Cyberscope

Audit Report

Hedgepie Finance

January 2023

Github	https://github.com/innovation-upstream/hedgepie-dev
Commit	c3cf61e13d509dc1874135346ac186ba62ff9972
Audited by	© cyberscope

Table of Contents

Review

Repository	https://github.com/innovation-upstream/hedgepie-dev/tree/audit
Commit	c3cf61e13d509dc1874135346ac186ba62ff9972

Audit Updates

Initial Audit	05 Jan 2023 https://github.com/cyberscope-io/audits/blob/main/hpie/v1/audit.pdf
Corrected Phase 2	03 Feb 2023

Source Files

Filename	SHA256
adapters/BaseAdapterBsc.sol	891f0210e4ec6e8809f51ade942d561f4d2583167e54a4fe2ebe0599b0077bcc
adapters/BaseAdapterEth.sol	4ea736ec3c48069b3e53a825d330169f5d91330365fc1d0f9ad0219b559261b6
adapters/BaseAdapterMatic.sol	e447d2664c0a2142132fe87562c5b3f838f651e8602dd7945ad8ddedb9ab4244
adapters/bnb/alpaca/alpaca-ausd-adapter.sol	0f2fc9caf942da5ff52ae174330e156a625282dd4275301d3aec608fa0b96e57
adapters/bnb/alpaca/alpaca-lend-adapter.sol	66d61006c58672db657bbd50d28cf15b56578401fd4ba05c14f3af4bd4fcefde
adapters/bnb/alpaca/alpaca-stake-adapter.sol	e32b986c88ecfc481094263165776b2815b10b3c4500dbd7dd5b3b7a4e52a5d5
adapters/bnb/apeswap/apeswap-banana-adapter.sol	1143fcffc2d5a55ca9f3b25bd883c1b552e962a070b363690889054d11616f14
adapters/bnb/apeswap/apeswap-farm-lp-adapter.sol	ca0c133299f61ee3f9bf22d547a395d0e5a8f00f9182c28df03298a96fc3ff7e
adapters/bnb/apeswap/apeswap-jungle-adapter.sol	813f886d4533ab062f717a08d7b072f0c2caa287ebe3fdf9433b75f9467e77c0
adapters/bnb/apeswap/apeswap-vault-adapter.sol	51e4eda8c398cb38ecc1573d36676847793e3d3b3345a672fe3d1e110c8bc984
adapters/bnb/autofarm/auto-vault-adapter.sol	039345e535161e44da2885829ac96f1a8595125cad584802dd6a0bda740f9cda
adapters/bnb/beefy/beefy-vault-adapter.sol	8e3591cc03ebb3c5aa9688c4ec2ff7cf55f6060edc310f15139d4202969fc015

adapters/bnb/belt.fi/belt-vault-adapter.sol	e202c1eb7b21e581f2f79c1a5b8e52f5cfa07a202ff93d35a8d86fbd1d4d5295
adapters/bnb/biswap/biswap-farm-lp-adapter.sol	070040f2cfa00a22835b2447f3e34e840743cb26342e35843e0d242e67bf3e8c
adapters/bnb/pancakeswap/pancake-farm-adapter.sol	bbc936f14509a5df1a0b2a5764dc5caaea8b98b8069cc53007c56153c30829ae
adapters/bnb/pancakeswap/pancake-stake-adapter.sol	f60ee9b3c5758f9d43f06a2839c374b3db945df3feddefa46e22ea76d50fe613
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	38bebe0e89a8a1faefd85c8ffa6a077e60ada0a7fdf178de1457b6eb33b4a201
adapters/bnb/venus/venus-lev-adapter.sol	f69a7260bfdfdb45b2b5cbf06b24a690d47cad1fd813606d5179459951214969
adapters/eth/aave/AaveLendAdapterEth.sol	d5356d6320c857fcec5ae1c4d8ec26c9d1228efddf8fe7de3b0fd6e5b75c7a88
adapters/eth/balancer/balancer-vault-adapter.sol	007c89b84e6057c3e84df72a02138582c15b38ca011b81d8c6fca8fafb4e6797

