



# Vesper Finance

Smart Contract Security Audit  
- OneOracle

Prepared by: **Halborn**

Date of Engagement: **May 31st, 2022 - June 7th, 2022**

Visit: **Halborn.com**

DOCUMENT REVISION HISTORY	4
CONTACTS	4
1 EXECUTIVE OVERVIEW	5
1.1 INTRODUCTION	6
1.2 AUDIT SUMMARY	6
1.3 TEST APPROACH & METHODOLOGY	6
RISK METHODOLOGY	7
1.4 SCOPE	9
2 ASSESSMENT SUMMARY & FINDINGS OVERVIEW	11
3 FINDINGS & TECH DETAILS	12
3.1 (HAL-01) SAFEAPPROVE(0) IS NOT CALLED BEFORE SETTING A NEW APPROVAL - <b>CRITICAL</b>	14
Description	14
Proof of Concept	16
Risk Level	16
Recommendation	16
Remediation Plan	17
3.2 (HAL-02) UNISWAPV2LIKEPRICEPROVIDER WILL REVERT WITH OVERFLOWS - <b>CRITICAL</b>	18
Description	18
Code Location	18
Risk Level	19
Recommendation	19
Remediation Plan	21
3.3 (HAL-03) SWAPPER.GETBESTAMOUNT FUNCTIONS MAY RETURN INCORRECT MAX AND MIN AMOUNTS - <b>HIGH</b>	22

Description	22
Proof of Concept	25
Risk Level	25
Recommendation	25
Remediation Plan	26
<b>3.4 (HAL-04) INCOMPATIBILITY WITH TRANSFER-ON-FEE OR DEFLATIONARY TOKENS - HIGH</b>	<b>27</b>
Description	27
Proof of Concept	28
Risk Level	28
Recommendation	28
Remediation Plan	29
<b>3.5 (HAL-05) MAX DEVIATION TOO HIGH IN SWAPPERORACLE - MEDIUM</b>	<b>30</b>
Description	30
Risk Level	31
Recommendation	32
Remediation Plan	32
<b>3.6 (HAL-06) STATE VARIABLES MISSING IMMUTABLE MODIFIER - INFORMATIONAL</b>	<b>33</b>
Description	33
Code Location	33
Risk Level	33
Recommendation	33
Remediation Plan	34
<b>4 AUTOMATED TESTING</b>	<b>35</b>
<b>4.1 STATIC ANALYSIS REPORT</b>	<b>36</b>
Description	36

Slither results	36
4.2 AUTOMATED SECURITY SCAN	58
Description	58
MythX results	58

## DOCUMENT REVISION HISTORY

VERSION	MODIFICATION	DATE	AUTHOR
0.1	Document Creation	05/31/2022	Roberto Reigada
0.2	Document Updates	06/07/2022	Roberto Reigada
0.3	Draft Review	06/07/2022	Gabi Urrutia
0.4	Document Updates	06/17/2022	Roberto Reigada
1.0	Remediation Plan	06/27/2022	Roberto Reigada
1.1	Remediation Plan Review	06/28/2022	Gabi Urrutia

## CONTACTS

CONTACT	COMPANY	EMAIL
Rob Behnke	Halborn	Rob.Behnke@halborn.com
Steven Walbroehl	Halborn	Steven.Walbroehl@halborn.com
Gabi Urrutia	Halborn	Gabi.Urrutia@halborn.com
Roberto Reigada	Halborn	Roberto.Reigada@halborn.com

# EXECUTIVE OVERVIEW

## 1.1 INTRODUCTION

Vesper Finance engaged Halborn to conduct a security audit on their smart contracts beginning on May 31st, 2022 and ending on June 7th, 2022. The security assessment was scoped to the smart contracts provided in the GitHub repository [bloqpriv/one-oracle](#).

## 1.2 AUDIT SUMMARY

The team at Halborn was provided two weeks for the engagement and assigned a full-time security engineer to audit the security of the smart contract. The security engineer is a blockchain and smart-contract security expert with advanced penetration testing, smart-contract hacking, and deep knowledge of multiple blockchain protocols.

The purpose of this audit is to:

- Ensure that smart contract functions operate as intended
- Identify potential security issues with the smart contracts

In summary, Halborn identified some security risks that were mostly addressed by the [Vesper Finance team](#).

## 1.3 TEST APPROACH & METHODOLOGY

Halborn performed a combination of manual and automated security testing to balance efficiency, timeliness, practicality, and accuracy in regard to the scope of this audit. While manual testing is recommended to uncover flaws in logic, process, and implementation; automated testing techniques help enhance coverage of the code and can quickly identify items that do not follow the security best practices. The following phases and associated tools were used during the audit:

- Research into architecture and purpose
- Smart contract manual code review and walkthrough
- Graphing out functionality and contract logic/connectivity/functions ([solgraph](#))
- Manual assessment of use and safety for the critical Solidity variables and functions in scope to identify any arithmetic related vulnerability classes
- Manual testing by custom scripts
- Scanning of solidity files for vulnerabilities, security hot-spots or bugs. ([MythX](#))
- Static Analysis of security for scoped contract, and imported functions. ([Slither](#))
- Testnet deployment ([Brownie](#), [Remix IDE](#))

#### RISK METHODOLOGY:

Vulnerabilities or issues observed by Halborn are ranked based on the risk assessment methodology by measuring the **LIKELIHOOD** of a security incident and the **IMPACT** should an incident occur. This framework works for communicating the characteristics and impacts of technology vulnerabilities. The quantitative model ensures repeatable and accurate measurement while enabling users to see the underlying vulnerability characteristics that were used to generate the Risk scores. For every vulnerability, a risk level will be calculated on a scale of **5 to 1** with **5** being the highest likelihood or impact.

#### RISK SCALE - LIKELIHOOD

- 5 - Almost certain an incident will occur.
- 4 - High probability of an incident occurring.
- 3 - Potential of a security incident in the long term.
- 2 - Low probability of an incident occurring.
- 1 - Very unlikely issue will cause an incident.

#### RISK SCALE - IMPACT

- 5 - May cause devastating and unrecoverable impact or loss.
- 4 - May cause a significant level of impact or loss.

- 3 - May cause a partial impact or loss to many.
- 2 - May cause temporary impact or loss.
- 1 - May cause minimal or un-noticeable impact.

The risk level is then calculated using a sum of these two values, creating a value of **10** to **1** with **10** being the highest level of security risk.

CRITICAL	HIGH	MEDIUM	LOW	INFORMATIONAL
----------	------	--------	-----	---------------

- 10** - CRITICAL
- 9** - **8** - HIGH
- 7** - **6** - MEDIUM
- 5** - **4** - LOW
- 3** - **1** - VERY LOW AND INFORMATIONAL

## 1.4 SCOPE

### IN-SCOPE:

The security assessment was scoped to the following smart contracts:

#### Initial commit ID:

- `84c39f610c30283ea23358f27304ef7783921c44` which included the following contracts:
  - `Governable.sol`
  - `ChainlinkArbitrumPriceProvider.sol`
  - `ChainlinkAvalanchePriceProvider.sol`
  - `ChainlinkMainnetPriceProvider.sol`
  - `ChainlinkPolygonPriceProvider.sol`
  - `ChainlinkPriceProvider.sol`
  - `PriceProvidersAggregator.sol`
  - `UniswapV2LikePriceProvider.sol`
  - `UniswapV3PriceProvider.sol`
  - `Oracle.sol`
  - `SwapperOracle.sol`
  - `Swapper.sol`
  - `UniswapV2LikeExchange.sol`
  - `CTokenOracle.sol`
  - `CurveLpTokenOracle.sol`
  - `SynthDefaultOracle.sol`
  - `UniswapV3CrossPoolOracle.sol`

#### Final commit ID:

- `db210d41025c45c273de3283c3950a54e8ae4298` which included the following contracts:
  - `Governable.sol`
  - `ChainlinkArbitrumPriceProvider.sol`
  - `ChainlinkAvalanchePriceProvider.sol`
  - `ChainlinkMainnetPriceProvider.sol`
  - `ChainlinkPolygonPriceProvider.sol`
  - `ChainlinkPriceProvider.sol`
  - `FluxPriceProvider.sol`
  - `PriceProvider.sol`

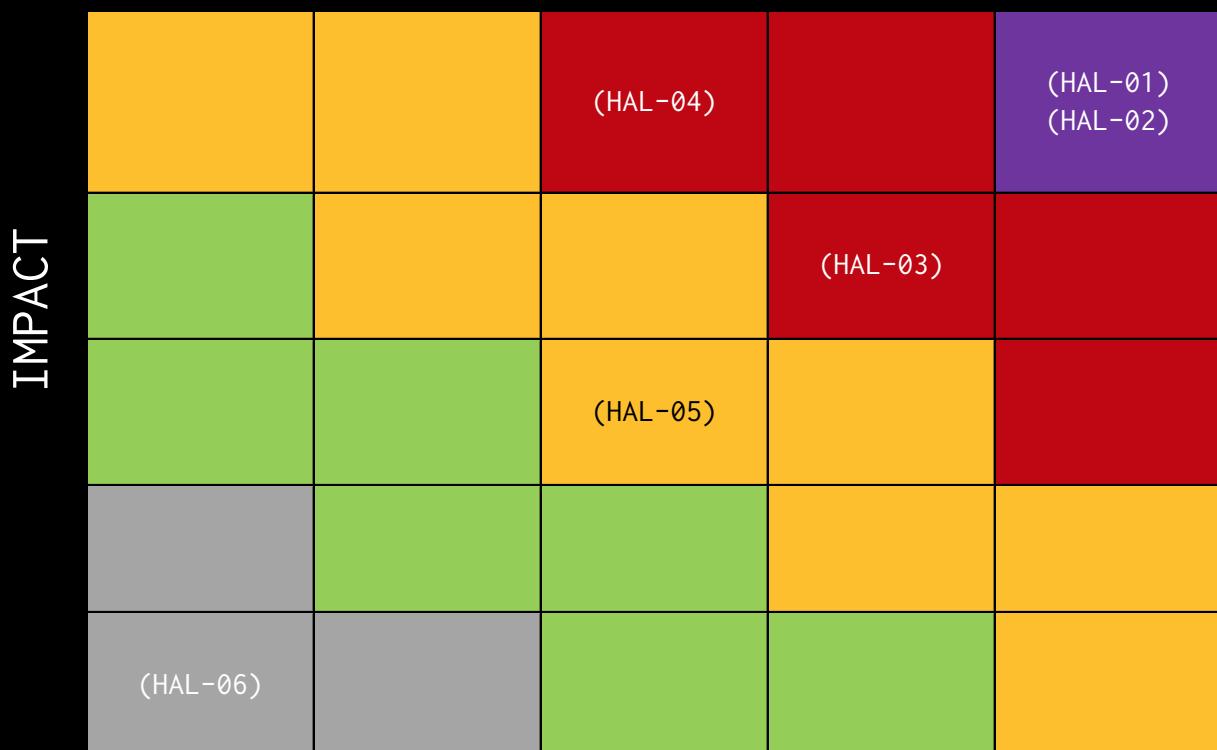
# EXECUTIVE OVERVIEW

- PriceProvidersAggregator.sol
- StableCoinProvider.sol
- UmbrellaPassportPriceProvider.sol
- UmbrellaPriceProvider.sol
- UniswapV2LikePriceProvider.sol
- UniswapV3PriceProvider.sol
- UsingMaxDeviation.sol
- UsingProvidersAggregator.sol
- UsingStableCoinProvider.sol
- UsingStalePeriod.sol
- UniswapV2LikeLpTokenOracle.sol
- SynthOracle
- Swapper.sol
- UniswapV2LikeExchange.sol
- UniswapV3Exchange.sol
- UniswapV3CrossPoolOracle.sol

## 2. ASSESSMENT SUMMARY & FINDINGS OVERVIEW

CRITICAL	HIGH	MEDIUM	LOW	INFORMATIONAL
2	2	1	0	1

### LIKELIHOOD

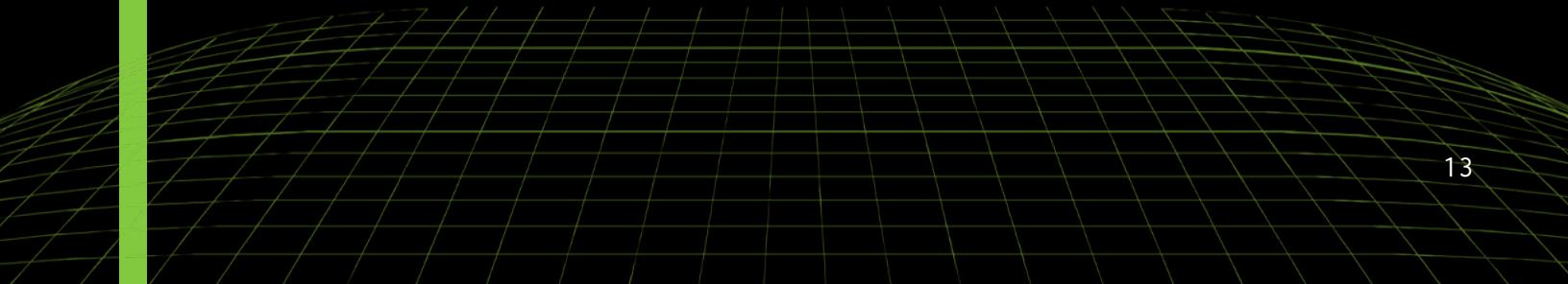


# EXECUTIVE OVERVIEW

SECURITY ANALYSIS	RISK LEVEL	REMEDIATION DATE
HAL01 - SAFEAPPROVE(0) IS NOT CALLED BEFORE SETTING A NEW APPROVAL	Critical	SOLVED - 06/21/2022
HAL02 - UNISWAPV2LIKEPRICEPROVIDER WILL REVERT WITH OVERFLOWS	Critical	SOLVED - 06/21/2022
HAL03 - SWAPPER.GETBESTAMOUNT FUNCTIONS MAY RETURN INCORRECT MAX AND MIN AMOUNTS	High	SOLVED - 06/21/2022
HAL04 - INCOMPATIBILITY WITH TRANSFER-ON-FEE OR DEFLATIONARY TOKENS	High	RISK ACCEPTED
HAL05 - MAX DEVIATION TOO HIGH IN SWAPPERORACLE	Medium	SOLVED - 06/21/2022
HAL06 - STATE VARIABLES MISSING IMMUTABLE MODIFIER	Informational	SOLVED - 06/21/2022



# FINDINGS & TECH DETAILS



### 3.1 (HAL-01) SAFEAPPROVE(0) IS NOT CALLED BEFORE SETTING A NEW APPROVAL - CRITICAL

Description:

In the `UniswapV2LikeExchange` contract, the functions `swapExactInput()` and `swapExactOutput()` do a `SafeApprove` call before calling `swapExactTokensForTokens()` in the Router contract:

```
Listing 1: UniswapV2LikeExchange.sol (Lines 114,128)

108 function swapExactInput(
109     address[] calldata path_,
110     uint256 amountIn_,
111     uint256 amountOutMin_,
112     address outReceiver_
113 ) external returns (uint256 _amountOut) {
114     IERC20(path_[0]).safeApprove(address(router), amountIn_);
115     _amountOut = router.swapExactTokensForTokens(amountIn_,
116         amountOutMin_, path_, outReceiver_, block.timestamp)[
117             path_.length - 1
118 ];
119
120 /// @inheritdoc IExchange
121 function swapExactOutput(
122     address[] calldata path_,
123     uint256 amountOut_,
124     uint256 amountInMax_,
125     address inSender_,
126     address outRecipient_
127 ) external returns (uint256 _amountIn) {
128     IERC20(path_[0]).safeApprove(address(router), amountInMax_);
129     _amountIn = router.swapTokensForExactTokens(amountOut_,
130         amountInMax_, path_, outRecipient_, block.timestamp)[0];
131     // If swap end up costly less than _amountInMax then return
132     remaining
133     uint256 _remainingAmountIn = IERC20(path_[0]).balanceOf(
134         address(this));
135     if (_remainingAmountIn > 0) {
```

```

133         IERC20(path_[0]).safeTransfer(inSender_,
134             _remainingAmountIn);
135     }

```

`SafeERC20.safeApprove()` function is used:

**Listing 2: SafeERC20.safeApprove (Lines 53-56)**

```

45 function safeApprove(
46     IERC20 token,
47     address spender,
48     uint256 value
49 ) internal {
50     // safeApprove should only be called when setting an initial
51     // allowance, or when resetting it to zero. To increase and decrease it,
52     // use 'safeIncreaseAllowance' and 'safeDecreaseAllowance'
53     require(
54         (value == 0) || (token.allowance(address(this), spender)
55         == 0),
56         "SafeERC20: approve from non-zero to non-zero allowance"
57     );
58     _callOptionalReturn(token, abi.encodeWithSelector(token.
59         approve.selector, spender, value));
60 }

```

As per the `SafeERC20` contract's comment: `safeApprove` should only be called when setting an initial allowance, or when resetting it to zero. To increase and decrease it, use `safeIncreaseAllowance` and `safeDecreaseAllowance`.

