INSPECTOR
LOVELY

SMART CONTRACT

SECURITY AUDIT

CYBERCONNECT TOKEN

inspector.lovely.finance

INSPECTOR LOVELY

TABLE OF CONTENTS

INSPECTOR
LOVELY

DISCLAIMER

This is a comprehensive report based on our automated and manual examination of cybersecurity vulnerabilities and framework flaws of the project's smart contract. Reading the full analysis report is essential to build your understanding of project's security level. It is crucial to take note, though we have done our best to perform this analysis and report, that you should not rely on the our research and cannot claim what it states or how we created it. Before making any judgments, you have to conduct your own independent research. We will discuss this in more depth in the following disclaimer - please read it fully. DISCLAIMER: You agree to the terms of this disclaimer by reading this report or any portion thereof. Please stop reading this report and remove and delete any copies of this report that you download and/or print if you do not agree to these conditions. Scan and verify report's presence in the GitHub repository by a qr-code on the title page. This report is for non-reliability information only and does not represent investment advice. No one shall be entitled to depend on the report or its contents, and Inspector Lovely and its affiliates shall not be held responsible to you or anyone else, nor shall Inspector Lovely provide any guarantee or representation to any person with regard to the accuracy or integrity of the report. Without any terms, warranties or other conditions other than as set forth in that exclusion and Inspector Lovely excludes hereby all representations, warrants, conditions and other terms (including, without limitation, guarantees implied by the law of satisfactory quality, fitness for purposes and the use of reasonable care and skills). The report is provided as "as is" and does not contain any terms and conditions. Except as legally banned, Inspector Lovely disclaims all responsibility and responsibilities and no claim against Inspector Lovely is made to any amount or type of loss or damages (without limitation, direct, indirect, special, punitive, consequential or pure economic loses or losses) that may be caused by you or any other person, or any damages or damages, including without limitations (whether innocent or negligent). Security analysis is based only on the smart contracts. No applications or operations were reviewed for security. No product code has been reviewed.

INSPECTOR LOVELY

AUDIT SCOPE

Name	Code Review and Security Analysis Report for Cyberconnect Token Coin Smart Contract
Platform	Ethereum
Language	Solidity
File1	ARKM.sol
File 2	EIP712.so
File 3	CyberNFTBase.sol
File 4	CyberEngine.sol
File 5	EssenceNFT.sol
File 6	ProfileNFT.sol
File 7	SubscribeNFT.sol
File 8	Auth.sol
File 9	ERC721.sol
File 10	Owned.sol
File 11	RolesAuthority.sol
File 12	Create2Deployer.sol
File 13	EssenceDeployer.sol
File 14	ProfileDeployer.sol
File 15	SubscribeDeployer.sol
File 16	FeeMw.sol
File 17	PermissionedMw.sol
File 18	Treasury.sol
File 19	CollectDisallowedMw.sol
File 20	CollectFlexPaidMw.sol
File 21	CollectLimitedTimePaidMw.sol
File 22	CollectMerkleDropMw.sol
File 23	CollectOnlySubscribedMw.sol
File 24	CollectPaidMw.sol
File 25	CollectPermissionMw.sol
File 26	CollectPermissionPaidMw.sol
File 27	FeeCreationMw.sol

INSPECTOR LOVELY

File 28	PermissionedFeeCreationMw.sol
File 29	StableFeeCreationMw.sol
File 30	SubscribeDisallowedMw.sol
File 31	SubscribeOnlyOnceMw.sol
File 32	SubscribePaidMw.sol
File 33	CyberBoxNFT.sol
File 34	CyberGrandNFT.sol
File 35	CyberVault.sol
File 36	FrameNFT.sol
File 37	Link3ProfileDescriptor.sol
File 38	Link3ProfileDescriptorV2.sol
File 39	Link3ProfileDescriptorV3.sol
File 40	MBNFT.sol
File 41	MiniShardNFT.sol
File 42	RelationshipChecker.sol
File 43	CyberBoxNFTStorage.sol
File 44	CyberEngineStorage.sol
File 45	CyberGrandNFTStorage.sol
File 46	EssenceNFTStorage.sol
File 47	Link3ProfileDescriptorStorage.sol
File 48	MBNFTStorage.sol
File 49	ProfileNFTStorage.sol
File 50	SubscribeNFTStorage.sol
File 51	UpgradeableBeacon.sol
File 52	CYBER.sol
File 53	Pausable.sol
File 54	ReentrancyGuard.sol
File 55	Link3ProfileDescriptorStorageV2.sol
Github commit Hash	e567208187d47e00c33fa0013b11b1ae769027e7
Audit Date	November 11th, 2023

PROPOSED SMART CONTRACT FEATURES

Claimed Feature Detail	Our Observation
File 1: CyberNFTBaseFlex.sol • This contract is the base for all NFT contracts.	Validated
File 2: CyberNFTBase.sol • This contract is the base for all NFT contracts.	Validated
File 3: CyberEngine.sol • This is the main entry point for the CyberConnect contract.	Validated
File 4: EssenceNFT.sol • This contract is used to create an Essence NFT.	Validated
File 5: ProfileNFT.sol • This contract is used to create a profile NFT.	Validated
File 6: SubscribeNFT.sol • This contract is used to create a Subscribe NFT. • This will be deployed as beacon contracts for gas efficiency.	Validated
File 7: Pausable.sol • The Contract module enables children to establish an emergency stop mechanism that can be activated by an authorized account.	Validated
File 8: ReentrancyGuard.sol • This module is designed to prevent reentrant calls to a function.	Validated
File 9: Auth.sol • The code has been adapted from Solmate's Auth.sol, with the initializer being replaced by the constructor.	Validated

Claimed Feature Detail	Our Observation
File 10: ERC721.sol • The code has been adapted from Solmate's ERC721.sol, with the initializer being replaced by the constructor.	Validated
File 11: Owned.sol • The code has been adapted from Solmate's Owned.sol file, with the initializer being replaced by the constructor.	Validated
File 12: RolesAuthority.sol • The authority is role-based and can support up to 256 roles.	Validated
File 13: Treasury.sol • This contract is used for treasury.	Validated
File 14: CollectDisallowedMw.sol • This contract is a middleware to disallow any collection to the essence that uses it.	Validated
File 15: CollectFlexPaidMw.sol • This contract is a middleware to only allow users to collect when they pay a flex fee to the essence owner.	Validated
File 16: CollectLimitedTimePaidMw.sol • This contract is a middleware to only allow users to collect when they pay a certain fee to the essence owner.	Validated
File 17: CollectMerkleDropMw.sol • This contract is a middleware to only allow users to collect an essence given the correct merkle proof.	Validated
File 18: CollectOnlySubscribedMw.sol • This contract is a middleware to allow the address to collect an essence only if they are subscribed.	Validated

Claimed Feature Detail	Our Observation
File 20: CollectPermissionMw.sol • This contract is a middleware to allow an address to collect an essence only if they have a valid signature from the essence owner.	Validated
File 21: CollectPermissionPaidMw.sol • This contract is a middleware to allow an address to collect an essence only if they have a valid signature from the signer and pay certain fees.	Validated
File 22: FeeCreationMw.sol • This contract is a middleware to create profiles with a fee.	Validated
File 23: PermissionedFeeCreationMw.sol • This contract is a middleware to create permissioned fees.	Validated
File 24: StableFeeCreationMw.sol • This contract is a middleware to charge a stable fee (USD) when creating a profile.	Validated
File 25: SubscribeDisallowedMw.sol • This contract is a middleware to disallow any subscriptions to the user.	Validated
File 26: SubscribeOnlyOnceMw.sol • This contract is a middleware to allow the address to subscribe only once to another address.	Validated
File 26: SubscribePaidMw.sol • This contract is a middleware to only allow users to subscribe when they pay a certain fee to the profile owner.	Validated
File 27: CyberBoxNFT.sol • This contract is used to create CyberBox NFT.	Validated

Claimed Feature Detail	Our Observation
File 28: CyberGrandNFT.sol • This contract is used to create CyberGrand NFT.	Validated
File 29: CyberVault.sol • This contract is used to create CyberVault.	Validated
File 30: Link3ProfileDescriptor.sol • This contract is used to create profile NFT token uri.	Validated
File 31: Link3ProfileDescriptorV2.sol • This contract is used to create profile NFT token uri.	Validated
File 32: Link3ProfileDescriptorV3.sol • This contract is used to create profile NFT token uri.	Validated
File 33: MBNFT.sol • This contract is used to create MB NFT.	Validated
File 34: MiniShardNFT.sol • The contract-deploying account will be granted minter and pauser roles, along with the default admin role, enabling it to grant these roles to other accounts.	Validated
File 35: CYBER.sol • Name: CyberConnect • Symbol: CYBER • Decimals: 18 • Total supply of cyber token is 100 Million. **Ownership control:** • The new TokenPool address can be set by the owner. • Current owner can transfer the ownership. • Owner can renounce ownership.	Validated

INSPECTOR LOVELY

AUDIT SUMMARY

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

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

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

We found 0 critical, 0 high, 0 medium and 0 low and 2 very low level issues.

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

INSPECTOR LOVELY

KEY TECHNICAL METRICS

MAIN CATEGORY	SUBCATEGORY	RESULT
Contract Programming	Solidity version is not specified	Passed
	Solidity version is too old	Passed
	Integer overflow/underflow	Passed
	Function input parameters lack 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	Moderated
	High consumption 'for/while' loop	Passed
	High consumption 'storage' storage	Passed
	Assert() misuse	Passed
Business Risk	The maximum limit for mintage is not set	Passed
	"Short Address" Attack	Passed
	"Double Spend" Attack	Passed

Overall Audit Result: PASSED

CODE QUALITY

This audit scope has 1 smart contract. Smart contract contains Libraries, Smart contracts, inherits and Interfaces. This is a compact and well written smart contract.

The libraries in CyberConnect 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 CyberConnect.

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

Code parts are well commented on in the smart contracts. Ethereum's NatSpec commenting style is recommended.

DOCUMENTATION

We were given a CyberConnect smart contract code in the form of a bscscan web link.

As mentioned above, code parts are well commented on. and the logic is straightforward. So it is easy to quickly understand the programming flow as well as complex code logic. Comments are very helpful in understanding the overall architecture of the protocol.

Another source of information was its official website: https://cyberconnect.me which provided rich information about the project architecture and tokenomics.

USE OF DEPENDENCIES

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

Apart from libraries, its functions are not used in external smart contract calls.

INSPECTOR LOVELY

PROJECT WEBSITE PERFORMANCE AUDIT

Performance Metrics

Browser Timings

Redirect Duration 0ms	Connection Duration 80ms	Backend Duration 29ms
Time to First Byte 109ms	First Paint 232s	DOM Interactive Time 232
DOM Content Loaded 254s	Onload Time 877s	Fully Loaded Time 1.4s

Grade

A	Performance 87%	Structure 97%

Web Vitals

LCP	TBT	CLS
316ms	462ms	0

Top Issues

IMPECT	AUDIT	
LOW	Allow back/forward cache restoration	1 failure reason
LOW	Avoid an excessive DOM size (TBT)	780 elements
LOW	Reduce JavaScript execution time (TBT)	536ms spent executing JavaScript
LOW	Avoid enormous network payloads LCP (LCP)	Total size was 1.36MB
LOW	Avoid long main-thread tasks (TBT)	2 long tasks found

Level of Criticality

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
Med	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.

INSPECTOR LOVELY

AUDIT FINDINGS TABLE

	Total	Resolved	UnResolved	Acknowledged
High Severity Issues Found	0	0	0	0
Moderate Severity Issues Found	0	0	0	0
Medium Severity Issues	0	0	0	0
Low Severity Issues	0	0	0	0
Informational Observations	2	0	2	0

The CyberConnect - Audit report identifies 2 issues with varying severity levels, discovered through manual review and static analysis techniques, alongside rigorous code reviews, highlighting the need for further investigation and vulnerability identification.

The smart contract is considered to **pass the audit**, as of the audit date, if no high severity or moderate severity issues are found.

INSPECTOR LOVELY

AUDIT FINDINGS

Critical Severity	No Critical severity vulnerabilities were found.
High Severity	No High severity vulnerabilities were found.
Medium	No Medium severity vulnerabilities were found.
Low	No Low severity vulnerabilities were found.
Very Low / Informational / Best practices:	1. Solidity Version: CYBER.sol `6 pragma solidity 0.8.14;` Use the latest solidity version. **Resolution:** Use the latest solidity version while contract deployment to prevent any compiler version level bugs. 2. Gas Optimization: The public functions that are never called by the contract could be declared external. **CYBER.sol** • mint **Ownable.sol** • renounceOwnership • transferOwnership **Resolution:** We suggest declaring these all functions external for better Gas optimization.

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:

CyberEngine.sol

- allowProfileMw: Profile middleware address can be allowed by the authorized owner.
- allowSubscribeMw: Subscribe middleware address can be allowed by the authorized owner.
- allowEssenceMw: Essence middleware address can be allowed by the authorized owner.
- createNamespace: Create a Namespace by the authorized owner.
- upgradeSubscribeNFT: Subscribe NFT can be updated by the authorized owner.
- upgradeEssenceNFT: Essence NFT can be updated by the authorized owner.
- upgradeProfileNFT: ProfileNFT can be updated by the authorized owner.
- setProfileMw: Profile middleware address can be updated by the authorized owner.

EssenceNFT.sol

- mint: Mint a new token by the profile owner.

ProfileNFT.sol

- pause: Pause a contract token by the namespace owner.
- setNamespaceOwner: A new Namespace owner address can be set by the namespace owner.
- setNFTDescriptor: NFT Descriptor address can be set by the namespace owner.
- setAvatar: Avatar Id can be set by the profile owner or operator.
- setOperatorApproval: Operator approval can be set by the profile owner.
- setMetadata: Metadata can be set by the profile owner or operator.
- setSubscribeData: Subscribe data can be set by the profile owner or operator.
- setEssenceData: Essence data can be set by the profile owner or operator.
- setPrimaryProfile: Primary profile can be set by the profile owner or operator.

SubscribeNFT.sol

- mint: Mint a new token by the profile owner.

INSPECTOR LOVELY

Auth.sol

- setOwner: A new owner address can be set by the current owner.

Owned.sol

- setOwner: A new owner address can be set by the current owner.

RolesAuthority.sol

- mint: Mint a new token by the owner.
- setPublicCapability: Public capability can be updated by the authorized owner.
- setRoleCapability: Role capability can be updated by the authorized owner.
- setUserRole: User role address can be updated by the authorized owner.

Treasury.sol

- setTreasuryAddress: Set the treasury address by the owner.
- setTreasuryFee: Set the treasury fee by the owner.
- allowCurrency: Allows a currency that will be used in a transaction by the owner.

CollectFlexPaidMw.sol

- setEssenceMwData: Essence middleware data can be set by the namespace owner.
- preProcess: Processes the transaction from the essence collector to the essence owner by the namespace owner.

CollectLimitedTimePaidMw.sol

- setEssenceMwData: Essence middleware data can be set by the namespace owner.
- preProcess: Processes the transaction from the essence collector to the essence owner by the namespace owner.

CollectPaidMw.sol

- setEssenceMwData: Essence middleware data can be set by the namespace owner.
- preProcess: Processes the transaction from the essence collector to the essence owner by the namespace owner.

CollectPermissionPaidMw.sol

- setEssenceMwData: Essence middleware data can be set by the namespace owner.
- preProcess: Processes the transaction from the essence collector to the essence owner by the namespace owner.

FeeCreationMw.sol

- preProcess: Processes the transaction from the essence collector to the essence owner by the namespace owner.
- setProfileMwData: Profile middleware data can be set by the engine owner.

StableFeeCreationMw.sol

- preProcess: Processes the transaction from the essence collector to the essence owner by the namespace owner.
- setProfileMwData: Profile middleware data can be set by the engine owner.

SubscribePaidMw.sol

- setSubscribeMwData: Subscribe middleware data can be set by the namespace owner.
- preProcess: Processes the transaction from the essence collector to the essence owner by the namespace owner.

CyberBoxNFT.sol

- pause: Changes the pause state of the box nft by the owner.
- setSigner: Set the new signer address by the owner.

CyberGrandNFT.sol

- pause: Changes the pause state of the box nft by the owner.
- setSigner: Set the new signer address by the owner.
- setTokenURI: Sets the new tokenURI by the owner.

CyberVault.sol

- setSigner: Set the new signer address by the owner.

Link3ProfileDescriptor.sol

- setAnimationTemplate: Animation template can be set by the owner.

MBNFT.sol

- pause: Changes the pause state of the box nft by the owner.
- setTokenURI: Set the new token URI by the owner.
- setBoxAddr: Set the new box and frame contract address by the owner.

CYBER.sol

- mint: Mint a new token by the owner.

Ownable.sol

- _checkOwner: Throws if the sender is not the owner.
- renounce Ownership: Deleting ownership will leave the contract without an owner, removing any owner-only functionality.
- transferOwnership: The current owner can transfer ownership of the contract to a new account.

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

CONCLUSION

We were given a contract code in the form of bscscan web links. And we have used all possible tests based on given objects as files. We had observed 2 very low issues in the smart contracts. but those are not critical.So, **it's good to go for the 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.

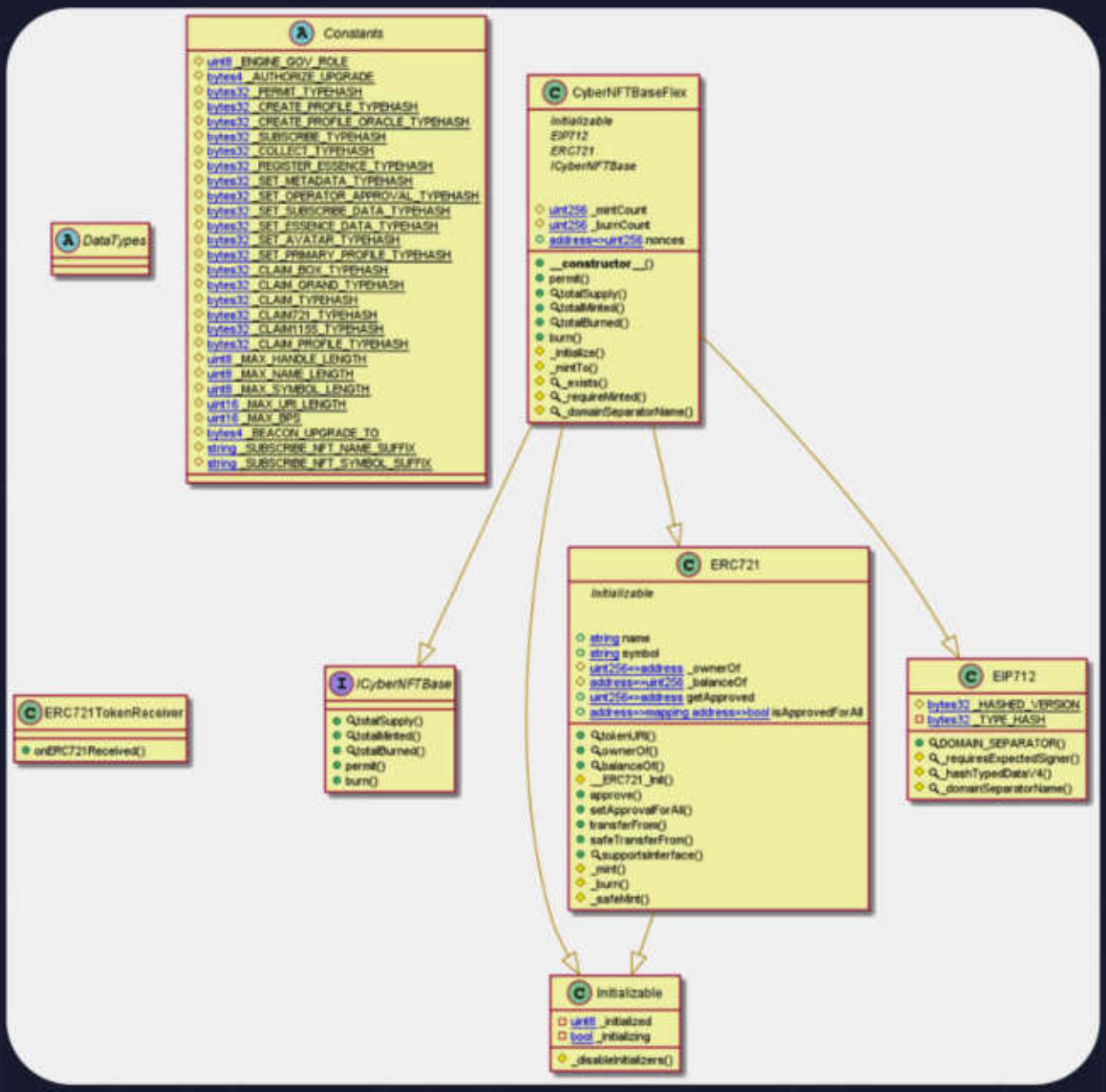
Security state of the reviewed smart contract, based on standard audit procedure scope, is **"Secured"**.

