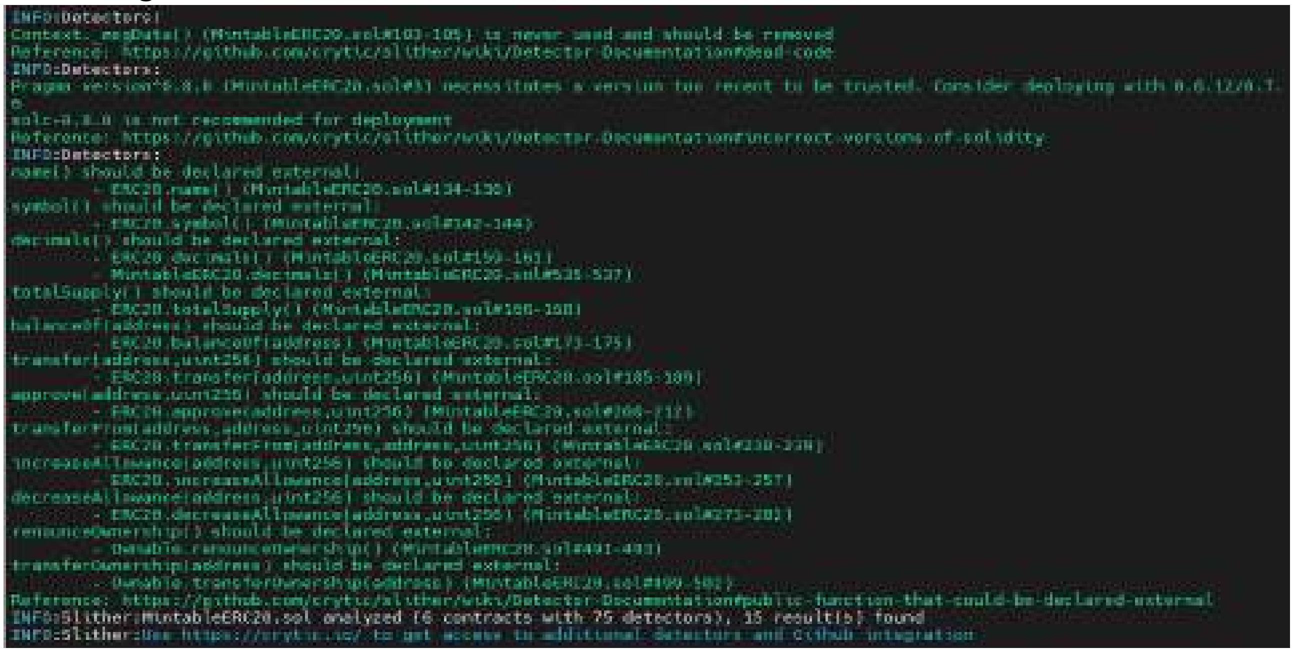


```
getTokenIds(address,uint256) should be declared external:
        - _10MBMasterChef.getTokenIds(address,uint256) (_10MBMasterchef.sol#1484-1492)
add(uint256,uint256,ERC20,bool,uint256,bool,string) should be declared external:
        - _10MBMasterChef.add(uint256,uint256,ERC20,bool,uint256,bool,string) (_10MBMasterchef.sol#1536-1564)
set(uint256,uint256,uint256,bool,uint256) should be declared external:
        - _10MBMasterChef.set(uint256,uint256,uint256,bool,uint256) (_10MBMasterchef.sol#1567-1577)
depositNFT(address,uint256,uint256,uint256) should be declared external:
        - _10MBMasterChef.depositNFT(address,uint256,uint256,uint256) (_10MBMasterchef.sol#1582-1602)
withdrawNFT(uint256,uint256) should be declared external:
        - _10MBMasterChef.withdrawNFT(uint256,uint256) (_10MBMasterchef.sol#1605-1630)
deposit(uint256,uint256) should be declared external:
        - _10MBMasterChef.deposit(uint256,uint256) (_10MBMasterchef.sol#1678-1710)
emergencyWithdraw(uint256) should be declared external:
        - _10MBMasterChef.emergencyWithdraw(uint256) (_10MBMasterchef.sol#1742-1761)
update10MBEmissionRate(uint256) should be declared external:
        - _10MBMasterChef.update10MBEmissionRate(uint256) (_10MBMasterchef.sol#1781-1786)
update10SHAREEmissionRate(uint256) should be declared external:
        - _10MBMasterChef.update10SHAREEmissionRate(uint256) (_10MBMasterchef.sol#1788-1793)
setNftBaseBoostRate(uint256) should be declared external:
        - _10MBMasterChef.setNftBaseBoostRate(uint256) (_10MBMasterchef.sol#1795-1799)
setNftSpecificBoost(address,uint256,uint256) should be declared external:
        - _10MBMasterChef.setNftSpecificBoost(address,uint256,uint256) (_10MBMasterchef.sol#1801-1807)
setNftIdSpecificBoostRange(address,uint256,uint256,uint256,uint256) should be declared external:
        - _10MBMasterChef.setNftIdSpecificBoostRange(address,uint256,uint256,uint256,uint256) (_10MBMasterchef.sol#1809-1820)
removeNftIdBoostRangeById(address,uint256) should be declared external:
        - _10MBMasterChef.removeNftIdBoostRangeById(address,uint256) (_10MBMasterchef.sol#1822-1826)
setNftWhitelist(address,bool) should be declared external:
        - _10MBMasterChef.setNftWhitelist(address,bool) (_10MBMasterchef.sol#1828-1831)
setReserveFund(address) should be declared external:
        - _10MBMasterChef.setReserveFund(address) (_10MBMasterchef.sol#1833-1836)
flipWhitelistAll() should be declared external:
        - _10MBMasterChef.flipWhitelistAll() (_10MBMasterchef.sol#1838-1840)
harvestAllRewards() should be declared external:
        - _10MBMasterChef.harvestAllRewards() (_10MBMasterchef.sol#1842-1849)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#public-function-that-could-be-declared-external
INFO:Slither:_10MBMasterchef.sol analyzed (22 contracts with 75 detectors), 108 result(s) found
INFO:Slither:Use https://crytic.io/ to get access to additional detectors and Github integration
```



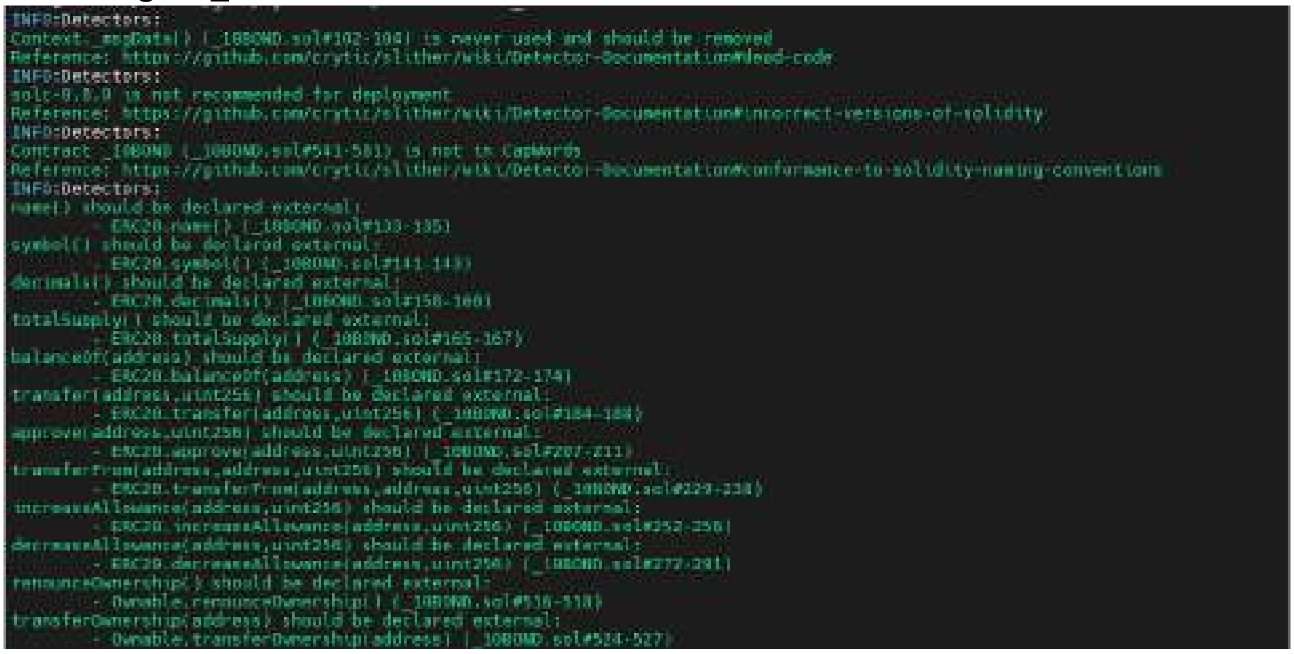
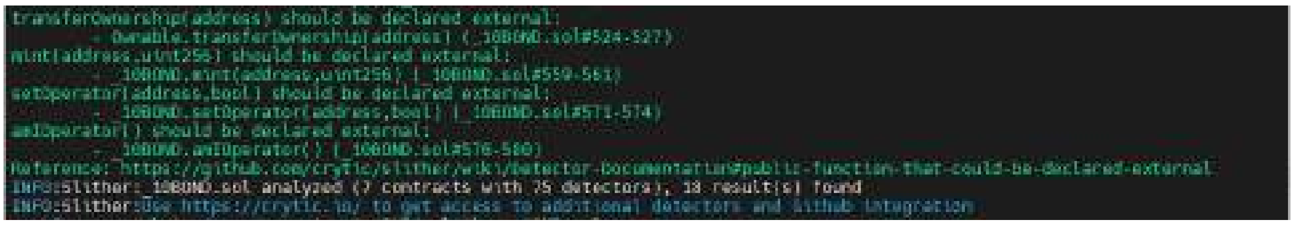
```
INFO:Detectors:
_10MBMasterChef.BURN_ADDRESS (_10MBMasterchef.sol#1339) is never used in _10MBMasterChef (_10MBMasterchef.sol#1293-1851)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#unused-state-variables
INFO:Detectors:
_10MBMasterChef.BURN_ADDRESS (_10MBMasterchef.sol#1339) should be constant
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#state-variables-that-could-be-declared-constant
INFO:Detectors:
name() should be declared external:
        - ERC20.name() (_10MBMasterchef.sol#190-192)
symbol() should be declared external:
        - ERC20.symbol() (_10MBMasterchef.sol#198-200)
decimals() should be declared external:
        - ERC20.decimals() (_10MBMasterchef.sol#215-217)
        - MintableERC20.decimals() (_10MBMasterchef.sol#1289-1291)
totalSupply() should be declared external:
        - ERC20.totalSupply() (_10MBMasterchef.sol#222-224)
balanceOf(address) should be declared external:
        - ERC20.balanceOf(address) (_10MBMasterchef.sol#229-231)
transfer(address,uint256) should be declared external:
        - ERC20.transfer(address,uint256) (_10MBMasterchef.sol#241-245)
approve(address,uint256) should be declared external:
        - ERC20.approve(address,uint256) (_10MBMasterchef.sol#264-268)
transferFrom(address,address,uint256) should be declared external:
        - ERC20.transferFrom(address,address,uint256) (_10MBMasterchef.sol#286-295)
increaseAllowance(address,uint256) should be declared external:
        - ERC20.increaseAllowance(address,uint256) (_10MBMasterchef.sol#309-313)
decreaseAllowance(address,uint256) should be declared external:
        - ERC20.decreaseAllowance(address,uint256) (_10MBMasterchef.sol#329-338)
renounceOwnership() should be declared external:
        - Ownable.renounceOwnership() (_10MBMasterchef.sol#986-988)
transferOwnership(address) should be declared external:
        - Ownable.transferOwnership(address) (_10MBMasterchef.sol#994-997)
getNftIdBoosters(address,uint256) should be declared external:
        - _10MBMasterChef.getNftIdBoosters(address,uint256) (_10MBMasterchef.sol#1438-1448)
getSlots(address,uint256) should be declared external:
        - _10MBMasterChef.getSlots(address,uint256) (_10MBMasterchef.sol#1404-1457)
```

Slither log >> MintableERC20.sol



```
INFO:Detectors:
Context._msgData() (MintableERC20.sol#103-105) is never used and should be removed
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#dead-code
INFO:Detectors:
Pragma version^0.8.0 (MintableERC20.sol#3) necessitates a version too recent to be trusted. Consider deploying with 0.6.12/0.7.
6
solc-0.8.0 is not recommended for deployment
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#incorrect-versions-of-solidity
INFO:Detectors:
name() should be declared external:
        - ERC20.name() (MintableERC20.sol#134-136)
symbol() should be declared external:
        - ERC20.symbol() (MintableERC20.sol#142-144)
decimals() should be declared external:
        - ERC20.decimals() (MintableERC20.sol#159-161)
        - MintableERC20.decimals() (MintableERC20.sol#535-537)
totalSupply() should be declared external:
        - ERC20.totalSupply() (MintableERC20.sol#166-168)
balanceOf(address) should be declared external:
        - ERC20.balanceOf(address) (MintableERC20.sol#173-175)
transfer(address,uint256) should be declared external:
        - ERC20.transfer(address,uint256) (MintableERC20.sol#185-189)
approve(address,uint256) should be declared external:
        - ERC20.approve(address,uint256) (MintableERC20.sol#208-212)
transferFrom(address,address,uint256) should be declared external:
        - ERC20.transferFrom(address,address,uint256) (MintableERC20.sol#230-239)
increaseAllowance(address,uint256) should be declared external:
        - ERC20.increaseAllowance(address,uint256) (MintableERC20.sol#253-257)
decreaseAllowance(address,uint256) should be declared external:
        - ERC20.decreaseAllowance(address,uint256) (MintableERC20.sol#273-282)
renounceOwnership() should be declared external:
        - Ownable.renounceOwnership() (MintableERC20.sol#491-493)
transferOwnership(address) should be declared external:
        - Ownable.transferOwnership(address) (MintableERC20.sol#499-502)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#public-function-that-could-be-declared-external
INFO:Slither:MintableERC20.sol analyzed (6 contracts with 75 detectors), 15 result(s) found