adapters/eth/compound/CompoundLendAdapterEth.sol	ec7d404ed904af220342ae025758d5f211a2de3200a0dc6d194ba6babcb20c0b
adapters/eth/curve/curve-gauge-adapter.sol	aad86f55ca343d77442e9c6e5d1f5e18cd65cca335e85b8332f757d0387e4f2b
adapters/eth/pickle/PickleCurveGaugeAdapter.sol	5886e4ec800f7f28a525fc4ab1712e9d4d600a61e31523b426f79abc92f03485
adapters/eth/pickle/PickleSingleGaugeAdapter.sol	8aceb567484b8d05a79be4774ef4e7bd72715422c8540046045d4d399e489c39
adapters/eth/pickle/PickleSushiGaugeAdapter.sol	ef0aee567878728ba3f8fe0541e29b8884c8d8f5037f94bd837a38d598509c9d
adapters/eth/pickle/PickleSushiMasterAdapter.sol	667e275ee27baa6c07458341195170895bae38d8f6937199f54a8fa2bf0c07bd
adapters/eth/sushiswap/SushiFarmAdapterEth.sol	680c5ecd16ab0de01574c2bbd31d9899263d8eddbd118e6e5a98ce059ff466dc
adapters/eth/sushiswap/SushiFarmV2AdapterEth.sol	7f251367473d61a3cc7199dcfbce2ddfb101c51552090605d5399288d64f5c61
adapters/eth/uniswap/uniswap-lp-v3-adapter.sol	dff18b4029564e5757af1f56cc3c59693ae8efb120ac631c88f2058191eb7b58
adapters/eth/yearn/yearn-curve-adaper.sol	2a4705251c2d1b02f2c60d579565c6fadea8ab273fb3142a1939e60202e6a3c6
adapters/eth/yearn/yearn-single-adapter.sol	f47fe232409a371bf00f0d2d5e470ff9d563494ba9485d36068c759719abe4e6
adapters/polygon/aave/aave-market-v2-adapter.sol	51d69dd80d4924dfc1e70df3f38cfce26d0587c3723a66e19f08dc85d08152db
adapters/polygon/aave/aave-market-v3-adapter.sol	38f4ba728a962944bb2cebd614b6e5a50729918b2bfa3349ce0dce09d7ad3674

adapters/polygon/apeswap/apeswap-farm-adapter.sol	a142421450dd495a427a8c1527b9620a76d80a0b009968de331ab923058252d7
adapters/polygon/beefy/beefy-balancer-adapter.sol	892167d52b08295b0f7a73a5f35541dd35a367587ca25811dfa8d4e428420fb3
adapters/polygon/beefy/beefy-stargate-adapter.sol	cdf598f0e5b75d6e0628542cf22caba89addb5672d8ac06bb6f4404dd758c775
adapters/polygon/beefy/beefy-vault-adapter.sol	dcdc2ebf1d95d2ab631ce39b43fd076f8158ed90438bfd8d466194e956f8efe6
adapters/polygon/curve/curve-lp-adapter.sol	b915a206448d918043de0b885036aef743d966163a9b5639622d99bf01382cec
adapters/polygon/quickswap/quick-lp-dual-adapter.sol	5847bf03e66cbddf8e2606e1f68309491fe665a50da43a34fb574f804788785f
adapters/polygon/quickswap/quick-lp-farm-adapter.sol	756c89712b635320739390576522e048832f3ae6e10cb0b244cdfc3ea3d74526
adapters/polygon/quickswap/quick-stake-adapter.sol	945f465955d454cc89b88fe5d97ef7ed75a52dadce0e8604ca6d86b070a70701
adapters/polygon/stargate/stargate-farm-adapter.sol	cd732e47624a19126bbe2c04d6d8374f62b8d05b7e5210ad493df035403609e7
adapters/polygon/sushi/sushi-farm-adapter.sol	c0e789ea87c214d2e362551f75b4ae4384ffbad6f3b97e7f0b4118c4a2bc8b66
adapters/polygon/uniswap/uniswap-lp-adapter.sol	93f4b09d9dfac445e93ba8d13ca1a9b9c98a714e9227466110c737aebf4ea497
HedgepieAdapterInfoBsc.sol	3091579f2e90e1d4ac9bc342e5e3a800f769087058c7f05b4b2ca1c72a0ee74a
HedgepieAdapterInfoEth.sol	fde19d46eac341f2ee5ac895edf8fcc42a46a702c5ae7a244335da9dc247e234

