

SMART CONTRACT

Security Audit Report

Project: Edov NFT Protocol
Website: <https://edoverse.io/>
Platform: Ethereum
Language: Solidity
Date: September 24th, 2022

Table of contents

Introduction	4
Project Background	4
Audit Scope	4
Claimed Smart Contract Features	5
Audit Summary	6
Technical Quick Stats	7
Code Quality	8
Documentation	8
Use of Dependencies	8
AS-IS overview	9
Severity Definitions	11
Audit Findings	12
Conclusion	16
Our Methodology	17
Disclaimers	19
Appendix	
• Code Flow Diagram	20
• Slither Results Log	23
• Solidity static analysis	26
• Solhint Linter	30

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

Introduction

EtherAuthority was contracted by Edov NFT protocol to perform the Security audit of the Edov NFT 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 September 24th, 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

Edov NFT is an NFT marketplace smart contract in which users can list/unlist their NFT, buy NFT by giving ERC20 tokens, etc. Edov NFT contract inherits IERC721, IERC721Receiver, IERC20, SafeMath, OwnableUpgradeable, Initializable 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 Edov NFT Protocol Smart Contracts
Platform	Ethereum / Solidity
File 1	EdoMarketplace.sol
File 1 MD5 Hash	8DBDEFA1FA11CEDD76F7339AC7B8EA47
Updated File 1 MD5 Hash	269D5D163EC08D1454347A508E478BE4
File 2	EdoMarketplaceManagement.sol
File 2 MD5 Hash	49864F564787FFCC543A2B672BD0C8D1
File 3	EdoMarketplaceManagementAdmin.sol
File 3 MD5 Hash	F7CA96695EB28A927E4A72BD336D5339
Updated File 3 MD5 Hash	1B0027A01C3B91F52D97E37BEA935D21
Audit Date	September 24th, 2022

Claimed Smart Contract Features

Claimed Feature Detail	Our Observation
<p>File 1 EdoMarketplace.sol</p> <ul style="list-style-type: none"> • EdoMarketplace can list NFT contract addresses. • EdoMarketplace owners can cancel trade. • EdoMarketplace can provide NFT information. • EdoMarketplace can provide listings for Edo NFT. 	<p>YES, This is valid.</p>
<p>File 2 EdoMarketplaceManagement.sol</p> <ul style="list-style-type: none"> • Minimum Fee Rate: 1% • Maximum Fee Rate: 10% • Non Agent Fee: 5% • Demominator: 100 • EdoMarketplaceManagement can set service fee rates. • EdoMarketplaceManagement owner can deposit fee amount. • EdoMarketplaceManagement owner can withdraw a token from the payment contract address. 	<p>YES, This is valid.</p>
<p>File 3 EdoMarketplaceManagementAdmin.sol</p> <ul style="list-style-type: none"> • EdoMarketplaceManagementAdmin owners can set deposit addresses, agent addresses, Edo verse foundation addresses, Edo verseInc addresses. • EdoMarketplaceManagementAdmin owners can set fee rate distribution. 	<p>YES, This is valid.</p>

Audit Summary

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



We used various tools like Slither, Solhint and Remix IDE. At the same time this finding is based on critical analysis of the manual audit.

All issues found during automated analysis were manually reviewed and applicable vulnerabilities are presented in the Audit overview section. General overview is presented in AS-IS section and all identified issues can be found in the Audit overview section.

We found 0 critical, 1 high, 1 medium and 1 low and some very low level issues.

All the issues have been resolved in the revised code.

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

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 3 smart contract files. Smart contracts contain Libraries, Smart contracts, inherits and Interfaces. This is a compact and well written smart contract.

The libraries in Edov NFT 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 Edov NFT Protocol.

The Edov NFT team has provided unit test scripts, which would have helped to determine the integrity of the code in an automated way.

All code parts are not well commented on smart contracts.

Documentation

We were given a Edov NFT smart contract code in the form of a file. The hash of that code is mentioned above in the table.

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

Another source of information was its official website <https://edoverse.io/> which provided rich information about the project architecture.

Use of Dependencies

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

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

AS-IS overview

EdoMarketplace.sol

Functions

Sl.	Functions	Type	Observation	Conclusion
1	constructor	write	Passed	No Issue
2	onlyEdoNft	modifier	Passed	No Issue
3	onlyNftOwner	modifier	Passed	No Issue
4	initialize	write	Passed	No Issue
5	approveErc20	external	Passed	No Issue
6	approveErc721	external	Passed	No Issue
7	setListingNFT	write	access only Edo Nft	No Issue
8	buyEdoNFT	write	access only Edo Nft	No Issue
9	cancelTrade	write	access only Nft Owner	No Issue
10	getEdoNFTInfo	write	access only Edo Nft	No Issue
11	getListingEdoNfts	read	Passed	No Issue
12	getHash	write	Passed	No Issue
13	setNftMeta	internal	Passed	No Issue
14	onlyAgent	modifier	Passed	No Issue
15	feeLimit	modifier	Passed	No Issue
16	setAgentRefundAddress	external	access only Agent	No Issue
17	setServiceFeeRate	write	Passed	No Issue
18	depositFeeAmount	external	access only Owner	No Issue
19	withdraw	external	access only Owner	No Issue

EdoMarketplaceManagement.sol

Functions

Sl.	Functions	Type	Observation	Conclusion
1	constructor	write	Passed	No Issue
2	onlyAgent	modifier	Passed	No Issue
3	feeLimit	modifier	Passed	No Issue
4	setAgentRefundAddress	external	access only Agent	No Issue
5	setServiceFeeRate	write	Passed	No Issue
6	depositFeeAmount	external	access only Owner	No Issue
7	withdraw	external	access only Owner	No Issue
8	setDepositAddress	external	access only Owner	No Issue
9	setAgentAddress	external	access only Owner	No Issue
10	setEdoVerseFoundationAddress	external	access only Owner	No Issue
11	setEdoVerseIncAddress	external	access only Owner	No Issue
12	includeNFTContractAddress	external	access only Owner	No Issue
13	excludeNFTContractAddress	external	access only Owner	No Issue

