Cyberscope

Audit Report

Hedgepie Finance

January 2023

Github https://github.com/innovation-upstream/hedgepie-dev

Commit d3d9834918219c613b50c8d6040a587102402a29

Audited by © cyberscope

Table of Contents

Table of Contents 1
Review 4
Audit Updates 4
Source Files 5
Introduction 13
Roles 14
Info Contracts 14
Manager Contracts 14
Investor Contracts 15
Master Chef Contract 16
Token contract 17
NFT contract 17
Contract Infrastructure Architecture Review 18
Libraries Dependency Review 18
Recommendations 18
Contracts Infrastructure Review 19
Recommendations 19
Adapters Review 20
Recommendations 20
Platform Integrated Adapters 21
Test Deployments 22
Analysis 27
MSI - Mocking Strategy Issue 29
Description 29
Recommendation 29
RII - Reward Inconsistency Issue 30
Description 30
Recommendation 31
UPEH - Underneath Protocols Error Handling 32
Description 32
Recommendation 32

PSSI - Potential State Synchronization Inconsistency 33
Description 33
Recommendation 33
RFA - Redundant Function Argument 34
Description 34
Recommendation 34
DSM - Data Structure Misuse 35
Description 35
Recommendation 35
CR - Code Repetition 36
Description 36
Recommendation 37
DDP - Decimal Division Precision 38
Description 38
Recommendation 38
RDSF - Redundant Data Structure Fields 39
Description 39
Recommendation 39
AAO - Accumulated Amount Overflow 40
Description 40
Recommendation 41
MSC - Missing Sanity Check 42
Description 42
Recommendation 43
RSML - Redundant SafeMath Library 44
Description 44
Recommendation 44
L04 - Conformance to Solidity Naming Conventions 45
Description 45
Recommendation 46
L06 - Missing Events Access Control 47
Description 47
Recommendation 47
L07 - Missing Events Arithmetic 48
Description 48

Recommendation 48
L13 - Divide before Multiply Operation 49
Description 49
Recommendation 49
L14 - Uninitialized Variables in Local Scope 50
Description 50
Recommendation 50
L19 - Stable Compiler Version 51
Description 51
Recommendation 51
Functions Analysis 52
Flow Graph 98
Summary 99
Disclaimer 100
About Cyberscope 101

Review

Repository	https://github.com/innovation-upstream/hedgepie-dev/tree/audit
Commit	04831d7fbe9d2a4b35ea873151e1ff8f4c596220

Audit Updates

Initial Audit	05 Jan 2023

Source Files

Filename	SHA256
adapters/BaseAdapterBsc.sol	dcb4ef104d89c772fa8a475c74db42c1d162d06b7f305ffce041d42ae8fdf4e3
adapters/BaseAdapterEth.sol	3f90d8f895ba5fea95e381b461cbd4ee382cf65918956b11e5832b40659b0c01
adapters/BaseAdapterMatic.sol	2ad8ca32318b9a9a9010ce60820878c72650a899a61a0489a22c97f8df75a28d
adapters/bnb/alpaca/alpaca-ausd-adapter.sol	aeea47788695d98988fd0b99042121863ab9a24f4f2bf9cb866b9615647dcea1
adapters/bnb/alpaca/alpaca-lend-adapter.sol	ca796503e891645d2d52ca969bda0fb27669dd693e76eebdb39baac42f590b9a
adapters/bnb/alpaca/alpaca-stake-adapter.sol	9f215e2b1f0931fbb0c2494f1cfd44ff68c2aa79ed27f7e497af989fdf6cd9d9
adapters/bnb/apeswap/apeswap-banana-adapter.sol	4f5a4bccd7b6a33414c056286e283e2038625e4dc327a0283bbe7fa0b8c47f42
adapters/bnb/apeswap/apeswap-farm-lp-adapter.sol	17403f30cb21910f6f6216e022978f5ae797ca3a588b907be32039a6412c658a
adapters/bnb/apeswap/apeswap-jungle-adapter.sol	10469124e3be87c9c997df52091d160c56cc092255645a427d8fdddb05c377d4
adapters/bnb/apeswap/apeswap-vault-adapter.sol	1d32d0223595afc215fa929c89fe66405340494347cd3d0e441061fc664bcd2b
adapters/bnb/autofarm/auto-vault-adapter.sol	cc74ee1bf3124c53c32f9aeef258b3a18994476358f4c6d983cd4a294fc88c44
adapters/bnb/beefy/beefy-vault-adapter.sol	0cd5e908a80d36f4f3253cba5adfc48a186a8ace956289c386551dd84493b69b
adapters/bnb/belt.fi/belt-vault-adapter.sol	c0671a3e51a3c2a05e9f4f9166ecf4978cd7bd5468076360a2ee1a356e032fe1

adapters/bnb/biswap/biswap-farm-lp-adapter.sol	0d0d3752cd5169d8ef920d0cacfb97750f8da3059d1c15050c21c60876c0ea38
adapters/bnb/pancakeswap/pancake-farm-adapter.sol	43d85f108dbc4b722bdb73c7d423c38c22539c794b2a6068e06b29a52d7cc99c
adapters/bnb/pancakeswap/pancake-stake-adapter.sol	19eafaf09fb476dea4468e9a9ecb87eca6d8a565307f28bbcb071bc23b583b16
adapters/bnb/venus/interface/ComptrollerInterface.sol	6c0c70495043338d37c88e4f15f07ddcb07921c47aab3162ffb99ddac97c7460
adapters/bnb/venus/interface/InterestRateModel.sol	c96eeb98fc0e4f5c392d664ceccf11769672886070ea7350752868538b99d949
adapters/bnb/venus/interface/VBep20Interface.sol	510f5af5f70b2ca67bb527adc977b68bd7f7bccdc6281f86044d56ed07d8b984
adapters/bnb/venus/interface/VTokenInterface.sol	67e4648e4624e2ff282c23f0f10116b919ac8e17ddabde400eb8a339f9438a14
adapters/bnb/venus/venus-adapter-mock.sol	89f20d2b03ba4c2b778fc61f1506441b970528144f7678fc92740270788f35ac
adapters/bnb/venus/venus-lend-adapter.sol	d09bc839333091c93c0b656cc23b805536b7e10cc3339f2ee124bec21d2f1d5a
adapters/bnb/venus/venus-lev-adapter.sol	3920a1ebc1cf5019d85c91f74643c489cb3cb6838c94f8595456eba2aa5b6df0
adapters/eth/aave/AaveLendAdapterEth.sol	5dd0da384c3a369f281502cebc47ee6f094006a8eb1ec9b09b0ae194a3cee105
adapters/eth/balancer/balancer-vault-adapter.sol	37a6bdbd3231d5942c3425e91384cc894378ea9d8b4e6b5bf57ffc424f2ebd8c
adapters/eth/compound/CompoundLendAdapterEth.sol	694ba77e1a8c0e6f7791e2b8898a7332872663ba1c8e2875e6dd5ac25497e113
adapters/eth/curve/curve-gauge-adapter.sol	4c17f2101229d7aecb316c68908f7ed84a9875b778eace54e73840bf3c724ae9
adapters/eth/pickle/PickleCurveGaugeAdapter.sol	bbdfacd078e835d4499177b6b51d15d9abdab037a3b562eb523230f56272bd5b

adapters/eth/pickle/PickleSingleGaugeAdapter.sol	a63e28780d20058080e105c25c7ed80c51046d5859a93b4ca21df73b46dee22d
adapters/eth/pickle/PickleSushiGaugeAdapter.sol	cf7ab55d3a20398cf0d3f0fda91ffdf1170b576dfd6e11ee057d622ccb886651
adapters/eth/pickle/PickleSushiMasterAdapter.sol	f68dda333f1ff7cd5cefa659dfdd165332d77a3ae5c28406b942ef5cdb5992b9
adapters/eth/sushiswap/SushiFarmAdapterEth.sol	bff970807312a8b1b5ee8b475f048974318c69d0c332d781aa277b92835955d8
adapters/eth/sushiswap/SushiFarmV2AdapterEth.sol	f665d251c61a2970d2b7802b10b3013ae0ac87a24b63c6457df7d8b51f1ec3ec
adapters/eth/uniswap/uniswap-lp-v3-adapter.sol	9b979a817d04a2f6fc5043829c6d9e5490525bebbbe47b6110114355c25f512f
adapters/eth/yearn/yearn-curve-adaper.sol	7b980014792cec3b260b58fa0220124b066e2759aef6e64fc7fd189c26bfd6c7
adapters/eth/yearn/yearn-single-adapter.sol	1a060b79822803141b3f2201d1f3f8851b471cbf5ec2c6d4e006c009d5d8ec5b
adapters/polygon/aave/aave-market-v2-adapter.sol	014cc055513695fd024f6fd3d026d60659016c7d0e45cec86e60ef8ace56a501
adapters/polygon/aave/aave-market-v3-adapter.sol	0d3fa7b57d7a43dc934579adc020af6960e94b44a19045cf680f42f3e8512a23
adapters/polygon/apeswap/apeswap-farm-adapter.sol	17bde191e69e93395c2410fc1cc7934dcd9574f569613480a93f0a4ffd19694e
adapters/polygon/beefy/beefy-balancer-adapter.sol	9c943901e176cd0323732813b768126aa8380f68c2b99d34823f9fc3a706547f
adapters/polygon/beefy/beefy-stargate-adapter.sol	c7f744dcba8fd73ef7e091224d7d7f391fde28586760bb1f75e911561d9f375b
adapters/polygon/beefy/beefy-vault-adapter.sol	d3038f62468e61712b5be12a2facb42eaf69c52451015f073e1903e6f0547e1d
adapters/polygon/curve/curve-lp-adapter.sol	2816d6a4610b47bda0396ff02032a60b055576dd02f1cbaaac1fb5ce5dfe068b

adapters/polygon/quickswap/quick-lp-dual-adapter.sol	7da20f8277b238dd55fdf6e34be680c225543e8ae90199ad8e84b1a7abc5bc1e
adapters/polygon/quickswap/quick-lp-farm-adapter.sol	5ec72a4d2b274edb3381c33a033bfd4fd08233308d9a8308a9dc65dc8c430e1c
adapters/polygon/quickswap/quick-stake-adapter.sol	12e080ab290e9801204acb994febc450510b6eba5c844b95ef8d3735841b752e
adapters/polygon/stargate/stargate-farm-adapter.sol	5936d06eae52e8f8d37587b7a823e8683adfa408029509a47a77de90608952f8
adapters/polygon/sushi/sushi-farm-adapter.sol	36178893a264f23d270b175b2f649776e557f273191843fc716b8f306dab4b0e
adapters/polygon/uniswap/uniswap-lp-adapter.sol	9e7fc9bc58b1fad4e06a1aa4c9f640c52eeff28262ee8af40b0fc60d89a57f4a
HedgepieAdapterInfoBsc.sol	3091579f2e90e1d4ac9bc342e5e3a800f769087058c7f05b4b2ca1c72a0ee74a
HedgepieAdapterInfoEth.sol	fde19d46eac341f2ee5ac895edf8fcc42a46a702c5ae7a244335da9dc247e234
HedgepieAdapterInfoMatic.sol	3b7346e4db8dd6f7c7e38a7f3ed448f31a88f7ff6e6ba103fdf26daf1b76c62e
HedgepieAdapterManagerBsc.sol	018697cf8eb9200b0a10882823846a5205d0854f47c4be5bd1c8e6bf20d6065b
HedgepieAdapterManagerEth.sol	9163bf69e1f4038c1656f2debb18c10a235c8b4628e99dd51e473f2da0aa0a92
HedgepieAdapterManagerMatic.sol	4c2f69098108e607fec81de9dc881c8d3c4a9fa6244cdbb480569717b58fee2d
HedgepieInvestorBsc.sol	0e10608a8fbfc4fe9a09c476e89a3245dac48d2f5ed461ec6df53661f80b6e83
HedgepieInvestorEth.sol	fa6163f0962e13ba5a95ad212f4e7d3f40f7d5bd17cfbd7bad4b49ff8a2bec92
HedgepieInvestorMatic.sol	46521112cfc12a5e5c3c2059e3686bcf14f4f49c802373b61445d99f8d06ab3d

