Ether Authority

www.EtherAuthority.io
audit@etherauthority.io

SMART CONTRACT

Security Audit Report

Project: Carbon XYZ Protocol
Platform: Polygon Network
Language: Solidity
Date: April 13th, 2022

Table of contents

Email: audit@EtherAuthority.io

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

Email: audit@EtherAuthority.io

Introduction

EtherAuthority was contracted by the Carbon XYZ team to perform the Security audit of the Carbon XYZ Protocol smart contracts code. The audit has been performed using manual analysis as well as using automated software tools. This report presents all the findings regarding the audit performed on April 13th, 2022.

The purpose of this audit was to address the following:

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

Project Background

The Carbon XYZ Contracts have functions like mint, burn, burnNFT, setStakingPool, mintNewNFT, setMembershipTrader, withdrawGEMS, etc. The Carbon XYZ contract inherits the AccessControl, ERC721, ERC721URIStorage, Counters, Strings, IERC20, Address, Pausable, SafeMath, ReentrancyGuard, Ownable standard smart contracts from the OpenZeppelin library. These OpenZeppelin contracts are considered community-audited and time-tested, and hence are not part of the audit scope.

Audit scope

Name	**Code Review and Security Analysis Report for Carbon XYZ Protocol Smart Contracts**
Platform	**Polygon / Solidity**
File 1	ETHToken.sol
File 1 MD5 Hash	676697E70A6EDDC5EBAA03BBCA7D4485
File 2	AdminRole.sol
File 2 MD5 Hash	C6750B6B82A5EEC557ACCBD0DA2F5143
File 3	GEMSNFTReceipt.sol
File 3 MD5 Hash	61535FA82782A82B460BF994D17703F2
File 4	GEMSStaking.sol

File 4 MD5 Hash	9D291FD0A77297B37E5D07D56F1EB8E2
Updated File 4 MD5 Hash	F44B5A5C1C9148E443ABD5730398990F
File 5	GEMSToken.sol
File 5 MD5 Hash	8B8C64C769FCD7DA5C34DB92D7BD67D2
File 6	CarbonMembership.sol
File 6 MD5 Hash	20F74964B3429F940D633BD60F91E0DA
File 7	MembershipTrader.sol
File 7 MD5 Hash	3AE63557743E8F68B9E522A1F6A5B14A
Updated File 7 MD5 Hash	C98A22C4830C6A939C548945CDD44A8B
File 8	ERC721NFTContract.sol
File 8 MD5 Hash	ED86B14B26BAC3EEA6C09FF16DEB5698
File 9	MintingFactory.sol
File 9 MD5 Hash	A5ADA1951E6E32AD46F01CF59E96E300
File 10	ExchangeCore.sol
File 10 MD5 Hash	5768D2999994906DA1C9C1645EBD507F
Updated File 10 MD5 Hash	D38BDA19B377DAC76895A034AD10328A
Audit Date	April 13th,2022
Revise Audit Date	April 16th,2022

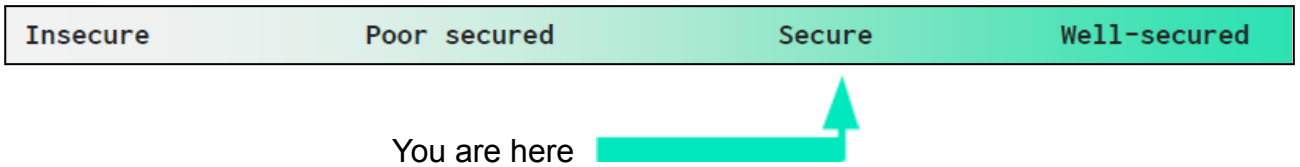
Claimed Smart Contract Features

Claimed Feature Detail	Our Observation
File 1 ETHToken.sol • Name: ETH Token • Symbol: ETH • Decimals: 18	**YES, This is valid.**
File 2 AdminRole.sol • AdminRole contract has functions like: isAdmin, addAdmin, leaveRole.	**YES, This is valid.**
File 3 GEMSNFTReceipt.sol • BaseURI: https://carbon.xyz • The GEMSNFTReceipt admin can set a staking pool.	**YES, This is valid.**
File 4 GEMSStaking.sol • Decimals: 18 • Tokens To Stake: 1,00,000 tokens for staking • The GEMSStaking contract has functions like: unstake	**YES, This is valid.**
File 5 GEMSToken.sol • Name: GEMS Token • Symbol: GEMS • Decimals: 18 • Total Supply: 1 Billion	**YES, This is valid.**
File 6 CarbonMembership.sol • Name: Carbon Membership Pass • Symbol: CMEM • BaseURI: https://carbon.xyz	**YES, This is valid.**
File 7 MembershipTrader.sol • Tokens to Deposit: 1,00,000 • The MembershipTrader contract has functions like: validate, executeOrder, withdrawGEMS.	**YES, This is valid.**

File 8 ERC721NFTContract.sol • BaseURI: https://carbon.xyz • The Factory can mint NFT Tokens.	**YES, This is valid.**
File 9 MintingFactory.sol • The MintingFactory contract creates an NFT contract and then it can mint NFT for that contract to keep track of all NFT contracts for the users.	**YES, This is valid.**
File 10 ExchangeCore.sol • Base Factor Maximum: 1025 • Buyers premium fees: 25	**YES, This is valid.**

Audit Summary

According to the standard audit assessment, Customer`s solidity smart contracts are **"Secured"**. 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 0 low and some very low level issues.
All these issues have been resolved / acknowledged.

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

Technical Quick Stats

Main Category	Subcategory	Result
Contract Programming	Solidity version not specified	Passed
	Solidity version too old	Passed
	Integer overflow/underflow	Passed
	Function input parameters lack of check	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	Passed
	“Short Address” Attack	Passed
	“Double Spend” Attack	Passed

Overall Audit Result: **PASSED**

Code Quality

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

The libraries in the Carbon XYZ 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 Carbon XYZ Protocol.

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

Code parts are **not** well commented on smart contracts.

Documentation

We were given a Carbon XYZ Protocol smart contract code in the form files. The hash of that code is mentioned above in the table.

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

Use of Dependencies

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

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

AS-IS overview

ETHToken.sol

Functions

Sl.	Functions	Type	Observation	Conclusion
1	constructor	write	Passed	No Issue
2	name	read	Passed	No Issue
3	symbol	read	Passed	No Issue
4	decimals	read	Passed	No Issue
5	totalSupply	read	Passed	No Issue
6	balanceOf	read	Passed	No Issue
7	transfer	write	Passed	No Issue
8	allowance	read	Passed	No Issue
9	approve	write	Passed	No Issue
10	transferFrom	write	Passed	No Issue
11	increaseAllowance	write	Passed	No Issue
12	decreaseAllowance	write	Passed	No Issue
13	_transfer	internal	Passed	No Issue
14	_mint	internal	Passed	No Issue
15	_burn	internal	Passed	No Issue
16	_approve	internal	Passed	No Issue
17	_spendAllowance	internal	Passed	No Issue

AdminRole.sol

Functions

Sl.	Functions	Type	Observation	Conclusion
1	constructor	write	Passed	No Issue
2	onlyRole	modifier	Passed	No Issue
3	supportsInterface	read	Passed	No Issue
4	hasRole	read	Passed	No Issue
5	_checkRole	internal	Passed	No Issue
6	_checkRole	internal	Passed	No Issue
7	getRoleAdmin	read	Passed	No Issue
8	grantRole	write	Passed	No Issue
9	revokeRole	write	Passed	No Issue
10	renounceRole	write	Passed	No Issue
11	_setupRole	internal	Passed	No Issue
12	_setRoleAdmin	internal	Passed	No Issue
13	_grantRole	internal	Passed	No Issue
14	_revokeRole	internal	Passed	No Issue
15	onlyAdmin	modifier	Passed	No Issue
16	isAdmin	internal	Passed	No Issue
17	addAdmin	external	access only Admin	No Issue

GEMSNFTReceipt.sol

Functions

Sl.	Functions	Type	Observation	Conclusion
1	constructor	write	Passed	No Issue
2	supportsInterface	write	Passed	No Issue
3	balanceOf	write	Passed	No Issue
4	ownerOf	write	Passed	No Issue
5	name	write	Passed	No Issue
6	symbol	write	Passed	No Issue
7	tokenURI	write	Passed	No Issue
8	_baseURI	internal	Passed	No Issue
9	approve	write	Passed	No Issue
10	getApproved	read	Passed	No Issue
11	setApprovalForAll	write	Passed	No Issue
12	isApprovedForAll	read	Passed	No Issue
13	transferFrom	write	Passed	No Issue
14	safeTransferFrom	write	Passed	No Issue
15	safeTransferFrom	write	Passed	No Issue
16	_safeTransfer	internal	Passed	No Issue
17	_exists	internal	Passed	No Issue
18	_isApprovedOrOwner	internal	Passed	No Issue
19	_safeMint	internal	Passed	No Issue
20	_safeMint	internal	Passed	No Issue
21	_mint	internal	Passed	No Issue
22	_burn	internal	Passed	No Issue
23	_transfer	internal	Passed	No Issue
24	_approve	internal	Passed	No Issue
25	_setApprovalForAll	internal	Passed	No Issue
26	_checkOnERC721Received	write	Passed	No Issue
27	_beforeTokenTransfer	internal	Passed	No Issue
28	_afterTokenTransfer	internal	Passed	No Issue
29	tokenURI	read	Passed	No Issue
30	_setTokenURI	internal	Passed	No Issue
31	_burn	internal	Passed	No Issue
32	onlyAuthorised	modifier	Passed	No Issue
33	mintNewNFT	write	access only Authorized	No Issue
34	getTotalNFTs	read	Passed	No Issue
35	burnNFT	write	Passed	No Issue
36	setStakingPool	write	access only Authorized	No Issue

GEMSStaking.sol

Functions

Sl.	Functions	Type	Observation	Conclusion
1	constructor	write	Passed	No Issue
2	onlyAuthorised	modifier	Passed	No Issue
3	mintNewNFT	write	access only Authorized	No Issue
4	getTotalNFTs	read	Passed	No Issue
5	burnNFT	write	Passed	No Issue
6	setStakingPool	write	access only Authorized	No Issue
7	stake	write	Passed	No Issue
8	unstake	write	Passed	No Issue

GEMSToken.sol

Functions

Sl.	Functions	Type	Observation	Conclusion
1	constructor	write	Passed	No Issue
2	name	read	Passed	No Issue
3	symbol	read	Passed	No Issue
4	decimals	read	Passed	No Issue
5	totalSupply	read	Passed	No Issue
6	balanceOf	read	Passed	No Issue
7	transfer	write	Passed	No Issue
8	allowance	read	Passed	No Issue
9	approve	write	Passed	No Issue
10	transferFrom	write	Passed	No Issue
11	increaseAllowance	write	Passed	No Issue
12	decreaseAllowance	write	Passed	No Issue
13	_transfer	internal	Passed	No Issue
14	_mint	internal	Passed	No Issue
15	_burn	internal	Passed	No Issue
16	_approve	internal	Passed	No Issue
17	_spendAllowance	internal	Passed	No Issue

CarbonMembership.sol

Functions

Sl.	Functions	Type	Observation	Conclusion
1	constructor	write	Unused constructor parameters	Refer Audit Findings
2	supportsInterface	write	Passed	No Issue
3	balanceOf	write	Passed	No Issue
4	ownerOf	write	Passed	No Issue

5	name	write	Passed	No Issue
6	symbol	write	Passed	No Issue
7	tokenURI	write	Passed	No Issue
8	_baseURI	internal	Passed	No Issue
9	approve	write	Passed	No Issue
10	getApproved	read	Passed	No Issue
11	setApprovalForAll	write	Passed	No Issue
12	isApprovedForAll	read	Passed	No Issue
13	transferFrom	write	Passed	No Issue
14	safeTransferFrom	write	Passed	No Issue
15	safeTransferFrom	write	Passed	No Issue
16	_safeTransfer	internal	Passed	No Issue
17	_exists	internal	Passed	No Issue
18	_isApprovedOrOwner	internal	Passed	No Issue
19	_safeMint	internal	Passed	No Issue
20	_safeMint	internal	Passed	No Issue
21	_mint	internal	Passed	No Issue
22	_burn	internal	Passed	No Issue
23	_transfer	internal	Passed	No Issue
24	_approve	internal	Passed	No Issue
25	_setApprovalForAll	internal	Passed	No Issue
26	_checkOnERC721Received	write	Passed	No Issue
27	_beforeTokenTransfer	internal	Passed	No Issue
28	_afterTokenTransfer	internal	Passed	No Issue
29	tokenURI	read	Passed	No Issue
30	_setTokenURI	internal	Passed	No Issue
31	_burn	internal	Passed	No Issue
32	owner	read	Passed	No Issue
33	onlyOwner	modifier	Passed	No Issue
34	renounceOwnership	write	access only Owner	No Issue
35	transferOwnership	write	access only Owner	No Issue
36	_transferOwnership	internal	Passed	No Issue
37	paused	read	Passed	No Issue
38	whenNotPaused	modifier	Passed	No Issue
39	whenPaused	modifier	Passed	No Issue
40	_pause	internal	Passed	No Issue
41	_unpause	internal	Passed	No Issue
42	onlyMembershipTrader	modifier	Passed	No Issue
43	mintNewNFT	write	access only Membership Trader	No Issue
44	setMembershipTrader	write	access only Owner	No Issue
45	pause	write	access only Owner	No Issue
46	unpause	write	access only Owner	No Issue
47	updateOwner	write	access only Owner	No Issue

MembershipTrader.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		No Issue
7	validate	internal	Passed	No Issue
8	executeOrder	write	Passed	No Issue
9	withdrawGEMS	write	access only Owner	No Issue
10	updateOwner	write	access only Owner	No Issue

ERC721NFTContract.sol

Functions

Sl.	Functions	Type	Observation	Conclusion
1	constructor	write	Passed	No Issue
2	supportsInterface	write	Passed	No Issue
3	balanceOf	write	Passed	No Issue
4	ownerOf	write	Passed	No Issue
5	name	write	Passed	No Issue
6	symbol	write	Passed	No Issue
7	tokenURI	write	Passed	No Issue
8	_baseURI	internal	Passed	No Issue
9	approve	write	Passed	No Issue
10	getApproved	read	Passed	No Issue
11	setApprovalForAll	write	Passed	No Issue
12	isApprovedForAll	read	Passed	No Issue
13	transferFrom	write	Passed	No Issue
14	safeTransferFrom	write	Passed	No Issue
15	safeTransferFrom	write	Passed	No Issue
16	_safeTransfer	internal	Passed	No Issue
17	_exists	internal	Passed	No Issue
18	_isApprovedOrOwner	internal	Passed	No Issue
19	_safeMint	internal	Passed	No Issue
20	_safeMint	internal	Passed	No Issue
21	_mint	internal	Passed	No Issue
22	_burn	internal	Passed	No Issue
23	_transfer	internal	Passed	No Issue
24	_approve	internal	Passed	No Issue

25	_setApprovalForAll	internal	Passed	No Issue
26	_checkOnERC721Received	write	Passed	No Issue
27	_beforeTokenTransfer	internal	Passed	No Issue
28	_afterTokenTransfer	internal	Passed	No Issue
29	tokenURI	read	Passed	No Issue
30	_setTokenURI	internal	Passed	No Issue
31	_burn	internal	Passed	No Issue
32	onlyFactory	modifier	Passed	No Issue
33	onlyAdmin	modifier	Passed	No Issue
34	mint	write	access only Factory	No Issue
35	getTotalNFTs	read	Passed	No Issue
36	changeAdmin	write	access only Admin	No Issue
37	updateFactory	external	access only Admin	No Issue

MintingFactory.sol

Functions

Sl.	Functions	Type	Observation	Conclusion
1	constructor	write	Passed	No Issue
2	onlyRole	modifier	Passed	No Issue
3	supportsInterface	read	Passed	No Issue
4	hasRole	read	Passed	No Issue
5	_checkRole	internal	Passed	No Issue
6	_checkRole	internal	Passed	No Issue
7	getRoleAdmin	read	Passed	No Issue
8	grantRole	write	Passed	No Issue
9	revokeRole	write	Passed	No Issue
10	renounceRole	write	Passed	No Issue
11	_setupRole	internal	Passed	No Issue
12	_setRoleAdmin	internal	Passed	No Issue
13	_grantRole	internal	Passed	No Issue
14	_revokeRole	internal	Passed	No Issue
15	onlyAdmin	modifier	Passed	No Issue
16	isAdmin	internal	Passed	No Issue
17	addAdmin	external	access only Admin	No Issue
18	onlyCreatorAdmin	modifier	Passed	No Issue
19	onlyExchange	modifier	Passed	No Issue
20	createNFTContract	external	access only Admin	No Issue
21	mintNFT	write	access only Creator Admin	No Issue
22	updateOwner	write	access only Exchange	No Issue
23	updateExchangeAddress	write	access only Admin	No Issue
24	getNFTsForOwner	read	Passed	No Issue

25	getTotalNFTsMinted	read	Passed	No Issue
26	transferFunds	external	Possibility to transfer fund to zero address	Refer Audit Findings
27	setCarbonMintingFactory FeeVault	external	access only Admin	No Issue

ExchangeCore.sol

Functions

Sl.	Functions	Type	Observation	Conclusion
1	constructor	write	Passed	No Issue
2	onlyRole	modifier	Passed	No Issue
3	supportsInterface	read	Passed	No Issue
4	hasRole	read	Passed	No Issue
5	_checkRole	internal	Passed	No Issue
6	_checkRole	internal	Passed	No Issue
7	getRoleAdmin	read	Passed	No Issue
8	grantRole	write	Passed	No Issue
9	revokeRole	write	Passed	No Issue
10	renounceRole	write	Passed	No Issue
11	_setupRole	internal	Passed	No Issue
12	_setRoleAdmin	internal	Passed	No Issue
13	_grantRole	internal	Passed	No Issue
14	_revokeRole	internal	Passed	No Issue
15	onlyAdmin	modifier	Passed	No Issue
16	isAdmin	internal	Passed	No Issue
17	addAdmin	external	access only Admin	No Issue
18	paused	read	Passed	No Issue
19	whenNotPaused	modifier	Passed	No Issue
20	whenPaused	modifier	Passed	No Issue
21	_pause	internal	Passed	No Issue
22	_unpause	internal	Passed	No Issue
23	nonReentrant	modifier	Passed	No Issue
24	validateSeller	internal	Passed	No Issue
25	validateBuyer	internal	Passed	No Issue
26	executeOrder	write	access only Admin	No Issue
27	_executeOrder	internal	Passed	No Issue
28	cancelOrder	write	access only Admin	No Issue
29	uncancelOrder	write	access only Admin	No Issue
30	updateFactory	external	access only Admin	No Issue
31	setCarbonFeeVaultAddress	external	Passed	No Issue
32	pause	write	access only Admin	No Issue
33	unpause	write	access only Admin	No Issue

Severity Definitions

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

Email: audit@EtherAuthority.io

Audit Findings

Critical Severity

No Critical severity vulnerabilities were found.