INSPECTOR LOVELY

ADDENDUM

Code Flow Diagram - CyberConnect

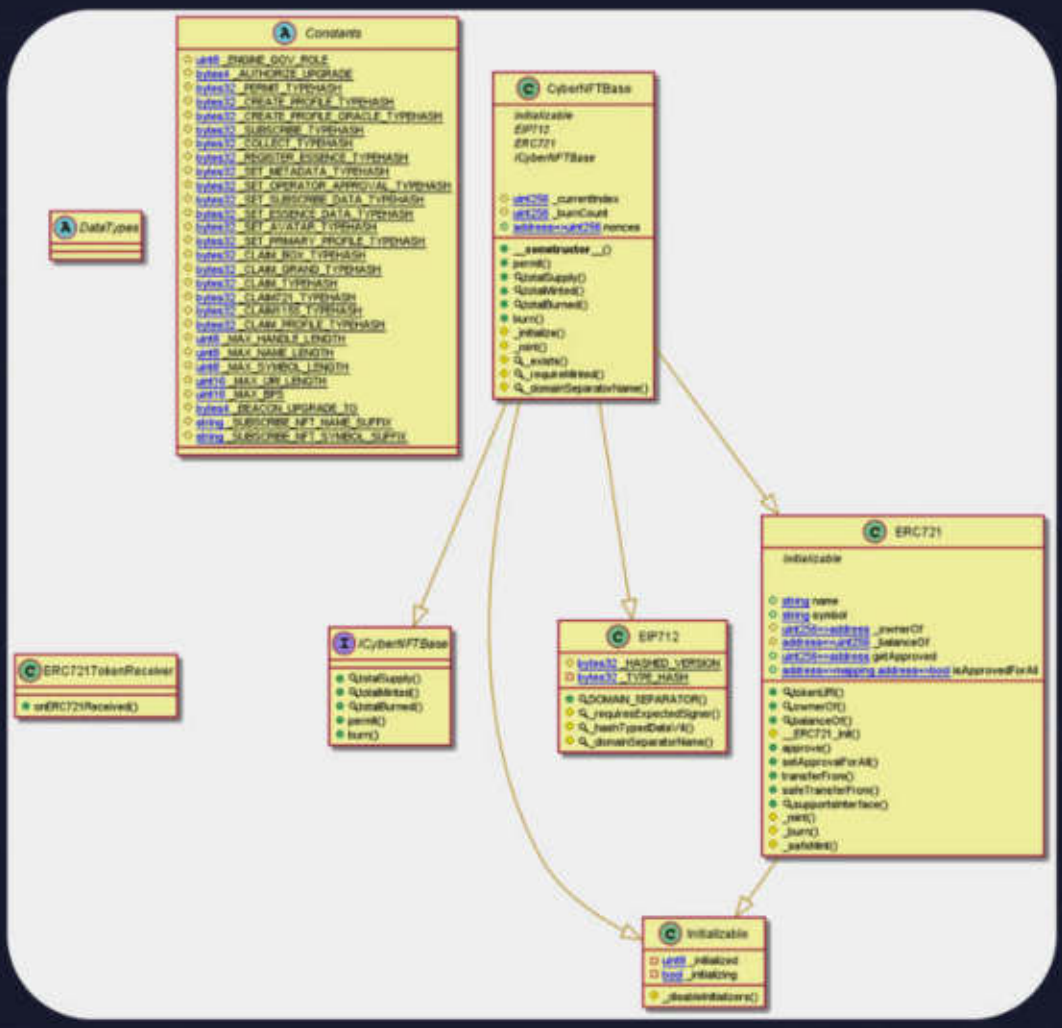
CyberNFTBaseFlex Diagram



INSPECTOR LOVELY

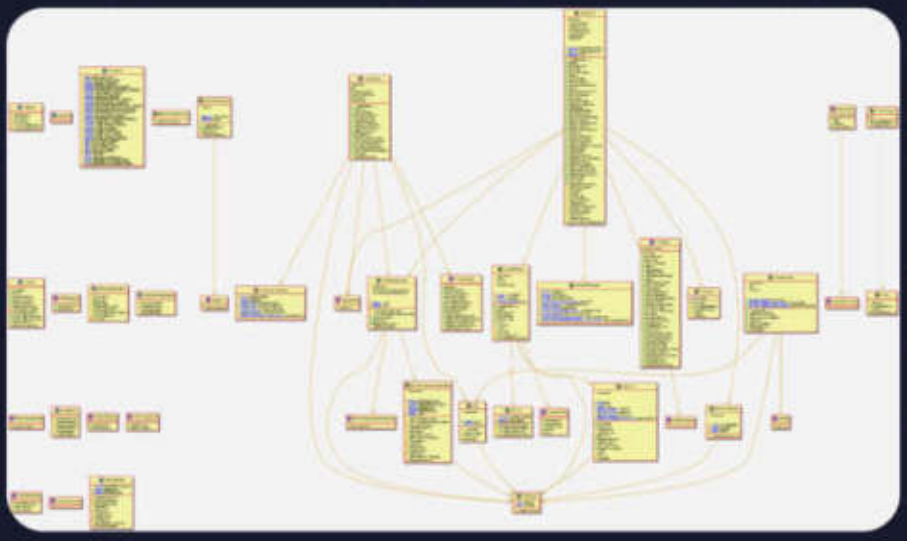
EIP712 Diagram

CyberNFTBase Diagram

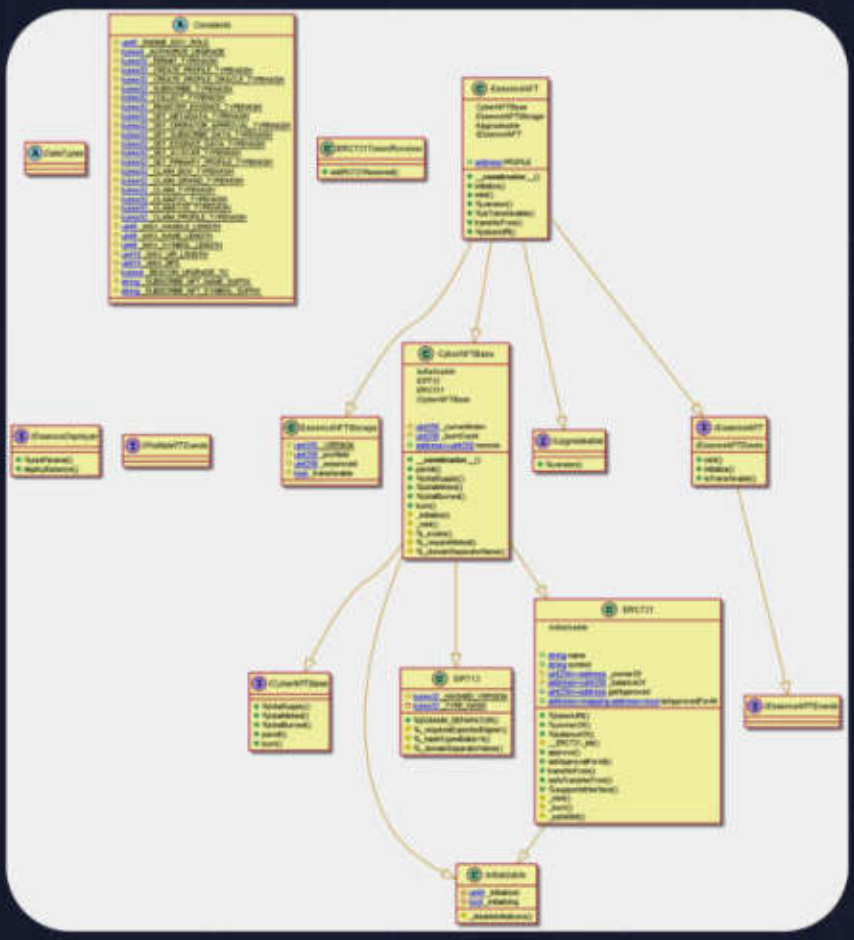


INSPECTOR LOVELY

CyberEngine Diagram

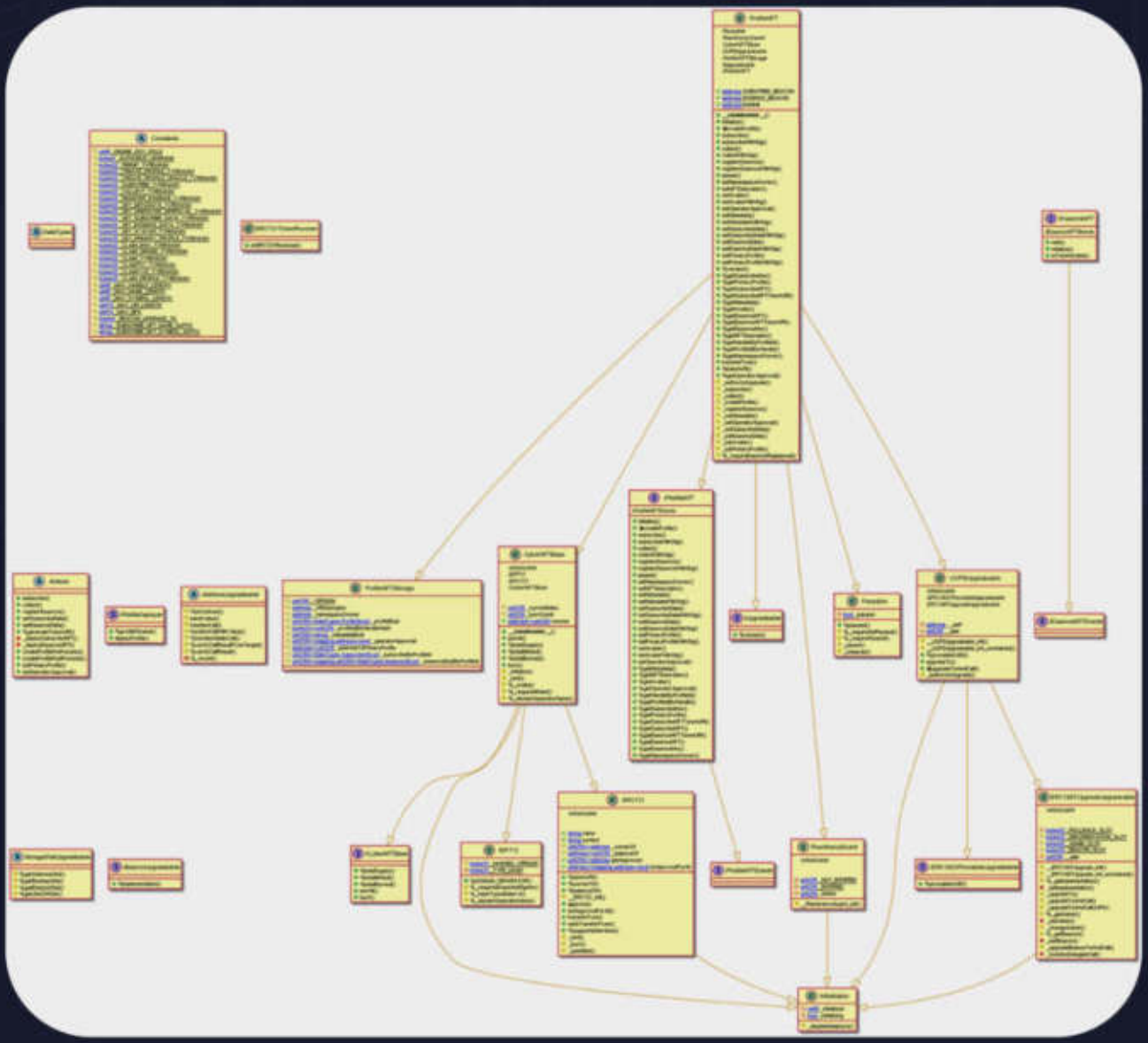


EssenceNFT Diagram



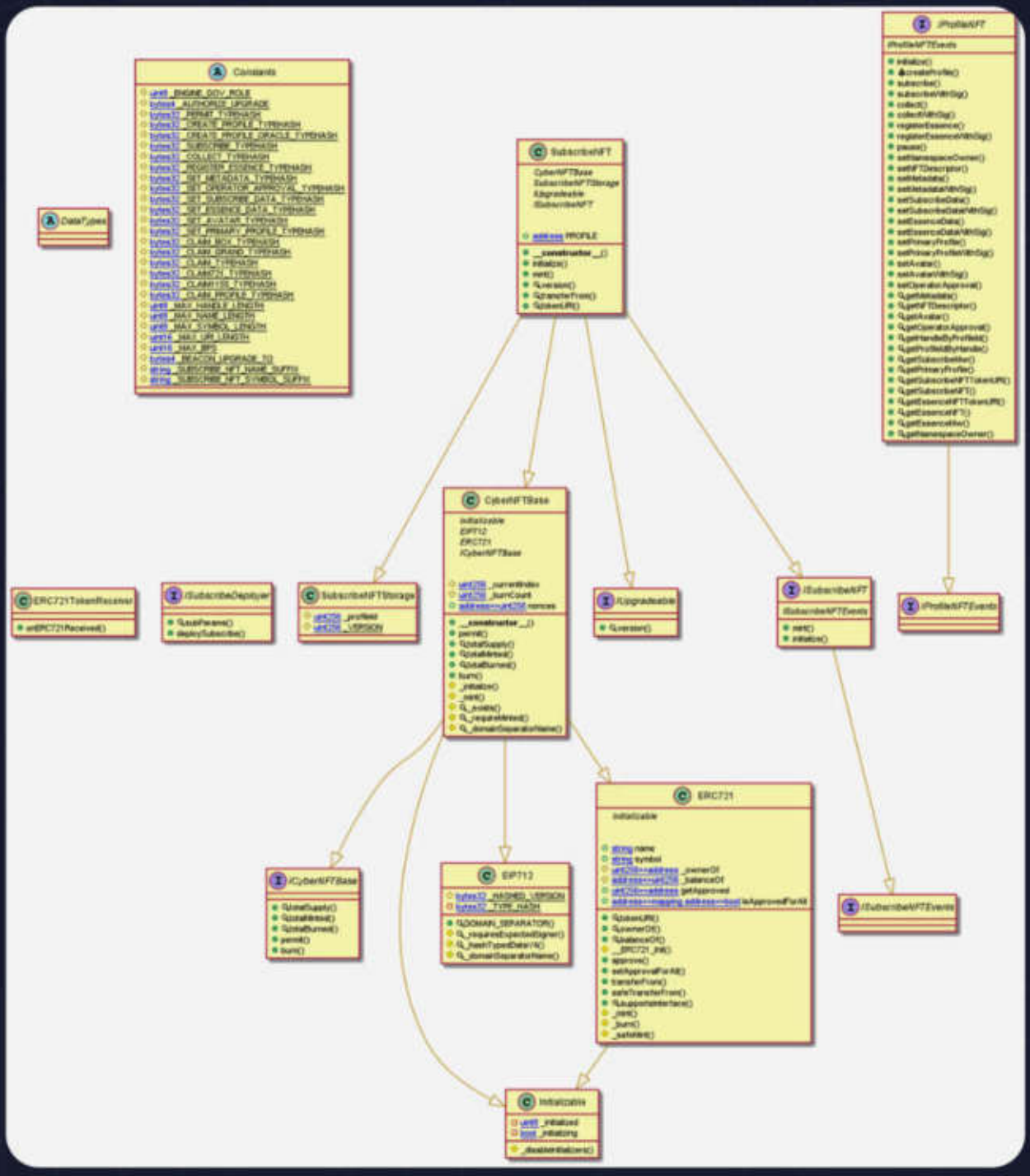
INSPECTOR LOVELY

ProfileNFT Diagram



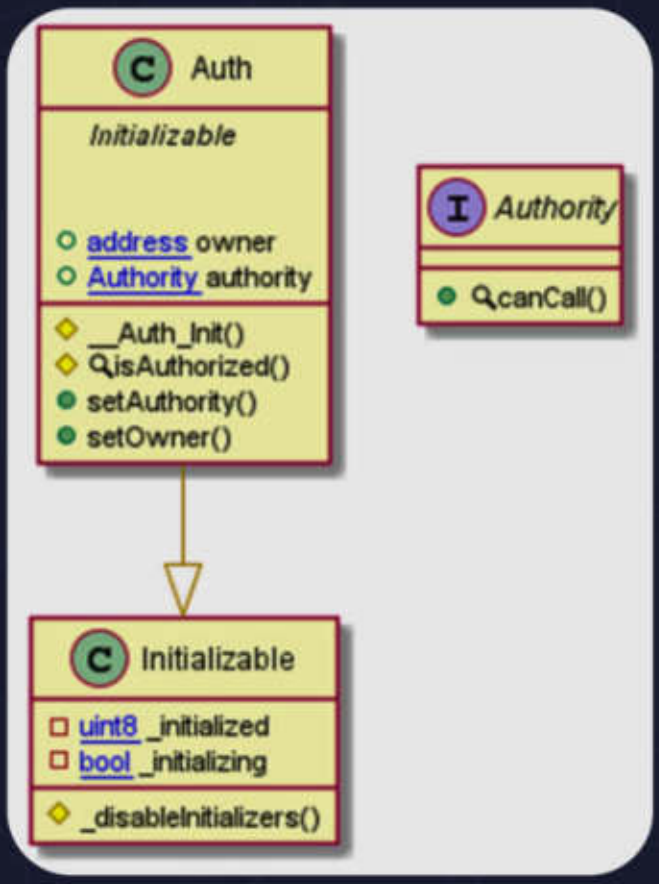
INSPECTOR LOVELY

SubscribeNFT Diagram



INSPECTOR LOVELY

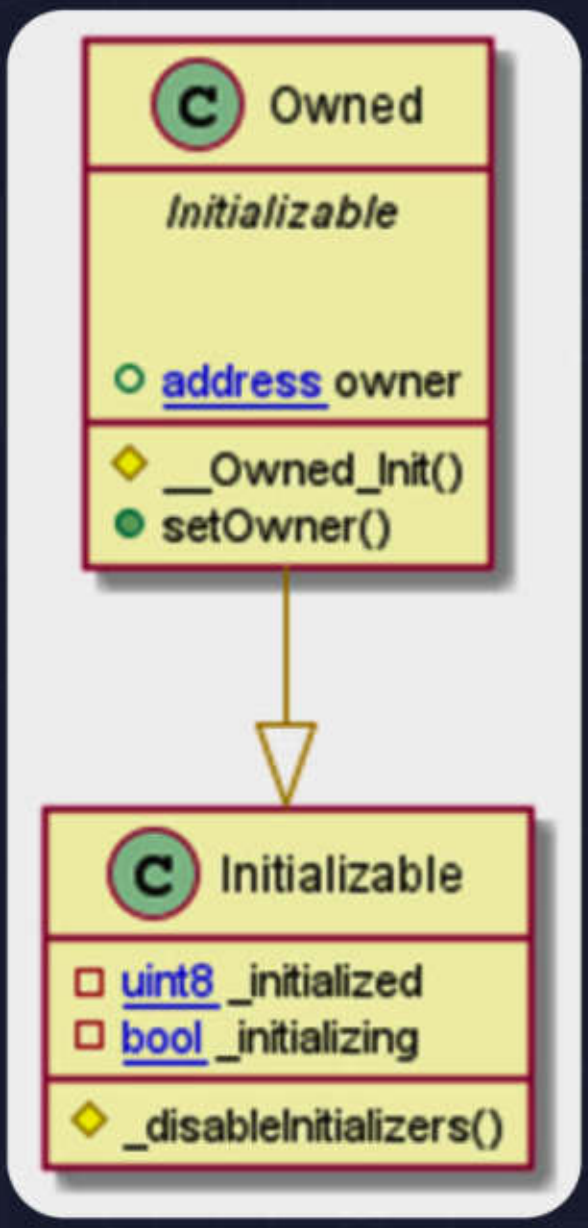
Auth Diagram



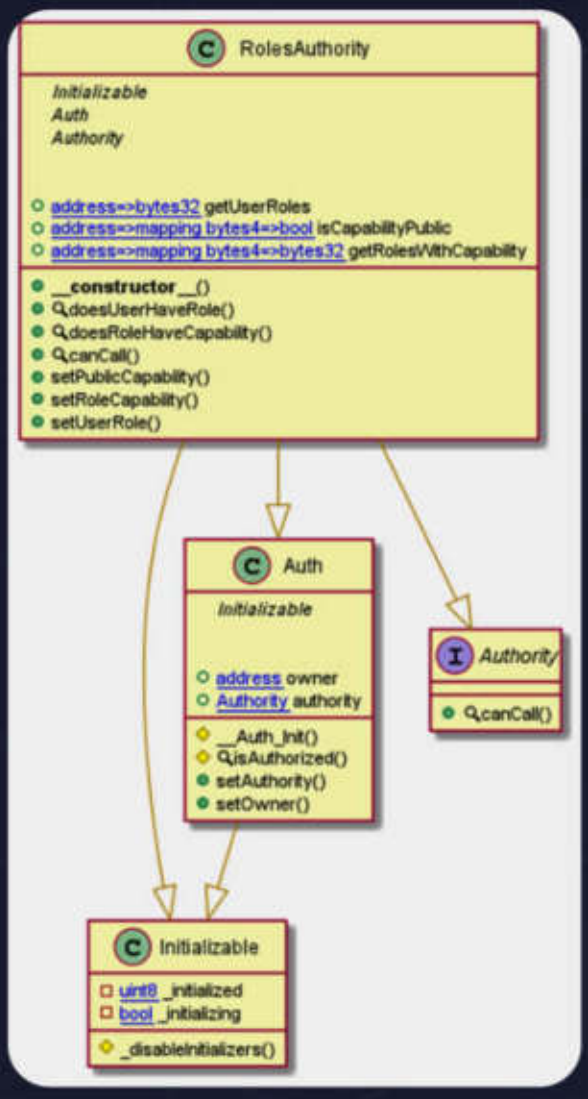
ERC721 Diagram

INSPECTOR LOVELY

Owned Diagram



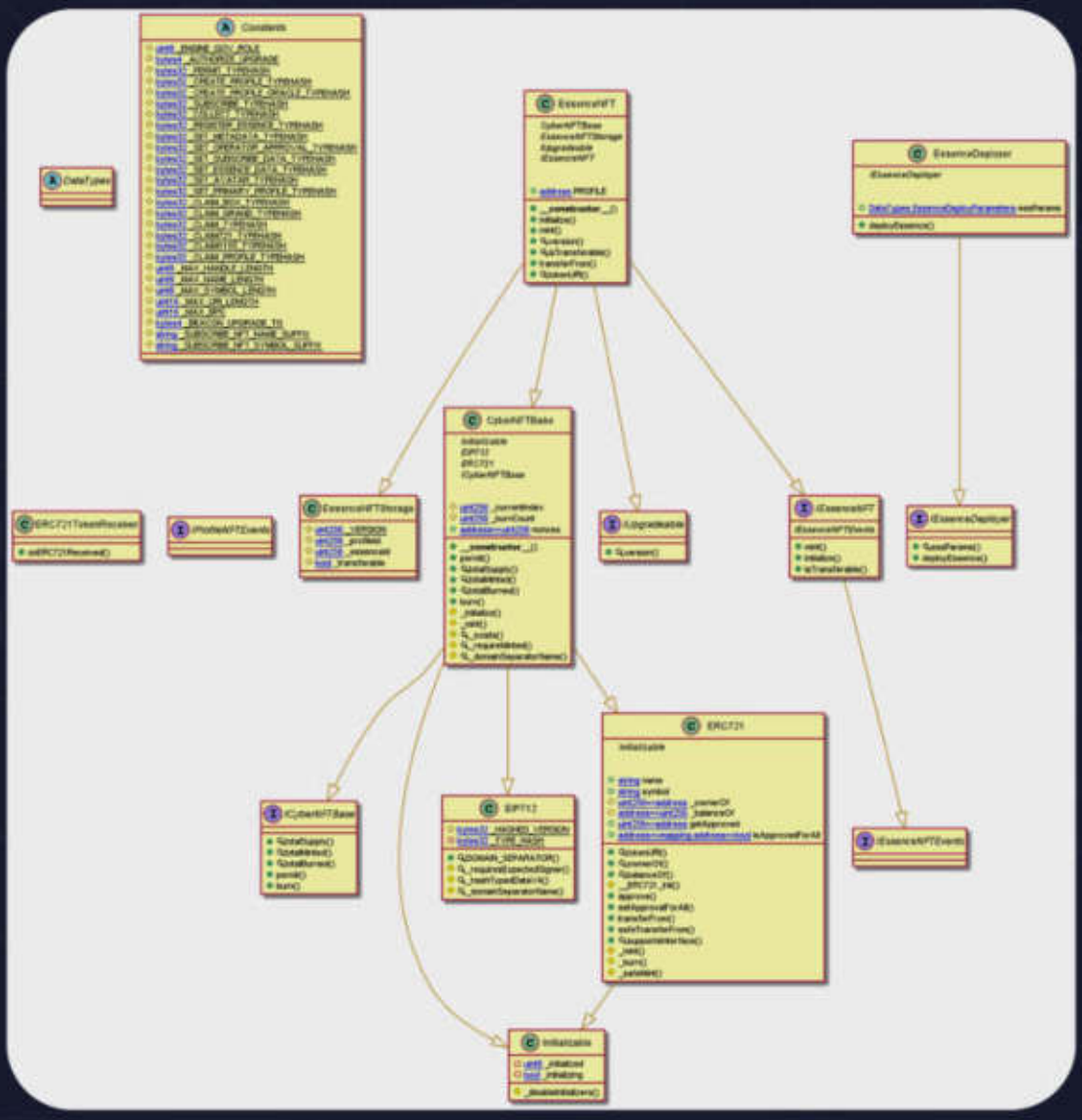
RolesAuthority Diagram



INSPECTOR LOVELY

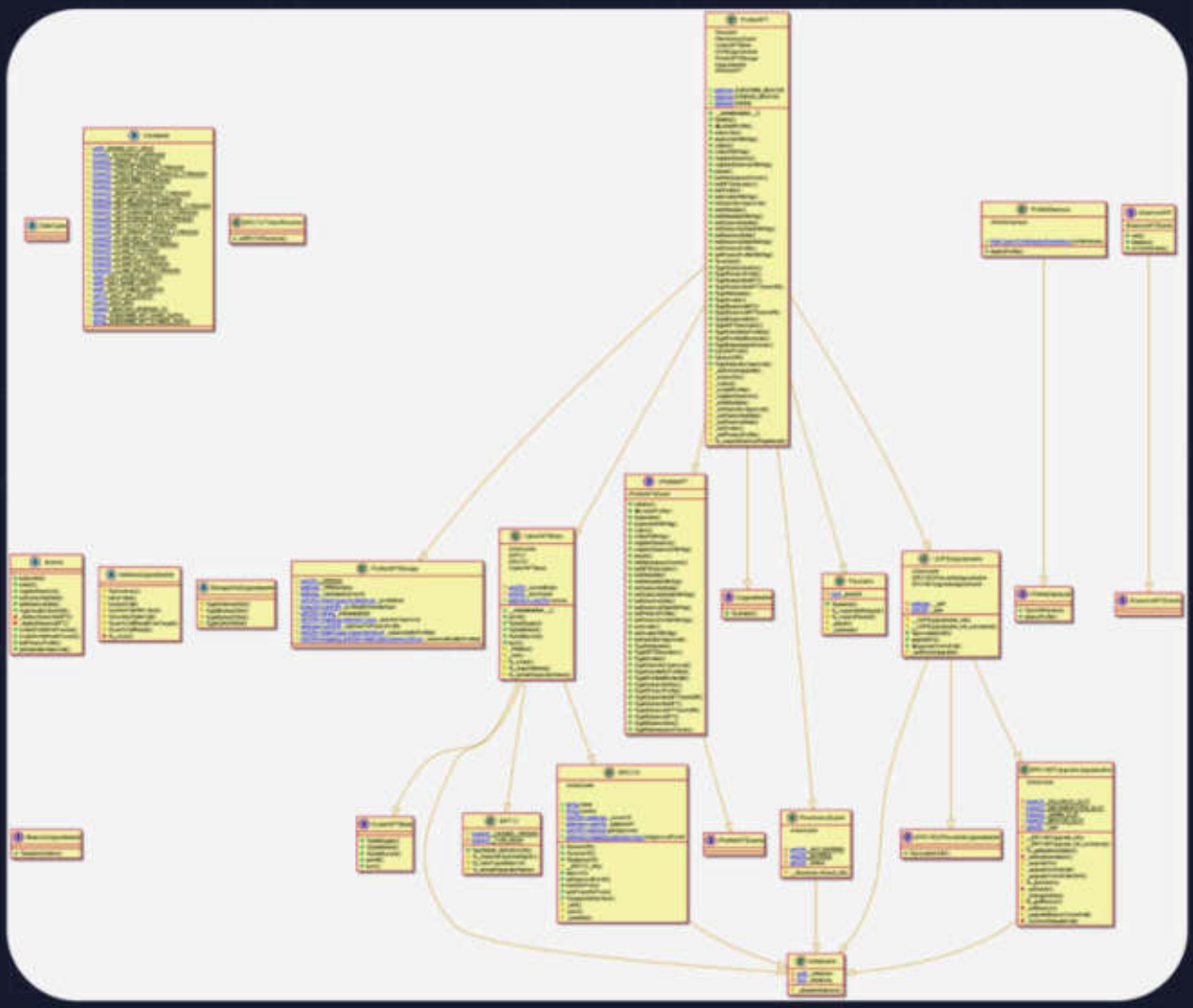
Create2Deployer Diagram

EssenceDeployer Diagram



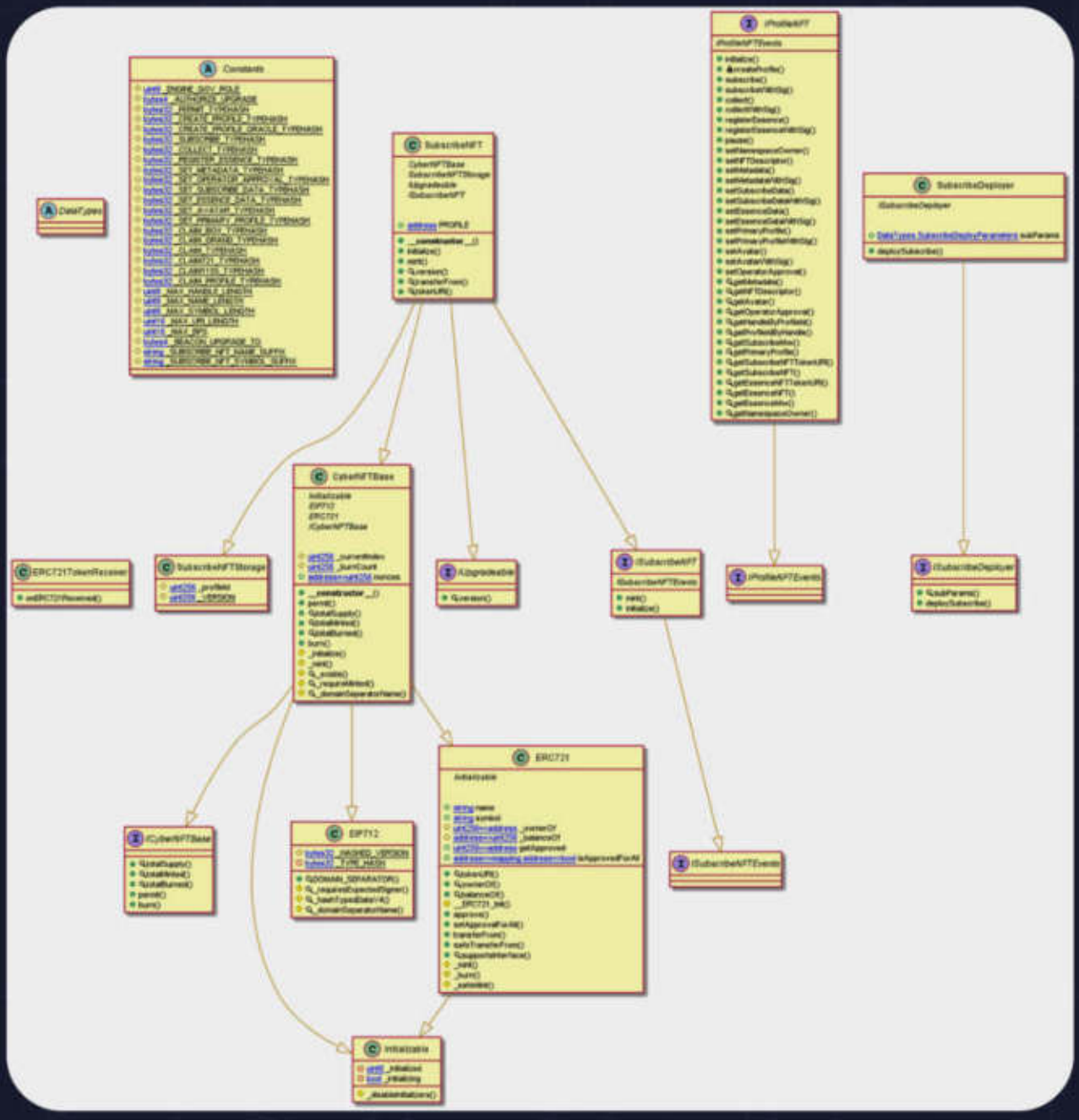
INSPECTOR LOVELY

ProfileDeployer Diagram



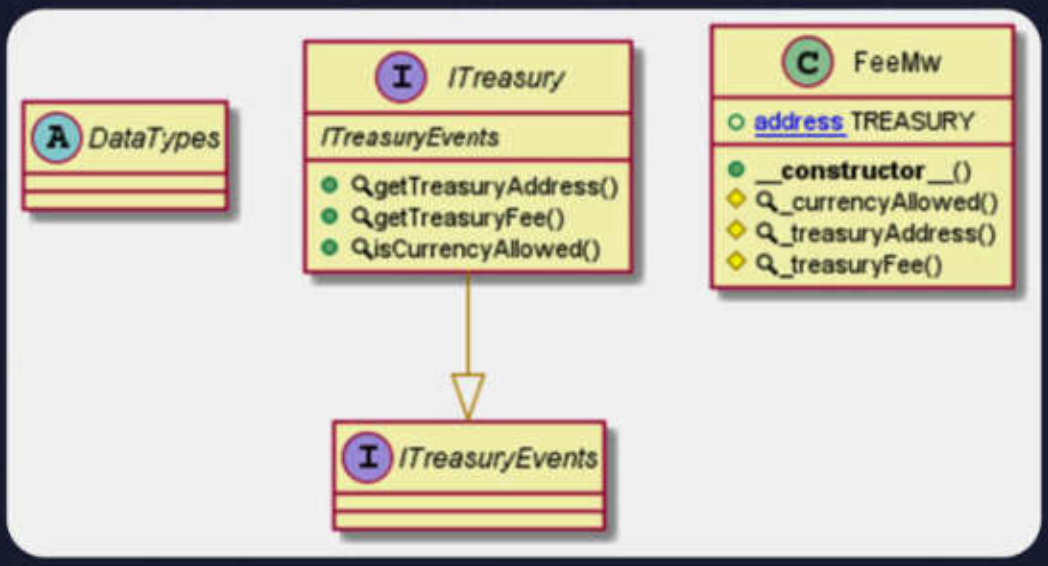
INSPECTOR LOVELY

SubscribeDeployer Diagram

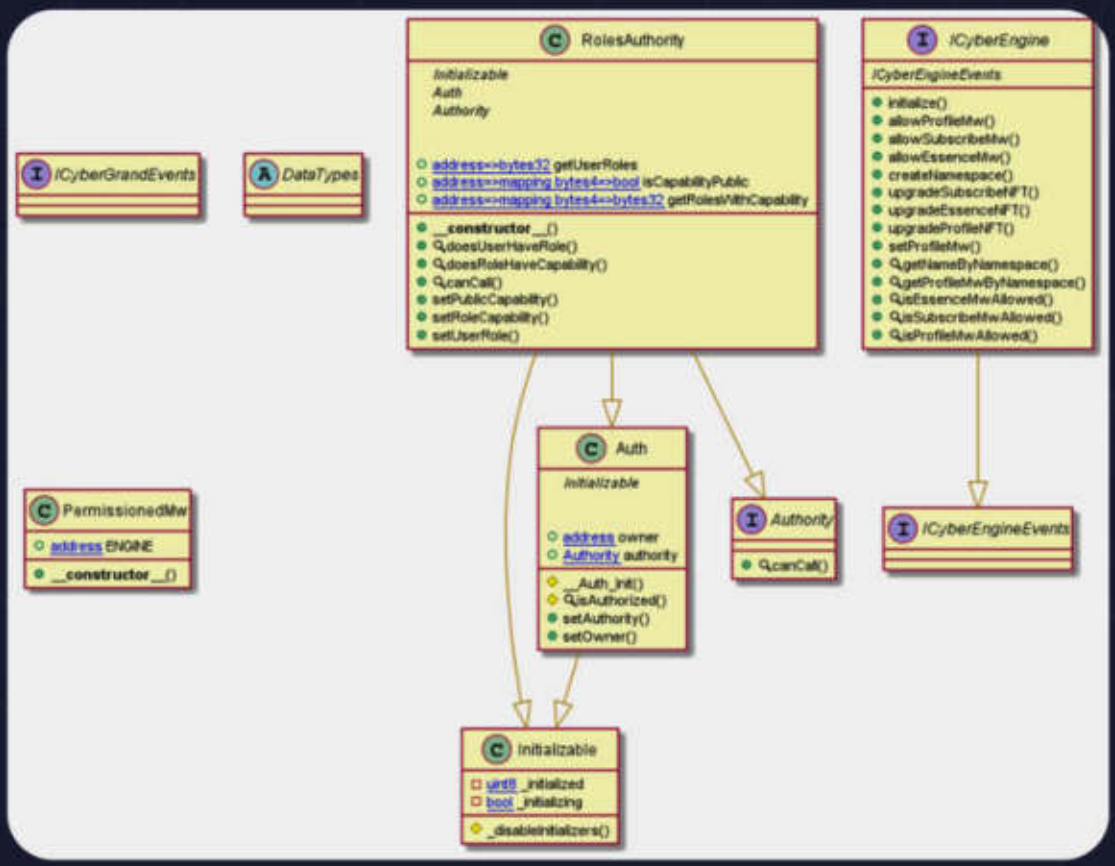


INSPECTOR LOVELY

FeeMw Diagram

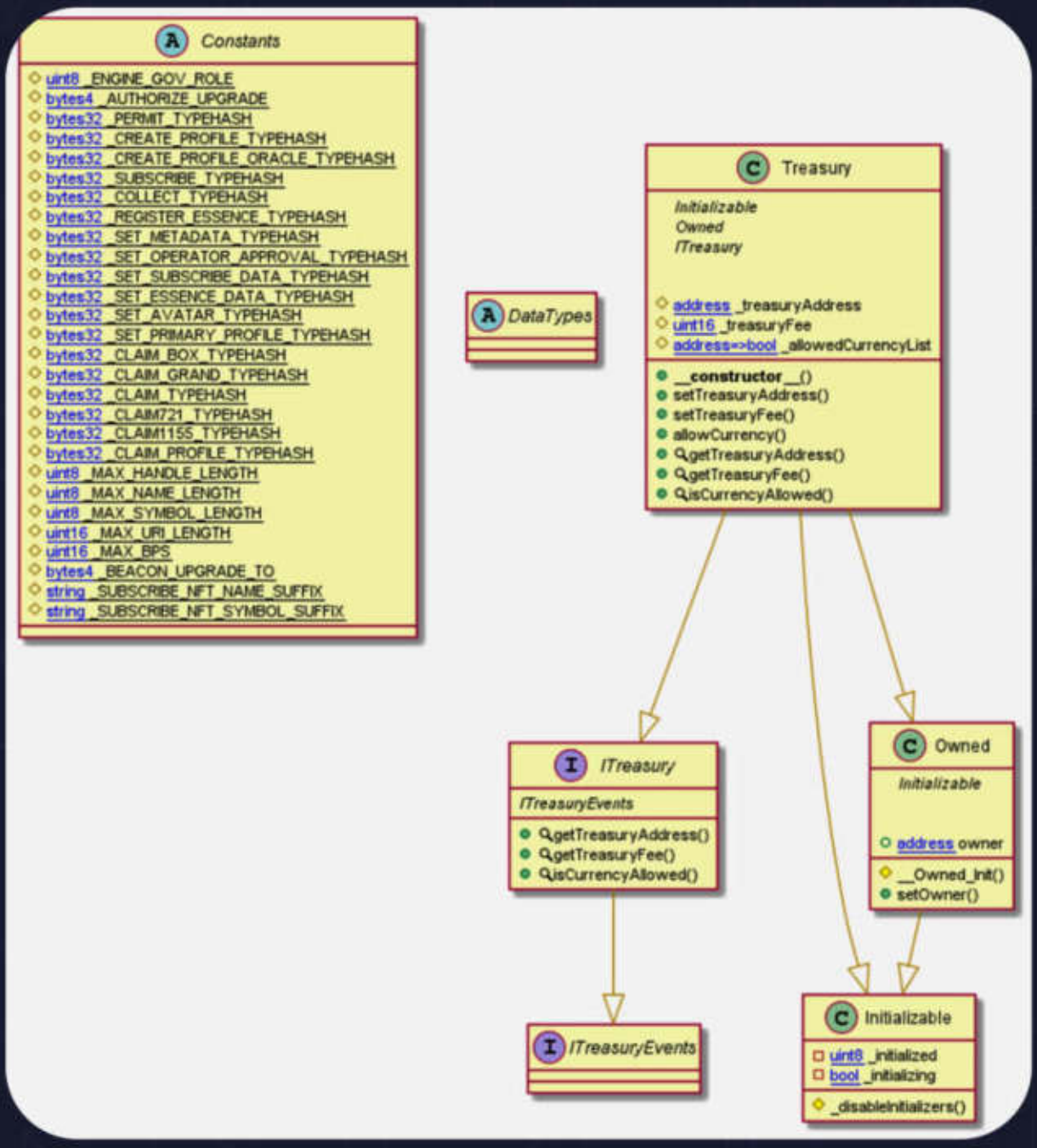


PermissionedMw Diagram



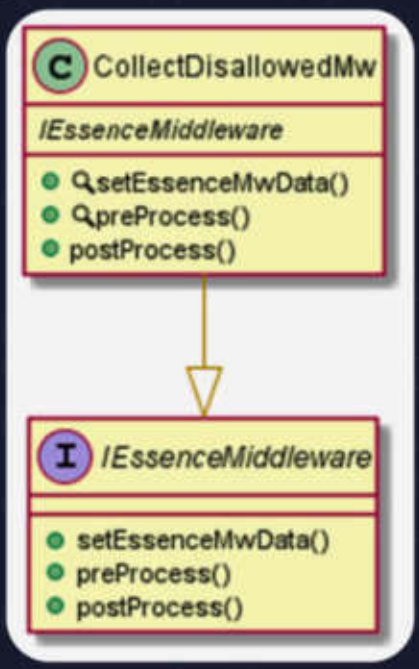
INSPECTOR LOVELY

Treasury Diagram

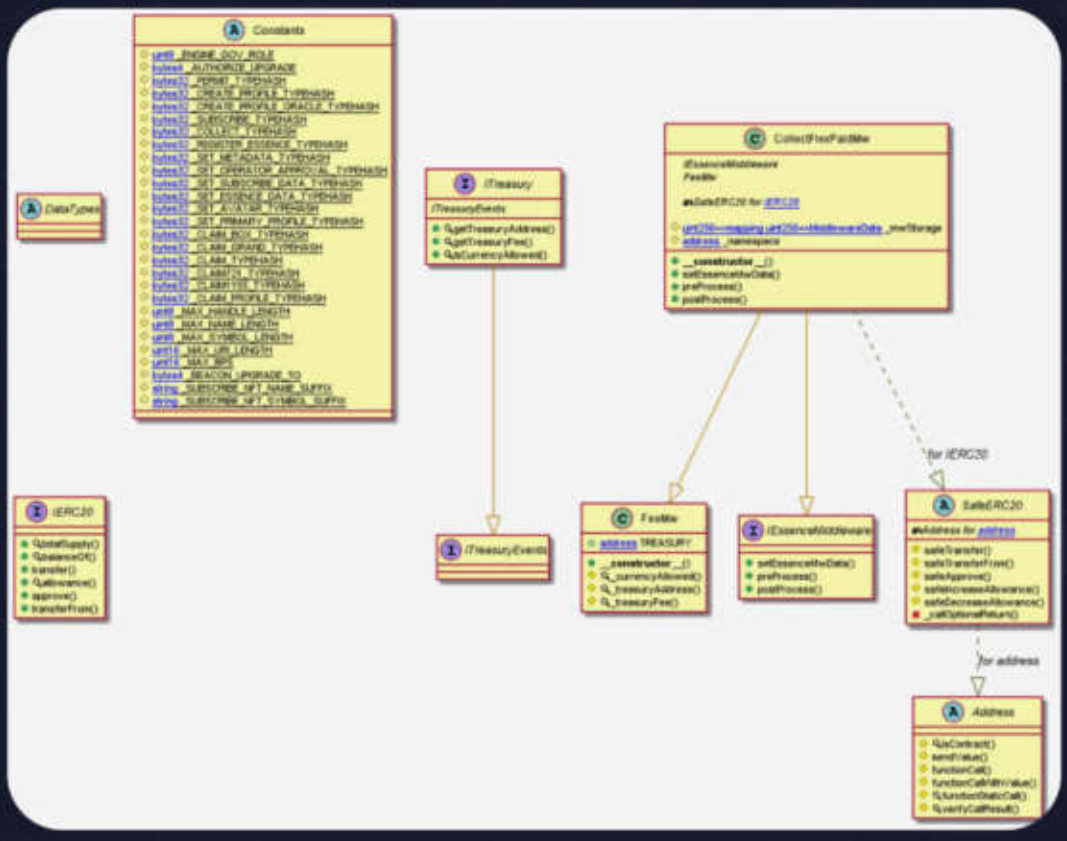


INSPECTOR LOVELY

CollectDisallowedMw Diagram

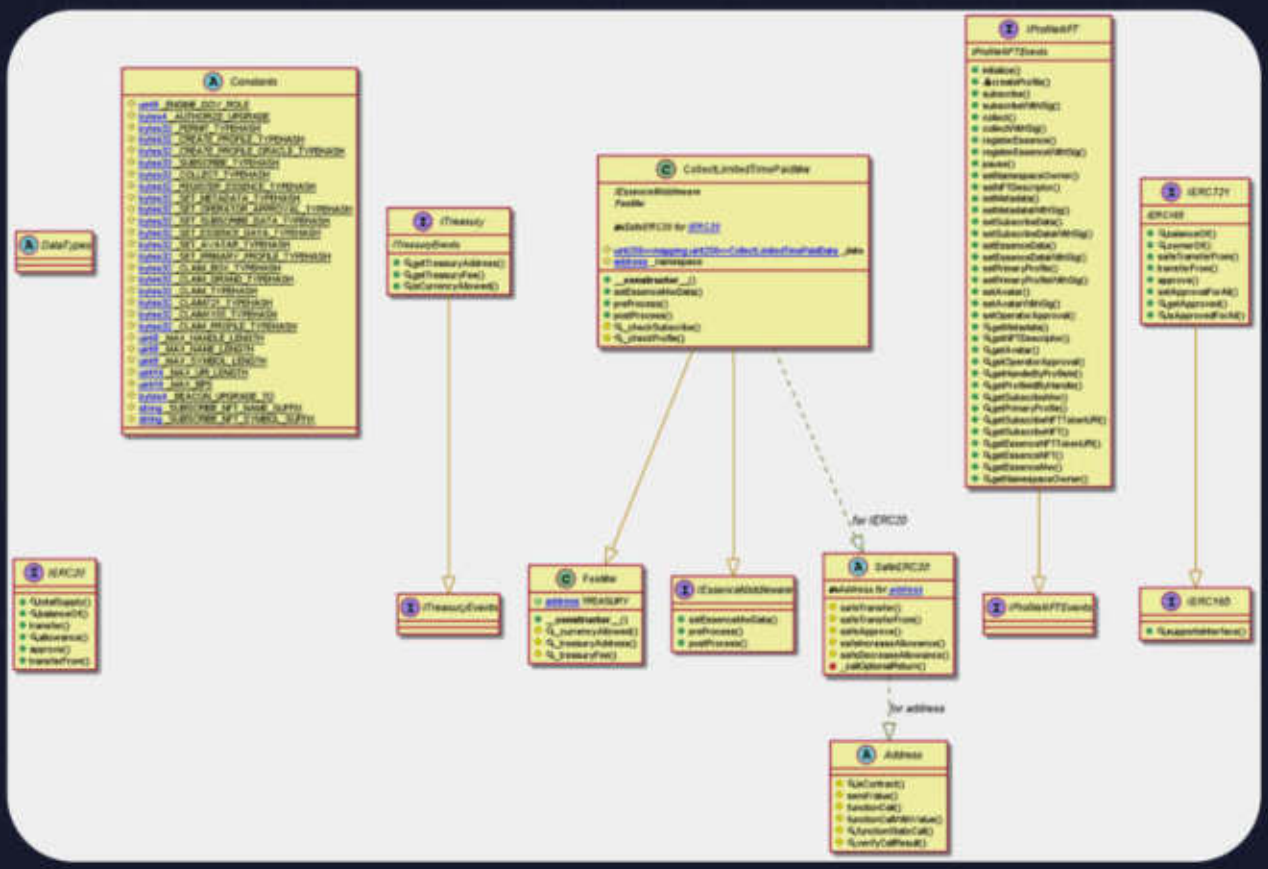


CollectDisallowedMw Diagram

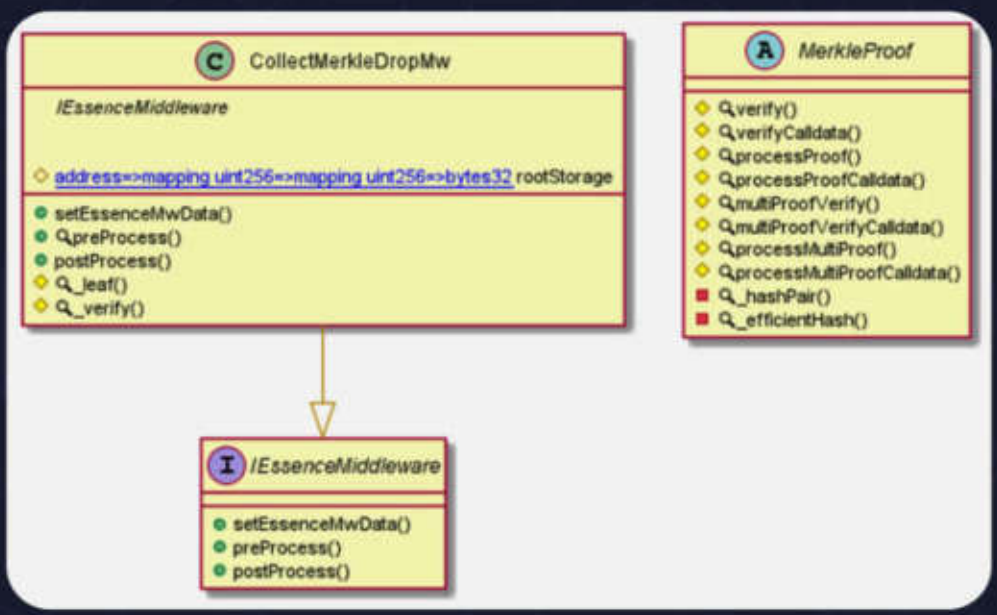


INSPECTOR LOVELY

CollectLimitedTimePaidMw Diagram

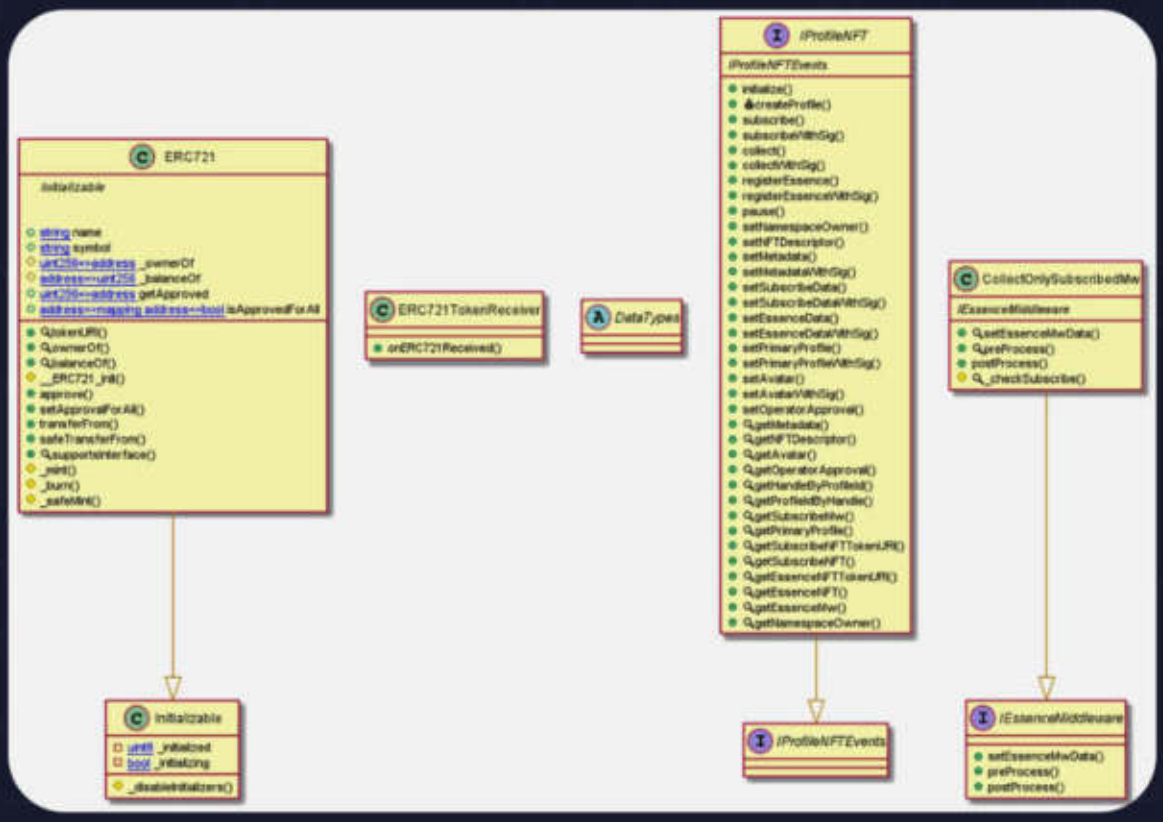


CollectMerkleDropMw Diagram

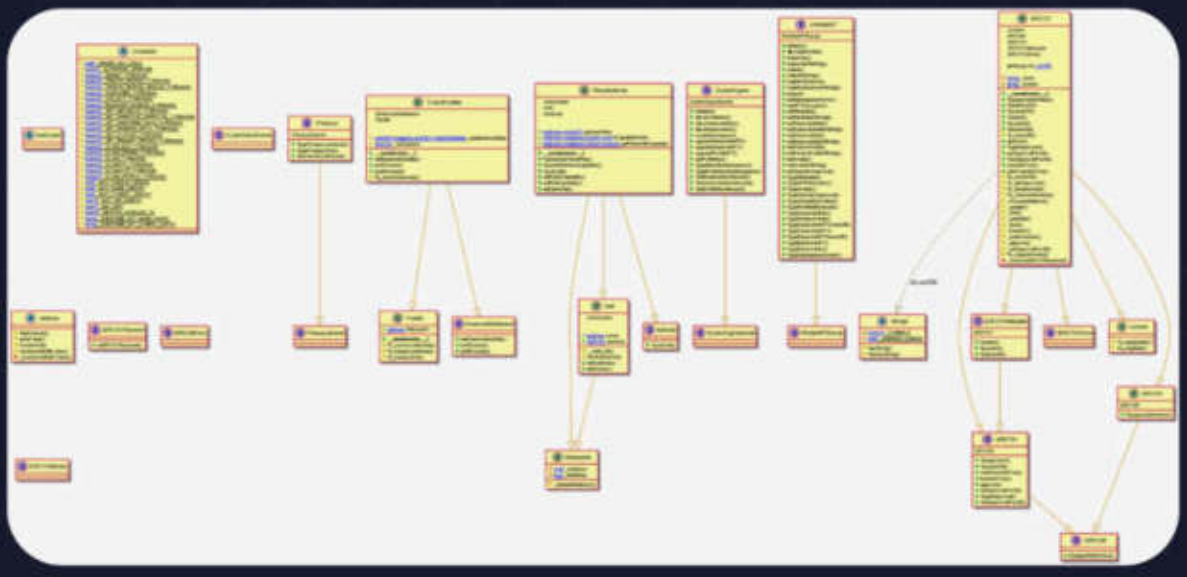


INSPECTOR LOVELY

CollectOnlySubscribedMw Diagram

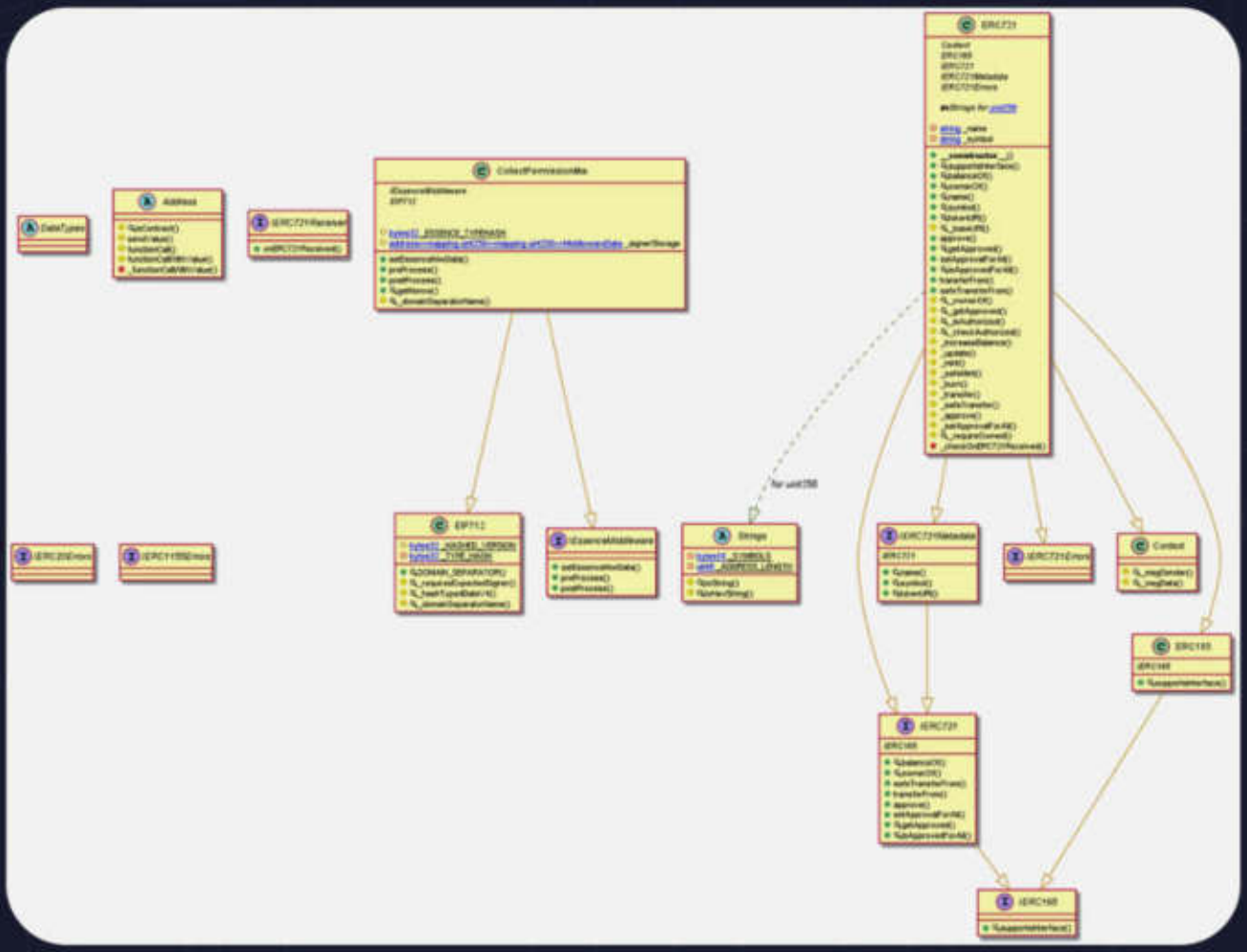


CollectPaidMw Diagram



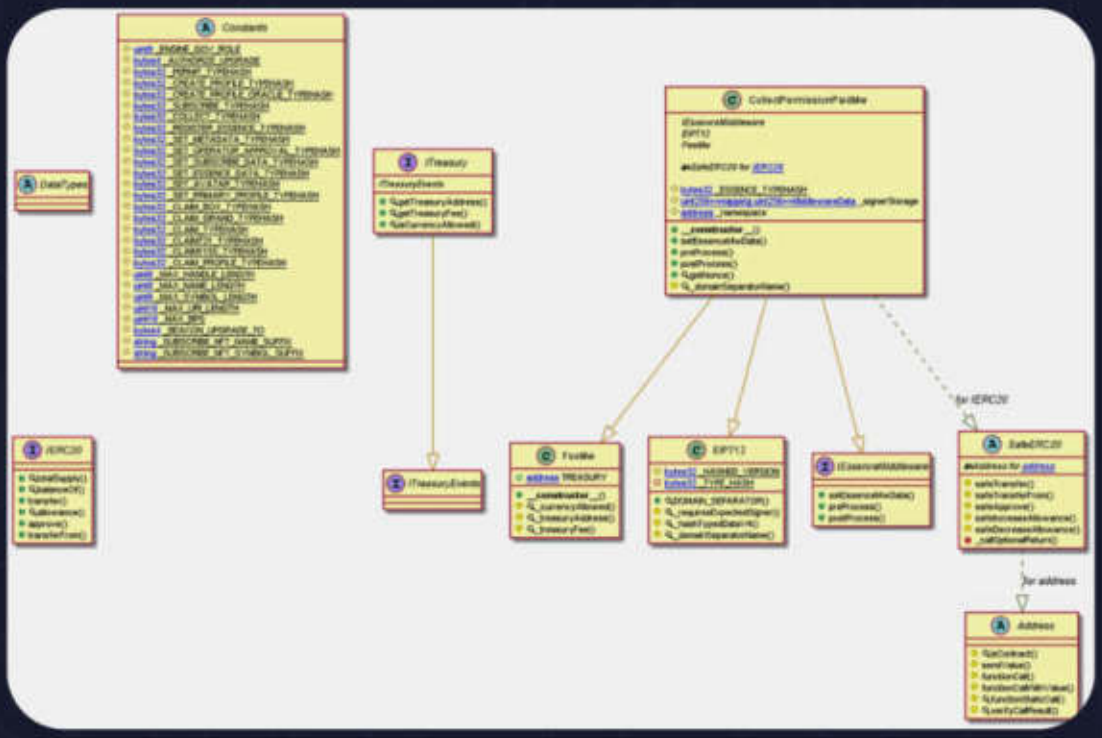
INSPECTOR LOVELY

CollectPermissionMw Diagram

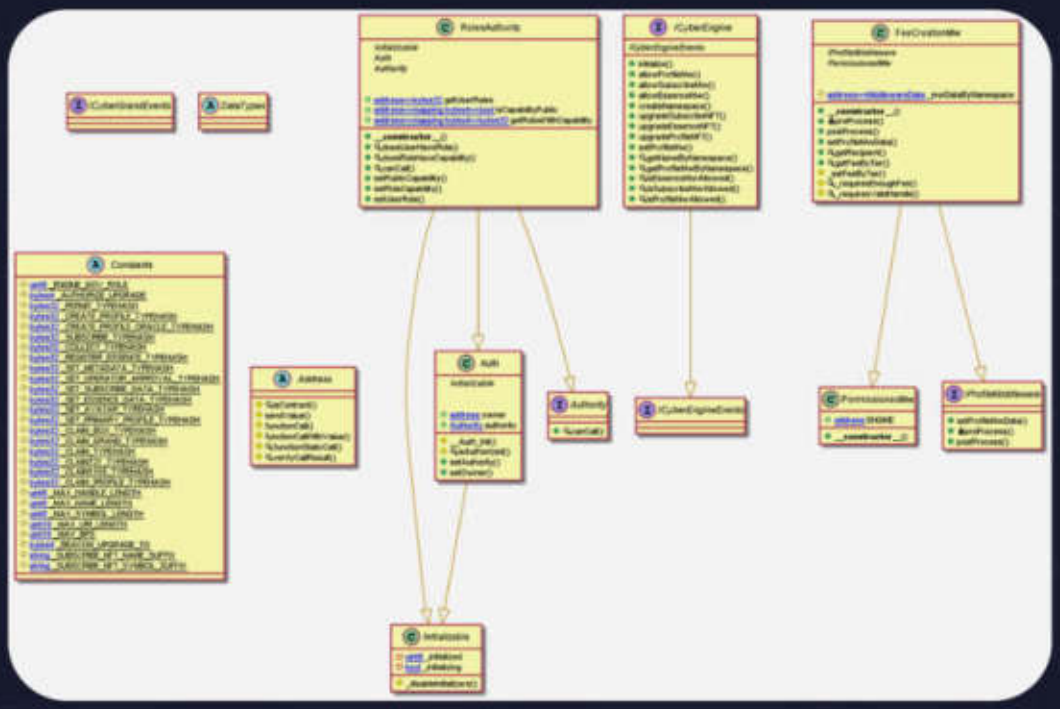


INSPECTOR LOVELY

CollectPermissionPaidMw Diagram

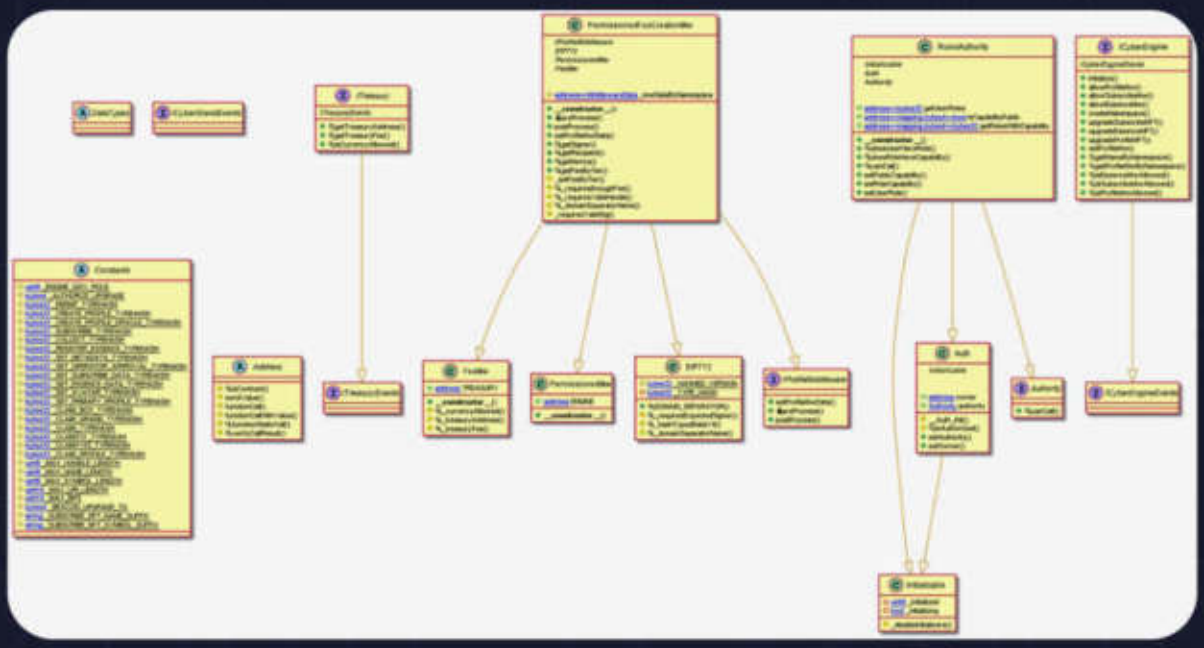


FeeCreationMw Diagram

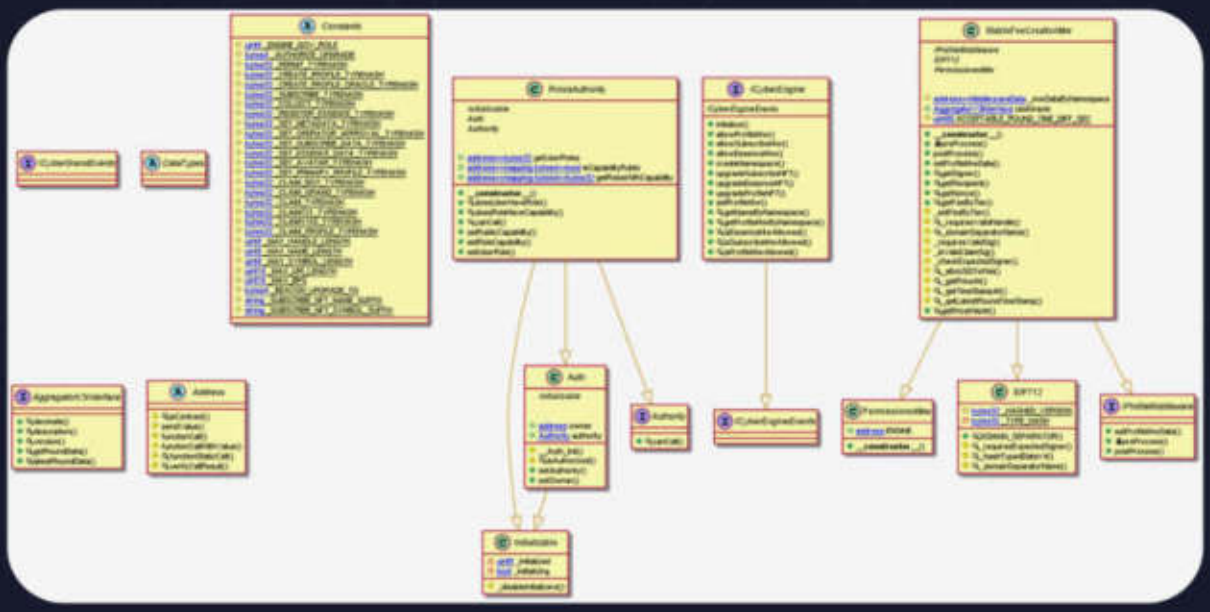


INSPECTOR LOVELY

PermissionedFeeCreationMw Diagram

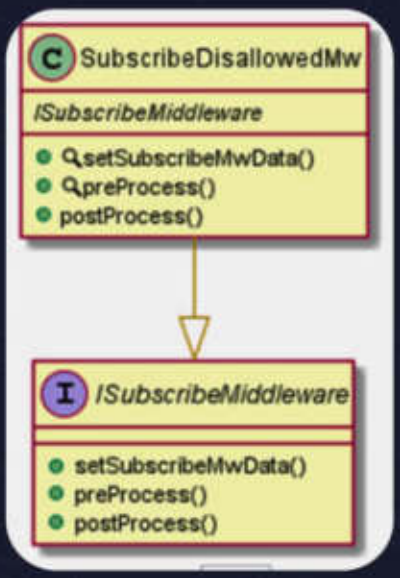


StableFeeCreationMw Diagram

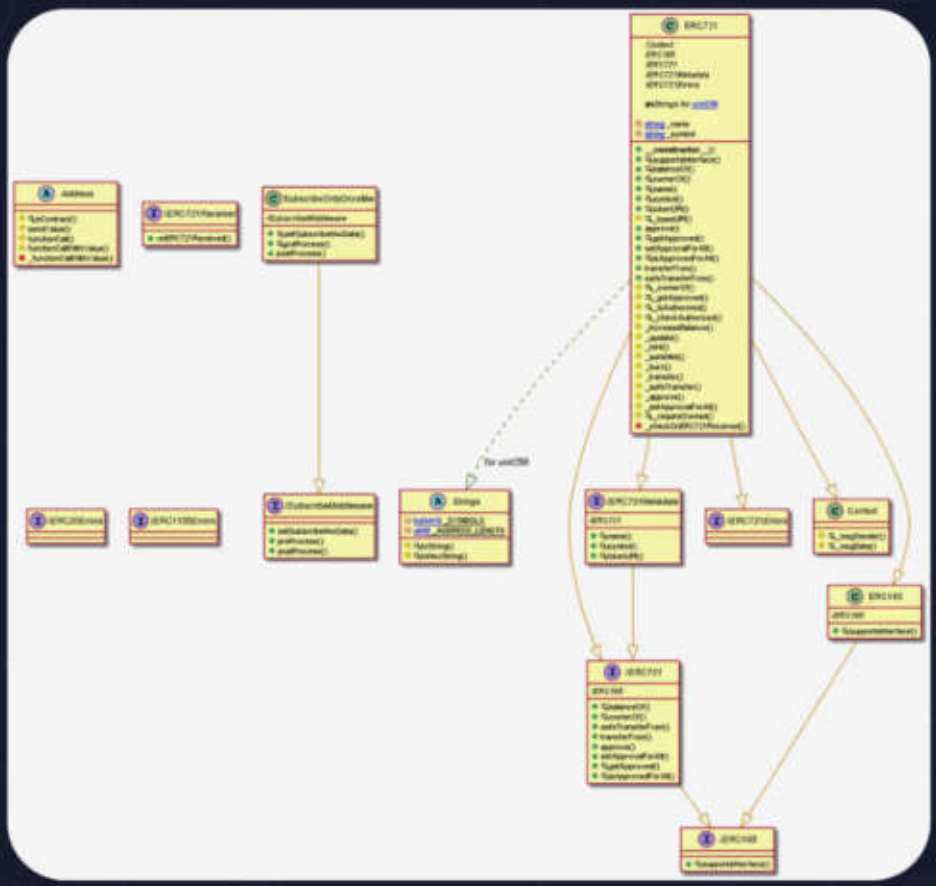


INSPECTOR LOVELY

SubscribeDisallowedMw Diagram

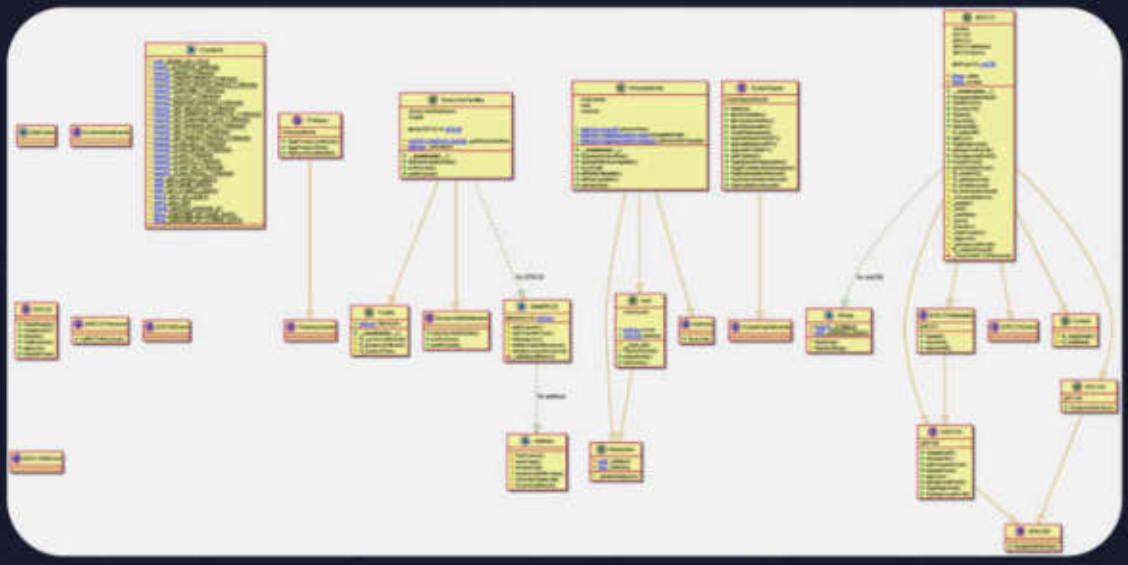


SubscribeOnlyOnceMw Diagram

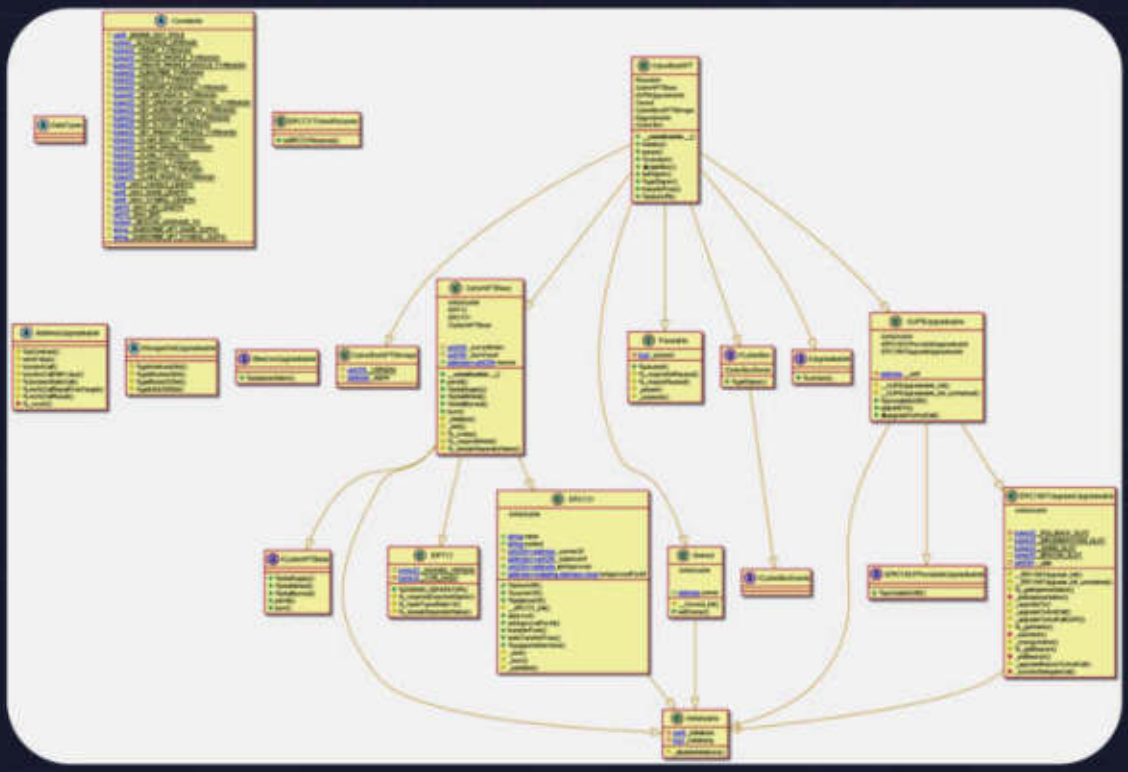


INSPECTOR LOVELY

SubscribePaidMw Diagram

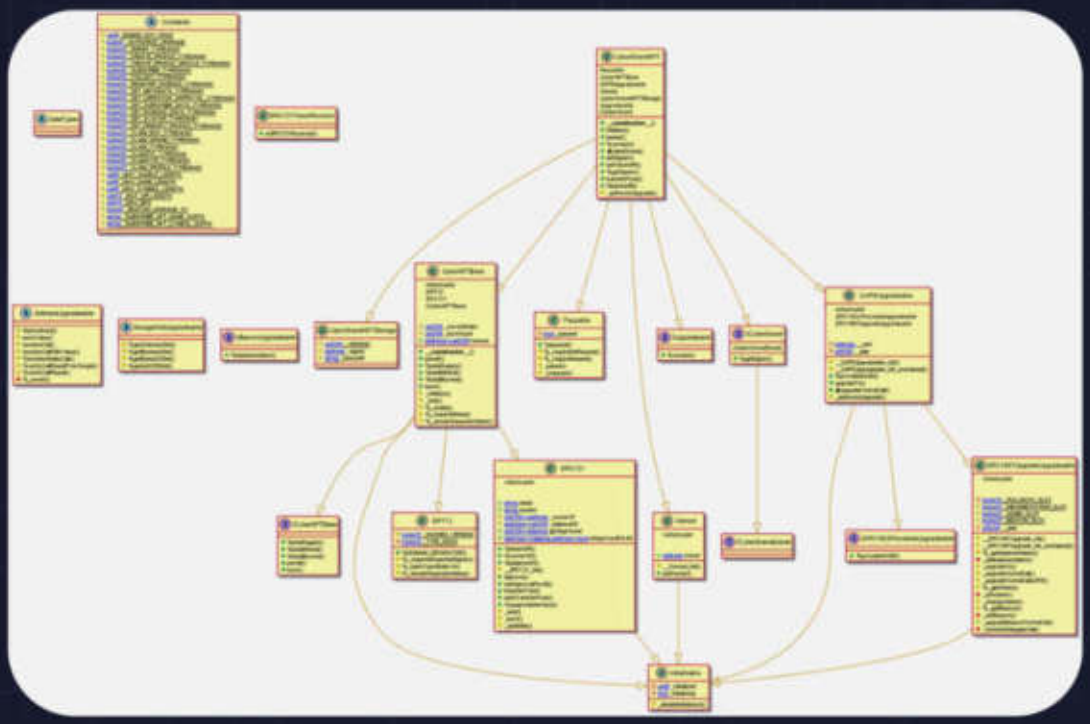


CyberBoxNFT Diagram

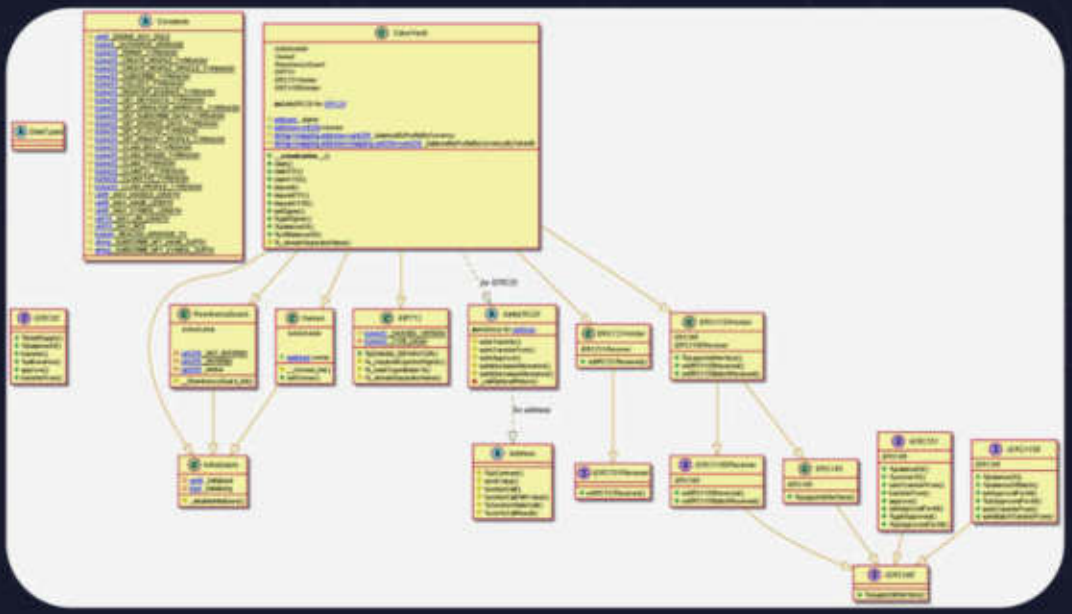


INSPECTOR LOVELY

CyberGrandNFT Diagram

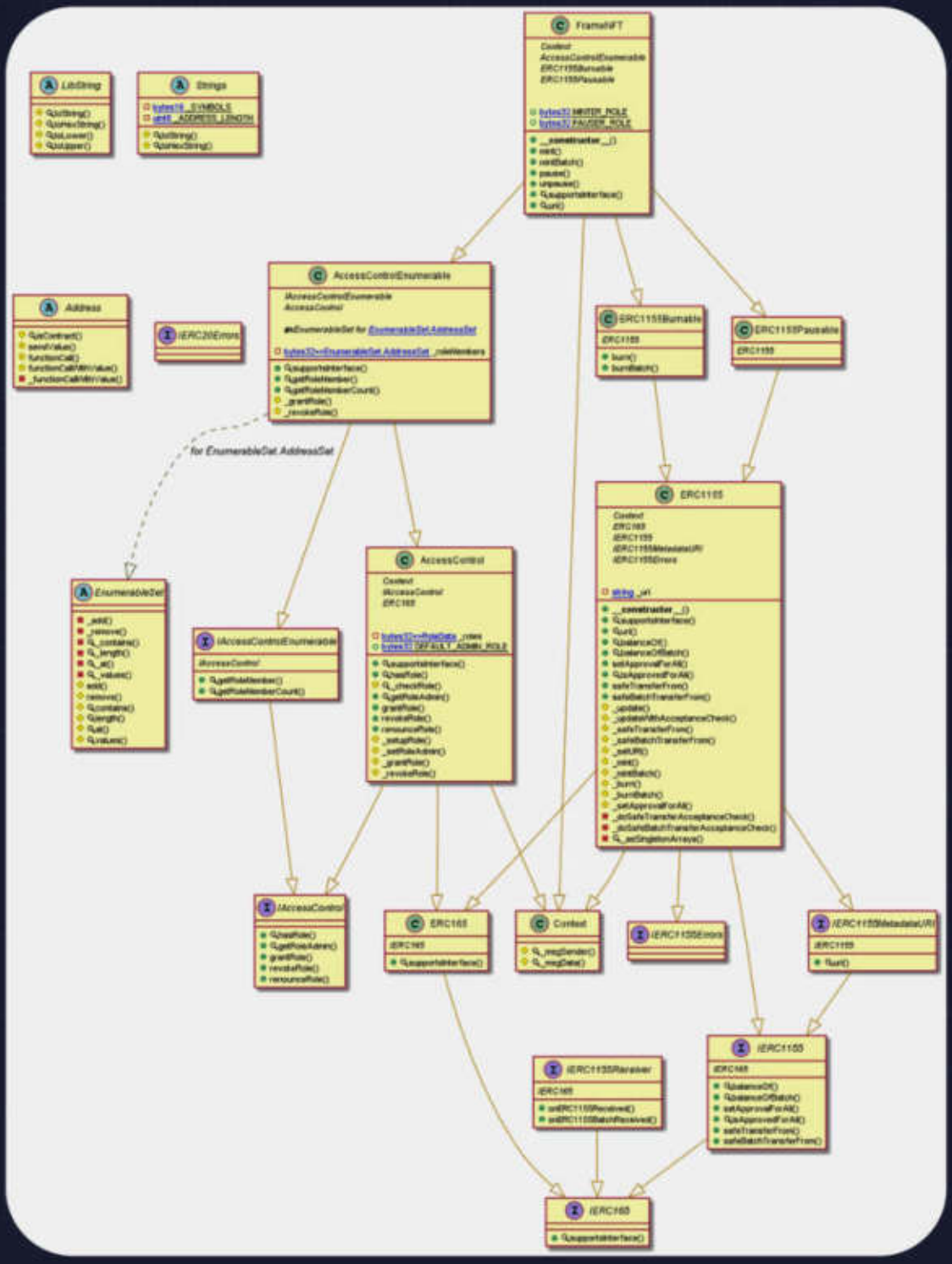


CyberVault Diagram



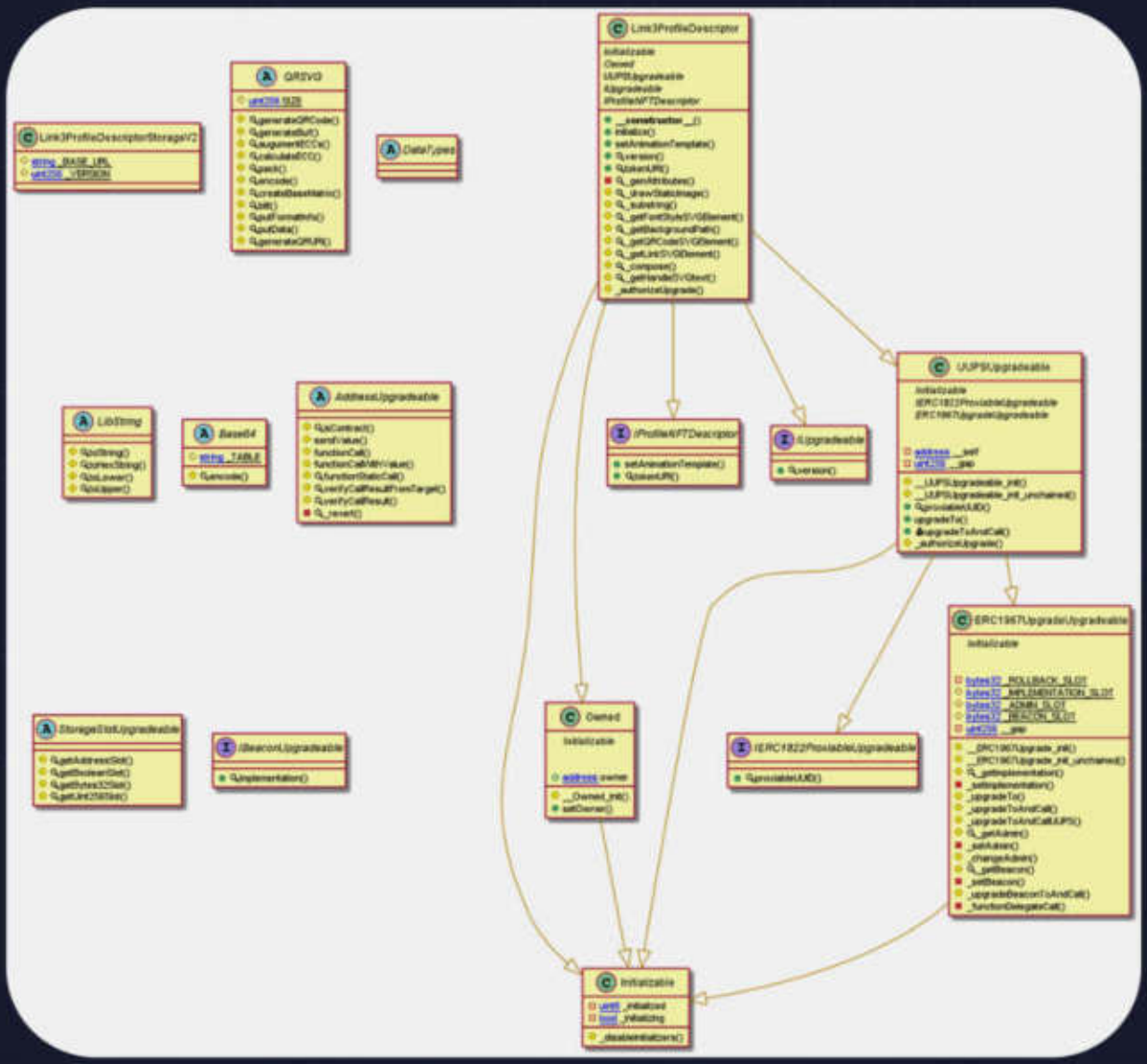
INSPECTOR LOVELY

FrameNFT Diagram



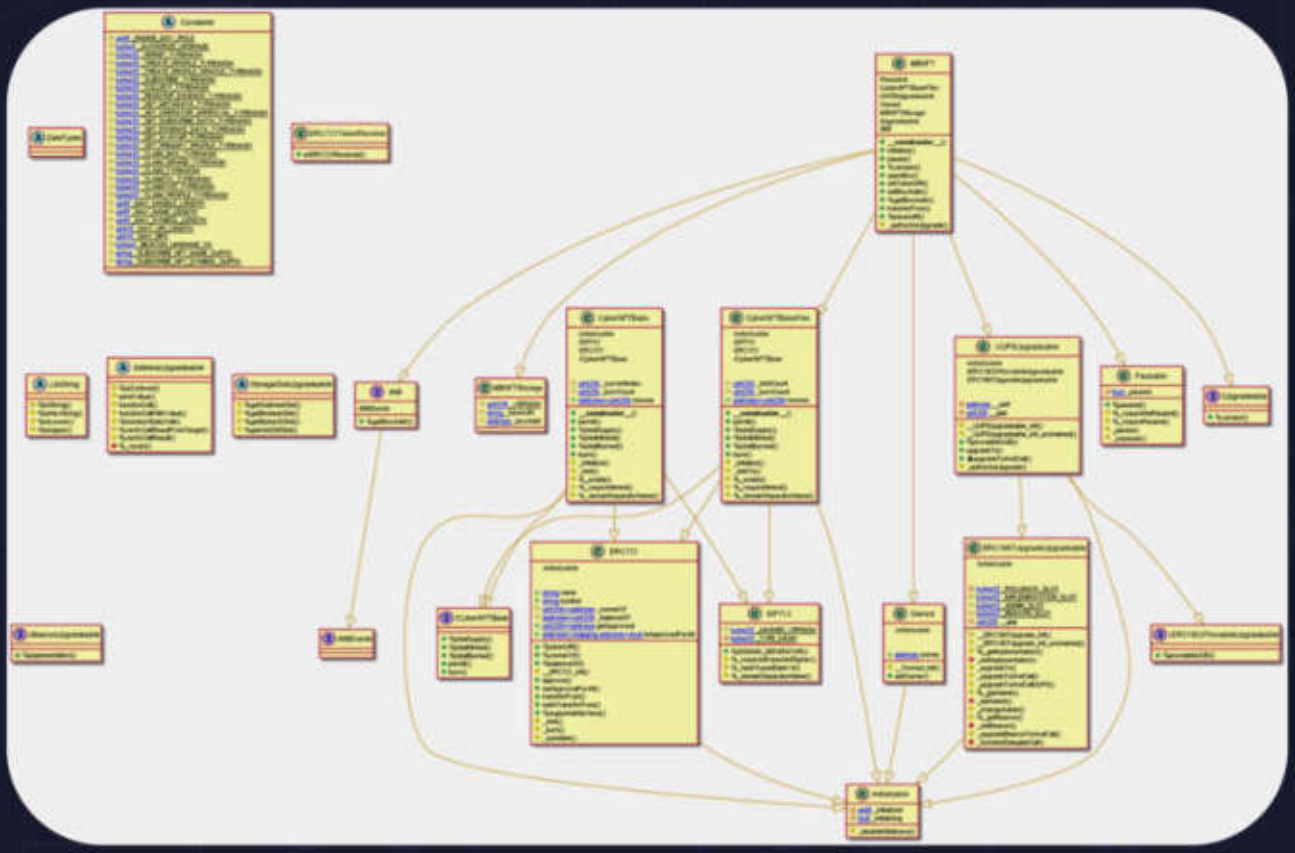
INSPECTOR LOVELY

Link3ProfileDescriptor Diagram