HedgepieAdapterInfoMatic.sol	3b7346e4db8dd6f7c7e38a7f3ed448f31a88f7ff6e6ba103fdf26daf1b76c62e
HedgepieAdapterManagerBsc.sol	c17924dfa1414aed43466cda9bdad3dc481eda65145cc627b96b56db8226fa0f
HedgepieAdapterManagerEth.sol	82d023815decab5a5f5a8a872050006c1428b3bce026c01dab7c681ba79b1cbe
HedgepieAdapterManagerMatic.sol	87b33144d8e0029d9d99d4f7392b7d7e73a1bdb8c53d4ad53fb87def65800eb7
HedgepieInvestorBsc.sol	ae1af039ad1526fd0cad2af9e30dc376f53f59ead6104400ed06a72667e82804
HedgepieInvestorEth.sol	922d43d0e9975d417d5a3553317ef190c53ce4e6f93d2d18db4d4c69c8e45239
HedgepieInvestorMatic.sol	9b316a1e7a1336c57ec0ccd8a0942bc78b2f814ecdf70f0b1b65645c7de183d2
HedgepieMasterChef.sol	1df29f15faa13439522f35a202d706d4e4e01a8114c564825827b45538eabb58
HedgepieToken.sol	4fa719e08ce69ee72ea8bb6fcf4ca4306a1b5a76eb8247468d9e1f0959cf75a3
HedgepieYBNFT.sol	d67d97012c025e0fe16e9f863cb165e1b90bdb7f656d244c61fb6c8eec2c2269
interfaces/IAdapter.sol	aee42ee6a0aa17d24402a535c9be09c6c2c5787385025504d2a8ba5919e93c6c
interfaces/IAdapterBsc.sol	1a3c9dcf8ce789cf24f79b1c29c054110c8be14fb47559af456c988e271586f0
interfaces/IAdapterEth.sol	e6cc12467ea3e9c3ee571d9dddd430004c757b4ee3b6ac9e0febbb1b0eb3847c

interfaces/IAdapterManager.sol	bd30412106454f4d62f844181cb6c8e5d6456bc7baf4f68d1c8e80a1b23b446f
interfaces/IAdapterManagerEth.sol	d517738b07503ebc8710f16f85ed1f9d8ecc1a78d97e77435f0b1ae444c07f10
interfaces/IAdapterManagerMatic.sol	24acbb7b62a484fec47967c8657aa8123777d764d90cbcd5eb195f68004502d7
interfaces/IAdapterMatic.sol	50568491e6b2c9a60b674ec32db1999ab4f46d591938536bd09f2b697c99aa33
interfaces/IBEP165.sol	e5c5014bfa05d512027982a43066cf8e01b0364d117e162f07b8f0d0c7758985
interfaces/IBEP20.sol	d7adbc5408c5d75e05bdd2d8618b2e013dc8c2539c26d2961f650de03ac3c3af
interfaces/IBEP721.sol	3e61cb926a68428c4aa547a10b912d0e08fb5fc7f33e77875a4ed48e8786c4c0
interfaces/IBEP721Metadata.sol	28bfdf127ede6fcc79c6d428e83b5f729f46eed2e36be466a3d850514de9cd5b
interfaces/IBEP721Receiver.sol	65da14007bd986bca6d87a1f597b5f0047f1b31cd6c9fa223413af57e45d78f9
interfaces/IHedgepieAdapterInfoBsc.sol	74d2551f93e3804cbe580739141546b6aeef37c911fe3ac428b07c1909595f73
interfaces/IHedgepieAdapterInfoEth.sol	ae5a8e0ee79af4dfa2e1e933b02517c72f2363a456564b61ebedb07669d3cb60
interfaces/IHedgepieAdapterInfoMatic.sol	638608c956796fb0e4b2c5b91afc513770a697b3158e226a156038e4bac89aa8
interfaces/IHedgepieInvestorBsc.sol	e814f41209bcea164fe3345a84bba7629da0d187c92c652982b1fbc31c7fa664