14	isApprovedNFTContractAddress	external	access only Owner	No Issue
15	includePaymentContractAddress	external	access only Owner	No Issue
16	excludePaymentContractAddress	external	access only Owner	No Issue
17	isApprovedPaymentContractAddress	external	access only Owner	No Issue
18	setFeeRateDistribution	external	access only Owner	No Issue

EdoMarketplaceManagementAdmin.sol

Functions

Sl.	Functions	Type	Observation	Conclusion
1	constructor	write	Passed	No Issue
2	setDepositAddress	external	access only Owner	No Issue
3	setAgentAddress	external	access only Owner	No Issue
4	setEdoVerseFoundationAddress	external	access only Owner	No Issue
5	setEdoVerseIncAddress	external	access only Owner	No Issue
6	includeNFTContractAddress	external	access only Owner	No Issue
7	excludeNFTContractAddress	external	access only Owner	No Issue
8	isApprovedNFTContractAddress	external	access only Owner	No Issue
9	includePaymentContractAddress	external	access only Owner	No Issue
10	excludePaymentContractAddress	external	access only Owner	No Issue
11	isApprovedPaymentContractAddress	external	access only Owner	No Issue
12	setFeeRateDistribution	external	access only Owner	No Issue
13	__Ownable_init	internal	access only Initializing	No Issue
14	__Ownable_init_unchained	internal	access only Initializing	No Issue
15	onlyOwner	modifier	Passed	No Issue
16	owner	read	Passed	No Issue
17	__checkOwner	internal	Passed	No Issue
18	renounceOwnership	write	access only Owner	No Issue
19	transferOwnership	write	access only Owner	No Issue
20	transferOwnership	internal	Passed	No Issue

Severity Definitions

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

Audit Findings

Critical Severity

No Critical severity vulnerabilities were found.

High Severity

(1) Wrong TotalFee calculation: [EdoMarketplaceManagementAdmin.sol](#)

```
function setFeeRateDistribution(  infinite gas
    uint256 _agentRefundFeeRate,
    uint256 _edoVerseFoundationFeeRate, 
    uint256 _edoVerseIncFeeRate
)
    external
    onlyOwner
{
    uint256 totalFeeRate = agentRefundFeeRate + edoVerseFoundationFeeRate + edoVerseIncFeeRate;
    require(totalFeeRate == 100, "EdoMarketplaceManagement: Total value is invalid.");

    agentRefundFeeRate = _agentRefundFeeRate; 
    edoVerseFoundationFeeRate = _edoVerseFoundationFeeRate; 
    edoVerseIncFeeRate = _edoVerseIncFeeRate; 
}
```

In the setFeeRateDistribution function, totalFeeRate is validated for the global variables which have been set while initialization. In case the admin has set wrong fees, totaling greater than 100, then admin cannot change fees again.

Resolution: We suggest using the input parameters for totaling and validation equal to 100 before setting the variables.

Status: Fixed.

Medium

(1) Irrelevant token standard and method: [EdoMarketplace.sol](#)

In the approveErc20 function, paymentContractAddress is IERC721 token standard and the approved method has the amount as the second parameter which is irrelevant. IERC721 should have tokenId as the second parameter.

Resolution: We suggest either to change the tokenStandard from IERC721 to IERC20 Or change the approved method's second parameter from amount to tokenId.

Status: Fixed.

Low

(1) Function input parameters lack of check: [EdoMarketplaceManagementAdmin.sol](#)

Variable validation is not performed in below functions:

- setDepositAddress = _depositAddress.

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

Status: Fixed.

Very Low / Informational / Best practices:

(1) Unused variable: [EdoMarketplaceManagement.sol](#)

NON_AGENT_FEE is defined as constant but not used anywhere.

Resolution: We suggest removing unused variables.

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:

- cancelTrade: EdoMarketplace owner can cancel trade.
- setAgentRefundAddress: EdoMarketplaceManagement agent owner can set agent refund address.
- setServiceFeeRate: EdoMarketplaceManagement agent owner can set service fee rate values.

- depositFeeAmount: EdoMarketplaceManagement owner can deposit fee amount.
- setDepositAddress: EdoMarketplaceManagementAdmin owner can set deposit address.
- setAgentAddress: EdoMarketplaceManagementAdmin owner can set agent address.
- setEdoVerseFoundationAddress: EdoMarketplaceManagementAdmin owner can set edo verse foundation address.
- setEdoVerseIncAddress: EdoMarketplaceManagementAdmin owner can set edo verse Inc address.
- includeNFTContractAddress: EdoMarketplaceManagementAdmin owner can include NFT contract address.
- excludeNFTContractAddress: EdoMarketplaceManagementAdmin owner can exclude NFT contract address.
- isApprovedNFTContractAddress: EdoMarketplaceManagementAdmin owner can set issapproved NFT contract address.
- includePaymentContractAddress: EdoMarketplaceManagementAdmin owner can include payment contract address.
- excludePaymentContractAddress: EdoMarketplaceManagementAdmin owner can exclude payment contract address.
- isApprovedPaymentContractAddress: EdoMarketplaceManagementAdmin owner can set approved payment contract address.
- setFeeRateDistribution: EdoMarketplaceManagementAdmin owner can set fee rate distribution value.

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

Conclusion

We were given a contract code in the form of a file. And we have used all possible tests based on given objects as files. We have observed 1 high severity issue, 1 medium severity issue, 1 low severity issue and some very low level issue in smart contracts. All the issues have been fixed in the revised code. **So smart contracts are ready for mainnet deployment.**

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

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

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

Security state of the reviewed contract, based on standard audit procedure scope, is **“Secure”**.

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.