HedgepieMasterChef.sol	1df29f15faa13439522f35a202d706d4e4e01a8114c564825827b45538eabb58
HedgepieToken.sol	4fa719e08ce69ee72ea8bb6fcf4ca4306a1b5a76eb8247468d9e1f0959cf75a3
HedgepieYBNFT.sol	d2e7056bc4c8a8e29b6d7ee17f54580e7713e5ffd854c54fb1de0a89ac5276c6
interfaces/IAdapter.sol	aee42ee6a0aa17d24402a535c9be09c6c2c5787385025504d2a8ba5919e93c6c
interfaces/IAdapterBsc.sol	81c6affa589eaf606954c73e669fcf3f5cb1eb1ff3d1d068455cd4ea35517e11
interfaces/IAdapterEth.sol	17ba164ec198329db32c183dce8b489c66e6d041858578c1e9625afa3a078a47
interfaces/IAdapterManager.sol	e453e36cb75ff0fce338b905002db2f71bac037a30b183e237c49292309c284d
interfaces/IAdapterManagerEth.sol	d517738b07503ebc8710f16f85ed1f9d8ecc1a78d97e77435f0b1ae444c07f10
interfaces/IAdapterManagerMatic.sol	24acbb7b62a484fec47967c8657aa8123777d764d90cbcd5eb195f68004502d7
interfaces/IAdapterMatic.sol	26c8056e7a57bea9d94c0011480c8bff8276f5585a58def23310faa2a6022632
interfaces/IBEP165.sol	e5c5014bfa05d512027982a43066cf8e01b0364d117e162f07b8f0d0c7758985
interfaces/IBEP20.sol	d7adbc5408c5d75e05bdd2d8618b2e013dc8c2539c26d2961f650de03ac3c3af
interfaces/IBEP721.sol	3e61cb926a68428c4aa547a10b912d0e08fb5fc7f33e77875a4ed48e8786c4c0
interfaces/IBEP721Metadata.sol	28bfdf127ede6fcc79c6d428e83b5f729f46eed2e36be466a3d850514de9cd5b
interfaces/IBEP721Receiver.sol	65da14007bd986bca6d87a1f597b5f0047f1b31cd6c9fa223413af57e45d78f9

interfaces/IHedgepieAdapterInfoBsc.sol	74d2551f93e3804cbe580739141546b6a eef37c911fe3ac428b07c1909595f73
interfaces/IHedgepieAdapterInfoEth.sol	ae5a8e0ee79af4dfa2e1e933b02517c72f 2363a456564b61ebedb07669d3cb60
interfaces/IHedgepieAdapterInfoMatic.sol	638608c956796fb0e4b2c5b91afc513770 a697b3158e226a156038e4bac89aa8
interfaces/IHedgepieInvestorBsc.sol	e814f41209bcea164fe3345a84bba7629d a0d187c92c652982b1fbc31c7fa664
interfaces/IHedgepieInvestorEth.sol	89ed6d73411d24b4d7eb1346e9fab7155 7eca978752a46e32f710e10d2862504
interfaces/IHedgepieInvestorMatic.sol	f464c54d10ab0e17e2cd25cededa1a3ac ad52fa10e282ec2f8a8eacc36e6eeac
interfaces/IPancakePair.sol	3b6fcda50c22a9db85abb57bcf2d87b4e 8ba7ca2e11d6366e34cc50499d6d236
interfaces/IPancakeRouter.sol	5b6a0ccde8aab4c23f0ee99a23148f832d c85570228e698e0ba9c6710b8b38be
interfaces/IRNG.sol	4de9b07c095e7edf6f80f2c33522a83620 165e4352833fd44db0a3def8abb235
interfaces/IVaultStrategy.sol	a1b933db9c687ca29ed66c0ea62bfd8ae 95abf12d18a6b0649e70ba9811a6695
interfaces/IWrap.sol	12228c23e9cc274bb322d165d596ccb97 c16be1aaa054d33bc6007e01145143d
interfaces/IYBNFT.sol	064e53be35ad75ad351476abadbddbbc 5e5d5b429724c31bf45b18c54d1021ff
interfaces/mock/IPancakeFarmMasterChef.sol	2874365e51a111866e8b6bfdb265f4167 7081419c5a2f4a51dfe10c7762940d0
interfaces/strategies/IPancakeswapStrategy.sol	b4ac909bafc7e637b846edb3ee517cff0e 11a1943305b0c4e65fdd56f74a33f3
interfaces/strategies/IVenusStrategy.sol	45a8ae93a5332af9946690de75e7c7e14 9151815112d006055f34e20334e8968

libraries/Address.sol	8591d74508d0e2526866a32fe460793bf149ae79338f847cdcbb50ffadc35953
libraries/Context.sol	b14f1609a44d1bc53805621e322cb609a510c86b22c9bc9cf1960e6adaf0fc0d
libraries/EnumerableSet.sol	88b261e7bf185d59e6f0a9d087cba9a1648ec53311a00f1f3189eea23a115ae5
libraries/FixedPoint.sol	0b381904ac838b09c8fbec7901aa44ce23939c4a0bad83852c38ecd5f0502d7d
libraries/HedgepieLibraryBsc.sol	714c7eadd94cb3855d8d09b1825d9606fc938d12e3590731d86bfa39062d5638
libraries/HedgepieLibraryEth.sol	7affb2f06c49aa7da9ecffc5ecffd7bcbbe0ed53e58426db73979221d066cd61
libraries/HedgepieLibraryMatic.sol	f2cb88eeba35ec628ed1105d0688b269b18482ef16f905ac720993778e9dd5a4
libraries/Ownable.sol	ccd7fffb22d8899cf2412b9b9e94faf8faacd6bd21c6c987637418245e2c13a1
libraries/SafeBEP20.sol	305ffde18d27c56ab1302e250156539f5d76a8316b53659773a7e89f6ee47a77
libraries/SafeMath.sol	48fc2979ca0b6fbec315cc2dc9ba86915f44de616f3f8d43c542c32dc1e12777
libraries/Strings.sol	1c14a4de4119fe78d8fbf3e8d0ae01f2cc1547ca8ebac77e2f35e0e33c4a9030
Multicall.sol	93441435df6d91d5fde8e6df8bfd2d37c74282b21706ddbeecef8d3d3638d165
type/AccessControl.sol	3982105b368b15fe22cc0c4a3fa6cec9d9c6bef4df67c9f069662bacb754a55b
type/AdminAccessRoles.sol	3b83a040d714ce41f07b7a554ed1f78862d5e6b6effd8a906b8cdc6068384b88
type/BEP165.sol	19aa178d1751cf4f7942ee644c0b2a6980e2d3f6e942d17ace5c82d244e7fc62

type/BEP20.sol	5dfebde7dc6d3ddcf3eb5d17df60fc4b309a3503c8c8a10232d50838fddcb568
type/BEP721.sol	e34d9a968d9bec98c9655b557a669a990ce30561a9a2694ca4832daf1377e3ea

Introduction

HedgePie is a platform that allows users to create and invest in hedge fund-like strategies using decentralized finance. It utilizes Non-Fungible Tokens (NFTs) to represent the strategies and assets being traded on the platform. Users can invest in a wide range of strategies, which are based on pools.

It is crucial for the platform to be configured properly in order for it to function as expected. A proper configuration can ensure the platform is secure and performs optimally. If the platform is not configured correctly, it can lead to security vulnerabilities and poor performance.

Roles

The HedgePie ecosystem consists of several different smart contracts that work together to enable the platform's functionality.

Info Contracts

The HedgePie Adapter Info contracts provide information about the underlying assets that are being traded on the platform. The information is only updated from the adapters, which are responsible for connecting the Non-Fungible Tokens (NFTs) to the strategies.

Managers are responsible for updating information related to the adapter, including trading volume and profit of the Adapter.

- `updateTVLInfo()`
- `updateTradedInfo()`
- `updateProfitInfo()`
- `updateParticipantInfo()`

Owner is responsible for configuring the manager of the contract.

- `setMetanager()`

Manager Contracts

The HedgePieAdapterManager contracts are responsible for managing and configuring the adapter contracts on the HedgePie platform. It allows for the addition and removal of adapters and keeps a registry of the currently active adapters. It also keeps track of the investor contract, which handles interactions between investors and the adapters on the platform.

Owner is responsible for configuring the adapters and maintaining the investor address within the contract.

- `setAdapter()`

- `setInvestor()`

ActiveAdapter can access the corresponding strategy of an adapter.

- `getAdapterStrat()`

Public `getAdapters,` any user can view the adapters info.

Investor Contracts

The HedgePie Investor contract allows users to invest in and manage their investments in different strategies offered on the platform. These contracts include functionality such as the ability to deposit and withdraw funds, claim rewards, and view pending rewards.

Owner is responsible for configuring the Adapter Manager and maintaining the treasury address within the contract.

- `setAdapterManager()`
- `setTreasury()`

Valid NFTs allowing addresses to invest in a strategy. For actions such as depositing, withdrawing, and claiming rewards.

- `depositBNB()`
- `withdrawBNB()`
- `claim()`

Public

`pendingReward`, Any user can view the pending rewards of an investment.

Master Chef Contract

The HedgePie Master Chef contract is a staking contract that allows users to stake a certain amount of tokens to receive rewards and benefits in return. The contract is composed of the owner's role.

Onwer is responsible for updating the reward multiplier, updating the allocation points for specific pools, and adding new liquidity pools to the contract.

- `updateMultiplier()`
- `set()`
- `add()`

Public

Any user can:

- `poolLength()`, view the number of pools that are available on the platform.
- `getMultiplier()`, view the reward multiplier over a given range of blocks.
- `pendingReward()`, view pending rewards.
- `updatePool()`, update reward variables of the given pool.
- `massUpdatePools()`, update reward variables for all pools.
- `deposit()`, deposit tokens to a pool.
- `withdraw()`, withdraw tokens from a pool.
- `emergencyWithdraw()`, withdraw their staked tokens without caring about the rewards.

Token contract

The HedgePie Token contract handles the issuance and transfer of the platform's native token. It is composed of two roles: Admin Role and Minter Role.

Admin Role is for managing the contract, such as adding or removing minters.

Minter Role allows the minter to create new tokens and transfer them to other addresses.

- `mint()`

NFT contract

HedgePie YBNFT contract is used for the creation and management of unique token assets on the platform. It is composed of an NFT owner role.

NFT onwer

Any NFT owner can,

- `updatePerformanceFee()`, update the performance fee of adapters.
- `updateAllocations()`, update the strategy’s allocations.
- `updateTokenURI()`, update token URI of an NFT.

Public

- `mint()`, Any user can mint NFTS.

Contract Infrastructure Architecture Review

This section of the audit focuses on the review of the platform's Infrastructure Architecture. The objective of this review is to assess the security and overall design of the contract infrastructure, including the management and storage of contracts, and any identified vulnerabilities or potential risks. The findings of this review will be used to make recommendations for improvement and to ensure the integrity and security of the HedgePie platform.

Libraries Dependency Review

The platform's contract infrastructure utilizes multiple similar libraries. This can potentially lead to issues such as increased security vulnerabilities, compatibility issues, and a lack of support. It also increases the risk of bugs and errors, which can impact the performance and reliability of the contract infrastructure.

```
import "@openzeppelin/contracts/access/Ownable.sol";
import "@openzeppelin/contracts/token/ERC20/IERC20.sol";
import "@openzeppelin/contracts/token/ERC20/utils/SafeERC20.sol";
import "./libraries/Ownable.sol";
import "./type/BEP721.sol";
import "./libraries/SafeBEP20.sol";
.
.
.
```

Recommendations

Using one library instead of multiple similar libraries throughout the platform is beneficial because it ensures consistency and predictability in the code. It is important to ensure that the use of multiple similar libraries is properly managed and that potential risks are identified and addressed in a timely manner to ensure the integrity and security of the platform.

Contracts Infrastructure Review

The platform infrastructure uses multiple contracts with the same functionality for different networks. This approach could lead to decreased code readability and maintenance issues.

```
HedgepieAdapterInfoEth.sol
HedgepieAdapterInfoMatic.sol
HedgepieAdapterManagerBsc.sol
HedgepieAdapterManagerEth.sol
HedgepieAdapterManagerMatic.sol
HedgepieInvestorBsc.sol
HedgepieInvestorEth.sol
HedgepieInvestorMatic.sol
```

Recommendations

It is recommended to evaluate the need for each contract and consider consolidating or removing unnecessary contracts to improve the overall efficiency and performance of the platform infrastructure.

Adapters Review

The HedgePie platform uses adapters to integrate various pools, decentralized exchanges, and ecosystems, allowing users to create and invest in new strategies.

All the adapters are configured in the same manner, which ensures consistency and predictability in the way the platform interacts with different underlying assets. This makes it easier for developers to understand and work with the code, and for users to understand how the platform operates. The Venus adapter is the only exception, as it utilizes the _repayAsset function which leaves 0.001% of the staked tokens in the strategy contract.

Recommendations

It is important that all the adapters are configured in the same way, so the platform can maintain its consistency and predictability, and to avoid any confusion or unexpected behavior. This will also help to improve the security and scalability of the platform, and make it more robust and reliable for users. By having a consistent and predictable way of managing the different adapters, the platform can ensure that all the users have a smooth and trustable experience.