INFO:Slither:Use https://crytic.io/ to get access to additional detectors and Github integration
```

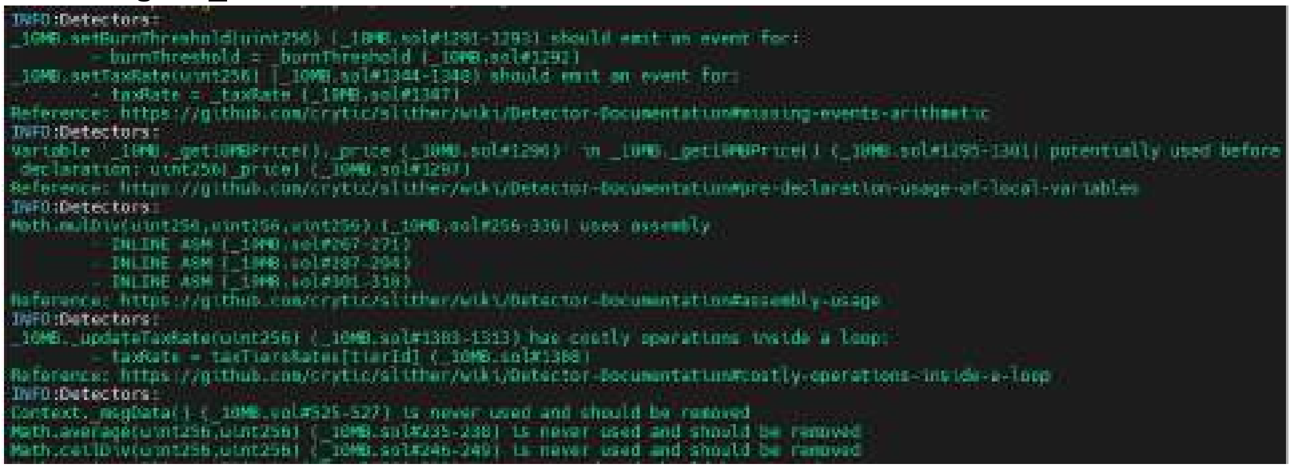
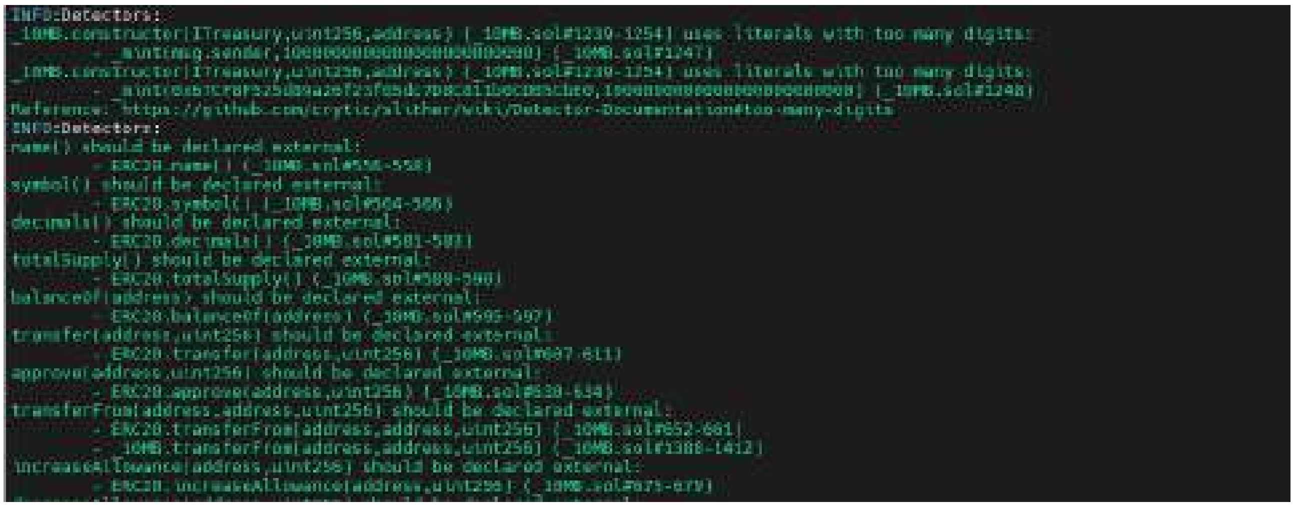
Slither log >> _10BOND.sol

```
INFO:Detectors:
Context._msgData() (_10BOND.sol#102-104) is never used and should be removed
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#dead-code
INFO:Detectors:
solc-0.8.9 is not recommended for deployment
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#incorrect-versions-of-solidity
INFO:Detectors:
Contract _10BOND (_10BOND.sol#541-581) is not in CapWords
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#conformance-to-solidity-naming-conventions
INFO:Detectors:
name() should be declared external:
        - ERC20.name() (_10BOND.sol#133-135)
symbol() should be declared external:
        - ERC20.symbol() (_10BOND.sol#141-143)
decimals() should be declared external:
        - ERC20.decimals() (_10BOND.sol#158-160)
totalSupply() should be declared external:
        - ERC20.totalSupply() (_10BOND.sol#165-167)
balanceOf(address) should be declared external:
        - ERC20.balanceOf(address) (_10BOND.sol#172-174)
transfer(address,uint256) should be declared external:
        - ERC20.transfer(address,uint256) (_10BOND.sol#184-188)
approve(address,uint256) should be declared external:
        - ERC20.approve(address,uint256) (_10BOND.sol#207-211)
transferFrom(address,address,uint256) should be declared external:
        - ERC20.transferFrom(address,address,uint256) (_10BOND.sol#229-238)
increaseAllowance(address,uint256) should be declared external:
        - ERC20.increaseAllowance(address,uint256) (_10BOND.sol#252-256)
decreaseAllowance(address,uint256) should be declared external:
        - ERC20.decreaseAllowance(address,uint256) (_10BOND.sol#272-281)
renounceOwnership() should be declared external:
        - Ownable.renounceOwnership() (_10BOND.sol#516-518)
transferOwnership(address) should be declared external:
        - Ownable.transferOwnership(address) (_10BOND.sol#524-527)
```

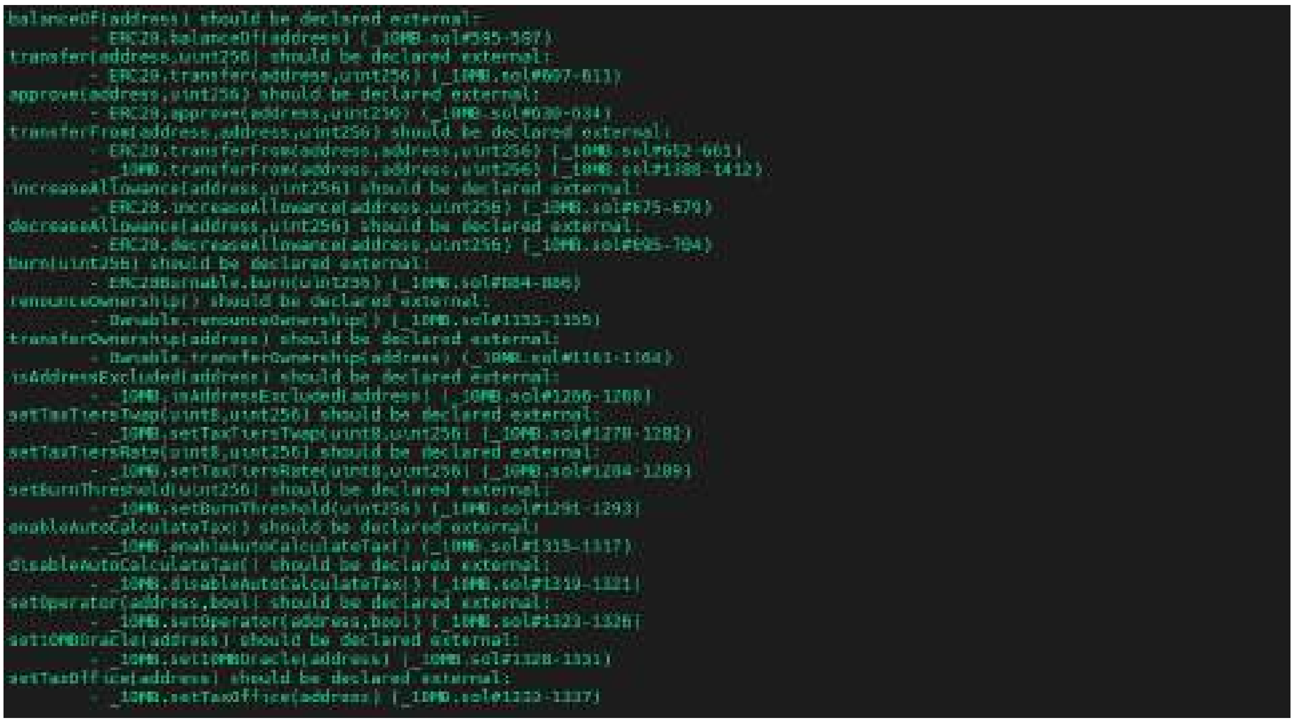
```
transferOwnership(address) should be declared external:
        - Ownable.transferOwnership(address) (_10BOND.sol#524-527)
mint(address,uint256) should be declared external:
        - _10BOND.mint(address,uint256) (_10BOND.sol#559-561)
setOperator(address,bool) should be declared external:
        - _10BOND.setOperator(address,bool) (_10BOND.sol#571-574)
amIOperator() should be declared external:
        - _10BOND.amIOperator() (_10BOND.sol#576-580)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#public-function-that-could-be-declared-external
INFO:Slither:_10BOND.sol analyzed (7 contracts with 75 detectors), 18 result(s) found
INFO:Slither:Use https://crytic.io/ to get access to additional detectors and Github integration
```

Slither log >> _10MB.sol

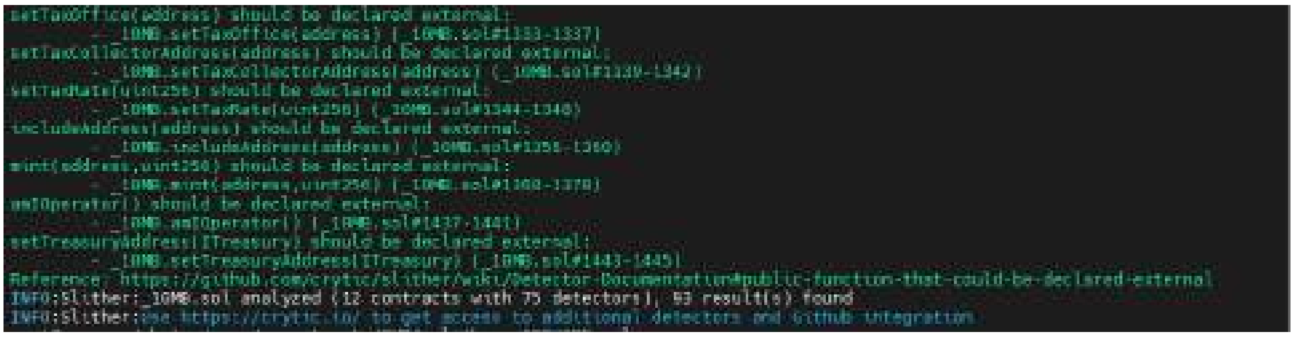



```
INFO:Detectors:
_10MB.setBurnThreshold(uint256) (_10MB.sol#1291-1293) should emit an event for:
        - burnThreshold = _burnThreshold (_10MB.sol#1292)
_10MB.setTaxRate(uint256) (_10MB.sol#1344-1348) should emit an event for:
        - taxRate = _taxRate (_10MB.sol#1347)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#missing-events-arithmetic
INFO:Detectors:
Variable '_10MB._get10MBPrice()._price (_10MB.sol#1296)' in _10MB._get10MBPrice() (_10MB.sol#1295-1301) potentially used before
 declaration: uint256(_price) (_10MB.sol#1297)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#pre-declaration-usage-of-local-variables
INFO:Detectors:
Math.mulDiv(uint256,uint256,uint256) (_10MB.sol#256-330) uses assembly
        - INLINE ASM (_10MB.sol#267-271)
        - INLINE ASM (_10MB.sol#287-294)
        - INLINE ASM (_10MB.sol#301-310)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#assembly-usage
INFO:Detectors:
_10MB._updateTaxRate(uint256) (_10MB.sol#1303-1313) has costly operations inside a loop:
        - taxRate = taxTiersRates[tierId] (_10MB.sol#1308)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#costly-operations-inside-a-loop
INFO:Detectors:
Context._msgData() (_10MB.sol#525-527) is never used and should be removed
Math.average(uint256,uint256) (_10MB.sol#235-238) is never used and should be removed
Math.ceilDiv(uint256,uint256) (_10MB.sol#246-249) is never used and should be removed
```

```
INFO:Detectors:
_10MB.constructor(ITreasury,uint256,address) (_10MB.sol#1239-1254) uses literals with too many digits:
        - _mint(msg.sender,1000000000000000000000000) (_10MB.sol#1247)
_10MB.constructor(ITreasury,uint256,address) (_10MB.sol#1239-1254) uses literals with too many digits:
        - _mint(0xb7CF0F525d09a20f23f05dc7D8C8110dC0D5Cbe0,1000000000000000000000000) (_10MB.sol#1248)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#too-many-digits
INFO:Detectors:
name() should be declared external:
        - ERC20.name() (_10MB.sol#556-558)
symbol() should be declared external:
        - ERC20.symbol() (_10MB.sol#564-566)
decimals() should be declared external:
        - ERC20.decimals() (_10MB.sol#581-583)
totalSupply() should be declared external:
        - ERC20.totalSupply() (_10MB.sol#588-590)
balanceOf(address) should be declared external:
        - ERC20.balanceOf(address) (_10MB.sol#595-597)
transfer(address,uint256) should be declared external:
        - ERC20.transfer(address,uint256) (_10MB.sol#607-611)
approve(address,uint256) should be declared external:
        - ERC20.approve(address,uint256) (_10MB.sol#630-634)
transferFrom(address,address,uint256) should be declared external:
        - ERC20.transferFrom(address,address,uint256) (_10MB.sol#652-661)
        - _10MB.transferFrom(address,address,uint256) (_10MB.sol#1388-1412)
increaseAllowance(address,uint256) should be declared external:
        - ERC20.increaseAllowance(address,uint256) (_10MB.sol#675-679)
```

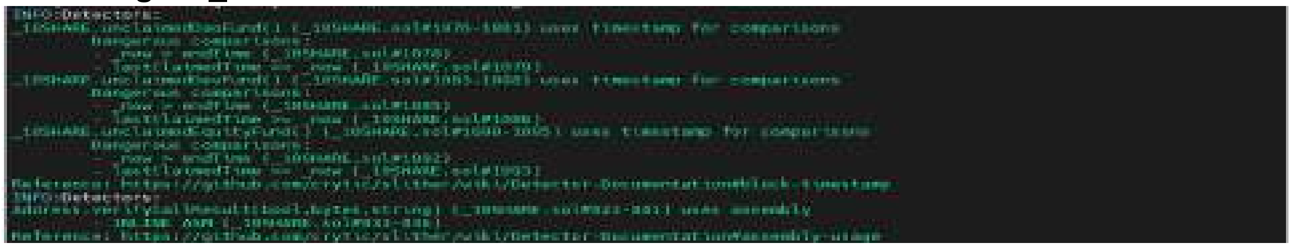


```
balanceOf(address) should be declared external:
        - ERC20.balanceOf(address) (_10MB.sol#595-597)
transfer(address,uint256) should be declared external:
        - ERC20.transfer(address,uint256) (_10MB.sol#607-611)
approve(address,uint256) should be declared external:
        - ERC20.approve(address,uint256) (_10MB.sol#630-634)
transferFrom(address,address,uint256) should be declared external:
        - ERC20.transferFrom(address,address,uint256) (_10MB.sol#652-661)
        - _10MB.transferFrom(address,address,uint256) (_10MB.sol#1388-1412)
increaseAllowance(address,uint256) should be declared external:
        - ERC20.increaseAllowance(address,uint256) (_10MB.sol#675-679)
decreaseAllowance(address,uint256) should be declared external:
        - ERC20.decreaseAllowance(address,uint256) (_10MB.sol#695-704)
burn(uint256) should be declared external:
        - ERC20Burnable.burn(uint256) (_10MB.sol#884-886)
renounceOwnership() should be declared external:
        - Ownable.renounceOwnership() (_10MB.sol#1153-1155)
transferOwnership(address) should be declared external:
        - Ownable.transferOwnership(address) (_10MB.sol#1161-1164)
isAddressExcluded(address) should be declared external:
        - _10MB.isAddressExcluded(address) (_10MB.sol#1266-1268)
setTaxTiersTwap(uint8,uint256) should be declared external:
        - _10MB.setTaxTiersTwap(uint8,uint256) (_10MB.sol#1270-1282)
setTaxTiersRate(uint8,uint256) should be declared external:
        - _10MB.setTaxTiersRate(uint8,uint256) (_10MB.sol#1284-1289)
setBurnThreshold(uint256) should be declared external:
        - _10MB.setBurnThreshold(uint256) (_10MB.sol#1291-1293)
enableAutoCalculateTax() should be declared external:
        - _10MB.enableAutoCalculateTax() (_10MB.sol#1315-1317)
disableAutoCalculateTax() should be declared external:
        - _10MB.disableAutoCalculateTax() (_10MB.sol#1319-1321)
setOperator(address,bool) should be declared external:
        - _10MB.setOperator(address,bool) (_10MB.sol#1323-1326)
set10MBOracle(address) should be declared external:
        - _10MB.set10MBOracle(address) (_10MB.sol#1328-1331)
setTaxOffice(address) should be declared external:
        - _10MB.setTaxOffice(address) (_10MB.sol#1333-1337)
```

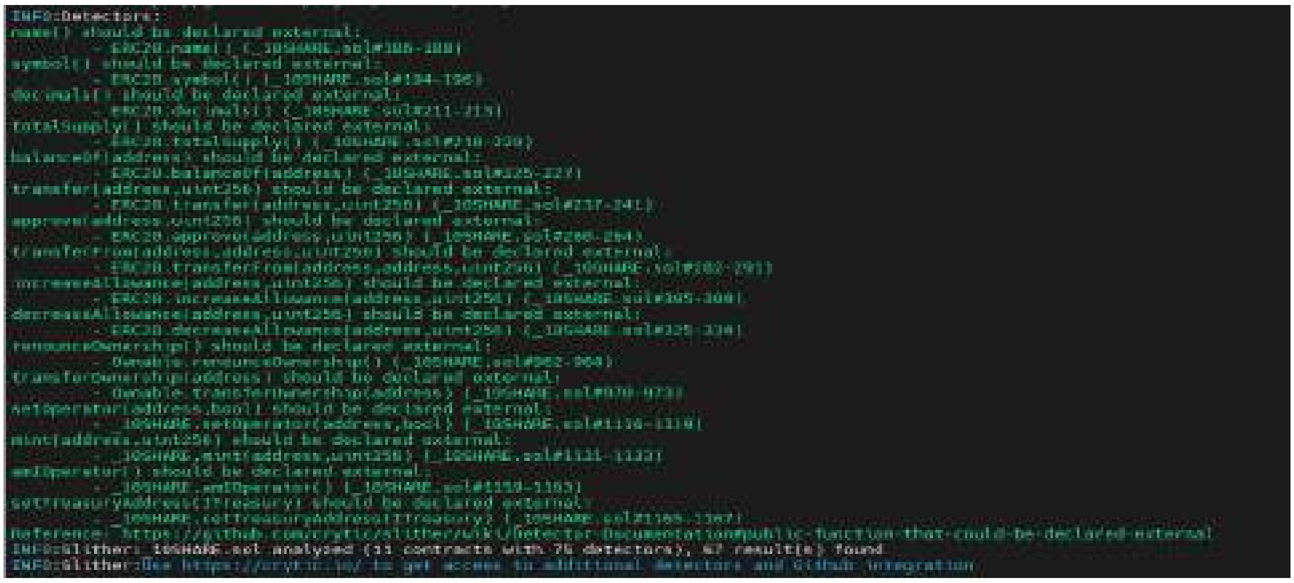


```
setTaxOffice(address) should be declared external:
        - _10MB.setTaxOffice(address) (_10MB.sol#1333-1337)
setTaxCollectorAddress(address) should be declared external:
        - _10MB.setTaxCollectorAddress(address) (_10MB.sol#1339-1342)
setTaxRate(uint256) should be declared external:
        - _10MB.setTaxRate(uint256) (_10MB.sol#1344-1348)
includeAddress(address) should be declared external:
        - _10MB.includeAddress(address) (_10MB.sol#1356-1360)
mint(address,uint256) should be declared external:
        - _10MB.mint(address,uint256) (_10MB.sol#1368-1378)
amIOperator() should be declared external:
        - _10MB.amIOperator() (_10MB.sol#1437-1441)
setTreasuryAddress(ITreasury) should be declared external:
        - _10MB.setTreasuryAddress(ITreasury) (_10MB.sol#1443-1445)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#public-function-that-could-be-declared-external
INFO:Slither:_10MB.sol analyzed (12 contracts with 75 detectors), 93 result(s) found
INFO:Slither:Use https://crytic.io/ to get access to additional detectors and Github integration
```

Slither log >> _10SHARE.sol



```
INFO:Detectors:
_10SHARE.unclaimedDaoFund() (_10SHARE.sol#1076-1081) uses timestamp for comparisons
        Dangerous comparisons:
        - _now > endTime (_10SHARE.sol#1078)
        - lastClaimedTime >= _now (_10SHARE.sol#1079)
_10SHARE.unclaimedDevFund() (_10SHARE.sol#1083-1088) uses timestamp for comparisons
        Dangerous comparisons:
        - _now > endTime (_10SHARE.sol#1085)
        - lastClaimedTime >= _now (_10SHARE.sol#1086)
_10SHARE.unclaimedEquityFund() (_10SHARE.sol#1090-1095) uses timestamp for comparisons
        Dangerous comparisons:
        - _now > endTime (_10SHARE.sol#1092)
        - lastClaimedTime >= _now (_10SHARE.sol#1093)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#block-timestamp
INFO:Detectors:
Address.verifyCallResult(bool,bytes,string) (_10SHARE.sol#822-831) uses assembly
        - INLINE ASM (_10SHARE.sol#831-836)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#assembly-usage
```

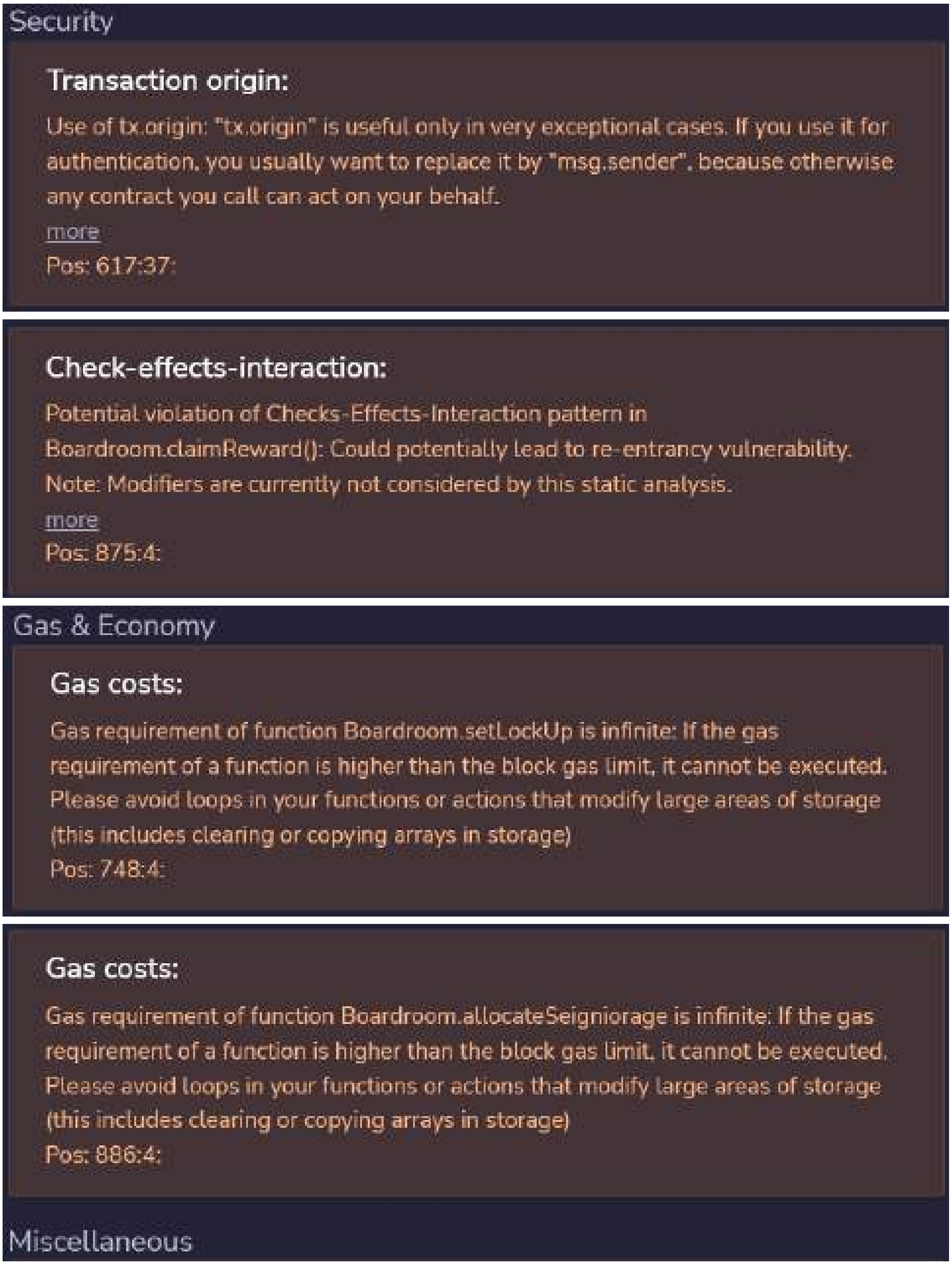


```
INFO:Detectors:
name() should be declared external:
        - ERC20.name() (_10SHARE.sol#186-188)
symbol() should be declared external:
        - ERC20.symbol() (_10SHARE.sol#194-196)
decimals() should be declared external:
        - ERC20.decimals() (_10SHARE.sol#211-213)
totalSupply() should be declared external:
        - ERC20.totalSupply() (_10SHARE.sol#218-220)
balanceOf(address) should be declared external:
        - ERC20.balanceOf(address) (_10SHARE.sol#225-227)
transfer(address,uint256) should be declared external:
        - ERC20.transfer(address,uint256) (_10SHARE.sol#237-241)
approve(address,uint256) should be declared external:
        - ERC20.approve(address,uint256) (_10SHARE.sol#260-264)
transferFrom(address,address,uint256) should be declared external:
        - ERC20.transferFrom(address,address,uint256) (_10SHARE.sol#282-291)
increaseAllowance(address,uint256) should be declared external:
        - ERC20.increaseAllowance(address,uint256) (_10SHARE.sol#305-308)
decreaseAllowance(address,uint256) should be declared external:
        - ERC20.decreaseAllowance(address,uint256) (_10SHARE.sol#325-334)
renounceOwnership() should be declared external:
        - Ownable.renounceOwnership() (_10SHARE.sol#962-964)
transferOwnership(address) should be declared external:
        - Ownable.transferOwnership(address) (_10SHARE.sol#970-973)
setOperator(address,bool) should be declared external:
        - _10SHARE.setOperator(address,bool) (_10SHARE.sol#1116-1119)
mint(address,uint256) should be declared external:
        - _10SHARE.mint(address,uint256) (_10SHARE.sol#1121-1133)
amIOperator() should be declared external:
        - _10SHARE.amIOperator() (_10SHARE.sol#1159-1163)
setTreasuryAddress(ITreasury) should be declared external:
        - _10SHARE.setTreasuryAddress(ITreasury) (_10SHARE.sol#1165-1167)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#public-function-that-could-be-declared-external
INFO:Slither:_10SHARE.sol analyzed (11 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

Solidity Static Analysis

Boardroom.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: 617:37:

Check-effects-interaction:

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

more

Pos: 875:4:

Gas & Economy

Gas costs:

Gas requirement of function Boardroom.setLockUp 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: 748:4:

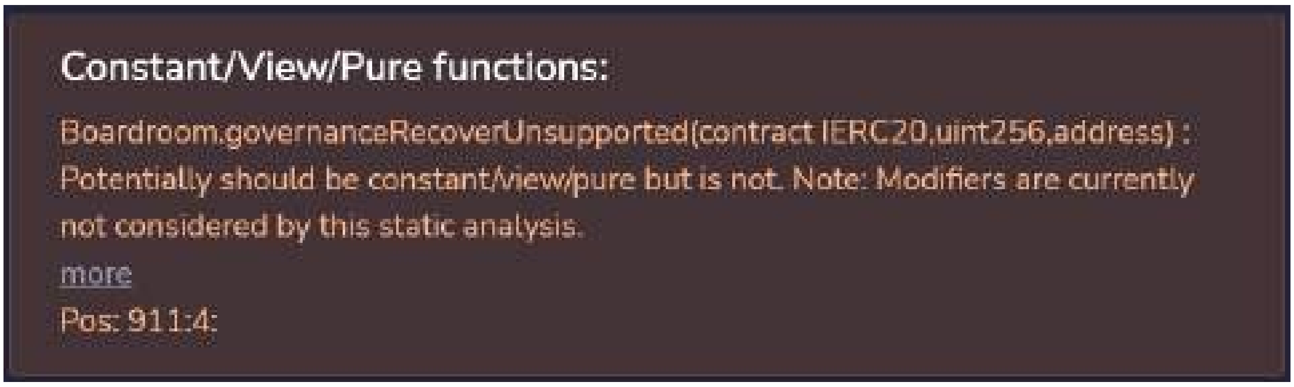
Gas costs:

Gas requirement of function Boardroom.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)

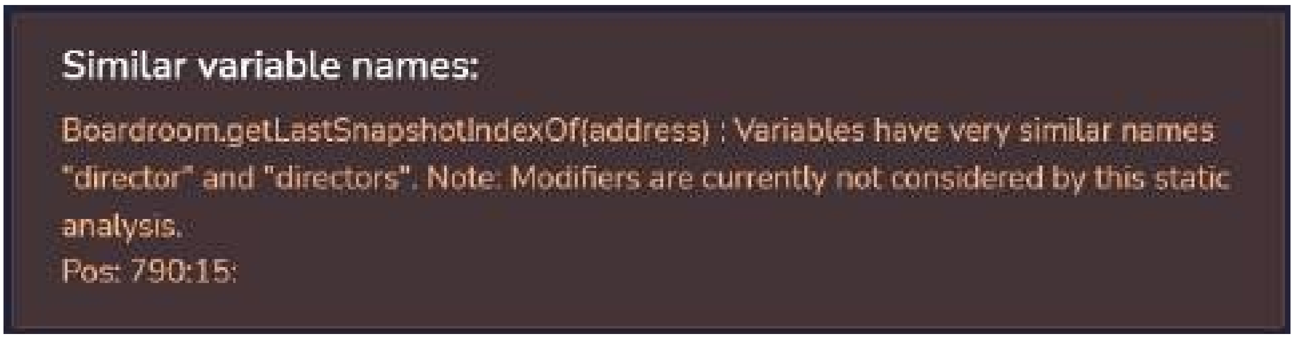
Pos: 886:4:

Miscellaneous

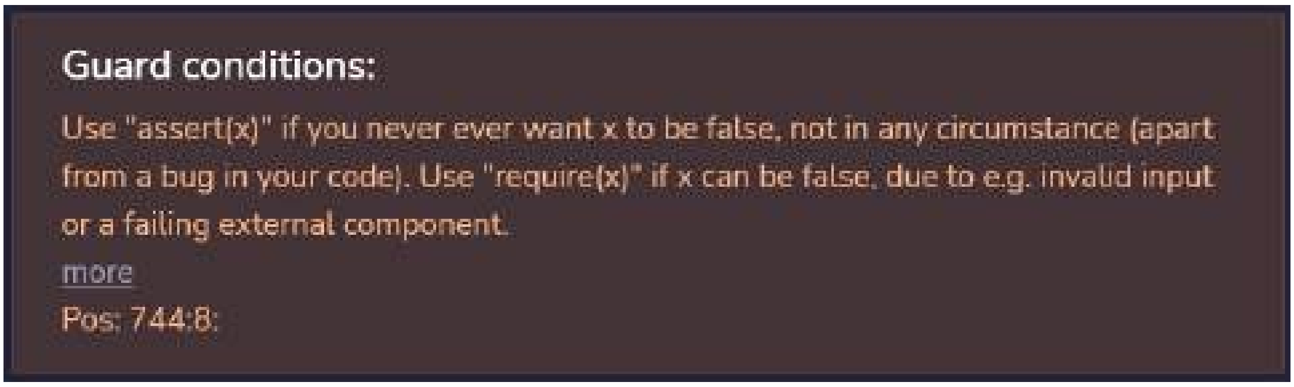
This is a private and confidential document. No part of this document should be disclosed to third party without prior written permission of EtherAuthority.
Email: audit@EtherAuthority.io


Constant/View/Pure functions:

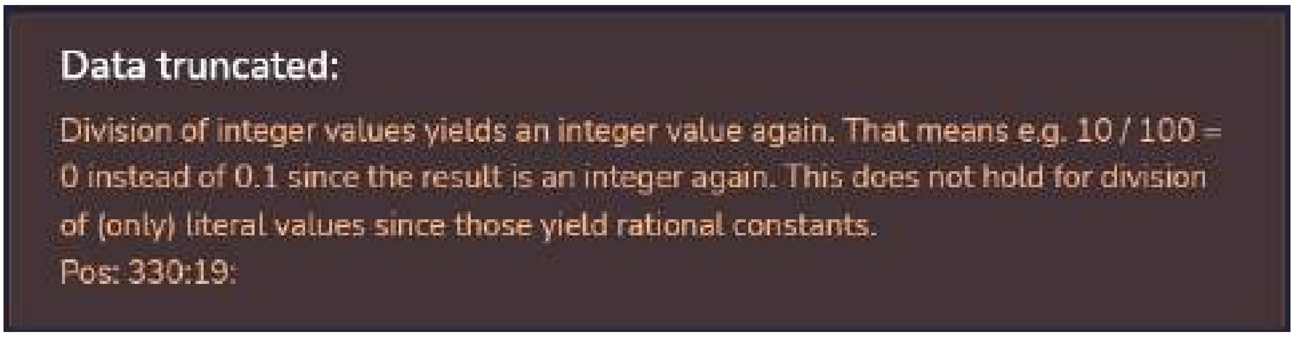
Boardroom.governanceRecoverUnsupported(contract IERC20,uint256,address) : Potentially should be constant/view/pure but is not. Note: Modifiers are currently not considered by this static analysis.
more
Pos: 911:4:


Similar variable names:

Boardroom.getLastSnapshotIndexOf(address) : Variables have very similar names "director" and "directors". Note: Modifiers are currently not considered by this static analysis.
Pos: 790:15:

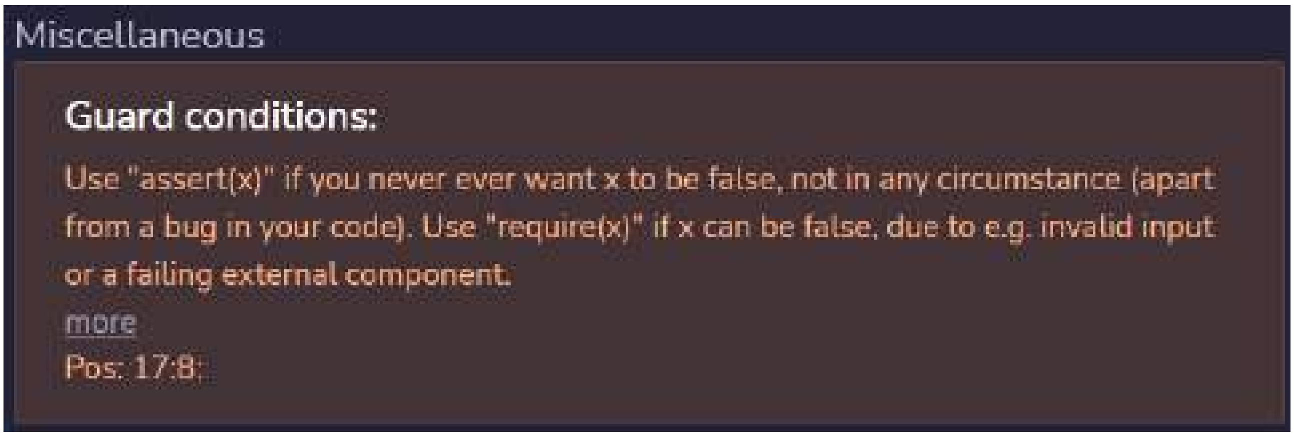

Guard conditions:

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: 744:8:


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: 330:19:

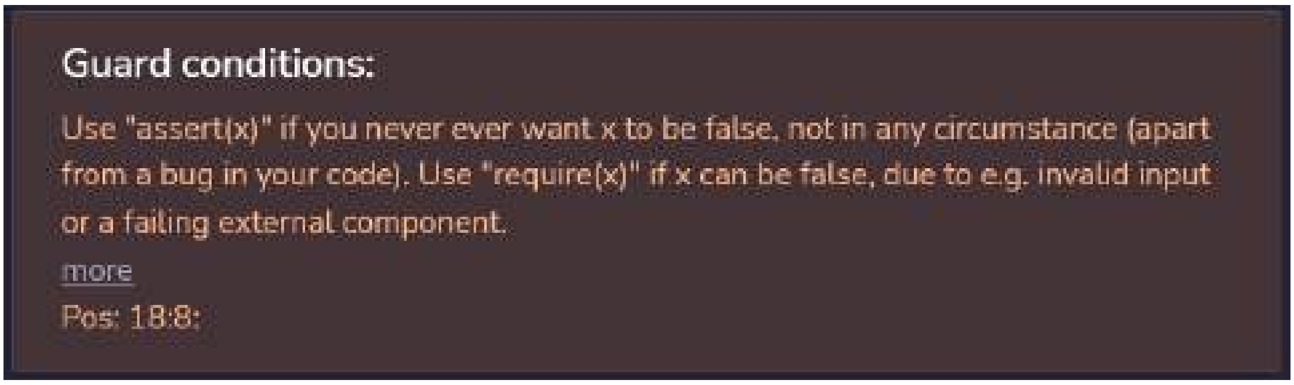
ContractGuard.sol


Miscellaneous

Guard conditions:

Use "assert(x)" if you never ever want x to be false, not in any circumstance (apart from a bug in your code). Use "require(x)" if x can be false, due to e.g. invalid input or a failing external component.
more
Pos: 17:8:

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



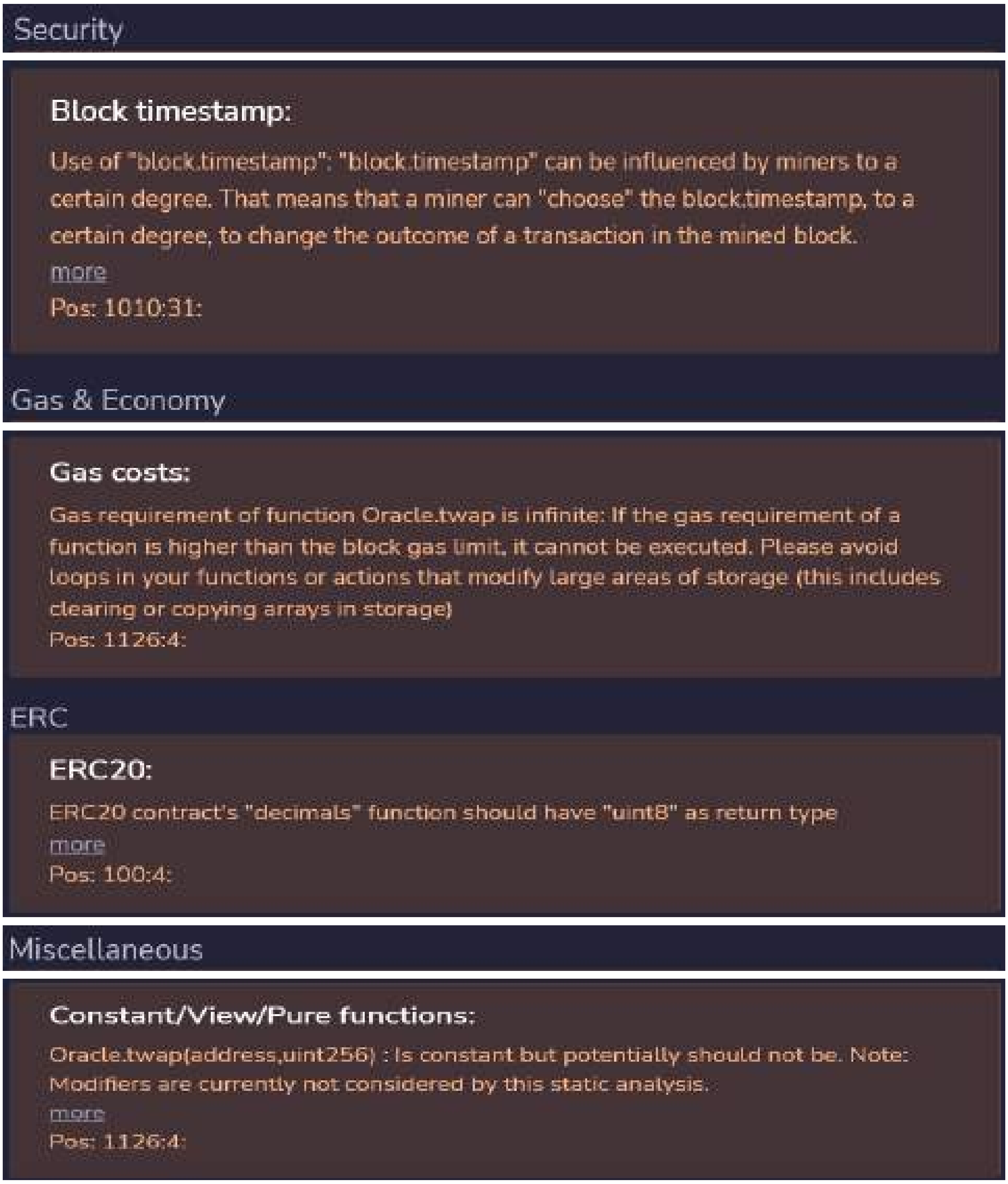
Guard conditions:

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: 18:8:

Oracle.sol



Security

Block timestamp:

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

more

Pos: 1010:31:

Gas & Economy

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: 1126:4:

ERC

ERC20:

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

more

Pos: 100:4:

Miscellaneous

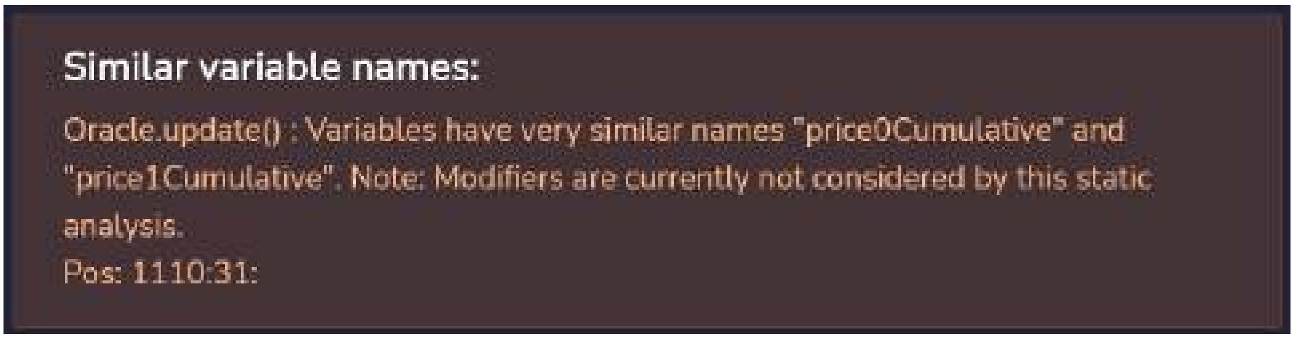
Constant/View/Pure functions:

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

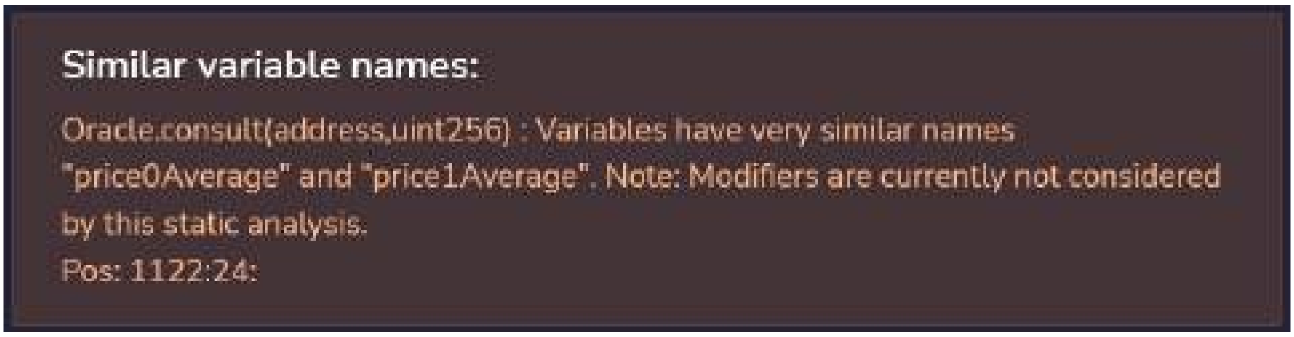
more

Pos: 1126:4:

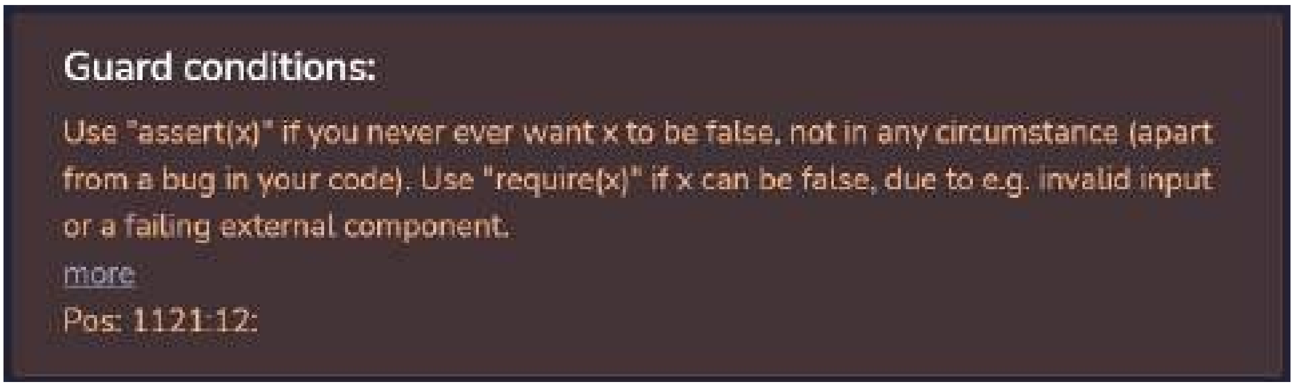
This is a private and confidential document. No part of this document should be disclosed to third party without prior written permission of EtherAuthority.