In the line highlighted, we can see that `SafeERC20.safeApprove()` checks that the allowance is zero before setting a new approval.

This means that every second call by a user to `Swapper.swapExactOutput()` with the same `tokenIn_` token will fail with the error: `SafeERC20: approve from non-zero to non-zero allowance` as the allowance after the first swap will be higher than zero.

## Proof of Concept:

In the image below, we can see how the second swap call for DAI as `tokenIn_` fails with the error: `SafeERC20: approve from non-zero to non-zero allowance`:

```

contract_DAI.balanceOf(user1) -> 40000000000000000000000000000000
contract_DAI.balanceOf(contract_Swapper) -> 0
contract_WETH.balanceOf(user1) -> 0
contract_WETH.balanceOf(contract_Swapper) -> 0
Calling -> contract_DAI.approve(contract_Swapper.address, 4000_0000000000000000000000000000, {'from': user1})
Transaction sent: 0xd614e5e3890f53fb61fe3c2ce69f32925180a258745db6e122cd42e7c56a9fc91
  Gas price: 0.0 gwei  Gas limit: 600000000  Nonce: 0
  DAI.approve confirmed  Block: 14898684  Gas used: 44050 (0.01%)
Calling -> contract_Swapper.swapExactOutput(contract_DAI.address, contract_WETH.address, 1_00000000000000000000000000000000, user1.address, {'from': user1})
Transaction sent: 0xcba91c9334f4fb0441f1dce5ebc1487a3857497d1df30dc8dd012e7d46ac600
  Gas price: 0.0 gwei  Gas limit: 600000000  Nonce: 1
  Swapper.swapExactOutput confirmed  Block: 14898685  Gas used: 278067 (0.05%)
contract_DAI.balanceOf(user1) -> 2248157218382782526549
contract_DAI.balanceOf(contract_Swapper) -> 0
contract_WETH.balanceOf(user1) -> 10000000000000000000000000000000
contract_WETH.balanceOf(contract_Swapper) -> 0
Calling -> contract_Swapper.swapExactOutput(contract_DAI.address, contract_WETH.address, 1_00000000000000000000000000000000, user1.address, {'from': user1})
Transaction sent: 0x4282ae95b69e6b49d49fa94d6340a123269915e0300771954f0e1773f5a4977b
  Gas price: 0.0 gwei  Gas limit: 600000000  Nonce: 2
  Swapper.swapExactOutput confirmed (SafeERC20: approve from non-zero to non-zero allowance)  Block: 14898686  Gas used: 160976 (0.03%)
contract_DAI.balanceOf(user1) -> 2248157218382782526549
contract_DAI.balanceOf(contract_Swapper) -> 0
contract_WETH.balanceOf(user1) -> 10000000000000000000000000000000
contract_WETH.balanceOf(contract_Swapper) -> 0

```

## Risk Level:

**Likelihood - 5**

**Impact - 5**

## Recommendation:

It is recommended to perform 2 different `SafeApprove()` calls in the swap functions. One resetting the allowance to 0 by calling `SafeApprove(0)` and another one with the amount needed to be approved:

**Listing 3: UniswapV2LikeExchange.sol (Lines 114,115,129,130)**

```

108 function swapExactInput(
109     address[] calldata path_,
110     uint256 amountIn_,
111     uint256 amountOutMin_,
112     address outReceiver_
113 ) external returns (uint256 _amountOut) {
114     IERC20(path_[0]).safeApprove(address(router), 0);
115     IERC20(path_[0]).safeApprove(address(router), amountIn_);

```

```
116     _amountOut = router.swapExactTokensForTokens(amountIn_,
117         amountOutMin_, path_, outReceiver_, block.timestamp)[
118             path_.length - 1
119     ];
120
121 /// @inheritdoc IExchange
122 function swapExactOutput(
123     address[] calldata path_,
124     uint256 amountOut_,
125     uint256 amountInMax_,
126     address inSender_,
127     address outRecipient_
128 ) external returns (uint256 _amountIn) {
129     IERC20(path_[0]).safeApprove(address(router), 0);
130     IERC20(path_[0]).safeApprove(address(router), amountInMax_);
131     _amountIn = router.swapTokensForExactTokens(amountOut_,
132         amountInMax_, path_, outRecipient_, block.timestamp)[0];
133     // If swap end up costly less than _amountInMax then return
134     // remaining
135     uint256 _remainingAmountIn = IERC20(path_[0]).balanceOf(
136         address(this));
137     if (_remainingAmountIn > 0) {
138         IERC20(path_[0]).safeTransfer(inSender_,
139             _remainingAmountIn);
140     }
141 }
```

#### Remediation Plan:

**SOLVED:** Vesper Finance team corrected the issue. Now, the `approve()` call will only be called once, as can be seen below:

**Listing 4: UniswapV2LikeExchange.sol (Line 135)**

```
135 if (_tokenIn.allowance(address(this), address(router)) <
136     amountInMax_) {
137     _tokenIn.approve(address(router), type(uint256).max);
138 }
```

## 3.2 (HAL-02)

# UNISWAPV2LIKEPRICEPROVIDER WILL REVERT WITH OVERFLOWS - CRITICAL

### Description:

The `UniswapV2LikePriceProvider` contract makes use of the library `UniswapV2OracleLibrary.sol` which only works with Solidity `< 0.8.0`.

According to [Uniswap official documentation](#):

The `UniswapV2Pair` cumulative price variables are designed to eventually overflow, i.e. `price0CumulativeLast` and `price1CumulativeLast` and `blockTimestampLast` will overflow through 0.

This means that `blockTimestamp` can be lower than `blockTimestampLast` and the new `price0Cumulative` can be less than `price0CumulativeLast` etc.

In that case, the current implementation of `_updateIfNeeded()` will revert due to underflow as the `UniswapV2LikePriceProvider` uses 0.8.9 Solidity version and after the Solidity version 0.8.0 arithmetic operations revert to underflow and overflow by default.

As a result, in some situations `price0Average` and `price1Average` will not be updated for a long time and any other contracts or functions that rely on this price will be affected, as the oracle would be displaying inaccurate information.

### Code Location:

Listing	5:	<code>UniswapV2LikePriceProvider.sol</code>	(Lines 199, 205, 206, 207, 208, 209, 210)
188	189	190	191
<code>/**</code>	<code>* @notice Update an oracle</code>	<code>* @param pair_ The pair to update</code>	<code>* @param twapPeriod_ The TWAP period</code>
<code>192</code>	<code>* @return True if updated was performed</code>		

```
193  */
194 function _updateIfNeeded(IUniswapV2Pair pair_, uint256 twapPeriod_
195 ) private returns (bool) {
196     Oracle storage _oracle = oracles[pair_][twapPeriod_];
197
198     (uint256 price0Cumulative, uint256 price1Cumulative, uint32
199      blockTimestamp) = UniswapV2OracleLibrary
200         .currentCumulativePrices(address(pair_));
201     uint32 timeElapsed = blockTimestamp - _oracle.
202     blockTimestampLast; // overflow is desired
203     // ensure that at least one full period has passed since the
204     // last update
205     if (timeElapsed < twapPeriod_) return false;
206
207     // overflow is desired, casting never truncates
208     // cumulative price is in (uq112x112 price * seconds) units so
209     // we simply wrap it after division by time elapsed
210     _oracle.price0Average = FixedPoint.uq112x112(
211         uint224((price0Cumulative - _oracle.price0CumulativeLast)
212         / timeElapsed)
213     );
214     _oracle.price1Average = FixedPoint.uq112x112(
215         uint224((price1Cumulative - _oracle.price1CumulativeLast)
216         / timeElapsed)
217     );
218     _oracle.price0CumulativeLast = price0Cumulative;
219     _oracle.price1CumulativeLast = price1Cumulative;
220     _oracle.blockTimestampLast = blockTimestamp;
221     return true;
222 }
```

Risk Level:

**Likelihood - 5**

**Impact - 5**

Recommendation:

It is recommended to update the `_updateIfNeeded()` function as shown below:

**Listing 6: UniswapV2LikePriceProvider.sol (Lines 12,13,14,18,19,20,21)**

```
1 /**
2  * @notice Update an oracle
3  * @param pair_ The pair to update
4  * @param twapPeriod_ The TWAP period
5  * @return True if updated was performed
6 */
7 function _updateIfNeeded(IUniswapV2Pair pair_, uint256 twapPeriod_
8 ) private returns (bool) {
9     Oracle storage _oracle = oracles[pair_][twapPeriod_];
10
11    (uint256 price0Cumulative, uint256 price1Cumulative, uint32
12     blockTimestamp) = UniswapV2OracleLibrary
13        .currentCumulativePrices(address(pair_));
14
15    unchecked {
16        uint32 timeElapsed = blockTimestamp - _oracle.
17        blockTimestampLast; // overflow is desired
18    }
19
20    // ensure that at least one full period has passed since the
21    // last update
22    if (timeElapsed < twapPeriod_) return false;
23
24    unchecked {
25        uint256 price0new = price0Cumulative - _oracle.
26        price0CumulativeLast;
27        uint256 price1new = price1Cumulative - _oracle.
28        price1CumulativeLast;
29    }
30
31    // overflow is desired, casting never truncates
32    // cumulative price is in (uq112x112 price * seconds) units so
33    // we simply wrap it after division by time elapsed
34    _oracle.price0Average = FixedPoint.uq112x112(
35        uint224(price0new / timeElapsed)
36    );
37    _oracle.price1Average = FixedPoint.uq112x112(
38        uint224(price1new / timeElapsed)
39    );
40    _oracle.price0CumulativeLast = price0Cumulative;
41    _oracle.price1CumulativeLast = price1Cumulative;
42    _oracle.blockTimestampLast = blockTimestamp;
43    return true;
44 }
```

## FINDINGS & TECH DETAILS

### Remediation Plan:

**SOLVED:** The Vesper Finance team solved the issue by implementing the suggested code that allows for overflows.

### 3.3 (HAL-03) SWAPPER.GETBESTAMOUNT FUNCTIONS MAY RETURN INCORRECT MAX AND MIN AMOUNTS - HIGH

Description:

In the `Swapper` contract, the function `getBestAmountIn()` given an input token, an output token and an output amount, calculates:

- The minimum amount of tokens that should be input by the user to get the amount of output tokens set => `_amountIn`
- The maximum amount of tokens that should be input by the user to get the amount of output tokens set => `_amountInMax`
- The exchange address from where the `_amountIn` data was pulled => `_exchange`
- The swap path that must be used in the `_exchange` to get that `_amountIn` => `_path`

**Listing 7: Swapper.sol (Lines 71,75,89,90)**

```

56 function getBestAmountIn(
57     address tokenIn_ ,
58     address tokenOut_ ,
59     uint256 amountOut_
60 )
61     public
62     view
63     returns (
64         uint256 _amountIn ,
65         uint256 _amountInMax ,
66         IExchange _exchange ,
67         address[] memory _path
68     )
69 {
70     _exchange = IExchange(exchanges.at(0));
71     (_amountIn , _path) = _getBestAmountIn(_exchange , tokenIn_ ,
72     tokenOut_ , amountOut_);
73     uint256 _len = exchanges.length();
74     for (uint256 i = 1; i < _len; ++i) {

```

```

74         IExchange _iExchange = IExchange(exchanges.at(i));
75         (uint256 _iAmountIn, address[] memory _iPath) =
L_     _getBestAmountIn(
76             _iExchange,
77             tokenIn_,
78             tokenOut_,
79             amountOut_
80         );
81         if (_iAmountIn > 0 && _iAmountIn < _amountIn) {
82             _amountIn = _iAmountIn;
83             _exchange = _iExchange;
84             _path = _iPath;
85         }
86     }
87     require(_path.length > 0, "no-path-found");
88
89     uint256 _amountInFromOracle = oracle.quote(tokenOut_, tokenIn_
L_ , amountOut_);
90     _amountInMax = (_amountInFromOracle * (1e18 + maxSlippage)) /
L_ 1e18;
91 }
```

In order to retrieve all that data, this function calls the `getBestAmountIn()` function in the Exchange Router Contract:

**Listing 8: UniswapV2LikeExchange.sol (Lines 51,64)**

```

42 function getBestAmountIn(
43     address tokenIn_,
44     address tokenOut_,
45     uint256 amountOut_
46 ) external view returns (uint256 _amountIn, address[] memory _path
L_ ) {
47     // 1. Check IN-OUT pair
48     address[] memory _pathA = new address[](2);
49     _pathA[0] = tokenIn_;
50     _pathA[1] = tokenOut_;
51     uint256 _amountInA = _getAmountsIn(amountOut_, _pathA);
52
53     if (tokenIn_ == wethLike || tokenOut_ == wethLike) {
54         // Returns if one of the token is WETH-Like
55         require(_amountInA > 0, "invalid-swap");
56         return (_amountInA, _pathA);
```

```

57     }
58
59     // 2. Check IN-WETH-OUT path
60     address[] memory _pathB = new address[](3);
61     _pathB[0] = tokenIn_;
62     _pathB[1] = wethLike;
63     _pathB[2] = tokenOut_;
64     uint256 _amountInB = _getAmountsIn(amountOut_, _pathB);   
65
66     // 3. Get best route between paths A and B
67     require(_amountInA > 0 || _amountInB > 0, "invalid-swap");
68
69     // Returns A if it's valid and better than B or if B isn't
70     // valid
71     if ((_amountInA > 0 && _amountInA < _amountInB) || _amountInB
72     == 0) {
73         return (_amountInA, _pathA);
74     }
75     return (_amountInB, _pathB);
76 }
```

Although occasionally the direct path (`tokenIn_ => tokenOut_`) will not be available, as it occurs here when trying to swap WBTC into ApeCoins:

```

>>> contract_UNISWAPV2_ROUTER.getAmountsIn(5000_0000000000000000, [contract_WBTC, contract_APECOIN])
File "<console>", line 1, in <module>
File "/usr/local/lib/python3.8/dist-packages/brownie/network/contract.py", line 1665, in __call__
    return self.call(*args, block_identifier=block_identifier)
File "/usr/local/lib/python3.8/dist-packages/brownie/network/contract.py", line 1461, in call
    raise VirtualMachineError(e) from None
VirtualMachineError: revert
>>> contract_SUSHISWAP_ROUTER.getAmountsIn(5000_0000000000000000, [contract_WBTC, contract_APECOIN])
File "<console>", line 1, in <module>
File "/usr/local/lib/python3.8/dist-packages/brownie/network/contract.py", line 1665, in __call__
    return self.call(*args, block_identifier=block_identifier)
File "/usr/local/lib/python3.8/dist-packages/brownie/network/contract.py", line 1461, in call
    raise VirtualMachineError(e) from None
VirtualMachineError: revert
```

In this case, the returned values will be taken from the path (`tokenIn_ => WETH => tokenOut_`):

```

>>> contract_UNISWAPV2_ROUTER.getAmountsIn(5000_0000000000000000, [contract_WBTC, contract_WETH, contract_APECOIN])
(129019022, 21202769103392587143, 5000000000000000000000000)
>>> contract_SUSHISWAP_ROUTER.getAmountsIn(5000_0000000000000000, [contract_WBTC, contract_WETH, contract_APECOIN])
(10362012, 50951847326946404459, 5000000000000000000)
```

The `_amountIn` retrieved when this last path is used is way higher, as there is an extra jump in the swap path. On the other hand, `SwapperOracle` is used to retrieve the `_amountInMax`. Chainlink price feeds are used here as primary oracle. This results in an `_amountInMax` way lower than the `_amountIn` value returned.

The same issue is also present in the `Swapper.getBestAmountOut()` function. But in this function, the `_amountOutMin` is higher than the `_amountOut` returned.