Platform Integrated Adapters

Adapter Name	URL
Alpaca Finance	https://docs.alpacafinance.org/
ApeSwap	https://apeswap.gitbook.io/apeswap-finance/welcome/master
Autofarm Network	https://autofarm.gitbook.io/autofarm-network/
Beefy.com	https://docs.beefy.finance/
Belt.fi	https://docs.belt.fi/
BitSwap	https://www.bitswap.network/blog
PancakeSwap	https://docs.pancakeswap.finance/get-started
Venus Protocol	https://docs.venus.io/docs/getstarted#introduction
Aave	https://docs.aave.com/hub/
Balancer	https://docs.balancer.fi/
Compound	https://docs.compound.finance/
Curve Finance	https://resources.curve.fi/
Pickle Finance	https://docs.pickle.finance/
SushiSwap	https://docs.sushi.com/
Yearn.finance	https://docs.yearn.finance/
QuickSwap	https://docs.quickswap.exchange/
Uniswap	https://docs.uniswap.org/

Test Deployments

Contract Name	Test Deploy
HedgepieLibraryEth	https://testnet.bscscan.com/address/0xCF8B7Cd25e16309D42168F77A0E49A368CB96AA1
HedgepieLibraryMatic	https://testnet.bscscan.com/address/0xAfCaaf96B6574B4e7C922D05D199E8f55ABB5cDc
HedgepieLibraryBsc	https://testnet.bscscan.com/address/0xf32E6913A64a6861183EA83D52BdBCA7e320CfFf
HedgepieYBNFT	https://testnet.bscscan.com/address/0xA99fc8bC9AdcE94C232945F1bEC3bEB88d739382
HedgepieToken	https://testnet.bscscan.com/address/0x47104f8EAb58d3C3957B249F90aD226FD6C34577
HedgepieMasterChef	https://testnet.bscscan.com/address/0x0dE858E2Aca15c89DB7DD261e821BE6eCE12858F
HedgepieAdapterInfoBsc	https://testnet.bscscan.com/address/0x91Ec1a3a50E5b50B3f0824968720635D22511ACf
HedgepieAdapterInfoEth	https://testnet.bscscan.com/address/0xd9651263079EF93b83B3617F035Fb3a4BFC3fE83
HedgepieAdapterInfoMatic	https://testnet.bscscan.com/address/0x9bb1AE5052E33EEd161fD29cAc3549CC4706a92A
HedgepieAdapterManagerBsc	https://testnet.bscscan.com/address/0x6e30038AF23aE4a5AFef3a77E8C9214cFA418D86
HedgepieAdapterManagerEth	https://testnet.bscscan.com/address/0xaebB837495D8744CC1bb00B03a6050FB16298BEa
HedgepieAdapterManagerMatic	https://testnet.bscscan.com/address/0x68272470caBd0E7dd6d1a23a5B5ed1Fe0F54C03B

HedgepieInvestorBsc	https://testnet.bscscan.com/address/0x57255E9c05857cDAd3016F23d5c1c07379efE611
HedgepieInvestorEth	https://testnet.bscscan.com/address/0x9068Cc9872BF3DcF62b6d30A1298BC89aF663f1A
HedgepieInvestorMatic	https://testnet.bscscan.com/address/0x1DdFC94Db944b456eB7338430C8c487AfD586148
AlpacaAUSDAdapter	https://testnet.bscscan.com/address/0x2f6Ca111639EDCF720cDf41487e51FE32CB42d89
AlpacaLendAdapter	https://testnet.bscscan.com/address/0x3E7a982993D2C33E42CCD6D64bf7828f72169bE2
AlpacaStakeAdapter	https://testnet.bscscan.com/address/0x050871f383c91B1520d0049Fd833BF10f5EfAE1a
ApeswapBananaAdapter	https://testnet.bscscan.com/address/0x61503391733F2486A85F73E01BD5C627896A45ab
ApeswapFarmLPAdapter	https://testnet.bscscan.com/address/0x37b83F80CD0aaC4B0697030010c3aaf8C0D7eBe8
ApeswapJungleAdapter	https://testnet.bscscan.com/address/0x15fCB8188BA8E70E5efA3a23CE79a1381F976235
ApeswapVaultAdapter	https://testnet.bscscan.com/address/0xB06BdaEA4DEe9d17a9Bd0E1cfd5768e6881D158B
AutoVaultAdapterBsc	https://testnet.bscscan.com/address/0x78FFE91A23abFb0dC03166211034BF6cac14371A
BeefyVaultAdapter	https://testnet.bscscan.com/address/0xb8CBD242296C67299C30F5F85782f6b7814Fa2b5
BeltVaultAdapter	https://testnet.bscscan.com/address/0xE5B054Cbb584841078151B745cDC4bD6A8D6a994
BiSwapFarmLPAdapterBsc	https://testnet.bscscan.com/address/0x5085536616e642588797Ed851631BaA37019aE39

PancakeSwapFarmLPAdapterBsc	https://testnet.bscscan.com/address/0x5Eca1148e109fcE4106215b601d679caC532c69c
PancakeStakeAdapterBsc	https://testnet.bscscan.com/address/0xb7A3C235D439CC7f14418BC2dDe1F5c8068Eb3AE
VenusAdapterMock	https://testnet.bscscan.com/address/0x9202F1583F8486f250841197cE33A846B2FB429d
VenusLendAdapterBsc	https://testnet.bscscan.com/address/0xE9379DC656934576a77522b8b428499fb9B1ab08
VenusLevAdapterBsc	https://testnet.bscscan.com/address/0xa3734e321e78D6e3a2856B6E07Be725BDcE8C3B9
AaveLendAdapterEth	https://testnet.bscscan.com/address/0x74CD5D6b9f2a70378d1FbfAa6e08fE9b4670A05A
BalancerVaultAdapterEth	https://testnet.bscscan.com/address/0xbf45f67B772A977c0a08B4354FB09Ec49ea0B296
CompoundLendAdapterEth	https://testnet.bscscan.com/address/0xD7Bac6Bd182bdFC1D348791f686cF812564D8751
CurveGaugeAdapter	https://testnet.bscscan.com/address/0x16FFb64eAcC3D6bCa4971c2Eb8683feBAEeB2DA1
PickleCurveGaugeAdapter	https://testnet.bscscan.com/address/0x9237e1A857aB3EA8Be266C069084eB1E0EBAC6a3
PickleSingleGaugeAdapter	https://testnet.bscscan.com/address/0xc192BC540d62F4daEA44393a6BCB0a057A85D869
PickleSushiGaugeAdapter	https://testnet.bscscan.com/address/0x139050D1ef612AAC19EC7b41B3f08215FbE22c19
PickleSushiMasterAdapter	https://testnet.bscscan.com/address/0x51495B0031DA9D837048CF8E688b7E9739085449
SushiFarmAdapterEth	https://testnet.bscscan.com/address/0x983e02b80d1B8eD008B01F2Ba099eb5fa884af22

SushiFarmV2AdapterEth	https://testnet.bscscan.com/address/0x5f46Bbc429877f368c72e4a3E2209F135B8429bc
UniswapV3LPAdapter	https://testnet.bscscan.com/address/0x06ac198bcc771105ca0C781b6b5ef9cFd0e0C227
YearnCurveAdapter	https://testnet.bscscan.com/address/0x8e8D8d6f892FaC9F82534003b4C62096148D0348
YearnSingleAdapter	https://testnet.bscscan.com/address/0xF92D5E4C5Aa1e683e25c3a6B284Bf4A887866333
AaveMarketV2AdapterMatic	https://testnet.bscscan.com/address/0xAF3604D2557f6D71D33Ff14DE11b0A7C9a5CA209
AaveMarketV3AdapterMatic	https://testnet.bscscan.com/address/0x4a7b78D607C2CECE097E2BbfD95A96ea6461015e
ApeswapFarmAdapter	https://testnet.bscscan.com/address/0x4568370179Ce46635e62e8Ce2C140eA73DA2f57A
BeefyBalancerAdapter	https://testnet.bscscan.com/address/0x3E3Ec9f7Ba7Ab7C15B9A8E560de9021eDdc856D2
BeefyStargateAdapter	https://testnet.bscscan.com/address/0x69124B15Edf83513814623EE164eC38561275763
BeefyVaultAdapterMatic	https://testnet.bscscan.com/address/0xdd811cFCf37Ca56F765a9185327Ec792Eb054728
CurveLPAdapter	https://testnet.bscscan.com/address/0x8Ae3cEC143b550c39d169c52988ea85265EC0027
QuickLPDualAdapter	https://testnet.bscscan.com/address/0xDc2A08F0c720A11387e0Bc2559D9bFD0DF53d264
QuickLPFarmAdapter	https://testnet.bscscan.com/address/0x3E669484DB1b3a45402E9F246DcD7829Cd92Cd35
QuickStakeAdapter	https://testnet.bscscan.com/address/0x796428bd64839f55FCf30fcA720C53ee59f46A87

StargateFarmAdapterMatic	https://testnet.bscscan.com/address/0x8f4Ccc4906d78754EF320d940d0d030C3052Db32
SushiSwapLPAdapterMatic	https://testnet.bscscan.com/address/0x366A7b8e5D767d865c8bE8037b293bbc89b01400
UniswapLPAdapter	https://testnet.bscscan.com/address/0x498E8Ec2599865d0380c1493B185D1D8a919F467

Analysis

● Critical ● Medium ● Minor / Informative ● Pass

Severity	Code	Description	Status
●	MSI	Mocking Strategy Issue	Unresolved
●	RII	Reward Inconsistency Issue	Unresolved
●	UPEH	Underneath Protocols Error Handling	Unresolved
●	PSSI	Potential State Synchronization Inconsistency	Unresolved
●	RFA	Redundant Function Argument	Unresolved
●	DSM	Data Structure Misuse	Unresolved
●	CR	Code Repetition	Unresolved
●	EGU	Excessive Gas Usage	Unresolved
●	DDP	Decimal Division Precision	Unresolved
●	RDSF	Redundant Data Structure Fields	Unresolved
●	AAO	Accumulated Amount Overflow	Unresolved
●	MSC	Missing Sanity Check	Unresolved
●	RSML	Redundant SafeMath Library	Unresolved

●	L04	Conformance to Solidity Naming Conventions	Unresolved
●	L06	Missing Events Access Control	Unresolved
●	L07	Missing Events Arithmetic	Unresolved
●	L13	Divide before Multiply Operation	Unresolved
●	L14	Uninitialized Variables in Local Scope	Unresolved
●	L19	Stable Compiler Version	Unresolved

MSI - Mocking Strategy Issue

Criticality	Critical
Location	HedgepieYBNFT.sol#L85
Status	Unresolved

Description

The HedgePie platform utilizes Non-Fundible Tokens (NFTs) to represent the strategies. Users can mint their own NFTs with adapter addresses that are not part of the Hedgepie platform. As a result, the platform is vulnerable to strategy mocking.

```
function mint(
    uint256[] calldata _adapterAllocations,
    address[] calldata _adapterTokens,
    address[] calldata _adapterAddrs,
    uint256 _performanceFee,
    string memory _tokenURI
) external {
    . . .
}
```

Recommendation

It is recommended that the team thoroughly verify that the appropriate adapters are integrated into the platform's ecosystem.

RII - Reward Inconsistency Issue

Criticality	Critical
Location	adapters/BaseAdapterBsc.sol#L169 adapters/BaseAdapterEth.sol#L169 adapters/BaseAdapterMatic.sol#L171
Status	Unresolved

Description

The platform's rewards formula is based on the variable `accTokenPerShare`. This variable can only be changed through the deposit and withdraw functions. Hence, the variable `accTokenPerShare` is not being properly synced with the actual rewards from the corresponding strategy, resulting in inconsistencies between the actual and pending rewards.

For instance, let's examine two cases:

1. If one user participates in a strategy and no one else joins, the accTokenPerShare `adapterInfo.accTokenPerShare += (rewardAmt0 * 1e12) / adapterInfo.totalStaked;` will accumulate to zero. Since accTokenPerShare is only changed in the deposit and withdraw functions. After a month the claim function will return zero rewards to the user. However, the pending rewards will return the actual reward earned.

2. If 10 users participate in a strategy during the first quarter of the year, and then try to claim their rewards a year later, the claimable amount will not be consistent with the pending rewards. This is because the accTokenPerShare, which is used to calculate rewards, is only mutated in the deposit and withdraw functions and is not updated over time. As a result, the claimable amount will not reflect the actual rewards earned throughout the year and will result in an inconsistency between the claimable rewards and the pending rewards.

```
function claim(uint256 _tokenId, address _account)
    external
    payable
    virtual
    returns (uint256 amountOut)
{}

function pendingReward(uint256 _tokenId, address _account)
    external
    view
    virtual
    returns (uint256 reward)
{}
}
```

Recommendation

The contract could synchronize the corresponding `accTokenPerShare` before the reward claim process.