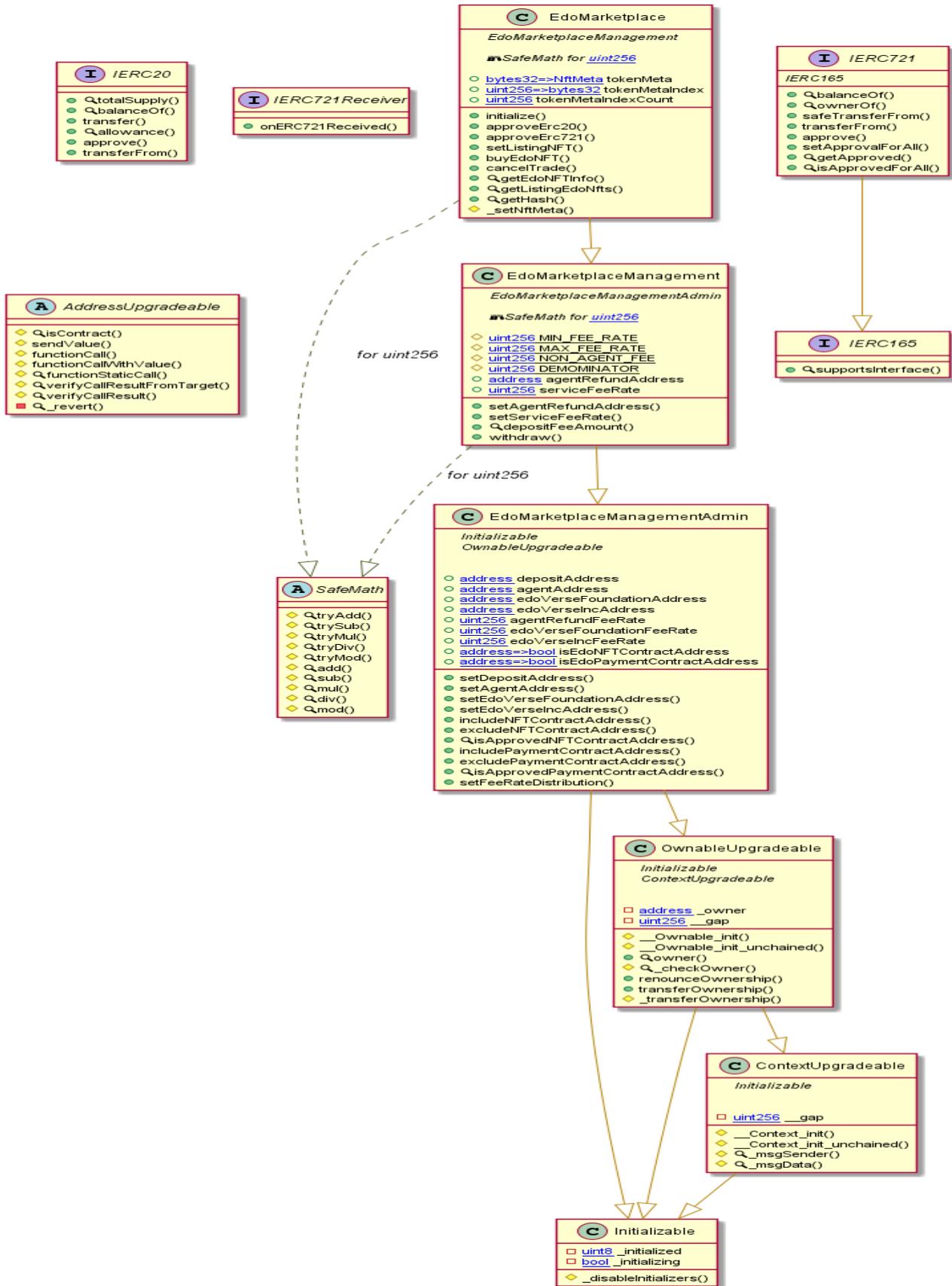
Technical Disclaimer

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 - EdoV NFT Protocol

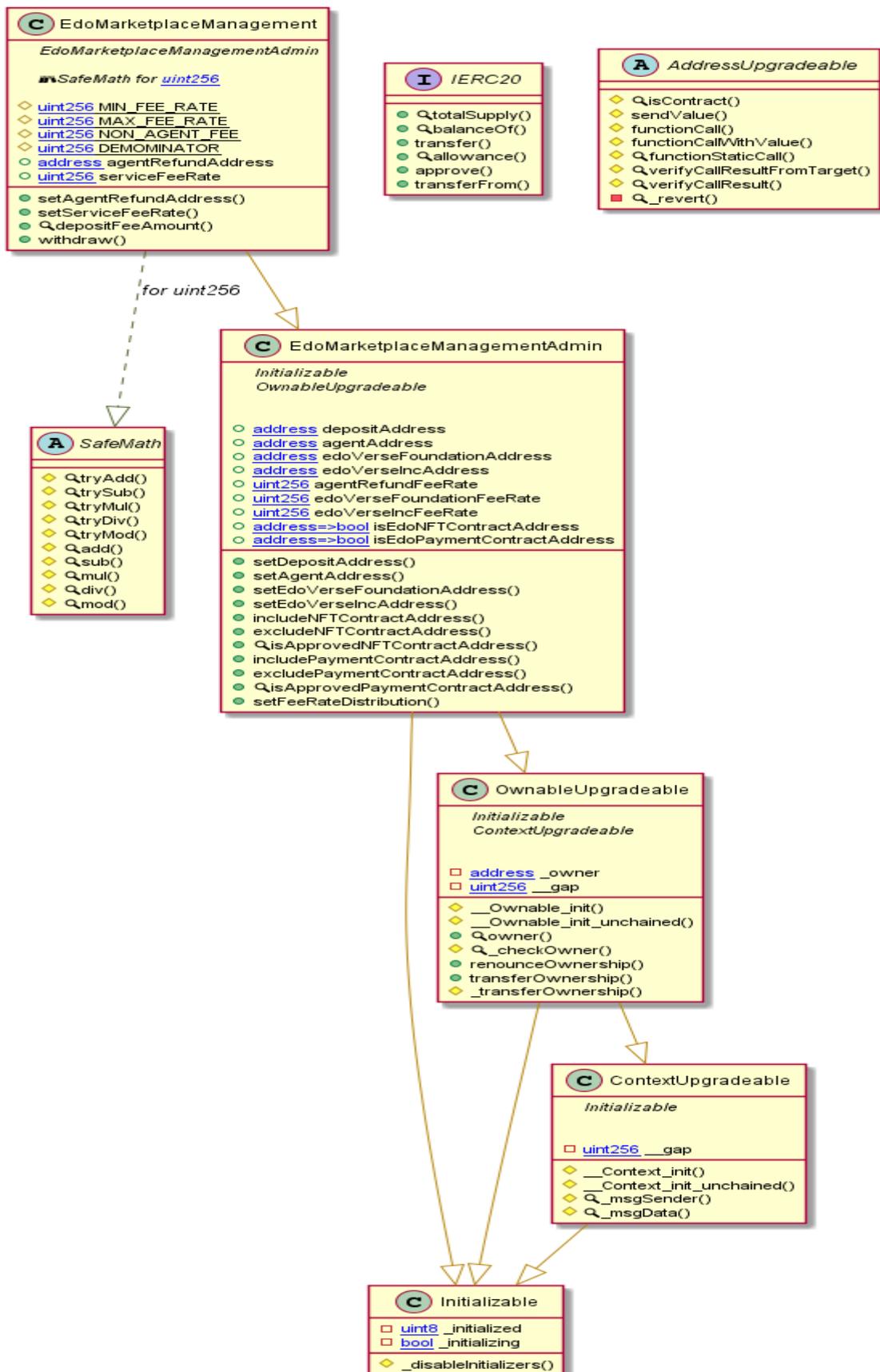
EdoMarketplace Diagram



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

Email: audit@EtherAuthority.io

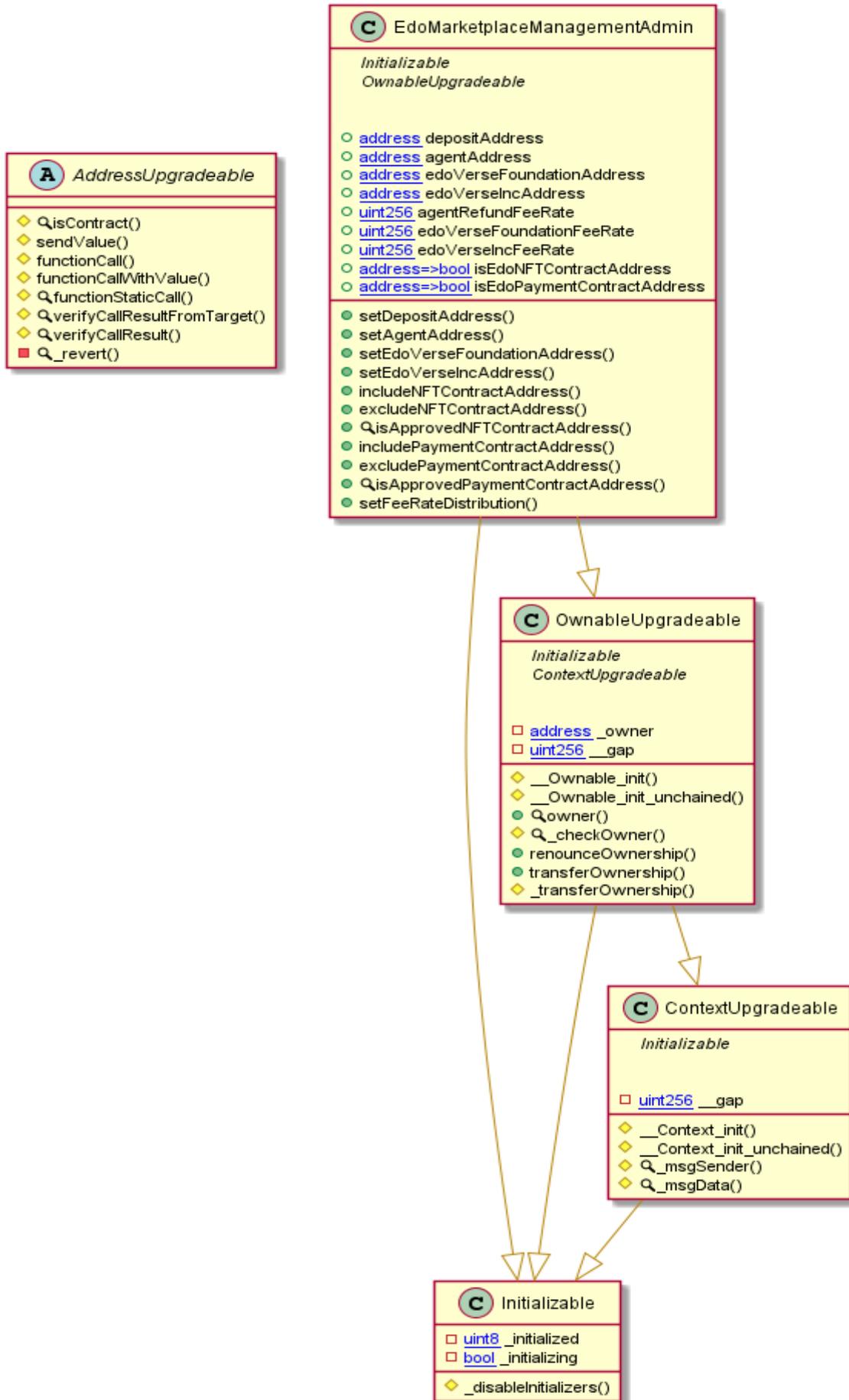
EdoMarketplaceManagement Diagram



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

Email: audit@EtherAuthority.io

EdoMarketplaceManagementAdmin Diagram



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

Email: audit@EtherAuthority.io

Slither Results Log

Slither log >> EdoMarketplace.sol

```
INFO:Detectors:
EdoMarketplace.getEdoNFTInfo(address,uint256).owner (EdoMarketplace.sol#976) shadows:
  - OwnableUpgradeable.owner() (EdoMarketplace.sol#642-644) (function)
EdoMarketplace._setNftMeta(bytes32,address,address,uint256,bool).owner (EdoMarketplace.sol#1018) shadows:
  - OwnableUpgradeable.owner() (EdoMarketplace.sol#642-644) (function)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#local-variable-shadowing
INFO:Detectors:
EdoMarketplace.initialize(uint256,uint256,uint256,uint256,address,address,address,address,address)._agentAddress (EdoMarketplace.sol#863) lacks a zero-check on:
  - agentAddress = _agentAddress (EdoMarketplace.sol#886)
EdoMarketplace.initialize(uint256,uint256,uint256,uint256,address,address,address,address,address)._agentRefundAddress (EdoMarketplace.sol#864) lacks a zero-check on:
  - agentRefundAddress = _agentRefundAddress (EdoMarketplace.sol#887)
EdoMarketplace.initialize(uint256,uint256,uint256,uint256,address,address,address,address,address)._edoVerseFoundationAddress (EdoMarketplace.sol#865) lacks a zero-check on:
  - edoVerseFoundationAddress = _edoVerseFoundationAddress (EdoMarketplace.sol#888)
EdoMarketplace.initialize(uint256,uint256,uint256,uint256,address,address,address,address,address,address)._edoVerseIncAddress (EdoMarketplace.sol#866) lacks a zero-check on:
  - edoVerseIncAddress = _edoVerseIncAddress (EdoMarketplace.sol#889)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#missing-zero-address-validation
INFO:Detectors:
AddressUpgradeable._revert(bytes,string) (EdoMarketplace.sol#539-548) uses assembly
  - INLINE ASM (EdoMarketplace.sol#541-544)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#assembly-usage
INFO:Detectors:
AddressUpgradeable._revert(bytes,string) (EdoMarketplace.sol#539-548) is never used and should be removed
AddressUpgradeable.functionCall(address,bytes) (EdoMarketplace.sol#467-469) is never used and should be removed