High Severity

No High severity vulnerabilities were found.

Medium

(1) User cannot stake after unstake: **GEMSStaking.sol**
Once an user unstakes his tokens, he cannot stake the tokens again.

Resolution: We suggest correcting this.
Status: Fixed

Low

No Low severity vulnerabilities were found.

Very Low / Informational / Best practices:

(1) Possibility to transfer fund to zero address: **MintingFactory.sol**

```
function transferFunds() external onlyAdmin {
    uint256 totalBalance = IERC20(ETH).balanceOf(address(this));
    IERC20(ETH).transfer(carbonMintingFactoryFeeVault, totalBalance);
}
```

The transferFunds function is used to transfer ETH tokens to "carbonMintingFactoryFeeVault" without checking whether it is set to some address or not.
.

Resolution: We suggest validating whether "carbonMintingFactoryFeeVault" has been set or not before transfer funds.

(2) Unused constructor parameters: **CarbonMembership.sol**

```
constructor(string memory _name, string memory _symbol)
    ERC721("Carbon Membership Pass", "CMEM")
{}
```

In the constructor _name and _symbol parameters are defined, but not used in the functionality.

.

Resolution: We suggest removing unused parameters from the constructor.

(3) Make variables constant: **ExchangeCore.sol**

```
uint256 public BUYERS_PREMIUM_FEES = 25; // 2.5%
```

These variables will be unchanged. So, please make it constant. It will save some gas.

.

Resolution: Declare those variables as constant. Just put a constant keyword.

(4) Variable should be immutable:

Variables that are defined within the constructor but further remain unchanged should be marked as immutable to save gas and to ease the reviewing process of third-parties.

Variables are:

- **GEMSStaking.sol**
  - GEMSToken
  - GEMSNFTAddress
- **GEMSNFTReceipt.sol**
  - admin
- **MintingFactory.sol**
  - ETH
- **ExchangeCore.sol**
  - ETH
  - carbonMembership

Resolution: Consider marking this variable as immutable.

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:

- mintNewNFT: The GEMSNFTReceipt Authorise can mint new NFT.
- setStakingPool: The GEMSNFTReceipt Authorise can set a staking pool.
- changeAdmin: The ERC721NFTContract admin can update a new admin address.
- updateFactory: The ERC721NFTContract admin can update factory addresses.
- createNFTContract: The MintingFactory admin can create a new NFT contract.
- mintNFT: The MintingFactory creator admin can mint NFT tokens.
- updateExchangeAddress: The MintingFactory admin can update the exchange address.
- transferFunds: The MintingFactory admin can transfer funds.
- setCarbonMintingFactoryFeeVault: The MintingFactory admin can set carbon minting factory fee vault address.
- setMembershipTrader: The CarbonMembership owner can set membership trader address.
- pause: The CarbonMembership owner can trigger a stopped state.
- unpause: The CarbonMembership owner can return to normal state.
- updateOwner: The CarbonMembership owner can update the new owner address.
- withdrawGEMS: The MembershipTrader owner can withdraw GEMS.
- updateOwner: The MembershipTrader owner can update the new owner address.
- executeOrder: The Exchange core owner can execute orders.
- cancelOrder: The Exchange core owner can cancel orders.
- uncancelOrder: The Exchange core owner can uncancel orders.
- updateFactory: The Exchange core owner can update the factory address.
- setCarbonFeeVaultAddress: The Exchange core owner can set a carbon fee vault address.
- pause: The ExchangeCore owner can trigger a stopped state.
- unpause: The ExchangeCore owner can return to normal state.

- createNFTContract: The MintingFactory admin can create NFT contracts.
- mintNFT: The MintingFactory admin can mint NFT tokens.
- updateExchangeAddress: The MintingFactory admin can update the exchange address.
- transferFunds: The MintingFactory admin can transfer funds.
- setCarbonMintingFactoryFeeVault: The MintingFactory admin can set carbon minting factory fee vault address.

Conclusion

We were given a contract code in the form of files. And we have used all possible tests based on given objects as files. We have not observed any major issues in the smart contracts. So, **it's good to go to production**.

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"**.

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.

Documenting Results:

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

Suggested Solutions:

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

Disclaimers

EtherAuthority.io Disclaimer

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

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

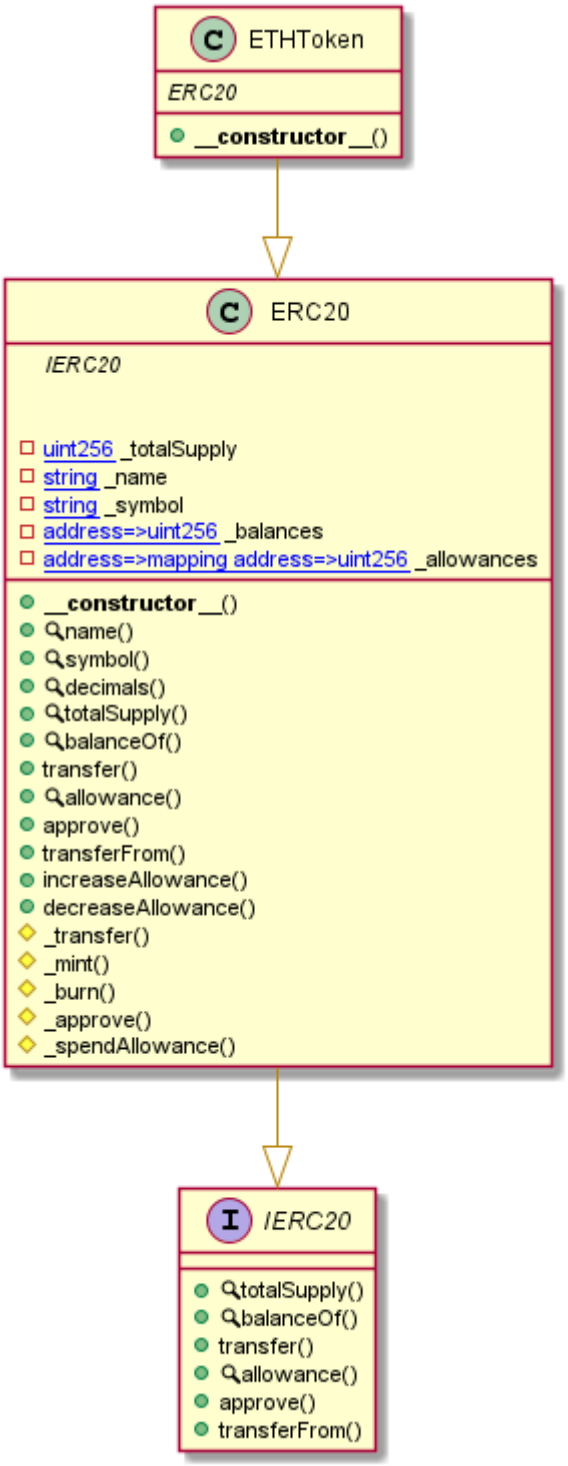
Technical Disclaimer

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

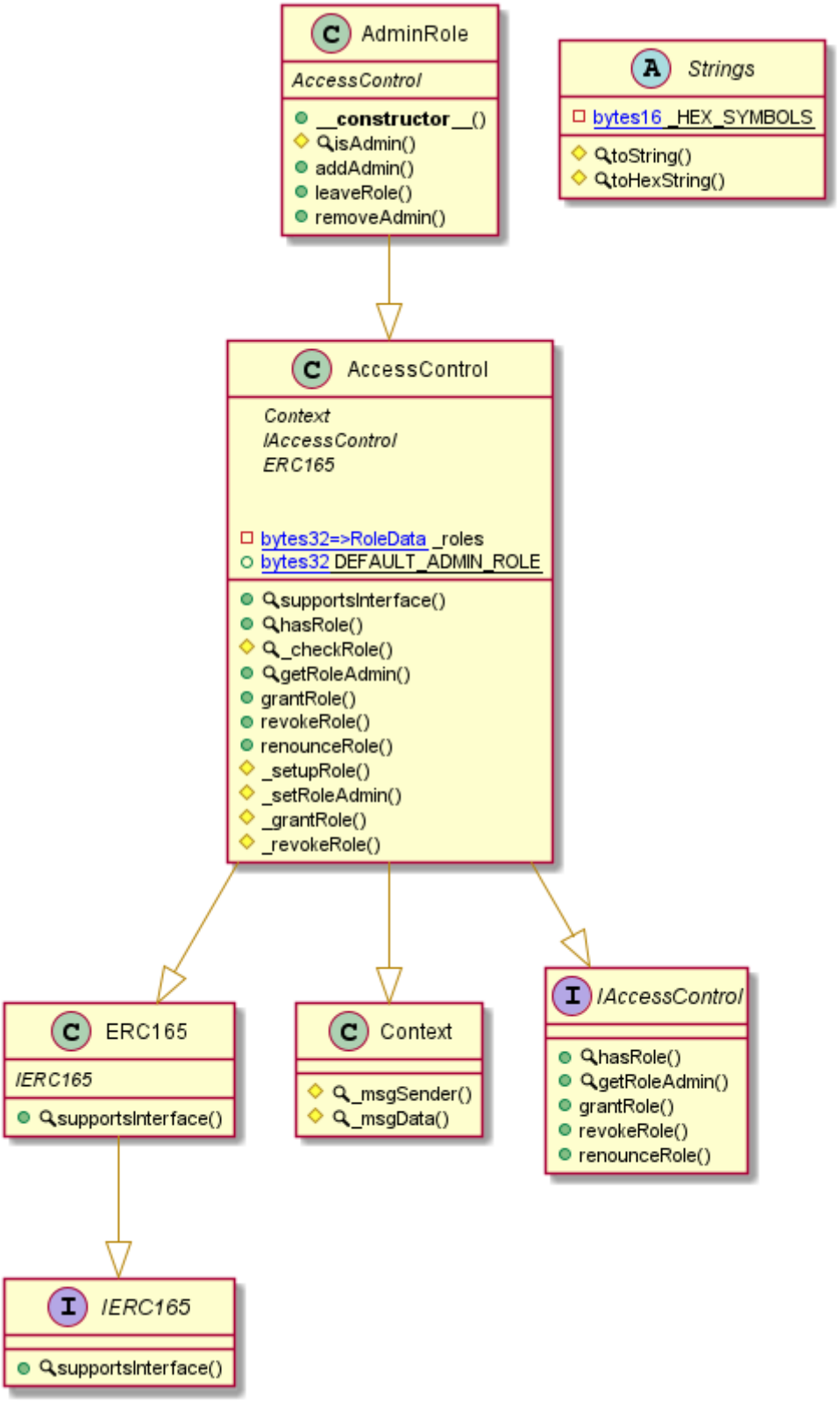
Appendix

Code Flow Diagram - Carbon XYZ Protocol

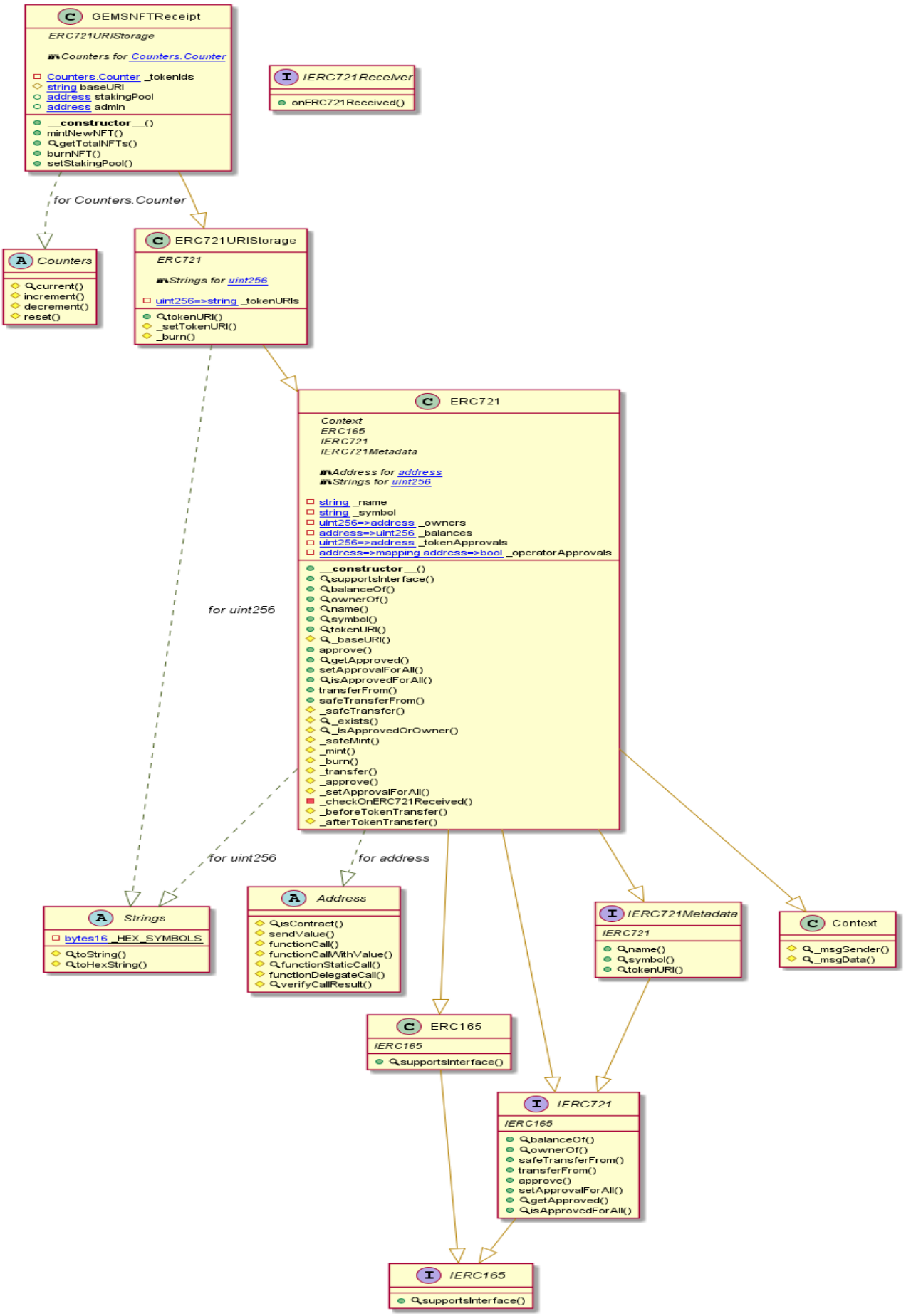
ETHToken Diagram



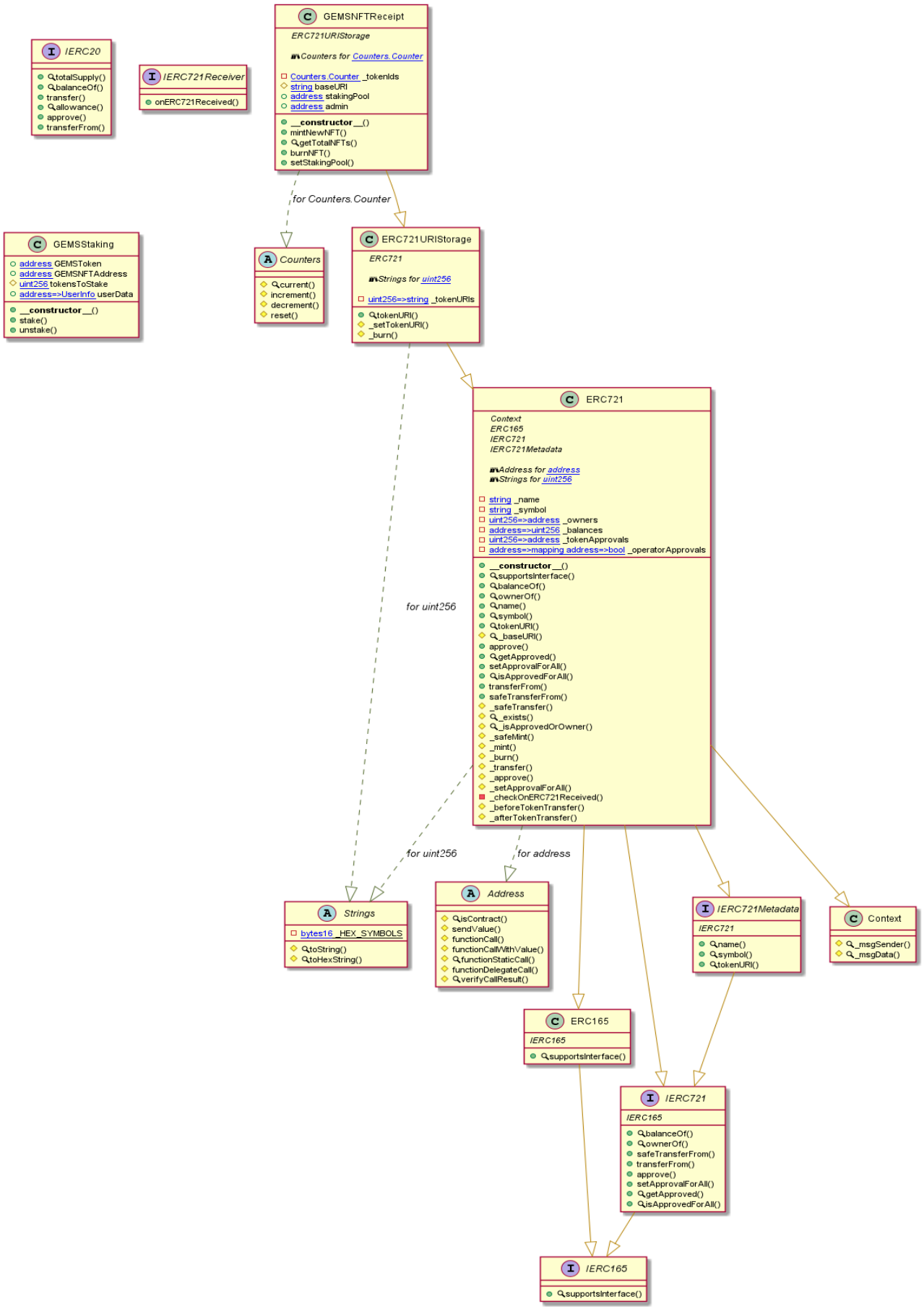
AdminRole Diagram



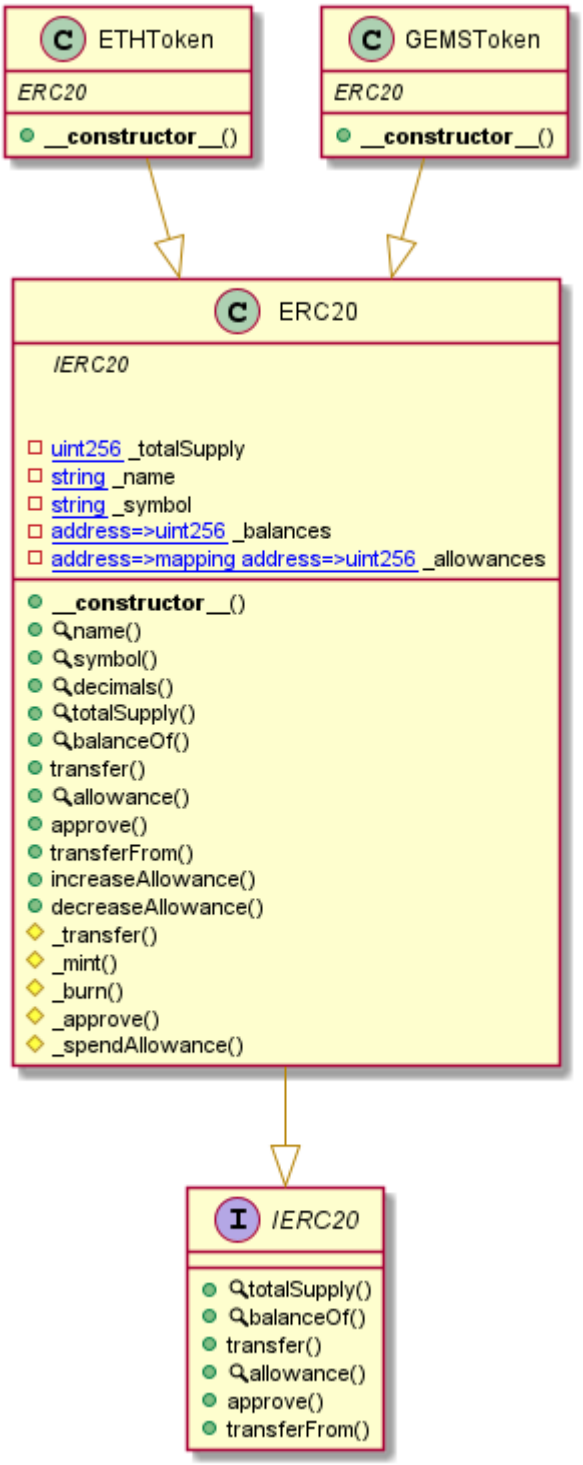
GEMSNFTReceipt Diagram



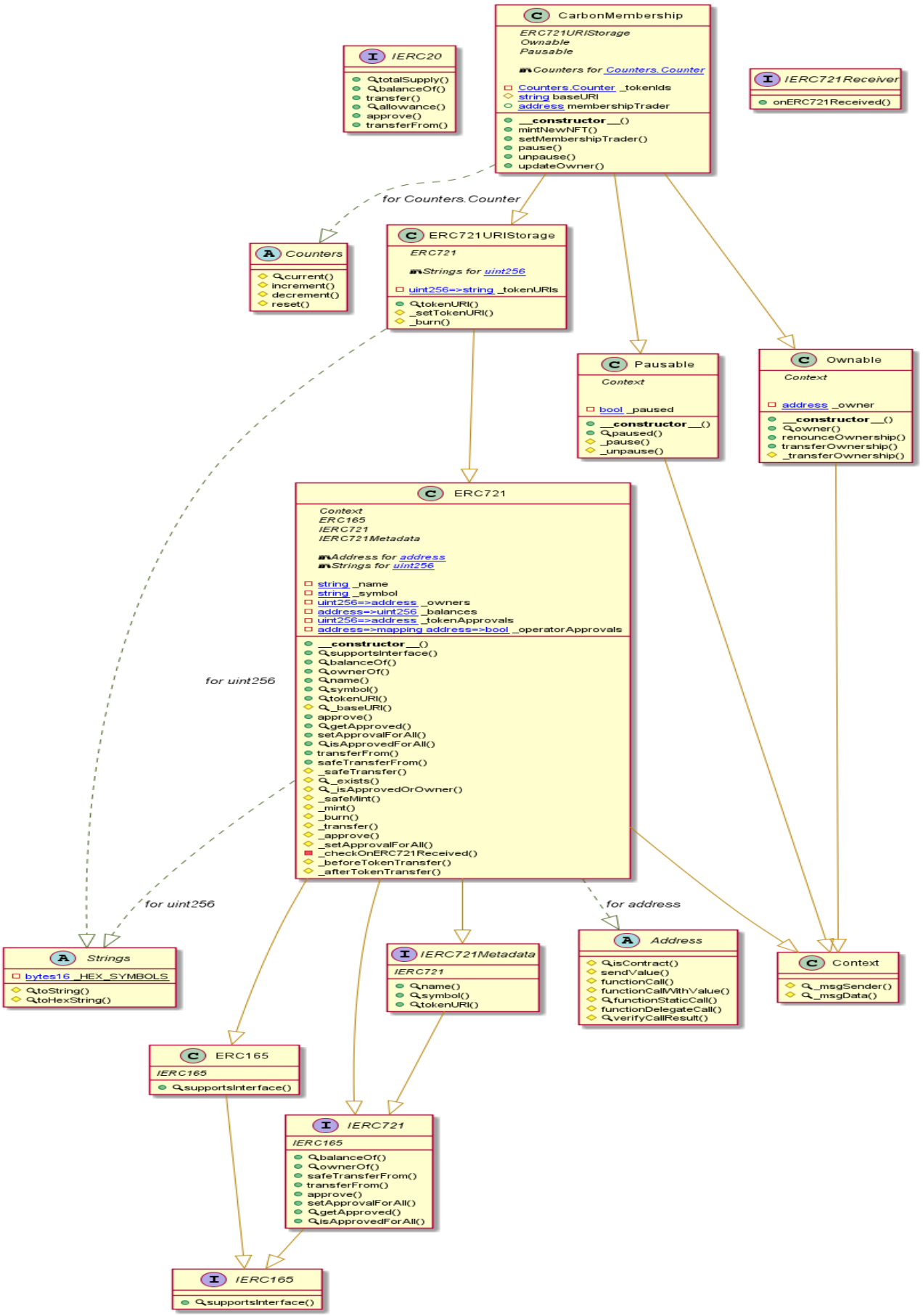
GEMSStaking Diagram



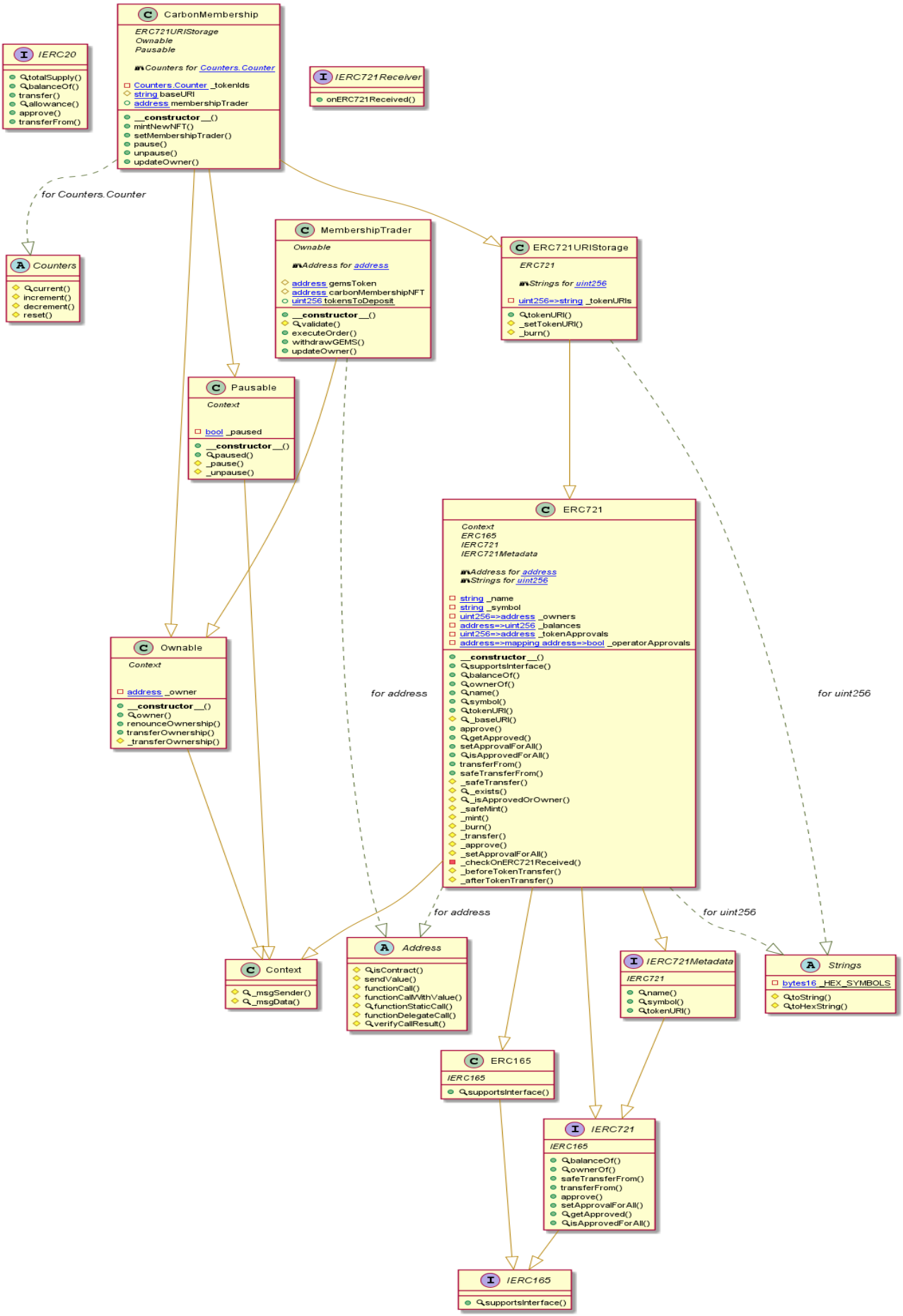
GEMSToken Diagram



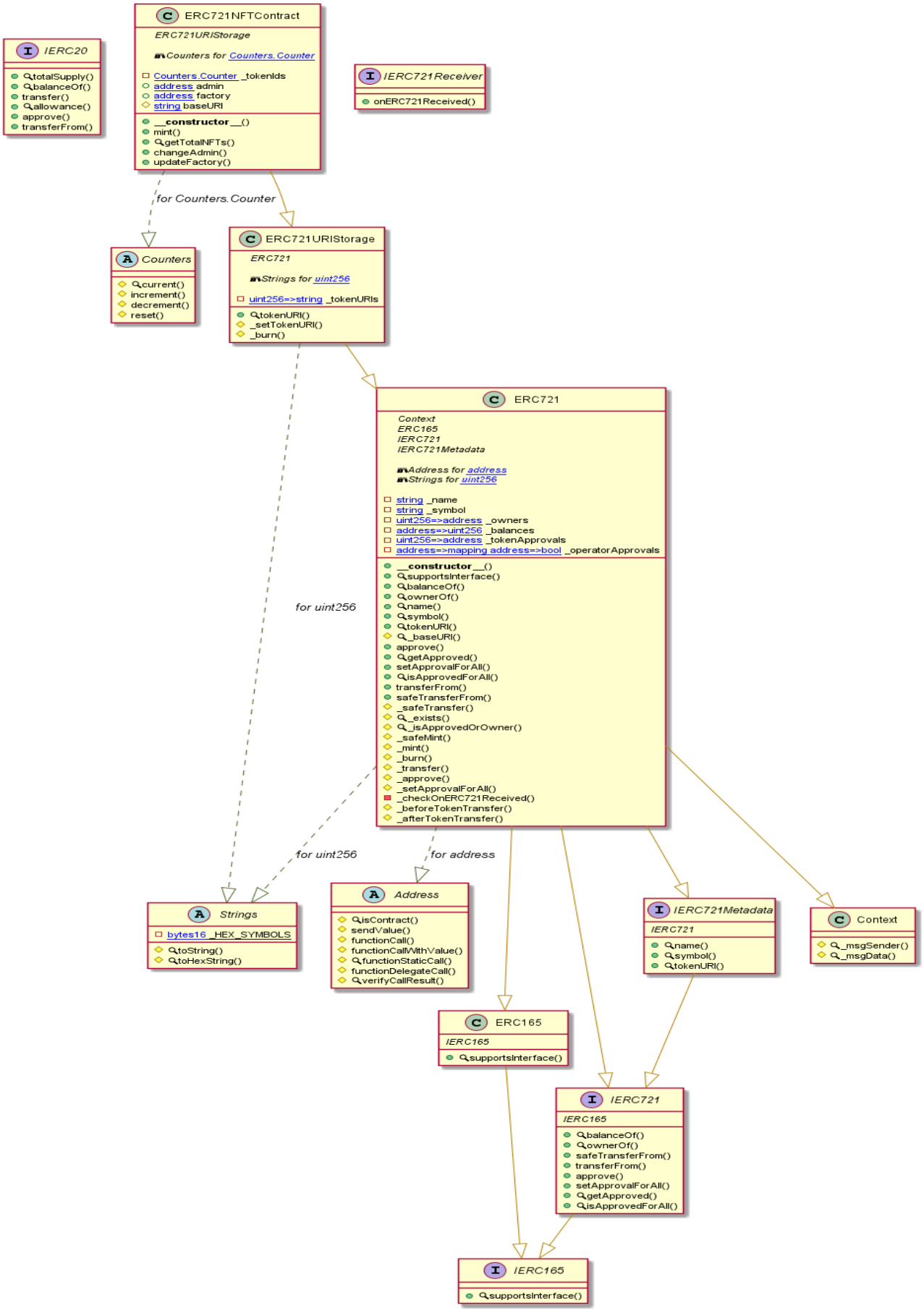
CarbonMembership Diagram



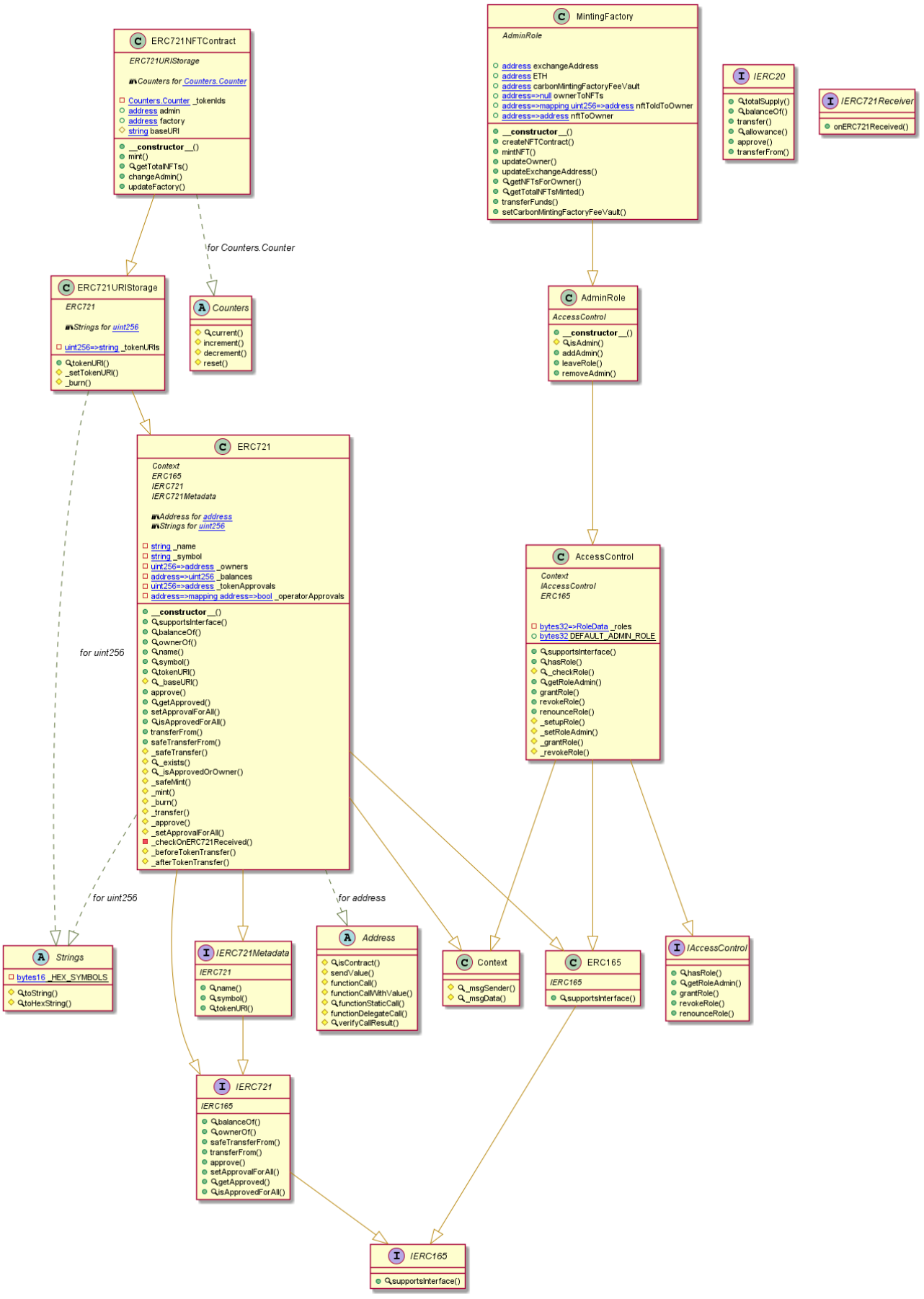
MembershipTrader Diagram



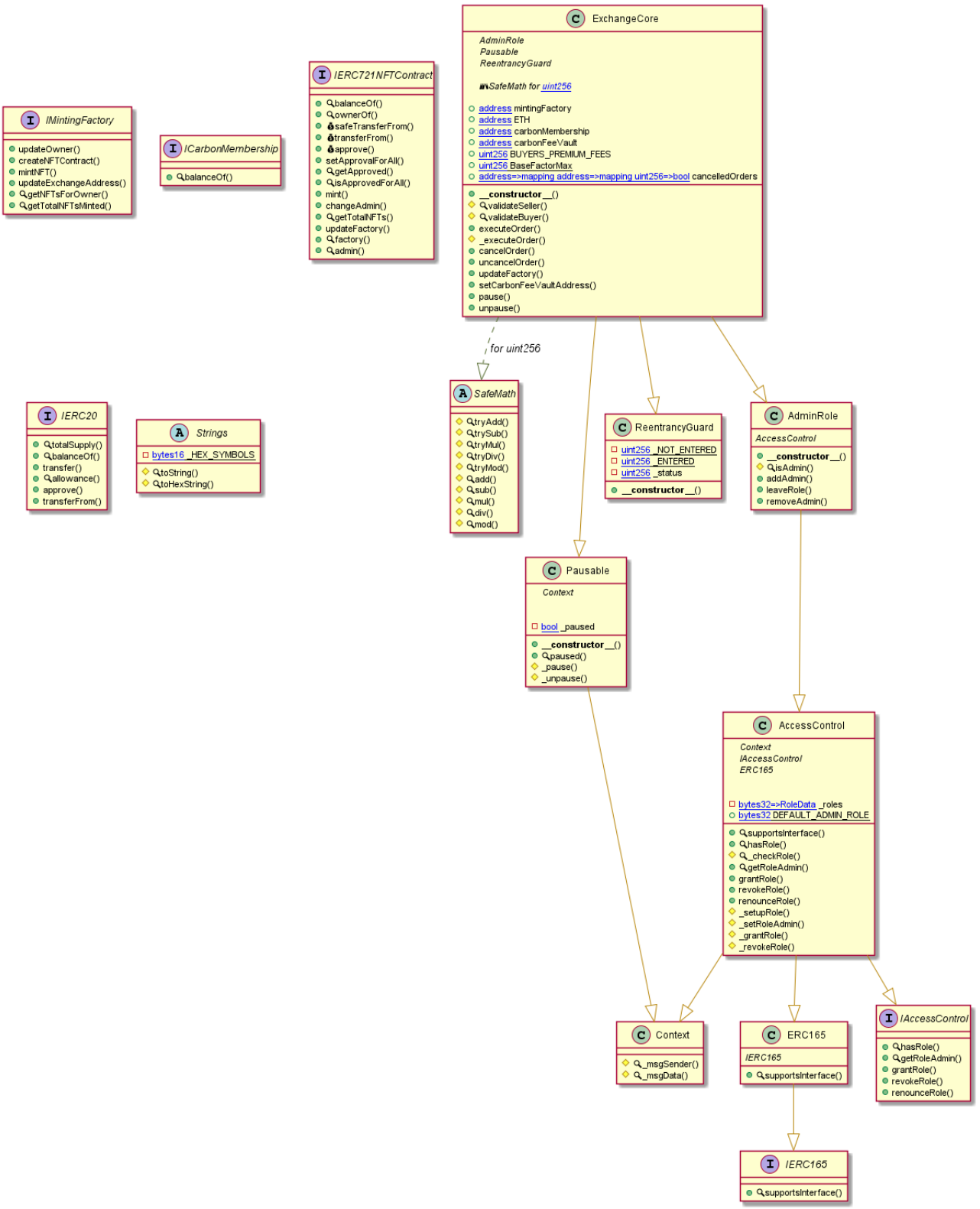
ERC721NFTContract Diagram



MintingFactory Diagram



ExchangeCore Diagram



Slither Results Log

Slither log >> ETHToken.sol

```
INFO:Detectors:
ERC20._burn(address,uint256) (ETHToken.sol#177-188) is never used and should be removed
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#dead-code
INFO:Detectors:
Pragma version^0.8.4 (ETHToken.sol#3) necessitates a version too recent to be trusted. Consider deploying with 0.6.12/0.7.6
solc-0.8.4 is not recommended for deployment
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#incorrect-versions-of-solidity
INFO:Detectors:
ETHToken.constructor() (ETHToken.sol#222-224) uses literals with too many digits:
        - _mint(msg.sender,100000000 * 10 ** 18) (ETHToken.sol#223)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#too-many-digits
INFO:Detectors:
name() should be declared external:
        - ERC20.name() (ETHToken.sol#51-53)
symbol() should be declared external:
        - ERC20.symbol() (ETHToken.sol#55-57)
decimals() should be declared external:
        - ERC20.decimals() (ETHToken.sol#59-61)
totalSupply() should be declared external:
        - ERC20.totalSupply() (ETHToken.sol#63-65)
balanceOf(address) should be declared external:
        - ERC20.balanceOf(address) (ETHToken.sol#67-75)
transfer(address,uint256) should be declared external:
        - ERC20.transfer(address,uint256) (ETHToken.sol#77-86)
approve(address,uint256) should be declared external:
        - ERC20.approve(address,uint256) (ETHToken.sol#98-107)
transferFrom(address,address,uint256) should be declared external:
        - ERC20.transferFrom(address,address,uint256) (ETHToken.sol#109-118)
increaseAllowance(address,uint256) should be declared external:
        - ERC20.increaseAllowance(address,uint256) (ETHToken.sol#120-128)
decreaseAllowance(address,uint256) should be declared external:
        - ERC20.decreaseAllowance(address,uint256) (ETHToken.sol#130-146)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#public-function-that-could-be-declared-external
INFO:Slither:ETHToken.sol analyzed (3 contracts with 75 detectors), 14 result(s) found
INFO:Slither:Use https://crytic.io/ to get access to additional detectors and Github integration
```

Slither log >> AdminRole.sol

```
INFO:Detectors:
AccessControl._setRoleAdmin(bytes32,bytes32) (AdminRole.sol#334-338) is never used and should be removed
Context._msgData() (AdminRole.sol#91-93) is never used and should be removed
Strings.toHexString(uint256) (AdminRole.sol#49-60) is never used and should be removed
Strings.toString(uint256) (AdminRole.sol#24-44) is never used and should be removed
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#dead-code
INFO:Detectors:
Pragma version^0.8.4 (AdminRole.sol#3) necessitates a version too recent to be trusted. Consider deploying with 0.6.12/0.7.6
solc-0.8.4 is not recommended for deployment
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#incorrect-versions-of-solidity
INFO:Slither:AdminRole.sol analyzed (7 contracts with 75 detectors), 6 result(s) found
INFO:Slither:Use https://crytic.io/ to get access to additional detectors and Github integration
```

Slither log >> GEMSNFTReceipt.sol

```
INFO:Detectors:
GEMSNFTReceipt.constructor(string,string,address)._name (GEMSNFTReceipt.sol#1008) shadows:
        - ERC721._name (GEMSNFTReceipt.sol#516) (state variable)
GEMSNFTReceipt.constructor(string,string,address)._symbol (GEMSNFTReceipt.sol#1009) shadows:
        - ERC721._symbol (GEMSNFTReceipt.sol#519) (state variable)
GEMSNFTReceipt.mintNewNFT(address).tokenURI (GEMSNFTReceipt.sol#1026-1028) shadows:
        - ERC721URIStorage.tokenURI(uint256) (GEMSNFTReceipt.sol#950-966) (function)
        - ERC721.tokenURI(uint256) (GEMSNFTReceipt.sol#585-590) (function)
        - IERC721Metadata.tokenURI(uint256) (GEMSNFTReceipt.sol#499) (function)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#local-variable-shadowing
INFO:Detectors:
GEMSNFTReceipt.setStakingPool(address) (GEMSNFTReceipt.sol#1043-1045) should emit an event for:
        - stakingPool = _stakingPool (GEMSNFTReceipt.sol#1044)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#missing-events-access-control
INFO:Detectors:
GEMSNFTReceipt.constructor(string,string,address)._admin (GEMSNFTReceipt.sol#1010) lacks a zero-check on :
                - admin = _admin (GEMSNFTReceipt.sol#1012)
GEMSNFTReceipt.setStakingPool(address)._stakingPool (GEMSNFTReceipt.sol#1043) lacks a zero-check on :
                - stakingPool = _stakingPool (GEMSNFTReceipt.sol#1044)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#missing-zero-address-validation
INFO:Detectors:
Variable 'ERC721._checkOnERC721Received(address,address,uint256,bytes).retval (GEMSNFTReceipt.sol#887)' in ERC721._checkOnER
C721Received(address,address,uint256,bytes) (GEMSNFTReceipt.sol#880-901) potentially used before declaration: retval == IERC
721Receiver.onERC721Received.selector (GEMSNFTReceipt.sol#888)
Variable 'ERC721._checkOnERC721Received(address,address,uint256,bytes).reason (GEMSNFTReceipt.sol#889)' in ERC721._checkOnER
C721Received(address,address,uint256,bytes) (GEMSNFTReceipt.sol#880-901) potentially used before declaration: reason.length