UPEH - Underneath Protocols Error Handling

Criticality	Minor / Informative
Status	Unresolved

Description

The adapters' main responsibility is to interact with the underneath protocols. Many of these protocols provide documentation about error handling. There are cases where the adapter does not handle potential errors. This may produce unexpected behavior since the adapter will wrongly assume that the process has been completed successfully.

For instance, The `VenusLevAdapterBsc` adapter calls the `redeemUnderlying()`. The `redeemUnderlying()` documentation returns 0 on success, otherwise a failure message.

We state that the Venus protocol is an example, the team could investigate the documentation about the error handling of all the underneath protocol methods.

Recommendation

The team is advised to properly handle the errors of the underneath protocols according to the documentation to ensure that the adapter will behave as expected.

PSSI - Potential State Synchronization Inconsistency

Criticality	Minor / Informative
Status	Unresolved

Description

The adapters heavily depend on the underneath implementations. These implementations are using local variables in order to be synchronized with the underneath contracts. In many cases, the underneath contracts provide the required state. Since the adapter can access the required information from the underneath contract, then the local variable may produce an inconsistency between the actual state and the real state.

For instance, the `VenusLevAdapterBsc` is using a local variable called `isEntered` that determines if the adapter has entered the Venus market. The Venus controller implements a method called `checkMembership` that determines if an account has entered a specific market.

```
ComptrollerInterface(comptroller).checkMembership(msg.sender, strategy)
```

We state that the Venus protocol is an example, the team could investigate all the possible variables that could be provided by the underneath implementations.

Recommendation

The team is advised to check the underneath protocol state rather than the internal state. This will prevent inconsistency issues that may be produced by potential upgrades or changes of the underneath implementations.

RFA - Redundant Function Argument

Criticality	Minor / Informative
Location	adapters/BaseAdapterBsc.sol#L157 adapters/BaseAdapterEth.sol#L134 adapters/BaseAdapterMatic.sol#L136
Status	Unresolved

Description

In the deposit function, the `_amountIn` variable is used to represent the `msg.value`. As a result the argument `_amountIn` is redundant.

```
function deposit(
    uint256 _tokenId,
    uint256 _amountIn,
    address _account
) external payable virtual returns (uint256 amountOut) {}
```

Recommendation

It is recommended to eliminate the redundant variable from the arguments in the deposit function, as it is already represented by msg.value.

DSM - Data Structure Misuse

Criticality	Minor / Informative
Location	HedgepieYBNFT.sol#L12,25,138
Status	Unresolved

Description

The contract uses variables that are created and modified within the Adapter structure, but the business logic of the contract does not necessitate iterating through the `Adapter` array of the `adapterInfo` mapping to update these variables `modified, created`. This leads to excessive gas consumption due to the linear storage of the modified and created data.

```
struct Adapter {
    uint256 allocation;
    address token;
    address addr;
    uint96 created;
    uint96 modified;
}

mapping(uint256 => Adapter[]) public adapterInfo;

for (uint256 i; i < adapterInfo[_tokenId].length; i++) {
    adapterInfo[_tokenId][i].modified = uint96(block.timestamp);
}
```

Recommendation

The contract could use a data structure that provides instant access. For instance, a Map would fit better to the business logic of the contract. This way the time complexity will be reduced from o(n) to o(1).

CR - Code Repetition

Criticality	Minor / Informative
Location	adapters/BaseAdapterBsc.sol#L134,145 adapters/BaseAdapterEth.sol#L134,145 adapters/BaseAdapterMatic.sol#L136,147
Status	Unresolved

Description

The contract includes repetitive code blocks in the deposit and withdraw functions of every adapter in the ecosystem. There are potential issues that can arise when using code segments in Solidity. Some of them can lead to issues like gas efficiency, complexity, readability, security, and maintainability of the source code. It is generally a good idea to try to minimize code repetition where possible.

```
//deposit
IHedgepieAdapterInfoBsc(adapterInfoBscAddr).updateTVLInfo(
    _tokenId,
    _amountIn,
    true
);
IHedgepieAdapterInfoBsc(adapterInfoBscAddr).updateTradedInfo(
    _tokenId,
    _amountIn,
    true
);
IHedgepieAdapterInfoBsc(adapterInfoBscAddr).updateParticipantInfo(
    _tokenId,
    _account,
    true
);
//withdraw
IHedgepieAdapterInfoBsc(adapterInfoBscAddr).updateTVLInfo(
    _tokenId,
    userInfo.invested,
    false
);
IHedgepieAdapterInfoBsc(adapterInfoBscAddr).updateTradedInfo(
    _tokenId,
    userInfo.invested,
    true
);
IHedgepieAdapterInfoBsc(adapterInfoBscAddr).updateParticipantInfo(
    _tokenId,
    _account,
    false
);
```

Recommendation

The team is advised to avoid repeating the same code in multiple places, which can make the contract easier to read and maintain. The authors could try to reuse code wherever possible, as this can help to reduce the complexity and size of the contract. For instance, the AdapterInfo contract could utilize the common code segments in an function in order to avoid repeating the same code in multiple places.

DDP - Decimal Division Precision

Criticality	Minor / Informative
Location	adapters/eth/uniswap/uniswap-lp-v3-adapter.sol#L224 adapters/polygon/uniswap/uniswap-lp-adapter.sol#L216
Status	Unresolved

Description

Division of decimal (fixed point) numbers can result in rounding errors due to the way that division is implemented in Solidity. Thus, it may produce issues with precise calculations with decimal numbers.

Solidity represents decimal numbers as integers, with the decimal point implied by the number of decimal places specified in the type (e.g. decimal with 18 decimal places). When a division is performed with decimal numbers, the result is also represented as an integer, with the decimal point implied by the number of decimal places in the type. This can lead to rounding errors, as the result may not be able to be accurately represented as an integer with the specified number of decimal places.

Hence, the splitted shares will not have the exact precision and some funds may not be calculated as expected.

```
tokenAmount[0] = _swapAndApprove(tokens[0], _amountIn / 2);
tokenAmount[1] = _swapAndApprove(tokens[1], _amountIn / 2);
```

Recommendation

The contract could calculate the subtraction of the divided funds in the last calculation in order to avoid the division rounding issue.

RDSF - Redundant Data Structure Fields

Criticality	Minor / Informative
Location	adapters/BaseAdapterMatic.sol#L7 HedgepieAdapterManagerEth.sol#L7 HedgepieAdapterManagerMatic.sol#L7
Status	Unresolved

Description

The contract employs the `AdapterInfo` structure to store adapter data. This structure contains the adapter name and its staking token address. However, as the adapter already holds this information, the name and staking fields in the structure are unnecessary and redundant.

```
struct AdapterInfo {
    address addr;
    string name;
    address stakingToken;
    bool status;
}
```

Recommendation

It is recommended to remove redundant data from smart contracts as it can optimize their performance and reduce the overall size of the contract. Removing unnecessary data structures and variables can make the contract more efficient and easier to understand. By eliminating redundant data, the contract will require less storage space, and less gas to execute the function.

AAO - Accumulated Amount Overflow

Criticality	Minor / Informative
Location	adapters/BaseAdapterBsc.sol#L14 adapters/BaseAdapterEth.sol#L14 adapters/BaseAdapterMatic.sol#L14 HedgepieAdapterInfoBsc.sol#L7 HedgepieAdapterInfoEth.sol#L7 HedgepieAdapterInfoMatic.sol#L7 HedgepieMasterChef.sol#L21,46
Status	Unresolved

Description

The contract is using variables to accumulate values. The contract could lead to an overflow when the total value of a variable exceeds the maximum value that can be stored in that variable's data type. This can happen when an accumulated value is updated repeatedly over time, and the value grows beyond the maximum value that can be represented by the data type.

```
struct AdapterInfo {
    uint256 accTokenPerShare; // Accumulated per share for first reward
token
    uint256 accTokenPerShare1; // Accumulated per share for second
reward token
    uint256 totalStaked; // Total staked staking token
}

struct AdapterInfo {
    ...
    ...
    uint256 traded;
    uint256 profit;
}

struct PoolInfo {
    ...
    ...
    uint256 accHpiePerShare;
    uint256 totalShares;
}

uint256 public totalAllocPoint = 0;
```

Recommendation

The team is advised to carefully investigate the usage of the variables that accumulate value. A suggestion is to add checks to the code to ensure that the value of a variable does not exceed the maximum value that can be stored in its data type.

MSC - Missing Sanity Check

Criticality	Minor / Informative
Status	Unresolved

Description

The Hedgepie contract does not adequately verify the initialized address in the adapters' constructor. If the adapter addresses are not initialized correctly, the adapter will not function as intended.

```
constructor(
    uint256 _pid,
    address _strategy,
    address _vStrategy,
    address _stakingToken,
    address _router,
    address _swapRouter,
    address _wbnb,
    string memory _name
) {

constructor(
    address _strategy,
    address _stakingToken,
    address _repayToken,
    address _swapRouter,
    address _wbnb,
    string memory _name
) {

.
.
.
```

The arguments `_lower` and `_upper` are not properly sanitized. The `_lower` variable can be set to values greater than `_upper`. If the tick values are not initialized correctly, the adapter will not function as intended.

```
constructor(
    address _strategy,
    address _stakingToken,
    address _router,
    int24 _lower,
    int24 _upper,
    address _weth,
    string memory _name
)
```

Recommendation

It is recommended that the Hedgepie contracts implement a proper address initialization check in the constructor to ensure that the adapter addresses and variables are correct. By adding a verification process, the contract can ensure that the adapters are set up correctly and will function as intended.

RSML - Redundant SafeMath Library

Criticality	Minor / Informative
Location	HedgepieMasterChef.sol#L11,12 HedgepieToken.sol#L12
Status	Unresolved

Description

SafeMath is a popular Solidity library that provides a set of functions for performing common arithmetic operations in a way that is resistant to integer overflows and underflows.

Starting with Solidity versions that are greater than or equal to 0.8.0, the arithmetic operations revert on underflow and overflow. As a result, the native functionality of the Solidity operations replaces the SafeMath library. Hence, the usage of the SafeMath library adds complexity, overhead and increases unnecessarily the gas consumption.

```
using SafeMath for uint256;
using SafeBEP20 for IBEP20;
```

Recommendation

The team is advised to remove the SafeMath library. Since the version of the contract is greater than `0.8.0` then the pure Solidity arithmetic operations produce the same result.

If the previous functionality is required, then the contract could exploit the `unchecked { ... }` statement.

Read more about the breaking change on https://docs.soliditylang.org/en/v0.8.16/080-breaking-changes.html#solidity-v0-8-0-breaking-changes.

L04 - Conformance to Solidity Naming Conventions

Criticality	Minor / Informative
Location	HedgepieYBNFT.sol#L49,61,74,86,87,88,89,90,127,149,150,173 HedgepieToken.sol#L22 HedgepieMasterChef.sol#L39,105,118,150,176,192,202,240,279,312 HedgepieInvestorMatic.sol#L77,110,134,160,183,194 HedgepieInvestorBsc.sol#L77,110,134,160,183,194 HedgepieAdapterManagerMatic.sol#L59,74,95,106 HedgepieAdapterManagerEth.sol#L59,74,95,106 HedgepieAdapterManagerBsc.sol#L59,74,95,106 HedgepieAdapterInfoEth.sol#L50,51,52,61,62,63,72,73,74,83,84,85,111 HedgepieAdapterInfoBsc.sol#L50,51,52,61,62,63,72,73,74,83,84,85,111
Status	Unresolved

Description

The Solidity style guide is a set of guidelines for writing clean and consistent Solidity code. Adhering to a style guide can help improve the readability and maintainability of the Solidity code, making it easier for others to understand and work with.

The followings are a few key points from the Solidity style guide:

1. Use camelCase for function and variable names, with the first letter in lowercase (e.g., myVariable, updateCounter).
2. Use PascalCase for contract, struct, and enum names, with the first letter in uppercase (e.g., MyContract, UserStruct, ErrorEnum).
3. Use uppercase for constant variables and enums (e.g., MAX_VALUE, ERROR_CODE).
4. Use indentation to improve readability and structure.
5. Use spaces between operators and after commas.
6. Use comments to explain the purpose and behavior of the code.
7. Keep lines short (around 120 characters) to improve readability.

```
uint256 _tokenId
uint256[] calldata _adapterAllocations
address[] calldata _adapterTokens
address[] calldata _adapterAddrs
uint256 _performanceFee
string memory _tokenURI
address _to
uint256 _amount
uint256 public BONUS_MULTIPLIER = 100
uint256 _from
uint256 _to
uint256 _pid
address _user
uint256 _allocPoint

...
```

Recommendation

By following the Solidity naming convention guidelines, the codebase increased the readability, maintainability, and makes it easier to work with.