interfaces/IHedgepieInvestorEth.sol	89ed6d73411d24b4d7eb1346e9fab71557eca978752a46e32f710e10d2862504
interfaces/IHedgepieInvestorMatic.sol	f464c54d10ab0e17e2cd25cededa1a3acad52fa10e282ec2f8a8eacc36e6eeac
interfaces/IPancakePair.sol	3b6fcda50c22a9db85abb57bcf2d87b4e8ba7ca2e11d6366e34cc50499d6d236
interfaces/IPancakeRouter.sol	5b6a0ccde8aab4c23f0ee99a23148f832dc85570228e698e0ba9c6710b8b38be
interfaces/IRNG.sol	4de9b07c095e7edf6f80f2c33522a83620165e4352833fd44db0a3def8abb235
interfaces/IVaultStrategy.sol	a1b933db9c687ca29ed66c0ea62bfd8ae95abf12d18a6b0649e70ba9811a6695
interfaces/IWrap.sol	12228c23e9cc274bb322d165d596ccb97c16be1aaa054d33bc6007e01145143d
interfaces/IYBNFT.sol	675eeaf5b7eb32797b4db90f98e38344c5ccfc557d4a08c8e76411449a1b741e
interfaces/strategies/IPancakeswapStrategy.sol	b4ac909bafc7e637b846edb3ee517cff0e11a1943305b0c4e65fdd56f74a33f3
interfaces/strategies/IVenusStrategy.sol	45a8ae93a5332af9946690de75e7c7e149151815112d006055f34e20334e8968
libraries/Address.sol	8591d74508d0e2526866a32fe460793bf149ae79338f847cdcbb50ffadc35953
libraries/Context.sol	b14f1609a44d1bc53805621e322cb609a510c86b22c9bc9cf1960e6adaf0fc0d
libraries/EnumerableSet.sol	88b261e7bf185d59e6f0a9d087cba9a1648ec53311a00f1f3189eea23a115ae5

libraries/FixedPoint.sol	0b381904ac838b09c8fbec7901aa44ce23939c4a0bad83852c38ecd5f0502d7d
libraries/HedgepieLibraryBsc.sol	e088c121b7443b7266e1a2341415c65cce94d35350f589c649255ca96692882b
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/

Analysis

● Critical ● Medium ● Minor / Informative ● Pass

Severity	Code	Description	Status
●	BLC	Business Logic Concern	Unresolved
●	RSML	Redundant SafeMath Library	Acknowledged
●	AAO	Accumulated Amount Overflow	Acknowledged
●	RDSF	Redundant Data Structure Fields	Acknowledged
●	CR	Code Repetition	Acknowledged
●	MSC	Missing Sanity Check	Acknowledged
●	PSSI	Potential State Synchronization Inconsistency	Acknowledged
●	UPEH	Underneath Protocols Error Handling	Unresolved
●	L02	State Variables could be Declared Constant	Unresolved
●	L04	Conformance to Solidity Naming Conventions	Acknowledged
●	L07	Missing Events Arithmetic	Acknowledged
●	L13	Divide before Multiply Operation	Acknowledged
●	L14	Uninitialized Variables in Local Scope	Acknowledged
●	L19	Stable Compiler Version	Acknowledged

BLC - Business Logic Concern

Criticality	Medium
Location	adapters/bnb/biswap/biswap-farm-lp-adapter.sol#L113 adapters/bnb/pancakeswap/pancake-farm-adapter.sol#L104 adapters/bnb/alpaca/alpaca-stake-adapter.sol#L100
Status	Unresolved

Description

The implementation may not follow the expected behavior. The contract utlizes the same accTokenPerShare variable for different rewards.

```
if (userInfo.amount != 0) {
    userInfo.rewardDebt +=
        (userInfo.amount *
            (mAdapter.accTokenPerShare - userInfo.userShares)) /
        1e12;
    userInfo.rewardDebt1 +=
        (userInfo.amount *
            (mAdapter.accTokenPerShare - userInfo.userShares)) /
        1e12;
}
```

Recommendation

The team is advised to carefully check if the implementation follows the expected business logic.

RSML - Redundant SafeMath Library

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

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.

Team Update

"Not related to current workflows"

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	Acknowledged

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.

RDSF - Redundant Data Structure Fields

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

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.