#### Proof of Concept:

1. `getBestAmountIn()`: USDC => WETH, 1\_00000000000000000000000000000000
2. `getBestAmountIn()`: USDC => DAI, 1\_00000000000000000000000000000000
3. `getBestAmountIn()`: WBTC => APECOIN, 5000\_000000000000000000000000000000: Issue occurs here.

```
contract Swapper.getBestAmountIn(contract USDC.address, contract WETH.address, 1_00000000000000000000000000000000 ->
(101506846, 105214765, "0x15997109021769040d0e1012204d910c683aE2C4", ("0xA0b66991c621b34c1d19D4a2e9B00c33606eB48", "0xC02aaA39b223FE8D0A0e5C4F27eAD9003C756Cc2"))
contract Swapper.getBestAmountIn(contract USDC.address, contract DAI.address, 1_00000000000000000000000000000000 ->
(1010107, 1009857, "0x0cd6D9A792d91CB1081e29340867663DD755AC", ("0x0b66991c621b36c1d19D4a2e9B00c33606eB48", "0x6B175474E89094c44D96b954EdeA495271d0F"))
contract Swapper.getBestAmountIn(contract WHTC.address, contract APECOIN.address, 5000_000000000000000000000000000000 ->
(129019022, 104629050, "0x774b4F4565ebcB2Fd033641C75D7F9c90159e02", ("0x2260FAC5E5542a773a448fBCf4eD7C193bc2C599", "0xC02aaA39b223FE8D0A0e5C4F27eAD9003C756Cc2", "0x4d24452801AcEd8B2F0aebE155379bb5D594381"))
```

We can see how in the last call `_amountInMax` is lower than `_amountIn`:

- `_amountIn` => 1\_29019022
- `_amountInMax` => 1\_04629050

And also below, we can appreciate the same issue in the `getBestAmountOut()` function:

```
contract Swapper.getBestAmountOut(contract APECOIN.address, contract WBTC.address, 5000_000000000000000000000000000000 ->
(105506525, 105597187, "0x774b4F4565ebcB2Fd033641C75D7F9c90159e02", ("0x4d24452801AcEd8B2F0aebE155379bb5D594381", "0xC02aaA39b223FE8D0A0e5C4F27eAD9003C756Cc2", "0x2260FAC5E5542a773a448fBCf4eD7C193bc2C599"))
```

In this case, we are getting an `_amountOutMin` higher than `_amountOut`.

#### Risk Level:

**Likelihood - 4**

**Impact - 4**

#### Recommendation:

An `_amountIn` higher than `_amountInMax` in the `getBestAmountIn()` return values will cause that the `swapExactOutput()` call fails, as the user would not have approved enough tokens. Something similar occurs with `getBestAmountOut()` and `swapExactInput()` with an `_amountOutMin` higher than `_amountOut`. Users in this case may receive an amount of tokens way lower

than what they expected from the swap.

It is recommended to revisit the logic of these functions, so these cases cannot happen.

Remediation Plan:

**SOLVED:** The Vesper Finance team solved the issue by discarding paths where `_amountIn` is greater than `_amountInMax` or with an `_amountOutMin` greater than `_amountOut`.

### 3.4 (HAL-04) INCOMPATIBILITY WITH TRANSFER-ON-FEE OR DEFLATIONARY TOKENS - HIGH

Description:

In the `Swapper` contract, the functions `swapExactInput()` and `swapExactOutput()` are used to swap tokens in an exchange:

Listing 9: `Swapper.sol` (Lines 148,149,165,166)

```
137 function swapExactInput(
138     address tokenIn_,
139     address tokenOut_,
140     uint256 amountIn_,
141     address receiver_
142 ) external returns (uint256 _amountOut) {
143     (, uint256 _amountOutMin, IExchange _exchange, address[] memory _path) = getBestAmountOut(
144         tokenIn_,
145         tokenOut_,
146         amountIn_
147     );
148     IERC20(tokenIn_).safeTransferFrom(_msgSender(), address(
149         _exchange), amountIn_);
150     return _exchange.swapExactInput(_path, amountIn_,
151         _amountOutMin, receiver_);
152 }
153 /// @inheritdoc ISwapper
154 function swapExactOutput(
155     address tokenIn_,
156     address tokenOut_,
157     uint256 amountOut_,
158     address receiver_
159 ) external returns (uint256 _amountIn) {
160     (, uint256 _amountInMax, IExchange _exchange, address[] memory _path) = getBestAmountIn(
161         tokenIn_,
162         tokenOut_,
163         amountOut_
```

```

163      );
164      address _caller = _msgSender();
165      IERC20(tokenIn_).safeTransferFrom(_caller, address(_exchange),
166      ↳ _amountInMax);
166      return _exchange.swapExactOutput(_path, amountOut_,
167      ↳ _amountInMax, _caller, receiver_);
167 }

```

The functions assume that the `IERC20(tokenIn_).safeTransferFrom()` call will transfer the full `amountIn_` and `_amountInMax` to the `Swapper` contract.

However, this may not be true if the `tokenIn_` is a transfer-on-fee token or a deflationary/rebasing token, causing the received amount to be less than the accounted amount.

If this occurs the swap call will revert as we can see in the proof of concept below.

#### Proof of Concept:

```

Calling -> contract_USDT.setParams(10, 20, {'from': contract_USDT.owner()}) SETTING A 1% FEE
Transaction sent: 0x5d2935a26f699f1796cf9d785ef1bd486fb50a3f0e8f7ebf7604367ab7ff71
Gas price: 0.0 gwei Gas limit: 600000000 Nonce: 1
USDT.setParams confirmed Block: 14891371 Gas used: 66957 (0.01%)

contract_USDT.balanceOf(user1) -> 50000000000
Calling -> contract_USDT.approve(contract_Swapper.address, 5000_000000, {'from': user1})
Transaction sent: 0x5f902b03d2e1bd4df9b599be2aceec1298bca22a4e5e92bf973a3bece9177d3
Gas price: 0.0 gwei Gas limit: 600000000 Nonce: 2
USDT.approve confirmed Block: 14891372 Gas used: 45961 (0.01%)

Calling -> contract_Swapper.swapExactInput(contract_USDT.address, contract_WETH.address, 5000_000000, user1.address, {'from': user1})
Transaction sent: 0x39b3d83670e11de125ab434b06a0cd49af8b67aca745b1fe51c6b40969cd83ed
Gas price: 0.0 gwei Gas limit: 600000000 Nonce: 3
Swapper.swapExactInput confirmed (TransferHelper: TRANSFER_FROM_FAILED) Block: 14891373 Gas used: 572322191 (95.39%)

```

#### Risk Level:

**Likelihood - 3**

**Impact - 5**

#### Recommendation:

It is recommended to get the exact received amount of `tokenIn_` by calculating the difference of `tokenIn_` balance before and after the transfer and use it in the `swapExactInput()` and `swapExactOutput()` calls.

## FINDINGS & TECH DETAILS

Remediation Plan:

**RISK ACCEPTED:** The Vesper Finance team accepted the risk of this finding.

## 3.5 (HAL-05) MAX DEVIATION TOO HIGH IN SWAPPERORACLE - MEDIUM

Description:

In the contract `SwapperOracle`, the constant `MAX_DEVIATION` sets the max deviation accepted between 2 different price providers:

**Listing 10: SwapperOracle.sol**

```
16 uint256 internal constant MAX_DEVIATION = 0.1e18;
```

**Listing 11: SwapperOracle.sol (Line 102)**

```
62 function quote(
63     address tokenIn_,
64     address tokenOut_,
65     uint256 amountIn_
66 ) external view returns (uint256 _amountOut) {
67     // 1. Get price from chainlink
68     uint256 _lastUpdatedAt;
69     (_amountOut, _lastUpdatedAt) = _quote(DataTypes.Provider.
↳ CHAINLINK, tokenIn_, tokenOut_, amountIn_);
70
71     // 2. If price from chainlink is OK return it
72     if (_amountOut > 0 && !_priceIsStale(_lastUpdatedAt)) {
73         return _amountOut;
74     }
75
76     // 3. Get price from fallback A
77     (uint256 _amountOutA, uint256 _lastUpdatedAtA) = _quote(
↳ fallbackProviderA, tokenIn_, tokenOut_, amountIn_);
78
79     // 4. If price from fallback A is OK and there isn't a
↳ fallback B, return price from fallback A
80     bool _aPriceOK = _amountOutA > 0 && !_priceIsStale(
↳ _lastUpdatedAtA);
81     if (fallbackProviderB == DataTypes.Provider.NONE) {
82         require(_aPriceOK, "fallback-a-failed");
83         return _amountOutA;
84     }
```

```
85
86     // 5. Get price from fallback B
87     (uint256 _amountOutB, uint256 _lastUpdatedAtB) = _quote(
88         ↳ fallbackProviderB, tokenIn_, tokenOut_, amountIn_);
89
90     // 6. If only one price from fallbacks is valid, return it
91     bool _bPriceOK = _amountOutB > 0 && !_priceIsStale(
92         ↳ _lastUpdatedAtB);
93     if (!_bPriceOK && _aPriceOK) {
94         return _amountOutA;
95     } else if (_bPriceOK && !_aPriceOK) {
96         return _amountOutB;
97     }
98
99     // 7. Check fallback prices deviation
100    require(_aPriceOK && _bPriceOK, "fallbacks-failed");
101    uint256 _deviation = _amountOutA > _amountOutB
102        ? ((_amountOutA - _amountOutB) * 1e18) / _amountOutA
103        : ((_amountOutB - _amountOutA) * 1e18) / _amountOutB;
104    require(_deviation <= MAX_DEVIATION, "prices-deviation-too-
105    ↳ high");
106 }
```

If the initial price provider does not return a correct price and there are 2 different fallback price providers set and both return correct prices, the price returned by these two will be compared with each other and if the difference is less than the 10% the price will be considered valid. Then the price of the fallback price provider A will be returned.

A 10% price deviation is way too high and may mean that the prices received are not correct.

Risk Level:

Likelihood - 3

Impact - 3

### Recommendation:

It is recommended to lower the MAX\_DEVIATION constant to a max. of 1-2%, instead of 10:

#### **Listing 12: SwapperOracle.sol**

```
16 uint256 internal constant MAX_DEVIATION = 0.02e18;
```

### Remediation Plan:

**SOLVED:** The [Vesper Finance team](#) addressed this issue by initially setting this parameter in the constructor. The `maxDeviation` can be adjusted later by the `governor` via the `UsingMaxDeviation.updateMaxDeviation()` function.

## 3.6 (HAL-06) STATE VARIABLES MISSING IMMUTABLE MODIFIER - INFORMATIONAL

### Description:

Some contracts have state variables that can be declared as `immutable` to save some gas.

The `immutable` keyword was added to Solidity in 0.6.5. State variables can be marked `immutable` which causes them to be read-only, but only assignable in the constructor.

### Code Location:

#### CTokenOracle.sol

- Line 18: `ISynthOracle public underlyingOracle;`
- Line 23: `address public wethLike;`

#### CurveLpTokenOracle.sol

- Line 13: `ICurveAddressProvider public addressProvider;`

#### SynthDefaultOracle.sol

- Line 31: `IChainlinkPriceProvider public chainlinkProvider;`

### Risk Level:

**Likelihood** - 1

**Impact** - 1

### Recommendation:

It is recommended to add the `immutable` modifier to the state variables mentioned.

## FINDINGS & TECH DETAILS

Remediation Plan:

**SOLVED:** The Vesper Finance team added the `immutable` modifier to these state variables.

# AUTOMATED TESTING

## 4.1 STATIC ANALYSIS REPORT

### Description:

Halborn used automated testing techniques to enhance the coverage of certain areas of the smart contracts in scope. Among the tools used was Slither, a Solidity static analysis framework. After Halborn verified the smart contracts in the repository and was able to compile them correctly into their ABIS and binary format, Slither was run against the contracts. This tool can statically verify mathematical relationships between Solidity variables to detect invalid or inconsistent usage of the contracts' APIs across the entire code-base.

### Slither results:

#### Governable.sol

```
Address.verifyAllResult(bool,bytes,string) (node_modules/@openzeppelin/contracts/utils/Address.sol#201-221) uses assembly
Address.functionCallWithValue(address,bytes,int256) (node_modules/@openzeppelin/contracts/utils/Address.sol#114-120) is never used and should be removed
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#assembly-useage

Different versions of Solidity is used:
- Version used: ['0.6.9', '^0.8.0', '^0.8.1', '^0.8.2']
- 0.8.0 (node_modules/@openzeppelin/contracts/proxy/utils/Initializable.sol#)
- 0.8.0 (node_modules/@openzeppelin/contracts/utils/Address.sol#)
- 0.8.0 (contracts/access/Governable.sol#)

Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#different-pragma-directives-are-used

Address.functionCall(address,bytes) (node_modules/@openzeppelin/contracts/utils/Address.sol#85-87) is never used and should be removed
Address.functionCall(address,bytes,bytes) (node_modules/@openzeppelin/contracts/utils/Address.sol#88-91) is never used and should be removed
Address.functionCallWithValue(address,bytes,int256) (node_modules/@openzeppelin/contracts/utils/Address.sol#114-120) is never used and should be removed
Address.functionCallWithValue(address,bytes,int256,string) (node_modules/@openzeppelin/contracts/utils/Address.sol#128-139) is never used and should be removed
Address.functionDelegateCall(address,bytes) (node_modules/@openzeppelin/contracts/proxy/Proxy.sol#)
Address.functionStaticCall(address,bytes) (node_modules/@openzeppelin/contracts/proxy/Proxy.sol#)
Address.functionStaticCall(address,bytes,bytes) (node_modules/@openzeppelin/contracts/proxy/Proxy.sol#)
Address.functionStaticCall(address,bytes,bytes,int256) (node_modules/@openzeppelin/contracts/proxy/Proxy.sol#)
Address.functionStaticCall(address,bytes,int256) (node_modules/@openzeppelin/contracts/proxy/Proxy.sol#)
Address.functionStaticCall(address,bytes,int256,string) (node_modules/@openzeppelin/contracts/proxy/Proxy.sol#)
Address.isContract(address) (node_modules/@openzeppelin/contracts/proxy/Proxy.sol#)
Address.sendValue(address,int256) (node_modules/@openzeppelin/contracts/proxy/Proxy.sol#)
Address.sendValue(address,int256,string) (node_modules/@openzeppelin/contracts/proxy/Proxy.sol#)
Address.transfer(address,int256) (node_modules/@openzeppelin/contracts/proxy/Proxy.sol#)
Governable.migrate() (node_modules/@openzeppelin/contracts/utils/Context.sol#21-23) is never used and should be removed
Initializable._initializable() (node_modules/@openzeppelin/contracts/proxy/Initializable.sol#)
Initializable._setInitializedVersion(uint256) (node_modules/@openzeppelin/contracts/proxy/Initializable.sol#133-140) is never used and should be removed
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#read-code

Pragma version="0.8.2" (node_modules/@openzeppelin/contracts/proxy/utils/Initializable.sol#8) allows old versions
Pragma version="0.8.1" (node_modules/@openzeppelin/contracts/proxy/utils/Initializable.sol#) allows old versions
Pragma version="0.8.0" (node_modules/@openzeppelin/contracts/proxy/utils/Initializable.sol#) necessitates a version too recent to be trusted. Consider deploying with 0.6.12/0.7.6/0.8.7
Pragma version="0.8.9" (contracts/interfaces/IGovernable.sol#) necessitates a version too recent to be trusted. Consider deploying with 0.6.12/0.7.6/0.8.7
pragma 0.8.9 is not recommended for deployment
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#incorrect-versions-of-solidity

Low level call in Address.sendValue(address,int256) (node_modules/@openzeppelin/contracts/proxy/utils/Initializable.sol#65):
- (success) = recipient.call{value: amount}() (node_modules/@openzeppelin/contracts/proxy/utils/Initializable.sol#40)
Low level call in Address.sendValue(address,int256,string) (node_modules/@openzeppelin/contracts/proxy/utils/Initializable.sol#128-139):
- (success,returnData) = target.call{value: value}(data) (node_modules/@openzeppelin/contracts/proxy/utils/Initializable.sol#137)
Low level call in Address.sendValue(address,int256,string,int256) (node_modules/@openzeppelin/contracts/proxy/utils/Initializable.sol#157-166):
- (success,returnData) = target.staticcall(data) (node_modules/@openzeppelin/contracts/proxy/utils/Initializable.sol#144)
Low level call in Address.functionDelegateCall(address,bytes) (node_modules/@openzeppelin/contracts/proxy/Proxy.sol#154-159):
- (success,returnData) = target.delegatecall(data) (node_modules/@openzeppelin/contracts/proxy/Proxy.sol#155)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#low-level-calls

Function Governor._Governable_init() (contracts/access/Governable.sol#9-12) is not in mixedCase
Parameter Governor.transferOwnership(address)_proposeGovernor (contracts/access/Governable.sol#6) is not in mixedCase
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#conformance-to-solidity-naming-conventions
```