Find more information on the Solidity documentation https://docs.soliditylang.org/en/v0.8.17/style-guide.html#naming-convention.

L06 - Missing Events Access Control

Criticality	Minor / Informative
Location	HedgepieAdapterManagerMatic.sol#L108 HedgepieAdapterManagerEth.sol#L108 HedgepieAdapterManagerBsc.sol#L108
Status	Unresolved

Description

Events are a way to record and log information about changes or actions that occur within a contract. They are often used to notify external parties or clients about events that have occurred within the contract, such as the transfer of tokens or the completion of a task. There are functions that have no event emitted, so it is difficult to track off-chain changes.

```
investor = _investor
```

Recommendation

To avoid this issue, it's important to carefully design and implement the events in a contract, and to ensure that all required events are included. It's also a good idea to test the contract to ensure that all events are being properly triggered and logged.

By including all required events in the contract and thoroughly testing the contract's functionality, the contract ensures that it performs as intended and does not have any missing events that could cause issues.

L07 - Missing Events Arithmetic

Criticality	Minor / Informative
Location	HedgepieMasterChef.sol#L158,181,194
Status	Unresolved

Description

Events are a way to record and log information about changes or actions that occur within a contract. They are often used to notify external parties or clients about events that have occurred within the contract, such as the transfer of tokens or the completion of a task.

It's important to carefully design and implement the events in a contract, and to ensure that all required events are included. It's also a good idea to test the contract to ensure that all events are being properly triggered and logged.

```
totalAllocPoint = totalAllocPoint.add(_allocPoint)

totalAllocPoint = totalAllocPoint.sub(prevAllocPoint).add(
                _allocPoint
            )
BONUS_MULTIPLIER = _multiplierNumber
```

Recommendation

By including all required events in the contract and thoroughly testing the contract's functionality, the contract ensures that it performs as intended and does not have any missing events that could cause issues with its arithmetic.

L13 - Divide before Multiply Operation

Criticality	Minor / Informative
Location	HedgepieMasterChef.sol#L132,136,213,217
Status	Unresolved

Description

It is important to be aware of the order of operations when performing arithmetic calculations. This is especially important when working with large numbers, as the order of operations can affect the final result of the calculation. Performing divisions before multiplications may cause loss of prediction.

```
uint256 hpieReward = multiplier
                .mul(rewardPerBlock)
                .mul(pool.allocPoint)
                .div(totalAllocPoint)
accHpiePerShare = accHpiePerShare.add(
                hpieReward.mul(1e12).div(lpSupply)
            )
```

Recommendation

To avoid this issue, it is recommended to carefully consider the order of operations when performing arithmetic calculations in Solidity. It's generally a good idea to use parentheses to specify the order of operations. The basic rule is that the multiplications should be prior to the divisions.

L14 - Uninitialized Variables in Local Scope

Criticality	Minor / Informative
Location	HedgepieYBNFT.sol#L138,162,179,235 HedgepieInvestorMatic.sol#L92,120,144,171 HedgepieInvestorBsc.sol#L92,120,144,171
Status	Unresolved

Description

Using an uninitialized local variable can lead to unpredictable behavior and potentially cause errors in the contract. It's important to always initialize local variables with appropriate values before using them.

```
uint256 i
uint8 i
```

Recommendation

By initializing local variables before using them, the contract ensures that the functions behave as expected and avoid potential issues.

L19 - Stable Compiler Version

Criticality	Minor / Informative
Location	HedgepieYBNFT.sol#L2 HedgepieToken.sol#L2 HedgepieMasterChef.sol#L2 HedgepieInvestorMatic.sol#L2 HedgepieInvestorBsc.sol#L2 HedgepieAdapterManagerMatic.sol#L2 HedgepieAdapterManagerEth.sol#L2 HedgepieAdapterManagerBsc.sol#L2 HedgepieAdapterInfoEth.sol#L2 HedgepieAdapterInfoBsc.sol#L2
Status	Unresolved

Description

The `^` symbol indicates that any version of Solidity that is compatible with the specified version (i.e., any version that is a higher minor or patch version) can be used to compile the contract. The version lock is a mechanism that allows the author to specify a minimum version of the Solidity compiler that must be used to compile the contract code. This is useful because it ensures that the contract will be compiled using a version of the compiler that is known to be compatible with the code.

```
pragma solidity ^0.8.4;
```

Recommendation

The team is advised to lock the pragma to ensure the stability of the codebase. The locked pragma version ensures that the contract will not be deployed with an unexpected version. An unexpected version may produce vulnerabilities and undiscovered bugs. The compiler should be configured to the lowest version that provides all the required functionality for the codebase. As a result, the project will be compiled in a well-tested LTS (Long Term Support) environment.

Functions Analysis

Contract	Type	Bases		
	Function Name	**Visibility**	**Mutability**	**Modifiers**
BaseAdapterBsc	Implementation	Ownable		
	getPaths	Public		-
	setPath	External	✓	onlyOwner
	setInvestor	External	✓	onlyOwner
	deposit	External	Payable	-
	withdraw	External	Payable	-
	claim	External	Payable	-
	pendingReward	External		-
BaseAdapterEth	Implementation	Ownable		
	getPaths	Public		-
	setPath	External	✓	onlyOwner
	setInvestor	External	✓	onlyOwner
	deposit	External	Payable	-
	withdraw	External	Payable	-
	claim	External	Payable	-
	pendingReward	External		-
BaseAdapterMatic	Implementation	Ownable		
	getPaths	Public		-
	setPath	External	✓	onlyOwner
	setInvestor	External	✓	onlyOwner
	deposit	External	Payable	-

	withdraw	External	Payable	-
	claim	External	Payable	-
	pendingReward	External		-
IStrategy	Interface			
	deposit	External	Payable	-
	withdraw	External	✓	-
	totalSupply	External		-
	totalToken	External		-
AlpacaAUSDAdapter	Implementation	BaseAdapterBsc		
		Public	✓	-
	deposit	External	Payable	onlyInvestor
	withdraw	External	Payable	onlyInvestor
	pendingReward	External		-
		External	Payable	-
IStrategy	Interface			
	deposit	External	Payable	-
	withdraw	External	✓	-
	totalSupply	External		-
	totalToken	External		-
AlpacaLendAdapter	Implementation	BaseAdapterBsc		
		Public	✓	-
	deposit	External	Payable	onlyInvestor
	withdraw	External	Payable	onlyInvestor
	pendingReward	External		-
		External	Payable	-

IFairLaunch	Interface			
	deposit	External	✓	-
	withdraw	External	✓	-
	pendingAlpaca	External		-
AlpacaStakeAdapter	Implementation	BaseAdapterBsc		
		Public	✓	-
	_getWrapToken	Internal	✓	
	_unwrapToken	Internal	✓	
	deposit	External	Payable	onlyInvestor
	withdraw	External	Payable	onlyInvestor
	claim	External	Payable	onlyInvestor
	pendingReward	External		-
		External	Payable	-
IStrategy	Interface			
	enterStaking	External	✓	-
	leaveStaking	External	✓	-
	pendingCake	External		-
ApeswapBananaAdapter	Implementation	BaseAdapterBsc		
		Public	✓	-
	deposit	External	Payable	onlyInvestor
	withdraw	External	Payable	onlyInvestor
	pendingReward	External		-
		External	Payable	-
IStrategy	Interface			

	deposit	External	✓	-
	withdraw	External	✓	-
	pendingCake	External		-
ApeswapFarmLPAdapter	Implementation	BaseAdapterBsc		
		Public	✓	-
	deposit	External	Payable	onlyInvestor
	withdraw	External	Payable	onlyInvestor
	claim	External	Payable	onlyInvestor
	pendingReward	External		-
		External	Payable	-
IStrategy	Interface			
	deposit	External	✓	-
	withdraw	External	✓	-
	pendingReward	External		-
ApeswapJungleAdapter	Implementation	BaseAdapterBsc		
		Public	✓	-
	deposit	External	Payable	onlyInvestor
	withdraw	External	Payable	onlyInvestor
	claim	External	Payable	onlyInvestor
	pendingReward	External		-
		External	Payable	-
IStrategy	Interface			
	deposit	External	✓	-
	withdraw	External	✓	-
	userInfo	External		-

IVStrategy	Interface			
	BANANA_VAULT	External		-
IVault	Interface			
	getPricePerFullShare	External		-
ApeswapVault Adapter	Implementation	BaseAdapterBsc		
		Public	✓	-
	deposit	External	Payable	onlyInvestor
	withdraw	External	Payable	onlyInvestor
	pendingReward	External		-
		External	Payable	-
IStrategy	Interface			
	pendingAUTO	External		-
	userInfo	External		-
	deposit	External	✓	-
	withdraw	External	✓	-
AutoVaultAdapterBsc	Implementation	BaseAdapterBsc		
		Public	✓	-
	deposit	External	Payable	onlyInvestor
	withdraw	External	Payable	onlyInvestor
	pendingReward	External		-
		External	Payable	-
IStrategy	Interface			
	deposit	External	✓	-

	withdraw	External	✓	-
	balance	External		-
	totalSupply	External		-
BeefyVaultAdapter	Implementation	BaseAdapterBsc		
		Public	✓	-
	deposit	External	Payable	onlyInvestor
	withdraw	External	Payable	onlyInvestor
	pendingReward	External		-
		External	Payable	-
IStrategy	Interface			
	deposit	External	Payable	-
	deposit	External	✓	-
	withdraw	External	✓	-
	withdrawBNB	External	✓	-
	balance	External		-
	totalSupply	External		-
BeltVaultAdapter	Implementation	BaseAdapterBsc		
		Public	✓	-
	deposit	External	Payable	onlyInvestor
	withdraw	External	Payable	onlyInvestor
	pendingReward	External		-
		External	Payable	-
IStrategy	Interface			
	pendingBSW	External		-
	deposit	External	✓	-

	withdraw	External	✓	-
	enterStaking	External	✓	-
	leaveStaking	External	✓	-
BiSwapFarmLPAdapterBsc	Implementation	BaseAdapterBsc		
		Public	✓	-
	deposit	External	Payable	onlyInvestor
	withdraw	External	Payable	onlyInvestor
	claim	External	Payable	onlyInvestor
	pendingReward	External		-
		External	Payable	-
IStrategy	Interface			
	pendingCake	External		-
	deposit	External	✓	-
	withdraw	External	✓	-
PancakeSwapFarmLPAdapterBsc	Implementation	BaseAdapterBsc		
		Public	✓	-
	deposit	External	Payable	onlyInvestor
	withdraw	External	Payable	onlyInvestor
	claim	External	Payable	onlyInvestor
	pendingReward	External		-
		External	Payable	-
IStrategy	Interface			
	pendingReward	External		-
	deposit	External	✓	-

	withdraw	External	✓	-
PancakeStakeAdapterBsc	Implementation	BaseAdapterBsc		
		Public	✓	-
	deposit	External	Payable	onlyInvestor
	withdraw	External	Payable	onlyInvestor
	claim	External	Payable	onlyInvestor
	pendingReward	External		-
		External	Payable	-
ComptrollerInterfaceG1	Interface			
	enterMarkets	External	✓	-
	exitMarket	External	✓	-
	mintAllowed	External	✓	-
	mintVerify	External	✓	-
	redeemAllowed	External	✓	-
	redeemVerify	External	✓	-
	borrowAllowed	External	✓	-
	borrowVerify	External	✓	-
	repayBorrowAllowed	External	✓	-
	repayBorrowVerify	External	✓	-
	liquidateBorrowAllowed	External	✓	-
	liquidateBorrowVerify	External	✓	-
	seizeAllowed	External	✓	-
	seizeVerify	External	✓	-
	transferAllowed	External	✓	-
	transferVerify	External	✓	-
	liquidateCalculateSeizeTokens	External		-
	setMintedVAIOf	External	✓	-