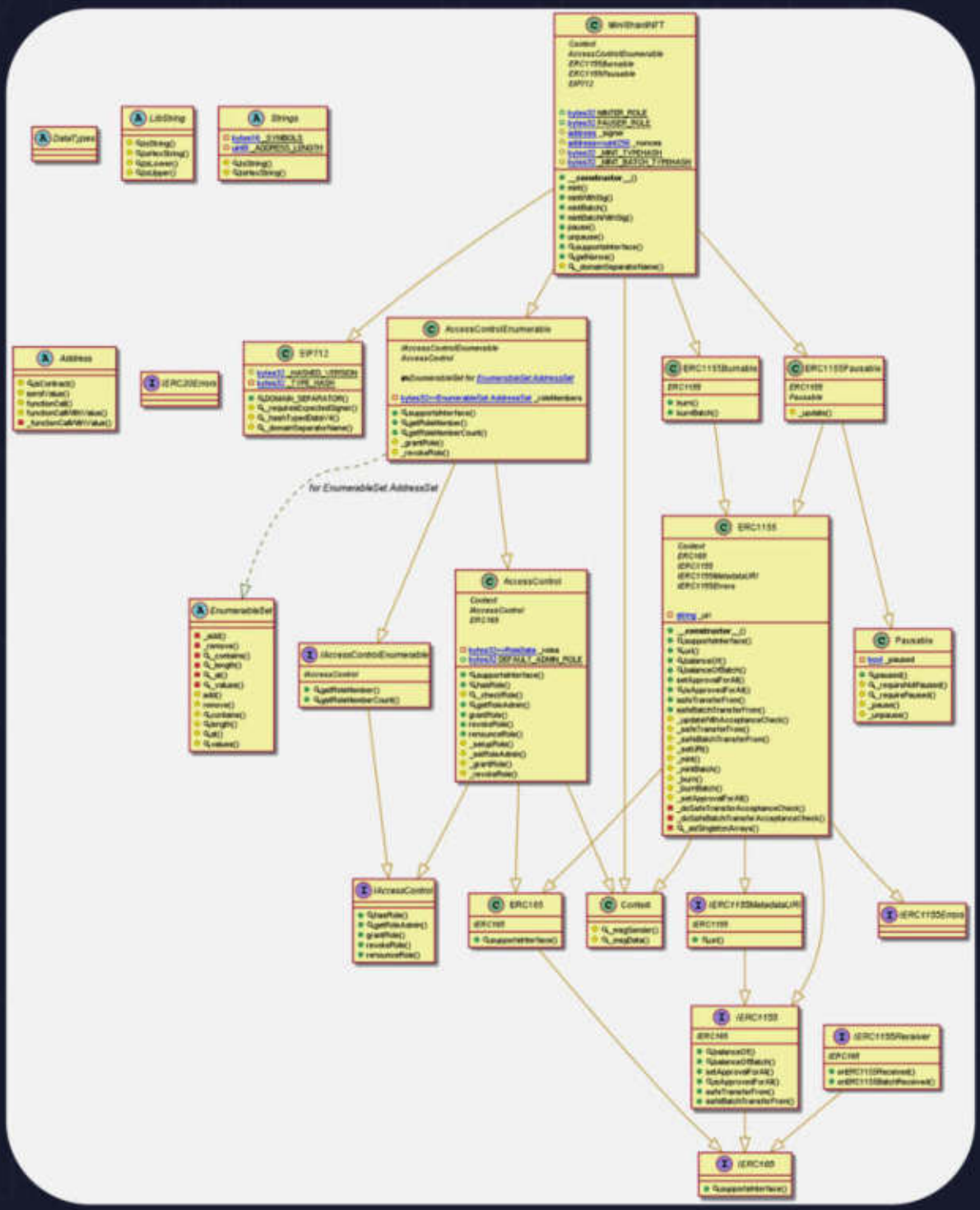
INSPECTOR LOVELY

MBNFT Diagram



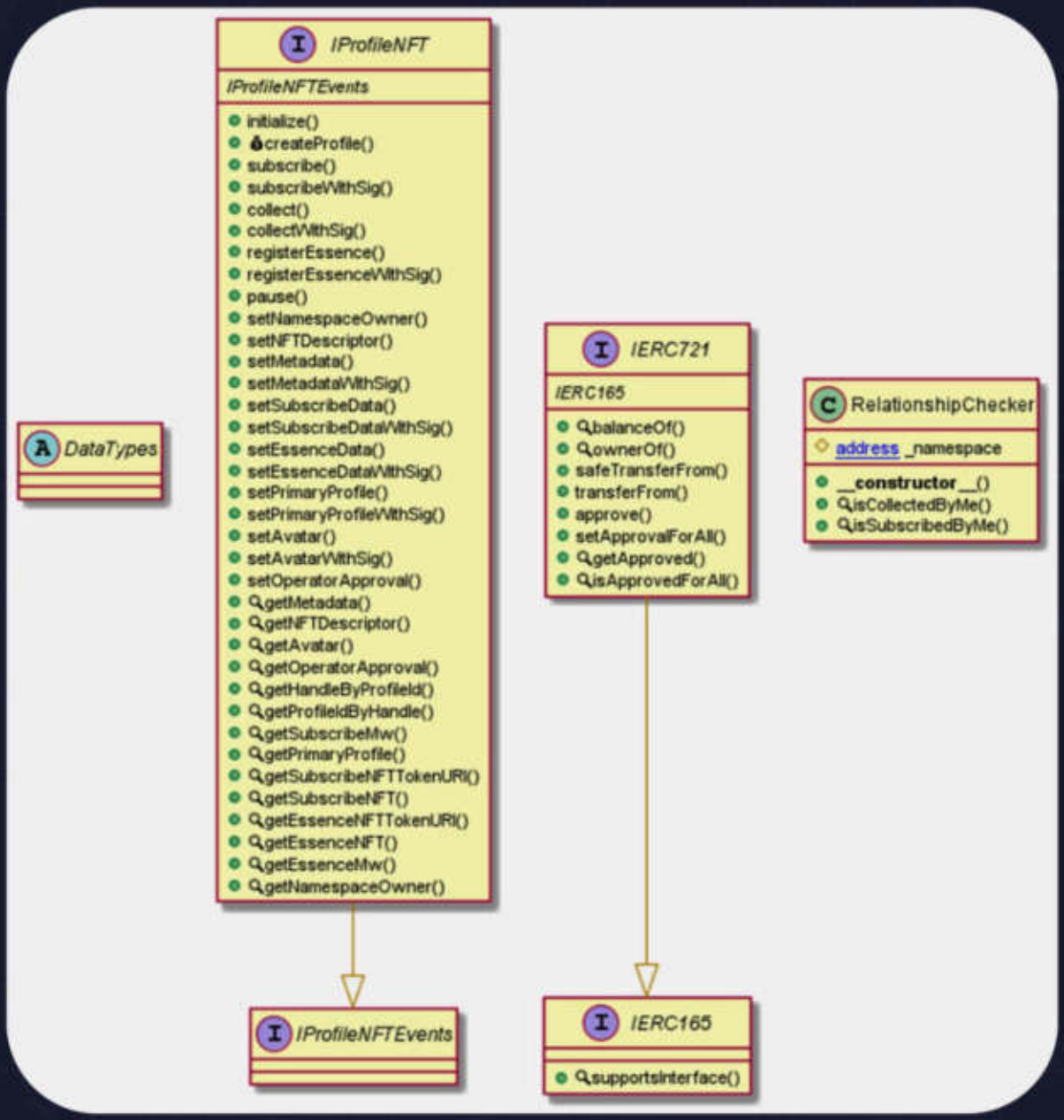
INSPECTOR LOVELY

MiniShardNFT Diagram



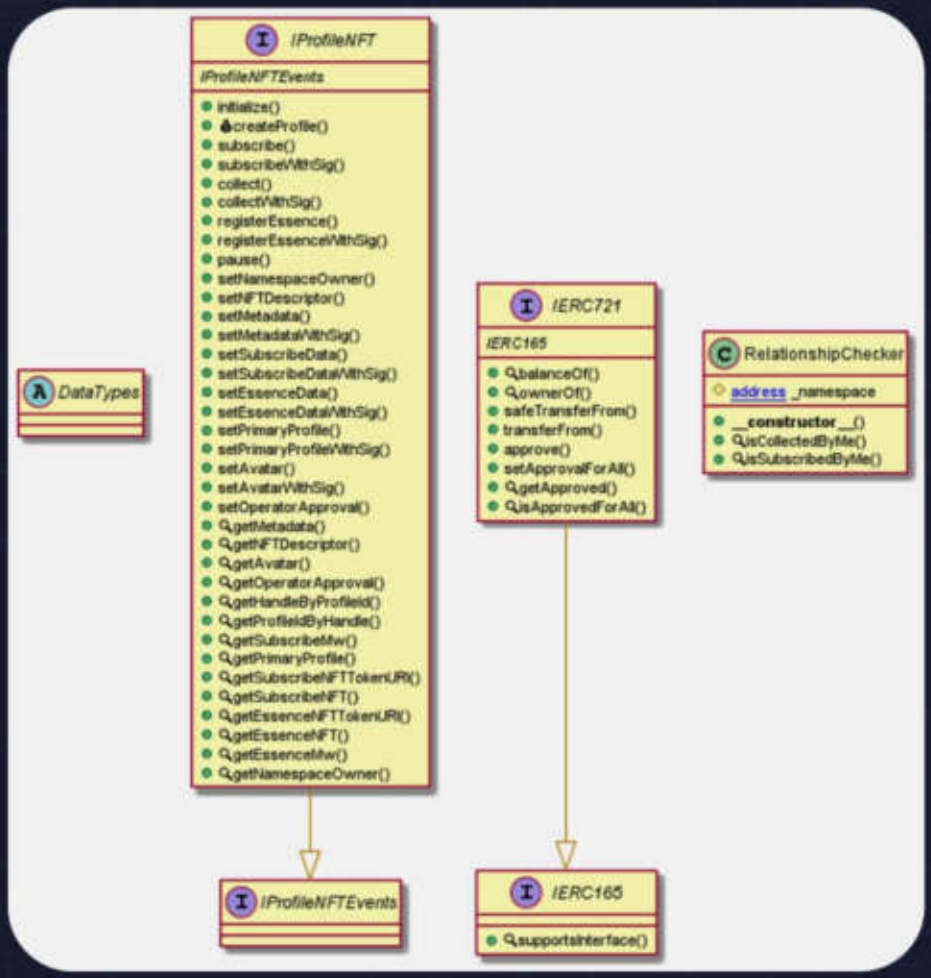
INSPECTOR LOVELY

RelationshipChecker Diagram



INSPECTOR LOVELY

RelationshipChecker Diagram



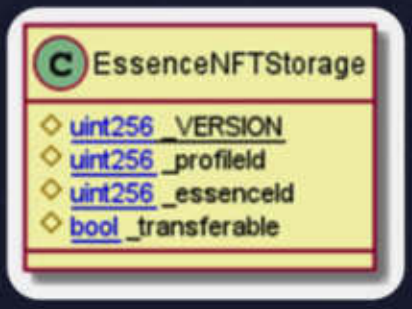
CyberBoxNFTStorage Diagram

INSPECTOR LOVELY

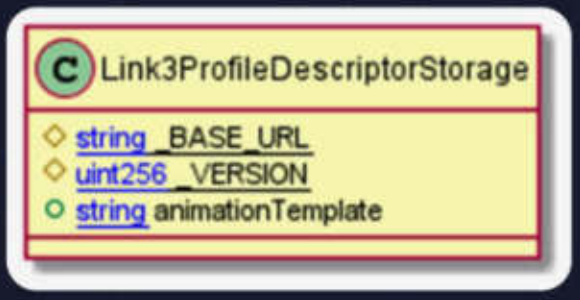
CyberEngineStorage Diagram

CyberGrandNFTStorage Diagram

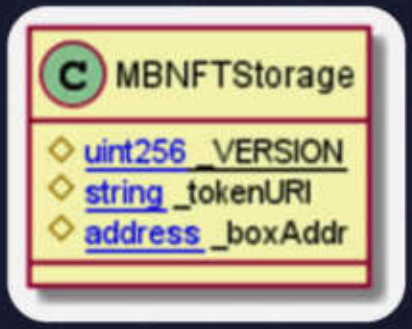
EssenceNFTStorage Diagram



Link3ProfileDescriptor Storage Diagram



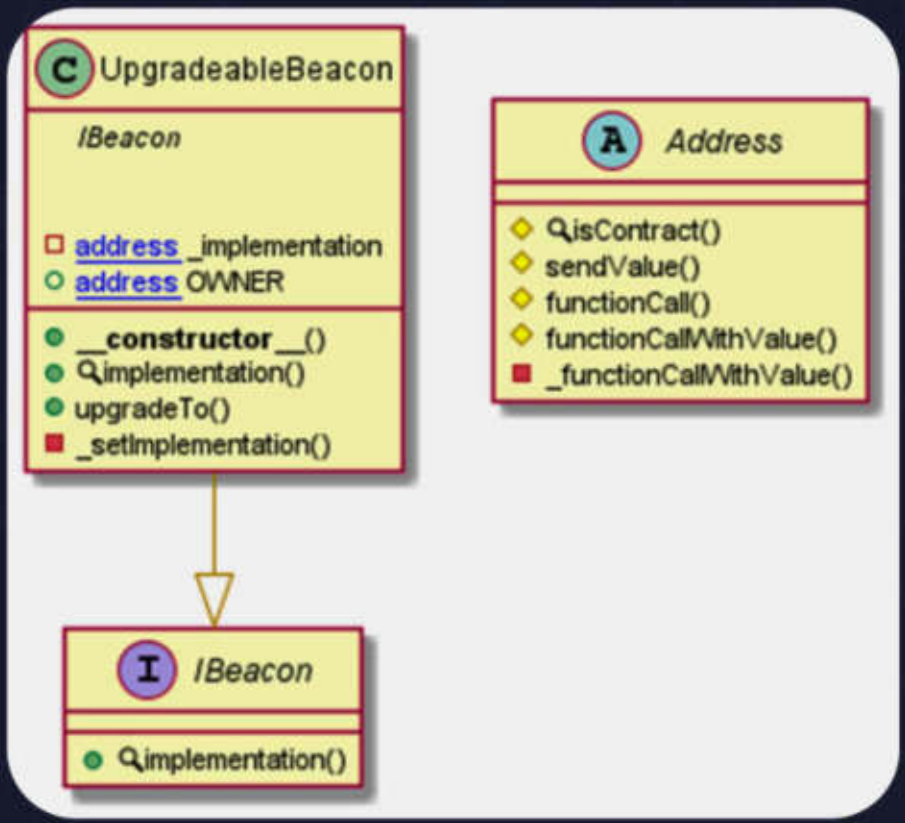
MBNFTStorage Diagram



ProfileNFTStorage Diagram

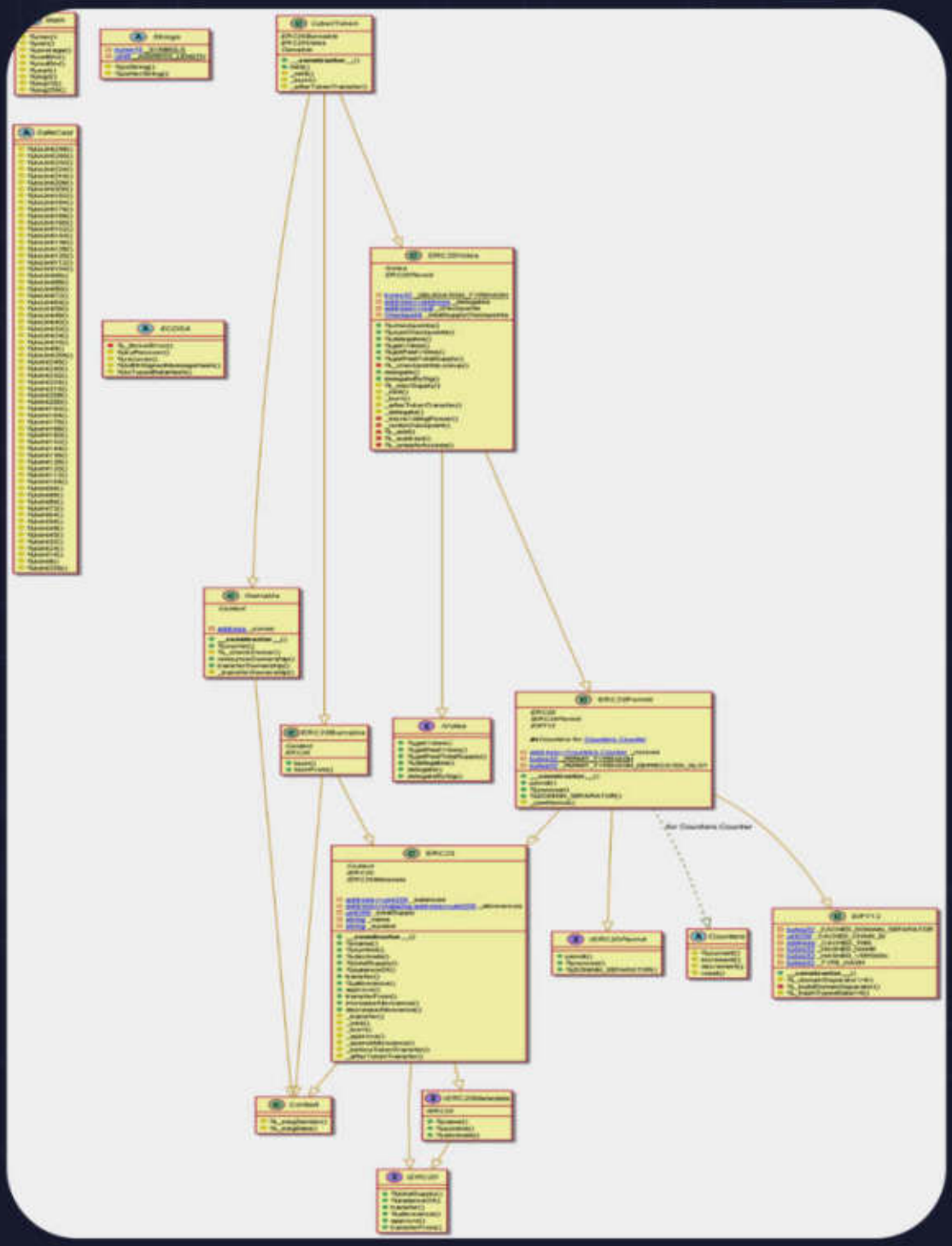
SubscribeNFTStorage Diagram

UpgradeableBeacon Diagram



INSPECTOR LOVELY

CYBER Diagram



INSPECTOR LOVELY

SECURITY ASSESSMENT REPORT

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

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

Slither Log >> CyberNFTBaseFlex.sol

```
Reentrancy in CyberNFTBaseFlex._mintTo(address,uint256) (CyberNFTBaseFlex.sol#747-750):
        External calls:
        - super._safeMint(_to,_id) (CyberNFTBaseFlex.sol#748)
                - require(bool,string)(ERC721TokenReceiver(to).onERC721Received(msg.sender,address(0),id,) == ERC721TokenReceive
r.onERC721Received.selector,UNSAFE_RECIPIENT) (CyberNFTBaseFlex.sol#524-528)
        State variables written after the call(s):
        - ++ _mintCount (CyberNFTBaseFlex.sol#749)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#reentrancy-vulnerabilities-2

EIP712._requiresExpectedSigner(bytes32,address,uint8,bytes32,bytes32,uint256) (CyberNFTBaseFlex.sol#599-616) uses timestamp for
comparisons
        Dangerous comparisons:
        - require(bool,string)(deadline >= block.timestamp,DEADLINE_EXCEEDED) (CyberNFTBaseFlex.sol#607)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#block-timestamp

CyberNFTBaseFlex._exists(uint256) (CyberNFTBaseFlex.sol#752-754) is never used and should be removed
CyberNFTBaseFlex._initialize(string,string) (CyberNFTBaseFlex.sol#740-745) is never used and should be removed
CyberNFTBaseFlex._mintTo(address,uint256) (CyberNFTBaseFlex.sol#747-750) is never used and should be removed
CyberNFTBaseFlex._requireMinted(uint256) (CyberNFTBaseFlex.sol#756-758) is never used and should be removed
ERC721._ERC721_Init(string,string) (CyberNFTBaseFlex.sol#381-387) is never used and should be removed
ERC721._mint(address,uint256) (CyberNFTBaseFlex.sol#484-497) is never used and should be removed
ERC721._safeMint(address,uint256) (CyberNFTBaseFlex.sol#520-529) is never used and should be removed
ERC721._safeMint(address,uint256,bytes) (CyberNFTBaseFlex.sol#531-544) is never used and should be removed
Initializable._disableInitializers() (CyberNFTBaseFlex.sol#317-323) is never used and should be removed