== 0 (GEMSNFTReceipt.sol#890)
Variable 'ERC721._checkOnERC721Received(address,address,uint256,bytes).reason (GEMSNFTReceipt.sol#889)' in ERC721._checkOnER
C721Received(address,address,uint256,bytes) (GEMSNFTReceipt.sol#880-901) potentially used before declaration: revert(uint256
,uint256)(32 + reason,mload(uint256)(reason)) (GEMSNFTReceipt.sol#894)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#pre-declaration-usage-of-local-variables
```

```
INFO:Detectors:
Address.verifyCallResult(bool,bytes,string) (GEMSNFTReceipt.sol#227-247) uses assembly
        - INLINE ASM (GEMSNFTReceipt.sol#239-242)
ERC721._checkOnERC721Received(address,address,uint256,bytes) (GEMSNFTReceipt.sol#880-901) uses assembly
        - INLINE ASM (GEMSNFTReceipt.sol#893-895)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#assembly-usage
INFO:Detectors:
Address.functionCall(address,bytes) (GEMSNFTReceipt.sol#111-113) is never used and should be removed
Address.functionCall(address,bytes,string) (GEMSNFTReceipt.sol#121-127) is never used and should be removed
Address.functionCallWithValue(address,bytes,uint256) (GEMSNFTReceipt.sol#140-146) is never used and should be removed
Address.functionCallWithValue(address,bytes,uint256,string) (GEMSNFTReceipt.sol#154-165) is never used and should be removed
Address.functionDelegateCall(address,bytes) (GEMSNFTReceipt.sol#200-202) is never used and should be removed
Address.functionDelegateCall(address,bytes,string) (GEMSNFTReceipt.sol#210-219) is never used and should be removed
Address.functionStaticCall(address,bytes) (GEMSNFTReceipt.sol#173-175) is never used and should be removed
Address.functionStaticCall(address,bytes,string) (GEMSNFTReceipt.sol#183-192) is never used and should be removed
Address.sendValue(address,uint256) (GEMSNFTReceipt.sol#86-91) is never used and should be removed
Address.verifyCallResult(bool,bytes,string) (GEMSNFTReceipt.sol#227-247) is never used and should be removed
Context._msgData() (GEMSNFTReceipt.sol#506-508) is never used and should be removed
Counters.decrement(Counters.Counter) (GEMSNFTReceipt.sol#23-29) is never used and should be removed
Counters.reset(Counters.Counter) (GEMSNFTReceipt.sol#31-33) is never used and should be removed
ERC721._safeMint(address,uint256) (GEMSNFTReceipt.sol#740-742) is never used and should be removed
ERC721._safeMint(address,uint256,bytes) (GEMSNFTReceipt.sol#748-758) is never used and should be removed
Strings.toHexString(uint256) (GEMSNFTReceipt.sol#281-292) is never used and should be removed
Strings.toHexString(uint256,uint256) (GEMSNFTReceipt.sol#297-307) is never used and should be removed
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#dead-code
INFO:Detectors:
Pragma version^0.8.4 (GEMSNFTReceipt.sol#2) necessitates a version too recent to be trusted. Consider deploying with 0.6.12/
0.7.6
solc-0.8.4 is not recommended for deployment
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#incorrect-versions-of-solidity
INFO:Detectors:
Low level call in Address.sendValue(address,uint256) (GEMSNFTReceipt.sol#86-91):
        - (success) = recipient.call{value: amount}() (GEMSNFTReceipt.sol#89)
Low level call in Address.functionCallWithValue(address,bytes,uint256,string) (GEMSNFTReceipt.sol#154-165):
        - (success,returndata) = target.call{value: value}(data) (GEMSNFTReceipt.sol#163)
Low level call in Address.functionStaticCall(address,bytes,string) (GEMSNFTReceipt.sol#183-192):
        - (success,returndata) = target.staticcall(data) (GEMSNFTReceipt.sol#190)
```

```
        - (success,returndata) = target.staticcall(data) (GEMSNFTReceipt.sol#190)
Low level call in Address.functionDelegateCall(address,bytes,string) (GEMSNFTReceipt.sol#210-219):
        - (success,returndata) = target.delegatecall(data) (GEMSNFTReceipt.sol#217)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#low-level-calls
INFO:Detectors:
Parameter ERC721.safeTransferFrom(address,address,uint256,bytes)._data (GEMSNFTReceipt.sol#671) is not in mixedCase
Parameter GEMSNFTReceipt.setStakingPool(address)._stakingPool (GEMSNFTReceipt.sol#1043) is not in mixedCase
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#conformance-to-solidity-naming-conventions
INFO:Detectors:
GEMSNFTReceipt.baseURI (GEMSNFTReceipt.sol#1003) should be constant
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#state-variables-that-could-be-declared-constant
INFO:Detectors:
balanceOf(address) should be declared external:
        - ERC721.balanceOf(address) (GEMSNFTReceipt.sol#554-557)
name() should be declared external:
        - ERC721.name() (GEMSNFTReceipt.sol#571-573)
symbol() should be declared external:
        - ERC721.symbol() (GEMSNFTReceipt.sol#578-580)
approve(address,uint256) should be declared external:
        - ERC721.approve(address,uint256) (GEMSNFTReceipt.sol#604-614)
setApprovalForAll(address,bool) should be declared external:
        - ERC721.setApprovalForAll(address,bool) (GEMSNFTReceipt.sol#628-630)
transferFrom(address,address,uint256) should be declared external:
        - ERC721.transferFrom(address,address,uint256) (GEMSNFTReceipt.sol#642-651)
safeTransferFrom(address,address,uint256) should be declared external:
        - ERC721.safeTransferFrom(address,address,uint256) (GEMSNFTReceipt.sol#656-662)
mintNewNFT(address) should be declared external:
        - GEMSNFTReceipt.mintNewNFT(address) (GEMSNFTReceipt.sol#1023-1033)
getTotalNFTs() should be declared external:
        - GEMSNFTReceipt.getTotalNFTs() (GEMSNFTReceipt.sol#1035-1037)
burnNFT(uint256) should be declared external:
        - GEMSNFTReceipt.burnNFT(uint256) (GEMSNFTReceipt.sol#1039-1041)
setStakingPool(address) should be declared external:
        - GEMSNFTReceipt.setStakingPool(address) (GEMSNFTReceipt.sol#1043-1045)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#public-function-that-could-be-declared-external
INFO:Slither:GEMSNFTReceipt.sol analyzed (12 contracts with 75 detectors), 49 result(s) found
INFO:Slither:Use https://crytic.io/ to get access to additional detectors and Github integration
```