CR - Code Repetition

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

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 contract could reuse the common code segments in an internal function in order to avoid repeating the same code in multiple places.

MSC - Missing Sanity Check

Criticality	Minor / Informative
Status	Acknowledged

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.

PSSI - Potential State Synchronization Inconsistency

Criticality	Minor / Informative
Status	Acknowledged

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.

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 `CompoundLendAdapterEth` adapter calls the `redeem()`. The `redeem()` documentation returns 0 on success, otherwise an Error code.

We state that the Compound 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.

L02 - State Variables could be Declared Constant

Criticality	Minor / Informative
Location	adapters/BaseAdapterMatic.sol#L20,22,24,26,28,30,32,34,36,40,42 adapters/BaseAdapterEth.sol#L20,22,24,26,28,30,32,34,38,40 adapters/BaseAdapterBsc.sol#L22,24,26,28,30,32,34,36,40,42
Status	Unresolved

Description

State variables can be declared as constant using the constant keyword. This means that the value of the state variable cannot be changed after it has been set. Additionally, the constant variables decrease gas consumption of the corresponding transaction.

```
uint256 public pid
address public stakingToken
address public liquidityToken
address public rewardToken
address public rewardToken1
address public repayToken
address public strategy
address public router
address public swapRouter
address public wmatic
string public name
address public weth
address public wbnb
```

Recommendation

Constant state variables can be useful when the contract wants to ensure that the value of a state variable cannot be changed by any function in the contract. This can be useful for storing values that are important to the contract's behavior, such as the contract's address or the maximum number of times a certain function can be called. The team is advised to add the constant keyword to state variables that never change.

L04 - Conformance to Solidity Naming Conventions

Criticality	Minor / Informative
Location	HedgepieYBNFT.sol#L56,68,81,93,94,95,96,97,155,174,175,199,272 HedgepieToken.sol#L22 HedgepieMasterChef.sol#L39,105,118,150,176,192,202,240,279,312 HedgepieInvestorMatic.sol#L77,109,133,159,182,193 HedgepieInvestorEth.sol#L77,109,133,159,182,193 HedgepieInvestorBsc.sol#L77,109,133,159,182,193 HedgepieAdapterManagerMatic.sol#L60,75,96,107 HedgepieAdapterManagerEth.sol#L60,75,96,107 HedgepieAdapterManagerBsc.sol#L59,85,100,121,132 HedgepieAdapterInfoMatic.sol#L50,51,52,61,62,63,72,73,74,83,84,85,111 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 adapters/BaseAdapterMatic.sol#L66,95,96,97,127 adapters/BaseAdapterEth.sol#L64,93,94,95,125 adapters/BaseAdapterBsc.sol#L68,97,98,99,129
Status	Acknowledged

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 _adapterManager
address _to
uint256 _amount
uint256 public BONUS_MULTIPLIER = 100
uint256 _from
uint256 _to