AddressUpgradeable.functionCall(address,bytes,string) (EdoMarketplace.sol#471-477) is never used and should be removed
AddressUpgradeable.functionCallWithValue(address,bytes,uint256) (EdoMarketplace.sol#479-485) is never used and should be removed
AddressUpgradeable.functionCallWithValue(address,bytes,uint256,string) (EdoMarketplace.sol#487-496) is never used and should be removed
AddressUpgradeable.functionStaticCall(address,bytes) (EdoMarketplace.sol#498-500) is never used and should be removed
AddressUpgradeable.functionStaticCall(address,bytes,string) (EdoMarketplace.sol#502-509) is never used and should be removed

INFO:Detectors:
Pragma version^0.8.4 (EdoMarketplace.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 AddressUpgradeable.sendValue(address,uint256) (EdoMarketplace.sol#460-465):
  - (success) = recipient.call{value: amount}() (EdoMarketplace.sol#463)
Low level call in AddressUpgradeable.functionCallWithValue(address,bytes,uint256,string) (EdoMarketplace.sol#487-496):
  - (success,returndata) = target.call{value: value}(data) (EdoMarketplace.sol#494)
Low level call in AddressUpgradeable.functionStaticCall(address,bytes,string) (EdoMarketplace.sol#502-509):
  - (success,returndata) = target.staticcall(data) (EdoMarketplace.sol#507)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#low-level-calls
INFO:Detectors:
Function ContextUpgradeable.__Context_init() (EdoMarketplace.sol#599-600) is not in mixedCase
Function ContextUpgradeable.__Context_init_unchained() (EdoMarketplace.sol#602-603) is not in mixedCase
Variable ContextUpgradeable.__gap (EdoMarketplace.sol#612) is not in mixedCase
Function OwnableUpgradeable.__Ownable_init() (EdoMarketplace.sol#623-625) is not in mixedCase
Function OwnableUpgradeable.__Ownable_init_unchained() (EdoMarketplace.sol#627-629) is not in mixedCase
Variable OwnableUpgradeable.__gap (EdoMarketplace.sol#688) is not in mixedCase
Parameter EdoMarketplaceManagementAdmin.setDepositAddress(address)._depositAddress (EdoMarketplace.sol#706) is not in mixedCase
Parameter EdoMarketplaceManagementAdmin.setAgentAddress(address)._agentAddress (EdoMarketplace.sol#714) is not in mixedCase
Parameter EdoMarketplaceManagementAdmin.setEdoVerseFoundationAddress(address)._edoVerseFoundationAddress (EdoMarketplace.sol#719) is not in mixedCase
Parameter EdoMarketplaceManagementAdmin.setEdoVerseIncAddress(address)._edoVerseIncAddress (EdoMarketplace.sol#724) is not in mixedCase
Parameter EdoMarketplaceManagementAdmin.includeNFTContractAddress(address)._NFTContractAddress (EdoMarketplace.sol#730) is not in mixedCase