Email: audit@EtherAuthority.io


Similar variable names:

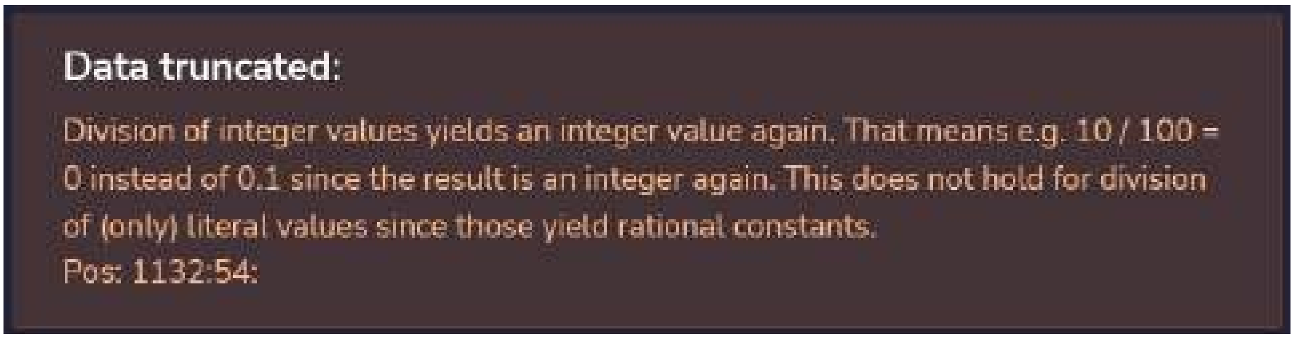
Oracle.update() : Variables have very similar names "price0Cumulative" and "price1Cumulative". Note: Modifiers are currently not considered by this static analysis.
Pos: 1110:31:


Similar variable names:

Oracle.consult(address,uint256) : Variables have very similar names "price0Average" and "price1Average". Note: Modifiers are currently not considered by this static analysis.
Pos: 1122:24:

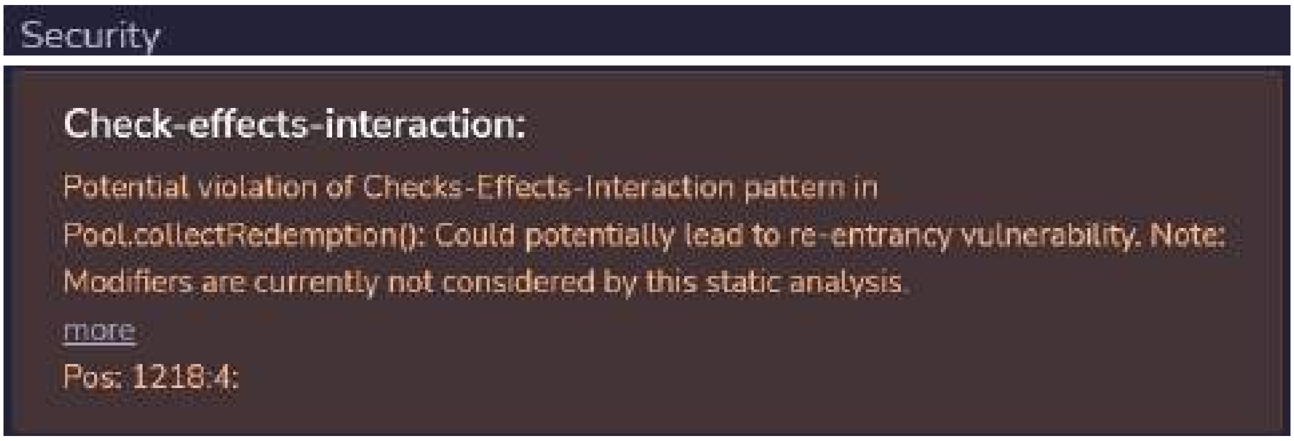

Guard conditions:

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: 1121:12:


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: 1132:54:

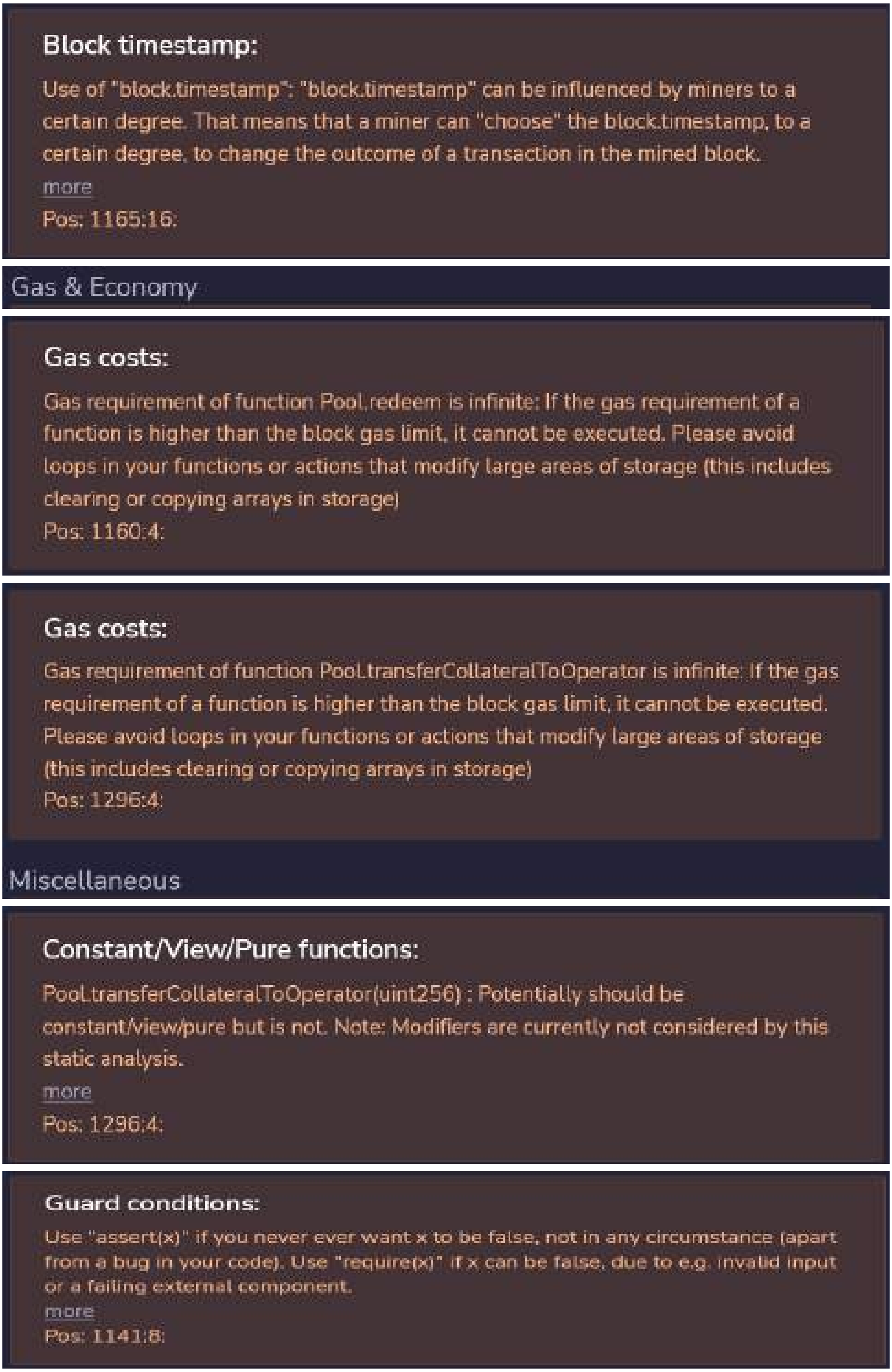
Pool.sol


Security

Check-effects-interaction:

Potential violation of Checks-Effects-Interaction pattern in Pool.collectRedemption(): Could potentially lead to re-entrancy vulnerability. Note: Modifiers are currently not considered by this static analysis.
more
Pos: 1218:4:

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



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: 1165:16:

Gas & Economy

Gas costs:

Gas requirement of function Pool.redeem 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: 1160:4:

Gas costs:

Gas requirement of function Pool.transferCollateralToOperator 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: 1296:4:

Miscellaneous

Constant/View/Pure functions:

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

more

Pos: 1296:4:

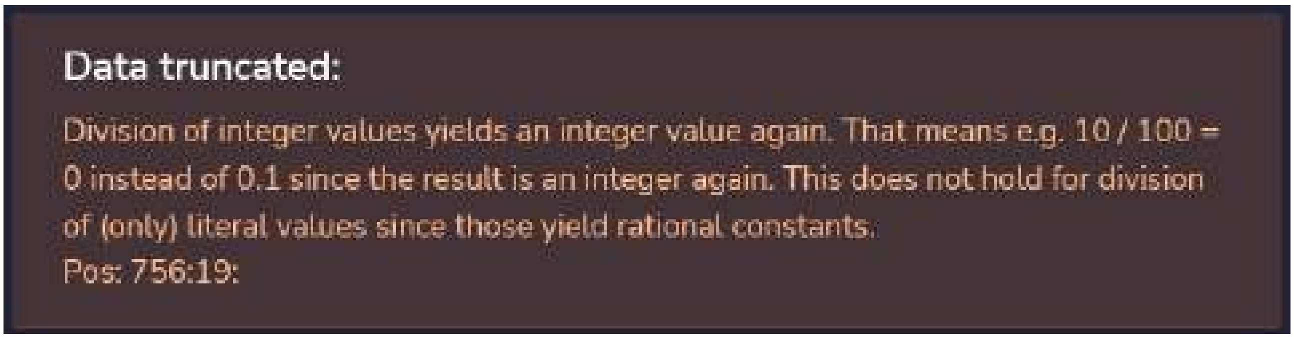
Guard conditions:

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: 1141:8:

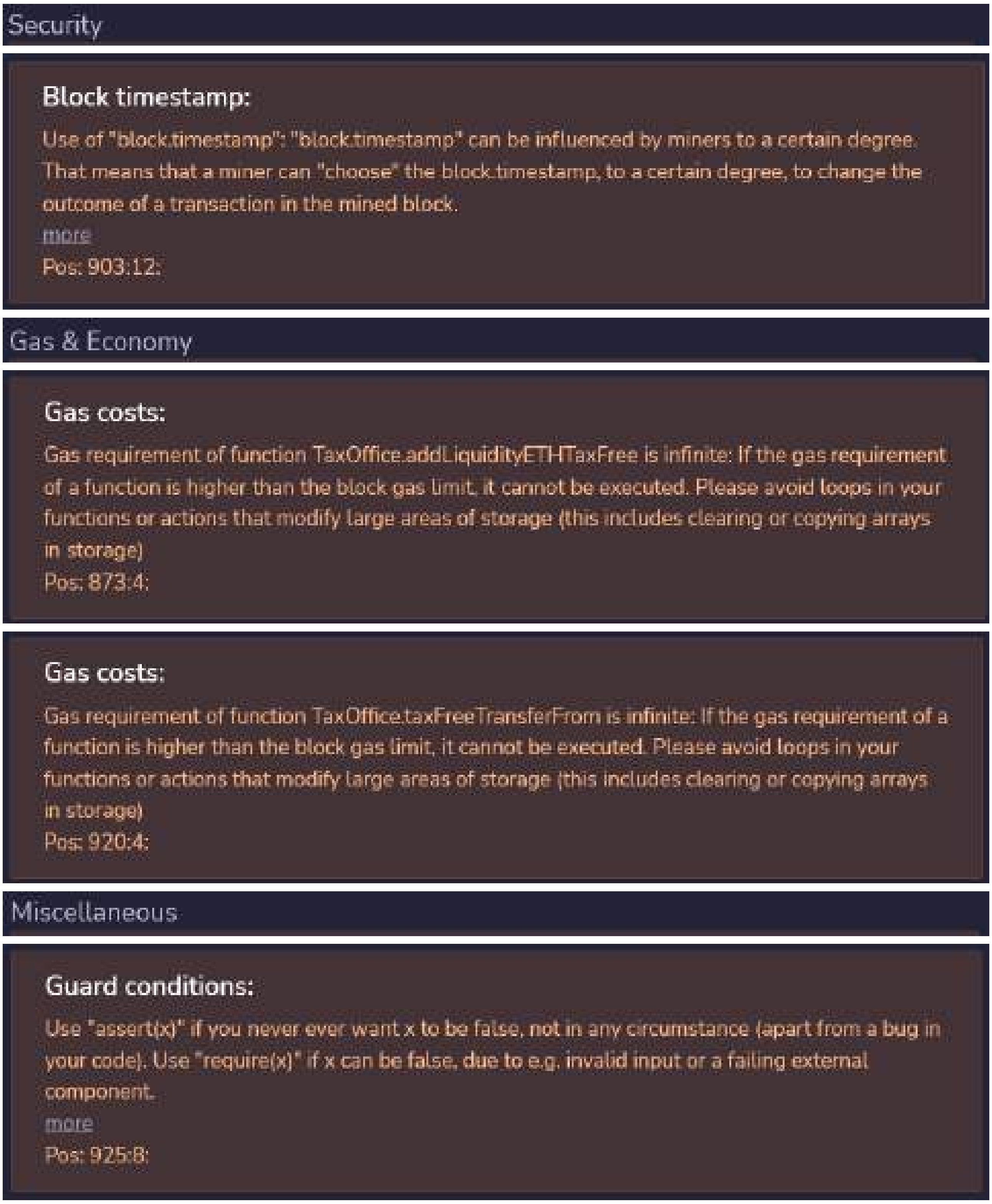
This is a private and confidential document. No part of this document should be disclosed to third party without prior written permission of EtherAuthority.
Email: audit@EtherAuthority.io



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: 756:19:

TaxOffice.sol



Security

Block timestamp:

Use of "block.timestamp": "block.timestamp" can be influenced by miners to a certain degree. That means that a miner can "choose" the block.timestamp, to a certain degree, to change the outcome of a transaction in the mined block.
more
Pos: 903:12:

Gas & Economy

Gas costs:

Gas requirement of function TaxOffice.addLiquidityETHTaxFree 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: 873:4:

Gas costs:

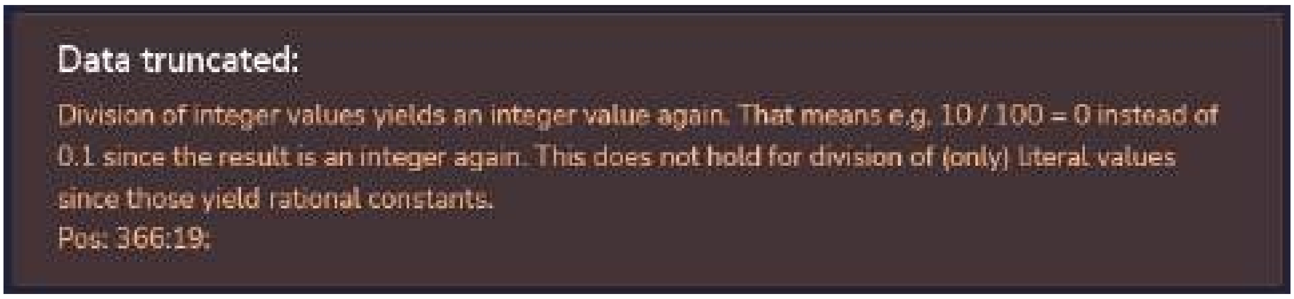
Gas requirement of function TaxOffice.taxFreeTransferFrom is infinite: If the gas requirement of a function is higher than the block gas limit, it cannot be executed. Please avoid loops in your functions or actions that modify large areas of storage (this includes clearing or copying arrays in storage)
Pos: 920:4:

Miscellaneous

Guard conditions:

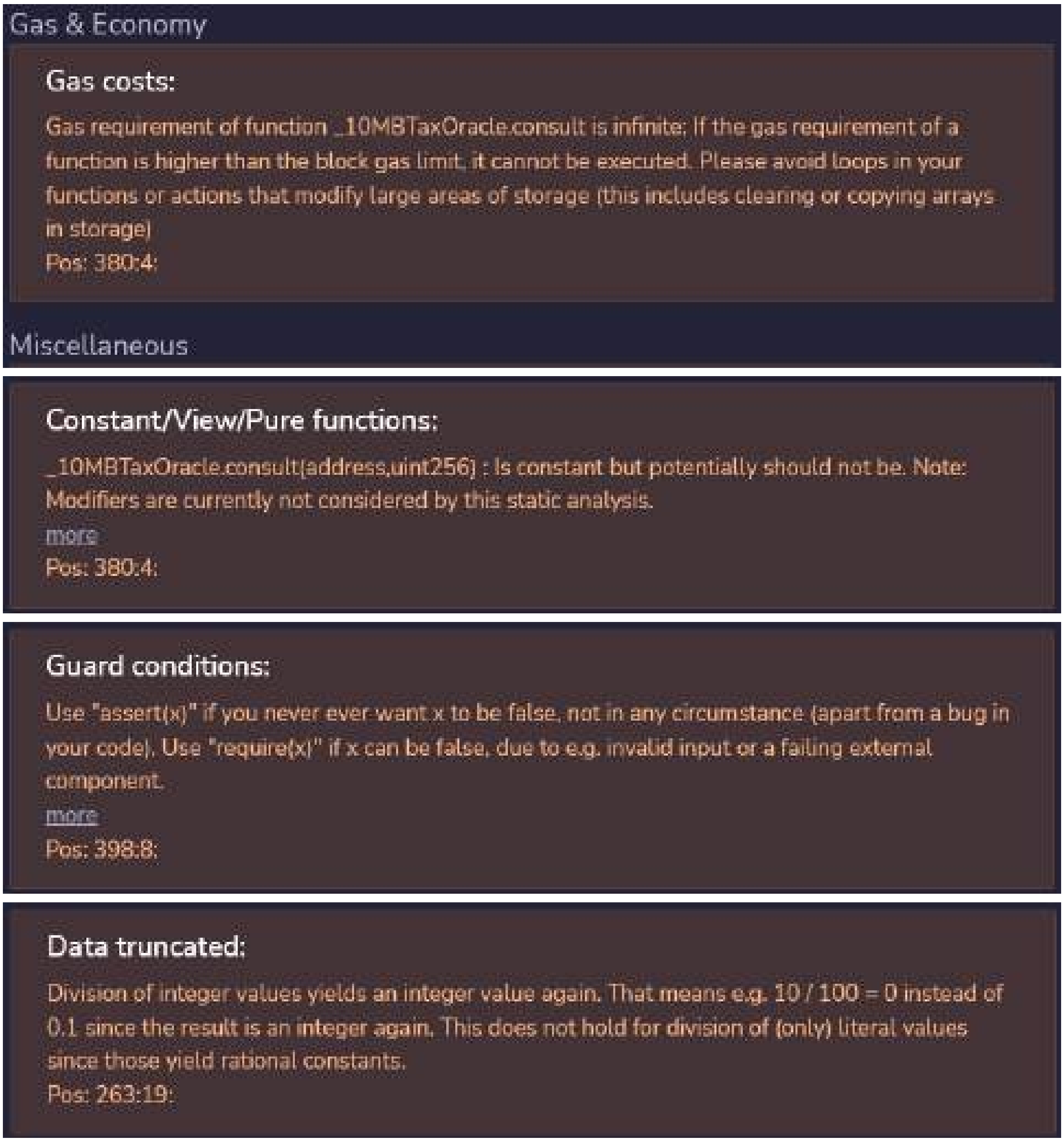
Use "assert(x)" if you never ever want x to be false, not in any circumstance (apart from a bug in your code). Use "require(x)" if x can be false, due to e.g. invalid input or a failing external component.
more
Pos: 925:8:

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


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: 366:19:

TaxOracle.sol


Gas & Economy

Gas costs:

Gas requirement of function _10MBTaxOracle.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: 380:4:

Miscellaneous

Constant/View/Pure functions:

_10MBTaxOracle.consult(address,uint256) : Is constant but potentially should not be. Note: Modifiers are currently not considered by this static analysis.