...
```

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.

L07 - Missing Events Arithmetic

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

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.

Team Update

"Not related to current workflows"

L13 - Divide before Multiply Operation

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

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.

Team Update

"Not related to current workflows"

L14 - Uninitialized Variables in Local Scope

Criticality	Minor / Informative
Location	HedgepieYBNFT.sol#L187,261 HedgepieInvestorMatic.sol#L92,119,143,170 HedgepieInvestorBsc.sol#L92,119,143,170 HedgepieAdapterManagerBsc.sol#L69 adapters/BaseAdapterMatic.sol#L106 adapters/BaseAdapterEth.sol#L104 adapters/BaseAdapterBsc.sol#L108
Status	Acknowledged

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	type/BEP721.sol#L2 type/BEP20.sol#L2 type/BEP165.sol#L2 type/AdminAccessRoles.sol#L2 type/AccessControl.sol#L2 libraries/Strings.sol#L2 libraries/SafeMath.sol#L2 libraries/SafeBEP20.sol#L2 libraries/Ownable.sol#L2 libraries/EnumerableSet.sol#L2 libraries/Context.sol#L2 libraries/Address.sol#L2 interfaces/IYBNFT.sol#L2 interfaces/IWrap.sol#L2 interfaces/IBEP721Receiver.sol#L2 interfaces/IBEP721Metadata.sol#L2 interfaces/IBEP721.sol#L2 interfaces/IBEP20.sol#L2 interfaces/IBEP165.sol#L2 interfaces/IAdapterMatic.sol#L2 interfaces/IAdapterManager.sol#L2 interfaces/IAdapterEth.sol#L2 interfaces/IAdapterBsc.sol#L2 HedgepieYBNFT.sol#L2 HedgepieToken.sol#L2 HedgepieMasterChef.sol#L2 HedgepieInvestorMatic.sol#L2 HedgepieInvestorEth.sol#L2 HedgepieInvestorBsc.sol#L2 HedgepieAdapterManagerMatic.sol#L2 HedgepieAdapterManagerEth.sol#L2 HedgepieAdapterManagerBsc.sol#L2 HedgepieAdapterInfoMatic.sol#L2 HedgepieAdapterInfoEth.sol#L2 HedgepieAdapterInfoBsc.sol#L2 adapters/BaseAdapterMatic.sol#L2 adapters/BaseAdapterEth.sol#L2 adapters/BaseAdapterBsc.sol#L2
Status	Acknowledged

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		-
BaseAdapter Matic	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
	claim	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		-
ApeswapVaultAdapter	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	✓	-
AaveMarketV2AdapterMatic	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	✓	-
AaveMarketV3AdapterMatic	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		-
BeefyStargate Adapter	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		-
QuickLPFarmAdapter	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		-
	getAdapterInfo	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		
	setAdapterManager	External	✓	onlyOwner
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		-
	mAdapter	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		-
	getAdapterInfo	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		-
	getMRewards	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	✓	

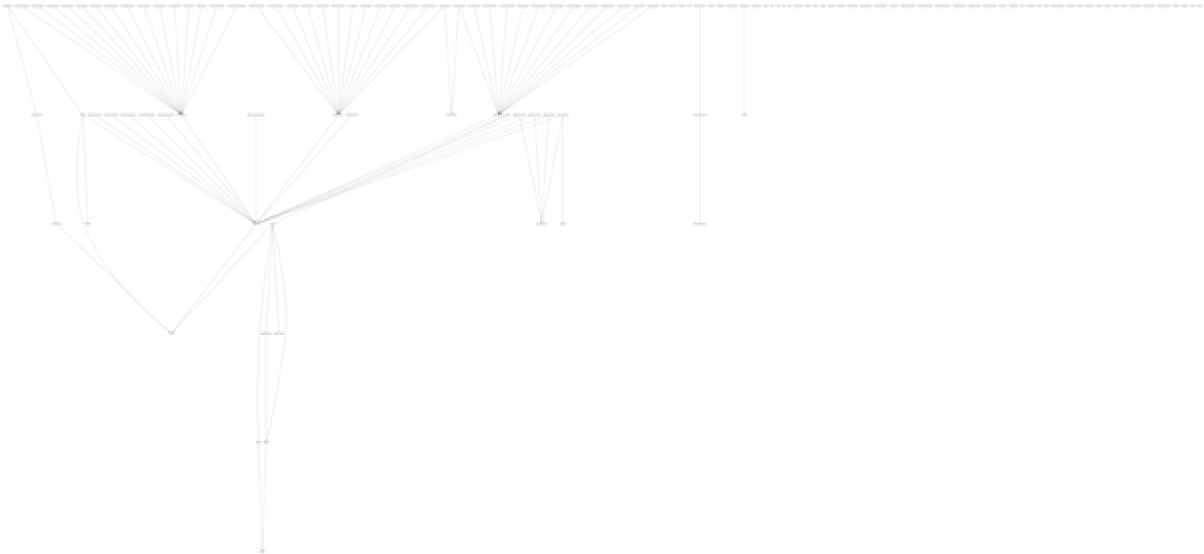
	safeApprove	Internal	✓	
	safeIncreaseAllowance	Internal	✓	
	safeDecreaseAllowance	Internal	✓	
	_callOptionalReturn	Private	✓	
SafeMath	Library			
	add	Internal		
	sub	Internal		
	sub	Internal		
	mul	Internal		
	div	Internal		
	div	Internal		
	mod	Internal		
	mod	Internal		
	sqrrt	Internal		
Strings	Library			
	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	✓	
AdminAccessRoles	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