Parameter EdoMarketplaceManagementAdmin.excludeNFTContractAddress(address)._NFTContractAddress (EdoMarketplace.sol#738) is not in mixedCase

Parameter EdoMarketplace.initialize(uint256,uint256,uint256,uint256,address,address,address,address,address)._edoNFTContractAddress (EdoMarketplace.sol#867) is not in mixedCase
Parameter EdoMarketplace.initialize(uint256,uint256,uint256,uint256,address,address,address,address,address)._edoPaymentContractAddress (EdoMarketplace.sol#868) is not in mixedCase
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#conformance-to-solidity-naming-conventions
INFO:Detectors:
EdoMarketplace (EdoMarketplace.sol#830-1042) does not implement functions:
  - ContextUpgradeable.__Context_init() (EdoMarketplace.sol#599-600)
  - ContextUpgradeable.__Context_init_unchained() (EdoMarketplace.sol#602-603)
  - OwnableUpgradeable.__Ownable_init() (EdoMarketplace.sol#623-625)
  - OwnableUpgradeable.__Ownable_init_unchained() (EdoMarketplace.sol#627-629)
  - OwnableUpgradeable._checkOwner() (EdoMarketplace.sol#649-651)
  - Initializable._disableInitializers() (EdoMarketplace.sol#589-595)
  - ContextUpgradeable._msgData() (EdoMarketplace.sol#608-610)
  - ContextUpgradeable._msgSender() (EdoMarketplace.sol#604-606)
  - EdoMarketplace._setNftMeta(bytes32,address,address,uint256,bool) (EdoMarketplace.sol#1016-1041)
  - OwnableUpgradeable._transferOwnership(address) (EdoMarketplace.sol#677-681)
  - EdoMarketplaceManagement.depositFeeAmount(address) (EdoMarketplace.sol#813-815)
  - EdoMarketplaceManagementAdmin.excludeNFTContractAddress(address) (EdoMarketplace.sol#738-740)
  - EdoMarketplaceManagementAdmin.excludePaymentContractAddress(address) (EdoMarketplace.sol#750-752)
  - EdoMarketplaceManagementAdmin.getHash(address,uint256) (EdoMarketplace.sol#1012-1014)
  - EdoMarketplaceManagementAdmin.includeNFTContractAddress(address) (EdoMarketplace.sol#729-736)
  - EdoMarketplaceManagementAdmin.includePaymentContractAddress(address) (EdoMarketplace.sol#746-748)
  - EdoMarketplaceManagementAdmin.isApprovedNFTContractAddress(address) (EdoMarketplace.sol#742-744)
  - EdoMarketplaceManagementAdmin.isApprovedPaymentContractAddress(address) (EdoMarketplace.sol#754-756)
  - OwnableUpgradeable.owner() (EdoMarketplace.sol#642-644)
  - OwnableUpgradeable.renounceOwnership() (EdoMarketplace.sol#660-662)
  - EdoMarketplaceManagementAdmin.setAgentAddress(address) (EdoMarketplace.sol#714-717)
  - EdoMarketplaceManagement.setAgentRefundAddress(address) (EdoMarketplace.sol#798-801)
  - EdoMarketplaceManagementAdmin.setDepositAddress(address) (EdoMarketplace.sol#705-712)
```