#### ChainlinkArbitrumPriceProvider.sol

```
Address.verifyAllResult(bool,bytes,string) (node_modules/@openzeppelin/contracts/utils/Address.sol#201-221) uses assembly
- INLINE ASM (node_modules/@openzeppelin/Contracts/Address.sol#21-216)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#assembly-useage

Different versions of Solidity is used:
- Version used: ['0.6.9', '^0.8.0', '^0.8.1', '^0.8.2']
- 0.8.0 (node_modules/@chainlink/contracts/src/v0.8/interfaces/AggregatorV3Interface.sol#)
- 0.8.0 (node_modules/@openzeppelin/contracts/proxy/utils/Initializable.sol#)
- 0.8.0 (node_modules/@openzeppelin/contracts/token/ERC20/IERC20.sol#)
- 0.8.0 (node_modules/@openzeppelin/contracts/token/ERC20/IERC20Metadata.sol#)
- 0.8.0 (node_modules/@openzeppelin/contracts/token/ERC20/IERC20Permit.sol#)
- 0.8.0 (node_modules/@openzeppelin/contracts/token/ERC20/IERC20URI.sol#)
- 0.8.0 (node_modules/@openzeppelin/contracts/math/Math.sol#)
- 0.8.0 (node_modules/@openzeppelin/contracts/math/SafeCast.sol#)
- 0.8.0 (node_modules/@openzeppelin/contracts/math/SafeMath.sol#)
- 0.8.9 (contracts/core/ChainlinkArbitrumPriceProvider.sol#9)
- 0.8.9 (contracts/interfaces/ICore.sol#)
- 0.8.9 (contracts/interfaces/ICore/ICore.sol#)
- 0.8.9 (contracts/interfaces/Core/ICore.sol#13)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#different-pragma-directives-are-used

Address.functionCall(address,bytes) (node_modules/@openzeppelin/Contracts/utils/Address.sol#85-87) is never used and should be removed
Address.functionCall(address,bytes,bytes) (node_modules/@openzeppelin/Contracts/utils/Address.sol#88-91) is never used and should be removed
Address.functionCallWithValue(address,bytes,int256) (node_modules/@openzeppelin/Contracts/utils/Address.sol#114-120) is never used and should be removed
Address.functionCallWithValue(address,bytes,int256,string) (node_modules/@openzeppelin/Contracts/utils/Address.sol#128-139) is never used and should be removed
Address.functionDelegatedCall(address,bytes) (node_modules/@openzeppelin/Contracts/proxy/Proxy.sol#)
Address.functionDelegatedCall(address,bytes,bytes) (node_modules/@openzeppelin/Contracts/proxy/Proxy.sol#)
Address.functionStaticCall(address,bytes) (node_modules/@openzeppelin/Contracts/proxy/Proxy.sol#)
Address.functionStaticCall(address,bytes,bytes) (node_modules/@openzeppelin/Contracts/proxy/Proxy.sol#)
Address.functionStaticCall(address,bytes,bytes,int256) (node_modules/@openzeppelin/Contracts/proxy/Proxy.sol#)
Address.functionStaticCall(address,bytes,int256) (node_modules/@openzeppelin/Contracts/proxy/Proxy.sol#)
Address.functionStaticCall(address,bytes,int256,string) (node_modules/@openzeppelin/Contracts/proxy/Proxy.sol#)
Address.sendValue(address,int256) (node_modules/@openzeppelin/Contracts/proxy/Proxy.sol#)
Address.sendValue(address,int256,string) (node_modules/@openzeppelin/Contracts/proxy/Proxy.sol#)
Address.transfer(address,int256) (node_modules/@openzeppelin/Contracts/proxy/Proxy.sol#)
Governable._Governable_init() (contracts/access/Governable.sol#6) is never used and should be removed
Governable._Governable_init() (contracts/access/Governable.sol#13) is never used and should be removed
```

## ChainlinkAvalanchePriceProvider.sol

```

Address.verifyAllRevert((bool,bytes,string)) (node_modules/@openzeppelin/contracts/proxy/utils/Initializable.sol#129-131) is never used and should be removed
Initializable.setInitializedVersion(uint3) (node_modules/@openzeppelin/contracts/proxy/utils/Initializable.sol#131-145) is never used and should be removed
Math.average(uint256,uint256) (node_modules/@openzeppelin/contracts/math/Math.sol#28-31) is never used and should be removed
Math.ceilDiv(uint256,uint256) (node_modules/@openzeppelin/contracts/math/Math.sol#149-152) is never used and should be removed
SafeCast.toInt20(int256) (node_modules/@openzeppelin/contracts/math/Math.sol#152-159) is never used and should be removed
SafeCast.toInt20(int256) (node_modules/@openzeppelin/contracts/math/Math.sol#206-209) is never used and should be removed
SafeCast.toInt20(int256) (node_modules/@openzeppelin/contracts/math/Math.sol#210-213) is never used and should be removed
SafeCast.toInt20(int256) (node_modules/@openzeppelin/contracts/math/Math.sol#109-191) is never used and should be removed
SafeCast.toInt20(int256) (node_modules/@openzeppelin/contracts/math/Math.sol#170-173) is never used and should be removed
SafeCast.toInt20(int256) (node_modules/@openzeppelin/contracts/math/Math.sol#174-177) is never used and should be removed
SafeCast.toInt128(uint256) (node_modules/@openzeppelin/contracts/math/Math.sol#47-50) is never used and should be removed
SafeCast.toInt16(uint256) (node_modules/@openzeppelin/contracts/math/Math.sol#107-110) is never used and should be removed
SafeCast.toInt32(uint256) (node_modules/@openzeppelin/contracts/math/Math.sol#92-95) is never used and should be removed
SafeCast.toInt4(uint256) (node_modules/@openzeppelin/contracts/math/Math.sol#77-80) is never used and should be removed
SafeCast.toInt8(uint256) (node_modules/@openzeppelin/contracts/math/Math.sol#41-44) is never used and should be removed
SafeCast.toInt96(uint256) (node_modules/@openzeppelin/Contracts/math/MathCast.sol#62-65) is never used and should be removed
SafeCast.toInt196(uint256) (node_modules/@openzeppelin/Contracts/math/MathCast.sol#62-65) is never used and should be removed
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#read-code

Pragma version=0.8.0 (node_modules/@chainlink/contracts/src/v0.8/interfaces/IAggregatorV3Interface.sol#2) allows old versions
Pragma version=0.8.2 (node_modules/@openzeppelin/contracts/proxy/v1/Initializable.sol#4) allows old versions
Pragma version=0.8.3 (node_modules/@openzeppelin/contracts/token/ERC20/IERC20.sol#4) allows old versions
Pragma version=0.8.4 (node_modules/@openzeppelin/contracts/token/ERC20/IERC20Metadata.sol#4) allows old versions
Pragma version=0.8.5 (node_modules/@openzeppelin/contracts/token/ERC20/IERC20.sol#4) allows old versions
Pragma version=0.8.6 (node_modules/@openzeppelin/contracts/token/ERC20/IERC20.sol#4) allows old versions
Pragma version=0.8.7 (node_modules/@openzeppelin/contracts/token/ERC20/IERC20.sol#4) allows old versions
Pragma version=0.8.8 (node_modules/@openzeppelin/contracts/token/ERC20/IERC20.sol#4) allows old versions
Pragma version=0.8.9 (node_modules/@openzeppelin/contracts/token/ERC20/IERC20.sol#4) necessitates a version too recent to be trusted. Consider deploying with 0.8.12/0.7.6/0.8.7
Pragma version=0.8.9 (contracts/core/ChainlinkPriceProvider.sol#3) necessitates a version too recent to be trusted. Consider deploying with 0.8.12/0.7.6/0.8.7
Pragma version=0.8.9 (contracts/interface/IDelegatable.sol#3) necessitates a version too recent to be trusted. Consider deploying with 0.8.12/0.7.6/0.8.7
Pragma version=0.8.9 (contracts/interface/IGovernable.sol#3) necessitates a version too recent to be trusted. Consider deploying with 0.8.12/0.7.6/0.8.7
Pragma version=0.8.9 (contracts/interface/ITraceable.sol#3) necessitates a version too recent to be trusted. Consider deploying with 0.8.12/0.7.6/0.8.7
solc=0.8.8 is not recommended for deployment
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#incorrect-versions-of-solidity

low level call in Address.sendValue(address,uint256) (node_modules/@openzeppelin/contracts/utils/Address.sol#6-6):
- (success, returnData) = target.call{value: value}();
Low level call in Address.functionCallWithValue(address,bytes,uint256,string) (node_modules/@openzeppelin/contracts/Address.sol#120-139):
- (success, returnData) = target.functionCallWithValue{value: value}(bytes);
Low level call in Address.functionDelegateCall(address,bytes,bytes) (node_modules/@openzeppelin/contracts/Address.sol#157-166):
- (success, returnData) = target.staticcall{data: data}(bytes);
Low level call in Address.functionDelegateCall(address,bytes,string) (node_modules/@openzeppelin/contracts/Address.sol#184-193):
- (success, returnData) = target.functionDelegateCall{data: data}(bytes, string);
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#solidity-level-calls

Function Governable._Governable_init() (contracts/access/Governable.sol#8-9) is not in mixedCase
Parameter Governable.transferOwnership(address) (contracts/access/Governable.sol#8) is not in mixedCase
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#conformance-to-solidity-naming-conventions

Variable ChainlinkPriceProvider._aggregatorOf(address) (aggregator/(contracts/core/ChainlinkPriceProvider.sol#7)) is too similar to ChainlinkPriceProvider.updateAggregator(address,AggregatorV3Interface.sol#1) aggregator_ (contracts/core/ChainlinkPriceProvider.sol#1)
Variable ChainlinkPriceProvider._aggregatorOf(address) (aggregator/(contracts/core/ChainlinkPriceProvider.sol#7)) is too similar to ChainlinkPriceProvider.aggregators_ (contracts/core/ChainlinkPriceProvider.sol#29)
Variable ChainlinkPriceProvider._updateAggregator(address) (aggregatorOf/ChainlinkPriceProvider.sol#9) is too similar to ChainlinkPriceProvider.aggregators_ (contracts/core/ChainlinkPriceProvider.sol#25)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#variable-names-are-too-similar

getPriceUsed(address) should be declared external:
- ChainlinkPriceProvider.getPriceUsed(address) (contracts/core/ChainlinkPriceProvider.sol#31-39)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#public-function-that-could-be-declared-external