Slither log >> GEMSStaking.sol

```
INFO:Detectors:
GEMSNFTReceipt.constructor(string,string,address)._name (GEMSStaking.sol#1037) shadows:
        - ERC721._name (GEMSStaking.sol#545) (state variable)
GEMSNFTReceipt.constructor(string,string,address)._symbol (GEMSStaking.sol#1038) shadows:
        - ERC721._symbol (GEMSStaking.sol#548) (state variable)
GEMSNFTReceipt.mintNewNFT(address).tokenURI (GEMSStaking.sol#1055-1057) shadows:
        - ERC721URIStorage.tokenURI(uint256) (GEMSStaking.sol#979-995) (function)
        - ERC721.tokenURI(uint256) (GEMSStaking.sol#614-619) (function)
        - IERC721Metadata.tokenURI(uint256) (GEMSStaking.sol#528) (function)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#local-variable-shadowing
INFO:Detectors:
GEMSNFTReceipt.setStakingPool(address) (GEMSStaking.sol#1072-1074) should emit an event for:
        - stakingPool = _stakingPool (GEMSStaking.sol#1073)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#missing-events-access-control
INFO:Detectors:
GEMSNFTReceipt.constructor(string,string,address)._admin (GEMSStaking.sol#1039) lacks a zero-check on :
                - admin = _admin (GEMSStaking.sol#1041)
GEMSNFTReceipt.setStakingPool(address)._stakingPool (GEMSStaking.sol#1072) lacks a zero-check on :
                - stakingPool = _stakingPool (GEMSStaking.sol#1073)
GEMSStaking.constructor(address,address)._gemsToken (GEMSStaking.sol#1082) lacks a zero-check on :
                - GEMSToken = _gemsToken (GEMSStaking.sol#1083)
GEMSStaking.constructor(address,address)._gemsNFTAddress (GEMSStaking.sol#1082) lacks a zero-check on :
                - GEMSNFTAddress = _gemsNFTAddress (GEMSStaking.sol#1084)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#missing-zero-address-validation
```

```
INFO:Detectors:
Variable 'ERC721._checkOnERC721Received(address,address,uint256,bytes).retval (GEMSStaking.sol#916)' in ERC721._checkOnERC72
1Received(address,address,uint256,bytes) (GEMSStaking.sol#909-930) potentially used before declaration: retval == IERC721Rec
eiver.onERC721Received.selector (GEMSStaking.sol#917)
Variable 'ERC721._checkOnERC721Received(address,address,uint256,bytes).reason (GEMSStaking.sol#918)' in ERC721._checkOnERC72
1Received(address,address,uint256,bytes) (GEMSStaking.sol#909-930) potentially used before declaration: reason.length == 0 (
GEMSStaking.sol#919)
Variable 'ERC721._checkOnERC721Received(address,address,uint256,bytes).reason (GEMSStaking.sol#918)' in ERC721._checkOnERC72
1Received(address,address,uint256,bytes) (GEMSStaking.sol#909-930) potentially used before declaration: revert(uint256,uint2
56)(32 + reason,mload(uint256)(reason)) (GEMSStaking.sol#923)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#pre-declaration-usage-of-local-variables
INFO:Detectors:
Reentrancy in GEMSStaking.stake(address,uint256) (GEMSStaking.sol#1098-1118):
        External calls:
        - IERC20(GEMSToken).transferFrom(user,address(this),_amount) (GEMSStaking.sol#1110)
        - tokenId = GEMSNFTReceipt(GEMSNFTAddress).mintNewNFT(user) (GEMSStaking.sol#1113)
        Event emitted after the call(s):
        - Staked(user,_amount) (GEMSStaking.sol#1117)
Reentrancy in GEMSStaking.unstake() (GEMSStaking.sol#1120-1129):
        External calls:
        - IERC20(GEMSToken).transfer(msg.sender,amount) (GEMSStaking.sol#1126)
        - GEMSNFTReceipt(GEMSNFTAddress).burnNFT(tokenId) (GEMSStaking.sol#1127)
        Event emitted after the call(s):
        - UnStaked(msg.sender,amount) (GEMSStaking.sol#1128)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#reentrancy-vulnerabilities-3
INFO:Detectors:
GEMSStaking.stake(address,uint256) (GEMSStaking.sol#1098-1118) uses timestamp for comparisons
        Dangerous comparisons:
        - require(bool,string)(amountStaked == 0,User had already staked tokens) (GEMSStaking.sol#1100)
GEMSStaking.unstake() (GEMSStaking.sol#1120-1129) uses timestamp for comparisons
        Dangerous comparisons:
        - require(bool,string)(amount != 0,User had no amount staked!) (GEMSStaking.sol#1123)
        - require(bool)(msg.sender == GEMSNFTReceipt(GEMSNFTAddress).ownerOf(tokenId)) (GEMSStaking.sol#1124)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#block-timestamp
```

```
INFO:Detectors:
Address.verifyCallResult(bool,bytes,string) (GEMSStaking.sol#258-278) uses assembly
        - INLINE ASM (GEMSStaking.sol#270-273)
ERC721._checkOnERC721Received(address,address,uint256,bytes) (GEMSStaking.sol#909-930) uses assembly
        - INLINE ASM (GEMSStaking.sol#922-924)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#assembly-usage
INFO:Detectors:
Address.functionCall(address,bytes) (GEMSStaking.sol#142-144) is never used and should be removed
Address.functionCall(address,bytes,string) (GEMSStaking.sol#152-158) is never used and should be removed
Address.functionCallWithValue(address,bytes,uint256) (GEMSStaking.sol#171-177) is never used and should be removed
Address.functionCallWithValue(address,bytes,uint256,string) (GEMSStaking.sol#185-196) is never used and should be removed
Address.functionDelegateCall(address,bytes) (GEMSStaking.sol#231-233) is never used and should be removed
Address.functionDelegateCall(address,bytes,string) (GEMSStaking.sol#241-250) is never used and should be removed
Address.functionStaticCall(address,bytes) (GEMSStaking.sol#204-206) is never used and should be removed
Address.functionStaticCall(address,bytes,string) (GEMSStaking.sol#214-223) is never used and should be removed
Address.sendValue(address,uint256) (GEMSStaking.sol#117-122) is never used and should be removed
Address.verifyCallResult(bool,bytes,string) (GEMSStaking.sol#258-278) is never used and should be removed
Context._msgData() (GEMSStaking.sol#535-537) is never used and should be removed
Counters.decrement(Counters.Counter) (GEMSStaking.sol#54-60) is never used and should be removed
Counters.reset(Counters.Counter) (GEMSStaking.sol#62-64) is never used and should be removed
ERC721._safeMint(address,uint256) (GEMSStaking.sol#769-771) is never used and should be removed
ERC721._safeMint(address,uint256,bytes) (GEMSStaking.sol#777-787) is never used and should be removed
Strings.toHexString(uint256) (GEMSStaking.sol#312-323) is never used and should be removed
Strings.toHexString(uint256,uint256) (GEMSStaking.sol#328-338) is never used and should be removed
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#dead-code
INFO:Detectors:
Pragma version^0.8.4 (GEMSStaking.sol#3) necessitates a version too recent to be trusted. Consider deploying with 0.6.12/0.7
.6
solc-0.8.4 is not recommended for deployment
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#incorrect-versions-of-solidity
INFO:Detectors:
Low level call in Address.sendValue(address,uint256) (GEMSStaking.sol#117-122):
        - (success) = recipient.call{value: amount}() (GEMSStaking.sol#120)
Low level call in Address.functionCallWithValue(address,bytes,uint256,string) (GEMSStaking.sol#185-196):
        - (success,returndata) = target.call{value: value}(data) (GEMSStaking.sol#194)
Low level call in Address.functionStaticCall(address,bytes,string) (GEMSStaking.sol#214-223):
        - (success,returndata) = target.staticcall(data) (GEMSStaking.sol#221)
```

```
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#low-level-calls
INFO:Detectors:
Parameter ERC721.safeTransferFrom(address,address,uint256,bytes)._data (GEMSStaking.sol#700) is not in mixedCase
Parameter GEMSNFTReceipt.setStakingPool(address)._stakingPool (GEMSStaking.sol#1072) is not in mixedCase
Parameter GEMSStaking.stake(address,uint256)._amount (GEMSStaking.sol#1098) is not in mixedCase
Variable GEMSStaking.GEMSToken (GEMSStaking.sol#1078) is not in mixedCase
Variable GEMSStaking.GEMSNFTAddress (GEMSStaking.sol#1079) is not in mixedCase
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#conformance-to-solidity-naming-conventions
INFO:Detectors:
GEMSStaking.slitherConstructorVariables() (GEMSStaking.sol#1077-1133) uses literals with too many digits:
        - tokensToStake = 100000 * 10 ** 18 (GEMSStaking.sol#1080)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#too-many-digits
INFO:Detectors:
GEMSNFTReceipt.baseURI (GEMSStaking.sol#1032) should be constant
GEMSStaking.tokensToStake (GEMSStaking.sol#1080) should be constant
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#state-variables-that-could-be-declared-constant
INFO:Detectors:
balanceOf(address) should be declared external:
        - ERC721.balanceOf(address) (GEMSStaking.sol#583-586)
name() should be declared external:
        - ERC721.name() (GEMSStaking.sol#600-602)
symbol() should be declared external:
        - ERC721.symbol() (GEMSStaking.sol#607-609)
approve(address,uint256) should be declared external:
        - ERC721.approve(address,uint256) (GEMSStaking.sol#633-643)
setApprovalForAll(address,bool) should be declared external:
        - ERC721.setApprovalForAll(address,bool) (GEMSStaking.sol#657-659)
transferFrom(address,address,uint256) should be declared external:
        - ERC721.transferFrom(address,address,uint256) (GEMSStaking.sol#671-680)
safeTransferFrom(address,address,uint256) should be declared external:
        - ERC721.safeTransferFrom(address,address,uint256) (GEMSStaking.sol#685-691)
mintNewNFT(address) should be declared external:
        - GEMSNFTReceipt.mintNewNFT(address) (GEMSStaking.sol#1052-1062)
getTotalNFTs() should be declared external:
        - GEMSNFTReceipt.getTotalNFTs() (GEMSStaking.sol#1064-1066)
burnNFT(uint256) should be declared external:
        - GEMSNFTReceipt.burnNFT(uint256) (GEMSStaking.sol#1068-1070)
```

```
burnNFT(uint256) should be declared external:
        - GEMSNFTReceipt.burnNFT(uint256) (GEMSStaking.sol#1068-1070)
setStakingPool(address) should be declared external:
        - GEMSNFTReceipt.setStakingPool(address) (GEMSStaking.sol#1072-1074)
stake(address,uint256) should be declared external:
        - GEMSStaking.stake(address,uint256) (GEMSStaking.sol#1098-1118)
unstake() should be declared external:
        - GEMSStaking.unstake() (GEMSStaking.sol#1120-1129)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#public-function-that-could-be-declared-external
INFO:Slither:GEMSStaking.sol analyzed (14 contracts with 75 detectors), 67 result(s) found
INFO:Slither:Use https://crytic.io/ to get access to additional detectors and Github integration
```

Slither log >> GEMSToken.sol

```
INFO:Detectors:
ERC20._burn(address,uint256) (GEMSToken.sol#176-187) is never used and should be removed
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#dead-code
INFO:Detectors:
Pragma version^0.8.4 (GEMSToken.sol#3) necessitates a version too recent to be trusted. Consider deploying with 0.6.12/0.7.6
solc-0.8.4 is not recommended for deployment
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#incorrect-versions-of-solidity
INFO:Detectors:
ETHToken.constructor() (GEMSToken.sol#220-222) uses literals with too many digits:
        - _mint(msg.sender,100000000 * 10 ** 18) (GEMSToken.sol#221)
GEMSToken.constructor() (GEMSToken.sol#226-228) uses literals with too many digits:
        - _mint(msg.sender,1000000000 * 10 ** 18) (GEMSToken.sol#227)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#too-many-digits
INFO:Detectors:
name() should be declared external:
        - ERC20.name() (GEMSToken.sol#50-52)
symbol() should be declared external:
        - ERC20.symbol() (GEMSToken.sol#54-56)
decimals() should be declared external:
        - ERC20.decimals() (GEMSToken.sol#58-60)
totalSupply() should be declared external:
        - ERC20.totalSupply() (GEMSToken.sol#62-64)
balanceOf(address) should be declared external:
        - ERC20.balanceOf(address) (GEMSToken.sol#66-74)
transfer(address,uint256) should be declared external:
        - ERC20.transfer(address,uint256) (GEMSToken.sol#76-85)
approve(address,uint256) should be declared external:
        - ERC20.approve(address,uint256) (GEMSToken.sol#97-106)
transferFrom(address,address,uint256) should be declared external:
        - ERC20.transferFrom(address,address,uint256) (GEMSToken.sol#108-117)
increaseAllowance(address,uint256) should be declared external:
        - ERC20.increaseAllowance(address,uint256) (GEMSToken.sol#119-127)
decreaseAllowance(address,uint256) should be declared external:
        - ERC20.decreaseAllowance(address,uint256) (GEMSToken.sol#129-145)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#public-function-that-could-be-declared-external
INFO:Slither:GEMSToken.sol analyzed (4 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 >> CarbonMembership.sol

```
INFO:Detectors:
CarbonMembership.constructor(string,string)._name (CarbonMembership.sol#1166) shadows:
        - ERC721._name (CarbonMembership.sol#602) (state variable)
CarbonMembership.constructor(string,string)._symbol (CarbonMembership.sol#1166) shadows:
        - ERC721._symbol (CarbonMembership.sol#605) (state variable)
CarbonMembership.mintNewNFT(address).tokenURI (CarbonMembership.sol#1186-1188) shadows:
        - ERC721URIStorage.tokenURI(uint256) (CarbonMembership.sol#1111-1127) (function)
        - ERC721.tokenURI(uint256) (CarbonMembership.sol#671-676) (function)
        - IERC721Metadata.tokenURI(uint256) (CarbonMembership.sol#528) (function)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#local-variable-shadowing
INFO:Detectors:
CarbonMembership.setMembershipTrader(address)._newMembershipTrader (CarbonMembership.sol#1195) lacks a zero-check on :
                - membershipTrader = _newMembershipTrader (CarbonMembership.sol#1199)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#missing-zero-address-validation
INFO:Detectors:
Variable 'ERC721._checkOnERC721Received(address,address,uint256,bytes).retval (CarbonMembership.sol#973)' in ERC721._checkOn
ERC721Received(address,address,uint256,bytes) (CarbonMembership.sol#966-987) potentially used before declaration: retval ==
IERC721Receiver.onERC721Received.selector (CarbonMembership.sol#974)
Variable 'ERC721._checkOnERC721Received(address,address,uint256,bytes).reason (CarbonMembership.sol#975)' in ERC721._checkOn
ERC721Received(address,address,uint256,bytes) (CarbonMembership.sol#966-987) potentially used before declaration: reason.len
gth == 0 (CarbonMembership.sol#976)
Variable 'ERC721._checkOnERC721Received(address,address,uint256,bytes).reason (CarbonMembership.sol#975)' in ERC721._checkOn
ERC721Received(address,address,uint256,bytes) (CarbonMembership.sol#966-987) potentially used before declaration: revert(uin
t256,uint256)(32 + reason,mload(uint256)(reason)) (CarbonMembership.sol#980)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#pre-declaration-usage-of-local-variables
INFO:Detectors:
Address.verifyCallResult(bool,bytes,string) (CarbonMembership.sol#258-278) uses assembly
        - INLINE ASM (CarbonMembership.sol#270-273)
ERC721._checkOnERC721Received(address,address,uint256,bytes) (CarbonMembership.sol#966-987) uses assembly
        - INLINE ASM (CarbonMembership.sol#979-981)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#assembly-usage
INFO:Detectors:
Address.functionCall(address,bytes) (CarbonMembership.sol#142-144) is never used and should be removed
Address.functionCall(address,bytes,string) (CarbonMembership.sol#152-158) is never used and should be removed
Address.functionCallWithValue(address,bytes,uint256) (CarbonMembership.sol#171-177) is never used and should be removed
Address.functionCallWithValue(address,bytes,uint256,string) (CarbonMembership.sol#185-196) is never used and should be remov
```

```
Address.functionDelegateCall(address,bytes) (CarbonMembership.sol#231-233) is never used and should be removed
Address.functionDelegateCall(address,bytes,string) (CarbonMembership.sol#241-250) is never used and should be removed
Address.functionStaticCall(address,bytes) (CarbonMembership.sol#204-206) is never used and should be removed
Address.functionStaticCall(address,bytes,string) (CarbonMembership.sol#214-223) is never used and should be removed
Address.sendValue(address,uint256) (CarbonMembership.sol#117-122) is never used and should be removed
Address.verifyCallResult(bool,bytes,string) (CarbonMembership.sol#258-278) is never used and should be removed
Context._msgData() (CarbonMembership.sol#535-537) is never used and should be removed
```