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

Email: audit@EtherAuthority.io

```

- EdoMarketplaceManagement.setAgentRefundAddress(address) (EdoMarketplace.sol#798-801)
- EdoMarketplaceManagementAdmin.setDepositAddress(address) (EdoMarketplace.sol#705-712)
- EdoMarketplaceManagementAdmin.setEdoVerseFoundationAddress(address) (EdoMarketplace.sol#719-722)
- EdoMarketplaceManagementAdmin.setEdoVerseIncAddress(address) (EdoMarketplace.sol#724-727)
- EdoMarketplaceManagementAdmin.setFeeRateDistribution(uint256,uint256,uint256) (EdoMarketplace.sol#758-772)
- EdoMarketplaceManagement.setServiceFeeRate(uint256) (EdoMarketplace.sol#803-811)
- OwnableUpgradeable.transferOwnership(address) (EdoMarketplace.sol#668-671)
- EdoMarketplaceManagement.withdraw(address) (EdoMarketplace.sol#817-827)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#unimplemented-functions
INFO:Detectors:
OwnableUpgradeable.__gap (EdoMarketplace.sol#688) is never used in EdoMarketplace (EdoMarketplace.sol#830-1042)
OwnableUpgradeable._owner (EdoMarketplace.sol#616) is never used in EdoMarketplace (EdoMarketplace.sol#830-1042)
EdoMarketplaceManagement.NON_AGENT_FEE (EdoMarketplace.sol#780) is never used in EdoMarketplace (EdoMarketplace.sol#830-1042)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#unused-state-variables
INFO:Detectors:
renounceOwnership() should be declared external:
- OwnableUpgradeable.renounceOwnership() (EdoMarketplace.sol#660-662)
transferOwnership(address) should be declared external:
- OwnableUpgradeable.transferOwnership(address) (EdoMarketplace.sol#668-671)
initialize(uint256,uint256,uint256,uint256,address,address,address,address,address,address,address,address,address) should be declared external:
- EdoMarketplace.initialize(uint256,uint256,uint256,uint256,address,address,address,address,address,address,address) (EdoMarketplace.sol#858-893)
setListingNFT(address,uint256,address,uint256) should be declared external:
- EdoMarketplace.setListingNFT(address,uint256,address,uint256) (EdoMarketplace.sol#903-926)
buyEdoNFT(address,uint256) should be declared external:
- EdoMarketplace.buyEdoNFT(address,uint256) (EdoMarketplace.sol#928-954)
cancelTrade(address,uint256) should be declared external:
- EdoMarketplace.cancelTrade(address,uint256) (EdoMarketplace.sol#956-966)
getEdoNFTInfo(address,uint256) should be declared external:
- EdoMarketplace.getEdoNFTInfo(address,uint256) (EdoMarketplace.sol#968-988)
getListingEdoNfts() should be declared external:
- EdoMarketplace.getListingEdoNfts() (EdoMarketplace.sol#990-1010)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#public-function-that-could-be-declared-external
INFO:Slither:EdoMarketplace.sol analyzed (12 contracts with 75 detectors), 92 result(s) found
INFO:Slither:Use https://crytic.io/ to get access to additional detectors and Github integration

```

Slither log >> EdoMarketplaceManagement.sol

```

INFO:Detectors:
EdoMarketplaceManagementAdmin.setDepositAddress(address)._depositAddress (EdoMarketplaceManagement.sol#544) lacks a zero-check on :
- depositAddress = _depositAddress (EdoMarketplaceManagement.sol#549)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#missing-zero-address-validation
INFO:Detectors:
AddressUpgradeable._revert(bytes,string) (EdoMarketplaceManagement.sol#377-386) uses assembly
- INLINE_ASM (EdoMarketplaceManagement.sol#379-382)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#assembly-usage

```

```

INFO:Detectors:
Pragma version^0.8.4 (EdoMarketplaceManagement.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 AddressUpgradeable.sendValue(address,uint256) (EdoMarketplaceManagement.sol#298-303):
- (success) = recipient.call{value: amount}() (EdoMarketplaceManagement.sol#301)
Low level call in AddressUpgradeable.functionCallWithValue(address,bytes,uint256,string) (EdoMarketplaceManagement.sol#325-334):
- (success,returndata) = target.call{value: value}(data) (EdoMarketplaceManagement.sol#332)
Low level call in AddressUpgradeable.functionStaticCall(address,bytes,string) (EdoMarketplaceManagement.sol#340-347):
- (success,returndata) = target.staticcall(data) (EdoMarketplaceManagement.sol#345)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#low-level-calls