ComptrollerInterfaceG2	Interface	ComptrollerInterfaceG1		
	liquidateVAICalculateSeizeTokens	External		-
ComptrollerInterface	Interface	ComptrollerInterfaceG2		
IVAIVault	Interface			
	updatePendingRewards	External	✓	-
IComptroller	Interface			
	liquidationIncentiveMantissa	External		-
	treasuryAddress	External		-
	treasuryPercent	External		-
InterestRateModel	Interface			
	getBorrowRate	External		-
	getSupplyRate	External		-
VBep20Interface	Interface	IERC20		
	mint	External	✓	-
	mintBehalf	External	✓	-
	redeem	External	✓	-
	redeemUnderlying	External	✓	-
	borrow	External	✓	-
	repayBorrow	External	✓	-
	repayBorrowBehalf	External	✓	-
	liquidateBorrow	External	✓	-
	isVToken	External		-

	underlying	External		-
	exchangeRateStored	External		-
	comptroller	External		-
	_addReserves	External	✓	-
VTokenInterface	Interface			
	transfer	External	✓	-
	transferFrom	External	✓	-
	approve	External	✓	-
	allowance	External		-
	balanceOf	External		-
	balanceOfUnderlying	External	✓	-
	getAccountSnapshot	External		-
	borrowRatePerBlock	External		-
	supplyRatePerBlock	External		-
	totalBorrowsCurrent	External	✓	-
	borrowBalanceCurrent	External	✓	-
	borrowBalanceStored	External		-
	exchangeRateCurrent	External	✓	-
	exchangeRateStored	External		-
	getCash	External		-
	accrueInterest	External	✓	-
	seize	External	✓	-
	_setPendingAdmin	External	✓	-
	_acceptAdmin	External	✓	-
	_setComptroller	External	✓	-
	_setReserveFactor	External	✓	-
	_reduceReserves	External	✓	-
	_setInterestRateModel	External	✓	-

VenusAdapterMock	Implementation	Ownable, Pausable, ReentrancyGuard		
		Public	✓	-
	_approveVToken	Internal	✓	
	supply	External	✓	onlyEOA whenNotPaused nonReentrant
	redeem	External	✓	onlyEOA whenNotPaused nonReentrant
	addVTokens	External	✓	onlyOwner
	pause	External	✓	onlyOwner
	unpause	External	✓	onlyOwner
IStrategy	Interface			
	mint	External	✓	-
	redeem	External	✓	-
VenusLendAdapterBsc	Implementation	BaseAdapterBsc		
		Public	✓	-
	deposit	External	Payable	onlyInvestor
	withdraw	External	Payable	onlyInvestor
		External	Payable	-
IStrategy	Interface			
	mint	External	✓	-
	redeem	External	✓	-
	redeemUnderlying	External	✓	-
	borrow	External	✓	-

	repayBorrow	External	✓	-
VenusLevAdapterBsc	Implementation	BaseAdapterBsc		
		Public	✓	-
	deposit	External	Payable	onlyInvestor
	withdraw	External	Payable	onlyInvestor
	_leverageAsset	Internal	✓	
	_repayAsset	Internal	✓	
		External	Payable	-
IStrategy	Interface			
	deposit	External	✓	-
	withdraw	External	✓	-
AaveLendAdapterEth	Implementation	BaseAdapterEth		
		Public	✓	-
	deposit	External	Payable	onlyInvestor
	withdraw	External	Payable	onlyInvestor
	claim	External	Payable	onlyInvestor
	pendingReward	External		-
		External	Payable	-
IStrategy	Interface			
	joinPool	External	Payable	-
	exitPool	External	Payable	-
BalancerVaultAdapterEth	Implementation	BaseAdapterEth		
		Public	✓	-

	deposit	External	Payable	onlyInvestor
	withdraw	External	Payable	onlyInvestor
		External	Payable	-
IStrategy	Interface			
	mint	External	✓	-
	redeem	External	✓	-
	exchangeRateStored	External		-
IComptroller	Interface			
	enterMarkets	External	✓	-
	exitMarket	External	✓	-
CompoundLendAdapterEth	Implementation	BaseAdapterEth		
		Public	✓	-
	deposit	External	Payable	onlyInvestor
	withdraw	External	Payable	onlyInvestor
	pendingReward	External		-
		External	Payable	-
IGauge	Interface			
	deposit	External	✓	-
	withdraw	External	✓	-
	integrate_fraction	External		-
IPool	Interface			
	add_liquidity	External	Payable	-
	add_liquidity	External	Payable	-
	add_liquidity	External	Payable	-

	remove_liquidity_one_coin	External	✓	-
IMint	Interface			
	mint	External	✓	-
	minted	External		-
CurveGaugeAdapter	Implementation	BaseAdapterEth		
		Public	✓	-
	_getCurveLP	Internal	✓	
	_removeCurveLP	Internal	✓	
	_getReward	Internal	✓	
	deposit	External	Payable	onlyInvestor
	withdraw	External	Payable	onlyInvestor
	claim	External	Payable	onlyInvestor
	pendingReward	External		-
		External	Payable	-
IStrategy	Interface			
	deposit	External	✓	-
	withdraw	External	✓	-
	getReward	External	✓	-
	earned	External		-
IJar	Interface			
	deposit	External	✓	-
	withdraw	External	✓	-
IPool	Interface			
	add_liquidity	External	Payable	-

	add_liquidity	External	Payable	-
	add_liquidity	External	Payable	-
	add_liquidity	External	Payable	-
	add_liquidity	External	Payable	-
	add_liquidity	External	Payable	-
	remove_liquidity_one_coin	External	✓	-
	remove_liquidity_one_coin	External	✓	-
PickleCurveGaugeAdapter	Implementation	BaseAdapterEth		
		Public	✓	-
	_getCurveLP	Internal	✓	
	_removeCurveLP	Internal	✓	
	_getReward	Internal	✓	
	deposit	External	Payable	onlyInvestor
	withdraw	External	Payable	onlyInvestor
	claim	External	Payable	onlyInvestor
	pendingReward	External		-
		External	Payable	-
IStrategy	Interface			
	deposit	External	✓	-
	withdraw	External	✓	-
	getReward	External	✓	-
	earned	External		-
IJar	Interface			
	deposit	External	✓	-
	withdraw	External	✓	-

PickleSingleGaugeAdapter	Implementation	BaseAdapterEth		
		Public	✓	-
	_getReward	Internal	✓	
	deposit	External	Payable	onlyInvestor
	withdraw	External	Payable	onlyInvestor
	claim	External	Payable	onlyInvestor
	pendingReward	External		-
		External	Payable	-
IStrategy	Interface			
	deposit	External	✓	-
	withdraw	External	✓	-
	getReward	External	✓	-
	earned	External		-
IJar	Interface			
	deposit	External	✓	-
	withdraw	External	✓	-
PickleSushiGaugeAdapter	Implementation	BaseAdapterEth		
		Public	✓	-
	_getReward	Internal	✓	
	deposit	External	Payable	onlyInvestor
	withdraw	External	Payable	onlyInvestor
	claim	External	Payable	onlyInvestor
	pendingReward	External		-
		External	Payable	-
IStrategy	Interface			

	deposit	External	✓	-
	withdraw	External	✓	-
	pendingPickle	External		-
IJar	Interface			
	deposit	External	✓	-
	withdraw	External	✓	-
PickleSushiMasterAdapter	Implementation	BaseAdapterEth		
		Public	✓	-
	deposit	External	Payable	onlyInvestor
	withdraw	External	Payable	onlyInvestor
	claim	External	Payable	onlyInvestor
	pendingReward	External		-
		External	Payable	-
IStrategy	Interface			
	deposit	External	✓	-
	withdraw	External	✓	-
	pendingSushi	External		-
SushiFarmAdapterEth	Implementation	BaseAdapterEth		
		Public	✓	-
	deposit	External	Payable	onlyInvestor
	withdraw	External	Payable	onlyInvestor
	claim	External	Payable	onlyInvestor
	pendingReward	External		-
		External	Payable	-

IStrategy	Interface			
	deposit	External	✓	-
	withdrawAndHarvest	External	✓	-
	pendingSushi	External		-
SushiFarmV2AdapterEth	Implementation	BaseAdapterEth		
		Public	✓	-
	deposit	External	Payable	onlyInvestor
	withdraw	External	Payable	onlyInvestor
	claim	External	Payable	onlyInvestor
	pendingReward	External		-
		External	Payable	-
UniswapV3LPAdapter	Implementation	BaseAdapterEth, IERC721Receiver		
		External	Payable	-
		Public	✓	-
	_swapAndApprove	Internal	✓	
	_removeRemain	Internal	✓	
	deposit	External	Payable	onlyInvestor
	withdraw	External	Payable	onlyInvestor
	_deposit	Internal	✓	
	_withdraw	Internal	✓	
	onERC721Received	External		-
IStrategy	Interface			
	deposit	External	✓	-
	withdraw	External	✓	-
	totalAssets	External		-

	totalSupply	External		-
IPool	Interface			
	add_liquidity	External	Payable	-
	add_liquidity	External	Payable	-
	add_liquidity	External	Payable	-
	add_liquidity	External	Payable	-
	add_liquidity	External	Payable	-
	add_liquidity	External	Payable	-
	remove_liquidity_one_coin	External	✓	-
	remove_liquidity_one_coin	External	✓	-
YearnCurveAdapter	Implementation	BaseAdapterEth		
		Public	✓	-
	_getCurveLP	Private	✓	
	_removeCurveLP	Private	✓	
	deposit	External	Payable	onlyInvestor
	withdraw	External	Payable	onlyInvestor
	pendingReward	External		-
		External	Payable	-
IStrategy	Interface			
	deposit	External	✓	-
	withdraw	External	✓	-
	totalAssets	External		-
	totalSupply	External		-
YearnSingleAdapter	Implementation	BaseAdapterEth		
		Public	✓	-

	deposit	External	Payable	onlyInvestor
	withdraw	External	Payable	onlyInvestor
	pendingReward	External		-
		External	Payable	-
IStrategy	Interface			
	deposit	External	✓	-
	withdraw	External	✓	-
AaveMarketV2 AdapterMatic	Implementation	BaseAdapterMatic		
		Public	✓	-
	deposit	External	Payable	onlyInvestor
	withdraw	External	Payable	onlyInvestor
	claim	External	Payable	onlyInvestor
	pendingReward	External		-
		External	Payable	-
IStrategy	Interface			
	supply	External	✓	-
	withdraw	External	✓	-
AaveMarketV3 AdapterMatic	Implementation	BaseAdapterMatic		
		Public	✓	-
	deposit	External	Payable	onlyInvestor
	withdraw	External	Payable	onlyInvestor
	claim	External	Payable	onlyInvestor
	pendingReward	External		-
		External	Payable	-

IStrategy	Interface			
	deposit	External	✓	-
	withdrawAndHarvest	External	✓	-
	harvest	External	✓	-
	pendingBanana	External		-
ApeswapFarmAdapter	Implementation	BaseAdapterMatic		
		Public	✓	-
	deposit	External	Payable	onlyInvestor
	withdraw	External	Payable	onlyInvestor
	claim	External	Payable	onlyInvestor
	pendingReward	External		-
		External	Payable	-
IAsset	Interface			
IStrategy	Interface			
	deposit	External	✓	-
	withdraw	External	✓	-
	balance	External		-
	totalSupply	External		-
IBalancerVault	Interface			
	getPoolTokens	External		-
	joinPool	External	✓	-
	exitPool	External	✓	-
BeefyBalancerAdapter	Implementation	BaseAdapterMatic		
		Public	✓	-

	_getBalancerLP	Internal	✓	
	_removeBalancerLP	Internal	✓	
	deposit	External	Payable	onlyInvestor
	withdraw	External	Payable	onlyInvestor
	pendingReward	External		-
		External	Payable	-
IStrategy	Interface			
	deposit	External	✓	-
	withdraw	External	✓	-
	balance	External		-
	totalSupply	External		-
IStargate	Interface			
	addLiquidity	External	✓	-
	instantRedeemLocal	External	✓	-
	totalSupply	External		-
	totalLiquidity	External		-
BeefyStargateAdapter	Implementation	BaseAdapterMatic		
		Public	✓	-
	_getStargate	Internal	✓	
	_removeStargate	Internal	✓	
	deposit	External	Payable	onlyInvestor
	withdraw	External	Payable	onlyInvestor
	pendingReward	External		-
		External	Payable	-
IStrategy	Interface			

	deposit	External	✓	-
	withdraw	External	✓	-
	balance	External		-
	totalSupply	External		-
BeefyVaultAdapterMatic	Implementation	BaseAdapterMatic		
		Public	✓	-
	deposit	External	Payable	onlyInvestor
	withdraw	External	Payable	onlyInvestor
	pendingReward	External		-
		External	Payable	-
IStrategy	Interface			
	deposit	External	✓	-
	withdraw	External	✓	-
	claim_rewards	External	✓	-
	claimable_reward	External		-
IPool	Interface			
	add_liquidity	External	Payable	-
	add_liquidity	External	Payable	-
	add_liquidity	External	Payable	-
	add_liquidity	External	Payable	-
	add_liquidity	External	Payable	-
	add_liquidity	External	Payable	-
	remove_liquidity_one_coin	External	✓	-
	remove_liquidity_one_coin	External	✓	-
CurveLPAdapter	Implementation	BaseAdapterMatic		