```
Counters.decrement(Counters.Counter) (CarbonMembership.sol#54-60) is never used and should be removed
Counters.reset(Counters.Counter) (CarbonMembership.sol#62-64) is never used and should be removed
ERC721._burn(uint256) (CarbonMembership.sol#882-896) is never used and should be removed
ERC721._safeMint(address,uint256) (CarbonMembership.sol#826-828) is never used and should be removed
ERC721._safeMint(address,uint256,bytes) (CarbonMembership.sol#834-844) is never used and should be removed
ERC721URIStorage._burn(uint256) (CarbonMembership.sol#1151-1157) is never used and should be removed
Strings.toHexString(uint256) (CarbonMembership.sol#312-323) is never used and should be removed
Strings.toHexString(uint256,uint256) (CarbonMembership.sol#328-338) is never used and should be removed
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#dead-code
INFO:Detectors:
Pragma version^0.8.4 (CarbonMembership.sol#2) necessitates a version too recent to be trusted. Consider deploying with 0.6.12/0.7.6
solc-0.8.4 is not recommended for deployment
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#incorrect-versions-of-solidity
INFO:Detectors:
Low level call in Address.sendValue(address,uint256) (CarbonMembership.sol#117-122):
        - (success) = recipient.call{value: amount}() (CarbonMembership.sol#120)
Low level call in Address.functionCallWithValue(address,bytes,uint256,string) (CarbonMembership.sol#185-196):
        - (success,returndata) = target.call{value: value}(data) (CarbonMembership.sol#194)
Low level call in Address.functionStaticCall(address,bytes,string) (CarbonMembership.sol#214-223):
        - (success,returndata) = target.staticcall(data) (CarbonMembership.sol#221)
Low level call in Address.functionDelegateCall(address,bytes,string) (CarbonMembership.sol#241-250):
        - (success,returndata) = target.delegatecall(data) (CarbonMembership.sol#248)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#low-level-calls
INFO:Detectors:
Parameter ERC721.safeTransferFrom(address,address,uint256,bytes)._data (CarbonMembership.sol#757) is not in mixedCase
Parameter CarbonMembership.setMembershipTrader(address)._newMembershipTrader (CarbonMembership.sol#1195) is not in mixedCase
Parameter CarbonMembership.updateOwner(address)._newOwner (CarbonMembership.sol#1210) is not in mixedCase
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#conformance-to-solidity-naming-conventions
INFO:Detectors:
CarbonMembership.baseURI (CarbonMembership.sol#1163) should be constant
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#state-variables-that-could-be-declared-constant
INFO:Detectors:
renounceOwnership() should be declared external:
        - Ownable.renounceOwnership() (CarbonMembership.sol#574-576)
transferOwnership(address) should be declared external:
        - Ownable.transferOwnership(address) (CarbonMembership.sol#582-585)
balanceOf(address) should be declared external:
        - ERC721.balanceOf(address) (CarbonMembership.sol#640-643)
name() should be declared external:
        - ERC721.name() (CarbonMembership.sol#657-659)
symbol() should be declared external:
        - ERC721.symbol() (CarbonMembership.sol#664-666)
approve(address,uint256) should be declared external:
        - ERC721.approve(address,uint256) (CarbonMembership.sol#690-700)
setApprovalForAll(address,bool) should be declared external:
        - ERC721.setApprovalForAll(address,bool) (CarbonMembership.sol#714-716)
transferFrom(address,address,uint256) should be declared external:
        - ERC721.transferFrom(address,address,uint256) (CarbonMembership.sol#728-737)
safeTransferFrom(address,address,uint256) should be declared external:
        - ERC721.safeTransferFrom(address,address,uint256) (CarbonMembership.sol#742-748)
mintNewNFT(address) should be declared external:
        - CarbonMembership.mintNewNFT(address) (CarbonMembership.sol#1178-1193)
setMembershipTrader(address) should be declared external:
        - CarbonMembership.setMembershipTrader(address) (CarbonMembership.sol#1195-1200)
pause() should be declared external:
        - CarbonMembership.pause() (CarbonMembership.sol#1202-1204)
unpause() should be declared external:
        - CarbonMembership.unpause() (CarbonMembership.sol#1206-1208)
updateOwner(address) should be declared external:
        - CarbonMembership.updateOwner(address) (CarbonMembership.sol#1210-1212)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#public-function-that-could-be-declared-external
INFO:Slither:CarbonMembership.sol analyzed (15 contracts with 75 detectors), 53 result(s) found
INFO:Slither:Use https://crytic.io/ to get access to additional detectors and Github integration
```

Slither log >> MembershipTrader.sol

```
INFO:Detectors:
CarbonMembership.constructor(string,string)._name (MembershipTrader.sol#1166) shadows:
        - ERC721._name (MembershipTrader.sol#601) (state variable)
CarbonMembership.constructor(string,string)._symbol (MembershipTrader.sol#1166) shadows:
        - ERC721._symbol (MembershipTrader.sol#604) (state variable)
CarbonMembership.mintNewNFT(address).tokenURI (MembershipTrader.sol#1186-1188) shadows:
        - ERC721URIStorage.tokenURI(uint256) (MembershipTrader.sol#1110-1126) (function)
        - ERC721.tokenURI(uint256) (MembershipTrader.sol#670-675) (function)
        - IERC721Metadata.tokenURI(uint256) (MembershipTrader.sol#527) (function)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#local-variable-shadowing
INFO:Detectors:
CarbonMembership.setMembershipTrader(address)._newMembershipTrader (MembershipTrader.sol#1195) lacks a zero-check on :
                - membershipTrader = _newMembershipTrader (MembershipTrader.sol#1199)
MembershipTrader.constructor(address,address)._gemsToken (MembershipTrader.sol#1222) lacks a zero-check on :
                - gemsToken = _gemsToken (MembershipTrader.sol#1223)
MembershipTrader.constructor(address,address)._carbonMembershipNFT (MembershipTrader.sol#1222) lacks a zero-check on :
                - carbonMembershipNFT = _carbonMembershipNFT (MembershipTrader.sol#1224)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#missing-zero-address-validation
INFO:Detectors:
Variable 'ERC721._checkOnERC721Received(address,address,uint256,bytes).retval (MembershipTrader.sol#972)' in ERC721._checkOnERC721Received(address,address,uint256,bytes) (MembershipTrader.sol#965-986) potentially used before declaration: retval == IERC721Receiver.onERC721Received.selector (MembershipTrader.sol#973)
Variable 'ERC721._checkOnERC721Received(address,address,uint256,bytes).reason (MembershipTrader.sol#974)' in ERC721._checkOnERC721Received(address,address,uint256,bytes) (MembershipTrader.sol#965-986) potentially used before declaration: reason.length == 0 (MembershipTrader.sol#975)
Variable 'ERC721._checkOnERC721Received(address,address,uint256,bytes).reason (MembershipTrader.sol#974)' in ERC721._checkOnERC721Received(address,address,uint256,bytes) (MembershipTrader.sol#965-986) potentially used before declaration: revert(uint256,uint256)(32 + reason,mload(uint256)(reason)) (MembershipTrader.sol#979)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#pre-declaration-usage-of-local-variables
```

```
INFO:Detectors:
Address.verifyCallResult(bool,bytes,string) (MembershipTrader.sol#257-277) uses assembly
        - INLINE ASM (MembershipTrader.sol#269-272)
ERC721._checkOnERC721Received(address,address,uint256,bytes) (MembershipTrader.sol#965-986) uses assembly
        - INLINE ASM (MembershipTrader.sol#978-980)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#assembly-usage
INFO:Detectors:
Address.functionCall(address,bytes) (MembershipTrader.sol#141-143) is never used and should be removed
Address.functionCall(address,bytes,string) (MembershipTrader.sol#151-157) is never used and should be removed
Address.functionCallWithValue(address,bytes,uint256) (MembershipTrader.sol#170-176) is never used and should be removed
```

```
Counters.decrement(Counters.Counter) (MembershipTrader.sol#53-59) is never used and should be removed
Counters.reset(Counters.Counter) (MembershipTrader.sol#61-63) is never used and should be removed
ERC721._burn(uint256) (MembershipTrader.sol#881-895) is never used and should be removed
ERC721._safeMint(address,uint256) (MembershipTrader.sol#825-827) is never used and should be removed
ERC721._safeMint(address,uint256,bytes) (MembershipTrader.sol#833-843) is never used and should be removed
ERC721URIStorage._burn(uint256) (MembershipTrader.sol#1150-1156) is never used and should be removed
Strings.toHexString(uint256) (MembershipTrader.sol#311-322) is never used and should be removed
Strings.toHexString(uint256,uint256) (MembershipTrader.sol#327-337) is never used and should be removed
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#dead-code
INFO:Detectors:
Pragma version^0.8.4 (MembershipTrader.sol#2) necessitates a version too recent to be trusted. Consider deploying with 0.6.1
2/0.7.6
solc-0.8.4 is not recommended for deployment
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#incorrect-versions-of-solidity
INFO:Detectors:
Low level call in Address.sendValue(address,uint256) (MembershipTrader.sol#116-121):
        - (success) = recipient.call{value: amount}() (MembershipTrader.sol#119)
Low level call in Address.functionCallWithValue(address,bytes,uint256,string) (MembershipTrader.sol#184-195):
        - (success,returndata) = target.call{value: value}(data) (MembershipTrader.sol#193)
Low level call in Address.functionStaticCall(address,bytes,string) (MembershipTrader.sol#213-222):
        - (success,returndata) = target.staticcall(data) (MembershipTrader.sol#220)
Low level call in Address.functionDelegateCall(address,bytes,string) (MembershipTrader.sol#240-249):
        - (success,returndata) = target.delegatecall(data) (MembershipTrader.sol#247)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#low-level-calls
INFO:Detectors:
Parameter ERC721.safeTransferFrom(address,address,uint256,bytes)._data (MembershipTrader.sol#756) is not in mixedCase
Parameter CarbonMembership.setMembershipTrader(address)._newMembershipTrader (MembershipTrader.sol#1195) is not in mixedCase
Parameter CarbonMembership.updateOwner(address)._newOwner (MembershipTrader.sol#1210) is not in mixedCase
Parameter MembershipTrader.updateOwner(address)._newOwner (MembershipTrader.sol#1256) is not in mixedCase
Constant MembershipTrader.tokensToDeposit (MembershipTrader.sol#1220) is not in UPPER_CASE_WITH_UNDERSCORES
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#conformance-to-solidity-naming-conventions
```

```
INFO:Detectors:
MembershipTrader.slitherConstructorConstantVariables() (MembershipTrader.sol#1215-1259) uses literals with too many digits:
        - tokensToDeposit = 100000 (MembershipTrader.sol#1220)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#too-many-digits
INFO:Detectors:
CarbonMembership.baseURI (MembershipTrader.sol#1163) should be constant
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#state-variables-that-could-be-declared-constant
INFO:Detectors:
renounceOwnership() should be declared external:
        - Ownable.renounceOwnership() (MembershipTrader.sol#573-575)
transferOwnership(address) should be declared external:
        - Ownable.transferOwnership(address) (MembershipTrader.sol#581-584)
balanceOf(address) should be declared external:
        - ERC721.balanceOf(address) (MembershipTrader.sol#639-642)
name() should be declared external:
        - ERC721.name() (MembershipTrader.sol#656-658)
symbol() should be declared external:
        - ERC721.symbol() (MembershipTrader.sol#663-665)
approve(address,uint256) should be declared external:
        - ERC721.approve(address,uint256) (MembershipTrader.sol#689-699)
setApprovalForAll(address,bool) should be declared external:
        - ERC721.setApprovalForAll(address,bool) (MembershipTrader.sol#713-715)
transferFrom(address,address,uint256) should be declared external:
        - ERC721.transferFrom(address,address,uint256) (MembershipTrader.sol#727-736)
safeTransferFrom(address,address,uint256) should be declared external:
        - ERC721.safeTransferFrom(address,address,uint256) (MembershipTrader.sol#741-747)
mintNewNFT(address) should be declared external:
        - CarbonMembership.mintNewNFT(address) (MembershipTrader.sol#1178-1193)
setMembershipTrader(address) should be declared external:
        - CarbonMembership.setMembershipTrader(address) (MembershipTrader.sol#1195-1200)
pause() should be declared external:
        - CarbonMembership.pause() (MembershipTrader.sol#1202-1204)
unpause() should be declared external:
        - CarbonMembership.unpause() (MembershipTrader.sol#1206-1208)
updateOwner(address) should be declared external:
        - CarbonMembership.updateOwner(address) (MembershipTrader.sol#1210-1212)
```

```
updateOwner(address) should be declared external:
        - CarbonMembership.updateOwner(address) (MembershipTrader.sol#1210-1212)
executeOrder(address) should be declared external:
        - MembershipTrader.executeOrder(address) (MembershipTrader.sol#1240-1249)
withdrawGEMS() should be declared external:
        - MembershipTrader.withdrawGEMS() (MembershipTrader.sol#1251-1254)
updateOwner(address) should be declared external:
        - MembershipTrader.updateOwner(address) (MembershipTrader.sol#1256-1258)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#public-function-that-could-be-declared-external
INFO:Slither:MembershipTrader.sol analyzed (16 contracts with 75 detectors), 64 result(s) found
INFO:Slither:Use https://crytic.io/ to get access to additional detectors and Github integration
```

Slither log >> ERC721NFTContract.sol

```
INFO:Detectors:
ERC721NFTContract.constructor(string,string,address)._name (ERC721NFTContract.sol#1047) shadows:
        - ERC721._name (ERC721NFTContract.sol#544) (state variable)
ERC721NFTContract.constructor(string,string,address)._symbol (ERC721NFTContract.sol#1048) shadows:
        - ERC721._symbol (ERC721NFTContract.sol#547) (state variable)
ERC721NFTContract.mint().tokenURI (ERC721NFTContract.sol#1059-1061) shadows:
        - ERC721URIStorage.tokenURI(uint256) (ERC721NFTContract.sol#978-994) (function)
        - ERC721.tokenURI(uint256) (ERC721NFTContract.sol#613-618) (function)
        - IERC721Metadata.tokenURI(uint256) (ERC721NFTContract.sol#527) (function)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#local-variable-shadowing
```

```
INFO:Detectors:
ERC721NFTContract.changeAdmin(address) (ERC721NFTContract.sol#1073-1076) should emit an event for:
        - admin = _newAdmin (ERC721NFTContract.sol#1075)
ERC721NFTContract.updateFactory(address) (ERC721NFTContract.sol#1078-1080) should emit an event for:
        - factory = _factory (ERC721NFTContract.sol#1079)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#missing-events-access-control
INFO:Detectors:
ERC721NFTContract.constructor(string,string,address)._admin (ERC721NFTContract.sol#1049) lacks a zero-check on :
                - admin = _admin (ERC721NFTContract.sol#1051)
ERC721NFTContract.updateFactory(address)._factory (ERC721NFTContract.sol#1078) lacks a zero-check on :
                - factory = _factory (ERC721NFTContract.sol#1079)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#missing-zero-address-validation
INFO:Detectors:
Variable 'ERC721._checkOnERC721Received(address,address,uint256,bytes).retval (ERC721NFTContract.sol#915)' in ERC721._checkO
nERC721Received(address,address,uint256,bytes) (ERC721NFTContract.sol#908-929) potentially used before declaration: retval =
= IERC721Receiver.onERC721Received.selector (ERC721NFTContract.sol#916)
Variable 'ERC721._checkOnERC721Received(address,address,uint256,bytes).reason (ERC721NFTContract.sol#917)' in ERC721._checkO
nERC721Received(address,address,uint256,bytes) (ERC721NFTContract.sol#908-929) potentially used before declaration: reason.l
ength == 0 (ERC721NFTContract.sol#918)
Variable 'ERC721._checkOnERC721Received(address,address,uint256,bytes).reason (ERC721NFTContract.sol#917)' in ERC721._checkO
nERC721Received(address,address,uint256,bytes) (ERC721NFTContract.sol#908-929) potentially used before declaration: revert(u
int256,uint256)(32 + reason,mload(uint256)(reason)) (ERC721NFTContract.sol#922)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#pre-declaration-usage-of-local-variables
INFO:Detectors:
Address.verifyCallResult(bool,bytes,string) (ERC721NFTContract.sol#257-277) uses assembly
        - INLINE ASM (ERC721NFTContract.sol#269-272)
ERC721._checkOnERC721Received(address,address,uint256,bytes) (ERC721NFTContract.sol#908-929) uses assembly
        - INLINE ASM (ERC721NFTContract.sol#921-923)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#assembly-usage
INFO:Detectors:
Address.functionCall(address,bytes) (ERC721NFTContract.sol#141-143) is never used and should be removed
Address.functionCall(address,bytes,string) (ERC721NFTContract.sol#151-157) is never used and should be removed
Address.functionCallWithValue(address,bytes,uint256) (ERC721NFTContract.sol#170-176) is never used and should be removed
```

```
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#dead-code
INFO:Detectors:
Pragma version^0.8.4 (ERC721NFTContract.sol#2) necessitates a version too recent to be trusted. Consider deploying with 0.6.
12/0.7.6
solc-0.8.4 is not recommended for deployment
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#incorrect-versions-of-solidity
INFO:Detectors:
Low level call in Address.sendValue(address,uint256) (ERC721NFTContract.sol#116-121):
        - (success) = recipient.call{value: amount}() (ERC721NFTContract.sol#119)
Low level call in Address.functionCallWithValue(address,bytes,uint256,string) (ERC721NFTContract.sol#184-195):
        - (success,returndata) = target.call{value: value}(data) (ERC721NFTContract.sol#193)
Low level call in Address.functionStaticCall(address,bytes,string) (ERC721NFTContract.sol#213-222):
        - (success,returndata) = target.staticcall(data) (ERC721NFTContract.sol#220)
Low level call in Address.functionDelegateCall(address,bytes,string) (ERC721NFTContract.sol#240-249):
        - (success,returndata) = target.delegatecall(data) (ERC721NFTContract.sol#247)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#low-level-calls
INFO:Detectors:
Parameter ERC721.safeTransferFrom(address,address,uint256,bytes)._data (ERC721NFTContract.sol#699) is not in mixedCase
Parameter ERC721NFTContract.changeAdmin(address)._newAdmin (ERC721NFTContract.sol#1073) is not in mixedCase
Parameter ERC721NFTContract.updateFactory(address)._factory (ERC721NFTContract.sol#1078) is not in mixedCase
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#conformance-to-solidity-naming-conventions
INFO:Detectors:
ERC721NFTContract.baseURI (ERC721NFTContract.sol#1034) should be constant
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#state-variables-that-could-be-declared-constant
INFO:Detectors:
balanceOf(address) should be declared external:
        - ERC721.balanceOf(address) (ERC721NFTContract.sol#582-585)
name() should be declared external:
        - ERC721.name() (ERC721NFTContract.sol#599-601)
symbol() should be declared external:
        - ERC721.symbol() (ERC721NFTContract.sol#606-608)
approve(address,uint256) should be declared external:
        - ERC721.approve(address,uint256) (ERC721NFTContract.sol#632-642)
setApprovalForAll(address,bool) should be declared external:
        - ERC721.setApprovalForAll(address,bool) (ERC721NFTContract.sol#656-658)
transferFrom(address,address,uint256) should be declared external:
        - ERC721.transferFrom(address,address,uint256) (ERC721NFTContract.sol#670-679)
```

```
transferFrom(address,address,uint256) should be declared external:
        - ERC721.transferFrom(address,address,uint256) (ERC721NFTContract.sol#670-679)
safeTransferFrom(address,address,uint256) should be declared external:
        - ERC721.safeTransferFrom(address,address,uint256) (ERC721NFTContract.sol#684-690)
mint() should be declared external:
        - ERC721NFTContract.mint() (ERC721NFTContract.sol#1055-1067)
getTotalNFTs() should be declared external:
        - ERC721NFTContract.getTotalNFTs() (ERC721NFTContract.sol#1069-1071)
changeAdmin(address) should be declared external:
        - ERC721NFTContract.changeAdmin(address) (ERC721NFTContract.sol#1073-1076)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#public-function-that-could-be-declared-external
INFO:Slither:ERC721NFTContract.sol analyzed (13 contracts with 75 detectors), 52 result(s) found
INFO:Slither:Use https://crytic.io/ to get access to additional detectors and Github integration
```

Slither log >> MintingFactory.sol