```

```

INFO:Detectors:
Function ContextUpgradeable.__Context_init() (EdoMarketplaceManagement.sol#437-438) is not in mixedCase
Function ContextUpgradeable.__Context_init_unchained() (EdoMarketplaceManagement.sol#440-441) is not in mixedCase
Variable ContextUpgradeable.__gap (EdoMarketplaceManagement.sol#450) is not in mixedCase
Function OwnableUpgradeable.__Ownable_init() (EdoMarketplaceManagement.sol#461-463) is not in mixedCase
Function OwnableUpgradeable.__Ownable_init_unchained() (EdoMarketplaceManagement.sol#465-467) is not in mixedCase
Variable OwnableUpgradeable.__gap (EdoMarketplaceManagement.sol#526) is not in mixedCase
Parameter EdoMarketplaceManagementAdmin.setDepositAddress(address)._depositAddress (EdoMarketplaceManagement.sol#544) is not in mixedCase
Parameter EdoMarketplaceManagementAdmin.setAgentAddress(address)._agentAddress (EdoMarketplaceManagement.sol#552) is not in mixedCase
Parameter EdoMarketplaceManagementAdmin.setEdoVerseFoundationAddress(address)._edoVerseFoundationAddress (EdoMarketplaceManagement.sol#557) is not in mixedCase
Parameter EdoMarketplaceManagementAdmin.setEdoVerseIncAddress(address)._edoVerseIncAddress (EdoMarketplaceManagement.sol#562) is not in mixedCase
Parameter EdoMarketplaceManagementAdmin.includeNFTContractAddress(address)._NFTContractAddress (EdoMarketplaceManagement.sol#568) is not in mixedCase
Parameter EdoMarketplaceManagementAdmin.excludeNFTContractAddress(address)._NFTContractAddress (EdoMarketplaceManagement.sol#576) is not in mixedCase
Parameter EdoMarketplaceManagementAdmin.isApprovedNFTContractAddress(address)._NFTContractAddress (EdoMarketplaceManagement.sol#580) is not in mixedCase

```

```

Parameter EdoMarketplaceManagementAdmin.setFeeRateDistribution(uint256,uint256,uint256)._edoVerseFoundationFeeRate (EdoMarketplaceManagement.sol#598) is not in mixedCase
Parameter EdoMarketplaceManagementAdmin.setFeeRateDistribution(uint256,uint256,uint256)._edoVerseIncFeeRate (EdoMarketplaceManagement.sol#599) is not in mixedCase
Parameter EdoMarketplaceManagement.setAgentRefundAddress(address)._agentRefundAddress (EdoMarketplaceManagement.sol#636) is not in mixedCase
Parameter EdoMarketplaceManagement.setServiceFeeRate(uint256)._serviceFeeRate (EdoMarketplaceManagement.sol#642) is not in mixedCase
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#conformance-to-solidity-naming-conventions
INFO:Detectors:
OwnableUpgradeable.__gap (EdoMarketplaceManagement.sol#526) is never used in EdoMarketplaceManagement (EdoMarketplaceManagement.sol#613-666)
EdoMarketplaceManagement.NON_AGENT_FEE (EdoMarketplaceManagement.sol#618) is never used in EdoMarketplaceManagement (EdoMarketplaceManagement.sol#613-666)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#unused-state-variables
INFO:Detectors:
renounceOwnership() should be declared external:
- OwnableUpgradeable.renounceOwnership() (EdoMarketplaceManagement.sol#498-500)
transferOwnership(address) should be declared external:
- OwnableUpgradeable.transferOwnership(address) (EdoMarketplaceManagement.sol#506-509)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#public-function-that-could-be-declared-external
INFO:Slither:EdoMarketplaceManagement.sol analyzed (8 contracts with 75 detectors), 63 result(s) found
INFO:Slither:Use https://crytic.io/ to get access to additional detectors and Github integration