```





## UniswapV2LikePriceProvider.sol

```

Address.functionCall(address,bytes) (node_modules/@openzeppelin/contracts/utils/Address.sol#88-87) is never used and should be removed
Address.functionCall(address,bytes,string) (node_modules/@openzeppelin/contracts/utils/Address.sol#90-101) is never used and should be removed
Address.functionCallWithValue(address,bytes,uint256) (node_modules/@openzeppelin/contracts/utils/Address.sol#114-120) is never used and should be removed
Address.functionDelegateCall(address,bytes) (node_modules/@openzeppelin/contracts/utils/Address.sol#144-154) is never used and should be removed
Address.functionStaticCall(address,bytes,string) (node_modules/@openzeppelin/contracts/utils/Address.sol#157-166) is never used and should be removed
Address.isContract(address) (node_modules/@openzeppelin/contracts/utils/Address.sol#42) is never used and should be removed
Address.sendValue(address,uint256) (node_modules/@openzeppelin/contracts/utils/Address.sol#125-131) is never used and should be removed
Address.verifyCallResult(bytes,string) (node_modules/@openzeppelin/contracts/utils/Address.sol#201-221) is never used and should be removed
Context._msgData() (node_modules/@openzeppelin/contracts/utils/Context.sol#23) is never used and should be removed
Context._msgDataBytes() (node_modules/@openzeppelin/contracts/utils/Context.sol#32) is never used and should be removed
Initializable._disableInitializers() (node_modules/@openzeppelin/contracts/proxy/utils/Initializable.sol#129-131) is never used and should be removed
Initializable._setInitializationVersion(uint32) (node_modules/@openzeppelin/contracts/proxy/utils/Initializable.sol#133-145) is never used and should be removed
Math._ceilDiv(uint256,uint256) (node modules/@openzeppelin/Contracts/utils/Math/Math.sol#89-92) is never used and should be removed
Math._maxUint256(uint256) (node modules/@openzeppelin/Contracts/utils/Math/Math.sol#13-15) is never used and should be removed
Reference: https://github.com/crytic/slither/wikis/Detector-Documentation#read-code

Frama version>0.8.2 (node_modules/@openzeppelin/contracts/proxy/utils/Initializable.sol#4) allows old versions
Frama version<=0.8.0 (node_modules/@openzeppelin/contracts/proxy/utils/Initializable.sol#4) does not allow old versions
Frama version>0.8.0 (node_modules/@openzeppelin/contracts/math/Math.sol#8) allows old versions
Frama version>0.8.0 (contracts/acces/Governable.sol#3) necessitates a version too recent to be trusted. Consider deploying with 0.6.12/0.7.6/0.8.7
Frama version<=0.8.0 (contracts/interfaces/IGovernable.sol#3) necessitates a version too recent to be trusted. Consider deploying with 0.6.12/0.7.6/0.8.7
Frama version>0.8.0 (contracts/interfaces/IFriceProvider.sol#3) necessitates a version too recent to be trusted. Consider deploying with 0.6.12/0.7.6/0.8.7
Frama version<=0.8.0 (contracts/libraries/DataType.sol#3) necessitates a version too recent to be trusted. Consider deploying with 0.6.12/0.7.6/0.8.7
Frama version>0.8.0 (contracts/libraries/IFriceProvider.sol#3) necessitates a version too recent to be trusted. Consider deploying with 0.6.12/0.7.6/0.8.7
solo<-0.8.0 is not recommended for deployment
Reference: https://github.com/crytic/slither/wikis/Detector-Documentation#incorrect-versions-of-solidity

low level call in Address.sendValue(address,uint256) (node_modules/@openzeppelin/contracts/utils/Address.sol#64):
- (success, returndata) = target.delegatecall(data) (node_modules/@openzeppelin/contracts/utils/Address.sol#128-139);
Low level call in Address.functionCallWithValue(address,bytes,uint256,string) (node_modules/@openzeppelin/contracts/utils/Address.sol#137):
- (success, returndata) = target.callValue(value)(data) (node_modules/@openzeppelin/contracts/utils/Address.sol#157-166);
Low level call in Address.functionDelegateCall(address,bytes) (node_modules/@openzeppelin/contracts/utils/Address.sol#144):
- (success, returndata) = target.staticCall(data) (node_modules/@openzeppelin/contracts/utils/Address.sol#144);
Low level call in Address.functionStaticCall(address,bytes,string) (node_modules/@openzeppelin/contracts/utils/Address.sol#157-166):
- (success, returndata) = target.staticCall(data) (node_modules/@openzeppelin/contracts/utils/Address.sol#157-166);
Reference: https://github.com/crytic/slither/wikis/Detector-Documentation#low-level-calls

Function Governor._Governable_init() (contracts/acces/Governable.sol#18-2) is not in mixedCase
Parameter Governor.transferOwnership(address) (contracts/acces/Governable.sol#6) is not in mixedCase
Reference: https://github.com/crytic/slither/wikis/Detector-Documentation#solidity-naming-conventions

Variable PriceProvidersAggregator.quote(DataType.Provider,address,DataType.Provider,address,uint256) (node_modules/@openzeppelin/contracts/acces/PriceProvidersAggregator.sol#6) is too similar to PriceProvidersAggregator.quote(DataType.Provider,address,DataType.Provider,address,uint256).providerIn (contracts/core/PriceProvidersAggregator.sol#56)
Variable PriceProvidersAggregator.quote(DataType.Provider,address,DataType.Provider,address,uint256).providerOut (contracts/core/PriceProvidersAggregator.sol#63) is too similar to PriceProvidersAggregator.quote(DataType.Provider,address,DataType.Provider,address,uint256).providerIn (contracts/core/PriceProvidersAggregator.sol#56)
Variable PriceProvidersAggregator.providerIn(DataType.Provider,address,DataType.Provider,address,uint256) (node_modules/@openzeppelin/contracts/acces/PriceProvidersAggregator.sol#17) is too similar to PriceProvidersAggregator.providerOut(DataType.Provider,address,DataType.Provider,address,uint256) (node_modules/@openzeppelin/contracts/acces/PriceProvidersAggregator.sol#18)
Variable PriceProvidersAggregator.providerOut(DataType.Provider,address,DataType.Provider,address,uint256) (node_modules/@openzeppelin/contracts/acces/PriceProvidersAggregator.sol#17) is too similar to PriceProvidersAggregator.providerIn(DataType.Provider,address,DataType.Provider,address,uint256) (node_modules/@openzeppelin/contracts/acces/PriceProvidersAggregator.sol#18)
Variable PriceProvidersAggregator.priceProvider(DataType.Provider,address,DataType.Provider,address,uint256) (node_modules/@openzeppelin/contracts/acces/PriceProvidersAggregator.sol#11) is too similar to PriceProvidersAggregator.priceProviders (contracts/core/PriceProvidersAggregator.sol#22)
Variable PriceProvidersAggregator.priceProviders(DataType.Provider,address,DataType.Provider,address,uint256) (node_modules/@openzeppelin/contracts/acces/PriceProvidersAggregator.sol#11) is too similar to PriceProvidersAggregator.priceProviders (contracts/core/PriceProvidersAggregator.sol#22)
Variable PriceProvidersAggregator.quote(DataType.Provider,address,DataType.Provider,address,uint256) (node_modules/@openzeppelin/contracts/acces/PriceProvidersAggregator.sol#56) is too similar to IPriceProvidersAggregator.quote(DataType.Provider,address,DataType.Provider,address,uint256).providerIn (contracts/core/PriceProvidersAggregator.sol#45)
Variable PriceProvidersAggregator.quote(DataType.Provider,address,DataType.Provider,address,uint256).providerOut (contracts/core/PriceProvidersAggregator.sol#63) is too similar to IPriceProvidersAggregator.quote(DataType.Provider,address,DataType.Provider,address,uint256).providerIn (contracts/core/PriceProvidersAggregator.sol#45)
Reference: https://github.com/crytic/slither/wikis/Detector-Documentation#variable-names-are-too-similar

quote(DataType.Provider,address,DataType.Provider,address,uint256) should be declared external:
- PriceProvidersAggregator.quote(DataType.Provider,address,DataType.Provider,address,uint256) (contracts/core/PriceProvidersAggregator.sol#6-70)
Reference: https://github.com/crytic/slither/wikis/Detector-Documentation#public-function-that-could-be-declared-external

UniswapV2OracleLibrary.currentBlockTimestamp() (node_modules/uniswap/v2-periphery/contracts/libraries/UniswapV2OracleLibrary.sol#11-13) uses a weak PENG: "uint32(block.timestamp % 2 ** 32) (node_modules/uniswap/v2-periphery/contracts/libraries/UniswapV2OracleLibrary.sol#11-13)
Reference: https://github.com/crytic/slither/wikis/Detector-Documentation#weak-PENG

FixedPoint._min(FixedPoint.uq128, uint256) (node_modules/uniswap/v2-core/contracts/libraries/FixedPoint.sol#4) is a local variable never initialized
Reference: https://github.com/crytic/slither/wikis/Detector-Documentation#uninitialized-local-variable

UniswapV2LikePriceProvider._computeSqrt(address,uint256,address,math.sqrtToken) (contracts/core/UniswapV2LikePriceProvider.sol#54) lacks a zero-check on :
- nativeToken = ratioToken (contract/core/UniswapV2LikePriceProvider.sol#59)
Reference: https://github.com/crytic/slither/wikis/Detector-Documentation#missing-zero-address-validation

UniswapV2OracleLibrary.currentCumulativePrices(address) (node_modules/uniswap/v2-periphery/contracts/libraries/UniswapV2OracleLibrary.sol#16-34) uses timestamp for comparisons
Dangerous comparisons:
- block.timestamp != block.timestamp (node_modules/uniswap/v2-periphery/contracts/libraries/UniswapV2OracleLibrary.sol#25)
Reference: https://github.com/crytic/slither/wikis/Detector-Documentation#block-timestamp

Address.verifyCallResult(bytes,string) (node_modules/@openzeppelin/contracts/utils/Address.sol#201-221) uses assembly
Reference: https://github.com/crytic/slither/wikis/Detector-Documentation#assembly-use

Different versions of Solidity is used:
- Version used [0.8.9, '>0.8.0', '>0.8.1', '>0.8.2']
- 0.8.2 (node_modules/@openzeppelin/contract/proxy/utils/Initializable.sol#4)
- 0.8.3 (node_modules/@openzeppelin/contract/proxy/utils/Initializable.sol#4)
- 0.8.4 (node_modules/@openzeppelin/contract/proxy/utils/Initializable.sol#4)
- 0.8.5 (node_modules/@openzeppelin/contract/proxy/utils/Initializable.sol#4)
- 0.8.6 (node_modules/@openzeppelin/contract/proxy/utils/Initializable.sol#4)
- 0.8.7 (node_modules/@openzeppelin/contract/proxy/utils/Initializable.sol#4)
- 0.8.8 (node_modules/@openzeppelin/contract/proxy/utils/Initializable.sol#4)
- 0.8.9 (node_modules/@openzeppelin/contract/proxy/utils/Initializable.sol#4)
- 0.8.10 (node_modules/@openzeppelin/contract/proxy/utils/Initializable.sol#4)
- 0.8.9 (contracts/acces/Governable.sol#3)
- 0.8.10 (contracts/acces/Governable.sol#3)
- 0.8.9 (contracts/interfaces/IFriceProvider.sol#3)
- 0.8.10 (contracts/interfaces/IFriceProvider.sol#3)
- 0.8.9 (contracts/interfaces/IFriceProvider.sol#3)
- 0.8.10 (contracts/interfaces/IFriceProvider.sol#3)
Reference: https://github.com/crytic/slither/wikis/Detector-Documentation#different-pragma-directives-are-used

Address.functionCall(address,bytes) (node_modules/@openzeppelin/contracts/utils/Address.sol#88-87) is never used and should be removed
Address.functionCallWithValue(address,bytes,uint256) (node_modules/@openzeppelin/contracts/utils/Address.sol#101-109) is never used and should be removed
Address.functionDelegateCall(address,bytes) (node_modules/@openzeppelin/contracts/utils/Address.sol#114-120) is never used and should be removed
Address.functionStaticCall(address,bytes,string) (node_modules/@openzeppelin/contracts/utils/Address.sol#154-159) is never used and should be removed
Address.functionStaticCall(address,bytes) (node_modules/@openzeppelin/contracts/utils/Address.sol#157-166) is never used and should be removed
Address.isContract(address) (node_modules/@openzeppelin/contracts/utils/Address.sol#42) is never used and should be removed
Address.sendValue(address,uint256) (node_modules/@openzeppelin/contracts/utils/Address.sol#125-131) is never used and should be removed
Address.verifyCallResult(bytes,string) (node_modules/@openzeppelin/contracts/utils/Address.sol#201-221) is never used and should be removed
Context._msgData() (node_modules/@openzeppelin/contracts/utils/Context.sol#23) is never used and should be removed
FixedPoint._decimals(FixedPoint.uq128) (node_modules/uniswap/v2-core/contracts/libraries/FixedPoint.sol#5) is never used and should be removed
FixedPoint._encodex4(FixedPoint.uq128) (node_modules/uniswap/v2-core/contracts/libraries/FixedPoint.sol#33) is never used and should be removed
FixedPoint._reciprocal(FixedPoint.uq128) (node_modules/uniswap/v2-core/contracts/libraries/FixedPoint.sol#67-70) is never used and should be removed
FixedPoint._sqrt(FixedPoint.uq128) (node_modules/uniswap/v2-core/contracts/libraries/FixedPoint.sol#111) is never used and should be removed
Governable._Governable_init() (contracts/acces/Governable.sol#18-2) is never used and should be removed
Initializable._disableInitializers() (node_modules/@openzeppelin/contracts/proxy/utils/Initializable.sol#129-131) is never used and should be removed
Math._average(uint256,uint256) (node_modules/@openzeppelin/Contracts/utils/Math/Math.sol#39-42) is never used and should be removed
Math._ceilDiv(uint256,uint256) (node_modules/@openzeppelin/Contracts/utils/Math/Math.sol#13-15) is never used and should be removed
Math._maxUint256(uint256) (node_modules/@openzeppelin/Contracts/utils/Math/Math.sol#15-18) is never used and should be removed
Reference: https://github.com/crytic/slither/wikis/Detector-Documentation#incorrect-versions-of-solidity

low level call in Address.sendValue(address,uint256) (node_modules/@openzeppelin/contracts/utils/Address.sol#65):
- (success) = recipient.call{value: amount}() (node_modules/@openzeppelin/contracts/utils/Address.sol#65)
Low level call in Address.functionCallWithValue(address,bytes,uint256,string) (node_modules/@openzeppelin/contracts/utils/Address.sol#128-139):
- (success) = target.callValue(value)(data) (node_modules/@openzeppelin/contracts/utils/Address.sol#137);
Low level call in Address.functionDelegateCall(address,bytes,string) (node_modules/@openzeppelin/contracts/utils/Address.sol#157-166):
- (success, returndata) = target.delegatecall(data) (node_modules/@openzeppelin/contracts/utils/Address.sol#157-166);
Low level call in Address.functionStaticCall(address,bytes,string) (node_modules/@openzeppelin/contracts/utils/Address.sol#157-166):
- (success, returndata) = target.staticCall(data) (node_modules/@openzeppelin/contracts/utils/Address.sol#157-166);
Reference: https://github.com/crytic/slither/wikis/Detector-Documentation#low-level-calls

```

```

Struct FixedPoint.uq112x112 (node_modules/@uniswap/lib/contracts/libraries/FixedPoint.sol#11-19) is not in CapWords
Struct FixedPoint.uq4x112 (node_modules/@uniswap/lib/contracts/libraries/FixedPoint.sol#11-19) is not in CapWords
Function UniSwapV2Pair._transferFeeRate_(uint256)(node_modules/@uniswap/v2-core/contracts/interfaces/IUniSwapV2Pair.sol#10) is not in mixedCase
Function UniSwapV2Pair._PERMIT_TYPE(uint256)(node_modules/@uniswap/v2-periphery/contracts/libraries/UniSwapV2OracleLibrary.sol#13) is not in mixedCase
Function Governable._governable_init()(contracts/access/Governable.sol#5-12) is not in mixedCase
Parameter Governor.GovernableAddress()(contracts/access/Governable.sol#13-16) is not in mixedCase
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#conformance-to-solidity-naming-conventions

Variable UniSwapV2OracleLibrary.currentCumulativePrices(address).priceCumulative (node_modules/@uniswap/v2-periphery/contracts/libraries/UniSwapV2OracleLibrary.sol#19) is too similar to UniSwapV2OracleLibrary.currentCumulativePrices(add)
read.getPriceCumulative (node_modules/@uniswap/v2-periphery/contracts/libraries/UniSwapV2OracleLibrary.sol#19)
Variable UniSwapV2LikePriceProvider._updateIfNeeded(IUniSwapV2Pair,uint256).priceCumulative (contracts/core/UniSwapV2LikePriceProvider.sol#19) is too similar to UniSwapV2LikePriceProvider._updateIfNeeded(IUniSwapV2Pair,uint256).priceC
umentation#public-function-that-could-be-declared-external
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#variable-names-are-too-similar

updateDefaultTwapPeriod(uint32) should be declared external;
- UniSwapV2LikePriceProvider.updateDefaultTwapPeriod(uint32) (contracts/core/UniSwapV2LikePriceProvider.sol#18-22)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#public-function-that-could-be-declared-external



## UniSwapV3PriceProvider.sol


Address.verifyCallResult(bool,bytes,string) (node_modules/@openzeppelin/contracts/utils/Address.sol#201-221) uses assembly
- INLINE ASM (node_modules/@openzeppelin/contracts/utils/Address.sol#213-214)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#assembly=usage

Different versions of Solidity is used:
- Version used: '0.8.0', '"0.8.1", "'0.8.2'"
- 0.8.2 (node_modules/@openzeppelin/contracts/proxy/utils/Initializable.sol#4)
- 0.8.1 (node_modules/@openzeppelin/contracts/uups/Initializable.sol#4)
- 0.8.0 (node_modules/@openzeppelin/contracts/uups/Initializable.sol#4)
- 0.8.9 (contracts/access/Governable.sol#3)
  - (contracts/interface/IERC20.sol#18)
  - 0.8.9 (contracts/interface/IERC20Metadata.sol#3)
  - 0.8.9 (contracts/interface/IUniswapV3LikePriceProvider.sol#3)
  - 0.8.9 (contracts/interface/IERC20.sol#4)
  - 0.8.9 (contracts/interface/IUniswapV3CrossPoolOracle.sol#3)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#different-pragma-directives=are-used

Address.functionCall(address,bytes) (node_modules/@openzeppelin/Contracts/Util/Address.sol#85-87) is never used and should be removed
Address.functionCallWithValue(address,bytes,int256) (node_modules/@openzeppelin/Contracts/Util/Address.sol#89-101) is never used and should be removed
Address.functionCallStatic(address,bytes,int256) (node_modules/@openzeppelin/Contracts/Util/Address.sol#114-120) is never used and should be removed
Address.functionDelegateCall(address,bytes) (node_modules/@openzeppelin/Contracts/Util/Address.sol#133-134) is never used and should be removed
Address.functionDelegateCall(address,bytes,int256) (node_modules/@openzeppelin/Contracts/Util/Address.sol#174-176) is never used and should be removed
Address.functionDelegateCall(address,bytes,int256,int256) (node_modules/@openzeppelin/Contracts/Util/Address.sol#194-195) is never used and should be removed
Address.functionDelegateCall(address,bytes,int256,int256,int256) (node_modules/@openzeppelin/Contracts/Util/Address.sol#196-198) is never used and should be removed
Address.functionStaticCall(address,bytes) (node_modules/@openzeppelin/Contracts/Util/Address.sol#199-200) is never used and should be removed
Address.isContract(address) (node_modules/@openzeppelin/Contracts/Util/Address.sol#205-206) is never used and should be removed
Address.sendValue(address,int256) (node_modules/@openzeppelin/Contracts/Util/Address.sol#209-215) is never used and should be removed
Address.transferValue(address,int256) (node_modules/@openzeppelin/Contracts/Util/Address.sol#216-218) is never used and should be removed
Context._msgData() (node_modules/@openzeppelin/Contracts/Util/Context.sol#4-6) is never used and should be removed
Governable._governable_init()(contracts/access/Governable.sol#4-9) is never used and should be removed
Initializable._setInitializationVersion(uint)(contracts/proxy/Initializable.sol#133-140) is never used and should be removed
SafeCast.toll128(uint256) (node_modules/@openzeppelin/Contracts/Util/Math/SafeCast.sol#152-155) is never used and should be removed
SafeCast.toll3224(uint256) (node_modules/@openzeppelin/Contracts/Util/Math/SafeCast.sol#156-159) is never used and should be removed
SafeCast.toll3236(uint256) (node_modules/@openzeppelin/Contracts/Util/Math/SafeCast.sol#159-161) is never used and should be removed
SafeCast.toll536(uint256) (node_modules/@openzeppelin/Contracts/Util/Math/SafeCast.sol#162-164) is never used and should be removed
SafeCast.toll8536(uint256) (node_modules/@openzeppelin/Contracts/Util/Math/SafeCast.sol#165-167) is never used and should be removed
SafeCast.toll1281256(uint256) (node_modules/@openzeppelin/Contracts/Util/Math/SafeCast.sol#168-170) is never used and should be removed
SafeCast.toll1281256(int256) (node_modules/@openzeppelin/Contracts/Util/Math/SafeCast.sol#171-173) is never used and should be removed
SafeCast.toll1281256(int256,int256) (node_modules/@openzeppelin/Contracts/Util/Math/SafeCast.sol#174-176) is never used and should be removed
SafeCast.toll1281256(int256,int256,int256) (node_modules/@openzeppelin/Contracts/Util/Math/SafeCast.sol#177-179) is never used and should be removed
SafeCast.toll321256(uint256) (node_modules/@openzeppelin/Contracts/Util/Math/SafeCast.sol#180-182) is never used and should be removed
SafeCast.toll321256(int256) (node_modules/@openzeppelin/Contracts/Util/Math/SafeCast.sol#183-185) is never used and should be removed
SafeCast.toll321256(int256,int256) (node_modules/@openzeppelin/Contracts/Util/Math/SafeCast.sol#186-188) is never used and should be removed
SafeCast.toll321256(int256,int256,int256) (node_modules/@openzeppelin/Contracts/Util/Math/SafeCast.sol#189-191) is never used and should be removed
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#incorrect-versions-of-solidity

Pragma version="0.8.2" (node_modules/@openzeppelin/Contracts/Proxy/Initializable.sol#4) allows old versions
Pragma version="0.8.0" (node_modules/@openzeppelin/Contracts/Util/Address.sol#4) allows old versions
Pragma version="0.8.0" (node_modules/@openzeppelin/Contracts/Util/Context.sol#4) allows old versions
Pragma version="0.8.0" (node_modules/@openzeppelin/Contracts/Util/Context.sol#10-11) is never used and should be removed
Pragma version="0.8.0" (contracts/core/IUniswapV3LikePriceProvider.sol#3) necessitates a version too recent to be trusted. Consider deploying with 0.6.12/0.7.6/0.8.7
Pragma version="0.8.0" (contracts/interface/IUniswapV3LikePriceProvider.sol#3) necessitates a version too recent to be trusted. Consider deploying with 0.6.12/0.7.6/0.8.7
Pragma version="0.8.0" (contracts/interface/IERC20.sol#3) necessitates a version too recent to be trusted. Consider deploying with 0.6.12/0.7.6/0.8.7
Pragma version="0.8.0" (contracts/interface/IUniswapV3CrossPoolOracle.sol#3) necessitates a version too recent to be trusted. Consider deploying with 0.6.12/0.7.6/0.8.7
Pragma version="0.8.0" (contracts/interface/IERC20.sol#4) necessitates a version too recent to be trusted. Consider deploying with 0.6.12/0.7.6/0.8.7
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#incorrect-versions-of-solidity

low level call in Address.sendValue(address,int256) (node_modules/@openzeppelin/Contracts/Util/Address.sol#85):
- (success) = recipient.call(value(amount)) (node_modules/@openzeppelin/Contracts/Util/Address.sol#85)
low level call in Address.functionCallWithValue(address,bytes,int256) (node_modules/@openzeppelin/Contracts/Util/Address.sol#89):
- (success) = target.functionCallWithValue(value,callData) (node_modules/@openzeppelin/Contracts/Util/Address.sol#89)
low level call in Address.functionCall(address,bytes) (node_modules/@openzeppelin/Contracts/Util/Address.sol#114):
- (success, returnData) = target.staticcall(data) (node_modules/@openzeppelin/Contracts/Util/Address.sol#114)
low level call in Address.functionCall(address,bytes,int256) (node_modules/@openzeppelin/Contracts/Util/Address.sol#133-139):
- (success, returnData) = target.functionCallWithValue(value,callData) (node_modules/@openzeppelin/Contracts/Util/Address.sol#133-139)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#low-level-calls

Function Governable._governable_init()(contracts/access/Governable.sol#15-18) is not in mixedCase
Parameter Governable.transferOwnership(address) _proposedOwner (contracts/access/Governable.sol#6) is not in mixedCase
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#conformance-to-solidity-naming-conventions

updateDefaultTwapPeriod(uint32) should be declared external;
- UniSwapV3PriceProvider.updateDefaultTwapPeriod(uint32) (contracts/core/UniSwapV3PriceProvider.sol#96-101)
- UniSwapV3PriceProvider.updateDefaultPoolFee(uint24) (contracts/core/UniSwapV3PriceProvider.sol#104-107)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#public-function-that-could-be-declared-external



## Oracle.sol


Oracle.setUSDEquivToken(address) (node_modules/@openzeppelin/contracts/interfaces/IERC20.sol#4) lacks a zero-check on :
Oracle.setUSDEquivToken(address) = setUSDEquivToken_. (contracts/erc20/Oracle.sol#14)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#missing-zero-address-validation

Address.verifyCallResult(bool,bytes,string) (node_modules/@openzeppelin/Contracts/Util/Address.sol#201-221) uses assembly
- INLINE ASM (node_modules/@openzeppelin/Contracts/Util/Address.sol#213-214)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#assembly=usage

Different versions of Solidity is used:
- Version used: '0.8.9', '"0.8.0", "'0.8.1", "'0.8.2"
- 0.8.2 (node_modules/@openzeppelin/Contracts/Proxy/Initializable.sol#4)
- 0.8.1 (node_modules/@openzeppelin/Contracts/Proxy/Initializable.sol#4)
- 0.8.0 (node_modules/@openzeppelin/Contracts/Proxy/Initializable.sol#4)
- 0.8.9 (contracts/access/Governable.sol#3)
  - (contracts/interface/IERC20.sol#18)
  - 0.8.9 (contracts/interface/IERC20Metadata.sol#3)
  - 0.8.9 (contracts/interface/IERC20.sol#4)
  - 0.8.9 (contracts/interface/IERC20Metadata.sol#4)
  - 0.8.9 (contracts/interface/IERC20.sol#18)
  - 0.8.9 (contracts/interface/IERC20Metadata.sol#18)
  - 0.8.9 (contracts/interface/IUniswapV3LikePriceProvider.sol#3)
  - 0.8.9 (contracts/interface/IUniswapV3LikePriceProvider.sol#3)
  - 0.8.9 (contracts/interface/IUniswapV3LikePriceProvider.sol#3)
  - 0.8.9 (contracts/interface/IUniswapV3LikePriceProvider.sol#3)
  - 0.8.9 (contracts/interface/IERC20.sol#4)
  - 0.8.9 (contracts/interface/IERC20Metadata.sol#4)
  - 0.8.9 (contracts/library/DataType.sol#3)
  - 0.8.9 (contracts/library/OracleTypes.sol#3)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#different-pragma-directives=are-used

Address.functionCall(address,bytes) (node_modules/@openzeppelin/Contracts/Util/Address.sol#85-87) is never used and should be removed
Address.functionCallWithValue(address,bytes,int256) (node_modules/@openzeppelin/Contracts/Util/Address.sol#89-101) is never used and should be removed
Address.functionCallStatic(address,bytes,int256) (node_modules/@openzeppelin/Contracts/Util/Address.sol#114-120) is never used and should be removed
Address.functionDelegateCall(address,bytes) (node_modules/@openzeppelin/Contracts/Util/Address.sol#133-134) is never used and should be removed
Address.functionDelegateCall(address,bytes,int256) (node_modules/@openzeppelin/Contracts/Util/Address.sol#174-176) is never used and should be removed
Address.functionStaticCall(address,bytes) (node_modules/@openzeppelin/Contracts/Util/Address.sol#194-195) is never used and should be removed
Address.isContract(address) (node_modules/@openzeppelin/Contracts/Util/Address.sol#205-206) is never used and should be removed
Address.sendValue(address,int256) (node_modules/@openzeppelin/Contracts/Util/Address.sol#209-215) is never used and should be removed
Address.transferValue(address,int256) (node_modules/@openzeppelin/Contracts/Util/Address.sol#216-218) is never used and should be removed
Context._msgData() (node_modules/@openzeppelin/Contracts/Util/Context.sol#4-6) is never used and should be removed
Governable._governable_init()(contracts/access/Governable.sol#4-9) is never used and should be removed
Initializable._setInitializationVersion(uint) (node_modules/@openzeppelin/Contracts/Proxy/Initializable.sol#133-140) is never used and should be removed
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#head-code

Pragma version="0.8.2" (node_modules/@openzeppelin/Contracts/Proxy/Initializable.sol#4) allows old versions
Pragma version="0.8.0" (node_modules/@openzeppelin/Contracts/Token/IERC20.sol#4) allows old versions
Pragma version="0.8.0" (node_modules/@openzeppelin/Contracts/Proxy/Initializable.sol#4) allows old versions
Pragma version="0.8.0" (node_modules/@openzeppelin/Contracts/Util/Address.sol#4) allows old versions
Pragma version="0.8.0" (node_modules/@openzeppelin/Contracts/Util/Context.sol#4) allows old versions