		Public	✓	-
	_getCurveLP	Internal	✓	
	_removeCurveLP	Internal	✓	
	_getReward	Internal	✓	
	deposit	External	Payable	onlyInvestor
	withdraw	External	Payable	onlyInvestor
	claim	External	Payable	onlyInvestor
	pendingReward	External		-
		External	Payable	-
IStrategy	Interface			
	stake	External	✓	-
	withdraw	External	✓	-
	getReward	External	✓	-
	earnedA	External		-
	earnedB	External		-
QuickLPDualAdapter	Implementation	BaseAdapterMatic		
		Public	✓	-
	_getReward	Internal	✓	
	deposit	External	Payable	onlyInvestor
	withdraw	External	Payable	onlyInvestor
	claim	External	Payable	onlyInvestor
	pendingReward	External		-
		External	Payable	-
IStrategy	Interface			
	stake	External	✓	-
	withdraw	External	✓	-

	getReward	External	✓	-
	earned	External		-
QuickLPFarm Adapter	Implementation	BaseAdapterMatic		
		Public	✓	-
	_getReward	Internal	✓	
	deposit	External	Payable	onlyInvestor
	withdraw	External	Payable	onlyInvestor
	claim	External	Payable	onlyInvestor
	pendingReward	External		-
		External	Payable	-
IStrategy	Interface			
	stake	External	✓	-
	withdraw	External	✓	-
	getReward	External	✓	-
	earned	External		-
QuickStakeAdapter	Implementation	BaseAdapterMatic		
		Public	✓	-
	_getReward	Internal	✓	
	deposit	External	Payable	onlyInvestor
	withdraw	External	Payable	onlyInvestor
	claim	External	Payable	onlyInvestor
	pendingReward	External		-
		External	Payable	-
IStrategy	Interface			
	pendingStargate	External		-

	balanceOf	External		-
	deposit	External	✓	-
	withdraw	External	✓	-
IProvider	Interface			
	instantRedeemLocal	External	✓	-
	addLiquidity	External	✓	-
StargateFarmAdapterMatic	Implementation	BaseAdapterMatic		
		Public	✓	-
	deposit	External	Payable	onlyInvestor
	withdraw	External	Payable	onlyInvestor
	claim	External	Payable	onlyInvestor
	pendingReward	External		-
		External	Payable	-
IStrategy	Interface			
	deposit	External	✓	-
	withdrawAndHarvest	External	✓	-
	pendingSushi	External		-
SushiSwapLPAdapterMatic	Implementation	BaseAdapterMatic		
		Public	✓	-
	deposit	External	Payable	onlyInvestor
	withdraw	External	Payable	onlyInvestor
	claim	External	Payable	onlyInvestor
	pendingReward	External		-
		External	Payable	-

UniswapLPAdapter	Implementation	BaseAdapterMatic, IERC721Receiver		
		External	Payable	-
		Public	✓	-
	_swapAndApprove	Internal	✓	
	_removeRemain	Internal	✓	
	deposit	External	Payable	onlyInvestor
	withdraw	External	Payable	onlyInvestor
	_deposit	Internal	✓	
	_withdraw	Internal	✓	
	onERC721Received	External		-
HedgepieAdapterInfoBsc	Implementation	Ownable		
	updateTVLInfo	External	✓	isManager
	updateTradedInfo	External	✓	isManager
	updateProfitInfo	External	✓	isManager
	updateParticipantInfo	External	✓	isManager
	setManager	External	✓	onlyOwner
	_emitEvent	Internal	✓	
HedgepieAdapterInfoEth	Implementation	Ownable		
	updateTVLInfo	External	✓	isManager
	updateTradedInfo	External	✓	isManager
	updateProfitInfo	External	✓	isManager
	updateParticipantInfo	External	✓	isManager
	setManager	External	✓	onlyOwner
	_emitEvent	Internal	✓	

HedgepieAdapterInfoMatic	Implementation	Ownable		
	updateTVLInfo	External	✓	isManager
	updateTradedInfo	External	✓	isManager
	updateProfitInfo	External	✓	isManager
	updateParticipantInfo	External	✓	isManager
	setManager	External	✓	onlyOwner
	_emitEvent	Internal	✓	
HedgepieAdapterManagerBsc	Implementation	Ownable		
	getAdapters	External		-
	getAdapterStrat	External		onlyActiveAdapter
	addAdapter	External	✓	onlyOwner
	setAdapter	External	✓	onlyOwner
	setInvestor	External	✓	onlyOwner
HedgepieAdapterManagerEth	Implementation	Ownable		
	getAdapters	External		-
	getAdapterStrat	External		onlyActiveAdapter
	addAdapter	External	✓	onlyOwner
	setAdapter	External	✓	onlyOwner
	setInvestor	External	✓	onlyOwner
HedgepieAdapterManagerMatic	Implementation	Ownable		
	getAdapters	External		-
	getAdapterStrat	External		onlyActiveAdapter

	addAdapter	External	✓	onlyOwner
	setAdapter	External	✓	onlyOwner
	setInvestor	External	✓	onlyOwner
HedgepieInvestorBsc	Implementation	Ownable, ReentrancyGuard		
		Public	✓	-
	depositBNB	External	Payable	nonReentrant onlyValidNFT
	withdrawBNB	External	✓	nonReentrant onlyValidNFT
	claim	External	✓	nonReentrant onlyValidNFT
	pendingReward	Public		-
	setAdapterManager	External	✓	onlyOwner
	setTreasury	External	✓	onlyOwner
		External	Payable	-
HedgepieInvestorEth	Implementation	Ownable, ReentrancyGuard		
		Public	✓	-
	depositETH	External	Payable	nonReentrant onlyValidNFT
	withdrawETH	External	✓	nonReentrant onlyValidNFT
	claim	External	✓	nonReentrant onlyValidNFT
	pendingReward	Public		-
	setAdapterManager	External	✓	onlyOwner
	setTreasury	External	✓	onlyOwner
		External	Payable	-
HedgepieInvestorMatic	Implementation	Ownable, Reentrancy		

		Guard		
		Public	✓	-
	depositMATIC	External	Payable	nonReentrant onlyValidNFT
	withdrawMATIC	External	✓	nonReentrant onlyValidNFT
	claim	External	✓	nonReentrant onlyValidNFT
	pendingReward	Public		-
	setAdapterManager	External	✓	onlyOwner
	setTreasury	External	✓	onlyOwner
		External	Payable	-
HedgepieMasterChef	Implementation	Ownable		
		Public	✓	-
	poolLength	External		-
	getMultiplier	Public		-
	pendingReward	External		-
	add	Public	✓	onlyOwner
	set	Public	✓	onlyOwner
	updateMultiplier	Public	✓	onlyOwner
	updatePool	Public	✓	-
	massUpdatePools	Public	✓	-
	deposit	Public	✓	-
	withdraw	Public	✓	-
	emergencyWithdraw	Public	✓	-
HedgepieToken	Implementation	AdminAccessRoles, BEP20		
		Public	✓	-
	mint	External	✓	onlyMintUser

	isCapReach	External		-
	maxCap	External		-
YBNFT	Implementation	BEP721, Ownable		
		Public	✓	BEP721
	getCurrentTokenId	Public		-
	getAdapterInfo	Public		-
	tokenURI	Public		-
	exists	Public		-
	mint	External	✓	-
	updatePerformanceFee	External	✓	-
	updateAllocations	External	✓	-
	updateTokenURI	External	✓	-
	_setTokenURI	Internal	✓	
	_setAdapterInfo	Internal	✓	
	_checkPercent	Internal		
IAdapter	Interface			
	getPaths	External		-
	stackWithdrawalAmounts	External		-
	DEEPTH	External		-
	isVault	External		-
	isEntered	External		-
	isLeverage	External		-
	borrowRate	External		-
	stakingToken	External		-
	strategy	External		-
	vStrategy	External		-
	pendingReward	External		-

	pendingShares	External		-
	name	External		-
	repayToken	External		-
	rewardToken	External		-
	wrapToken	External		-
	router	External		-
	getAdapterStrategy	External		-
	getWithdrawalAmount	External		-
	getInvestCallData	External		-
	getDevestCallData	External		-
	getEnterMarketCallData	External		-
	getLoanCallData	External		-
	getDeLoanCallData	External		-
	getReward	External		-
	increaseWithdrawalAmount	External	✓	-
	increaseWithdrawalAmount	External	✓	-
	setWithdrawalAmount	External	✓	-
	setIsEntered	External	✓	-
	setInvestor	External	✓	-
IAdapterBsc	Interface			
	getPaths	External		-
	stakingToken	External		-
	strategy	External		-
	name	External		-
	rewardToken	External		-
	rewardToken1	External		-
	router	External		-
	swapRouter	External		-

	deposit	External	Payable	-
	withdraw	External	Payable	-
	claim	External	Payable	-
	pendingReward	External		-
	adapterInfos	External		-
	userAdapterInfos	External		-
IAdapterEth	Interface			
	getPaths	External		-
	stakingToken	External		-
	strategy	External		-
	name	External		-
	rewardToken	External		-
	rewardToken1	External		-
	router	External		-
	swapRouter	External		-
	deposit	External	Payable	-
	withdraw	External	Payable	-
	claim	External	Payable	-
	pendingReward	External		-
	adapterInfos	External		-
	userAdapterInfos	External		-
IAdapterManager	Interface			
	getAdapterStrat	External		-
	getDepositCallData	External		-
	getWithdrawCallData	External		-
	getLoanCallData	External		-
	getDeLoanCallData	External		-

	getEnterMarketCallData	External		-
IAdapterManagerEth	Interface			
	getAdapterStrat	External		-
IAdapterManagerMatic	Interface			
	getAdapterStrat	External		-
	getDepositCallData	External		-
	getWithdrawCallData	External		-
	getRewardCallData	External		-
	getAddLiqCallData	External		-
	getRemoveLiqCallData	External		-
IAdapterMatic	Interface			
	getPaths	External		-
	stakingToken	External		-
	strategy	External		-
	name	External		-
	rewardToken	External		-
	rewardToken1	External		-
	router	External		-
	swapRouter	External		-
	deposit	External	Payable	-
	withdraw	External	Payable	-
	claim	External	Payable	-
	pendingReward	External		-
	adapterInfos	External		-
	userAdapterInfos	External		-

IBEP165	Interface			
	supportsInterface	External		-
IBEP20	Interface			
	totalSupply	External		-
	decimals	External		-
	symbol	External		-
	name	External		-
	getOwner	External		-
	balanceOf	External		-
	transfer	External	✓	-
	allowance	External		-
	approve	External	✓	-
	transferFrom	External	✓	-
IBEP721	Interface	IBEP165		
	balanceOf	External		-
	ownerOf	External		-
	safeTransferFrom	External	✓	-
	transferFrom	External	✓	-
	approve	External	✓	-
	getApproved	External		-
	setApprovalForAll	External	✓	-
	isApprovedForAll	External		-
	safeTransferFrom	External	✓	-
IBEP721Metadata	Interface	IBEP721		
	name	External		-
	symbol	External		-

	tokenURI	External		-
IBEP721Receiver	Interface			
	onBEP721Received	External	✓	-
IHedgepieAdapterInfoBsc	Interface			
	updateTVLInfo	External	✓	-
	updateTradedInfo	External	✓	-
	updateProfitInfo	External	✓	-
	updateParticipantInfo	External	✓	-
IHedgepieAdapterInfoEth	Interface			
	updateTVLInfo	External	✓	-
	updateTradedInfo	External	✓	-
	updateProfitInfo	External	✓	-
	updateParticipantInfo	External	✓	-
IHedgepieAdapterInfoMatic	Interface			
	updateTVLInfo	External	✓	-
	updateTradedInfo	External	✓	-
	updateProfitInfo	External	✓	-
	updateParticipantInfo	External	✓	-
IHedgepieInvestorBsc	Interface			
	ybnft	External		-
	treasury	External		-
	adapterManager	External		-
	adapterInfo	External		-