more
Pos: 380:4:

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: 398:8:

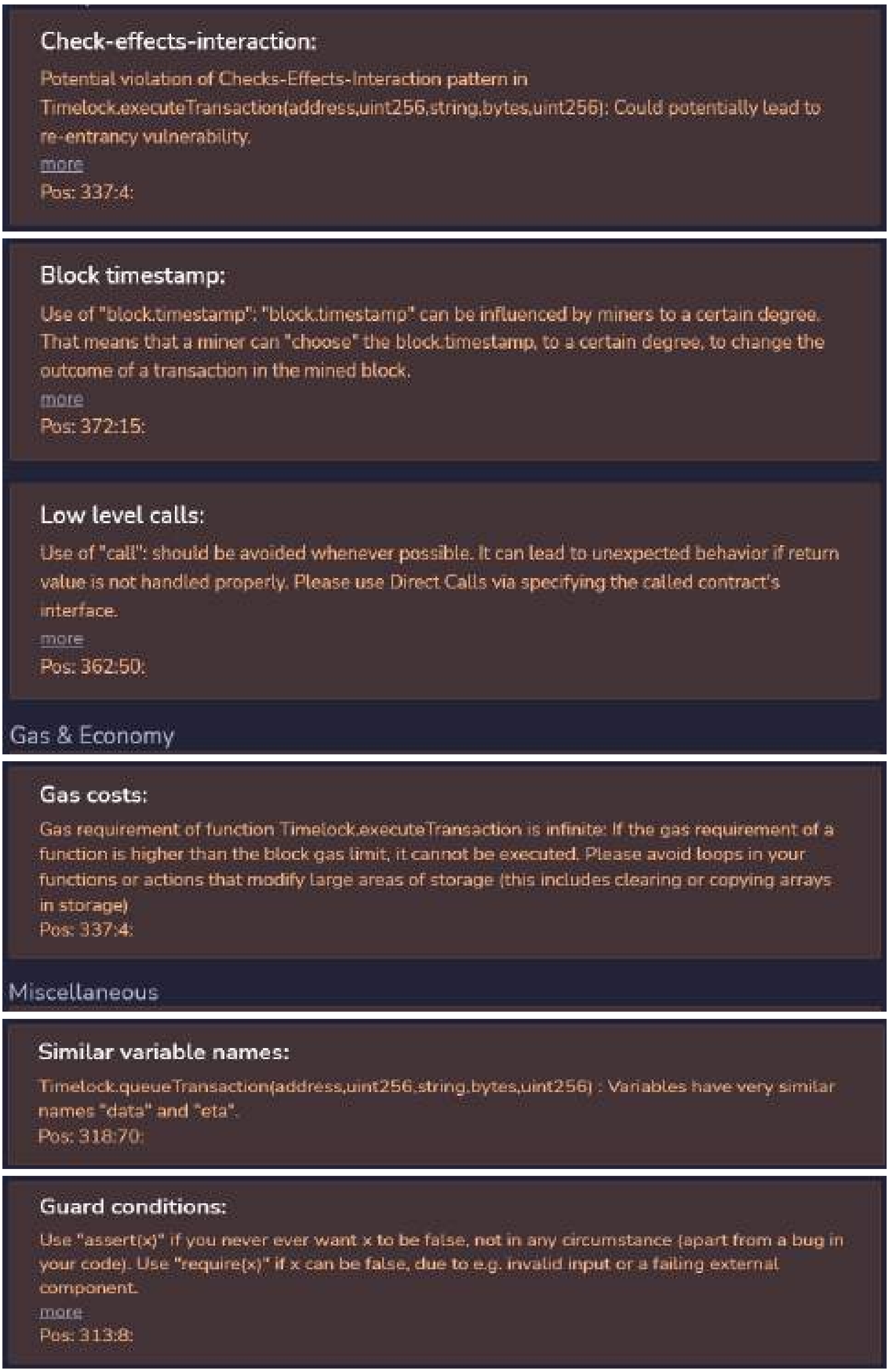
Data truncated:

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: 263:19:

Timelock.sol

Security

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

more

Pos: 337:4:

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: 372:15:

Low level calls:

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

more

Pos: 362:50:

Gas & Economy

Gas costs:

Gas requirement of function Timelock.executeTransaction 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: 337:4:

Miscellaneous

Similar variable names:

Timelock.queueTransaction(address,uint256,string,bytes,uint256) : Variables have very similar names "data" and "eta".

Pos: 318:70:

Guard conditions:

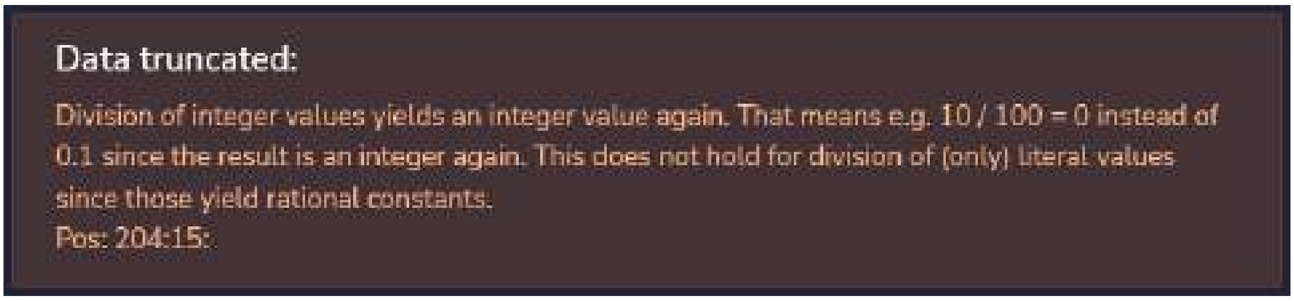
Use "assert(x)" if you never ever want x to be false, not in any circumstance (apart from a bug in your code). Use "require(x)" if x can be false, due to e.g. invalid input or a failing external component.

more

Pos: 313:8:

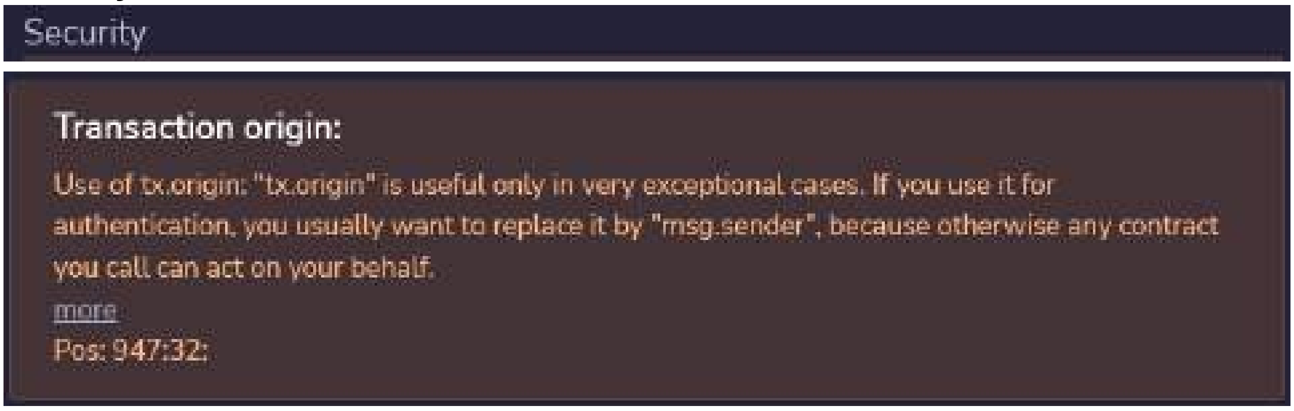
This is a private and confidential document. No part of this document should be disclosed to third party without prior written permission of EtherAuthority.

Email: audit@EtherAuthority.io


Data truncated:

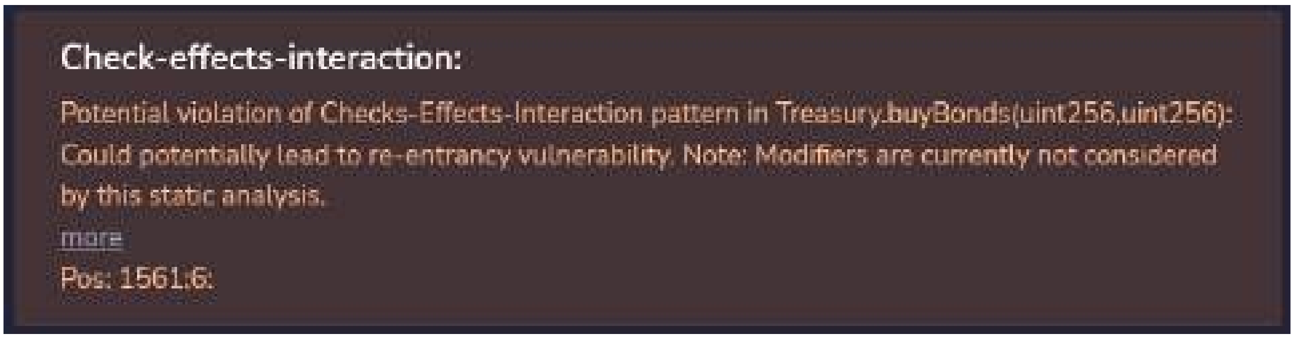
Division of integer values yields an integer value again. That means e.g. 10 / 100 = 0 instead of 0.1 since the result is an integer again. This does not hold for division of (only) literal values since those yield rational constants.
Pos: 204:15:

Treasury.sol

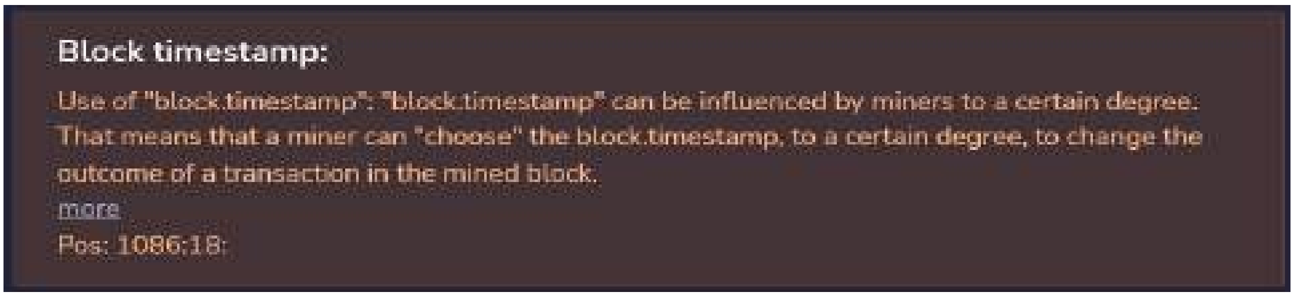

Security

Transaction origin:

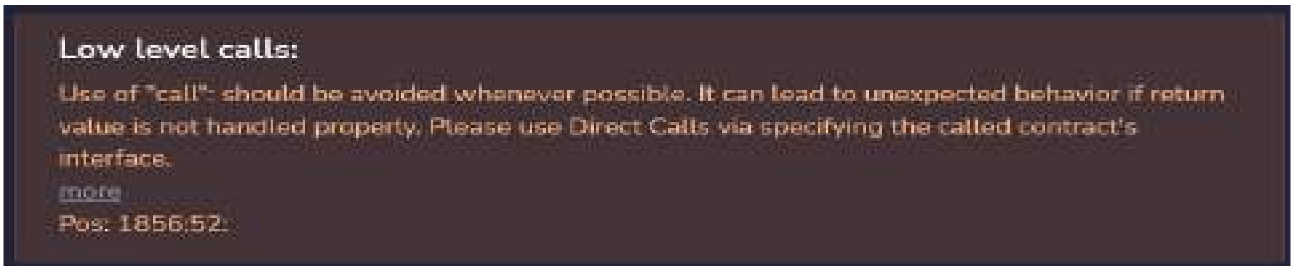
Use of tx.origin: "tx.origin" is useful only in very exceptional cases. If you use it for authentication, you usually want to replace it by "msg.sender", because otherwise any contract you call can act on your behalf.
more
Pos: 947:32:


Check-effects-interaction:

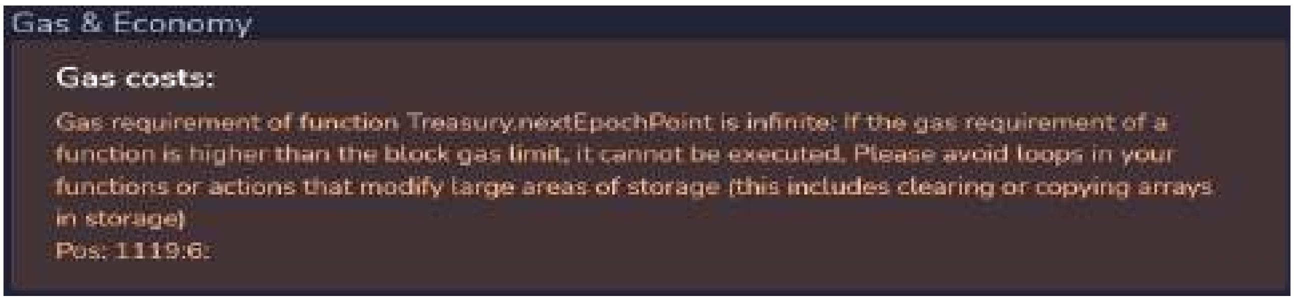
Potential violation of Checks-Effects-Interaction pattern in Treasury.buyBonds(uint256,uint256): Could potentially lead to re-entrancy vulnerability. Note: Modifiers are currently not considered by this static analysis.
more
Pos: 1561:6:


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: 1086:18:


Low level calls:

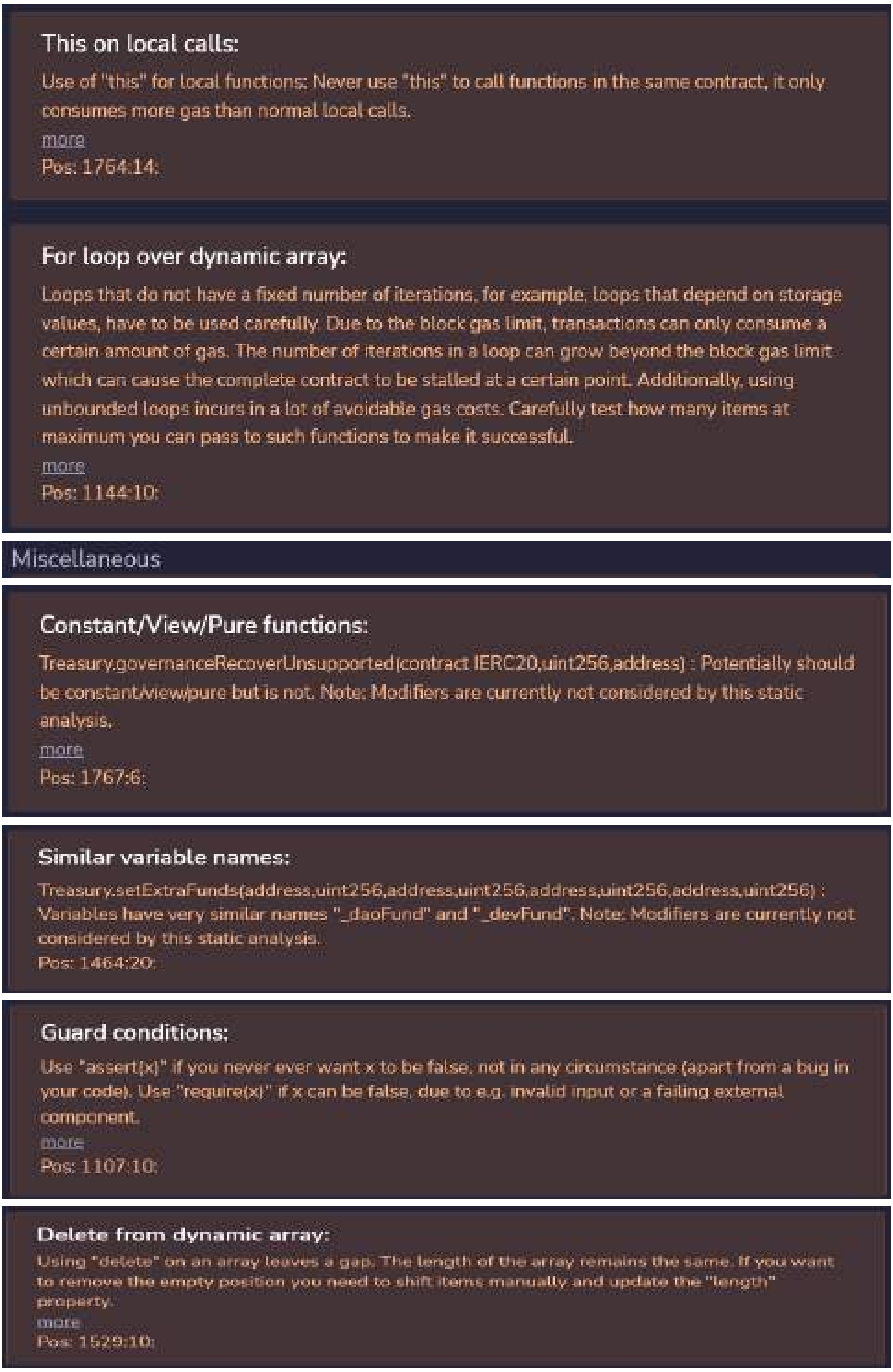
Use of "call": should be avoided whenever possible. It can lead to unexpected behavior if return value is not handled properly. Please use Direct Calls via specifying the called contract's interface.
more
Pos: 1856:52:


Gas & Economy

Gas costs:

Gas requirement of function Treasury.nextEpochPoint 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: 1119:6:

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 on local calls:

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

more

Pos: 1764:14:

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: 1144:10:

Miscellaneous

Constant/View/Pure functions:

Treasury.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: 1767:6:

Similar variable names:

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

Pos: 1464:20:

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: 1107:10:

Delete from dynamic array:

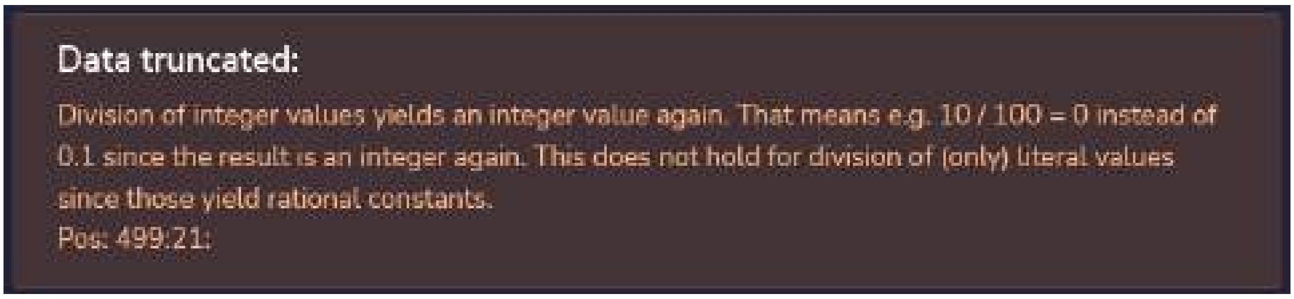
Using "delete" on an array leaves a gap. The length of the array remains the same. If you want to remove the empty position you need to shift items manually and update the "length" property.