```

Pragma version0.8.0 (contracts/access/Governable.sol#0) necessitates a version too recent to be trusted. Consider deploying with 0.6.12/0.7.6/0.8.7

Pragma version0.8.0 (contracts/interfaces/IDownloadable.sol#3) necessitates a version too recent to be trusted. Consider deploying with 0.6.12/0.7.6/0.8.7

Pragma version0.8.0 (contracts/interfaces/core/IChainlinkPriceProvider.sol#3) necessitates a version too recent to be trusted. Consider deploying with 0.6.12/0.7.6/0.8.7

Pragma version0.8.0 (contracts/interfaces/core/IPriceProvider.sol#3) necessitates a version too recent to be trusted. Consider deploying with 0.6.12/0.7.6/0.8.7

Pragma version0.8.0 (contracts/interfaces/periphery/IOracle.sol#3) necessitates a version too recent to be trusted. Consider deploying with 0.6.12/0.7.6/0.8.7

Pragma version0.8.0 (contracts/interfaces/periphery/Storage.sol#3) necessitates a version too recent to be trusted. Consider deploying with 0.6.12/0.7.6/0.8.7

Pragma version0.8.0 (contracts/libraries/DataTypes.sol#3) necessitates a version too recent to be trusted. Consider deploying with 0.6.12/0.7.6/0.8.7

pragma 0.8.9 is not recommended for deployment

Reference: <https://github.com/crytic/slither/wiki/Detector-Documentation#incorrect-versions-of-solidity>

Low level call in Address.sendValue(address,uint24) (node\_modules/@openzeppelin/contracts/utils/Address.sol#6-6);  
 - (success, returnedData) = target.call{value:(amount)}(address,bytes,Uint256,string); (node\_modules/@openzeppelin/contracts/utils/Address.sol#120-139):  
 Low level call in Address.functionCallWithValue(address,bytes,uint256,string) (node\_modules/@openzeppelin/contracts/utils/Address.sol#137-146):  
 - (success, returnedData) = target.staticcall(data)(address,bytes,Uint256,string); (node\_modules/@openzeppelin/contracts/utils/Address.sol#157-160):  
 Low level call in Address.functionCallWithAddress(address,bytes,string) (node\_modules/@openzeppelin/contracts/utils/Address.sol#164-193):  
 - (success, returnedData) = target.staticcall(data)(address,bytes,Uint256,string); (node\_modules/@openzeppelin/contracts/utils/Address.sol#191):  
 Reference: <https://github.com/crytic/slither/wiki/Detector-Documentation#low-level-calls>

Function Governor.\_Governable\_init() (contracts/governor/Governable.sol#18-2) is not in mixedCase  
 Parameter Governor.transferOwnership(address) \_proposedGovernor (contracts/governor/Governable.sol#6) is not in mixedCase  
 Parameter OracleHelpers.scaledDecimal(uint256,uint256,uint256)\_scaledDecimal (contracts/libraries/OracleHelpers.sol#9) is not in mixedCase  
 Parameter OracleHelpers.scaledDecimal(uint256,uint256,uint256)\_scaledDecimal (contracts/libraries/OracleHelpers.sol#10) is not in mixedCase  
 Reference: <https://github.com/crytic/slither/wiki/Detector-Documentation#conformance-to-solidity-naming-conventions>

Variable IPricProvidersAggregator.getPriceProvider(DataTypes.Provider) provider (contracts/interfaces/core/IPricProvidersAggregator.sol#12) is too similar to IPriceProvidersAggregator.setPriceProvider(DataTypes.Provider,IPriceProv  
 osit,priceProvider) (contracts/libraries/Interface.sol#515):  
 Reference: <https://github.com/crytic/slither/wiki/Detector-Documentation#variable-names-are-too-similar>

## SwapperOracle.sol

SwapperOracle.\_quote(DataTypes.Provider,address,address,uint256)\_amountOut (contracts/periphery/SwapperOracle.sol#19) is a local variable never initialized  
 SwapperOracle.\_quote(DataTypes.Provider,address,address,uint256)\_lastUpdatedAt (contracts/periphery/SwapperOracle.sol#20) is a local variable never initialized  
 Reference: <https://github.com/crytic/slither/wiki/Detector-Documentation#unnecessary-local-variables>

SwapperOracle.\_quote(DataTypes.Provider,address,address,uint256) (contracts/periphery/SwapperOracle.sol#112-125) ignores return value by providersAggregator.quote(providerIn\_,tokenInOut\_,amountIn\_) (contracts/periphery/SwapperOracle.sol#118-124):  
 Reference: <https://github.com/crytic/slither/wiki/Detector-Documentation#unneeded-return>

Variable SwapperOracle.\_quote(DataTypes.Provider,address,address,uint256)\_amountOut (contracts/periphery/SwapperOracle.sol#119)' is in SwapperOracle.\_quote(DataTypes.Provider,address,address,uint256) (contracts/periphery/SwapperOracle.sol#112-125) potentially used for declaration: amountOut = amountOut (contracts/periphery/SwapperOracle.sol#9):  
 Variable SwapperOracle.\_quote(DataTypes.Provider,address,address,uint256)\_lastUpdatedAt (contracts/periphery/SwapperOracle.sol#120)' is in SwapperOracle.\_quote(DataTypes.Provider,address,address,uint256) (contracts/periphery/SwapperOracle.sol#112-125) potentially used before declaration: lastUpdatedAt = lastUpdatedAt (contracts/periphery/SwapperOracle.sol#123):  
 Reference: <https://github.com/crytic/slither/wiki/Detector-Documentation#pre-declaration-use-of-local-variables>

SwapperOracle.quote(address,address,uint256) (contracts/periphery/SwapperOracle.sol#62-106) uses timestamp for comparisons  
 - amountOut > 0 || ! priceIsStale(.lastUpdatedAt) (contracts/periphery/SwapperOracle.sol#72)  
 - aPriceOk = amountOut > 0 && priceIsStale(.lastUpdatedAt) (contracts/periphery/SwapperOracle.sol#80)  
 - bPriceOk = amountOut <= 0 && ! priceIsStale(.lastUpdatedAt) (contracts/periphery/SwapperOracle.sol#88)  
 - ! bPriceOk && ! aPriceOk (contracts/periphery/SwapperOracle.sol#95)  
 - require(bool(string)) bPriceOk || bPriceOk.fallbackFailed() (contracts/periphery/SwapperOracle.sol#96):  
 - require(bool(string)) ! bPriceOk || bPriceOk.fallbackFailed() (contracts/periphery/SwapperOracle.sol#98):  
 SwapperOracle.\_priceIsStale(uint256) (contracts/periphery/SwapperOracle.sol#63-105) uses timestamp for comparisons  
 - block.timestamp > timeOfLastUpdate > stalePeriod (contracts/periphery/SwapperOracle.sol#64):  
 Reference: <https://github.com/crytic/slither/wiki/Detector-Documentation#block-timestamp>

Address.verifyCallResult(bool,bytes,string) (node\_modules/@openzeppelin/contracts/utils/Address.sol#201-221) uses assembly  
 - INLINE ASM (node\_modules/@openzeppelin/contracts/utils/Address.sol#219-216):  
 Reference: <https://github.com/crytic/slither/wiki/Detector-Documentation#assembly-use>

Different versions of Solidity is used:  
 Version used: [0.8.0, "0.8.8", "+0.8.8", "+0.8.8"]  
 - 0.8.0 mode modules/@openzeppelin/contracts/token/IERC20/IERC20.sol#4)  
 - 0.8.0 mode modules/@openzeppelin/contracts/token/IERC20/extensions/IERC20Metadata.sol#4)  
 - 0.8.0 mode modules/@openzeppelin/contracts/token/ERC20/ERC20.sol#4)  
 - 0.8.0 mode modules/@openzeppelin/contracts/token/ERC20/extensions/IERC20Metadata.sol#4)  
 - 0.8.9 (contracts/access/Governable.sol#3)  
 - 0.8.9 (contracts/math/Math.sol#3)  
 - 0.8.9 (contracts/interfaces/core/IChainlinkPriceProvider.sol#3)  
 - 0.8.9 (contracts/interfaces/core/IPriceProvider.sol#3)  
 - 0.8.9 (contracts/interfaces/core/IFeeCalculator.sol#3)  
 - 0.8.9 (contracts/interfaces/core/IStorage.sol#3)  
 - 0.8.9 (contracts/libraries/Storage.sol#3)  
 - 0.8.9 (contracts/libraries/DataTypes.sol#3)  
 Reference: <https://github.com/crytic/slither/wiki/Detector-Documentation#different-pragmas-are-used>

Address.functionCall(address,bytes) (node\_modules/@openzeppelin/contracts/utils/Address.sol#89-97) is never used and should be removed  
 Address.functionCall(address,bytes,bytes) (node\_modules/@openzeppelin/contracts/utils/Address.sol#10-101) is never used and should be removed  
 Address.functionCallWithValue(address,bytes,uint256) (node\_modules/@openzeppelin/contracts/utils/Address.sol#114-120) is never used and should be removed  
 Address.functionCallWithValue(address,bytes,uint256,bytes) (node\_modules/@openzeppelin/contracts/utils/Address.sol#155-159) is never used and should be removed  
 Address.functionGetByteCode(address) (node\_modules/@openzeppelin/contracts/utils/Address.sol#163-165) is never used and should be removed  
 Address.functionStaticCall(address,bytes,bytes) (node\_modules/@openzeppelin/contracts/utils/Address.sol#164-193) is never used and should be removed  
 Address.functionStaticCall(address,bytes,bytes,bytes) (node\_modules/@openzeppelin/contracts/utils/Address.sol#164-193) is never used and should be removed  
 Address.isContract(address) (node\_modules/@openzeppelin/contracts/utils/Address.sol#43-42) is never used and should be removed  
 Address.sendValue(address,uint256) (node\_modules/@openzeppelin/contracts/utils/Address.sol#40-45) is never used and should be removed  
 Address.transferOwnership(address) (node\_modules/@openzeppelin/contracts/utils/Address.sol#40-45) is never used and should be removed  
 Context.\_msgData() (node\_modules/@openzeppelin/contracts/context/Context.sol#21-23) is never used and should be removed  
 Governor.\_Governable\_init() (contracts/governor/Governable.sol#4-6) is never used and should be removed  
 InterestRateMath.\_getInterestRate(DataTypes.Address)(address) (node\_modules/@openzeppelin/contracts/math/InterestRateMath.sol#19-21) is never used and should be removed  
 Initializable.\_Initializable\_init() (node\_modules/@openzeppelin/contracts/math/Initializable.sol#138-146) is never used and should be removed  
 Reference: <https://github.com/crytic/slither/wiki/Detector-Documentation#dead-code>