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#dead-code

Pragma version0.8.14 (CyberNFTBaseFlex.sol#3) allows old versions
solc-0.8.14 is not recommended for deployment
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#incorrect-versions-of-solidity

Function ERC721._ERC721_Init(string,string) (CyberNFTBaseFlex.sol#381-387) is not in mixedCase
Parameter ERC721._ERC721_Init(string,string)._name (CyberNFTBaseFlex.sol#381) is not in mixedCase
Parameter ERC721._ERC721_Init(string,string)._symbol (CyberNFTBaseFlex.sol#381) is not in mixedCase
Variable ERC721._ownerOf (CyberNFTBaseFlex.sol#355) is not in mixedCase
Variable ERC721._balanceOf (CyberNFTBaseFlex.sol#357) is not in mixedCase
Function EIP712.DOMAIN_SEPARATOR() (CyberNFTBaseFlex.sol#582-593) is not in mixedCase
Variable CyberNFTBaseFlex._mintCount (CyberNFTBaseFlex.sol#656) is not in mixedCase
Variable CyberNFTBaseFlex._burnCount (CyberNFTBaseFlex.sol#657) is not in mixedCase
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#conformance-to-solidity-naming-conventions

CyberNFTBaseFlex (CyberNFTBaseFlex.sol#652-769) does not implement functions:
        - ERC721.tokenURI(uint256) (CyberNFTBaseFlex.sol#349)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#unimplemented-functions
CyberNFTBaseFlex.sol analyzed (8 contracts with 84 detectors), 22 result(s) found
```

INSPECTOR LOVELY

Slither Log >> EIP712.sol

```
EIP712._requiresExpectedSigner(bytes32,address,uint8,bytes32,bytes32,uint256) (EIP712.sol#178-195) uses timestamp for comparison
s
        Dangerous comparisons:
        - require(bool,string)(deadline >= block.timestamp,DEADLINE_EXCEEDED) (EIP712.sol#186)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#block-timestamp

EIP712._hashTypedDataV4(bytes32) (EIP712.sol#212-222) is never used and should be removed
EIP712._requiresExpectedSigner(bytes32,address,DataTypes.EIP712Signature) (EIP712.sol#197-210) is never used and should be remov
ed
EIP712._requiresExpectedSigner(bytes32,address,uint8,bytes32,bytes32,uint256) (EIP712.sol#178-195) is never used and should be r
emoved
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#dead-code

Pragma version0.8.14 (EIP712.sol#3) allows old versions
solc-0.8.14 is not recommended for deployment
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#incorrect-versions-of-solidity

Function EIP712.DOMAIN_SEPARATOR() (EIP712.sol#161-172) is not in mixedCase
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#conformance-to-solidity-naming-conventions

EIP712 (EIP712.sol#141-229) does not implement functions:
        - EIP712._domainSeparatorName() (EIP712.sol#224-228)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#unimplemented-functions
EIP712.sol analyzed (2 contracts with 84 detectors), 8 result(s) found
```