```
INFO:Detectors:
ERC721NFTContract.constructor(string,string,address)._name (MintingFactory.sol#1049) shadows:
        - ERC721._name (MintingFactory.sol#546) (state variable)
ERC721NFTContract.constructor(string,string,address)._symbol (MintingFactory.sol#1050) shadows:
        - ERC721._symbol (MintingFactory.sol#549) (state variable)
ERC721NFTContract.mint().tokenURI (MintingFactory.sol#1061-1063) shadows:
        - ERC721URIStorage.tokenURI(uint256) (MintingFactory.sol#980-996) (function)
        - ERC721.tokenURI(uint256) (MintingFactory.sol#615-620) (function)
        - IERC721Metadata.tokenURI(uint256) (MintingFactory.sol#529) (function)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#local-variable-shadowing
```

```
INFO:Detectors:
ERC721NFTContract.changeAdmin(address) (MintingFactory.sol#1075-1078) should emit an event for:
        - admin = _newAdmin (MintingFactory.sol#1077)
ERC721NFTContract.updateFactory(address) (MintingFactory.sol#1080-1082) should emit an event for:
        - factory = _factory (MintingFactory.sol#1081)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#missing-events-access-control
INFO:Detectors:
ERC721NFTContract.constructor(string,string,address)._admin (MintingFactory.sol#1051) lacks a zero-check on :
                - admin = _admin (MintingFactory.sol#1053)
ERC721NFTContract.updateFactory(address)._factory (MintingFactory.sol#1080) lacks a zero-check on :
                - factory = _factory (MintingFactory.sol#1081)
MintingFactory.constructor(address,address)._eth (MintingFactory.sol#1394) lacks a zero-check on :
                - ETH = _eth (MintingFactory.sol#1395)
MintingFactory.updateExchangeAddress(address)._newExchange (MintingFactory.sol#1466) lacks a zero-check on :
                - exchangeAddress = _newExchange (MintingFactory.sol#1468)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#missing-zero-address-validation
INFO:Detectors:
Variable 'ERC721._checkOnERC721Received(address,address,uint256,bytes).retval (MintingFactory.sol#917)' in ERC721._checkOnER
C721Received(address,address,uint256,bytes) (MintingFactory.sol#910-931) potentially used before declaration: retval == IERC
721Receiver.onERC721Received.selector (MintingFactory.sol#918)
Variable 'ERC721._checkOnERC721Received(address,address,uint256,bytes).reason (MintingFactory.sol#919)' in ERC721._checkOnER
C721Received(address,address,uint256,bytes) (MintingFactory.sol#910-931) potentially used before declaration: reason.length
== 0 (MintingFactory.sol#920)
Variable 'ERC721._checkOnERC721Received(address,address,uint256,bytes).reason (MintingFactory.sol#919)' in ERC721._checkOnER
C721Received(address,address,uint256,bytes) (MintingFactory.sol#910-931) potentially used before declaration: revert(uint256
,uint256)(32 + reason,mload(uint256)(reason)) (MintingFactory.sol#924)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#pre-declaration-usage-of-local-variables
INFO:Detectors:
Reentrancy in MintingFactory.createNFTContract(string,string,address) (MintingFactory.sol#1427-1444):
        External calls:
        - ERC721NFTContract(nftContract).setApprovalForAll(exchangeAddress,true) (MintingFactory.sol#1439)
        Event emitted after the call(s):
        - NFTContractCreated(_name,_symbol,nftContract,_creator) (MintingFactory.sol#1441)
Reentrancy in MintingFactory.mintNFT(address) (MintingFactory.sol#1446-1453):
        External calls:
        - _tokenId = ERC721NFTContract(_nftContract).mint() (MintingFactory.sol#1450)
        - NFTMinted(_nftContract,_tokenId) (MintingFactory.sol#1452)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#reentrancy-vulnerabilities-3
INFO:Detectors:
Address.verifyCallResult(bool,bytes,string) (MintingFactory.sol#329-349) uses assembly
        - INLINE ASM (MintingFactory.sol#341-344)
ERC721._checkOnERC721Received(address,address,uint256,bytes) (MintingFactory.sol#910-931) uses assembly
        - INLINE ASM (MintingFactory.sol#923-925)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#assembly-usage
INFO:Detectors:
AccessControl._setRoleAdmin(bytes32,bytes32) (MintingFactory.sol#1324-1328) is never used and should be removed
Address.functionCall(address,bytes) (MintingFactory.sol#213-215) is never used and should be removed
Address.functionCall(address,bytes,string) (MintingFactory.sol#223-229) is never used and should be removed
Address.functionCallWithValue(address,bytes,uint256) (MintingFactory.sol#242-248) is never used and should be removed
Address.functionCallWithValue(address,bytes,uint256,string) (MintingFactory.sol#256-267) is never used and should be removed
Address.functionDelegateCall(address,bytes) (MintingFactory.sol#302-304) is never used and should be removed
Address.functionDelegateCall(address,bytes,string) (MintingFactory.sol#312-321) is never used and should be removed
Address.functionStaticCall(address,bytes) (MintingFactory.sol#275-277) is never used and should be removed
INFO:Detectors:
ERC721NFTContract.baseURI (MintingFactory.sol#1036) should be constant
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#state-variables-that-could-be-declared-constant
INFO:Detectors:
balanceOf(address) should be declared external:
        - ERC721.balanceOf(address) (MintingFactory.sol#584-587)
name() should be declared external:
        - ERC721.name() (MintingFactory.sol#601-603)
symbol() should be declared external:
        - ERC721.symbol() (MintingFactory.sol#608-610)
approve(address,uint256) should be declared external:
        - ERC721.approve(address,uint256) (MintingFactory.sol#634-644)
setApprovalForAll(address,bool) should be declared external:
        - ERC721.setApprovalForAll(address,bool) (MintingFactory.sol#658-660)
transferFrom(address,address,uint256) should be declared external:
        - ERC721.transferFrom(address,address,uint256) (MintingFactory.sol#672-681)
safeTransferFrom(address,address,uint256) should be declared external:
        - ERC721.safeTransferFrom(address,address,uint256) (MintingFactory.sol#686-692)
mint() should be declared external:
        - ERC721NFTContract.mint() (MintingFactory.sol#1057-1069)
getTotalNFTs() should be declared external:
        - ERC721NFTContract.getTotalNFTs() (MintingFactory.sol#1071-1073)
changeAdmin(address) should be declared external:
        - ERC721NFTContract.changeAdmin(address) (MintingFactory.sol#1075-1078)
mintNFT(address) should be declared external:
        - MintingFactory.mintNFT(address) (MintingFactory.sol#1446-1453)
updateOwner(address,uint256,address) should be declared external:
        - MintingFactory.updateOwner(address,uint256,address) (MintingFactory.sol#1456-1464)
updateExchangeAddress(address) should be declared external:
        - MintingFactory.updateExchangeAddress(address) (MintingFactory.sol#1466-1470)
getNFTsForOwner(address) should be declared external:
        - MintingFactory.getNFTsForOwner(address) (MintingFactory.sol#1473-1479)
getTotalNFTsMinted(address) should be declared external:
        - MintingFactory.getTotalNFTsMinted(address) (MintingFactory.sol#1482-1488)
getNFTsForOwner(address) should be declared external:
        - MintingFactory.getNFTsForOwner(address) (MintingFactory.sol#1473-1479)
getTotalNFTsMinted(address) should be declared external:
        - MintingFactory.getTotalNFTsMinted(address) (MintingFactory.sol#1482-1488)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#public-function-that-could-be-declared-external
INFO:Slither:MintingFactory.sol analyzed (17 contracts with 75 detectors), 73 result(s) found
INFO:Slither:Use https://crytic.io/ to get access to additional detectors and Github integration
```

Slither log >> ExchangeCore.sol

```
INFO:Detectors:
ExchangeCore.constructor(address,address,address,address)._mintingFactory (ExchangeCore.sol#872) lacks a zero-check on :
                - mintingFactory = _mintingFactory (ExchangeCore.sol#877)
ExchangeCore.constructor(address,address,address,address)._eth (ExchangeCore.sol#873) lacks a zero-check on :
                - ETH = _eth (ExchangeCore.sol#878)
ExchangeCore.constructor(address,address,address,address)._carbonMembership (ExchangeCore.sol#874) lacks a zero-check on :
                - carbonMembership = _carbonMembership (ExchangeCore.sol#879)
ExchangeCore.updateFactory(address)._factory (ExchangeCore.sol#1065) lacks a zero-check on :
                - mintingFactory = _factory (ExchangeCore.sol#1066)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#missing-zero-address-validation
INFO:Detectors:
Reentrancy in ExchangeCore._executeOrder(uint256,uint256,address,uint256,address,address,uint256) (ExchangeCore.sol#1007-103
6):
        External calls:
        - IERC20(ETH).transferFrom(_buyer,carbonFeeVault,_totalCarbonFee) (ExchangeCore.sol#1017)
        - IERC20(ETH).transferFrom(_buyer,_seller,_creatorRoyalties) (ExchangeCore.sol#1020)
        - IERC721NFTContract(_nftContract).transferFrom(_seller,_buyer,_tokenId) (ExchangeCore.sol#1023-1027)
        - IMintingFactory(mintingFactory).updateOwner(_nftContract,_tokenId,_buyer) (ExchangeCore.sol#1029-1033)
        Event emitted after the call(s):
        - OrderExecuted(_nftContract,_tokenId,_seller,_buyer,_totalCarbonFee,_creatorRoyalties,_mode) (ExchangeCore.sol#1035
)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#reentrancy-vulnerabilities-3
INFO:Detectors:
ExchangeCore.executeOrder(address,uint256,address,address,uint256,uint256,uint256) (ExchangeCore.sol#942-1005) uses timestam
p for comparisons
        Dangerous comparisons:
        - require(bool,string)(_auctionEndTime > block.timestamp,Auction has ended) (ExchangeCore.sol#952)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#block-timestamp
INFO:Detectors:
AccessControl._setRoleAdmin(bytes32,bytes32) (ExchangeCore.sol#796-800) is never used and should be removed
Context._msgData() (ExchangeCore.sol#353-355) is never used and should be removed
SafeMath.add(uint256,uint256) (ExchangeCore.sol#212-214) is never used and should be removed
```

```
Strings.toHexString(uint256) (ExchangeCore.sol#520-531) is never used and should be removed
Strings.toString(uint256) (ExchangeCore.sol#495-515) is never used and should be removed
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#dead-code
INFO:Detectors:
Pragma version^0.8.4 (ExchangeCore.sol#3) necessitates a version too recent to be trusted. Consider deploying with 0.6.12/0.
7.6
solc-0.8.4 is not recommended for deployment
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#incorrect-versions-of-solidity
INFO:Detectors:
Parameter ExchangeCore.validateSeller(address,uint256,address)._nftContract (ExchangeCore.sol#902) is not in mixedCase
Parameter ExchangeCore.validateSeller(address,uint256,address)._tokenId (ExchangeCore.sol#903) is not in mixedCase
Parameter ExchangeCore.validateSeller(address,uint256,address)._seller (ExchangeCore.sol#904) is not in mixedCase
Parameter ExchangeCore.validateBuyer(address,uint256)._buyer (ExchangeCore.sol#926) is not in mixedCase
Parameter ExchangeCore.validateBuyer(address,uint256)._amount (ExchangeCore.sol#926) is not in mixedCase
Parameter ExchangeCore.executeOrder(address,uint256,address,address,uint256,uint256,uint256)._nftContract (ExchangeCore.sol#
943) is not in mixedCase
Parameter ExchangeCore.executeOrder(address,uint256,address,address,uint256,uint256,uint256)._tokenId (ExchangeCore.sol#944)
 is not in mixedCase
Parameter ExchangeCore.executeOrder(address,uint256,address,address,uint256,uint256,uint256)._buyer (ExchangeCore.sol#945) i
s not in mixedCase
Parameter ExchangeCore.executeOrder(address,uint256,address,address,uint256,uint256,uint256)._seller (ExchangeCore.sol#946)
is not in mixedCase
Parameter ExchangeCore.executeOrder(address,uint256,address,address,uint256,uint256,uint256)._amount (ExchangeCore.sol#947)
is not in mixedCase
```