Pragma version0.9.2 (node\_modules/@openzeppelin/contracts/proxy/ERC20/ERC20Burnable.sol#4) allows old versions  
 Pragma version0.9.0 (node\_modules/@openzeppelin/contracts/token/ERC20/IERC20.sol#4) allows old versions  
 Pragma version0.9.0 (node\_modules/@openzeppelin/contracts/token/ERC20/extension/IERC20Metadata.sol#4) allows old versions  
 Pragma version0.8.0 (node\_modules/@openzeppelin/contracts/math/Math.sol#4) allows old versions  
 Pragma version0.8.0 (contracts/access/Governable.sol#3) necessitates a version too recent to be trusted. Consider deploying with 0.6.12/0.7.6/0.8.7  
 Pragma version0.8.0 (contracts/interfaces/core/IChainlinkPriceProvider.sol#3) necessitates a version too recent to be trusted. Consider deploying with 0.6.12/0.7.6/0.8.7  
 Pragma version0.8.0 (contracts/interfaces/core/IPriceProvider.sol#3) necessitates a version too recent to be trusted. Consider deploying with 0.6.12/0.7.6/0.8.7  
 Pragma version0.8.0 (contracts/interfaces/core/IFeeCalculator.sol#3) necessitates a version too recent to be trusted. Consider deploying with 0.6.12/0.7.6/0.8.7  
 Pragma version0.8.0 (contracts/libraries/DataTypes.sol#3) necessitates a version too recent to be trusted. Consider deploying with 0.6.12/0.7.6/0.8.7  
 Pragma version0.8.9 (contracts/periphery/SwapperOracle.sol#3) necessitates a version too recent to be trusted. Consider deploying with 0.6.12/0.7.6/0.8.7  
 Pragma version0.8.9 (contracts/libraries/Storage.sol#3) necessitates a version too recent to be trusted. Consider deploying with 0.6.12/0.7.6/0.8.7  
 Reference: <https://github.com/crytic/slither/wiki/Detector-Documentation#incorrect-versions-of-solidity>

Low level call in Address.sendValue(address,uint256) (node\_modules/@openzeppelin/contracts/utils/Address.sol#6-6):  
 - (success) = recipient.call{value:(amount)}(address,bytes,Uint256,string); (node\_modules/@openzeppelin/contracts/utils/Address.sol#120-139):  
 Low level call in Address.functionCallWithValue(address,bytes,uint256,string) (node\_modules/@openzeppelin/contracts/utils/Address.sol#137-146):  
 - (success, returnedData) = target.staticcall(data)(address,bytes,Uint256,string); (node\_modules/@openzeppelin/contracts/utils/Address.sol#157-160):  
 Low level call in Address.functionCallWithAddress(address,bytes,string) (node\_modules/@openzeppelin/contracts/utils/Address.sol#164-193):  
 - (success, returnedData) = target.staticcall(data)(address,bytes,Uint256,string); (node\_modules/@openzeppelin/contracts/utils/Address.sol#191):  
 Reference: <https://github.com/crytic/slither/wiki/Detector-Documentation#low-level-calls>

Function Governorable.\_Governable\_init() (contracts/governor/Governable.sol#18-2) is not in mixedCase  
 Parameter Governorable.transferOwnership(address) \_proposedGovernor (contracts/governor/Governable.sol#6) is not in mixedCase  
 Reference: <https://github.com/crytic/slither/wiki/Detector-Documentation#conformance-to-solidity-naming-conventions>

Variable IPricProvidersAggregator.getPriceProvider(DataTypes.Provider) provider (contracts/interfaces/core/IPricProvidersAggregator.sol#12):  
 - provider.priceProvider = contractInterface(IPricProvidersAggregator.sol#18-22):  
 Variable SwapperOracle.\_quote(DataTypes.Provider,address,address,uint256)\_amountOut (contracts/periphery/SwapperOracle.sol#19) is too similar to SwapperOracle.\_quote(address,address,uint256)\_amountOut (contracts/periphery/SwapperOracle.sol#18-22):  
 Variable SwapperOracle.\_quote(DataTypes.Provider,address,address,uint256)\_lastUpdatedAt (contracts/periphery/SwapperOracle.sol#20) is too similar to SwapperOracle.\_quote(address,address,uint256)\_lastUpdatedAt (contracts/periphery/SwapperOracle.sol#18-22):  
 Variable SwapperOracle.\_updateFallbackProviders(DataTypes.Provider,fallbackProviderB,(contracts/periphery/SwapperOracle.sol#11)) is too similar to SwapperOracle.updateFallbackProviders(DataTypes.Provider,DataProvider,FallbackProviderB):  
 Variable SwapperOracle.\_updateFallbackProviders(DataTypes.Provider,fallbackProviderF,(contracts/periphery/SwapperOracle.sol#11)) is too similar to SwapperOracle.updateFallbackProviders(DataTypes.Provider,DataProvider,FallbackProviderF):  
 Variable SwapperOracle.\_updateFallbackProviders(DataTypes.Provider,fallbackProviderR,(contracts/periphery/SwapperOracle.sol#11)) is too similar to SwapperOracle.updateFallbackProviders(DataTypes.Provider,DataProvider,FallbackProviderR):  
 Variable SwapperOracle.\_updateFallbackProviders(DataTypes.Provider,fallbackProviderS,(contracts/periphery/SwapperOracle.sol#11)) is too similar to SwapperOracle.updateFallbackProviders(DataTypes.Provider,DataProvider,FallbackProviderS):  
 Reference: <https://github.com/crytic/slither/wiki/Detector-Documentation#variable-names-are-too-similar>

updateFallbackProviders(DataTypes.Provider,DataProvider,FallbackProvider) should be declared external;  
 SwapperOracle.\_updateFallbackProviders(DataTypes.Provider,DataProvider,fallbackProvider) (contracts/periphery/SwapperOracle.sol#131-139):  
 updateFallbackProviders(DataTypes.Provider,DataProvider,fallbackProvider) should be declared external;  
 - SwapperOracle.updateFallbackProvidersIPricProvidersAggregator (contracts/periphery/SwapperOracle.sol#144-148):  
 updateFallbackProvidersIPricProvidersAggregator(DataTypes.Provider,fallbackProvider) should be declared external;  
 - SwapperOracle.updateFallbackProvidersIPricProvidersAggregator (contracts/periphery/SwapperOracle.sol#144-148):  
 Reference: <https://github.com/crytic/slither/wiki/Detector-Documentation#public-function-that-could-be-declared-external>





```

Pragma version0.8.0 (contracts/interfaces/IDeniable.sol#8) necessitates a version too recent to be trusted. Consider deploying with 0.6.12/0.7.6/0.8.7
Pragma version0.8.0 (contracts/interfaces/external/ICurveAddressProvider.sol#3) necessitates a version too recent to be trusted. Consider deploying with 0.6.12/0.7.6/0.8.7
Pragma version0.8.0 (contracts/interfaces/external/ICurveRegistry.sol#3) necessitates a version too recent to be trusted. Consider deploying with 0.6.12/0.7.6/0.8.7
Pragma version0.8.0 (contracts/interfaces/external/ISynthOracle.sol#3) necessitates a version too recent to be trusted. Consider deploying with 0.6.12/0.7.6/0.8.7
solc=0.8.9 is not recommended for deployment
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#incorrect-versions-of-solidity

Low level call in Address.sendValue(address,uint256) (node_modules/Openzeppelin/contracts/utils/Address.sol#60-65):
- (success) = recipient.call.value(amount) (node_modules/Openzeppelin/contracts/utils/Address.sol#60)
- (success,returndata) = target.call.value(amount)(data) (node_modules/Openzeppelin/contracts/utils/Address.sol#128-139)
Low level call in Address.functionStaticCall(address,bytes,string) (node_modules/Openzeppelin/contracts/utils/Address.sol#157-160):
Low level call in Address.functionDelegateCall(address,bytes,string) (node_modules/Openzeppelin/contracts/utils/Address.sol#184-189):
Low level call in Address.functionReturnDataCall(address,bytes,string) (node_modules/Openzeppelin/contracts/utils/Address.sol#184-191):
- (success,returndata) = target.delegatecall(data) (node_modules/Openzeppelin/contracts/utils/Address.sol#191)

Function Governable : Governable_init() (contracts/access/Governable.sol#38-42) is not in mixedCase
Parameter CurvedPriceOracle.getPriceInUSD(ERC20) (contracts/interfaces/external/ICurveAddressProvider.sol#6) is not in mixedCase
Function KovanAddressProvider.getRegistry() (contracts/interfaces/external/ICurveAddressProvider.sol#16) is not in mixedCase
Function KovanRegistry.get_virtual_price_from_ip_token(address) (contracts/interfaces/external/ICurveRegistry.sol#16) is not in mixedCase
Parameter CurvePriceOracle.getPriceInUSD(ERC20) .asset (contracts/interfaces/periphery/synth/CurvePriceOracle.sol#2) is not in mixedCase
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#conformance-to-solidity-naming-conventions

SynthDefaultOracle.sol
SynthDefaultOracle.gp.getPriceInUSD(ERC20) (contracts/periphery/synth/SynthDefaultOracle.sol#85-94) uses timestamp for comparisons
Dangerous comparisons:
- require(bool,string){_userData.statePeriod > block.timestamp - lastUpdatedAt,price-is-stale} (contracts/periphery/synth/SynthDefaultOracle.sol#93)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#block-timestamp

Address.verifyAllResult(bool,bytes,string) (node_modules/Openzeppelin/contracts/utils/Address.sol#201-221) uses assembly
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#assembly-use

Different versions of Solidity is used:
- 0.6.0, "0.8.1", "0.8.2"
- 0.8.2 (node_modules/Openzeppelin/contracts/proxy/UtilityInitializable.sol#4)
- 0.8.0 (node_modules/Openzeppelin/contracts/token/ERC20/ERC20.sol#4)
- 0.8.1 (node_modules/Openzeppelin/contracts/token/ERC20/ERC20Metadata.sol#4)
- 0.8.1 (node_modules/Openzeppelin/contracts/token/ERC20/ERC20.sol#4)
- 0.8.0 (node_modules/Openzeppelin/contracts/token/ERC20/ERC20.sol#4)
- 0.8.0 (node_modules/Openzeppelin/contracts/token/ERC20/ERC20.sol#4)
- 0.8.0 (node_modules/Openzeppelin/contracts/token/ERC20/ERC20.sol#4)
- 0.8.0 (node_modules/Openzeppelin/contracts/token/ERC20/ERC20.sol#4)
- 0.8.9 (contracts/interface/IGovernable.sol#8)
- 0.8.9 (contracts/interface/Core/IChainlinkPriceProvider.sol#8)
- 0.8.9 (contracts/interface/Core/IGovernable.sol#8)
- 0.8.9 (contracts/interface/Periphery/ISynthOracle.sol#8)
- 0.8.8 (contracts/periphery/synth/SynthDefaultOracle.sol#3)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#different-pragma-directives-are-used

Address.functionCall(address,bytes) (node_modules/Openzeppelin/contracts/utils/Address.sol#65-87) is never used and should be removed
Address.functionCallWithValue(address,bytes,string) (node_modules/Openzeppelin/contracts/utils/Address.sol#101-120) is never used and should be removed
Address.functionDelegateCall(address,bytes,uint256) (node_modules/Openzeppelin/contracts/utils/Address.sol#128-139) is never used and should be removed
Address.functionStaticCall(address,bytes,string) (node_modules/Openzeppelin/contracts/utils/Address.sol#176-178) is never used and should be removed
Address.functionReturnDataCall(address,bytes,string) (node_modules/Openzeppelin/contracts/utils/Address.sol#184-191) is never used and should be removed
Address.functionStaticCall(address,bytes) (node_modules/Openzeppelin/contracts/utils/Address.sol#194-195) is never used and should be removed
Address.functionStaticCall(address,bytes,string) (node_modules/Openzeppelin/contracts/utils/Address.sol#197-199) is never used and should be removed
Address.sendValue(address,uint256) (node_modules/Openzeppelin/contracts/utils/Address.sol#40-45) is never used and should be removed
Address.verifyAllResult(bool,bytes,string) (node_modules/Openzeppelin/contracts/utils/Address.sol#201-221) is never used and should be removed
Contract Governor : Governor_init() (contracts/access/Governable.sol#38-42) is never used and should be removed
Governable : Governorable_init() (contracts/access/Governable.sol#129-133) is never used and should be removed
Initializable : _disableInitializers() (node_modules/Openzeppelin/contracts/proxy/Initializable.sol#129-133) is never used and should be removed
Initializable : _setInitializers() (node_modules/Openzeppelin/contracts/proxy/Initializable.sol#133-148) is never used and should be removed
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#functions

Pragma version0.8.2 (node_modules/Openzeppelin/contracts/math/FractionalMath.sol#4) allows old versions
Pragma version0.8.0 (node_modules/Openzeppelin/contracts/token/ERC20/ERC20.sol#4) allows old versions
Pragma version0.8.0 (node_modules/Openzeppelin/contracts/token/ERC20/ERC20Metadata.sol#4) allows old versions
Pragma version0.8.0 (node_modules/Openzeppelin/contracts/token/ERC20/ERC20.sol#4) allows old versions
Pragma version0.8.0 (node_modules/Openzeppelin/contracts/token/ERC20/ERC20Metadata.sol#4) allows old versions
Pragma version0.8.0 (contracts/interfaces/Governable.sol#8) necessitates a version too recent to be trusted. Consider deploying with 0.6.12/0.7.6/0.8.7
Pragma version0.8.0 (contracts/interface/IGovernable.sol#8) necessitates a version too recent to be trusted. Consider deploying with 0.6.12/0.7.6/0.8.7
Pragma version0.8.0 (contracts/interface/Core/IFaceProvider.sol#8) necessitates a version too recent to be trusted. Consider deploying with 0.6.12/0.7.6/0.8.7
Pragma version0.8.0 (contracts/interface/Periphery/ISynthOracle.sol#8) necessitates a version too recent to be trusted. Consider deploying with 0.6.12/0.7.6/0.8.7
solc=0.8.9 is not recommended for deployment
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#incorrect-versions-of-solidity

Function Governor : _Governable_init() (contracts/access/Governable.sol#38-42) is not in mixedCase
Parameter Governor.transferOwnership(address) (contracts/access/Governable.sol#16) is not in mixedCase
Parameter SynthDefaultOracle.addOrUpdateAssetThatUsesChainlink(ERC20,address,uint256).asset (contracts/periphery/synth/SynthDefaultOracle.sol#77) is not in mixedCase
Parameter SynthDefaultOracle.addOrUpdateAssetThatUsesChainlink(ERC20,address,uint256).underlyingAsset (contracts/periphery/synth/SynthDefaultOracle.sol#79) is not in mixedCase
Parameter SynthDefaultOracle.addOrUpdateAssetThatUsesChainlink(ERC20,address,uint256).underlyingAsset._relatedAssets (contracts/periphery/synth/SynthDefaultOracle.sol#79) is not in mixedCase
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#conformance-to-solidity-naming-conventions