Slither Log >> CyberNFTBase.sol

```
EIP712._requiresExpectedSigner(bytes32,address,uint8,bytes32,bytes32,uint256) (CyberNFTBase.sol#349-366) uses timestamp for comp
arisons
        Dangerous comparisons:
        - require(bool,string)(deadline >= block.timestamp,DEADLINE_EXCEEDED) (CyberNFTBase.sol#357)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#block-timestamp

CyberNFTBase._exists(uint256) (CyberNFTBase.sol#742-744) is never used and should be removed
CyberNFTBase._initialize(string,string) (CyberNFTBase.sol#730-735) is never used and should be removed
CyberNFTBase._mint(address) (CyberNFTBase.sol#737-740) is never used and should be removed
CyberNFTBase._requireMinted(uint256) (CyberNFTBase.sol#746-748) is never used and should be removed
ERC721._ERC721_Init(string,string) (CyberNFTBase.sol#457-463) is never used and should be removed
ERC721._mint(address,uint256) (CyberNFTBase.sol#560-573) is never used and should be removed
ERC721._safeMint(address,uint256) (CyberNFTBase.sol#596-605) is never used and should be removed
ERC721._safeMint(address,uint256,bytes) (CyberNFTBase.sol#607-620) is never used and should be removed
Initializable._disableInitializers() (CyberNFTBase.sol#302-308) is never used and should be removed
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#dead-code

Pragma version0.8.14 (CyberNFTBase.sol#3) allows old versions
solc-0.8.14 is not recommended for deployment
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#incorrect-versions-of-solidity

Function EIP712.DOMAIN_SEPARATOR() (CyberNFTBase.sol#332-343) is not in mixedCase
Function ERC721._ERC721_Init(string,string) (CyberNFTBase.sol#457-463) is not in mixedCase
Parameter ERC721._ERC721_Init(string,string)._name (CyberNFTBase.sol#457) is not in mixedCase
Parameter ERC721._ERC721_Init(string,string)._symbol (CyberNFTBase.sol#457) is not in mixedCase
Variable ERC721._ownerOf (CyberNFTBase.sol#431) is not in mixedCase
Variable ERC721._balanceOf (CyberNFTBase.sol#433) is not in mixedCase
Variable CyberNFTBase._currentIndex (CyberNFTBase.sol#646) is not in mixedCase
Variable CyberNFTBase._burnCount (CyberNFTBase.sol#647) is not in mixedCase
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#conformance-to-solidity-naming-conventions

CyberNFTBase (CyberNFTBase.sol#642-759) does not implement functions:
        - ERC721.tokenURI(uint256) (CyberNFTBase.sol#425)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#unimplemented-functions
CyberNFTBase.sol analyzed (8 contracts with 84 detectors), 21 result(s) found
```

INSPECTOR LOVELY

Slither Log >> CyberEngine.sol

```
ProfileNFT.initialize(address,string,string).name (CyberEngine.sol#2560) shadows:
        - ERC721.name (CyberEngine.sol#615) (state variable)
ProfileNFT.initialize(address,string,string).symbol (CyberEngine.sol#2570) shadows:
        - ERC721.symbol (CyberEngine.sol#617) (state variable)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#local-variable-shadowing

Auth.setOwner(address).newOwner (CyberEngine.sol#3526) lacks a zero-check on :
                - owner = newOwner (CyberEngine.sol#3527)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#missing-zero-address-validation

Variable 'ERC1967UpgradeUpgradeable._upgradeToAndCallUUPS(address,bytes,bool).slot (CyberEngine.sol#2374)' in ERC1967UpgradeUpgr
adeable._upgradeToAndCallUUPS(address,bytes,bool) (CyberEngine.sol#2366-2381) potentially used before declaration: require(bool,
string)(slot == _IMPLEMENTATION_SLOT,ERC1967Upgrade: unsupported proxiableUUID) (CyberEngine.sol#2375)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#pre-declaration-usage-of-local-variables

Reentrancy in Actions.collect(DataTypes.CollectData,mapping(uint256 => mapping(uint256 => DataTypes.EssenceStruct))) (CyberEngin
e.sol#1035-1081):
        External calls:
        - tokenId = IEssenceNFT(essenceNFT).mint(data.collector) (CyberEngine.sol#1071)
        Event emitted after the call(s):
        - CollectEssence(data.collector,data.profileId,data.essenceId,tokenId,data.preData,data.postData) (CyberEngine.sol#1073-
1080)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#reentrancy-vulnerabilities-3

EIP712._requiresExpectedSigner(bytes32,address,uint8,bytes32,bytes32,uint256) (CyberEngine.sol#543-560) uses timestamp for compa
risons
        Dangerous comparisons:
        - require(bool,string)(deadline >= block.timestamp,DEADLINE_EXCEEDED) (CyberEngine.sol#551)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#block-timestamp

Function Auth.__Auth_Init(address,Authority) (CyberEngine.sol#3494-3500) is not in mixedCase
Parameter Auth.__Auth_Init(address,Authority)._owner (CyberEngine.sol#3494) is not in mixedCase
Parameter Auth.__Auth_Init(address,Authority)._authority (CyberEngine.sol#3494) is not in mixedCase
Parameter CyberEngine.initialize(address,RolesAuthority)._owner (CyberEngine.sol#4166) is not in mixedCase
Parameter CyberEngine.initialize(address,RolesAuthority)._rolesAuthority (CyberEngine.sol#4166) is not in mixedCase
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#conformance-to-solidity-naming-conventions

Variable UpgradeableBeacon._implementation (CyberEngine.sol#97) is too similar to UpgradeableBeacon.constructor(address,address)
.implementation_ (CyberEngine.sol#109)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#variable-names-too-similar

CyberEngineStorage._VERSION_STRING (CyberEngine.sol#288) is never used in CyberEngine (CyberEngine.sol#4116-4407)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#unused-state-variable
CyberEngine.sol analyzed (42 contracts with 84 detectors), 134 result(s) found
```

```
EssenceNFT.initialize(uint256,uint256,string,string,bool).name (EssenceNFT.sol#1095) shadows:
        - ERC721.name (EssenceNFT.sol#443) (state variable)
EssenceNFT.initialize(uint256,uint256,string,string,bool).symbol (EssenceNFT.sol#1097) shadows:
        - ERC721.symbol (EssenceNFT.sol#445) (state variable)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#local-variable-shadowing

EIP712._requiresExpectedSigner(bytes32,address,uint8,bytes32,bytes32,uint256) (EssenceNFT.sol#371-388) uses timestamp for compar
isons
        Dangerous comparisons:
        - require(bool,string)(deadline >= block.timestamp,DEADLINE_EXCEEDED) (EssenceNFT.sol#379)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#block-timestamp

ERC721._safeMint(address,uint256,bytes) (EssenceNFT.sol#629-642) is never used and should be removed
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#dead-code

Pragma version0.8.14 (EssenceNFT.sol#1) allows old versions
solc-0.8.14 is not recommended for deployment
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#incorrect-versions-of-solidity

Variable EssenceNFTStorage._profileId (EssenceNFT.sol#12) is not in mixedCase
Variable EssenceNFTStorage._essenceId (EssenceNFT.sol#13) is not in mixedCase
Variable EssenceNFTStorage._transferable (EssenceNFT.sol#14) is not in mixedCase
Function EIP712.DOMAIN_SEPARATOR() (EssenceNFT.sol#354-365) is not in mixedCase
Function ERC721.__ERC721_Init(string,string) (EssenceNFT.sol#479-485) is not in mixedCase
Parameter ERC721.__ERC721_Init(string,string)._name (EssenceNFT.sol#479) is not in mixedCase
Parameter ERC721.__ERC721_Init(string,string)._symbol (EssenceNFT.sol#479) is not in mixedCase
Variable ERC721._ownerOf (EssenceNFT.sol#453) is not in mixedCase
Variable ERC721._balanceOf (EssenceNFT.sol#455) is not in mixedCase
Variable CyberNFTBase._currentIndex (EssenceNFT.sol#668) is not in mixedCase
Variable CyberNFTBase._burnCount (EssenceNFT.sol#669) is not in mixedCase
Variable EssenceNFT.PROFILE (EssenceNFT.sol#1075) is not in mixedCase
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#conformance-to-solidity-naming-conventions
EssenceNFT.sol analyzed (15 contracts with 84 detectors), 18 result(s) found
```

INSPECTOR LOVELY

Slither Log >> ProfileNFT.sol

```
ProfileNFT.initialize(address,string,string).name (ProfileNFT.sol#2340) shadows:
        - ERC721.name (ProfileNFT.sol#455) (state variable)
ProfileNFT.initialize(address,string,string).symbol (ProfileNFT.sol#2341) shadows:
        - ERC721.symbol (ProfileNFT.sol#457) (state variable)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#local-variable-shadowing

Variable 'ERC1967UpgradeUpgradeable._upgradeToAndCallUUPS(address,bytes,bool).slot (ProfileNFT.sol#2145)' in ERC1967UpgradeUpgra
deable._upgradeToAndCallUUPS(address,bytes,bool) (ProfileNFT.sol#2137-2152) potentially used before declaration: require(bool,st
ring)(slot == _IMPLEMENTATION_SLOT,ERC1967Upgrade: unsupported proxiableUUID) (ProfileNFT.sol#2146)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#pre-declaration-usage-of-local-variables

EIP712._requiresExpectedSigner(bytes32,address,uint8,bytes32,bytes32,uint256) (ProfileNFT.sol#383-400) uses timestamp for compar
isons
        Dangerous comparisons:
        - require(bool,string)(deadline >= block.timestamp,DEADLINE_EXCEEDED) (ProfileNFT.sol#391)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#block-timestamp

AddressUpgradeable._revert(bytes,string) (ProfileNFT.sol#1914-1923) uses assembly
        - INLINE ASM (ProfileNFT.sol#1916-1919)
StorageSlotUpgradeable.getAddressSlot(bytes32) (ProfileNFT.sol#1943-1947) uses assembly
        - INLINE ASM (ProfileNFT.sol#1944-1946)
StorageSlotUpgradeable.getBooleanSlot(bytes32) (ProfileNFT.sol#1949-1953) uses assembly
        - INLINE ASM (ProfileNFT.sol#1950-1952)
StorageSlotUpgradeable.getBytes32Slot(bytes32) (ProfileNFT.sol#1955-1959) uses assembly
        - INLINE ASM (ProfileNFT.sol#1956-1958)
StorageSlotUpgradeable.getUint256Slot(bytes32) (ProfileNFT.sol#1961-1965) uses assembly
        - INLINE ASM (ProfileNFT.sol#1962-1964)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#assembly-usage
```

```
Pragma version0.8.14 (ProfileNFT.sol#3) allows old versions
solc-0.8.14 is not recommended for deployment
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#incorrect-versions-of-solidity

Low level call in AddressUpgradeable.sendValue(address,uint256) (ProfileNFT.sol#1835-1840):
        - (success) = recipient.call{value: amount}() (ProfileNFT.sol#1838)
Low level call in AddressUpgradeable.functionCallWithValue(address,bytes,uint256,string) (ProfileNFT.sol#1862-1871):
        - (success,returndata) = target.call{value: value}(data) (ProfileNFT.sol#1869)
Low level call in AddressUpgradeable.functionStaticCall(address,bytes,string) (ProfileNFT.sol#1877-1884):
        - (success,returndata) = target.staticcall(data) (ProfileNFT.sol#1882)
Low level call in ERC1967UpgradeUpgradeable._functionDelegateCall(address,bytes) (ProfileNFT.sol#2201-2206):
        - (success,returndata) = target.delegatecall(data) (ProfileNFT.sol#2204)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#low-level-calls

Variable ProfileNFTStorage._nftDescriptor (ProfileNFT.sol#148) is not in mixedCase
Variable ProfileNFTStorage._namespaceOwner (ProfileNFT.sol#149) is not in mixedCase
Variable ProfileNFTStorage._profileById (ProfileNFT.sol#150) is not in mixedCase
Variable ProfileNFTStorage._profileIdByHandleHash (ProfileNFT.sol#151) is not in mixedCase
Variable ProfileNFTStorage._metadataById (ProfileNFT.sol#152) is not in mixedCase
Variable ProfileNFTStorage._operatorApproval (ProfileNFT.sol#153) is not in mixedCase
Variable ProfileNFTStorage._addressToPrimaryProfile (ProfileNFT.sol#154) is not in mixedCase
Variable ProfileNFTStorage._subscribeByProfileId (ProfileNFT.sol#155-156) is not in mixedCase
Variable ProfileNFTStorage._essenceByIdByProfileId (ProfileNFT.sol#157-158) is not in mixedCase
Function EIP712.DOMAIN_SEPARATOR() (ProfileNFT.sol#366-377) is not in mixedCase
Function ERC721.__ERC721_Init(string,string) (ProfileNFT.sol#491-497) is not in mixedCase
Parameter ERC721.__ERC721_Init(string,string)._name (ProfileNFT.sol#491) is not in mixedCase
Parameter ERC721.__ERC721_Init(string,string)._symbol (ProfileNFT.sol#491) is not in mixedCase
Variable ERC721._ownerOf (ProfileNFT.sol#465) is not in mixedCase
Variable ERC721._balanceOf (ProfileNFT.sol#467) is not in mixedCase
Variable CyberNFTBase._currentIndex (ProfileNFT.sol#680) is not in mixedCase
Variable CyberNFTBase._burnCount (ProfileNFT.sol#681) is not in mixedCase
Parameter Actions.collect(DataTypes.CollectData,mapping(uint256 => mapping(uint256 => DataTypes.EssenceStruct)))._essenceByIdByP
rofileId (ProfileNFT.sol#877-878) is not in mixedCase
Parameter Actions.registerEssence(DataTypes.RegisterEssenceData,address,mapping(uint256 => DataTypes.ProfileStruct),mapping(uint
256 => mapping(uint256 => DataTypes.EssenceStruct)))._profileById (ProfileNFT.sol#900) is not in mixedCase
```

```
Parameter Actions.setOperatorApproval(uint256,address,bool,mapping(uint256 => mapping(address => bool)))._operatorApproval (Prof
ileNFT.sol#1085) is not in mixedCase
Function ReentrancyGuard.__ReentrancyGuard_init() (ProfileNFT.sol#1991-1993) is not in mixedCase
Function ERC1967UpgradeUpgradeable.__ERC1967Upgrade_init() (ProfileNFT.sol#2101-2102) is not in mixedCase
Function ERC1967UpgradeUpgradeable.__ERC1967Upgrade_init_unchained() (ProfileNFT.sol#2104-2105) is not in mixedCase
Variable ERC1967UpgradeUpgradeable.__gap (ProfileNFT.sol#2208) is not in mixedCase
Function UUPSUpgradeable.__UUPSUpgradeable_init() (ProfileNFT.sol#2211-2212) is not in mixedCase
Function UUPSUpgradeable.__UUPSUpgradeable_init_unchained() (ProfileNFT.sol#2214-2215) is not in mixedCase
Variable UUPSUpgradeable.__self (ProfileNFT.sol#2216) is not in mixedCase
Variable UUPSUpgradeable.__gap (ProfileNFT.sol#2245) is not in mixedCase
Parameter ProfileNFT.initialize(address,string,string)._owner (ProfileNFT.sol#2330) is not in mixedCase
Variable ProfileNFT.SUBSCRIBE_BEACON (ProfileNFT.sol#2268) is not in mixedCase
Variable ProfileNFT.ESSENCE_BEACON (ProfileNFT.sol#2269) is not in mixedCase
Variable ProfileNFT.ENGINE (ProfileNFT.sol#2270) is not in mixedCase
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#conformance-to-solidity-naming-conventions
ProfileNFT.sol analyzed (25 contracts with 84 detectors), 85 result(s) found
```

INSPECTOR LOVELY

Slither Log >> SubscribeNFT.sol

```
SubscribeNFT.initialize(uint256,string,string).name (SubscribeNFT.sol#1502) shadows:
        - ERC721.name (SubscribeNFT.sol#429) (state variable)
SubscribeNFT.initialize(uint256,string,string).symbol (SubscribeNFT.sol#1503) shadows:
        - ERC721.symbol (SubscribeNFT.sol#431) (state variable)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#local-variable-shadowing

EIP712._requiresExpectedSigner(bytes32,address,uint8,bytes32,bytes32,uint256) (SubscribeNFT.sol#357-374) uses timestamp for comp
arisons
        Dangerous comparisons:
        - require(bool,string)(deadline >= block.timestamp,DEADLINE_EXCEEDED) (SubscribeNFT.sol#365)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#block-timestamp

ERC721._safeMint(address,uint256,bytes) (SubscribeNFT.sol#615-620) is never used and should be removed
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#dead-code

Pragma version0.8.14 (SubscribeNFT.sol#3) allows old versions
solc-0.8.14 is not recommended for deployment
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#incorrect-versions-of-solidity

Variable SubscribeNFTStorage._profileId (SubscribeNFT.sol#8) is not in mixedCase
Function EIP712.DOMAIN_SEPARATOR() (SubscribeNFT.sol#340-351) is not in mixedCase
Function ERC721.__ERC721_Init(string,string) (SubscribeNFT.sol#465-471) is not in mixedCase
Parameter ERC721.__ERC721_Init(string,string)._name (SubscribeNFT.sol#465) is not in mixedCase
Parameter ERC721.__ERC721_Init(string,string)._symbol (SubscribeNFT.sol#465) is not in mixedCase
Variable ERC721._ownerOf (SubscribeNFT.sol#439) is not in mixedCase
Variable ERC721._balanceOf (SubscribeNFT.sol#441) is not in mixedCase
Variable CyberNFTBase._currentIndex (SubscribeNFT.sol#654) is not in mixedCase
Variable CyberNFTBase._burnCount (SubscribeNFT.sol#655) is not in mixedCase
Variable SubscribeNFT.PROFILE (SubscribeNFT.sol#1482) is not in mixedCase
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#conformance-to-solidity-naming-conventions
SubscribeNFT.sol analyzed (16 contracts with 84 detectors), 16 result(s) found
```

Slither Log >> Auth.sol

```
Auth.setOwner(address).newOwner (Auth.sol#83) lacks a zero-check on :
                - owner = newOwner (Auth.sol#84)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#missing-zero-address-validation

Auth.__Auth_Init(address,Authority) (Auth.sol#51-57) is never used and should be removed
Initializable._disableInitializers() (Auth.sol#32-38) is never used and should be removed
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#dead-code

Pragma version>=0.8.0 (Auth.sol#3) allows old versions
solc-0.8.14 is not recommended for deployment
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#incorrect-versions-of-solidity

Function Auth.__Auth_Init(address,Authority) (Auth.sol#51-57) is not in mixedCase
Parameter Auth.__Auth_Init(address,Authority)._owner (Auth.sol#51) is not in mixedCase
Parameter Auth.__Auth_Init(address,Authority)._authority (Auth.sol#51) is not in mixedCase
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#conformance-to-solidity-naming-conventions
Auth.sol analyzed (3 contracts with 84 detectors), 8 result(s) found
```

Slither Log >> ERC721.sol

```
ERC721.__ERC721_Init(string,string) (ERC721.sol#96-102) is never used and should be removed
ERC721._burn(uint256) (ERC721.sol#214-229) is never used and should be removed
ERC721._mint(address,uint256) (ERC721.sol#199-212) is never used and should be removed
ERC721._safeMint(address,uint256) (ERC721.sol#235-244) is never used and should be removed
ERC721._safeMint(address,uint256,bytes) (ERC721.sol#246-259) is never used and should be removed
Initializable._disableInitializers() (ERC721.sol#32-38) is never used and should be removed
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#dead-code

Pragma version>=0.8.0 (ERC721.sol#3) allows old versions
solc-0.8.14 is not recommended for deployment
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#incorrect-versions-of-solidity

Function ERC721.__ERC721_Init(string,string) (ERC721.sol#96-102) is not in mixedCase
Parameter ERC721.__ERC721_Init(string,string)._name (ERC721.sol#96) is not in mixedCase
Parameter ERC721.__ERC721_Init(string,string)._symbol (ERC721.sol#96) is not in mixedCase
Variable ERC721._ownerOf (ERC721.sol#70) is not in mixedCase
Variable ERC721._balanceOf (ERC721.sol#72) is not in mixedCase
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#conformance-to-solidity-naming-conventions

ERC721 (ERC721.sol#45-260) does not implement functions:
        - ERC721.tokenURI(uint256) (ERC721.sol#64)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#unimplemented-functions
ERC721.sol analyzed (3 contracts with 84 detectors), 14 result(s) found
```

```
Initializable._disableInitializers() (Owned.sol#36-42) is never used and should be removed
Owned.__Owned_Init(address) (Owned.sol#69-74) is never used and should be removed
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#dead-code

Pragma version>=0.8.0 (Owned.sol#4) allows old versions
solc-0.8.14 is not recommended for deployment
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#incorrect-versions-of-solidity

Function Owned.__Owned_Init(address) (Owned.sol#69-74) is not in mixedCase
Parameter Owned.__Owned_Init(address)._owner (Owned.sol#69) is not in mixedCase
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#conformance-to-solidity-naming-conventions
Owned.sol analyzed (2 contracts with 84 detectors), 6 result(s) found
```

Slither Log >> RolesAuthority.sol

```
Auth.setOwner(address).newOwner (RolesAuthority.sol#82) lacks a zero-check on :
                - owner = newOwner (RolesAuthority.sol#83)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#missing-zero-address-validation

Initializable._disableInitializers() (RolesAuthority.sol#31-37) is never used and should be removed
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#dead-code

Pragma version>=0.8.0 (RolesAuthority.sol#3) allows old versions
solc-0.8.14 is not recommended for deployment
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#incorrect-versions-of-solidity

Function Auth.__Auth_Init(address,Authority) (RolesAuthority.sol#50-56) is not in mixedCase
Parameter Auth.__Auth_Init(address,Authority)._owner (RolesAuthority.sol#50) is not in mixedCase
Parameter Auth.__Auth_Init(address,Authority)._authority (RolesAuthority.sol#50) is not in mixedCase
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#conformance-to-solidity-naming-conventions
RolesAuthority.sol analyzed (4 contracts with 84 detectors), 7 result(s) found
```

Slither Log >> Create2Deployer.sol

```
Create2Deployer.deploy(bytes,bytes32) (Create2Deployer.sol#7-19) uses assembly
        - INLINE ASM (Create2Deployer.sol#10-15)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#assembly-usage

Pragma version0.8.14 (Create2Deployer.sol#2) allows old versions
solc-0.8.14 is not recommended for deployment
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#incorrect-versions-of-solidity
Create2Deployer.sol analyzed (1 contracts with 84 detectors), 3 result(s) found
```

Slither Log >> EssenceDeployer.sol

```
EssenceNFT.initialize(uint256,uint256,string,string,bool).name (EssenceDeployer.sol#1001) shadows:
        - ERC721.name (EssenceDeployer.sol#420) (state variable)
EssenceNFT.initialize(uint256,uint256,string,string,bool).symbol (EssenceDeployer.sol#1002) shadows:
        - ERC721.symbol (EssenceDeployer.sol#430) (state variable)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#local-variable-shadowing

EIP712._requiresExpectedSigner(bytes32,address,uint8,bytes32,bytes32,uint256) (EssenceDeployer.sol#356-373) uses timestamp for c
omparisons
        Dangerous comparisons:
        - require(bool,string)(deadline >= block.timestamp,DEADLINE_EXCEEDED) (EssenceDeployer.sol#364)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#block-timestamp

ERC721._safeMint(address,uint256,bytes) (EssenceDeployer.sol#514-527) is never used and should be removed
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#dead-code

solc-0.8.14 is not recommended for deployment
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#incorrect-versions-of-solidity

Variable EssenceNFTStorage._profileId (EssenceDeployer.sol#7) is not in mixedCase
Variable EssenceNFTStorage._essenceId (EssenceDeployer.sol#8) is not in mixedCase
Variable EssenceNFTStorage._transferable (EssenceDeployer.sol#9) is not in mixedCase
Function EIP712.DOMAIN_SEPARATOR() (EssenceDeployer.sol#338-350) is not in mixedCase
Function ERC721.__ERC721_Init(string,string) (EssenceDeployer.sol#464-470) is not in mixedCase
Parameter ERC721.__ERC721_Init(string,string)._name (EssenceDeployer.sol#464) is not in mixedCase
Parameter ERC721.__ERC721_Init(string,string)._symbol (EssenceDeployer.sol#464) is not in mixedCase
Variable ERC721._ownerOf (EssenceDeployer.sol#438) is not in mixedCase
Variable ERC721._balanceOf (EssenceDeployer.sol#440) is not in mixedCase
Variable CyberNFTBase._currentIndex (EssenceDeployer.sol#653) is not in mixedCase
Variable CyberNFTBase._burnCount (EssenceDeployer.sol#654) is not in mixedCase
Variable EssenceNFT.PROFILE (EssenceDeployer.sol#1000) is not in mixedCase
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#conformance-to-solidity-naming-conventions
EssenceDeployer.sol analyzed (16 contracts with 84 detectors), 17 result(s) found
```

INSPECTOR LOVELY

Slither Log >> ProfileDeployer.sol

```
ProfileNFT.initialize(address,string,string).name (ProfileDeployer.sol#2323) shadows:
        - ERC721.name (ProfileDeployer.sol#459) (state variable)
ProfileNFT.initialize(address,string,string).symbol (ProfileDeployer.sol#2324) shadows:
        - ERC721.symbol (ProfileDeployer.sol#461) (state variable)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#local-variable-shadowing

Variable 'ERC1967Upgrade_upgradeToAndCallUUPS(address,bytes,bool).slot (ProfileDeployer.sol#2128)' in ERC1967Upgrade
Upgradeable._upgradeToAndCallUUPS(address,bytes,bool) (ProfileDeployer.sol#2120-2135) potentially used before declaration: requi
re(bool,string)(slot == _IMPLEMENTATION_SLOT,ERC1967Upgrade: unsupported proxiableUUID) (ProfileDeployer.sol#2129)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#pre-declaration-usage-of-local-variables

EIP712._requiresExpectedSigner(bytes32,address,uint8,bytes32,bytes32,uint256) (ProfileDeployer.sol#387-404) uses timestamp for c
omparisons
        Dangerous comparisons:
        - require(bool,string)(deadline >= block.timestamp,DEADLINE_EXCEEDED) (ProfileDeployer.sol#395)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#block-timestamp

Parameter Actions.createProfilePostProcess(DataTypes.CreateProfileParams,bytes,DataTypes.CreateProfilePostProcessData,mapping(ui
nt256 => DataTypes.ProfileStruct),mapping(uint256 => string),mapping(bytes32 => uint256),mapping(address => uint256),mapping(uin
t256 => mapping(address => bool)))._profileById (ProfileDeployer.sol#1033) is not in mixedCase
Parameter Actions.createProfilePostProcess(DataTypes.CreateProfileParams,bytes,DataTypes.CreateProfilePostProcessData,mapping(ui
nt256 => DataTypes.ProfileStruct),mapping(uint256 => string),mapping(bytes32 => uint256),mapping(address => uint256),mapping(uin
t256 => mapping(address => bool)))._metadataById (ProfileDeployer.sol#1034) is not in mixedCase
Parameter Actions.createProfilePostProcess(DataTypes.CreateProfileParams,bytes,DataTypes.CreateProfilePostProcessData,mapping(ui
nt256 => DataTypes.ProfileStruct),mapping(uint256 => string),mapping(bytes32 => uint256),mapping(address => uint256),mapping(uin
t256 => mapping(address => bool)))._profileIdByHandleHash (ProfileDeployer.sol#1035) is not in mixedCase
Parameter Actions.setPrimaryProfile(address,uint256,mapping(address => uint256))._addressToPrimaryProfile (ProfileDeployer.sol#1
058) is not in mixedCase
Parameter Actions.setOperatorApproval(uint256,address,bool,mapping(uint256 => mapping(address => bool)))._operatorApproval (Prof
ileDeployer.sol#1068) is not in mixedCase
Function ReentrancyGuard.__ReentrancyGuard_init() (ProfileDeployer.sol#1974-1976) is not in mixedCase
Function ERC1967UpgradeUpgradeable.__ERC1967Upgrade_init() (ProfileDeployer.sol#2084-2085) is not in mixedCase
Function ERC1967UpgradeUpgradeable.__ERC1967Upgrade_init_unchained() (ProfileDeployer.sol#2087-2088) is not in mixedCase
Variable ERC1967UpgradeUpgradeable.__gap (ProfileDeployer.sol#2191) is not in mixedCase
Function UUPSUpgradeable.__UUPSUpgradeable_init() (ProfileDeployer.sol#2194-2195) is not in mixedCase
Function UUPSUpgradeable.__UUPSUpgradeable_init_unchained() (ProfileDeployer.sol#2197-2198) is not in mixedCase
Variable UUPSUpgradeable.__self (ProfileDeployer.sol#2199) is not in mixedCase
Variable UUPSUpgradeable.__gap (ProfileDeployer.sol#2228) is not in mixedCase
Parameter ProfileNFT.initialize(address,string,string)._owner (ProfileDeployer.sol#2322) is not in mixedCase
Variable ProfileNFT.SUBSCRIBE_BEACON (ProfileDeployer.sol#2251) is not in mixedCase
Variable ProfileNFT.ESSENCE_BEACON (ProfileDeployer.sol#2252) is not in mixedCase
Variable ProfileNFT.ENGINE (ProfileDeployer.sol#2253) is not in mixedCase
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#conformance-to-solidity-naming-conventions
ProfileDeployer.sol analyzed (26 contracts with 84 detectors), 80 result(s) found
```

Slither Log >> SubscribeDeployer.sol

```
SubscribeNFT.initialize(uint256,string,string).name (SubscribeDeployer.sol#1503) shadows:
        - ERC721.name (SubscribeDeployer.sol#429) (state variable)
SubscribeNFT.initialize(uint256,string,string).symbol (SubscribeDeployer.sol#1504) shadows:
        - ERC721.symbol (SubscribeDeployer.sol#431) (state variable)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#local-variable-shadowing

EIP712._requiresExpectedSigner(bytes32,address,uint8,bytes32,bytes32,uint256) (SubscribeDeployer.sol#357-374) uses timestamp for
 comparisons
        Dangerous comparisons:
        - require(bool,string)(deadline >= block.timestamp,DEADLINE_EXCEEDED) (SubscribeDeployer.sol#365)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#block-timestamp

ERC721._safeMint(address,uint256,bytes) (SubscribeDeployer.sol#615-628) is never used and should be removed
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#dead-code

Pragma version0.8.14 (SubscribeDeployer.sol#3) allows old versions
solc-0.8.14 is not recommended for deployment
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#incorrect-versions-of-solidity

Variable SubscribeNFTStorage._profileId (SubscribeDeployer.sol#8) is not in mixedCase
Function EIP712.DOMAIN_SEPARATOR() (SubscribeDeployer.sol#340-351) is not in mixedCase
Function ERC721.__ERC721_Init(string,string) (SubscribeDeployer.sol#465-471) is not in mixedCase
Parameter ERC721.__ERC721_Init(string,string)._name (SubscribeDeployer.sol#465) is not in mixedCase
Parameter ERC721.__ERC721_Init(string,string)._symbol (SubscribeDeployer.sol#465) is not in mixedCase
Variable ERC721._ownerOf (SubscribeDeployer.sol#439) is not in mixedCase
Variable ERC721._balanceOf (SubscribeDeployer.sol#441) is not in mixedCase
Variable CyberNFTBase._currentIndex (SubscribeDeployer.sol#654) is not in mixedCase
Variable CyberNFTBase._burnCount (SubscribeDeployer.sol#655) is not in mixedCase
Variable SubscribeNFT.PROFILE (SubscribeDeployer.sol#1483) is not in mixedCase
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#conformance-to-solidity-naming-conventions
SubscribeDeployer.sol analyzed (17 contracts with 84 detectors), 16 result(s) found
```

INSPECTOR LOVELY

Slither Log >> FeeMw.sol

```
FeeMw._currencyAllowed(address) (FeeMw.sol#222-224) is never used and should be removed
FeeMw._treasuryAddress() (FeeMw.sol#220-228) is never used and should be removed
FeeMw._treasuryFee() (FeeMw.sol#230-232) is never used and should be removed
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#dead-code

Pragma version0.8.14 (FeeMw.sol#3) allows old versions
solc-0.8.14 is not recommended for deployment
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#incorrect-versions-of-solidity

Variable FeeMw.TREASURY (FeeMw.sol#207) is not in mixedCase
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#conformance-to-solidity-naming-conventions
FeeMw.sol analyzed (4 contracts with 84 detectors), 6 result(s) found
```

Slither Log >> PermissionedMw.sol

```
Auth.setOwner(address).newOwner (PermissionedMw.sol#82) lacks a zero-check on :
        - owner = newOwner (PermissionedMw.sol#83)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#missing-zero-address-validation

Initializable._disableInitializers() (PermissionedMw.sol#31-37) is never used and should be removed
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#dead-code

Pragma version>=0.8.0 (PermissionedMw.sol#3) allows old versions
solc-0.8.14 is not recommended for deployment
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#incorrect-versions-of-solidity

Function Auth.__Auth_Init(address,Authority) (PermissionedMw.sol#50-56) is not in mixedCase
Parameter Auth.__Auth_Init(address,Authority)._owner (PermissionedMw.sol#50) is not in mixedCase
Parameter Auth.__Auth_Init(address,Authority)._authority (PermissionedMw.sol#50) is not in mixedCase
Variable PermissionedMw.ENGINE (PermissionedMw.sol#591) is not in mixedCase
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#conformance-to-solidity-naming-conventions
PermissionedMw.sol analyzed (9 contracts with 84 detectors), 8 result(s) found
```

```
Pragma version0.8.14 (CollectDisallowedMw.sol#3) allows old versions
solc-0.8.14 is not recommended for deployment
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#incorrect-versions-of-solidity
CollectDisallowedMw.sol analyzed (2 contracts with 84 detectors), 2 result(s) found
```

Slither Log >> CollectFlexPaidMw.sol

```
CollectFlexPaidMw.constructor(address,address).namespace (CollectFlexPaidMw.sol#625) lacks a zero-check on :
        - _namespace = namespace (CollectFlexPaidMw.sol#626)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#missing-zero-address-validation

Reentrancy in CollectFlexPaidMw.preProcess(uint256,uint256,address,address,bytes) (CollectFlexPaidMw.sol#654-694):
        External calls:
        - IERC20(currency).safeTransferFrom(collector,_mwStorage[profileId][essenceId].recipient,amount - treasuryCollected) (Co
llectFlexPaidMw.sol#671-675)
        - IERC20(currency).safeTransferFrom(collector,_treasuryAddress(),treasuryCollected) (CollectFlexPaidMw.sol#678-682)
        Event emitted after the call(s):
        - CollectFlexPaidMwPreprocessed(profileId,essenceId,collector,_mwStorage[profileId][essenceId].recipient,currency,amount
,metadataId) (CollectFlexPaidMw.sol#685-693)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#reentrancy-vulnerabilities-3

Address.verifyCallResult(bool,bytes,string) (CollectFlexPaidMw.sol#467-485) uses assembly
        - INLINE ASM (CollectFlexPaidMw.sol#477-480)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#assembly-usage
```

```
Pragma version0.8.14 (CollectFlexPaidMw.sol#3) allows old versions
solc-0.8.14 is not recommended for deployment
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#incorrect-versions-of-solidity

Low level call in Address.sendValue(address,uint256) (CollectFlexPaidMw.sol#409-414):
        - (success) = recipient.call{value: amount}() (CollectFlexPaidMw.sol#412)
Low level call in Address.functionCallWithValue(address,bytes,uint256,string) (CollectFlexPaidMw.sol#436-447):
        - (success,returndata) = target.call{value: value}(data) (CollectFlexPaidMw.sol#445)
Low level call in Address.functionStaticCall(address,bytes,string) (CollectFlexPaidMw.sol#453-462):
        - (success,returndata) = target.staticcall(data) (CollectFlexPaidMw.sol#460)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#low-level-calls

Variable FeeMw.TREASURY (CollectFlexPaidMw.sol#207) is not in mixedCase
Variable CollectFlexPaidMw._mwStorage (CollectFlexPaidMw.sol#618) is not in mixedCase
Variable CollectFlexPaidMw._namespace (CollectFlexPaidMw.sol#619) is not in mixedCase
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#conformance-to-solidity-naming-conventions

CollectFlexPaidMw._namespace (CollectFlexPaidMw.sol#619) should be immutable
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#state-variables-that-could-be-declared-immutable
CollectFlexPaidMw.sol analyzed (10 contracts with 84 detectors), 24 result(s) found
```