more

Pos: 1529:10:

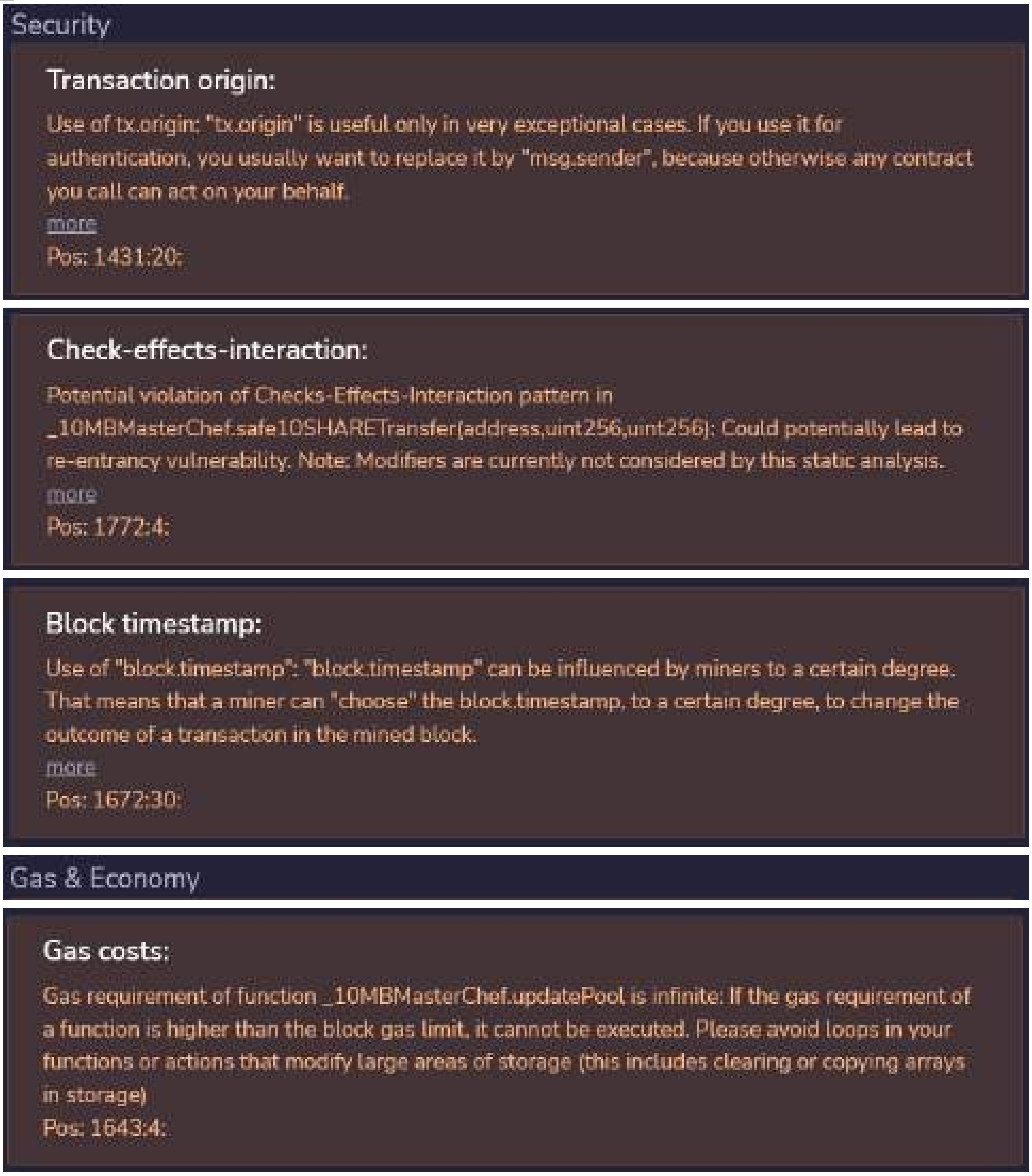
This is a private and confidential document. No part of this document should be disclosed to third party without prior written permission of EtherAuthority.

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: 499:21:

_10MBMasterchef.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: 1431:20:

Check-effects-interaction:

Potential violation of Checks-Effects-Interaction pattern in _10MBMasterChef.safe10SHARETransfer(address,uint256,uint256): Could potentially lead to re-entrancy vulnerability. Note: Modifiers are currently not considered by this static analysis.
more
Pos: 1772:4:

Block timestamp:

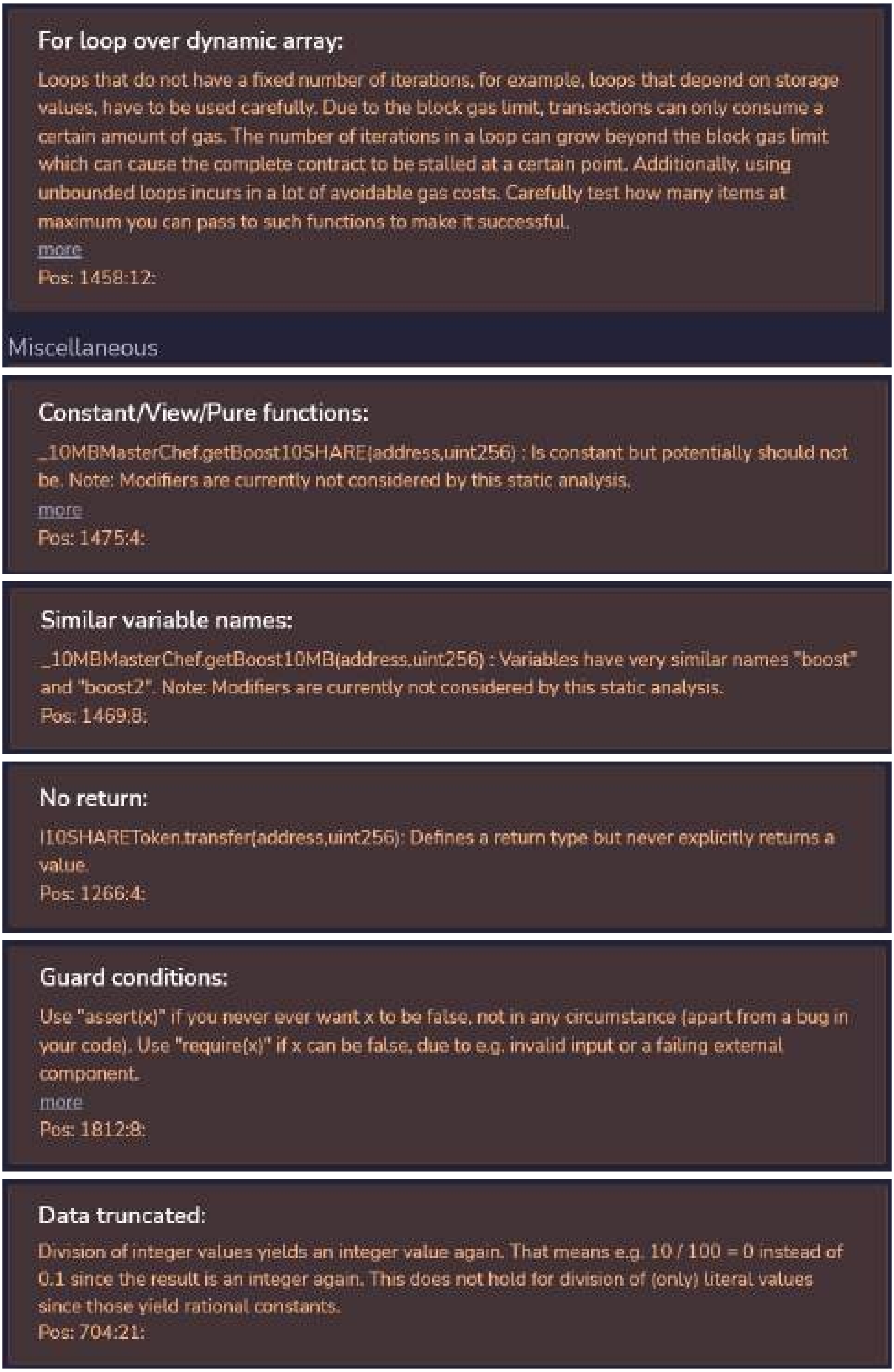
Use of "block.timestamp": "block.timestamp" can be influenced by miners to a certain degree. That means that a miner can "choose" the block.timestamp, to a certain degree, to change the outcome of a transaction in the mined block.
more
Pos: 1672:30:

Gas & Economy

Gas costs:

Gas requirement of function _10MBMasterChef.updatePool 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: 1643:4:

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



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: 1458:12:

Miscellaneous

Constant/View/Pure functions:

_10MBMasterChef.getBoost10SHARE(address,uint256) : Is constant but potentially should not be. Note: Modifiers are currently not considered by this static analysis.
more
Pos: 1475:4:

Similar variable names:

_10MBMasterChef.getBoost10MB(address,uint256) : Variables have very similar names "boost" and "boost2". Note: Modifiers are currently not considered by this static analysis.
Pos: 1469:8:

No return:

I10SHAREToken.transfer(address,uint256): Defines a return type but never explicitly returns a value.
Pos: 1266:4:

Guard conditions:

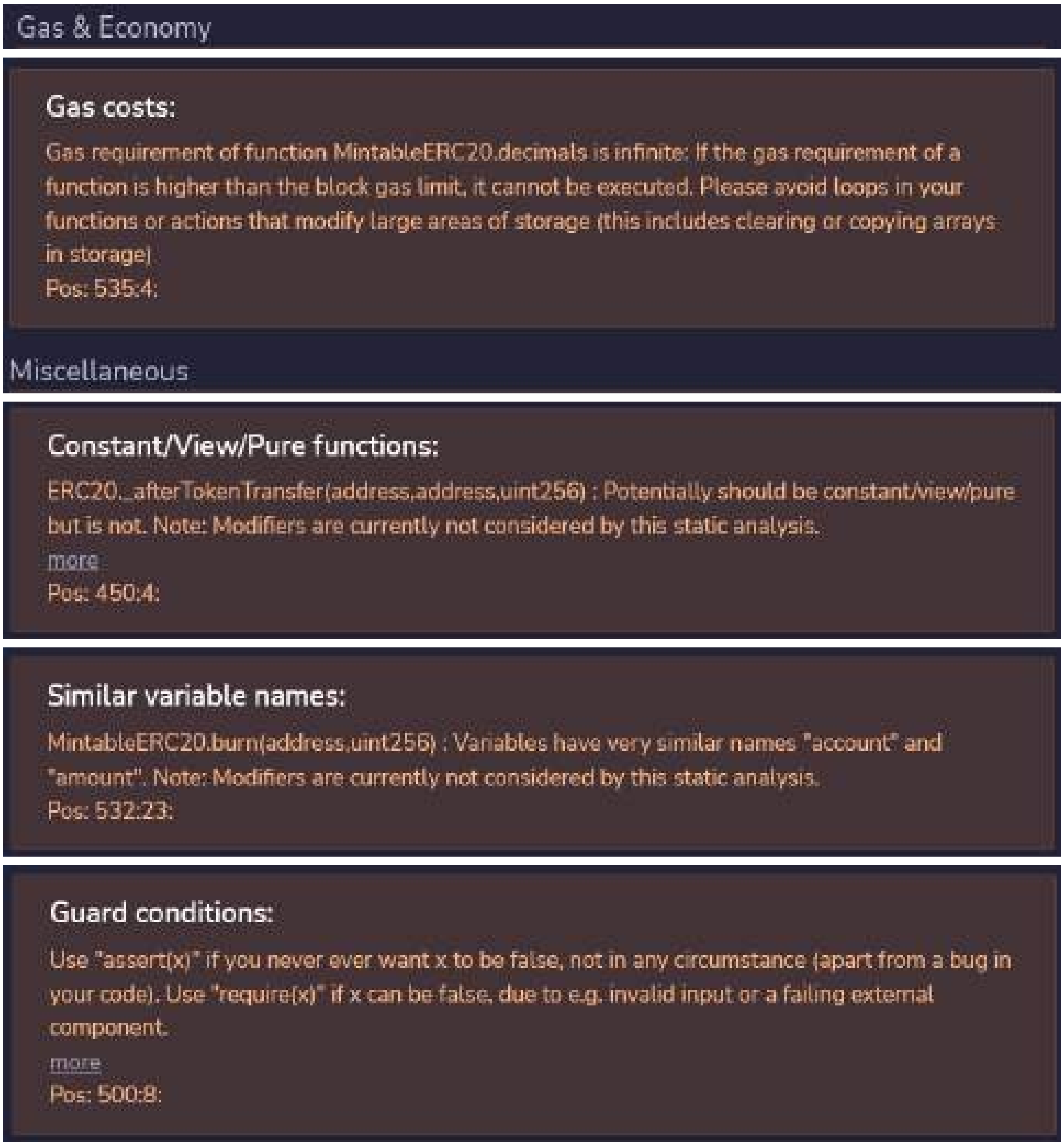
Use "assert(x)" if you never ever want x to be false, not in any circumstance (apart from a bug in your code). Use "require(x)" if x can be false, due to e.g. invalid input or a failing external component.
more
Pos: 1812:8:

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: 704:21:

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

MintableERC20.sol



Gas & Economy

Gas costs:

Gas requirement of function MintableERC20.decimals 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: 535:4:

Miscellaneous

Constant/View/Pure functions:

ERC20._afterTokenTransfer(address,address,uint256) : Potentially should be constant/view/pure but is not. Note: Modifiers are currently not considered by this static analysis.
more
Pos: 450:4:

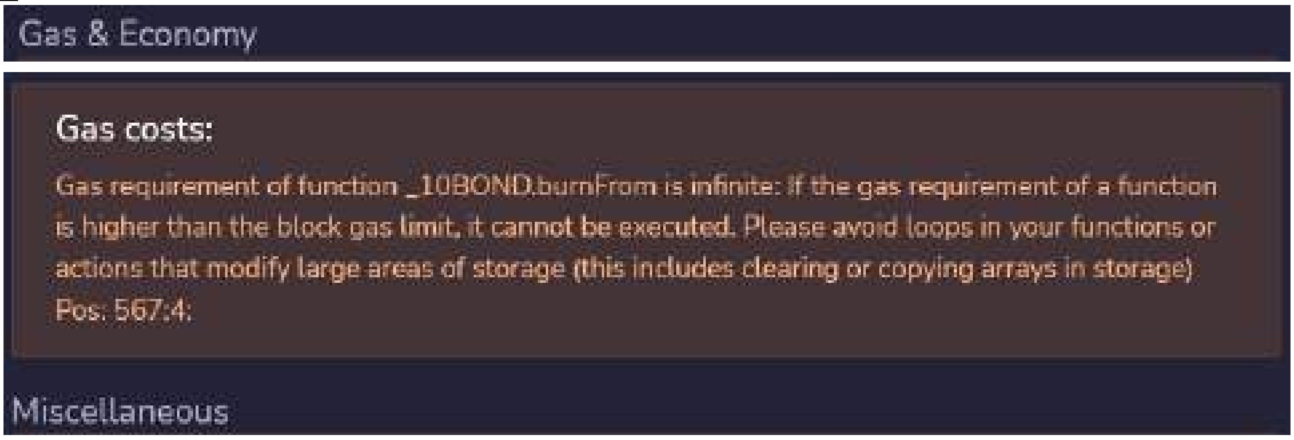
Similar variable names:

MintableERC20.burn(address,uint256) : Variables have very similar names "account" and "amount". Note: Modifiers are currently not considered by this static analysis.
Pos: 532:23:

Guard conditions:

Use "assert(x)" if you never ever want x to be false, not in any circumstance (apart from a bug in your code). Use "require(x)" if x can be false, due to e.g. invalid input or a failing external component.
more
Pos: 500:8:

_10BOND.sol



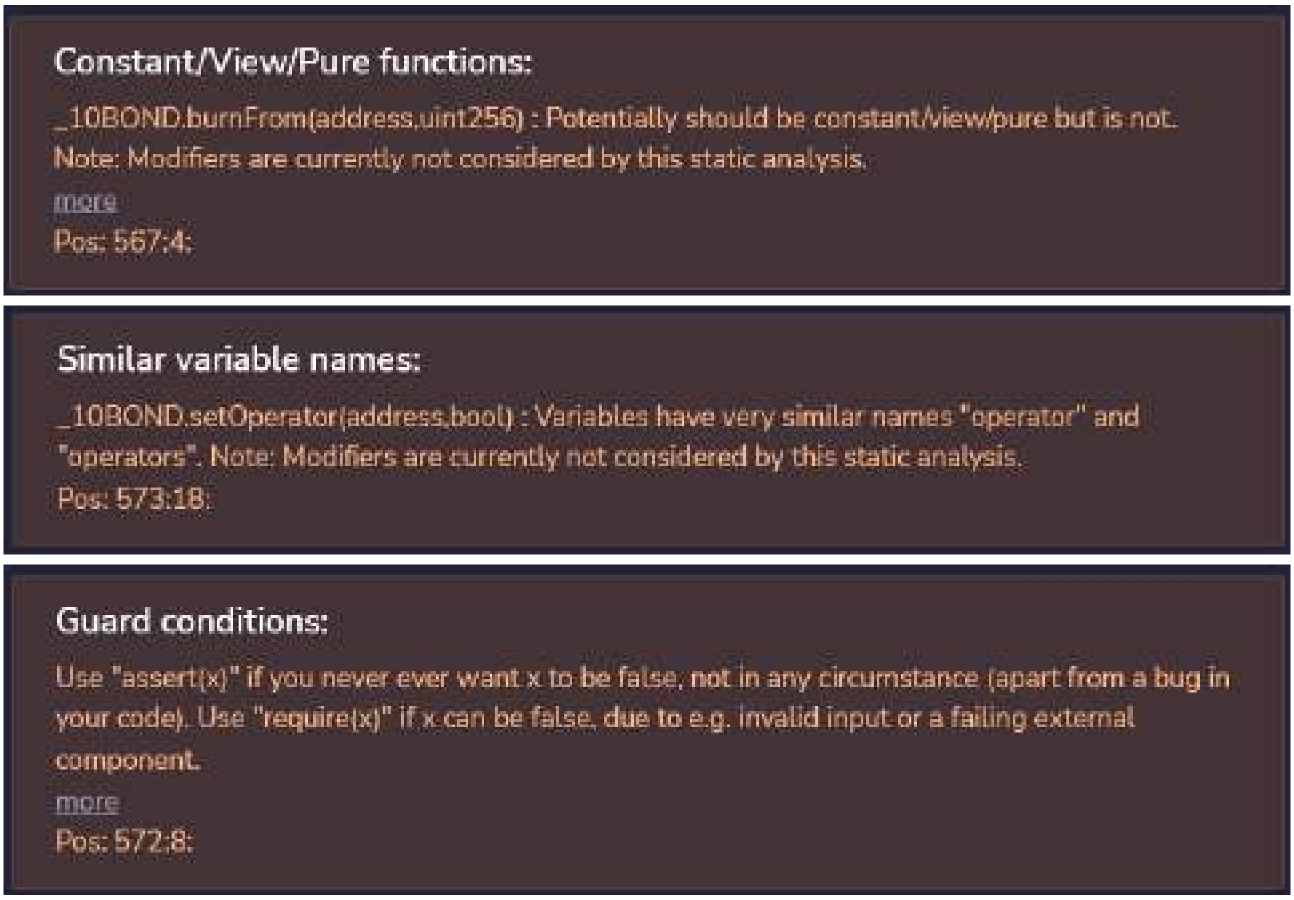
Gas & Economy

Gas costs:

Gas requirement of function _10BOND.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: 567:4:

Miscellaneous

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


Constant/View/Pure functions:

_10BOND.burnFrom(address,uint256) : Potentially should be constant/view/pure but is not. Note: Modifiers are currently not considered by this static analysis.

more

Pos: 567:4:

Similar variable names:

_10BOND.setOperator(address,bool) : Variables have very similar names "operator" and "operators". Note: Modifiers are currently not considered by this static analysis.