```
Parameter ExchangeCore.executeOrder(address,uint256,address,address,uint256,uint256,uint256)._mode (ExchangeCore.sol#949) is
 not in mixedCase
Parameter ExchangeCore.cancelOrder(address,uint256,address)._nftContract (ExchangeCore.sol#1039) is not in mixedCase
Parameter ExchangeCore.cancelOrder(address,uint256,address)._tokenId (ExchangeCore.sol#1040) is not in mixedCase
Parameter ExchangeCore.cancelOrder(address,uint256,address)._buyer (ExchangeCore.sol#1041) is not in mixedCase
Parameter ExchangeCore.uncancelOrder(address,uint256,address)._nftContract (ExchangeCore.sol#1053) is not in mixedCase
Parameter ExchangeCore.uncancelOrder(address,uint256,address)._tokenId (ExchangeCore.sol#1054) is not in mixedCase
Parameter ExchangeCore.uncancelOrder(address,uint256,address)._buyer (ExchangeCore.sol#1055) is not in mixedCase
Parameter ExchangeCore.updateFactory(address)._factory (ExchangeCore.sol#1065) is not in mixedCase
Parameter ExchangeCore.setCarbonFeeVaultAddress(address)._carbonFeeVault (ExchangeCore.sol#1069) is not in mixedCase
Variable ExchangeCore.ETH (ExchangeCore.sol#862) is not in mixedCase
Variable ExchangeCore.BUYERS_PREMIUM_FEES (ExchangeCore.sol#867) is not in mixedCase
Constant ExchangeCore.BaseFactorMax (ExchangeCore.sol#868) is not in UPPER_CASE_WITH_UNDERSCORES
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#conformance-to-solidity-naming-conventions
INFO:Detectors:
ExchangeCore.BUYERS_PREMIUM_FEES (ExchangeCore.sol#867) should be constant
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#state-variables-that-could-be-declared-constant
INFO:Detectors:
executeOrder(address,uint256,address,address,uint256,uint256,uint256) should be declared external:
        - ExchangeCore.executeOrder(address,uint256,address,address,uint256,uint256,uint256) (ExchangeCore.sol#942-1005)
cancelOrder(address,uint256,address) should be declared external:
        - ExchangeCore.cancelOrder(address,uint256,address) (ExchangeCore.sol#1038-1050)
uncancelOrder(address,uint256,address) should be declared external:
        - ExchangeCore.uncancelOrder(address,uint256,address) (ExchangeCore.sol#1052-1063)
pause() should be declared external:
        - ExchangeCore.pause() (ExchangeCore.sol#1077-1079)
unpause() should be declared external:
        - ExchangeCore.unpause() (ExchangeCore.sol#1081-1083)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#public-function-that-could-be-declared-external
INFO:Slither:ExchangeCore.sol analyzed (15 contracts with 75 detectors), 54 result(s) found
INFO:Slither:Use https://crytic.io/ to get access to additional detectors and Github integration
```

Solidity Static Analysis

ETHToken.sol

Gas & Economy

Gas costs:

Gas requirement of function ERC20.name 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: 51:4:

Miscellaneous

Constant/View/Pure functions:

IERC20.transfer(address,uint256) : Potentially should be constant/view/pure but is not.
more
Pos: 10:4:

Similar variable names:

ERC20._burn(address,uint256) : Variables have very similar names "account" and "amount".
Pos: 187:43:

No return:

IERC20.totalSupply(): Defines a return type but never explicitly returns a value.
Pos: 6:4:

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: 209:12:

Email: audit@EtherAuthority.io

AdminRole.sol

Gas & Economy

Gas costs:

Gas requirement of function AdminRole.getRoleAdmin 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: 256:4:

Gas costs:

Gas requirement of function AdminRole.removeAdmin 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: 390:4:

Miscellaneous

Constant/View/Pure functions:

AccessControl._checkRole(bytes32,address) : Is constant but potentially should not be. Note: Modifiers are currently not considered by this static analysis.
more
Pos: 235:4:

Similar variable names:

AccessControl._revokeRole(bytes32,address) : Variables have very similar names "_roles" and "role". Note: Modifiers are currently not considered by this static analysis.
Pos: 360:29:

Guard conditions:

Use "assert(x)" if you never ever want x to be false, not in any circumstance (apart from a bug in your code). Use "require(x)" if x can be false, due to e.g. invalid input or a failing external component.
more
Pos: 373:8:

GEMSNFTReceipt.sol

Security

Check-effects-interaction:

Potential violation of Checks-Effects-Interaction pattern in Address.functionCallWithValue(address,bytes,uint256,string): Could potentially lead to re-entrancy vulnerability. Note: Modifiers are currently not considered by this static analysis.
more
Pos: 154:4:

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: 893:20:

Gas & Economy

Gas costs:

Gas requirement of function ERC721.name 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: 571:4:

Gas costs:

Gas requirement of function GEMSNFTReceipt.burnNFT 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: 1039:4:

Delete dynamic array:

The "delete" operation when applied to a dynamically sized array in Solidity generates code to delete each of the elements contained. If the array is large, this operation can surpass the block gas limit and raise an OOG exception. Also nested dynamically sized objects can produce the same results.
more
Pos: 994:12:

Miscellaneous

Constant/View/Pure functions:

ERC721._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: 934:4:

Similar variable names:

GEMSNFTReceipt.burnNFT(uint256) : Variables have very similar names "_tokenIds" and "tokenId". Note: Modifiers are currently not considered by this static analysis.
Pos: 1040:14:

Guard conditions:

Use "assert(x)" if you never ever want x to be false, not in any circumstance (apart from a bug in your code). Use "require(x)" if x can be false, due to e.g. invalid input or a failing external component.
more
Pos: 1016:8:

GEMSStaking.sol

Security

Check-effects-interaction:

Potential violation of Checks-Effects-Interaction pattern in GEMSStaking.stake(address,uint256): Could potentially lead to re-entrancy vulnerability. Note: Modifiers are currently not considered by this static analysis.
more
Pos: 1098:4:

Gas & Economy

Gas costs:

Gas requirement of function ERC721.name 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: 600:4:

Gas costs:

Gas requirement of function GEMSStaking.unstake 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: 1120:4:

Delete dynamic array:

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

more

Pos: 1023:12:

Miscellaneous

Constant/View/Pure functions:

ERC721._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: 963:4:

Similar variable names:

GEMSNFTReceipt.burnNFT(uint256) : Variables have very similar names "_tokenIds" and "tokenId". Note: Modifiers are currently not considered by this static analysis.

Pos: 1069:14:

Guard conditions:

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

more

Pos: 1124:8:

Delete from dynamic array:

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

more

Pos: 1023:12:

Gas & Economy

Gas costs:

Gas requirement of function ERC20.name 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: 50:4:

Gas costs:

Gas requirement of function GEMSToken.decreaseAllowance is infinite: If the gas requirement of a function is higher than the block gas limit, it cannot be executed. Please avoid loops in your functions or actions that modify large areas of storage (this includes clearing or copying arrays in storage)
Pos: 129:4:

Miscellaneous

Constant/View/Pure functions:

IERC20.transferFrom(address,address,uint256) : Potentially should be constant/view/pure but is not.
more
Pos: 21:4:

Similar variable names:

ERC20._burn(address,uint256) : Variables have very similar names "account" and "amount".
Pos: 186:43:

Guard conditions:

Use "assert(x)" if you never ever want x to be false, not in any circumstance (apart from a bug in your code). Use "require(x)" if x can be false, due to e.g. invalid input or a failing external component.
more
Pos: 208:12:

CarbonMembership.sol

Security

Check-effects-interaction:

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

more

Pos: 185:4:

Gas & Economy

Gas costs:

Gas requirement of function CarbonMembership.name 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: 657:4:

Gas costs:

Gas requirement of function CarbonMembership.unpause 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: 1206:4:

Delete dynamic array:

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

more

Pos: 1155:12:

Email: audit@EtherAuthority.io

Miscellaneous

Constant/View/Pure functions:

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

more

Pos: 9:4:

Constant/View/Pure functions:

ERC721._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: 1020:4:

Similar variable names:

CarbonMembership.mintNewNFT(address) : Variables have very similar names "_tokenURIs" and "tokenURI". Note: Modifiers are currently not considered by this static analysis.

Pos: 1190:32:

Guard conditions:

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

more

Pos: 1171:8:

Delete from dynamic array:

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

more

Pos: 1155:12:

MembershipTrader.sol

Security

Check-effects-interaction:

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

more

Pos: 184:4:

Gas & Economy

Gas costs:

Gas requirement of function MembershipTrader.withdrawGEMS 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: 1251:4:

Delete dynamic array:

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

more

Pos: 1154:12:

Miscellaneous

Constant/View/Pure functions:

ERC721._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: 1019:4:

Email: audit@EtherAuthority.io

Similar variable names:

CarbonMembership.mintNewNFT(address) : Variables have very similar names "_tokenURIs" and "tokenURI". Note: Modifiers are currently not considered by this static analysis.
Pos: 1190:32:

Guard conditions:

Use "assert(x)" if you never ever want x to be false, not in any circumstance (apart from a bug in your code). Use "require(x)" if x can be false, due to e.g. invalid input or a failing external component.
more
Pos: 1243:8:

Delete from dynamic array:

Using "delete" on an array leaves a gap. The length of the array remains the same. If you want to remove the empty position you need to shift items manually and update the "length" property.
more
Pos: 1154:12:

ERC721NFTContract.sol

Security

Check-effects-interaction:

Potential violation of Checks-Effects-Interaction pattern in Address.functionCallWithValue(address,bytes,uint256,string): Could potentially lead to re-entrancy vulnerability. Note: Modifiers are currently not considered by this static analysis.
more
Pos: 184:4:

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: 921:20:

Gas & Economy

Gas costs:

Gas requirement of function ERC721NFTContract.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: 1055:4:

Delete dynamic array:

The "delete" operation when applied to a dynamically sized array in Solidity generates code to delete each of the elements contained. If the array is large, this operation can surpass the block gas limit and raise an OOG exception. Also nested dynamically sized objects can produce the same results.
more
Pos: 1022:12:

Miscellaneous

Constant/View/Pure functions:

ERC721._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: 962:4:

Similar variable names:

ERC721NFTContract.mint() : Variables have very similar names "_tokenURIs" and "tokenURI". Note: Modifiers are currently not considered by this static analysis.
Pos: 1064:32:

Guard conditions:

Use "assert(x)" if you never ever want x to be false, not in any circumstance (apart from a bug in your code). Use "require(x)" if x can be false, due to e.g. invalid input or a failing external component.
more
Pos: 1074:8:

MintingFactory.sol

Security

Check-effects-interaction:

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

more

Pos: 256:4:

Check-effects-interaction:

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

more

Pos: 1427:4:

Gas & Economy

Gas costs:

Gas requirement of function ERC721.name 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: 601:4:

Gas costs:

Gas requirement of function MintingFactory.transferFunds 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: 1490:4:

Email: audit@EtherAuthority.io

Delete dynamic array:

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

more

Pos: 1024:12:

Miscellaneous

Constant/View/Pure functions:

AccessControl._checkRole(bytes32,address) : Is constant but potentially should not be. Note: Modifiers are currently not considered by this static analysis.

more

Pos: 1225:4:

Similar variable names:

MintingFactory.createNFTContract(string,string,address) : Variables have very similar names "nftContract" and "_nftcontract". Note: Modifiers are currently not considered by this static analysis.

Pos: 1443:15:

No return:

IAccessControl.getRoleAdmin(bytes32): Defines a return type but never explicitly returns a value.

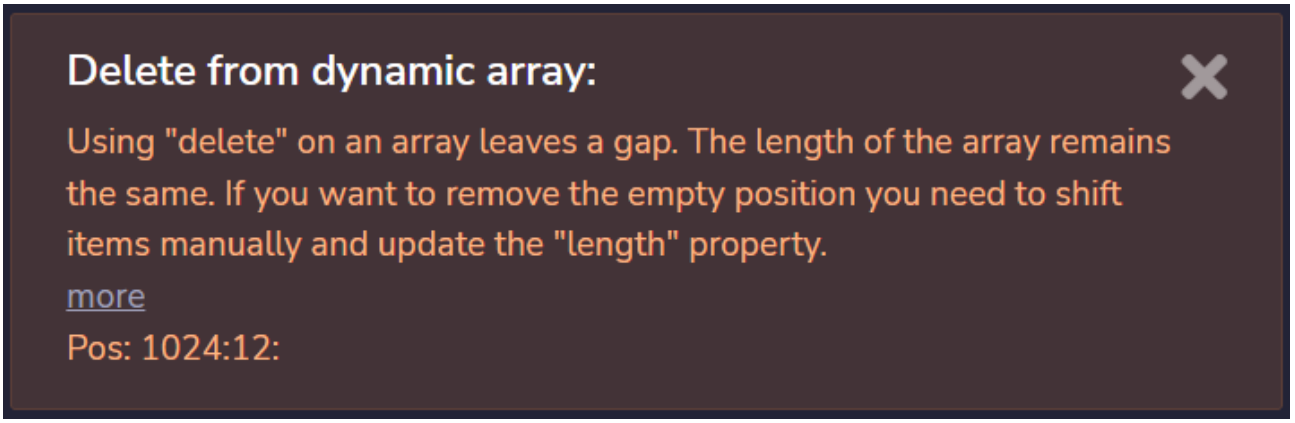
Pos: 1125:4:

Guard conditions:

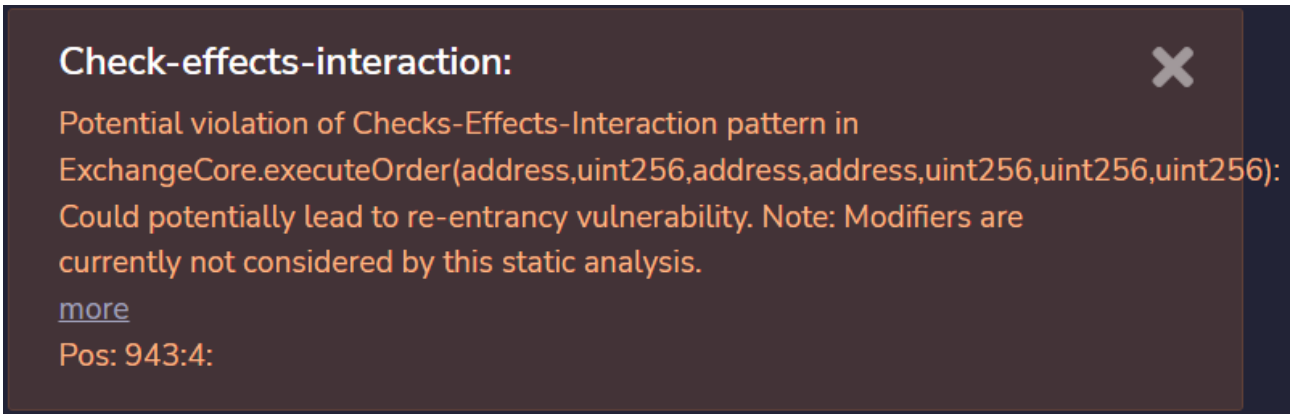
Use "assert(x)" if you never ever want x to be false, not in any circumstance (apart from a bug in your code). Use "require(x)" if x can be false, due to e.g. invalid input or a failing external component.

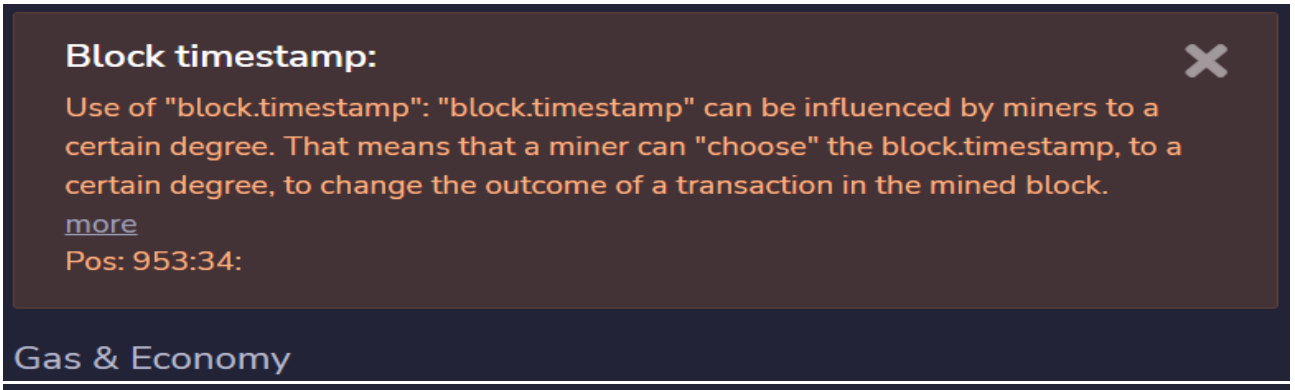
more

Pos: 1499:8:

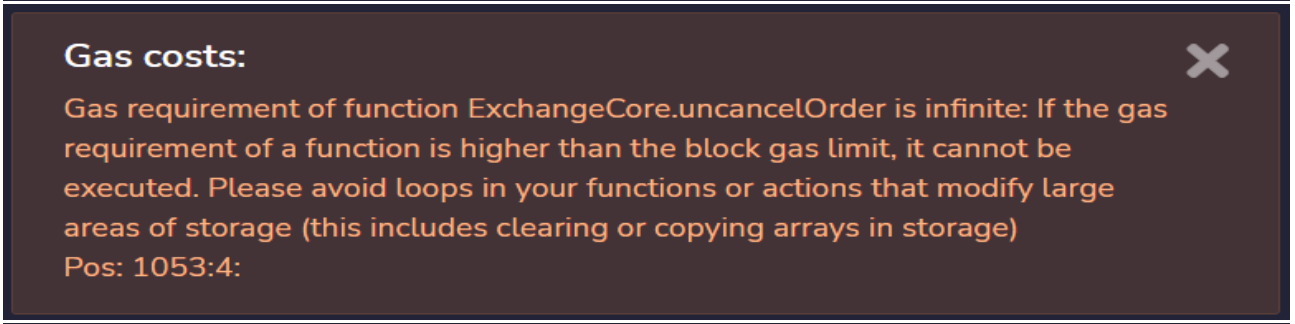


ExchangeCore.sol

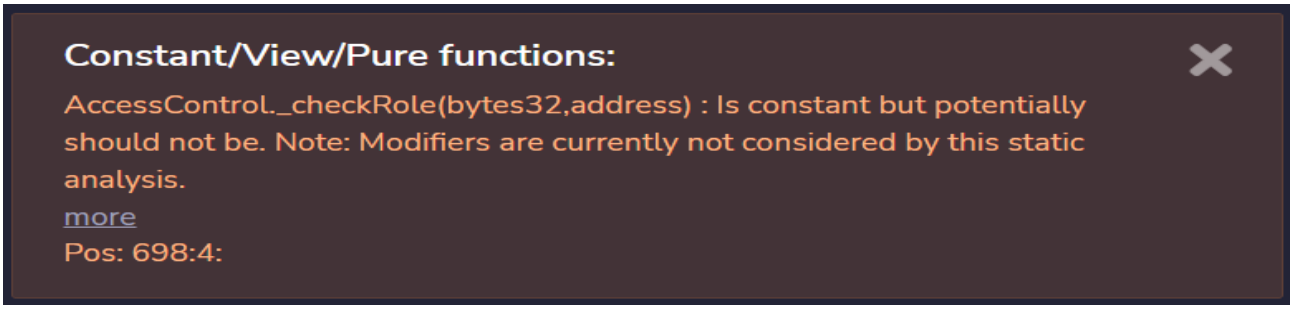




Gas & Economy



Miscellaneous



Similar variable names:

AccessControl._revokeRole(bytes32,address) : Variables have very similar names "_roles" and "role". Note: Modifiers are currently not considered by this static analysis.
Pos: 822:19:

Similar variable names:

AccessControl._revokeRole(bytes32,address) : Variables have very similar names "_roles" and "role". Note: Modifiers are currently not considered by this static analysis.
Pos: 823:29:

Guard conditions:

Use "assert(x)" if you never ever want x to be false, not in any circumstance (apart from a bug in your code). Use "require(x)" if x can be false, due to e.g. invalid input or a failing external component.
more
Pos: 958:8:

Guard conditions:

Use "assert(x)" if you never ever want x to be false, not in any circumstance (apart from a bug in your code). Use "require(x)" if x can be false, due to e.g. invalid input or a failing external component.
more
Pos: 962:8:

Guard conditions:

Use "assert(x)" if you never ever want x to be false, not in any circumstance (apart from a bug in your code). Use "require(x)" if x can be false, due to e.g. invalid input or a failing external component.
more
Pos: 1074:8:

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: 317:19:

Solhint Linter

ETHToken.sol

```
ETHToken.sol:141:18: Error: Parse error: missing ';' at '{'
ETHToken.sol:161:18: Error: Parse error: missing ';' at '{'
ETHToken.sol:182:18: Error: Parse error: missing ';' at '{'
ETHToken.sol:213:22: Error: Parse error: missing ';' at '{'
```

AdminRole.sol

```
AdminRole.sol:3:1: Error: Compiler version ^0.8.0 does not satisfy
the r semver requirement
AdminRole.sol:368:5: Error: Explicitly mark visibility in function
(Set ignoreConstructors to true if using solidity >=0.7.0)
```

GEMSNFTReceipt.sol

```
GEMSNFTReceipt.sol:18:18: Error: Parse error: missing ';' at '{'
GEMSNFTReceipt.sol:26:18: Error: Parse error: missing ';' at '{'
```

GEMSStaking.sol

```
GEMSStaking.sol:49:18: Error: Parse error: missing ';' at '{'
GEMSStaking.sol:57:18: Error: Parse error: missing ';' at '{'
```

GEMSToken.sol

```
GEMSToken.sol:140:18: Error: Parse error: missing ';' at '{'
GEMSToken.sol:160:18: Error: Parse error: missing ';' at '{'
GEMSToken.sol:181:18: Error: Parse error: missing ';' at '{'
GEMSToken.sol:212:22: Error: Parse error: missing ';' at '{'
```

CarbonMembership.sol

```
CarbonMembership.sol:49:18: Error: Parse error: missing ';' at '{'
CarbonMembership.sol:57:18: Error: Parse error: missing ';' at '{'
```

MembershipTrader.sol

```
MembershipTrader.sol:48:18: Error: Parse error: missing ';' at '{'
MembershipTrader.sol:56:18: Error: Parse error: missing ';' at '{'
```

ERC721NFTContract.sol

```
ERC721NFTContract.sol:48:18: Error: Parse error: missing ';' at '{'
ERC721NFTContract.sol:56:18: Error: Parse error: missing ';' at '{'
```

MintingFactory.sol

```
MintingFactory.sol:120:18: Error: Parse error: missing ';' at '{'
MintingFactory.sol:128:18: Error: Parse error: missing ';' at '{'
```

ExchangeCore.sol

```
ExchangeCore.sol:142:18: Error: Parse error: missing ';' at '{'
ExchangeCore.sol:155:18: Error: Parse error: missing ';' at '{'
ExchangeCore.sol:167:18: Error: Parse error: missing ';' at '{'
ExchangeCore.sol:184:18: Error: Parse error: missing ';' at '{'
ExchangeCore.sol:196:18: Error: Parse error: missing ';' at '{'
ExchangeCore.sol:292:18: Error: Parse error: missing ';' at '{'
ExchangeCore.sol:315:18: Error: Parse error: missing ';' at '{'
ExchangeCore.sol:341:18: 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.

Email: audit@EtherAuthority.io