INSPECTOR LOVELY

Slither Log >> CollectLimitedTimePaidMw.sol

```
CollectLimitedTimePaidMw.constructor(address,address).namespace (CollectLimitedTimePaidMw.sol#1304) lacks a zero-check on :
                - _namespace = namespace (CollectLimitedTimePaidMw.sol#1305)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#missing-zero-address-validation

CollectLimitedTimePaidMw.preProcess(uint256,uint256,address,address,bytes) (CollectLimitedTimePaidMw.sol#1381-1440) uses timesta
mp for comparisons
        Dangerous comparisons:
        - require(bool,string)(block.timestamp >= _data[profileId][essenceId].startTimestamp,NOT_STARTED) (CollectLimitedTimePai
dMw.sol#1395-1398)
        - require(bool,string)(block.timestamp <= _data[profileId][essenceId].endTimestamp,ENDED) (CollectLimitedTimePaidMw.sol#
1400-1403)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#block-timestamp

Address.verifyCallResult(bool,bytes,string) (CollectLimitedTimePaidMw.sol#1130-1156) uses assembly
        - INLINE ASM (CollectLimitedTimePaidMw.sol#1148-1151)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#assembly-usage

CollectLimitedTimePaidMw.preProcess(uint256,uint256,address,address,bytes) (CollectLimitedTimePaidMw.sol#1381-1440) compares to
a boolean constant:
        -_data[profileId][essenceId].profileRequired == true (CollectLimitedTimePaidMw.sol#1412)
CollectLimitedTimePaidMw.preProcess(uint256,uint256,address,address,bytes) (CollectLimitedTimePaidMw.sol#1381-1440) compares to
a boolean constant:
        -_data[profileId][essenceId].subscribeRequired == true (CollectLimitedTimePaidMw.sol#1405)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#boolean-equality
```

Slither Log >> CollectMerkleDropMw.sol

```
MerkleProof._efficientHash(bytes32,bytes32) (CollectMerkleDropMw.sol#255-262) uses assembly
        - INLINE ASM (CollectMerkleDropMw.sol#257-261)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#assembly-usage

MerkleProof.multiProofVerify(bytes32[],bool[],bytes32,bytes32[]) (CollectMerkleDropMw.sol#105-112) is never used and should be r
emoved
MerkleProof.multiProofVerifyCalldata(bytes32[],bool[],bytes32,bytes32[]) (CollectMerkleDropMw.sol#119-126) is never used and sho
uld be removed
MerkleProof.processMultiProof(bytes32[],bool[],bytes32[]) (CollectMerkleDropMw.sol#138-187) is never used and should be removed
MerkleProof.processMultiProofCalldata(bytes32[],bool[],bytes32[]) (CollectMerkleDropMw.sol#194-243) is never used and should be
removed
MerkleProof.processProofCalldata(bytes32[],bytes32) (CollectMerkleDropMw.sol#91-97) is never used and should be removed
MerkleProof.verifyCalldata(bytes32[],bytes32,bytes32) (CollectMerkleDropMw.sol#70-72) is never used and should be removed
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#dead-code

Pragma version0.8.14 (CollectMerkleDropMw.sol#3) allows old versions
solc-0.8.14 is not recommended for deployment
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#incorrect-versions-of-solidity
CollectMerkleDropMw.sol analyzed (3 contracts with 84 detectors), 9 result(s) found
```

```
ERC721._ERC721_Init(string,string) (CollectOnlySubscribedMw.sol#95-101) is never used and should be removed
ERC721._burn(uint256) (CollectOnlySubscribedMw.sol#213-228) is never used and should be removed
ERC721._mint(address,uint256) (CollectOnlySubscribedMw.sol#198-211) is never used and should be removed
ERC721._safeMint(address,uint256) (CollectOnlySubscribedMw.sol#234-243) is never used and should be removed
ERC721._safeMint(address,uint256,bytes) (CollectOnlySubscribedMw.sol#245-250) is never used and should be removed
Initializable._disableInitializers() (CollectOnlySubscribedMw.sol#31-37) is never used and should be removed
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#dead-code

Pragma version>=0.8.0 (CollectOnlySubscribedMw.sol#3) allows old versions
solc-0.8.14 is not recommended for deployment
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#incorrect-versions-of-solidity

Function ERC721._ERC721_Init(string,string) (CollectOnlySubscribedMw.sol#95-101) is not in mixedCase
Parameter ERC721._ERC721_Init(string,string)._name (CollectOnlySubscribedMw.sol#95) is not in mixedCase
Parameter ERC721._ERC721_Init(string,string)._symbol (CollectOnlySubscribedMw.sol#95) is not in mixedCase
Variable ERC721._ownerOf (CollectOnlySubscribedMw.sol#69) is not in mixedCase
Variable ERC721._balanceOf (CollectOnlySubscribedMw.sol#71) is not in mixedCase
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#conformance-to-solidity-naming-conventions

ERC721 (CollectOnlySubscribedMw.sol#44-259) does not implement functions:
        - ERC721.tokenURI(uint256) (CollectOnlySubscribedMw.sol#63)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#unimplemented-functions
CollectOnlySubscribedMw.sol analyzed (8 contracts with 84 detectors), 14 result(s) found
```

INSPECTOR LOVELY

Slither Log >> CollectPaidMw.sol

```
Auth.setOwner(address).newOwner (CollectPaidMw.sol#313) lacks a zero-check on :
                - owner = newOwner (CollectPaidMw.sol#314)
CollectPaidMw.constructor(address,address).namespace (CollectPaidMw.sol#2177) lacks a zero-check on :
                - _namespace = namespace (CollectPaidMw.sol#2178)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#missing-zero-address-validation

Variable 'ERC721._checkOnERC721Received(address,address,uint256,bytes).retval (CollectPaidMw.sol#2104)' in ERC721._checkOnERC721
Received(address,address,uint256,bytes) (CollectPaidMw.sol#2102-2119) potentially used before declaration: retval != IERC721Rece
iver.onERC721Received.selector (CollectPaidMw.sol#2105)
Variable 'ERC721._checkOnERC721Received(address,address,uint256,bytes).reason (CollectPaidMw.sol#2108)' in ERC721._checkOnERC721
Received(address,address,uint256,bytes) (CollectPaidMw.sol#2102-2119) potentially used before declaration: reason.length == 0 (C
ollectPaidMw.sol#2109)
Variable 'ERC721._checkOnERC721Received(address,address,uint256,bytes).reason (CollectPaidMw.sol#2108)' in ERC721._checkOnERC721
Received(address,address,uint256,bytes) (CollectPaidMw.sol#2102-2119) potentially used before declaration: revert(uint256,uint25
6)(32 + reason,mload(uint256)(reason)) (CollectPaidMw.sol#2114)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#pre-declaration-usage-of-local-variables

Address.isContract(address) (CollectPaidMw.sol#1460-1467) uses assembly
        - INLINE ASM (CollectPaidMw.sol#1463-1465)
Address._functionCallWithValue(address,bytes,uint256,string) (CollectPaidMw.sol#1506-1528) uses assembly
        - INLINE ASM (CollectPaidMw.sol#1520-1523)
ERC721._checkOnERC721Received(address,address,uint256,bytes) (CollectPaidMw.sol#2102-2119) uses assembly
        - INLINE ASM (CollectPaidMw.sol#2113-2115)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#assembly-usage

Pragma version0.8.14 (CollectPaidMw.sol#3) allows old versions
solc-0.8.14 is not recommended for deployment
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#incorrect-versions-of-solidity

Low level call in Address.sendValue(address,uint256) (CollectPaidMw.sol#1460-1474):
        - (success) = recipient.call{value: amount}() (CollectPaidMw.sol#1472)
Low level call in Address._functionCallWithValue(address,bytes,uint256,string) (CollectPaidMw.sol#1506-1528):
        - (success,returndata) = target.call{value: weiValue}(data) (CollectPaidMw.sol#1514)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#low-level-calls

Variable FeeMw.TREASURY (CollectPaidMw.sol#207) is not in mixedCase
Function Auth.__Auth_Init(address,Authority) (CollectPaidMw.sol#281-287) is not in mixedCase
Parameter Auth.__Auth_Init(address,Authority)._owner (CollectPaidMw.sol#281) is not in mixedCase
Parameter Auth.__Auth_Init(address,Authority)._authority (CollectPaidMw.sol#281) is not in mixedCase
Variable CollectPaidMw._paidEssenceData (CollectPaidMw.sol#2169-2170) is not in mixedCase
Variable CollectPaidMw._namespace (CollectPaidMw.sol#2171) is not in mixedCase
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#conformance-to-solidity-naming-conventions

Redundant expression "this (CollectPaidMw.sol#1645)" inContext (CollectPaidMw.sol#1639-1648)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#redundant-statements

CollectPaidMw._namespace (CollectPaidMw.sol#2171) should be immutable
ERC721._name (CollectPaidMw.sol#1660) should be immutable
ERC721._symbol (CollectPaidMw.sol#1663) should be immutable
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#state-variables-that-could-be-declared-immutable
CollectPaidMw.sol analyzed (28 contracts with 84 detectors), 46 result(s) found
```

Slither Log >> CollectPermissionMw.sol

```
Variable 'ERC721._checkOnERC721Received(address,address,uint256,bytes).retval (CollectPermissionMw.sol#931)' in ERC721._checkOnE
RC721Received(address,address,uint256,bytes) (CollectPermissionMw.sol#929-946) potentially used before declaration: retval != IE
RC721Receiver.onERC721Received.selector (CollectPermissionMw.sol#932)
Variable 'ERC721._checkOnERC721Received(address,address,uint256,bytes).reason (CollectPermissionMw.sol#935)' in ERC721._checkOnE
RC721Received(address,address,uint256,bytes) (CollectPermissionMw.sol#929-946) potentially used before declaration: reason.lengt
h == 0 (CollectPermissionMw.sol#936)
Variable 'ERC721._checkOnERC721Received(address,address,uint256,bytes).reason (CollectPermissionMw.sol#935)' in ERC721._checkOnE
RC721Received(address,address,uint256,bytes) (CollectPermissionMw.sol#929-946) potentially used before declaration: revert(uint2
56,uint256)(32 + reason,mload(uint256)(reason)) (CollectPermissionMw.sol#941)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#pre-declaration-usage-of-local-variables

EIP712._requiresExpectedSigner(bytes32,address,uint8,bytes32,bytes32,uint256) (CollectPermissionMw.sol#178-195) uses timestamp f
or comparisons
        Dangerous comparisons:
        - require(bool,string)(deadline >= block.timestamp,DEADLINE_EXCEEDED) (CollectPermissionMw.sol#186)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#block-timestamp

Address.isContract(address) (CollectPermissionMw.sol#286-293) uses assembly
        - INLINE ASM (CollectPermissionMw.sol#289-291)
Address._functionCallWithValue(address,bytes,uint256,string) (CollectPermissionMw.sol#332-354) uses assembly
        - INLINE ASM (CollectPermissionMw.sol#346-349)
ERC721._checkOnERC721Received(address,address,uint256,bytes) (CollectPermissionMw.sol#929-946) uses assembly
        - INLINE ASM (CollectPermissionMw.sol#940-942)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#assembly-usage
```

```
Pragma version0.8.14 (CollectPermissionMw.sol#3) allows old versions
solc-0.8.14 is not recommended for deployment
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#incorrect-versions-of-solidity

Low level call in Address.sendValue(address,uint256) (CollectPermissionMw.sol#295-300):
        - (success) = recipient.call{value: amount}() (CollectPermissionMw.sol#298)
Low level call in Address._functionCallWithValue(address,bytes,uint256,string) (CollectPermissionMw.sol#332-354):
        - (success,returndata) = target.call{value: weiValue}(data) (CollectPermissionMw.sol#340)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#low-level-calls

Function EIP712.DOMAIN_SEPARATOR() (CollectPermissionMw.sol#161-172) is not in mixedCase
Variable CollectPermissionMw._signerStorage (CollectPermissionMw.sol#981-982) is not in mixedCase
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#conformance-to-solidity-naming-conventions

Redundant expression "this (CollectPermissionMw.sol#472)" inContext (CollectPermissionMw.sol#466-475)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#redundant-statements

ERC721._name (CollectPermissionMw.sol#487) should be immutable
ERC721._symbol (CollectPermissionMw.sol#490) should be immutable
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#state-variables-that-could-be-declared-immutable
CollectPermissionMw.sol analyzed (16 contracts with 84 detectors), 38 result(s) found
```

Slither Log >> CollectPermissionPaidMw.sol

```
CollectPermissionPaidMw.constructor(address,address).namespace (CollectPermissionPaidMw.sol#725) lacks a zero-check on :
                _namespace = namespace (CollectPermissionPaidMw.sol#726)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#missing-zero-address-validation

EIP712._requiresExpectedSigner(bytes32,address,uint8,bytes32,bytes32,uint256) (CollectPermissionPaidMw.sol#274-291) uses timesta
mp for comparisons
        Dangerous comparisons:
        - require(bool,string)(deadline >= block.timestamp,DEADLINE_EXCEEDED) (CollectPermissionPaidMw.sol#282)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#block-timestamp

Address.verifyCallResult(bool,bytes,string) (CollectPermissionPaidMw.sol#550-577) uses assembly
        - INLINE ASM (CollectPermissionPaidMw.sol#569-572)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#assembly-usage
```

```
Pragma version0.8.14 (CollectPermissionPaidMw.sol#3) allows old versions
solc-0.8.14 is not recommended for deployment
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#incorrect-versions-of-solidity

Low level call in Address.sendValue(address,uint256) (CollectPermissionPaidMw.sol#501-506):
        - (success) = recipient.call{value: amount}() (CollectPermissionPaidMw.sol#504)
Low level call in Address.functionCallWithValue(address,bytes,uint256,string) (CollectPermissionPaidMw.sol#528-539):
        - (success,returndata) = target.call{value: value}(data) (CollectPermissionPaidMw.sol#537)
Low level call in Address.functionStaticCall(address,bytes,string) (CollectPermissionPaidMw.sol#545-554):
        - (success,returndata) = target.staticcall(data) (CollectPermissionPaidMw.sol#552)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#low-level-calls

Variable FeeMw.TREASURY (CollectPermissionPaidMw.sol#207) is not in mixedCase
Function EIP712.DOMAIN_SEPARATOR() (CollectPermissionPaidMw.sol#257-268) is not in mixedCase
Variable CollectPermissionPaidMw._signerStorage (CollectPermissionPaidMw.sol#717-718) is not in mixedCase
Variable CollectPermissionPaidMw._namespace (CollectPermissionPaidMw.sol#719) is not in mixedCase
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#conformance-to-solidity-naming-conventions

CollectPermissionPaidMw._namespace (CollectPermissionPaidMw.sol#719) should be immutable
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#state-variables-that-could-be-declared-immutable
CollectPermissionPaidMw.sol analyzed (11 contracts with 84 detectors), 28 result(s) found
```

INSPECTOR LOVELY

Slither Log >> FeeCreationMw.sol

```
Auth.setOwner(address).newOwner (FeeCreationMw.sol#84) lacks a zero-check on :
                - owner = newOwner (FeeCreationMw.sol#85)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#missing-zero-address-validation

Address.verifyCallResult(bool,bytes,string) (FeeCreationMw.sol#812-830) uses assembly
        - INLINE ASM (FeeCreationMw.sol#822-825)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#assembly-usage

Address.functionCall(address,bytes) (FeeCreationMw.sol#761-763) is never used and should be removed
Address.functionCall(address,bytes,string) (FeeCreationMw.sol#765-771) is never used and should be removed
Address.functionCallWithValue(address,bytes,uint256) (FeeCreationMw.sol#773-779) is never used and should be removed
Address.functionCallWithValue(address,bytes,uint256,string) (FeeCreationMw.sol#781-792) is never used and should be removed
Address.functionStaticCall(address,bytes) (FeeCreationMw.sol#794-796) is never used and should be removed
Address.functionStaticCall(address,bytes,string) (FeeCreationMw.sol#798-807) is never used and should be removed
Address.isContract(address) (FeeCreationMw.sol#749-752) is never used and should be removed
Address.verifyCallResult(bool,bytes,string) (FeeCreationMw.sol#812-830) is never used and should be removed
Initializable._disableInitializers() (FeeCreationMw.sol#33-39) is never used and should be removed
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#dead-code

Pragma version>=0.8.0 (FeeCreationMw.sol#3) allows old versions
solc-0.8.14 is not recommended for deployment
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#incorrect-versions-of-solidity

Low level call in Address.sendValue(address,uint256) (FeeCreationMw.sol#754-759):
        - (success) = recipient.call{value: amount}() (FeeCreationMw.sol#757)
Low level call in Address.functionCallWithValue(address,bytes,uint256,string) (FeeCreationMw.sol#781-792):
        - (success,returndata) = target.call{value: value}(data) (FeeCreationMw.sol#790)
Low level call in Address.functionStaticCall(address,bytes,string) (FeeCreationMw.sol#798-807):
        - (success,returndata) = target.staticcall(data) (FeeCreationMw.sol#805)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#low-level-calls

Function Auth.__Auth_Init(address,Authority) (FeeCreationMw.sol#52-58) is not in mixedCase
Parameter Auth.__Auth_Init(address,Authority)._owner (FeeCreationMw.sol#52) is not in mixedCase
Parameter Auth.__Auth_Init(address,Authority)._authority (FeeCreationMw.sol#52) is not in mixedCase
Variable PermissionedMw.ENGINE (FeeCreationMw.sol#506) is not in mixedCase
Variable FeeCreationMw._mwDataByNamespace (FeeCreationMw.sol#865) is not in mixedCase
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#conformance-to-solidity-naming-conventions
FeeCreationMw.sol analyzed (13 contracts with 84 detectors), 21 result(s) found
```

Slither Log >> PermissionedFeeCreationMw.sol

```
Auth.setOwner(address).newOwner (PermissionedFeeCreationMw.sol#313) lacks a zero-check on :
                - owner = newOwner (PermissionedFeeCreationMw.sol#314)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#missing-zero-address-validation

EIP712._requiresExpectedSigner(bytes32,address,uint8,bytes32,bytes32,uint256) (PermissionedFeeCreationMw.sol#754-771) uses times
tamp for comparisons
        Dangerous comparisons:
        - require(bool,string)(deadline >= block.timestamp,DEADLINE_EXCEEDED) (PermissionedFeeCreationMw.sol#762)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#block-timestamp

Address.verifyCallResult(bool,bytes,string) (PermissionedFeeCreationMw.sol#1001-1019) uses assembly
        - INLINE ASM (PermissionedFeeCreationMw.sol#1011-1014)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#assembly-usage

Address.functionCall(address,bytes) (PermissionedFeeCreationMw.sol#950-952) is never used and should be removed
Address.functionCall(address,bytes,string) (PermissionedFeeCreationMw.sol#954-960) is never used and should be removed
Address.functionCallWithValue(address,bytes,uint256) (PermissionedFeeCreationMw.sol#962-968) is never used and should be removed
Address.functionCallWithValue(address,bytes,uint256,string) (PermissionedFeeCreationMw.sol#970-981) is never used and should be
removed
Address.functionStaticCall(address,bytes) (PermissionedFeeCreationMw.sol#983-985) is never used and should be removed
Address.functionStaticCall(address,bytes,string) (PermissionedFeeCreationMw.sol#987-996) is never used and should be removed
Address.isContract(address) (PermissionedFeeCreationMw.sol#938-941) is never used and should be removed
Address.verifyCallResult(bool,bytes,string) (PermissionedFeeCreationMw.sol#1001-1019) is never used and should be removed
EIP712._requiresExpectedSigner(bytes32,address,DataTypes.EIP712Signature) (PermissionedFeeCreationMw.sol#773-786) is never used
and should be removed
FeeMw._currencyAllowed(address) (PermissionedFeeCreationMw.sol#222-224) is never used and should be removed
Initializable._disableInitializers() (PermissionedFeeCreationMw.sol#262-268) is never used and should be removed
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#dead-code
```

```
Low level call in Address.sendValue(address,uint256) (PermissionedFeeCreationMw.sol#943-948):
        - (success) = recipient.call{value: amount}() (PermissionedFeeCreationMw.sol#946)
Low level call in Address.functionCallWithValue(address,bytes,uint256,string) (PermissionedFeeCreationMw.sol#970-981):
        - (success,returndata) = target.call{value: value}(data) (PermissionedFeeCreationMw.sol#979)
Low level call in Address.functionStaticCall(address,bytes,string) (PermissionedFeeCreationMw.sol#987-996):
        - (success,returndata) = target.staticcall(data) (PermissionedFeeCreationMw.sol#994)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#low-level-calls

Variable FeeMw.TREASURY (PermissionedFeeCreationMw.sol#207) is not in mixedCase
Function Auth.__Auth_Init(address,Authority) (PermissionedFeeCreationMw.sol#281-287) is not in mixedCase
Parameter Auth.__Auth_Init(address,Authority)._owner (PermissionedFeeCreationMw.sol#281) is not in mixedCase
Parameter Auth.__Auth_Init(address,Authority)._authority (PermissionedFeeCreationMw.sol#281) is not in mixedCase
Variable PermissionedMw.ENGINE (PermissionedFeeCreationMw.sol#689) is not in mixedCase
Function EIP712.DOMAIN_SEPARATOR() (PermissionedFeeCreationMw.sol#737-748) is not in mixedCase
Variable PermissionedFeeCreationMw._mwDataByNamespace (PermissionedFeeCreationMw.sol#1061) is not in mixedCase
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#conformance-to-solidity-naming-conventions
PermissionedFeeCreationMw.sol analyzed (17 contracts with 84 detectors), 26 result(s) found
```

Slither Log >> StableFeeCreationMw.sol

```
Auth.setOwner(address).newOwner (StableFeeCreationMw.sol#83) lacks a zero-check on :
                - owner = newOwner (StableFeeCreationMw.sol#84)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#missing-zero-address-validation

EIP712._requiresExpectedSigner(bytes32,address,uint8,bytes32,bytes32,uint256) (StableFeeCreationMw.sol#658-675) uses timestamp f
or comparisons
        Dangerous comparisons:
        - require(bool,string)(deadline >= block.timestamp,DEADLINE_EXCEEDED) (StableFeeCreationMw.sol#666)
StableFeeCreationMw._checkExpectedSigner(bytes32,address,uint8,bytes32,bytes32,uint256) (StableFeeCreationMw.sol#1251-1271) uses
 timestamp for comparisons
        Dangerous comparisons:
        - deadline < block.timestamp (StableFeeCreationMw.sol#1259)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#block-timestamp

Address.verifyCallResult(bool,bytes,string) (StableFeeCreationMw.sol#923-941) uses assembly
        - INLINE ASM (StableFeeCreationMw.sol#933-936)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#assembly-usage
```

```
Pragma version0.8.14 (StableFeeCreationMw.sol#3) allows old versions
solc-0.8.14 is not recommended for deployment
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#incorrect-versions-of-solidity

Low level call in Address.sendValue(address,uint256) (StableFeeCreationMw.sol#865-870):
        - (success) = recipient.call{value: amount}() (StableFeeCreationMw.sol#868)
Low level call in Address.functionCallWithValue(address,bytes,uint256,string) (StableFeeCreationMw.sol#892-903):
        - (success,returndata) = target.call{value: value}(data) (StableFeeCreationMw.sol#901)
Low level call in Address.functionStaticCall(address,bytes,string) (StableFeeCreationMw.sol#909-918):
        - (success,returndata) = target.staticcall(data) (StableFeeCreationMw.sol#916)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#low-level-calls

Function Auth.__Auth_Init(address,Authority) (StableFeeCreationMw.sol#51-57) is not in mixedCase
Parameter Auth.__Auth_Init(address,Authority)._owner (StableFeeCreationMw.sol#51) is not in mixedCase
Parameter Auth.__Auth_Init(address,Authority)._authority (StableFeeCreationMw.sol#51) is not in mixedCase
Variable PermissionedMw.ENGINE (StableFeeCreationMw.sol#595) is not in mixedCase
Function EIP712.DOMAIN_SEPARATOR() (StableFeeCreationMw.sol#641-652) is not in mixedCase
Variable StableFeeCreationMw._mwDataByNamespace (StableFeeCreationMw.sol#992) is not in mixedCase
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#conformance-to-solidity-naming-conventions
StableFeeCreationMw.sol analyzed (15 contracts with 84 detectors), 26 result(s) found
```

Slither Log >> SubscribeDisallowedMw.sol

```
Pragma version0.8.14 (SubscribeDisallowedMw.sol#3) allows old versions
solc-0.8.14 is not recommended for deployment
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#incorrect-versions-of-solidity
SubscribeDisallowedMw.sol analyzed (2 contracts with 84 detectors), 2 result(s) found
```

INSPECTOR LOVELY

Slither Log >> SubscribeOnlyOnceMw.sol

```
Pragma version0.8.14 (SubscribeOnlyOnceMw.sol#3) allows old versions
solc-0.8.14 is not recommended for deployment
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#incorrect-versions-of-solidity

Low level call in Address.sendValue(address,uint256) (SubscribeOnlyOnceMw.sol#57-62):
        - (success) = recipient.call{value: amount}() (SubscribeOnlyOnceMw.sol#60)
Low level call in Address._functionCallWithValue(address,bytes,uint256,string) (SubscribeOnlyOnceMw.sol#94-116):
        - (success,returndata) = target.call{value: weiValue}(data) (SubscribeOnlyOnceMw.sol#102)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#low-level-calls

Redundant expression "this (SubscribeOnlyOnceMw.sol#250)" inContext (SubscribeOnlyOnceMw.sol#244-253)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#redundant-statements

ERC721._name (SubscribeOnlyOnceMw.sol#265) should be immutable
ERC721._symbol (SubscribeOnlyOnceMw.sol#268) should be immutable
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#state-variables-that-could-be-declared-immutable
SubscribeOnlyOnceMw.sol analyzed (14 contracts with 84 detectors), 35 result(s) found
```

Slither Log >> SubscribePaidMw.sol

```
Auth.setOwner(address).newOwner (SubscribePaidMw.sol#314) lacks a zero-check on :
                - owner = newOwner (SubscribePaidMw.sol#315)
SubscribePaidMw.constructor(address,address).namespace (SubscribePaidMw.sol#1654) lacks a zero-check on :
                - _namespace = namespace (SubscribePaidMw.sol#1655)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#missing-zero-address-validation

Variable 'ERC721._checkOnERC721Received(address,address,uint256,bytes).retval (SubscribePaidMw.sol#1586)' in ERC721._checkOnERC7
21Received(address,address,uint256,bytes) (SubscribePaidMw.sol#1584-1601) potentially used before declaration: retval != IERC721
Receiver.onERC721Received.selector (SubscribePaidMw.sol#1587)
Variable 'ERC721._checkOnERC721Received(address,address,uint256,bytes).reason (SubscribePaidMw.sol#1590)' in ERC721._checkOnERC7
21Received(address,address,uint256,bytes) (SubscribePaidMw.sol#1584-1601) potentially used before declaration: reason.length ==
0 (SubscribePaidMw.sol#1591)
Variable 'ERC721._checkOnERC721Received(address,address,uint256,bytes).reason (SubscribePaidMw.sol#1590)' in ERC721._checkOnERC7
21Received(address,address,uint256,bytes) (SubscribePaidMw.sol#1584-1601) potentially used before declaration: revert(uint256,ui
nt256)(32 + reason,mload(uint256)(reason)) (SubscribePaidMw.sol#1596)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#pre-declaration-usage-of-local-variables

Address.verifyCallResult(bool,bytes,string) (SubscribePaidMw.sol#908-926) uses assembly
        - INLINE ASM (SubscribePaidMw.sol#918-921)
ERC721._checkOnERC721Received(address,address,uint256,bytes) (SubscribePaidMw.sol#1584-1601) uses assembly
        - INLINE ASM (SubscribePaidMw.sol#1595-1597)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#assembly-usage
```

```
Pragma version0.8.14 (SubscribePaidMw.sol#3) allows old versions
solc-0.8.14 is not recommended for deployment
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#incorrect-versions-of-solidity

Low level call in Address.sendValue(address,uint256) (SubscribePaidMw.sol#850-855):
        - (success) = recipient.call{value: amount}() (SubscribePaidMw.sol#853)
Low level call in Address.functionCallWithValue(address,bytes,uint256,string) (SubscribePaidMw.sol#877-888):
        - (success,returndata) = target.call{value: value}(data) (SubscribePaidMw.sol#886)
Low level call in Address.functionStaticCall(address,bytes,string) (SubscribePaidMw.sol#894-903):
        - (success,returndata) = target.staticcall(data) (SubscribePaidMw.sol#901)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#low-level-calls

Variable FeeMw.TREASURY (SubscribePaidMw.sol#207) is not in mixedCase
Function Auth.__Auth_Init(address,Authority) (SubscribePaidMw.sol#282-288) is not in mixedCase
Parameter Auth.__Auth_Init(address,Authority)._owner (SubscribePaidMw.sol#282) is not in mixedCase
Parameter Auth.__Auth_Init(address,Authority)._authority (SubscribePaidMw.sol#282) is not in mixedCase
Variable SubscribePaidMw._paidSubscribeData (SubscribePaidMw.sol#1647) is not in mixedCase
Variable SubscribePaidMw._namespace (SubscribePaidMw.sol#1648) is not in mixedCase
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#conformance-to-solidity-naming-conventions

Redundant expression "this (SubscribePaidMw.sol#1127)" inContext (SubscribePaidMw.sol#1121-1130)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#redundant-statements

ERC721._name (SubscribePaidMw.sol#1142) should be immutable
ERC721._symbol (SubscribePaidMw.sol#1145) should be immutable
SubscribePaidMw._namespace (SubscribePaidMw.sol#1648) should be immutable
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#state-variables-that-could-be-declared-immutable
SubscribePaidMw.sol analyzed (28 contracts with 84 detectors), 47 result(s) found
```

INSPECTOR LOVELY

Slither Log >> CyberBoxNFT.sol

```
CyberBoxNFT.initialize(address,address,string,string).owner (CyberBoxNFT.sol#1296) shadows:
        - Owned.owner (CyberBoxNFT.sol#790) (state variable)
CyberBoxNFT.initialize(address,address,string,string).name (CyberBoxNFT.sol#1298) shadows:
        - ERC721.name (CyberBoxNFT.sol#444) (state variable)
CyberBoxNFT.initialize(address,address,string,string).symbol (CyberBoxNFT.sol#1299) shadows:
        - ERC721.symbol (CyberBoxNFT.sol#446) (state variable)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#local-variable-shadowing

CyberBoxNFT.initialize(address,address,string,string).signer (CyberBoxNFT.sol#1297) lacks a zero-check on :
                - _signer = signer (CyberBoxNFT.sol#1301)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#missing-zero-address-validation

Variable 'ERC1967UpgradeUpgradeable._upgradeToAndCallUUPS(address,bytes,bool).slot (CyberBoxNFT.sol#1161)' in ERC1967UpgradeUpgr
adeable._upgradeToAndCallUUPS(address,bytes,bool) (CyberBoxNFT.sol#1153-1168) potentially used before declaration: require(bool,
string)(slot == _IMPLEMENTATION_SLOT,ERC1967Upgrade: unsupported proxiableUUID) (CyberBoxNFT.sol#1162)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#pre-declaration-usage-of-local-variables
```

```
Pragma version0.8.14 (CyberBoxNFT.sol#3) allows old versions
solc-0.8.14 is not recommended for deployment
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#incorrect-versions-of-solidity

Low level call in AddressUpgradeable.sendValue(address,uint256) (CyberBoxNFT.sol#979-984):
        - (success) = recipient.call{value: amount}() (CyberBoxNFT.sol#982)
Low level call in AddressUpgradeable.functionCallWithValue(address,bytes,uint256,string) (CyberBoxNFT.sol#1006-1015):
        - (success,returndata) = target.call{value: value}(data) (CyberBoxNFT.sol#1013)
Low level call in AddressUpgradeable.functionStaticCall(address,bytes,string) (CyberBoxNFT.sol#1021-1028):
        - (success,returndata) = target.staticcall(data) (CyberBoxNFT.sol#1026)
Low level call in ERC1967UpgradeUpgradeable._functionDelegateCall(address,bytes) (CyberBoxNFT.sol#1217-1222):
        - (success,returndata) = target.delegatecall(data) (CyberBoxNFT.sol#1220)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#low-level-calls

Variable CyberBoxNFTStorage._signer (CyberBoxNFT.sol#11) is not in mixedCase
Function EIP712.DOMAIN_SEPARATOR() (CyberBoxNFT.sol#355-366) is not in mixedCase
Function ERC721.__ERC721_Init(string,string) (CyberBoxNFT.sol#480-486) is not in mixedCase
Parameter ERC721.__ERC721_Init(string,string)._name (CyberBoxNFT.sol#480) is not in mixedCase
Parameter ERC721.__ERC721_Init(string,string)._symbol (CyberBoxNFT.sol#480) is not in mixedCase
Variable ERC721._ownerOf (CyberBoxNFT.sol#454) is not in mixedCase
Variable ERC721._balanceOf (CyberBoxNFT.sol#456) is not in mixedCase
Variable CyberNFTBase._currentIndex (CyberBoxNFT.sol#669) is not in mixedCase
Variable CyberNFTBase._burnCount (CyberBoxNFT.sol#670) is not in mixedCase
Function Owned.__Owned_Init(address) (CyberBoxNFT.sol#810-815) is not in mixedCase
Parameter Owned.__Owned_Init(address)._owner (CyberBoxNFT.sol#810) is not in mixedCase
Function ERC1967UpgradeUpgradeable.__ERC1967Upgrade_init() (CyberBoxNFT.sol#1117-1118) is not in mixedCase
Function ERC1967UpgradeUpgradeable.__ERC1967Upgrade_init_unchained() (CyberBoxNFT.sol#1120-1121) is not in mixedCase
Variable ERC1967UpgradeUpgradeable.__gap (CyberBoxNFT.sol#1224) is not in mixedCase
Function UUPSUpgradeable.__UUPSUpgradeable_init() (CyberBoxNFT.sol#1227-1228) is not in mixedCase
Function UUPSUpgradeable.__UUPSUpgradeable_init_unchained() (CyberBoxNFT.sol#1230-1231) is not in mixedCase
Variable UUPSUpgradeable.__self (CyberBoxNFT.sol#1232) is not in mixedCase
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#conformance-to-solidity-naming-conventions
CyberBoxNFT.sol analyzed (21 contracts with 84 detectors), 59 result(s) found
```