Pos: 573:18:

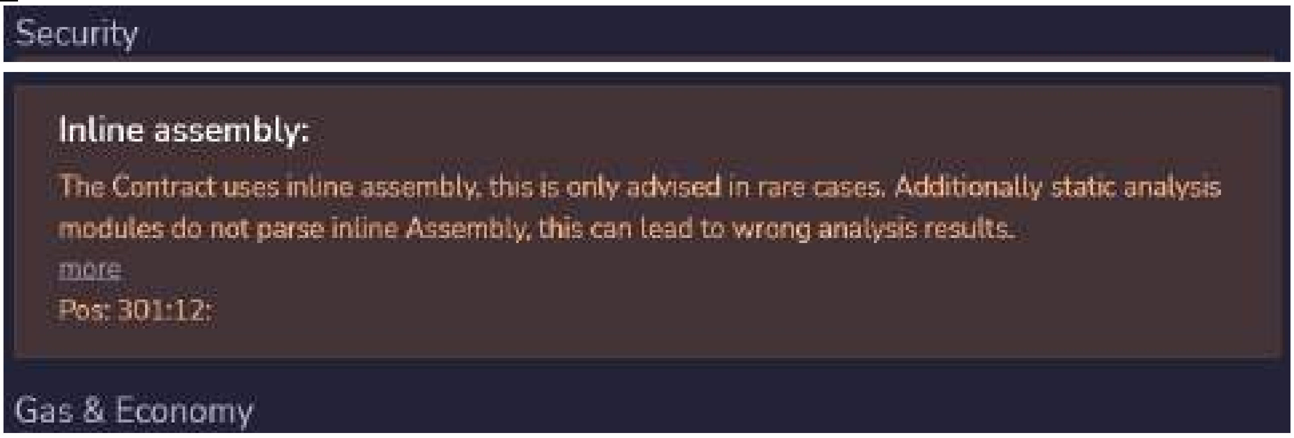
Guard conditions:

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

more

Pos: 572:8:

_10MB.sol


Security

Inline assembly:

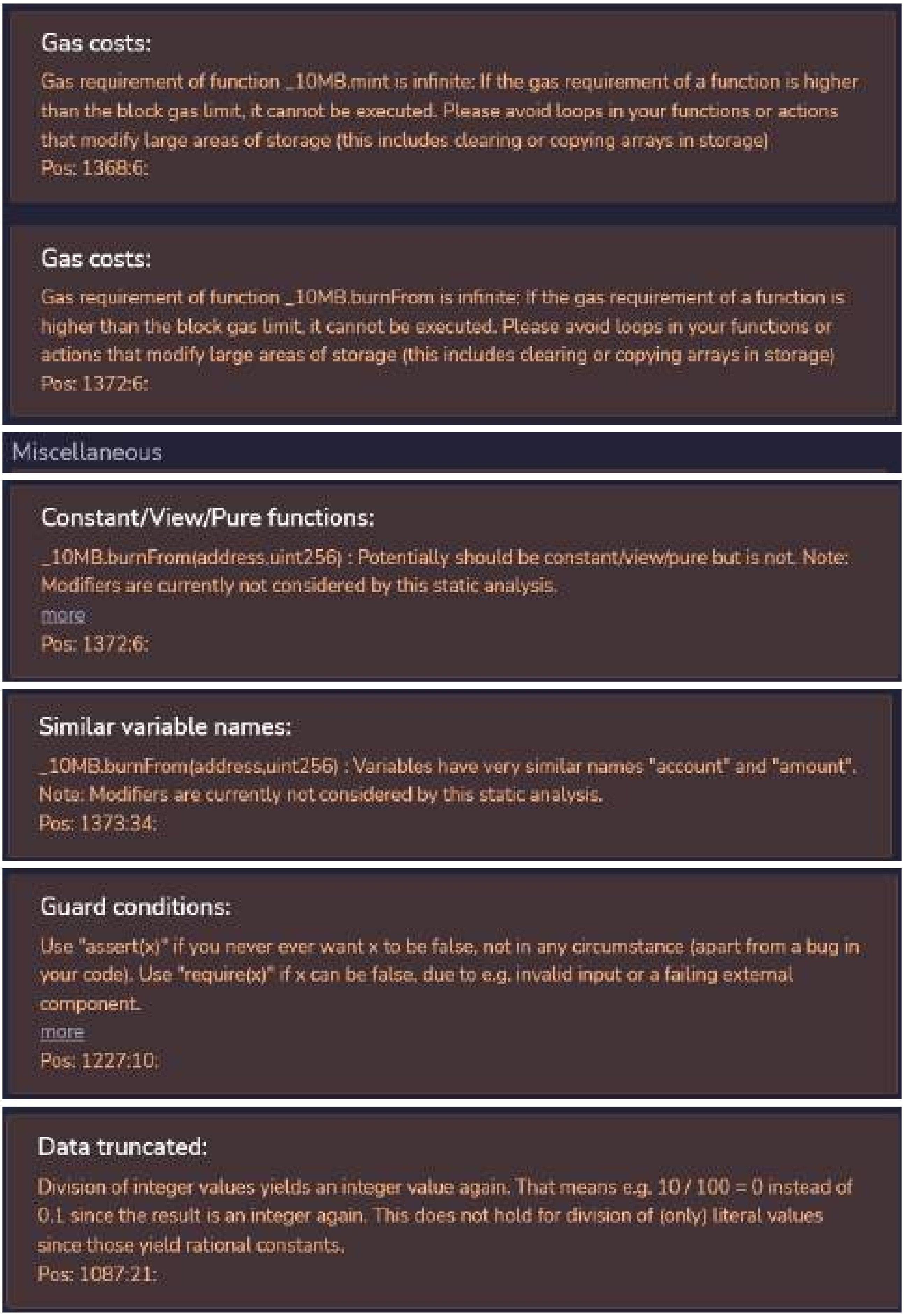
The Contract uses inline assembly, this is only advised in rare cases. Additionally static analysis modules do not parse inline Assembly, this can lead to wrong analysis results.

more

Pos: 301:12:

Gas & Economy

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 _10MB.mint is infinite: If the gas requirement of a function is higher than the block gas limit, it cannot be executed. Please avoid loops in your functions or actions that modify large areas of storage (this includes clearing or copying arrays in storage)
Pos: 1368:6:

Gas costs:

Gas requirement of function _10MB.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: 1372:6:

Miscellaneous

Constant/View/Pure functions:

_10MB.burnFrom(address,uint256) : Potentially should be constant/view/pure but is not. Note: Modifiers are currently not considered by this static analysis.
more
Pos: 1372:6:

Similar variable names:

_10MB.burnFrom(address,uint256) : Variables have very similar names "account" and "amount". Note: Modifiers are currently not considered by this static analysis.
Pos: 1373:34:

Guard conditions:

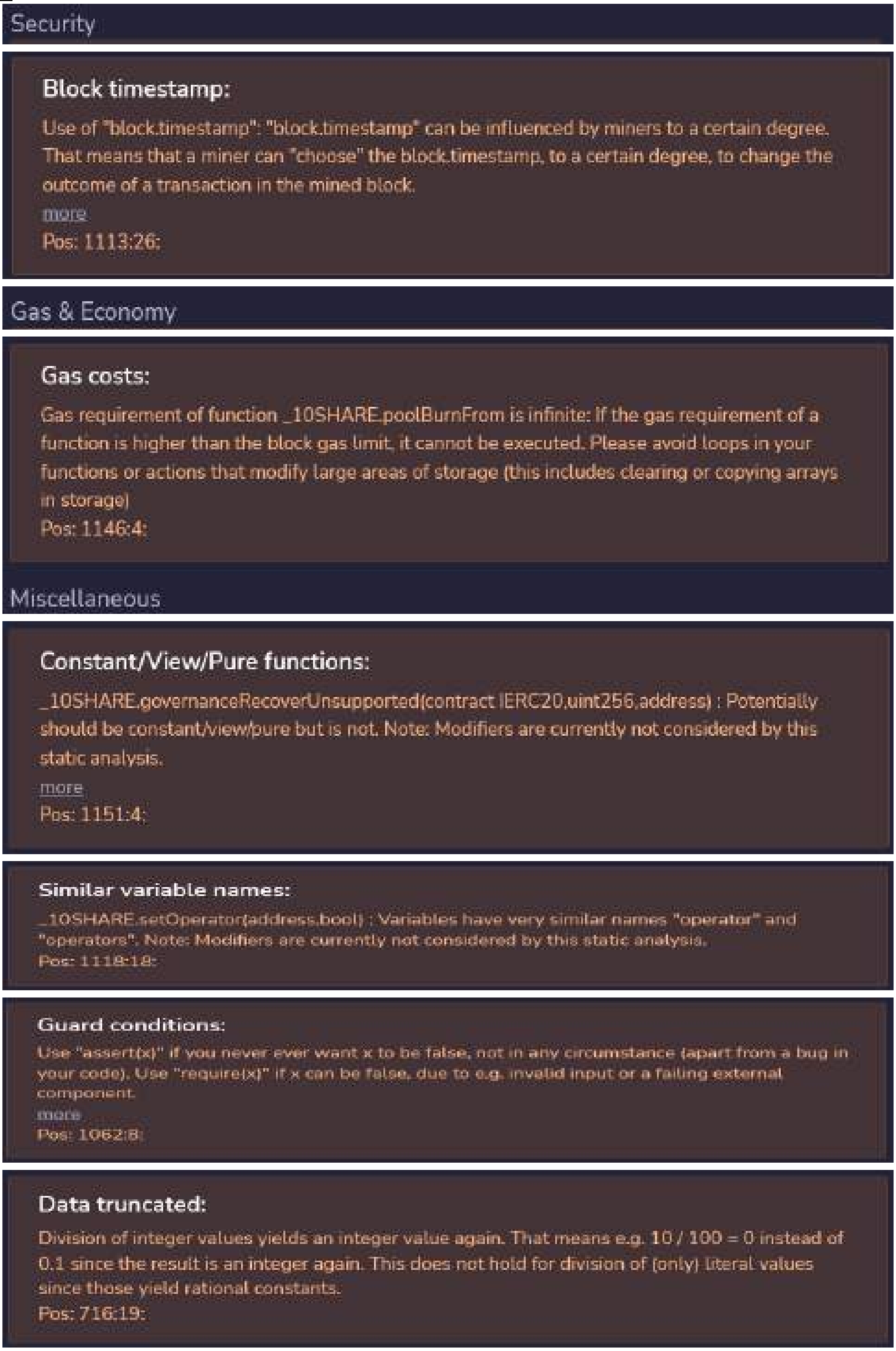
Use "assert(x)" if you never ever want x to be false, not in any circumstance (apart from a bug in your code). Use "require(x)" if x can be false, due to e.g. invalid input or a failing external component.
more
Pos: 1227:10:

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: 1087:21:

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

_10SHARE.sol



Security

Block timestamp:

Use of "block.timestamp": "block.timestamp" can be influenced by miners to a certain degree. That means that a miner can "choose" the block.timestamp, to a certain degree, to change the outcome of a transaction in the mined block.
more
Pos: 1113:26:

Gas & Economy

Gas costs:

Gas requirement of function _10SHARE.poolBurnFrom 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: 1146:4:

Miscellaneous

Constant/View/Pure functions:

_10SHARE.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: 1151:4:

Similar variable names:

_10SHARE.setOperator(address,bool) : Variables have very similar names "operator" and "operators". Note: Modifiers are currently not considered by this static analysis.
Pos: 1118:18:

Guard conditions:

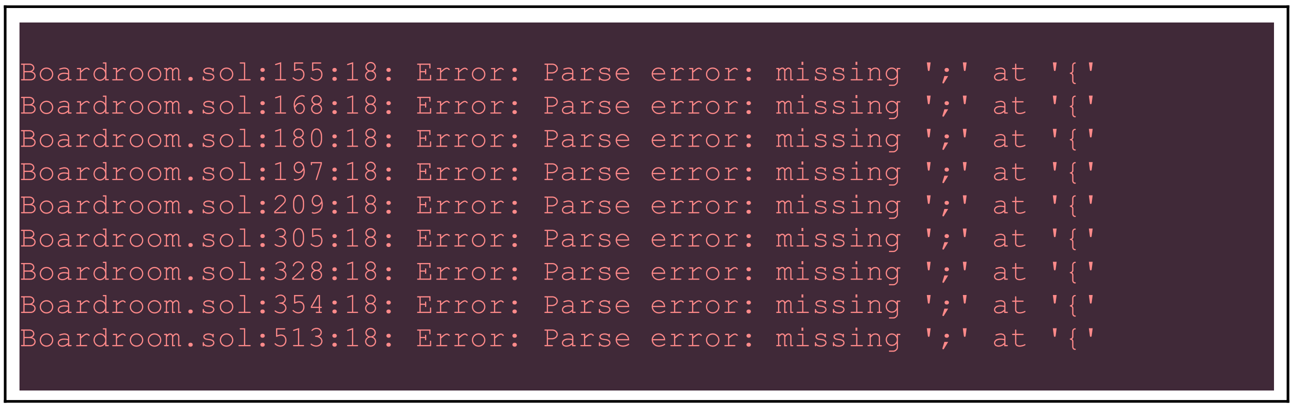
Use "assert(x)" if you never ever want x to be false, not in any circumstance (apart from a bug in your code). Use "require(x)" if x can be false, due to e.g. invalid input or a failing external component.
more
Pos: 1062:8:

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: 716:19:

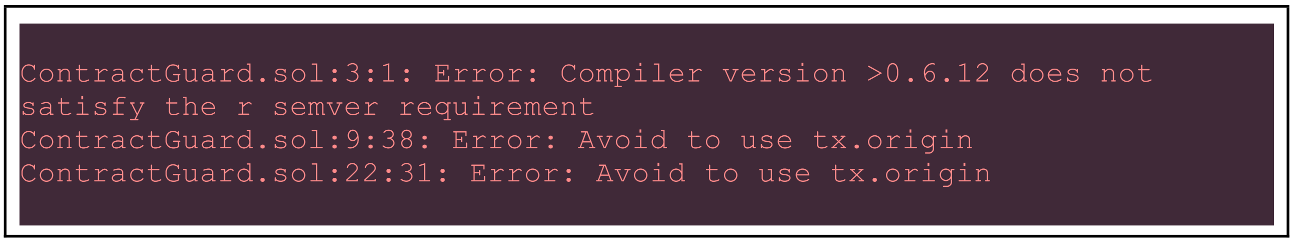
Solhint Linter

Boardroom.sol



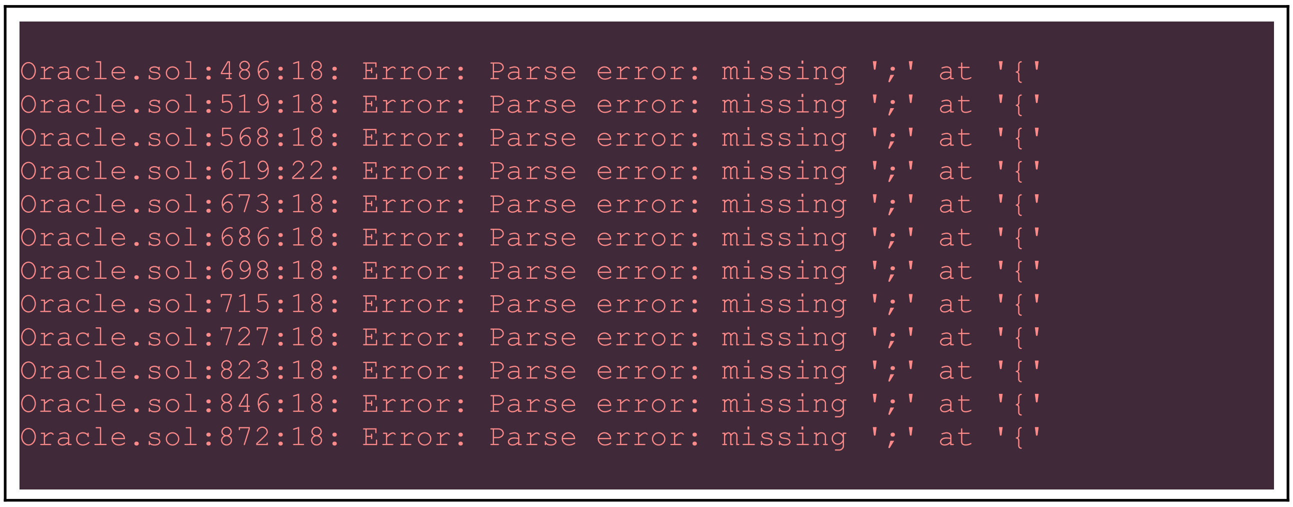
```
Boardroom.sol:155:18: Error: Parse error: missing ';' at '{'
Boardroom.sol:168:18: Error: Parse error: missing ';' at '{'
Boardroom.sol:180:18: Error: Parse error: missing ';' at '{'
Boardroom.sol:197:18: Error: Parse error: missing ';' at '{'
Boardroom.sol:209:18: Error: Parse error: missing ';' at '{'
Boardroom.sol:305:18: Error: Parse error: missing ';' at '{'
Boardroom.sol:328:18: Error: Parse error: missing ';' at '{'
Boardroom.sol:354:18: Error: Parse error: missing ';' at '{'
Boardroom.sol:513:18: Error: Parse error: missing ';' at '{'
```

ContractGuard.sol



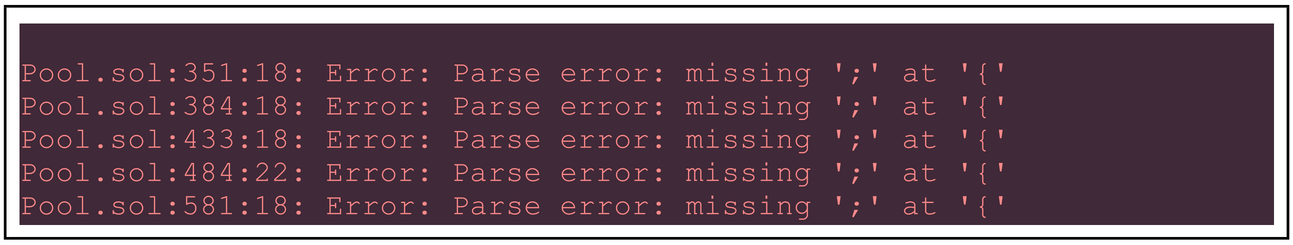
```
ContractGuard.sol:3:1: Error: Compiler version >0.6.12 does not
satisfy the r semver requirement
ContractGuard.sol:9:38: Error: Avoid to use tx.origin
ContractGuard.sol:22:31: Error: Avoid to use tx.origin
```

Oracle.sol



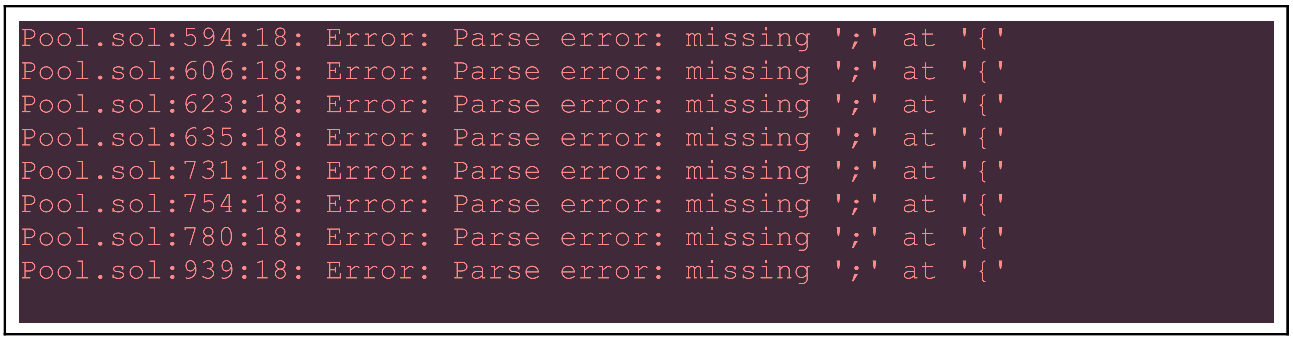
```
Oracle.sol:486:18: Error: Parse error: missing ';' at '{'
Oracle.sol:519:18: Error: Parse error: missing ';' at '{'
Oracle.sol:568:18: Error: Parse error: missing ';' at '{'
Oracle.sol:619:22: Error: Parse error: missing ';' at '{'
Oracle.sol:673:18: Error: Parse error: missing ';' at '{'
Oracle.sol:686:18: Error: Parse error: missing ';' at '{'
Oracle.sol:698:18: Error: Parse error: missing ';' at '{'
Oracle.sol:715:18: Error: Parse error: missing ';' at '{'
Oracle.sol:727:18: Error: Parse error: missing ';' at '{'
Oracle.sol:823:18: Error: Parse error: missing ';' at '{'
Oracle.sol:846:18: Error: Parse error: missing ';' at '{'
Oracle.sol:872:18: Error: Parse error: missing ';' at '{'
```

Pool.sol



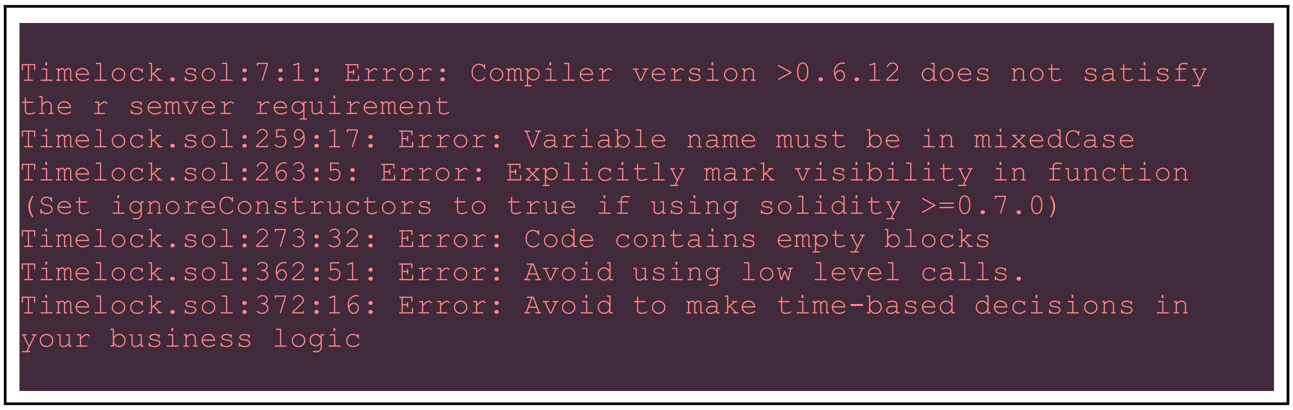
```
Pool.sol:351:18: Error: Parse error: missing ';' at '{'
Pool.sol:384:18: Error: Parse error: missing ';' at '{'
Pool.sol:433:18: Error: Parse error: missing ';' at '{'
Pool.sol:484:22: Error: Parse error: missing ';' at '{'
Pool.sol:581:18: Error: Parse error: missing ';' at '{'
```

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



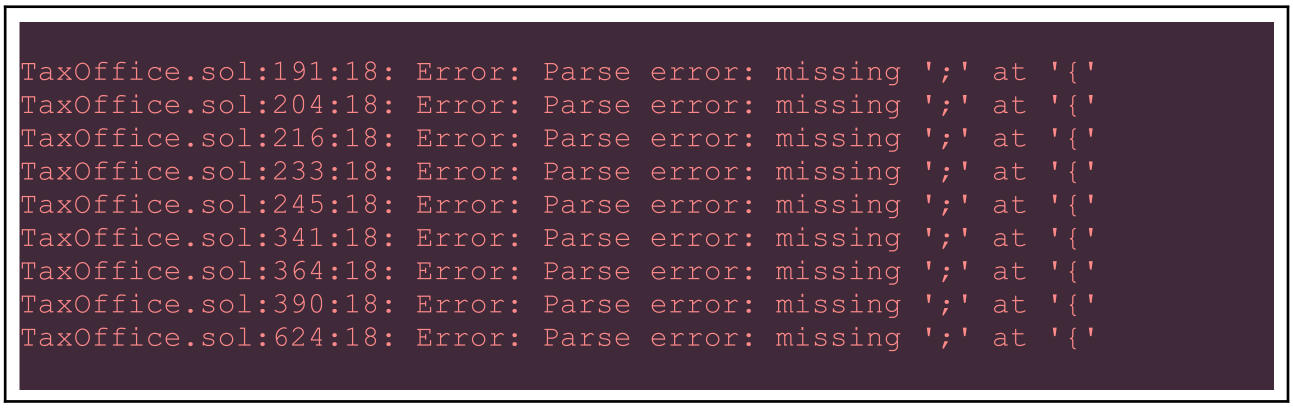
```
Pool.sol:594:18: Error: Parse error: missing ';' at '{'
Pool.sol:606:18: Error: Parse error: missing ';' at '{'
Pool.sol:623:18: Error: Parse error: missing ';' at '{'
Pool.sol:635:18: Error: Parse error: missing ';' at '{'
Pool.sol:731:18: Error: Parse error: missing ';' at '{'
Pool.sol:754:18: Error: Parse error: missing ';' at '{'
Pool.sol:780:18: Error: Parse error: missing ';' at '{'
Pool.sol:939:18: Error: Parse error: missing ';' at '{'
```

Timelock.sol



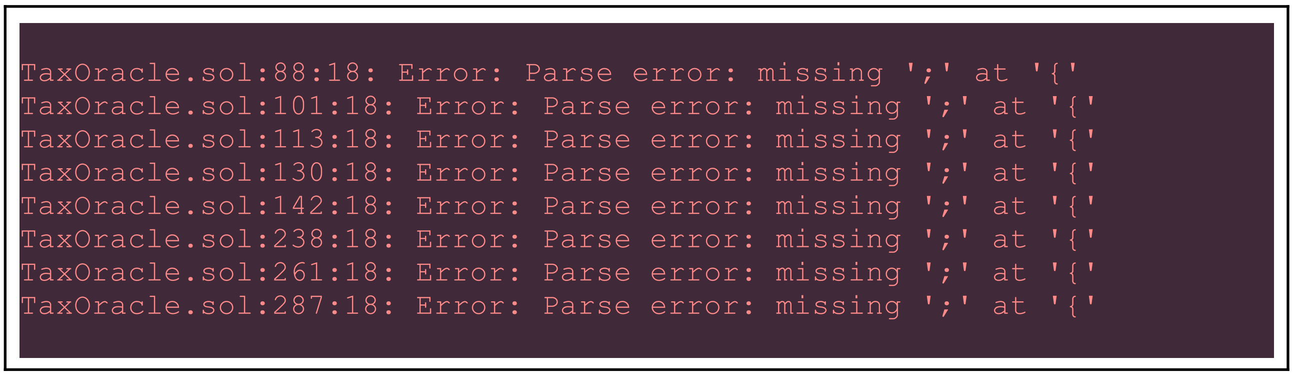
```
Timelock.sol:7:1: Error: Compiler version >0.6.12 does not satisfy
the r semver requirement
Timelock.sol:259:17: Error: Variable name must be in mixedCase
Timelock.sol:263:5: Error: Explicitly mark visibility in function
(Set ignoreConstructors to true if using solidity >=0.7.0)
Timelock.sol:273:32: Error: Code contains empty blocks
Timelock.sol:362:51: Error: Avoid using low level calls.