```

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

Email: audit@EtherAuthority.io

Slither log >> EdoMarketplaceManagementAdmin.sol

```
INFO:Detectors:
EdoMarketplaceManagementAdmin.setDepositAddress(address)._depositAddress (EdoMarketplaceManagementAdmin.sol#256) lacks a zero-check on :
- depositAddress = _depositAddress (EdoMarketplaceManagementAdmin.sol#261)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#missing-zero-address-validation
INFO:Detectors:
AddressUpgradeable._revert(bytes,string) (EdoMarketplaceManagementAdmin.sol#89-98) uses assembly
- INLINE ASM (EdoMarketplaceManagementAdmin.sol#91-94)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#assembly-usage
```

```
INFO:Detectors:
Pragma version^0.8.4 (EdoMarketplaceManagementAdmin.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 AddressUpgradeable.sendValue(address,uint256) (EdoMarketplaceManagementAdmin.sol#10-15):
- (success) = recipient.call{value: amount}{} (EdoMarketplaceManagementAdmin.sol#13)
Low level call in AddressUpgradeable.functionCallWithValue(address,bytes,uint256,string) (EdoMarketplaceManagementAdmin.sol#37-46):
- (success,returndata) = target.call{value: value}(data) (EdoMarketplaceManagementAdmin.sol#44)
Low level call in AddressUpgradeable.functionStaticCall(address,bytes,string) (EdoMarketplaceManagementAdmin.sol#52-59):
- (success,returndata) = target.staticcall(data) (EdoMarketplaceManagementAdmin.sol#57)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#low-level-calls
```

```
INFO:Detectors:
Function ContextUpgradeable.__Context_init() (EdoMarketplaceManagementAdmin.sol#149-150) is not in mixedCase
Function ContextUpgradeable.__Context_init_unchained() (EdoMarketplaceManagementAdmin.sol#152-153) is not in mixedCase
Variable ContextUpgradeable.__gap (EdoMarketplaceManagementAdmin.sol#162) is not in mixedCase
Function OwnableUpgradeable.__Ownable_init() (EdoMarketplaceManagementAdmin.sol#173-175) is not in mixedCase
Function OwnableUpgradeable.__Ownable_init_unchained() (EdoMarketplaceManagementAdmin.sol#177-179) is not in mixedCase
Variable OwnableUpgradeable.__gap (EdoMarketplaceManagementAdmin.sol#238) is not in mixedCase
Parameter EdoMarketplaceManagementAdmin.setDepositAddress(address)._depositAddress (EdoMarketplaceManagementAdmin.sol#256) is not in mixedCase
Parameter EdoMarketplaceManagementAdmin.setAgentAddress(address)._agentAddress (EdoMarketplaceManagementAdmin.sol#264) is not in mixedCase
Parameter EdoMarketplaceManagementAdmin.setEdoVerseFoundationAddress(address)._edoVerseFoundationAddress (EdoMarketplaceManagementAdmin.sol#269) is not in mixedCase
Parameter EdoMarketplaceManagementAdmin.setEdoVerseIncAddress(address)._edoVerseIncAddress (EdoMarketplaceManagementAdmin.sol#274) is not in mixedCase
Parameter EdoMarketplaceManagementAdmin.includeNFTContractAddress(address)._NFTContractAddress (EdoMarketplaceManagementAdmin.sol#280) is not in mixedCase
Parameter EdoMarketplaceManagementAdmin.excludeNFTContractAddress(address)._NFTContractAddress (EdoMarketplaceManagementAdmin.sol#288) is not in mixedCase
Parameter EdoMarketplaceManagementAdmin.isApprovedNFTContractAddress(address)._NFTContractAddress (EdoMarketplaceManagementAdmin.sol#292) is not in mixedCase
Parameter EdoMarketplaceManagementAdmin.includePaymentContractAddress(address)._edoPaymentContractAddress (EdoMarketplaceManagementAdmin.sol#296) is not in mixedCase
Parameter EdoMarketplaceManagementAdmin.excludePaymentContractAddress(address)._edoPaymentContractAddress (EdoMarketplaceManagementAdmin.sol#300) is not in mixedCase
Parameter EdoMarketplaceManagementAdmin.isApprovedPaymentContractAddress(address)._edoPaymentContractAddress (EdoMarketplaceManagementAdmin.sol#304) is not in mixedCase
Parameter EdoMarketplaceManagementAdmin.setFeeRateDistribution(uint256,uint256,uint256)._agentRefundFeeRate (EdoMarketplaceManagementAdmin.sol#309) is not in mixedCase
Parameter EdoMarketplaceManagementAdmin.setFeeRateDistribution(uint256,uint256,uint256)._edoVerseFoundationFeeRate (EdoMarketplaceManagementAdmin.sol#310) is not in mixedCase
Parameter EdoMarketplaceManagementAdmin.setFeeRateDistribution(uint256,uint256,uint256)._edoVerseIncFeeRate (EdoMarketplaceManagementAdmin.sol#311) is not in mixedCase
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#conformance-to-solidity-naming-conventions
```

```
INFO:Detectors:
OwnableUpgradeable.__gap (EdoMarketplaceManagementAdmin.sol#238) is never used in EdoMarketplaceManagementAdmin (EdoMarketplaceManagementAdmin.sol#242-323)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#unused-state-variables
INFO:Detectors:
renounceOwnership() should be declared external:
- OwnableUpgradeable.renounceOwnership() (EdoMarketplaceManagementAdmin.sol#210-212)
transferOwnership(address) should be declared external:
- OwnableUpgradeable.transferOwnership(address) (EdoMarketplaceManagementAdmin.sol#218-221)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#public-function-that-could-be-declared-external
INFO:Slither:EdoMarketplaceManagementAdmin.sol analyzed (5 contracts with 75 detectors), 47 result(s) found
INFO:Slither:Use https://crytic.io/ to get access to additional detectors and Github integration
```

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

Email: audit@EtherAuthority.io

Solidity Static Analysis

EdoMarketplace.sol

Security

Check-effects-interaction:

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

[more](#)

Pos: 84:4:

Check-effects-interaction:

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

[more](#)

Pos: 109:4:

Check-effects-interaction:

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

[more](#)

Pos: 137:4:

Gas & Economy

Gas costs:

Gas requirement of function EdoMarketplace.depositFeeAmount 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: 47:4:

Gas costs:

Gas requirement of function EdoMarketplaceManagement.withdraw 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:

Gas costs:

Gas requirement of function `EdoMarketplaceManagementAdmin.setFeeRateDistribution` 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: 74:4:

Miscellaneous

Constant/View/Pure functions:

`EdoMarketplace.getListingEdoNfts()` : Is constant but potentially should not be. Note: Modifiers are currently not considered by this static analysis.

[more](#)

Pos: 171:4:

Constant/View/Pure functions:

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

[more](#)

Pos: 193:4:

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: 41:8:

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: 83:8:

EdoMarketplaceManagement.sol

Gas & Economy

Gas costs:

Gas requirement of function `EdoMarketplaceManagement.depositFeeAmount` 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: 47:4:

Gas costs:

Gas requirement of function `EdoMarketplaceManagement.setFeeRateDistribution` 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: 74:4:

Gas costs:

Gas requirement of function `EdoMarketplaceManagementAdmin.setFeeRateDistribution` 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: 74:4:

Miscellaneous

Guard conditions:

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

[more](#)

Pos: 41:8:

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: 83:8:

EdoMarketplaceManagementAdmin.sol

Gas & Economy

Gas costs:

Gas requirement of function `EdoMarketplaceManagementAdmin.setFeeRateDistribution` 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: 74:4:

Miscellaneous

Guard conditions:

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

[more](#)

Pos: 31:8:

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: 36:8:

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: 41:8:

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: 83:8:

Solhint Linter

EdoMarketplace.sol

```
EdoMarketplace.sol:2:1: Error: Compiler version ^0.8.15 does not satisfy the r semver requirement
```

EdoMarketplaceManagement.sol

```
EdoMarketplaceManagement.sol:2:1: Error: Compiler version ^0.8.15 does not satisfy the r semver requirement
EdoMarketplaceManagement.sol:12:5: Error: Explicitly mark visibility of state
EdoMarketplaceManagement.sol:13:5: Error: Explicitly mark visibility of state
EdoMarketplaceManagement.sol:14:5: Error: Explicitly mark visibility of state
EdoMarketplaceManagement.sol:15:5: Error: Explicitly mark visibility of state
```

EdoMarketplaceManagementAdmin.sol

```
EdoMarketplaceManagementAdmin.sol:2:1: Error: Compiler version ^0.8.15 does not satisfy the r semver requirement
EdoMarketplaceManagementAdmin.sol:46:9: Error: Variable name must be in mixedCase
EdoMarketplaceManagementAdmin.sol:54:40: Error: Variable name must be in mixedCase
EdoMarketplaceManagementAdmin.sol:58:43: Error: Variable name must be in mixedCase
```

Software analysis result:

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



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

Email: audit@EtherAuthority.io