Slither Log >> CyberGrandNFT.sol

```
CyberGrandNFT.initialize(address,address,string,string,string).owner (CyberGrandNFT.sol#1296) shadows:
        - Owned.owner (CyberGrandNFT.sol#880) (state variable)
CyberGrandNFT.initialize(address,address,string,string,string).name (CyberGrandNFT.sol#1298) shadows:
        - ERC721.name (CyberGrandNFT.sol#445) (state variable)
CyberGrandNFT.initialize(address,address,string,string,string).symbol (CyberGrandNFT.sol#1299) shadows:
        - ERC721.symbol (CyberGrandNFT.sol#447) (state variable)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#local-variable-shadowing

CyberGrandNFT.initialize(address,address,string,string,string).signer (CyberGrandNFT.sol#1297) lacks a zero-check on :
                - _signer = signer (CyberGrandNFT.sol#1302)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#missing-zero-address-validation

Variable 'ERC1967UpgradeUpgradeable._upgradeToAndCallUUPS(address,bytes,bool).slot (CyberGrandNFT.sol#1160)' in ERC1967UpgradeUp
gradeable._upgradeToAndCallUUPS(address,bytes,bool) (CyberGrandNFT.sol#1152-1167) potentially used before declaration: require(b
ool,string)(slot == _IMPLEMENTATION_SLOT,ERC1967Upgrade: unsupported proxiableUUID) (CyberGrandNFT.sol#1161)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#pre-declaration-usage-of-local-variables
```

```
Pragma version0.8.14 (CyberGrandNFT.sol#3) allows old versions
solc-0.8.14 is not recommended for deployment
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#incorrect-versions-of-solidity

Low level call in AddressUpgradeable.sendValue(address,uint256) (CyberGrandNFT.sol#979-984):
        - (success) = recipient.call{value: amount}() (CyberGrandNFT.sol#982)
Low level call in AddressUpgradeable.functionCallWithValue(address,bytes,uint256,string) (CyberGrandNFT.sol#1006-1015):
        - (success,returndata) = target.call{value: value}(data) (CyberGrandNFT.sol#1013)
Low level call in AddressUpgradeable.functionStaticCall(address,bytes,string) (CyberGrandNFT.sol#1021-1028):
        - (success,returndata) = target.staticcall(data) (CyberGrandNFT.sol#1026)
Low level call in ERC1967UpgradeUpgradeable._functionDelegateCall(address,bytes) (CyberGrandNFT.sol#1216-1221):
        - (success,returndata) = target.delegatecall(data) (CyberGrandNFT.sol#1219)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#low-level-calls

Variable CyberGrandNFTStorage._signer (CyberGrandNFT.sol#10) is not in mixedCase
Variable CyberGrandNFTStorage._tokenURI (CyberGrandNFT.sol#11) is not in mixedCase
Function EIP712.DOMAIN_SEPARATOR() (CyberGrandNFT.sol#356-367) is not in mixedCase
Function ERC721._ERC721_Init(string,string) (CyberGrandNFT.sol#481-487) is not in mixedCase
Parameter ERC721._ERC721_Init(string,string)._name (CyberGrandNFT.sol#481) is not in mixedCase
Parameter ERC721._ERC721_Init(string,string)._symbol (CyberGrandNFT.sol#481) is not in mixedCase
Variable ERC721._ownerOf (CyberGrandNFT.sol#455) is not in mixedCase
Variable ERC721._balanceOf (CyberGrandNFT.sol#457) is not in mixedCase
Variable CyberNFTBase._currentIndex (CyberGrandNFT.sol#670) is not in mixedCase
Variable CyberNFTBase._burnCount (CyberGrandNFT.sol#671) is not in mixedCase
Function Owned._Owned_Init(address) (CyberGrandNFT.sol#892-897) is not in mixedCase
Parameter Owned._Owned_Init(address)._owner (CyberGrandNFT.sol#892) is not in mixedCase
Function ERC1967UpgradeUpgradeable._ERC1967Upgrade_init() (CyberGrandNFT.sol#1116-1117) is not in mixedCase
Function ERC1967UpgradeUpgradeable._ERC1967Upgrade_init_unchained() (CyberGrandNFT.sol#1119-1120) is not in mixedCase
Variable ERC1967UpgradeUpgradeable.__gap (CyberGrandNFT.sol#1223) is not in mixedCase
Function UUPSUpgradeable._UUPSUpgradeable_init() (CyberGrandNFT.sol#1226-1227) is not in mixedCase
Function UUPSUpgradeable._UUPSUpgradeable_init_unchained() (CyberGrandNFT.sol#1229-1230) is not in mixedCase
Variable UUPSUpgradeable.__self (CyberGrandNFT.sol#1231) is not in mixedCase
Variable UUPSUpgradeable.__gap (CyberGrandNFT.sol#1260) is not in mixedCase
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#conformance-to-solidity-naming-conventions
CyberGrandNFT.sol analyzed (21 contracts with 84 detectors), 61 result(s) found
```

Slither Log >> CyberVault.sol

```
CyberVault.constructor(address).owner (CyberVault.sol#965) shadows:
        - Owned.owner (CyberVault.sol#96) (state variable)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#local-variable-shadowing

CyberVault.constructor(address).owner (CyberVault.sol#965) lacks a zero-check on :
                - _signer = owner (CyberVault.sol#966)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#missing-zero-address-validation

Reentrancy in CyberVault.deposit(string,address,uint256) (CyberVault.sol#1133-1146):
        External calls:
        - IERC20(currency).safeTransferFrom(msg.sender,address(this),amount) (CyberVault.sol#1142)
        State variables written after the call(s):
        - _balanceByProfileByCurrency[profileId][currency] += amount (CyberVault.sol#1144)
Reentrancy in CyberVault.deposit1155(string,address,uint256,uint256) (CyberVault.sol#1178-1200):
        External calls:
        - IERC1155(currency).safeTransferFrom(msg.sender,address(this),tokenId,amount,) (CyberVault.sol#1188-1194)
        State variables written after the call(s):
        - _balanceByProfileByCurrencyByTokenID[profileId][currency][tokenId] += amount (CyberVault.sol#1196-1198)
Reentrancy in CyberVault.deposit721(string,address,uint256) (CyberVault.sol#1155-1168):
        External calls:
        - IERC721(currency).safeTransferFrom(msg.sender,address(this),tokenId) (CyberVault.sol#1164)
        State variables written after the call(s):
        - _balanceByProfileByCurrencyByTokenID[profileId][currency][tokenId] = 1 (CyberVault.sol#1166)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#reentrancy-vulnerabilities-2

Pragma version0.8.14 (CyberVault.sol#3) allows old versions
solc-0.8.14 is not recommended for deployment
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#incorrect-versions-of-solidity

Low level call in Address.sendValue(address,uint256) (CyberVault.sol#693-698):
        - (success) = recipient.call{value: amount}() (CyberVault.sol#696)
Low level call in Address.functionCallWithValue(address,bytes,uint256,string) (CyberVault.sol#720-731):
        - (success,returndata) = target.call{value: value}(data) (CyberVault.sol#729)
Low level call in Address.functionStaticCall(address,bytes,string) (CyberVault.sol#737-746):
        - (success,returndata) = target.staticcall(data) (CyberVault.sol#744)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#low-level-calls

Function ReentrancyGuard.__ReentrancyGuard_init() (CyberVault.sol#50-60) is not in mixedCase
Function Owned._Owned_Init(address) (CyberVault.sol#108-113) is not in mixedCase
Parameter Owned._Owned_Init(address)._owner (CyberVault.sol#108) is not in mixedCase
Function EIP712.DOMAIN_SEPARATOR() (CyberVault.sol#282-293) is not in mixedCase
Variable CyberVault._signer (CyberVault.sol#956) is not in mixedCase
Variable CyberVault._balanceByProfileByCurrency (CyberVault.sol#958) is not in mixedCase
Variable CyberVault._balanceByProfileByCurrencyByTokenID (CyberVault.sol#959) is not in mixedCase
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#conformance-to-solidity-naming-conventions
CyberVault.sol analyzed (10 contracts with 84 detectors), 28 result(s) found
```

INSPECTOR LOVELY

Slither Log >> FrameNFT.sol

```
FrameNFT.constructor(string,address).uri (FrameNFT.sol#1546) shadows:
        - FrameNFT.uri(uint256) (FrameNFT.sol#1641-1652) (function)
        - ERC1155.uri(uint256) (FrameNFT.sol#1078-1080) (function)
        - IERC1155MetadataURI.uri(uint256) (FrameNFT.sol#1039) (function)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#local-variable-shadowing

Variable 'ERC1155._doSafeTransferAcceptanceCheck(address,address,address,uint256,uint256,bytes).response (FrameNFT.sol#1398)' in
 ERC1155._doSafeTransferAcceptanceCheck(address,address,address,uint256,uint256,bytes) (FrameNFT.sol#1389-1415) potentially used
 before declaration: response != IERC1155Receiver.onERC1155Received.selector (FrameNFT.sol#1399)
Variable 'ERC1155._doSafeTransferAcceptanceCheck(address,address,address,uint256,uint256,bytes).reason (FrameNFT.sol#1403)' in E
RC1155._doSafeTransferAcceptanceCheck(address,address,address,uint256,uint256,bytes) (FrameNFT.sol#1389-1415) potentially used b
efore declaration: reason.length == 0 (FrameNFT.sol#1404)
Variable 'ERC1155._doSafeTransferAcceptanceCheck(address,address,address,uint256,uint256,bytes).reason (FrameNFT.sol#1403)' in E
RC1155._doSafeTransferAcceptanceCheck(address,address,address,uint256,uint256,bytes) (FrameNFT.sol#1389-1415) potentially used b
efore declaration: revert(uint256,uint256)(32 + reason,mload(uint256)(reason)) (FrameNFT.sol#1410)
Variable 'ERC1155._doSafeTransferAcceptanceCheck(address,address,address,uint256,uint256,bytes).reason (FrameNFT.sol#1403)' in E
RC1155._doSafeTransferAcceptanceCheck(address,address,address,uint256,uint256,bytes) (FrameNFT.sol#1389-1415) potentially used b
efore declaration: revert(uint256,uint256)(32 + reason,mload(uint256)(reason)) (FrameNFT.sol#1410)
Variable 'ERC1155._doSafeBatchTransferAcceptanceCheck(address,address,address,uint256[],uint256[],bytes).response (FrameNFT.sol#
1431)' in ERC1155._doSafeBatchTransferAcceptanceCheck(address,address,address,uint256[],uint256[],bytes) (FrameNFT.sol#1421-1449
) potentially used before declaration: response != IERC1155Receiver.onERC1155BatchReceived.selector (FrameNFT.sol#1433)
Variable 'ERC1155._doSafeBatchTransferAcceptanceCheck(address,address,address,uint256[],uint256[],bytes).reason (FrameNFT.sol#14
37)' in ERC1155._doSafeBatchTransferAcceptanceCheck(address,address,address,uint256[],uint256[],bytes) (FrameNFT.sol#1421-1449)
potentially used before declaration: reason.length == 0 (FrameNFT.sol#1438)
Variable 'ERC1155._doSafeBatchTransferAcceptanceCheck(address,address,address,uint256[],uint256[],bytes).reason (FrameNFT.sol#14
37)' in ERC1155._doSafeBatchTransferAcceptanceCheck(address,address,address,uint256[],uint256[],bytes) (FrameNFT.sol#1421-1449)
potentially used before declaration: revert(uint256,uint256)(32 + reason,mload(uint256)(reason)) (FrameNFT.sol#1444)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#pre-declaration-usage-of-local-variables

Pragma version^0.8.0 (FrameNFT.sol#4) allows old versions
solc-0.8.14 is not recommended for deployment
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#incorrect-versions-of-solidity

Low level call in Address.sendValue(address,uint256) (FrameNFT.sol#915-920):
        - (success) = recipient.call{value: amount}() (FrameNFT.sol#918)
Low level call in Address._functionCallWithValue(address,bytes,uint256,string) (FrameNFT.sol#952-974):
        - (success,returndata) = target.call{value: weiValue}(data) (FrameNFT.sol#960)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#low-level-calls
FrameNFT.sol analyzed (20 contracts with 84 detectors), 59 result(s) found
```

Slither Log >> Link3ProfileDescriptor.sol

```
Pragma version0.8.14 (Link3ProfileDescriptor.sol#3) allows old versions
solc-0.8.14 is not recommended for deployment
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#incorrect-versions-of-solidity

Low level call in AddressUpgradeable.sendValue(address,uint256) (Link3ProfileDescriptor.sol#1033-1038):
        - (success) = recipient.call{value: amount}() (Link3ProfileDescriptor.sol#1036)
Low level call in AddressUpgradeable.functionCallWithValue(address,bytes,uint256,string) (Link3ProfileDescriptor.sol#1060-1069):
        - (success,returndata) = target.call{value: value}(data) (Link3ProfileDescriptor.sol#1067)
Low level call in AddressUpgradeable.functionStaticCall(address,bytes,string) (Link3ProfileDescriptor.sol#1075-1082):
        - (success,returndata) = target.staticcall(data) (Link3ProfileDescriptor.sol#1080)
Low level call in ERC1967UpgradeUpgradeable._functionDelegateCall(address,bytes) (Link3ProfileDescriptor.sol#1270-1275):
        - (success,returndata) = target.delegatecall(data) (Link3ProfileDescriptor.sol#1273)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#low-level-calls

Function Owned.__Owned_Init(address) (Link3ProfileDescriptor.sol#81-86) is not in mixedCase
Parameter Owned.__Owned_Init(address)._owner (Link3ProfileDescriptor.sol#81) is not in mixedCase
Function ERC1967UpgradeUpgradeable.__ERC1967Upgrade_init() (Link3ProfileDescriptor.sol#1170-1171) is not in mixedCase
Function ERC1967UpgradeUpgradeable.__ERC1967Upgrade_init_unchained() (Link3ProfileDescriptor.sol#1173-1174) is not in mixedCase
Variable ERC1967UpgradeUpgradeable.__gap (Link3ProfileDescriptor.sol#1277) is not in mixedCase
Function UUPSUpgradeable.__UUPSUpgradeable_init() (Link3ProfileDescriptor.sol#1280-1281) is not in mixedCase
Function UUPSUpgradeable.__UUPSUpgradeable_init_unchained() (Link3ProfileDescriptor.sol#1283-1284) is not in mixedCase
Variable UUPSUpgradeable.__self (Link3ProfileDescriptor.sol#1285) is not in mixedCase
Variable UUPSUpgradeable.__gap (Link3ProfileDescriptor.sol#1314) is not in mixedCase
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#conformance-to-solidity-naming-conventions
Link3ProfileDescriptor.sol analyzed (16 contracts with 84 detectors), 85 result(s) found
```

INSPECTOR LOVELY

Slither Log >> MBNFT.sol

```
MBNFT.initialize(address,address,string,string,string).owner (MBNFT.sol#1629) shadows:
        - Owned.owner (MBNFT.sol#1042) (state variable)
MBNFT.initialize(address,address,string,string,string).name (MBNFT.sol#1631) shadows:
        - ERC721.name (MBNFT.sol#488) (state variable)
MBNFT.initialize(address,address,string,string,string).symbol (MBNFT.sol#1632) shadows:
        - ERC721.symbol (MBNFT.sol#490) (state variable)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#local-variable-shadowing

MBNFT.initialize(address,address,string,string,string).boxAddr (MBNFT.sol#1630) lacks a zero-check on :
                - _boxAddr = boxAddr (MBNFT.sol#1635)
MBNFT.setBoxAddr(address).boxAddr (MBNFT.sol#1699) lacks a zero-check on :
                - _boxAddr = boxAddr (MBNFT.sol#1700)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#missing-zero-address-validation

Variable 'ERC1967UpgradeUpgradeable._upgradeToAndCallUUPS(address,bytes,bool).slot (MBNFT.sol#1492)' in ERC1967UpgradeUpgradeabl
e._upgradeToAndCallUUPS(address,bytes,bool) (MBNFT.sol#1484-1499) potentially used before declaration: require(bool,string)(slot
 == _IMPLEMENTATION_SLOT,ERC1967Upgrade: unsupported proxiableUUID) (MBNFT.sol#1493)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#pre-declaration-usage-of-local-variables

Reentrancy in CyberNFTBaseFlex._mintTo(address,uint256) (MBNFT.sol#923-926):
        External calls:
        - super._safeMint(_to,_id) (MBNFT.sol#924)
                - require(bool,string)(ERC721TokenReceiver(to).onERC721Received(msg.sender,address(0),id,) == ERC721TokenReceive
r.onERC721Received.selector,UNSAFE_RECIPIENT) (MBNFT.sol#667-671)
        State variables written after the call(s):
        - ++ _mintCount (MBNFT.sol#925)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#reentrancy-vulnerabilities-2
```

```
Pragma version0.8.14 (MBNFT.sol#3) allows old versions
solc-0.8.14 is not recommended for deployment
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#incorrect-versions-of-solidity

Low level call in AddressUpgradeable.sendValue(address,uint256) (MBNFT.sol#1311-1316):
        - (success) = recipient.call{value: amount}() (MBNFT.sol#1314)
Low level call in AddressUpgradeable.functionCallWithValue(address,bytes,uint256,string) (MBNFT.sol#1338-1347):
        - (success,returndata) = target.call{value: value}(data) (MBNFT.sol#1345)
Low level call in AddressUpgradeable.functionStaticCall(address,bytes,string) (MBNFT.sol#1353-1360):
        - (success,returndata) = target.staticcall(data) (MBNFT.sol#1358)
Low level call in ERC1967UpgradeUpgradeable._functionDelegateCall(address,bytes) (MBNFT.sol#1548-1553):
        - (success,returndata) = target.delegatecall(data) (MBNFT.sol#1551)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#low-level-calls
```

```
Variable MBNFTStorage._tokenURI (MBNFT.sol#53) is not in mixedCase
Variable MBNFTStorage._boxAddr (MBNFT.sol#54) is not in mixedCase
Function EIP712.DOMAIN_SEPARATOR() (MBNFT.sol#399-410) is not in mixedCase
Function ERC721.__ERC721_Init(string,string) (MBNFT.sol#524-530) is not in mixedCase
Parameter ERC721.__ERC721_Init(string,string)._name (MBNFT.sol#524) is not in mixedCase
Parameter ERC721.__ERC721_Init(string,string)._symbol (MBNFT.sol#524) is not in mixedCase
Variable ERC721._ownerOf (MBNFT.sol#498) is not in mixedCase
Variable ERC721._balanceOf (MBNFT.sol#500) is not in mixedCase
Variable CyberNFTBase._currentIndex (MBNFT.sol#713) is not in mixedCase
Variable CyberNFTBase._burnCount (MBNFT.sol#714) is not in mixedCase
Variable CyberNFTBaseFlex._mintCount (MBNFT.sol#832) is not in mixedCase
Variable CyberNFTBaseFlex._burnCount (MBNFT.sol#833) is not in mixedCase
Function Owned.__Owned_Init(address) (MBNFT.sol#1054-1059) is not in mixedCase
Parameter Owned.__Owned_Init(address)._owner (MBNFT.sol#1054) is not in mixedCase
Function ERC1967UpgradeUpgradeable.__ERC1967Upgrade_init() (MBNFT.sol#1440-1449) is not in mixedCase
Function ERC1967UpgradeUpgradeable.__ERC1967Upgrade_init_unchained() (MBNFT.sol#1451-1452) is not in mixedCase
Variable ERC1967UpgradeUpgradeable.__gap (MBNFT.sol#1555) is not in mixedCase
Function UUPSUpgradeable.__UUPSUpgradeable_init() (MBNFT.sol#1558-1559) is not in mixedCase
Function UUPSUpgradeable.__UUPSUpgradeable_init_unchained() (MBNFT.sol#1561-1562) is not in mixedCase
Variable UUPSUpgradeable.__self (MBNFT.sol#1563) is not in mixedCase
Variable UUPSUpgradeable.__gap (MBNFT.sol#1592) is not in mixedCase
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#conformance-to-solidity-naming-conventions

CyberNFTBase (MBNFT.sol#709-826) does not implement functions:
        - ERC721.tokenURI(uint256) (MBNFT.sol#492)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#unimplemented-functions
MBNFT.sol analyzed (23 contracts with 84 detectors), 79 result(s) found
```

INSPECTOR LOVELY

Slither Log >> MiniShardNFT.sol

```
MiniShardNFT.constructor(string,address).uri (MiniShardNFT.sol#1856) shadows:
        - ERC1155.uri(uint256) (MiniShardNFT.sol#1307-1309) (function)
        - IERC1155MetadataURI.uri(uint256) (MiniShardNFT.sol#1265) (function)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#local-variable-shadowing

MiniShardNFT.constructor(string,address).owner (MiniShardNFT.sol#1856) lacks a zero-check on :
                - _signer = owner (MiniShardNFT.sol#1857)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#missing-zero-address-validation
```

```
Pragma version0.8.14 (MiniShardNFT.sol#3) allows old versions
solc-0.8.14 is not recommended for deployment
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#incorrect-versions-of-solidity

Low level call in Address.sendValue(address,uint256) (MiniShardNFT.sol#1143-1148):
        - (success) = recipient.call{value: amount}() (MiniShardNFT.sol#1146)
Low level call in Address._functionCallWithValue(address,bytes,uint256,string) (MiniShardNFT.sol#1180-1202):
        - (success,returndata) = target.call{value: weiValue}(data) (MiniShardNFT.sol#1188)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#low-level-calls

Function EIP712.DOMAIN_SEPARATOR() (MiniShardNFT.sol#161-172) is not in mixedCase
Variable MiniShardNFT._signer (MiniShardNFT.sol#1839) is not in mixedCase
Variable MiniShardNFT._nonces (MiniShardNFT.sol#1840) is not in mixedCase
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#conformance-to-solidity-naming-conventions

MiniShardNFT._signer (MiniShardNFT.sol#1839) should be immutable
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#state-variables-that-could-be-declared-immutable
MiniShardNFT.sol analyzed (23 contracts with 84 detectors), 67 result(s) found
```

Slither Log >> RelationshipChecker.sol

```
RelationshipChecker.constructor(address).namespace (RelationshipChecker.sol#820) lacks a zero-check on :
                - _namespace = namespace (RelationshipChecker.sol#821)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#missing-zero-address-validation

Pragma version0.8.14 (RelationshipChecker.sol#3) allows old versions
solc-0.8.14 is not recommended for deployment
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#incorrect-versions-of-solidity

Variable RelationshipChecker._namespace (RelationshipChecker.sol#814) is not in mixedCase
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#conformance-to-solidity-naming-conventions

RelationshipChecker._namespace (RelationshipChecker.sol#814) should be immutable
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#state-variables-that-could-be-declared-immutable
RelationshipChecker.sol analyzed (6 contracts with 84 detectors), 5 result(s) found
```

Slither Log >> CyberBoxNFTStorage.sol

```
Pragma version0.8.14 (CyberBoxNFTStorage.sol#3) allows old versions
solc-0.8.14 is not recommended for deployment
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#incorrect-versions-of-solidity

Variable CyberBoxNFTStorage._signer (CyberBoxNFTStorage.sol#10) is not in mixedCase
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#conformance-to-solidity-naming-conventions

CyberBoxNFTStorage._signer (CyberBoxNFTStorage.sol#10) should be constant
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#state-variables-that-could-be-declared-constant
CyberBoxNFTStorage.sol analyzed (1 contracts with 84 detectors), 4 result(s) found
```

Slither Log >> CyberEngineStorage.sol

```
Pragma version0.8.14 (CyberEngineStorage.sol#3) allows old versions
solc-0.8.14 is not recommended for deployment
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#incorrect-versions-of-solidity

Variable CyberEngineStorage._profileMwAllowlist (CyberEngineStorage.sol#146) is not in mixedCase
Variable CyberEngineStorage._essenceMwAllowlist (CyberEngineStorage.sol#147) is not in mixedCase
Variable CyberEngineStorage._subscribeMwAllowlist (CyberEngineStorage.sol#148) is not in mixedCase
Variable CyberEngineStorage._namespaceByName (CyberEngineStorage.sol#149) is not in mixedCase
Variable CyberEngineStorage._namespaceInfo (CyberEngineStorage.sol#150) is not in mixedCase
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#conformance-to-solidity-naming-conventions
CyberEngineStorage.sol analyzed (2 contracts with 84 detectors), 7 result(s) found
```

INSPECTOR
LOVELY

Slither Log >> CyberGrandNFTStorage.sol

```
Pragma version0.8.14 (CyberGrandNFTStorage.sol#3) allows old versions
solc-0.8.14 is not recommended for deployment
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#incorrect-versions-of-solidity

Variable CyberGrandNFTStorage._signer (CyberGrandNFTStorage.sol#10) is not in mixedCase
Variable CyberGrandNFTStorage._tokenURI (CyberGrandNFTStorage.sol#11) is not in mixedCase
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#conformance-to-solidity-naming-conventions

CyberGrandNFTStorage._signer (CyberGrandNFTStorage.sol#10) should be constant
CyberGrandNFTStorage._tokenURI (CyberGrandNFTStorage.sol#11) should be constant
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#state-variables-that-could-be-declared-constant
CyberGrandNFTStorage.sol analyzed (1 contracts with 84 detectors), 6 result(s) found
```

Slither Log >> EssenceNFTStorage.sol

```
Pragma version0.8.14 (EssenceNFTStorage.sol#3) allows old versions
solc-0.8.14 is not recommended for deployment
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#incorrect-versions-of-solidity

Variable EssenceNFTStorage._profileId (EssenceNFTStorage.sol#10) is not in mixedCase
Variable EssenceNFTStorage._essenceId (EssenceNFTStorage.sol#11) is not in mixedCase
Variable EssenceNFTStorage._transferable (EssenceNFTStorage.sol#12) is not in mixedCase
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#conformance-to-solidity-naming-conventions

EssenceNFTStorage._essenceId (EssenceNFTStorage.sol#11) should be constant
EssenceNFTStorage._profileId (EssenceNFTStorage.sol#10) should be constant
EssenceNFTStorage._transferable (EssenceNFTStorage.sol#12) should be constant
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#state-variables-that-could-be-declared-constant
EssenceNFTStorage.sol analyzed (1 contracts with 84 detectors), 8 result(s) found
```

Slither Log >> Link3ProfileDescriptorStorage.sol

```
Pragma version0.8.14 (Link3ProfileDescriptorStorage.sol#3) allows old versions
solc-0.8.14 is not recommended for deployment
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#incorrect-versions-of-solidity

Link3ProfileDescriptorStorage.animationTemplate (Link3ProfileDescriptorStorage.sol#11) should be constant
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#state-variables-that-could-be-declared-constant
Link3ProfileDescriptorStorage.sol analyzed (1 contracts with 84 detectors), 3 result(s) found
```

Slither Log >> MBNFTStorage.sol

```
Pragma version0.8.14 (MBNFTStorage.sol#1) allows old versions
solc-0.8.14 is not recommended for deployment
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#incorrect-versions-of-solidity

Variable MBNFTStorage._tokenURI (MBNFTStorage.sol#10) is not in mixedCase
Variable MBNFTStorage._boxAddr (MBNFTStorage.sol#11) is not in mixedCase
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#conformance-to-solidity-naming-conventions

MBNFTStorage._boxAddr (MBNFTStorage.sol#11) should be constant
MBNFTStorage._tokenURI (MBNFTStorage.sol#10) should be constant
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#state-variables-that-could-be-declared-constant
MBNFTStorage.sol analyzed (1 contracts with 84 detectors), 6 result(s) found
```

INSPECTOR LOVELY

Slither Log >> ProfileNFTStorage.sol

```
Pragma version0.8.14 (ProfileNFTStorage.sol#3) allows old versions
solc-0.8.14 is not recommended for deployment
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#incorrect-versions-of-solidity

Variable ProfileNFTStorage._nftDescriptor (ProfileNFTStorage.sol#146) is not in mixedCase
Variable ProfileNFTStorage._namespaceOwner (ProfileNFTStorage.sol#147) is not in mixedCase
Variable ProfileNFTStorage._profileById (ProfileNFTStorage.sol#148) is not in mixedCase
Variable ProfileNFTStorage._profileIdByHandleHash (ProfileNFTStorage.sol#149) is not in mixedCase
Variable ProfileNFTStorage._metadataById (ProfileNFTStorage.sol#150) is not in mixedCase
Variable ProfileNFTStorage._operatorApproval (ProfileNFTStorage.sol#151) is not in mixedCase
Variable ProfileNFTStorage._addressToPrimaryProfile (ProfileNFTStorage.sol#152) is not in mixedCase
Variable ProfileNFTStorage._subscribeByProfileId (ProfileNFTStorage.sol#153-154) is not in mixedCase
Variable ProfileNFTStorage._essenceByIdByProfileId (ProfileNFTStorage.sol#155-156) is not in mixedCase
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#conformance-to-solidity-naming-conventions
```

Slither Log >> SubscribeNFTStorage.sol

```
Pragma version0.8.14 (SubscribeNFTStorage.sol#3) allows old versions
solc-0.8.14 is not recommended for deployment
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#incorrect-versions-of-solidity

Variable SubscribeNFTStorage._profileId (SubscribeNFTStorage.sol#7) is not in mixedCase
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#conformance-to-solidity-naming-conventions

SubscribeNFTStorage._profileId (SubscribeNFTStorage.sol#7) should be constant
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#state-variables-that-could-be-declared-constant
SubscribeNFTStorage.sol analyzed (1 contracts with 84 detectors), 4 result(s) found
```

Slither Log >> UpgradeableBeacon.sol

```
Address.isContract(address) (UpgradeableBeacon.sol#17-24) uses assembly
        - INLINE ASM (UpgradeableBeacon.sol#20-22)
Address._functionCallWithValue(address,bytes,uint256,string) (UpgradeableBeacon.sol#63-85) uses assembly
        - INLINE ASM (UpgradeableBeacon.sol#77-80)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#assembly-usage

Address._functionCallWithValue(address,bytes,uint256,string) (UpgradeableBeacon.sol#63-85) is never used and should be removed
Address.functionCall(address,bytes) (UpgradeableBeacon.sol#33-35) is never used and should be removed
Address.functionCall(address,bytes,string) (UpgradeableBeacon.sol#37-43) is never used and should be removed
Address.functionCallWithValue(address,bytes,uint256) (UpgradeableBeacon.sol#45-51) is never used and should be removed
Address.functionCallWithValue(address,bytes,uint256,string) (UpgradeableBeacon.sol#53-61) is never used and should be removed
Address.sendValue(address,uint256) (UpgradeableBeacon.sol#26-31) is never used and should be removed
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#dead-code

Pragma version^0.8.0 (UpgradeableBeacon.sol#4) allows old versions
solc-0.8.14 is not recommended for deployment
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#incorrect-versions-of-solidity

Low level call in Address.sendValue(address,uint256) (UpgradeableBeacon.sol#26-31):
        - (success) = recipient.call{value: amount}() (UpgradeableBeacon.sol#29)
Low level call in Address._functionCallWithValue(address,bytes,uint256,string) (UpgradeableBeacon.sol#63-85):
        - (success,returndata) = target.call{value: weiValue}(data) (UpgradeableBeacon.sol#71)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#low-level-calls

Variable UpgradeableBeacon.OWNER (UpgradeableBeacon.sol#99) is not in mixedCase
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#conformance-to-solidity-naming-conventions

Variable UpgradeableBeacon._implementation (UpgradeableBeacon.sol#98) is too similar to UpgradeableBeacon.constructor(address,address).implementation_ (UpgradeableBeacon.sol#110)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#variable-names-too-similar
UpgradeableBeacon.sol analyzed (3 contracts with 84 detectors), 14 result(s) found
```

INSPECTOR LOVELY

Slither Log >> CYBER.sol

```
ERC20Permit.permit(address,address,uint256,uint256,uint8,bytes32,bytes32) (CyberToken.sol#2507-2526) uses timestamp for compar
isons
        Dangerous comparisons:
        - require(bool,string)(block.timestamp <= deadline,ERC20Permit: expired deadline) (CyberToken.sol#2516)
ERC20Votes.delegateBySig(address,uint256,uint256,uint8,bytes32,bytes32) (CyberToken.sol#2675-2692) uses timestamp for comparis
ons
        Dangerous comparisons:
        - require(bool,string)(block.timestamp <= expiry,ERC20Votes: signature expired) (CyberToken.sol#2683)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#block-timestamp

Math.mulDiv(uint256,uint256,uint256) (CyberToken.sol#227-307) uses assembly
        - INLINE ASM (CyberToken.sol#238-242)
        - INLINE ASM (CyberToken.sol#258-265)
        - INLINE ASM (CyberToken.sol#272-281)
Strings.toString(uint256) (CyberToken.sol#526-546) uses assembly
        - INLINE ASM (CyberToken.sol#532-534)
        - INLINE ASM (CyberToken.sol#538-540)
ECDSA.tryRecover(bytes32,bytes) (CyberToken.sol#1700-1806) uses assembly
        - INLINE ASM (CyberToken.sol#1797-1801)
ERC20Votes._unsafeAccess(ERC20Votes.Checkpoint[],uint256) (CyberToken.sol#2798-2803) uses assembly
        - INLINE ASM (CyberToken.sol#2799-2802)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#assembly-usage

Context._msgData() (CyberToken.sol#2033-2035) is never used and should be removed
Counters.decrement(Counters.Counter) (CyberToken.sol#167-173) is never used and should be removed
Counters.reset(Counters.Counter) (CyberToken.sol#175-177) is never used and should be removed
ECDSA.recover(bytes32,bytes) (CyberToken.sol#1822-1826) is never used and should be removed
ECDSA.recover(bytes32,bytes32,bytes32) (CyberToken.sol#1850-1858) is never used and should be removed
SafeCast.toUint8(uint256) (CyberToken.sol#1150-1159) is never used and should be removed
SafeCast.toUint80(uint256) (CyberToken.sol#1003-1006) is never used and should be removed
SafeCast.toUint88(uint256) (CyberToken.sol#988-989) is never used and should be removed
SafeCast.toUint96(uint256) (CyberToken.sol#969-972) is never used and should be removed
Strings.toHexString(address) (CyberToken.sol#575-577) is never used and should be removed
Strings.toHexString(uint256) (CyberToken.sol#551-555) is never used and should be removed
Strings.toHexString(uint256,uint256) (CyberToken.sol#560-570) is never used and should be removed
Strings.toString(uint256) (CyberToken.sol#526-546) is never used and should be removed
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#dead-code

Pragma version^0.8.0 (CyberToken.sol#4) allows old versions
solc-0.8.0 is not recommended for deployment
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#incorrect-versions-of-solidity

Function IERC20Permit.DOMAIN_SEPARATOR() (CyberToken.sol#146) is not in mixedCase
Variable EIP712._CACHED_DOMAIN_SEPARATOR (CyberToken.sol#1953) is not in mixedCase
Variable EIP712._CACHED_CHAIN_ID (CyberToken.sol#1954) is not in mixedCase
Variable EIP712._CACHED_THIS (CyberToken.sol#1955) is not in mixedCase
Variable EIP712._HASHED_NAME (CyberToken.sol#1957) is not in mixedCase
Variable EIP712._HASHED_VERSION (CyberToken.sol#1958) is not in mixedCase
Variable EIP712._TYPE_HASH (CyberToken.sol#1959) is not in mixedCase
Function ERC20Permit.DOMAIN_SEPARATOR() (CyberToken.sol#2539-2541) is not in mixedCase
Variable ERC20Permit._PERMIT_TYPEHASH_DEPRECATED_SLOT (CyberToken.sol#2495) is not in mixedCase
Parameter CyberToken.mint(address,uint256)._account (CyberToken.sol#2829) is not in mixedCase
Parameter CyberToken.mint(address,uint256)._amount (CyberToken.sol#2829) is not in mixedCase
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#conformance-to-solidity-naming-conventions
CyberToken.sol analyzed (17 contracts with 84 detectors), 120 result(s) found
```

INSPECTOR LOVELY

SOLIDITY STATIC ANALYSIS

Static code analysis is used to identify many common coding problems before a program is released. It involves examining the code manually or using tools to automate the process. Static code analysis tools can automatically scan the code without executing it.