IHedgepieInvestorEth	Interface			
	ybnft	External		-
	treasury	External		-
	adapterManager	External		-
	adapterInfo	External		-
IHedgepieInvestorMatic	Interface			
	ybnft	External		-
	treasury	External		-
	adapterManager	External		-
	adapterInfo	External		-
IPancakePair	Interface			
	token0	External		-
	token1	External		-
	totalSupply	External		-
	fee	External		-
	getReserves	External		-
IPancakeRouter	Interface			
	getAmountsIn	External		-
	swapExactTokensForTokens	External	✓	-
	swapExactETHForTokensSupportingFeeOnTransferTokens	External	Payable	-
	swapExactTokensForETHSupportingFeeOnTransferTokens	External	✓	-
	addLiquidity	External	✓	-
	addLiquidityETH	External	Payable	-
	removeLiquidity	External	✓	-

	removeLiquidityETH	External	✓	-
	getAmountsOut	External		-
IRNG	Interface			
	getRandomNumber	External	✓	-
	randomResults	External	✓	-
IVaultStrategy	Interface			
	wantLockedTotal	External		-
	totalSupply	External		-
	sharesTotal	External		-
	earn	External	✓	-
	deposit	External	✓	-
	withdraw	External	✓	-
	inCaseTokensGetStuck	External	✓	-
IWrap	Interface			
	deposit	External	✓	-
	withdraw	External	✓	-
	deposit	External	Payable	-
IYBNFT	Interface			
	getCurrentTokenId	External		-
	performanceFee	External		-
	getAdapterInfo	External		-
	exists	External		-
	mint	External	✓	-
IPancakeswap Strategy	Interface			

	deposit	External	✓	-
	withdraw	External	✓	-
IVenusStrategy	Interface			
	deposit	External	✓	-
	requestWithdrawal	External	✓	-
Address	Library			
	isContract	Internal		
	sendValue	Internal	✓	
	functionCall	Internal	✓	
	functionCall	Internal	✓	
	functionCallWithValue	Internal	✓	
	functionCallWithValue	Internal	✓	
	_functionCallWithValue	Private	✓	
Context	Implementation			
		Public	✓	-
	_msgSender	Internal		
	_msgData	Internal		
EnumerableSet	Library			
	_add	Private	✓	
	_remove	Private	✓	
	_contains	Private		
	_length	Private		
	_at	Private		
	addrToUint	Internal		
	add	Internal	✓	

	remove	Internal	✓	
	contains	Internal		
	length	Internal		
	at	Internal		
	add	Internal	✓	
	remove	Internal	✓	
	contains	Internal		
	length	Internal		
	at	Internal		
FullMath	Library			
	fullMul	Private		
	fullDiv	Private		
	mulDiv	Internal		
Babylonian	Library			
	sqrt	Internal		
BitMath	Library			
	mostSignificantBit	Internal		
FixedPoint	Library			
	decode	Internal		
	decode112with18	Internal		
	fraction	Internal		
	sqrt	Internal		
HedgepieLibraryBsc	Library			
	swapOnRouter	Public	✓	-

	swapforBnb	Public	✓	-
	getRewards	Public		-
	getLP	Public	✓	-
	withdrawLP	Public	✓	-
HedgepieLibraryEth	Library			
	swapOnRouter	Public	✓	-
	swapforEth	Public	✓	-
	getRewards	Public		-
	getLP	Public	✓	-
	withdrawLP	Public	✓	-
HedgepieLibraryMatic	Library			
	swapOnRouter	Public	✓	-
	swapforMatic	Public	✓	-
	getRewards	Public		-
	getLP	Public	✓	-
	withdrawLP	Public	✓	-
Ownable	Implementation	Context		
		Public	✓	-
	owner	Public		-
	renounceOwnership	Public	✓	onlyOwner
	transferOwnership	Public	✓	onlyOwner
	_transferOwnership	Internal	✓	
SafeBEP20	Library			
	safeTransfer	Internal	✓	
	safeTransferFrom	Internal	✓	

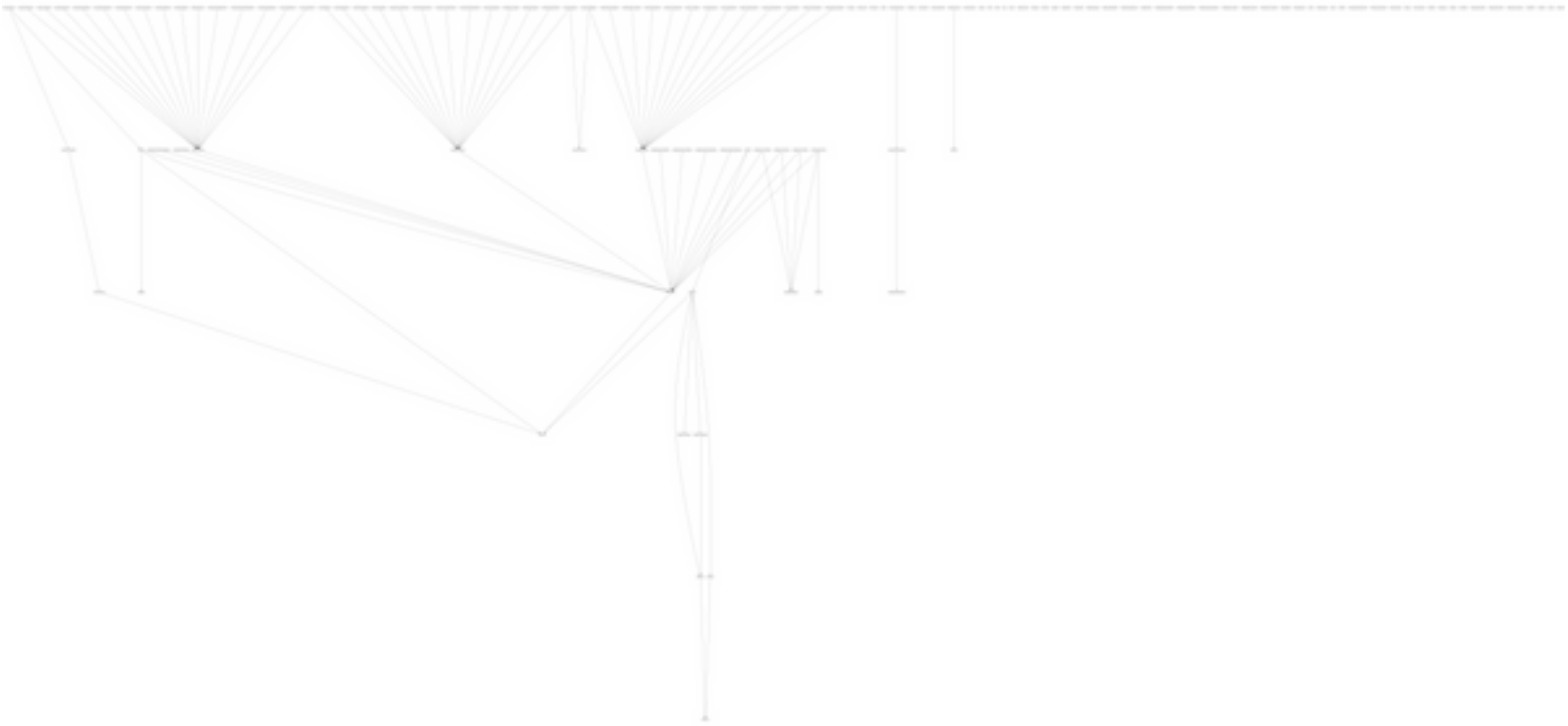
	safeApprove	Internal	✓	
	safeIncreaseAllowance	Internal	✓	
	safeDecreaseAllowance	Internal	✓	
	_callOptionalReturn	Private	✓	
SafeMath	Library			
	add	Internal		
	sub	Internal		
	sub	Internal		
	mul	Internal		
	div	Internal		
	div	Internal		
	mod	Internal		
	mod	Internal		
	sqrrt	Internal		
Strings	Library			
	toString	Internal		
	toHexString	Internal		
	toHexString	Internal		
Multicall	Implementation			
	aggregate	Public	✓	-
	getEthBalance	Public		-
	getBlockHash	Public		-
	getLastBlockHash	Public		-
	getCurrentBlockTimestamp	Public		-
	getCurrentBlockDifficulty	Public		-
	getCurrentBlockGasLimit	Public		-

	getCurrentBlockCoinbase	Public		-
AccessControl	Implementation	Context		
	hasRole	Public		-
	getRoleMemberCount	Public		-
	getRoleMember	Public		-
	getRoleAdmin	Public		-
	grantRole	Public	✓	-
	revokeRole	Public	✓	-
	renounceRole	Public	✓	-
	_setupRole	Internal	✓	
	_setRoleAdmin	Internal	✓	
	_grantRole	Private	✓	
	_revokeRole	Private	✓	
AdminAccess Roles	Implementation	AccessControl		
		Public	✓	-
	isAdmin	Public		-
	isMintUser	Public		-
	addMintUser	Public	✓	onlyAdmin
	addAdmin	Public	✓	onlyAdmin
	removeMintUser	Public	✓	onlyAdmin
	renounceAdmin	Public	✓	-
BEP165	Implementation	IBEP165		
	supportsInterface	Public		-
BEP20	Implementation	Context, IBEP20, Ownable		

		Public	✓	-
	getOwner	External		-
	name	Public		-
	decimals	Public		-
	symbol	Public		-
	totalSupply	Public		-
	balanceOf	Public		-
	transfer	Public	✓	-
	allowance	Public		-
	approve	Public	✓	-
	transferFrom	Public	✓	-
	increaseAllowance	Public	✓	-
	decreaseAllowance	Public	✓	-
	_transfer	Internal	✓	
	_mint	Internal	✓	
	_burn	Internal	✓	
	_approve	Internal	✓	
	_burnFrom	Internal	✓	
	_beforeTokenTransfer	Internal	✓	
BEP721	Implementation	Context, BEP165, IBEP721, IBEP721Receiver, IBEP721Metadata		
		Public	✓	-
	supportsInterface	Public		-
	balanceOf	Public		-
	ownerOf	Public		-
	name	Public		-

	symbol	Public		-
	tokenURI	Public		-
	_baseURI	Internal		
	approve	Public	✓	-
	getApproved	Public		-
	setApprovalForAll	Public	✓	-
	isApprovedForAll	Public		-
	transferFrom	Public	✓	-
	safeTransferFrom	Public	✓	-
	safeTransferFrom	Public	✓	-
	onBEP721Received	Public	✓	-
	_safeTransfer	Internal	✓	
	_exists	Internal		
	_isApprovedOrOwner	Internal		
	_safeMint	Internal	✓	
	_safeMint	Internal	✓	
	_mint	Internal	✓	
	_burn	Internal	✓	
	_transfer	Internal	✓	
	_approve	Internal	✓	
	_setApprovalForAll	Internal	✓	
	_checkOnBEP721Received	Private	✓	
	_beforeTokenTransfer	Internal	✓	
	_afterTokenTransfer	Internal	✓	

Inheritance Graph



Original grapth

https://github.com/cyberscope-io/audits/blob/main/hpie

Flow Graph

Original graph

https://github.com/cyberscope-io/audits/blob/main/hpie

Summary

Hedgepie ecosystem contracts implements utility, financial and token mechanism. This audit investigates security issues, business logic concerns and potential improvements.

Disclaimer

The information provided in this report does not constitute investment, financial or trading advice and you should not treat any of the document's content as such. This report may not be transmitted, disclosed, referred to or relied upon by any person for any purposes nor may copies be delivered to any other person other than the Company without Cyberscope’s prior written consent. This report is not nor should be considered an “endorsement” or “disapproval” of any particular project or team. This report is not nor should be regarded as an indication of the economics or value of any “product” or “asset” created by any team or project that contracts Cyberscope to perform a security assessment. This document does not provide any warranty or guarantee regarding the absolute bug-free nature of the technology analyzed, nor do they provide any indication of the technologies proprietors' business, business model or legal compliance. This report should not be used in any way to make decisions around investment or involvement with any particular project. This report represents an extensive assessment process intending to help our customers increase the quality of their code while reducing the high level of risk presented by cryptographic tokens and blockchain technology.

Blockchain technology and cryptographic assets present a high level of ongoing risk Cyberscope’s position is that each company and individual are responsible for their own due diligence and continuous security Cyberscope’s goal is to help reduce the attack vectors and the high level of variance associated with utilizing new and consistently changing technologies and in no way claims any guarantee of security or functionality of the technology we agree to analyze. The assessment services provided by Cyberscope are subject to dependencies and are under continuing development. You agree that your access and/or use including but not limited to any services reports and materials will be at your sole risk on an as-is where-is and as-available basis Cryptographic tokens are emergent technologies and carry with them high levels of technical risk and uncertainty. The assessment reports could include false positives false negatives and other unpredictable results. The services may access and depend upon multiple layers of third parties.

About Cyberscope

Cyberscope is a blockchain cybersecurity company that was founded with the vision to make web3.0 a safer place for investors and developers. Since its launch, it has worked with thousands of projects and is estimated to have secured tens of millions of investors' funds.

Cyberscope is one of the leading smart contract audit firms in the crypto space and has built a high-profile network of clients and partners.

The Cyberscope team

https://www.cyberscope.io