Timelock.sol:372:16: Error: Avoid to make time-based decisions in
your business logic
```

TaxOffice.sol



```
TaxOffice.sol:191:18: Error: Parse error: missing ';' at '{'
TaxOffice.sol:204:18: Error: Parse error: missing ';' at '{'
TaxOffice.sol:216:18: Error: Parse error: missing ';' at '{'
TaxOffice.sol:233:18: Error: Parse error: missing ';' at '{'
TaxOffice.sol:245:18: Error: Parse error: missing ';' at '{'
TaxOffice.sol:341:18: Error: Parse error: missing ';' at '{'
TaxOffice.sol:364:18: Error: Parse error: missing ';' at '{'
TaxOffice.sol:390:18: Error: Parse error: missing ';' at '{'
TaxOffice.sol:624:18: Error: Parse error: missing ';' at '{'
```

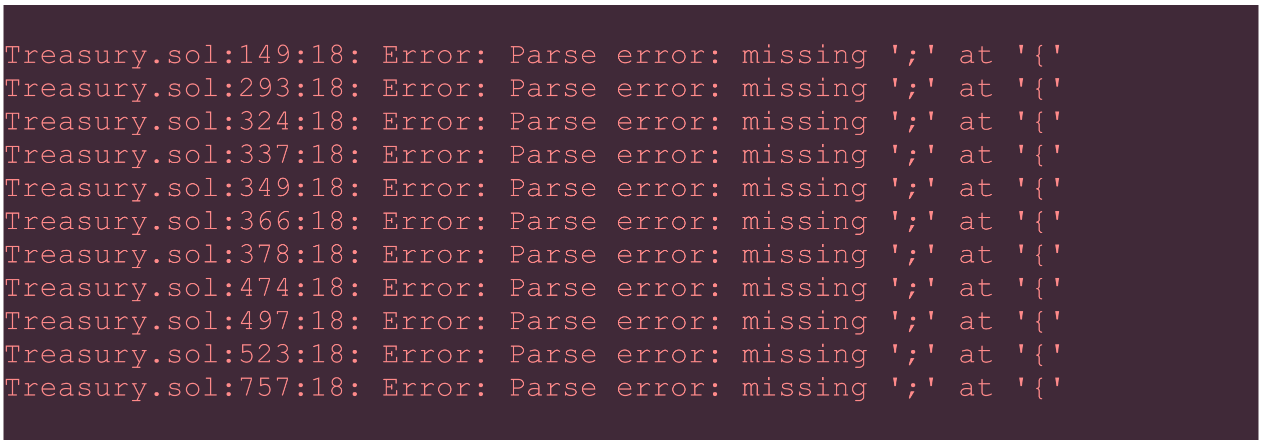
TaxOracle.sol



```
TaxOracle.sol:88:18: Error: Parse error: missing ';' at '{'
TaxOracle.sol:101:18: Error: Parse error: missing ';' at '{'
TaxOracle.sol:113:18: Error: Parse error: missing ';' at '{'
TaxOracle.sol:130:18: Error: Parse error: missing ';' at '{'
TaxOracle.sol:142:18: Error: Parse error: missing ';' at '{'
TaxOracle.sol:238:18: Error: Parse error: missing ';' at '{'
TaxOracle.sol:261:18: Error: Parse error: missing ';' at '{'
TaxOracle.sol:287:18: Error: Parse error: missing ';' at '{'
```

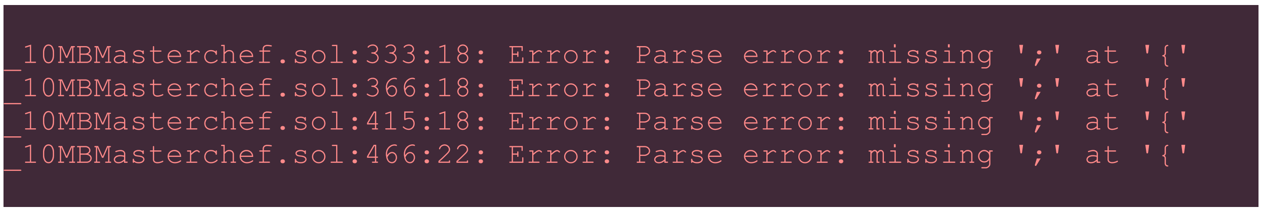
Treasury.sol

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



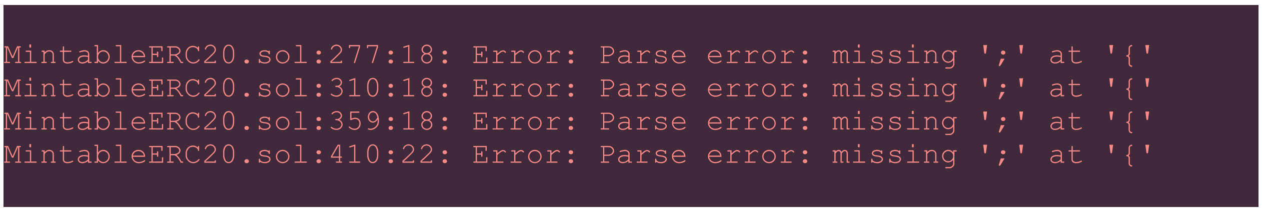
```
Treasury.sol:149:18: Error: Parse error: missing ';' at '{'
Treasury.sol:293:18: Error: Parse error: missing ';' at '{'
Treasury.sol:324:18: Error: Parse error: missing ';' at '{'
Treasury.sol:337:18: Error: Parse error: missing ';' at '{'
Treasury.sol:349:18: Error: Parse error: missing ';' at '{'
Treasury.sol:366:18: Error: Parse error: missing ';' at '{'
Treasury.sol:378:18: Error: Parse error: missing ';' at '{'
Treasury.sol:474:18: Error: Parse error: missing ';' at '{'
Treasury.sol:497:18: Error: Parse error: missing ';' at '{'
Treasury.sol:523:18: Error: Parse error: missing ';' at '{'
Treasury.sol:757:18: Error: Parse error: missing ';' at '{'
```

_10MBMasterchef.sol



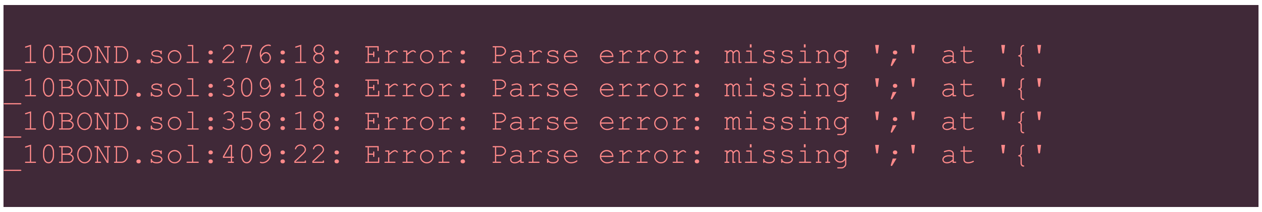
```
_10MBMasterchef.sol:333:18: Error: Parse error: missing ';' at '{'
_10MBMasterchef.sol:366:18: Error: Parse error: missing ';' at '{'
_10MBMasterchef.sol:415:18: Error: Parse error: missing ';' at '{'
_10MBMasterchef.sol:466:22: Error: Parse error: missing ';' at '{'
```

MintableERC20.sol



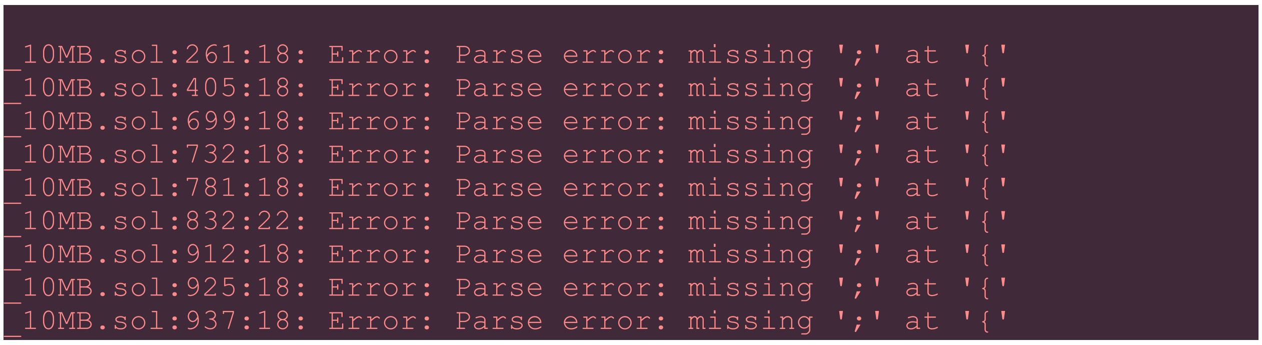
```
MintableERC20.sol:277:18: Error: Parse error: missing ';' at '{'
MintableERC20.sol:310:18: Error: Parse error: missing ';' at '{'
MintableERC20.sol:359:18: Error: Parse error: missing ';' at '{'
MintableERC20.sol:410:22: Error: Parse error: missing ';' at '{'
```

_10BOND.sol



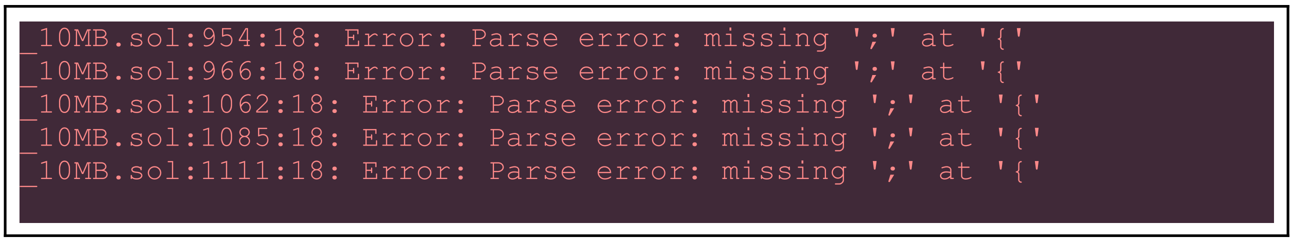
```
_10BOND.sol:276:18: Error: Parse error: missing ';' at '{'
_10BOND.sol:309:18: Error: Parse error: missing ';' at '{'
_10BOND.sol:358:18: Error: Parse error: missing ';' at '{'
_10BOND.sol:409:22: Error: Parse error: missing ';' at '{'
```

_10MB.sol



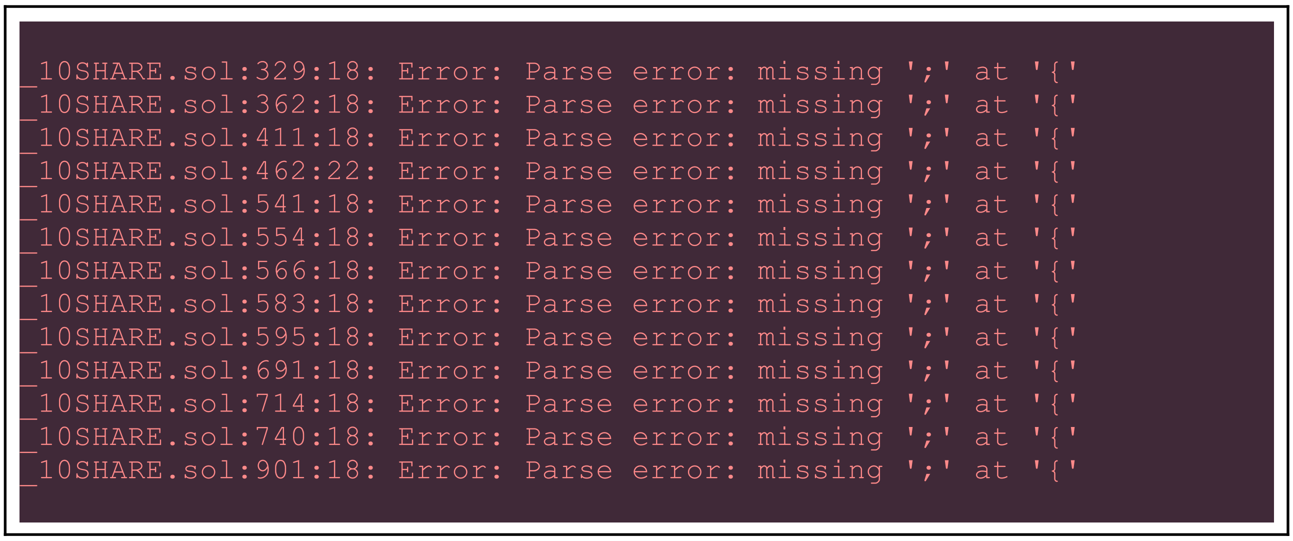
```
_10MB.sol:261:18: Error: Parse error: missing ';' at '{'
_10MB.sol:405:18: Error: Parse error: missing ';' at '{'
_10MB.sol:699:18: Error: Parse error: missing ';' at '{'
_10MB.sol:732:18: Error: Parse error: missing ';' at '{'
_10MB.sol:781:18: Error: Parse error: missing ';' at '{'
_10MB.sol:832:22: Error: Parse error: missing ';' at '{'
_10MB.sol:912:18: Error: Parse error: missing ';' at '{'
_10MB.sol:925:18: Error: Parse error: missing ';' at '{'
_10MB.sol:937:18: Error: Parse error: missing ';' at '{'
```

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



```
_10MB.sol:954:18: Error: Parse error: missing ';' at '{'
_10MB.sol:966:18: Error: Parse error: missing ';' at '{'
_10MB.sol:1062:18: Error: Parse error: missing ';' at '{'
_10MB.sol:1085:18: Error: Parse error: missing ';' at '{'
_10MB.sol:1111:18: Error: Parse error: missing ';' at '{'
```

_10SHARE.sol



```
_10SHARE.sol:329:18: Error: Parse error: missing ';' at '{'
_10SHARE.sol:362:18: Error: Parse error: missing ';' at '{'
_10SHARE.sol:411:18: Error: Parse error: missing ';' at '{'
_10SHARE.sol:462:22: Error: Parse error: missing ';' at '{'
_10SHARE.sol:541:18: Error: Parse error: missing ';' at '{'
_10SHARE.sol:554:18: Error: Parse error: missing ';' at '{'
_10SHARE.sol:566:18: Error: Parse error: missing ';' at '{'
_10SHARE.sol:583:18: Error: Parse error: missing ';' at '{'
_10SHARE.sol:595:18: Error: Parse error: missing ';' at '{'
_10SHARE.sol:691:18: Error: Parse error: missing ';' at '{'
_10SHARE.sol:714:18: Error: Parse error: missing ';' at '{'
_10SHARE.sol:740:18: Error: Parse error: missing ';' at '{'
_10SHARE.sol:901:18: 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