CyberNFTBaseFlex.sol

Block timestamp:

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

more

Pos: 52:28:

Constant/View/Pure functions:

CyberNFTBaseFlex._initialize(string,string) : Potentially should be constant/view/pure but is not. Note: Modifiers are currently not considered by this static analysis.

more

Pos: 108:4:

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: 220:12:

Delete from dynamic array:

Using "delete" on an array leaves a gap. The length of the array remains the same. If you want to remove the empty position you need to shift items manually and update the "length" property.
more
Pos: 192:8:

No return:

ICyberNFTBase.totalBurned(): Defines a return type but never explicitly returns a value.
Pos: 27:4:

EIP712.sol

Block timestamp:

Use of "block.timestamp": "block.timestamp" can be influenced by miners to a certain degree. That means that a miner can "choose" the block.timestamp, to a certain degree, to change the outcome of a transaction in the mined block.
more
Pos: 52:28:

Constant/View/Pure functions:

EIP712.DOMAIN_SEPARATOR() : Is constant but potentially should not be.
more
Pos: 27:4:

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: 60:8:

INSPECTOR LOVELY

CyberNFTBase.sol

Block timestamp:

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

more

Pos: 52:28:

Constant/View/Pure functions:

CyberNFTBase._initialize(string,string) : Potentially should be constant/view/pure but is not. Note: Modifiers are currently not considered by this static analysis.

more

Pos: 108:4:

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: 220:12:

Delete from dynamic array:

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

more

Pos: 192:8:

INSPECTOR LOVELY

CyberEngine.sol

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: 186:16:

Block timestamp:

Use of "block.timestamp": "block.timestamp" can be influenced by miners to a certain degree. That means that a miner can "choose" the block.timestamp, to a certain degree, to change the outcome of a transaction in the mined block.
more
Pos: 52:28:

Low level calls:

Use of "delegatecall": should be avoided whenever possible. External code, that is called can change the state of the calling contract and send ether from the caller's balance. If this is wanted behaviour, use the Solidity library feature if possible.
more
Pos: 173:50:

For loop over dynamic array:

Loops that do not have a fixed number of iterations, for example, loops that depend on storage values, have to be used carefully. Due to the block gas limit, transactions can only consume a certain amount of gas. The number of iterations in a loop can grow beyond the block gas limit which can cause the complete contract to be stalled at a certain point. Additionally, using unbounded loops incurs in a lot of avoidable gas costs. Carefully test how many items at maximum you can pass to such functions to make it successful.
more
Pos: 700:8:

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: 255:8:

EssenceNFT.sol

Check-effects-interaction:

Potential violation of Checks-Effects-Interaction pattern in EssenceNFT.(): Could potentially lead to re-entrancy vulnerability. Note: Modifiers are currently not considered by this static analysis.
more
Pos: 34:4:

Block timestamp:

Use of "block.timestamp": "block.timestamp" can be influenced by miners to a certain degree. That means that a miner can "choose" the block.timestamp, to a certain degree, to change the outcome of a transaction in the mined block.
more
Pos: 52:28:

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: 62:8:

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: 192:8:

ProfileNFT.sol

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: 157:8:

Block timestamp:

Use of "block.timestamp": "block.timestamp" can be influenced by miners to a certain degree. That means that a miner can "choose" the block.timestamp, to a certain degree, to change the outcome of a transaction in the mined block.
more
Pos: 52:28:

For loop over dynamic array:

Loops that do not have a fixed number of iterations, for example, loops that depend on storage values, have to be used carefully. Due to the block gas limit, transactions can only consume a certain amount of gas. The number of iterations in a loop can grow beyond the block gas limit which can cause the complete contract to be stalled at a certain point. Additionally, using unbounded loops incurs in a lot of avoidable gas costs. Carefully test how many items at maximum you can pass to such functions to make it successful.
more
Pos: 700:8:

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: 57:8:

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: 192:8:

SubscribeNFT.sol

Check-effects-interaction:

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

more

Pos: 37:4:

Block timestamp:

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

more

Pos: 52:28:

Constant/View/Pure functions:

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

more

Pos: 60:4:

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: 61:8:

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: 190:8:

INSPECTOR LOVELY

Auth.sol

Check-effects-interaction:

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

more

Pos: 43:4:

No return:

Authority.canCall(address,address,bytes4): Defines a return type but never explicitly returns a value.

Pos: 64:4:

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: 46:8:

ERC721.sol

Constant/View/Pure functions:

ERC721TokenReceiver.onERC721Received(address,address,uint256,bytes) : Potentially should be constant/view/pure but is not. Note: Modifiers are currently not considered by this static analysis.

more

Pos: 231:4:

Similar variable names:

ERC721.transferFrom(address,address,uint256) : Variables have very similar names "to" and "id". Note: Modifiers are currently not considered by this static analysis.

Pos: 95:33:

No return:

ERC721.tokenURI(uint256): Defines a return type but never explicitly returns a value.
Pos: 30:4:

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:

Owned.sol

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: 47:8:

INSPECTOR LOVELY

RolesAuthority.sol

Check-effects-interaction:

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

more

Pos: 43:4:

Gas costs:

Gas requirement of function RolesAuthority.setUserRole 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: 101:4:

No return:

Authority.canCall(address,address,bytes4): Defines a return type but never explicitly returns a value.

Pos: 64:4:

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: 30:8:

INSPECTOR LOVELY

Create2Deployer.sol

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: 10:8:

Gas costs:

Gas requirement of function Create2Deployer.deploy 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: 7:4:

Check-effects-interaction:

Potential violation of Checks-Effects-Interaction pattern in EssenceNFT.(): Could potentially lead to re-entrancy vulnerability. Note: Modifiers are currently not considered by this static analysis.
more
Pos: 34:4:

Block timestamp:

Use of "block.timestamp": "block.timestamp" can be influenced by miners to a certain degree. That means that a miner can "choose" the block.timestamp, to a certain degree, to change the outcome of a transaction in the mined block.
more
Pos: 52:28:

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: 62:8:

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: 114:8:

INSPECTOR LOVELY

ProfileDeployer.sol

Block timestamp:

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

more

Pos: 52:28:

For loop over dynamic array:

Loops that do not have a fixed number of iterations, for example, loops that depend on storage values, have to be used carefully. Due to the block gas limit, transactions can only consume a certain amount of gas. The number of iterations in a loop can grow beyond the block gas limit which can cause the complete contract to be stalled at a certain point. Additionally, using unbounded loops incurs in a lot of avoidable gas costs. Carefully test how many items at maximum you can pass to such functions to make it successful.

more

Pos: 105:8:

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: 57:8:

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: 114:8:

INSPECTOR LOVELY

SubscribeDeployer.sol

Check-effects-interaction:

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

more

Pos: 37:4:

Block timestamp:

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

more

Pos: 52:28:

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: 192:8:

FeeMw.sol

No return:

ITreasury.isCurrencyAllowed(address): Defines a return type but never explicitly returns a value.

Pos: 27:4:

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: 18:8:

INSPECTOR LOVELY

PermissionedMw.sol

Check-effects-interaction:

Potential violation of Checks-Effects-Interaction pattern in Auth.setAuthority(contract Authority): Could potentially lead to re-entrancy vulnerability. Note: Modifiers are currently not considered by this static analysis.
more
Pos: 43:4:

Treasury.sol

No return:

ITreasury.isCurrencyAllowed(address): Defines a return type but never explicitly returns a value.
Pos: 27:4:

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: 26:8:

CollectDisallowedMw.sol

Gas costs:

Gas requirement of function CollectDisallowedMw.postProcess 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: 42:4:

INSPECTOR LOVELY

CollectFlexPaidMw.sol

Transaction origin:

Use of tx.origin: "tx.origin" is useful only in very exceptional cases. If you use it for authentication, you usually want to replace it by "msg.sender", because otherwise any contract you call can act on your behalf.
more
Pos: 108:16:

Gas costs:

Gas requirement of function CollectFlexPaidMw.setEssenceMwData is infinite: If the gas requirement of a function is higher than the block gas limit, it cannot be executed. Please avoid loops in your functions or actions that modify large areas of storage (this includes clearing or copying arrays in storage)
Pos: 77:4:

Guard conditions:

Use "assert(x)" if you never ever want x to be false, not in any circumstance (apart from a bug in your code). Use "require(x)" if x can be false, due to e.g. invalid input or a failing external component.
more
Pos: 18:8:

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: 111:36:

CollectLimitedTimePaidMw.sol

Transaction origin:

Use of tx.origin: "tx.origin" is useful only in very exceptional cases. If you use it for authentication, you usually want to replace it by "msg.sender", because otherwise any contract you call can act on your behalf.
more
Pos: 162:16:

Block timestamp:

Use of "block.timestamp": "block.timestamp" can be influenced by miners to a certain degree. That means that a miner can "choose" the block.timestamp, to a certain degree, to change the outcome of a transaction in the mined block.
more
Pos: 170:12:

Guard conditions:

Use "assert(x)" if you never ever want x to be false, not in any circumstance (apart from a bug in your code). Use "require(x)" if x can be false, due to e.g. invalid input or a failing external component.
more
Pos: 18:8:

Gas costs:

Gas requirement of function CollectMerkleDropMw.postProcess 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: 82:4:

Constant/View/Pure functions:

CollectMerkleDropMw.preProcess(uint256,uint256,address,address,bytes) : Is constant but potentially should not be.
more
Pos: 64:4:

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: 71:8:

CollectOnlySubscribedMw.sol

Gas costs:

Gas requirement of function CollectOnlySubscribedMw.postProcess 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: 48:4:

Constant/View/Pure functions:

CollectOnlySubscribedMw.setEssenceMwData(uint256,uint256,bytes) : Is constant but potentially should not be. Note: Modifiers are currently not considered by this static analysis.
more
Pos: 21:4:

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: 192:8:

CollectPaidMw.sol

Transaction origin:

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

more

Pos: 140:16:

Check-effects-interaction:

Potential violation of Checks-Effects-Interaction pattern in Auth.setAuthority(contract Authority): Could potentially lead to re-entrancy vulnerability. Note: Modifiers are currently not considered by this static analysis.Note: Import aliases are currently not supported by this static analysis.

more

Pos: 43:4:

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: 192:8:

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: 144:36:

INSPECTOR
LOVELY

CollectPermissionMw.sol

Block timestamp:

Use of "block.timestamp": "block.timestamp" can be influenced by miners to a certain degree. That means that a miner can "choose" the block.timestamp, to a certain degree, to change the outcome of a transaction in the mined block.
more
Pos: 52:28:

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: 220:12:

Delete from dynamic array:

Using "delete" on an array leaves a gap. The length of the array remains the same. If you want to remove the empty position you need to shift items manually and update the "length" property.
more
Pos: 114:8:

CollectPermissionPaidMw.sol

Transaction origin:

Use of tx.origin: "tx.origin" is useful only in very exceptional cases. If you use it for authentication, you usually want to replace it by "msg.sender", because otherwise any contract you call can act on your behalf.
more
Pos: 141:16:

Block timestamp:

Use of "block.timestamp": "block.timestamp" can be influenced by miners to a certain degree. That means that a miner can "choose" the block.timestamp, to a certain degree, to change the outcome of a transaction in the mined block.
more
Pos: 52:28:

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: 53:8:

FeeCreationMw.sol

Block timestamp:

Use of "block.timestamp": "block.timestamp" can be influenced by miners to a certain degree. That means that a miner can "choose" the block.timestamp, to a certain degree, to change the outcome of a transaction in the mined block.
more
Pos: 52:28:

Gas costs:

Gas requirement of function CollectPermissionMw.preProcess is infinite: If the gas requirement of a function is higher than the block gas limit, it cannot be executed. Please avoid loops in your functions or actions that modify large areas of storage (this includes clearing or copying arrays in storage)
Pos: 77:4:

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: 190:8:

PermissionedFeeCreationMw.sol

Check-effects-interaction:

Potential violation of Checks-Effects-Interaction pattern in Auth.setAuthority(contract Authority): Could potentially lead to re-entrancy vulnerability. Note: Modifiers are currently not considered by this static analysis.
more
Pos: 43:4:

Block timestamp:

Use of "block.timestamp": "block.timestamp" can be influenced by miners to a certain degree. That means that a miner can "choose" the block.timestamp, to a certain degree, to change the outcome of a transaction in the mined block.
more
Pos: 52:28:

Gas costs:

Gas requirement of function RolesAuthority.setUserRole 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: 101:4:

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: 30:8:

StableFeeCreationMw.sol

Check-effects-interaction:

Potential violation of Checks-Effects-Interaction pattern in Auth.setAuthority(contract Authority): Could potentially lead to re-entrancy vulnerability. Note: Modifiers are currently not considered by this static analysis.
more
Pos: 43:4:

Block timestamp:

Use of "block.timestamp": "block.timestamp" can be influenced by miners to a certain degree. That means that a miner can "choose" the block.timestamp, to a certain degree, to change the outcome of a transaction in the mined block.
more
Pos: 52:28:

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: 118:8:

SubscribeDisallowedMw.sol

Gas costs:

Gas requirement of function SubscribeDisallowedMw.preProcess 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: 32:4:

Constant/View/Pure functions:

SubscribeDisallowedMw.postProcess(uint256,address,address,bytes) : Potentially should be constant/view/pure but is not.
more
Pos: 42:4:

SubscribeOnlyOnceMw.sol

Gas costs:

Gas requirement of function SubscribeOnlyOnceMw.postProcess 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:

Constant/View/Pure functions:

ERC721TokenReceiver.onERC721Received(address,address,uint256,bytes) : Potentially should be constant/view/pure but is not. Note: Modifiers are currently not considered by this static analysis.
more
Pos: 231:4:

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: 190:8:

SubscribePaidMw.sol

Check-effects-interaction:

Potential violation of Checks-Effects-Interaction pattern in Auth.setAuthority(contract Authority): Could potentially lead to re-entrancy vulnerability. Note: Modifiers are currently not considered by this static analysis.Note: Import aliases are currently not supported by this static analysis.
more
Pos: 43:4:

Gas costs:

Gas requirement of function RolesAuthority.setOwner 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: 53:4:

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: 46:8:

Gas costs:

Gas requirement of function CyberBoxNFT.initialize is infinite: If the gas requirement of a function is higher than the block gas limit, it cannot be executed. Please avoid loops in your functions or actions that modify large areas of storage (this includes clearing or copying arrays in storage)
Pos: 51:4:

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: 26:8:

CyberGrandNFT.sol

Block timestamp:

Use of "block.timestamp": "block.timestamp" can be influenced by miners to a certain degree. That means that a miner can "choose" the block.timestamp, to a certain degree, to change the outcome of a transaction in the mined block.
more
Pos: 52:28:

Gas costs:

Gas requirement of function CyberGrandNFT.initialize is infinite: If the gas requirement of a function is higher than the block gas limit, it cannot be executed. Please avoid loops in your functions or actions that modify large areas of storage (this includes clearing or copying arrays in storage)
Pos: 50:4:

Gas costs:

Gas requirement of function MiniShardNFT.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: 72:4:

RelationshipChecker.sol

Gas costs:

Gas requirement of function RelationshipChecker.isSubscribedByMe 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: 53:4:

No return:

IProfileNFT.getEssenceNFT(uint256,uint256): Defines a return type but never explicitly returns a value.
Pos: 422:4:

Link3ProfileDescriptorStorage.sol

Gas costs:

Gas requirement of function Link3ProfileDescriptorStorage.animationTemplate 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: 11:4:

Guard conditions:

Use "assert(x)" if you never ever want x to be false, not in any circumstance (apart from a bug in your code). Use "require(x)" if x can be false, due to e.g. invalid input or a failing external component.
more
Pos: 45:8:

CyberVault.sol

Check-effects-interaction:

Potential violation of Checks-Effects-Interaction pattern in CyberVault.claim1155(string,address,address,uint256,uint256,struct DataTypes.EIP712Signature): Could potentially lead to re-entrancy vulnerability. Note: Modifiers are currently not considered by this static analysis.Note: Import aliases are currently not supported by this static analysis.
more
Pos: 195:4:

Gas costs:

Gas requirement of function CyberVault.claim 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: 102:4:

Gas costs:

Gas requirement of function CyberVault.claim 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: 102:4:

FrameNFT.sol

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: 157:8:

No return:

LibString.toHexString(address): Defines a return type but never explicitly returns a value.
Pos: 152:4:

Link3ProfileDescriptor.sol

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: 54:8:

Gas costs:

Gas requirement of function Link3ProfileDescriptor.animationTemplate 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: 11:4:

INSPECTOR LOVELY

Link3ProfileDescriptorV2.sol

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: 11:8:

For loop over dynamic array:

Loops that do not have a fixed number of iterations, for example, loops that depend on storage values, have to be used carefully. Due to the block gas limit, transactions can only consume a certain amount of gas. The number of iterations in a loop can grow beyond the block gas limit which can cause the complete contract to be stalled at a certain point. Additionally, using unbounded loops incurs in a lot of avoidable gas costs. Carefully test how many items at maximum you can pass to such functions to make it successful.
more
Pos: 203:8:

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: 47:8:

MBNFT.sol

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: 11:8:

For loop over dynamic array:

Loops that do not have a fixed number of iterations, for example, loops that depend on storage values, have to be used carefully. Due to the block gas limit, transactions can only consume a certain amount of gas. The number of iterations in a loop can grow beyond the block gas limit which can cause the complete contract to be stalled at a certain point. Additionally, using unbounded loops incurs in a lot of avoidable gas costs. Carefully test how many items at maximum you can pass to such functions to make it successful.

more

Pos: 203:8:

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: 47:8:

MiniShardNFT.sol

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: 157:8:

Block timestamp:

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

more

Pos: 52:28:

INSPECTOR LOVELY

CyberBoxNFT.sol

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: 95:8:

Block timestamp:

Use of "block.timestamp": "block.timestamp" can be influenced by miners to a certain degree. That means that a miner can "choose" the block.timestamp, to a certain degree, to change the outcome of a transaction in the mined block.
more
Pos: 52:28:

Gas costs:

Gas requirement of function CyberBoxNFT.initialize is infinite: If the gas requirement of a function is higher than the block gas limit, it cannot be executed. Please avoid loops in your functions or actions that modify large areas of storage (this includes clearing or copying arrays in storage)
Pos: 51:4:

CyberBoxNFT.sol

Block timestamp:

Use of "block.timestamp": "block.timestamp" can be influenced by miners to a certain degree. That means that a miner can "choose" the block.timestamp, to a certain degree, to change the outcome of a transaction in the mined block.
more
Pos: 52:28:

INSPECTOR LOVELY

UpgradeableBeacon.sol

Low level calls:

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

more

Pos: 125:50:

Check-effects-interaction:

Potential violation of Checks-Effects-Interaction pattern in Address.functionCallWithValue(address,bytes,uint256,string): Could potentially lead to re-entrancy vulnerability.

more

Pos: 120:4:

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: 39:8:

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: 242:8:

INSPECTOR LOVELY

CYBER.sol

Block timestamp:

Use of "block.timestamp": "block.timestamp" can be influenced by miners to a certain degree. That means that a miner can "choose" the block.timestamp, to a certain degree, to change the outcome of a transaction in the mined block.
more
Pos: 2522:23:

Gas costs:

Gas requirement of function CyberToken.permit 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: 2513:11:

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: 2805:15:

Gas costs:

Gas requirement of function CyberToken.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: 2835:11:

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: 359:35:

INSPECTOR LOVELY

COMPLIANCE ANALYSIS

Linters are the utility tools that analyze the given source code and report programming errors, bugs, and stylistic errors. For the Solidity language, there are some linter tools available that a developer can use to improve the quality of their Solidity contracts

CyberNFTBaseFlex.sol

```
Compiler version 0.8.14 does not satisfy the ^0.5.8 semver requirement
Pos: 1:2
Explicitly mark visibility in function (Set ignoreConstructors to true if using solidity >=0.7.0)
Pos: 5:31
```

EIP712.sol

```
Compiler version 0.8.14 does not satisfy the ^0.5.8 semver requirement
Pos: 1:2
Avoid making time-based decisions in your business logic
Pos: 29:51
```

CyberNFTBase.sol

```
Compiler version 0.8.14 does not satisfy the ^0.5.8 semver requirement
Pos: 1:2
Explicitly mark visibility in function (Set ignoreConstructors to true if using solidity >=0.7.0)
Pos: 5:31
```

CyberEngine.sol

```
Compiler version 0.8.14 does not satisfy the ^0.5.8 semver requirement
Pos: 1:2
Explicitly mark visibility in function (Set ignoreConstructors to true if using solidity >=0.7.0)
Pos: 5:71
Variable "newImpl" is unused
Pos: 32:224
Code contains empty blocks
Pos: 70:320
```

INSPECTOR LOVELY

EssenceNFT.sol

Compiler version 0.8.14 does not satisfy the ^0.5.8 semver requirement
Pos: 1:2
Explicitly mark visibility in function (Set ignoreConstructors to true if using solidity >=0.7.0)
Pos: 5:33

ProfileNFT.sol

Compiler version 0.8.14 does not satisfy the ^0.5.8 semver requirement
Pos: 1:2

SubscribeNFT.sol

Compiler version 0.8.14 does not satisfy the ^0.5.8 semver requirement
Pos: 1:2
Explicitly mark visibility in function (Set ignoreConstructors to true if using solidity >=0.7.0)
Pos: 5:36

Auth.sol

Compiler version >=0.8.0 does not satisfy the ^0.5.8 semver requirement
Pos: 1:2
global import of path @openzeppelin/contracts/proxy/utils/Initializable.sol is not allowed. Specify names to import individually or bind all exports of the module into a name (import "path" as Name)
Pos: 1:4
Function name must be in mixedCase
Pos: 5:20

ERC721.sol

Compiler version 0.8.14 does not satisfy the ^0.5.8 semver requirement
Pos: 1:2
Function name must be in mixedCase
Pos: 5:61

INSPECTOR LOVELY

Owned.sol

Compiler version =0.8.14 does not satisfy the ^0.5.8 semver requirement
Pos: 1:2
Function name must be in mixedCase
Pos: 5:34

RolesAuthority.sol

Compiler version 0.8.14 does not satisfy the ^0.5.8 semver requirement
Pos: 1:2
Explicitly mark visibility in function (Set ignoreConstructors to true if using solidity >=0.7.0)
Pos: 5:27

Create2Deployer.sol

Compiler version 0.8.14 does not satisfy the ^0.5.8 semver requirement
Pos: 1:1
Avoid using inline assembly. It is acceptable only in rare cases
Pos: 9:9

EssenceDeployer.sol

Compiler version 0.8.14 does not satisfy the ^0.5.8 semver requirement
Pos: 1:2

ProfileDeployer.sol

Compiler version 0.8.14 does not satisfy the ^0.5.8 semver requirement
Pos: 1:2

SubscribeDeployer.sol

Compiler version 0.8.14 does not satisfy the ^0.5.8 semver requirement
Pos: 1:2

INSPECTOR LOVELY

FeeMw.sol

Compiler version 0.8.14 does not satisfy the ^0.5.8 semver requirement
Pos: 1:2
Explicitly mark visibility in function (Set ignoreConstructors to true if using solidity >=0.7.0)
Pos: 5:16

PermissionedMw.sol

Compiler version 0.8.14 does not satisfy the ^0.5.8 semver requirement
Pos: 1:2
Explicitly mark visibility in function (Set ignoreConstructors to true if using solidity >=0.7.0)
Pos: 5:28

Treasury.sol

Compiler version 0.8.14 does not satisfy the ^0.5.8 semver requirement
Pos: 1:2
Explicitly mark visibility in function (Set ignoreConstructors to true if using solidity >=0.7.0)
Pos: 5:28

CollectDisallowedMw.sol

Compiler version 0.8.14 does not satisfy the ^0.5.8 semver requirement
Pos: 1:2
Code contains empty blocks
Pos: 16:47

CollectFlexPaidMw.sol

Compiler version 0.8.14 does not satisfy the ^0.5.8 semver requirement
Pos: 1:2
Explicitly mark visibility in function (Set ignoreConstructors to true if using solidity >=0.7.0)
Pos: 5:67
Avoid to use tx.origin
Pos: 17:107
Variable "treasuryCollected" is unused
Pos: 9:110
Code contains empty blocks
Pos: 16:132

INSPECTOR LOVELY

CollectLimitedTimePaidMw.sol

Compiler version 0.8.14 does not satisfy the ^0.5.8 semver requirement
Pos: 1:2
Explicitly mark visibility in function (Set ignoreConstructors to true if using solidity >=0.7.0)
Pos: 5:77
Avoid to use tx.origin
Pos: 17:161
Avoid making time-based decisions in your business logic
Pos: 13:169
Avoid making time-based decisions in your business logic
Pos: 13:174
Variable "currency" is unused
Pos: 13:192
Variable "actualPaid" is unused
Pos: 13:195
Code contains empty blocks
Pos: 16:208

CollectMerkleDropMw.sol

Compiler version 0.8.14 does not satisfy the ^0.5.8 semver requirement
Pos: 1:2
Explicitly mark visibility of state
Pos: 5:29
Code contains empty blocks
Pos: 16:87

CollectOnlySubscribedMw.sol

Compiler version 0.8.14 does not satisfy the ^0.5.8 semver requirement
Pos: 1:2
Code contains empty blocks
Pos: 16:53

CollectPaidMw.sol

Compiler version 0.8.14 does not satisfy the ^0.5.8 semver requirement
Pos: 1:2
Explicitly mark visibility in function (Set ignoreConstructors to true if using solidity >=0.7.0)
Pos: 5:71
Avoid to use tx.origin
Pos: 17:139
Variable "currency" is unused
Pos: 9:141
Variable "actualPaid" is unused
Pos: 9:145
Code contains empty blocks
Pos: 16:164

INSPECTOR LOVELY

CollectPermissionMw.sol

Compiler version 0.8.14 does not satisfy the ^0.5.8 semver requirement
Pos: 1:2
Code contains empty blocks
Pos: 16:120

CollectPermissionPaidMw.sol

Compiler version 0.8.20 does not satisfy the ^0.5.8 semver requirement
Pos: 1:2
Explicitly mark visibility in function (Set ignoreConstructors to true if using solidity >=0.7.0)
Pos: 5:75
Avoid to use tx.origin
Pos: 17:140
Variable "currency" is unused
Pos: 13:174
Variable "actualPaid" is unused
Pos: 13:177
Code contains empty blocks
Pos: 16:191

FeeCreationMw.sol

Compiler version 0.8.14 does not satisfy the ^0.5.8 semver requirement
Pos: 1:2
Explicitly mark visibility in function (Set ignoreConstructors to true if using solidity >=0.7.0)
Pos: 5:63
Code contains empty blocks
Pos: 56:63
Variable "data" is unused
Pos: 9:72
Code contains empty blocks
Pos: 25:86

PermissionedFeeCreationMw.sol

Compiler version 0.8.14 does not satisfy the ^0.5.8 semver requirement
Pos: 1:2
Explicitly mark visibility in function (Set ignoreConstructors to true if using solidity >=0.7.0)
Pos: 5:72
Code contains empty blocks
Pos: 5:75
Code contains empty blocks
Pos: 25:111

INSPECTOR LOVELY

StableFeeCreationMw.sol

Compiler version 0.8.20 does not satisfy the ^0.5.8 semver requirement
Pos: 1:2
Explicitly mark visibility in function (Set ignoreConstructors to true if using solidity >=0.7.0)
Pos: 5:85
Code contains empty blocks
Pos: 76:85
Code contains empty blocks
Pos: 25:132
Avoid making time-based decisions in your business logic
Pos: 24:328
Code contains empty blocks
Pos: 73:351
Code contains empty blocks
Pos: 78:357
Code contains empty blocks
Pos: 73:362

SubscribeDisallowedMw.sol

Compiler version 0.8.14 does not satisfy the ^0.5.8 semver requirement
Pos: 1:2
Code contains empty blocks
Pos: 25:46

SubscribeOnlyOnceMw.sol

Compiler version 0.8.14 does not satisfy the ^0.5.8 semver requirement
Pos: 1:2
Code contains empty blocks
Pos: 25:51

SubscribePaidMw.sol

Compiler version 0.8.20 does not satisfy the ^0.5.8 semver requirement
Pos: 1:2
Explicitly mark visibility in function (Set ignoreConstructors to true if using solidity >=0.7.0)
Pos: 5:65
Variable "currency" is unused
Pos: 9:124
Variable "actualPaid" is unused
Pos: 9:128
Code contains empty blocks
Pos: 16:147

INSPECTOR LOVELY

CyberBoxNFT.sol

Compiler version 0.8.14 does not satisfy the ^0.5.8 semver requirement
Pos: 1:2
Explicitly mark visibility in function (Set ignoreConstructors to true if using solidity >=0.7.0)
Pos: 5:35
Code contains empty blocks
Pos: 69:191

CyberGrandNFT.sol

Compiler version 0.8.20 does not satisfy the ^0.5.8 semver requirement
Pos: 1:2
Explicitly mark visibility in function (Set ignoreConstructors to true if using solidity >=0.7.0)
Pos: 5:35
Code contains empty blocks
Pos: 69:202

CyberVault.sol

Compiler version 0.8.20 does not satisfy the ^0.5.8 semver requirement
Pos: 1:2
Explicitly mark visibility of state
Pos: 5:74
Explicitly mark visibility of state
Pos: 5:75
Explicitly mark visibility in function (Set ignoreConstructors to true if using solidity >=0.7.0)
Pos: 5:81

FrameNFT.sol

Compiler version ^0.8.20 does not satisfy the ^0.5.8 semver requirement
Pos: 1:3
Explicitly mark visibility in function (Set ignoreConstructors to true if using solidity >=0.7.0)
Pos: 5:37
Code contains empty blocks
Pos: 20:37
Code contains empty blocks
Pos: 22:55
Code contains empty blocks
Pos: 22:66
Code contains empty blocks
Pos: 37:79
Code contains empty blocks
Pos: 39:92

INSPECTOR LOVELY

Link3ProfileDescriptor.sol

Compiler version 0.8.20 does not satisfy the ^0.5.8 semver requirement
Pos: 1:2
Use double quotes for string literals
Pos: 29:95
Use double quotes for string literals
Pos: 17:133
Use double quotes for string literals
Pos: 17:135
Use double quotes for string literals
Pos: 17:137

Link3ProfileDescriptorV2.sol

Compiler version 0.8.20 does not satisfy the ^0.5.8 semver requirement
Pos: 1:2
Use double quotes for string literals
Pos: 29:78
Use double quotes for string literals
Pos: 29:84
Use double quotes for string literals
Pos: 17:110
Use double quotes for string literals
Pos: 17:112
Use double quotes for string literals
Pos: 17:114
Use double quotes for string literals
Pos: 17:116

Link3ProfileDescriptorV3.sol

Compiler version 0.8.20 does not satisfy the ^0.5.8 semver requirement
Pos: 1:2
Use double quotes for string literals
Pos: 29:78
Use double quotes for string literals
Pos: 29:80
Use double quotes for string literals
Pos: 29:82
Use double quotes for string literals
Pos: 29:84
Use double quotes for string literals
Pos: 17:110
Use double quotes for string literals
Pos: 17:112
Use double quotes for string literals
Pos: 17:114
Use double quotes for string literals
Pos: 17:116

INSPECTOR LOVELY

MBNFT.sol

Compiler version 0.8.14 does not satisfy the ^0.5.8 semver requirement
Pos: 1:2
Explicitly mark visibility in function (Set ignoreConstructors to true if using solidity >=0.7.0)
Pos: 5:37
Code contains empty blocks
Pos: 69:189

MiniShardNFT.sol

Compiler version ^0.8.20 does not satisfy the ^0.5.8 semver requirement
Pos: 1:3
Explicitly mark visibility in function (Set ignoreConstructors to true if using solidity >=0.7.0)
Pos: 5:55
Error message for require is too long
Pos: 9:77
Error message for require is too long
Pos: 9:121
Error message for require is too long
Pos: 9:166
Error message for require is too long
Pos: 9:183

RelationshipChecker.sol

Compiler version 0.8.14 does not satisfy the ^0.5.8 semver requirement
Pos: 1:2
Explicitly mark visibility in function (Set ignoreConstructors to true if using solidity >=0.7.0)
Pos: 5:15

CyberBoxNFTStorage.sol

Compiler version 0.8.14 does not satisfy the ^0.5.8 semver requirement
Pos: 1:2

CyberEngineStorage.sol

Compiler version 0.8.14 does not satisfy the ^0.5.8 semver requirement
Pos: 1:2

INSPECTOR LOVELY

CyberGrandNFTStorage.sol

Compiler version 0.8.14 does not satisfy the ^0.5.8 semver requirement
Pos: 1:2

EssenceNFTStorage.sol

Compiler version 0.8.14 does not satisfy the ^0.5.8 semver requirement
Pos: 1:2

Link3ProfileDescriptorStorage.sol

Compiler version 0.8.14 does not satisfy the ^0.5.8 semver requirement
Pos: 1:2

MBNFTStorage.sol

Compiler version 0.8.14 does not satisfy the ^0.5.8 semver requirement
Pos: 1:2

ProfileNFTStorage.sol

Compiler version 0.8.14 does not satisfy the ^0.5.8 semver requirement
Pos: 1:2

SubscribeNFTStorage.sol

Compiler version 0.8.14 does not satisfy the ^0.5.8 semver requirement
Pos: 1:2

INSPECTOR LOVELY

UpgradeableBeacon.sol

```
Compiler version ^0.8.14 does not satisfy the ^0.5.8 semver requirement
Pos: 1:3
Error message for require is too long
Pos: 9:63
Error message for require is too long
Pos: 9:120
Error message for require is too long
Pos: 9:145
Error message for require is too long
Pos: 9:169
Variable name must be in mixedCase
Pos: 5:206
Explicitly mark visibility in function (Set ignoreConstructors to true if using solidity >=0.7.0)
Pos: 5:217
Error message for require is too long
Pos: 9:254
```

CYBER.sol

```
Compiler version 0.8.20 does not satisfy the ^0.5.8 semver requirement
Pos: 1:2
Variable name must be in mixedCase
Pos: 5:10
Explicitly mark visibility in function (Set ignoreConstructors to true if using solidity >=0.7.0)
Pos: 5:12
```

SOFTWARE ANALYSIS RESULT

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

INSPECTOR
LOVELY

INSPECTOR LOVELY

INFO

Website: Inspector.lovely.finance

Telegram community: t.me/inspectorlovely

Twitter: twitter.com/InspectorLovely

inspector.lovely.finance