```

```

UniswapV3CrossPoolOracle.sol
FullMath.mulDiv(uint256,uint256,uint256) (node_modules/Uniswap/v3-core/contracts/libraries/FullMath.sol#14-106) performs a multiplication on the result of a division:
- denominator = denominator / two (node_modules/Uniswap/v3-core/contracts/libraries/FullMath.sol#67)
FullMath.mulDiv(uint256,uint256,uint256) (node_modules/Uniswap/v3-core/contracts/libraries/FullMath.sol#14-106) performs a multiplication on the result of a division:
- denominator = denominator / two (node_modules/Uniswap/v3-core/contracts/libraries/FullMath.sol#67)
FullMath.mulInv(uint256,uint256,uint256) (node_modules/Uniswap/v3-core/contracts/libraries/FullMath.sol#14-106) performs a multiplication on the result of a division:
- denominator = denominator / two (node_modules/Uniswap/v3-core/contracts/libraries/FullMath.sol#67)
FullMath.inv(uint256) (node_modules/Uniswap/v3-core/contracts/libraries/FullMath.sol#14-106) performs a multiplication on the result of a division:
- denominator = denominator / two (node_modules/Uniswap/v3-core/contracts/libraries/FullMath.sol#67)
FullMath.mulInv(uint256,uint256,uint256) (node_modules/Uniswap/v3-core/contracts/libraries/FullMath.sol#14-106) performs a multiplication on the result of a division:
- denominator = denominator / two (node_modules/Uniswap/v3-core/contracts/libraries/FullMath.sol#67)
FullMath.inv2x2(uint256,uint256,uint256,uint256) (node_modules/Uniswap/v3-core/contracts/libraries/FullMath.sol#14-106) performs a multiplication on the result of a division:
- denominator = denominator / two (node_modules/Uniswap/v3-core/contracts/libraries/FullMath.sol#67)
FullMath.inv2x2(uint256,uint256,uint256,uint256) (node_modules/Uniswap/v3-core/contracts/libraries/FullMath.sol#14-106) performs a multiplication on the result of a division:
- denominator = denominator / two (node_modules/Uniswap/v3-core/contracts/libraries/FullMath.sol#67)
FullMath.mulDiv(uint256,uint256,uint256) (node_modules/Uniswap/v3-core/contracts/libraries/FullMath.sol#14-106) performs a multiplication on the result of a division:
- inv = 1 / denominator = inv (node_modules/Uniswap/v3-core/contracts/libraries/FullMath.sol#99)
FullMath.mulDiv(uint256,uint256,uint256) (node_modules/Uniswap/v3-core/contracts/libraries/FullMath.sol#14-106) performs a multiplication on the result of a division:
- denominator = denominator * two (node_modules/Uniswap/v3-core/contracts/libraries/FullMath.sol#67)
FullMath.mulDiv(uint256,uint256,uint256) (node_modules/Uniswap/v3-core/contracts/libraries/FullMath.sol#14-106) performs a multiplication on the result of a division:
- inv = 2 * denominator = inv (node_modules/Uniswap/v3-core/contracts/libraries/FullMath.sol#99)
FullMath.mulDiv(uint256,uint256,uint256) (node_modules/Uniswap/v3-core/contracts/libraries/FullMath.sol#14-106) performs a multiplication on the result of a division:
- prod = prod * two (node_modules/Uniswap/v3-core/contracts/libraries/FullMath.sol#72)
- result = prod * inv (node_modules/Uniswap/v3-core/contracts/libraries/FullMath.sol#104)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#divide-before-multiply

OracleLibrary.getWeightedArithmeticalMeanTick(oracleLibrary,WeightedTickData).l (node_modules/Uniswap/v3-periphery/contracts/libraries/OracleLibrary.sol#152) is a local variable never initialized
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#uninitialized-local-variables

UniswapV3CrossPoolOracle.constructor(address,nativeToken) (contracts/utils/UniswapV3CrossPoolOracle.sol#18) lacks a zero-check on :
- nativeToken = nativeToken (contracts/utils/UniswapV3CrossPoolOracle.sol#18)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#assess-zero-address-validation

OracleLibrary.getLiquidityTimestamp(address) (node_modules/Uniswap/v3-periphery/contracts/libraries/OracleLibrary.sol#93-126) uses timestamp for comparisons
OracleLibrary.getLiquidityTimestampAndLiquidity(address) (node_modules/Uniswap/v3-periphery/contracts/libraries/OracleLibrary.sol#93-126) uses timestamp for comparisons
OracleLibrary.getObservationTimestamp(uint32,uint32) (node_modules/Uniswap/v3-periphery/contracts/libraries/OracleLibrary.sol#104)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#block-timestamp

FullMath.mulDiv(uint256,uint256,uint256) (node_modules/Uniswap/v3-core/contracts/libraries/FullMath.sol#14-106) uses assembly
- INLINE ASM (node_modules/Uniswap/v3-core/contracts/libraries/FullMath.sol#26-30)
- INLINE ASM (node_modules/Uniswap/v3-core/contracts/libraries/FullMath.sol#52-54)
- INLINE ASM (node_modules/Uniswap/v3-core/contracts/libraries/FullMath.sol#56-59)
- INLINE ASM (node_modules/Uniswap/v3-core/contracts/libraries/FullMath.sol#61-63)
- INLINE ASM (node_modules/Uniswap/v3-core/contracts/libraries/FullMath.sol#71-73)
- INLINE ASM (node_modules/Uniswap/v3-core/contracts/libraries/FullMath.sol#77-79)

```

## StableCoinProvider.sol

```

StableCoinProvider.pricesInUSD(fixed) (contracts/core/StableCoinProvider.sol#57-70) uses timestamp for comparisons
  - <= priceSale.lastUpdated() && !isDeviationOk(_priceInUSD) (contracts/core/StableCoinProvider.sol#61)
    => require(priceSale.lastUpdated() <= priceSale.lastUpdated() && !isDeviationOk(_priceInUSD), "stable-prices-invalid") (contracts/core/StableCoinProvider.sol#67)

UsingSafeMathForIntegers(100) (contracts/math/SafeMathForIntegers.sol#47-54) uses timestamp for comparisons
  - dangerous comparison: block.timestamp - timerLastEpoch >= timerPeriod (contracts/math/SafeMathForIntegers.sol#46)
    => require(block.timestamp - timerLastEpoch >= timerPeriod, "block-timestamp")

Address.verifyCallResult(bool, bytes, string) (code/modules/Openseppelin/contracts/Address.sol#201-221) uses assembly
  - AND ADD (code/modules/Openseppelin/contracts/Address.sol#213-214)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#assembly-usage

Different versions of Solidity is used:
  - Version used: ['0.8.9', '>0.6.2', '<=0.8.0', '>0.8.1', '>0.8.2']
  - 0.8.2 (code/modules/Openseppelin/contracts/proxy/utils/Initializable.sol#4)
    => require(!_initializable, "Initializable.sol: _initializable is not set") (code/modules/Openseppelin/contracts/proxy/utils/Initializable.sol#10)
  - 0.8.0 (code/modules/Openseppelin/contracts/token/ERC20/extensions/IERC20Metadata.sol#4)
    => 0.8.1 (code/modules/Openseppelin/contracts/token/ERC20/IERC20.sol#4)
  - 0.8.1 (code/modules/Openseppelin/contracts/token/ERC20/IERC20.sol#10)
    => 0.8.0 (code/modules/Openseppelin/contracts/token/ERC20/IERC20.sol#14)
  - 0.8.5 (code/modules/Openseppelin/contracts/token/ERC20/IERC20.sol#16)
    => 0.8.6 (code/modules/Openseppelin/contracts/token/ERC20/IERC20.sol#17)
  - 0.8.6 (code/modules/Openseppelin/contracts/token/ERC20/IERC20.sol#18)
    => 0.8.7 (code/modules/Openseppelin/contracts/token/ERC20/IERC20.sol#19)
  - 0.8.7 (code/modules/Openseppelin/contracts/token/ERC20/IERC20.sol#20)
    => 0.8.8 (code/modules/Openseppelin/contracts/token/ERC20/IERC20.sol#21)
  - 0.8.8 (code/modules/Openseppelin/contracts/token/ERC20/IERC20.sol#22)
    => 0.8.9 (code/modules/Openseppelin/contracts/token/ERC20/IERC20.sol#23)

Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#different-pragma-directives-are-used

Address.functionCall(address, bytes) (code/modules/Openseppelin/contracts/utils/Address.sol#85-97) is never used and should be removed
Address.functionCall(address, bytes, uint32) (code/modules/Openseppelin/contracts/utils/Address.sol#101-103) is never used and should be removed
Address.functionCallWithValue(address, bytes, uint32, string) (code/modules/Openseppelin/contracts/utils/Address.sol#128-139) is never used and should be removed
Address.functionDelegateCall(address, bytes) (code/modules/Openseppelin/contracts/utils/Address.sol#174-176) is never used and should be removed
Address.functionStaticCall(address, bytes) (code/modules/Openseppelin/contracts/utils/Address.sol#187-191) is never used and should be removed
Address.functionStaticCall(address, bytes, string) (code/modules/Openseppelin/contracts/utils/Address.sol#157-160) is never used and should be removed
Address.functionStaticCall(address, bytes, string, uint32) (code/modules/Openseppelin/contracts/utils/Address.sol#161-164) is never used and should be removed
Address.sendValue(address, uint256) (code/modules/Openseppelin/contracts/utils/Address.sol#40-45) is never used and should be removed
Address.verifyCallResult(bool, bytes, string) (code/modules/Openseppelin/contracts/Address.sol#201-221) is never used and should be removed
Contract.requiresInitializable(Initializable, address) (code/modules/Openseppelin/contracts/proxy/utils/Initializable.sol#18-20) is never used and should be removed
Initializable.disableInitializable(Initializable) (code/modules/Openseppelin/contracts/proxy/utils/Initializable.sol#129-131) is never used and should be removed
Initializable._initializeInitializableVersion(uint32) (code/modules/Openseppelin/contracts/proxy/utils/Initializable.sol#133-148) is never used and should be removed
SafeMathForIntegers._addSafeIntegers(int128, int128) (code/modules/Openseppelin/contracts/math/SafeMathForIntegers.sol#10-14) is never used and should be removed

Pragma version 0.8.2 (code/modules/Openseppelin/contracts/token/IERC20.sol#4) allows old versions
Pragma version 0.8.0 (code/modules/Openseppelin/contracts/token/IERC20.sol#14) allows old versions
Pragma version 0.8.1 (code/modules/Openseppelin/contracts/token/IERC20.sol#10) allows old versions
Pragma version 0.8.5 (code/modules/Openseppelin/contracts/token/IERC20.sol#16) necessitates a version too recent to be trusted. Consider deploying with 0.6.12/0.7.6/0.8.7
Pragma version 0.8.0 (code/modules/Openseppelin/contracts/token/IERC20.sol#14) necessitates a version too recent to be trusted. Consider deploying with 0.6.12/0.7.6/0.8.7
Pragma version 0.8.9 (code/features/UsingProviderAggregator.sol#3) necessitates a version too recent to be trusted. Consider deploying with 0.6.12/0.7.6/0.8.7
Pragma version 0.8.9 (code/features/UsingSafePeriod.sol#3) necessitates a version too recent to be trusted. Consider deploying with 0.6.12/0.7.6/0.8.7
Pragma version 0.8.9 (code/interfaces/core/IForceProvider.sol#3) necessitates a version too recent to be trusted. Consider deploying with 0.6.12/0.7.6/0.8.7
Pragma version 0.8.9 (code/interfaces/core/IForceProvidersAggregator.sol#3) necessitates a version too recent to be trusted. Consider deploying with 0.6.12/0.7.6/0.8.7

```

Pragma version0.8.0 (contracts/interfaces/core/IStableCoinProvider.sol#4) necessitates a version too recent to be trusted. Consider deploying with 0.6.12/0.7.6/0.8.7  
 Pragma version0.8.9 (contracts/libraries/DataTypes.sol#3) necessitates a version too recent to be trusted. Consider deploying with 0.6.12/0.7.6/0.8.7  
 pragma0.8.9 is not recommended for deployment  
 Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#incorrect-versions-of-solidity

Low level call in Address.sendValue(address,uint256) (node\_modules/@openzeppelin/contracts/utils/Address.sol#G-65):  
  - (success) = recipient.call.value(amount) (node\_modules/@openzeppelin/contracts/utils/Address.sol#43)  
  - low level call in Address.functionCallWithValue(address,bytes,uint256,string) (node\_modules/@openzeppelin/contracts/utils/Address.sol#128-139):  
  - (success,returnData) = target.call.value(value)(data) (node\_modules/@openzeppelin/contracts/utils/Address.sol#137)  
  - low level call in Address.functionDelegateCall(address,bytes,bytes) (node\_modules/@openzeppelin/contracts/utils/Address.sol#157-166):  
  - (success,returnData) = target.staticcall(data) (node\_modules/@openzeppelin/contracts/utils/Address.sol#164)  
  - low level call in Address.functionDelegateCall(address,bytes,bytes) (node\_modules/@openzeppelin/contracts/utils/Address.sol#184-193):  
  - (success,returnData) = target.delegatecall(data) (node\_modules/@openzeppelin/contracts/utils/Address.sol#191)

Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#low-level-calls

Function Governable.\_Governable\_init() (contracts/access/Governable.sol#19-23) is not in mixedCase  
  Parameter Governable.transferOwnership(address) \_proposedGovernor (contracts/access/Governable.sol#6) is not in mixedCase  
  Variable StableCoinProvider.\_primaryStablecoin (contracts/core/IStableCoinProvider.sol#27) is not in mixedCase  
  Variable OracleHelpers.\_scaledDecimal (uint128,uint128,uint256) fromDecimal (contracts/libraries/OracleHelpers.sol#8) is not in mixedCase  
  Parameter OracleHelpers.scaledDecimal (uint128,uint128,uint256).toDecimal (contracts/libraries/OracleHelpers.sol#9) is not in mixedCase  
  Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#conformance-to-solidity-naming-conventions

Variable IPriceProvidersAggregator.priceProviders(DataTypes.Provider).priceProvider (contracts/interfaces/core/IPriceProvidersAggregator.sol#36) is too similar to IPriceProvidersAggregator.setPriceProvider(DataTypes.Provider,IPriceProvider).priceProvider (contracts/interfaces/core/IPriceProvidersAggregator.sol#107)  
  Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#variable-names-are-too-similar

**UmbrellaPassportPriceProvider.sol**

UmbrellaPriceProvider.\_toKey(Bytes).i (contracts/core/UmbrellaPriceProvider.sol#64) is a local variable never initialized  
  Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#uninitialized-local-variables

Address.verifyCallResult(bool,Bytes,Bytes) (node\_modules/@openzeppelin/contracts/utils/Address.sol#201-221) uses assembly  
  - INLINE ASM (contract/dependencies/#BmB-network/lib/ValueDecoder.sol#17-18) uses assembly

ValueDecoder.toUnit(bytes) (contract/dependencies/#BmB-network/lib/ValueDecoder.sol#17-19) uses assembly  
  - INLINE ASM (contract/dependencies/#BmB-network/lib/ValueDecoder.sol#17-18) uses assembly

ValueDecoder.toInt(uint256) (contract/dependencies/#BmB-network/lib/ValueDecoder.sol#13-15) uses assembly  
  - INLINE ASM (contract/dependencies/#BmB-network/lib/ValueDecoder.sol#11-13) uses assembly

ValueDecoder.toInt(uint256) (contract/dependencies/#BmB-network/lib/ValueDecoder.sol#11-13) uses assembly  
  - INLINE ASM (contract/dependencies/#BmB-network/lib/ValueDecoder.sol#9-11) uses assembly

Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#assembly-use

Different versions of Solidity is used:  
  - Version used: ['0.8.9', '>=0.6.8', '<=0.8.0', '\*\*0.8.1', '<>0.8.2']  
  - 0.8.0 (node\_modules/@openzeppelin/contracts/token/ERC20/IERC20.sol#4)  
  - 0.8.0 (node\_modules/@openzeppelin/contracts/token/ERC20/extensions/IERC20Metadata.sol#4)  
  - 0.8.0 (node\_modules/@openzeppelin/contracts/math/Math.sol#4)  
  - 0.8.0 (node\_modules/@openzeppelin/contracts/math/SafeCast.sol#16)  
  - 0.8.0 (node\_modules/@openzeppelin/contracts/math/SafeCast.sol#16-17)  
  - v2 (node\_modules/#BmB-network/toolbox/dist/contracts/IChain.sol#2)  
  - 0.8.0 (node\_modules/#BmB-network/toolbox/dist/contracts/IRegistry.sol#2)  
  - 0.8.0 (node\_modules/#BmB-network/toolbox/dist/contracts/ISafeCast.sol#1)  
  - 0.8.0 (contract/core/IPriceProvider.sol#1)  
  - 0.8.0 (contract/core/UmbrellaPassportPriceProvider.sol#3)  
  - 0.8.0 (contract/dependencies/#BmB-network/interfaces/IDatumReceiver.sol#3)  
  - 0.8.0 (contract/dependencies/#BmB-network/lib/PassportStructs.sol#3)  
  - 0.8.0 (contract/dependencies/#BmB-network/lib/ValueDecoder.sol#3)  
  - 0.8.0 (contract/interfaces/IGovernable.sol#3)  
  - 0.8.0 (contract/interfaces/core/IPriceProvider.sol#3)  
  - 0.8.0 (contract/interfaces/core/IUmberllaPriceProvider.sol#3)  
  - 0.8.0 (contract/interfaces/core/IUmberllaPriceProvider.sol#3)  
  Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#different-pragma-directives-are-used

Address.functionCall(address,bytes) (node\_modules/@openzeppelin/contracts/utils/Address.sol#95-101) is never used and should be removed  
  Address.functionCall(address,bytes,bytes) (node\_modules/@openzeppelin/contracts/utils/Address.sol#95-101) is never used and should be removed  
  Address.functionCallWithValue(address,bytes,bytes,uint256,string) (node\_modules/@openzeppelin/contracts/utils/Address.sol#128-139) is never used and should be removed  
  Address.functionDelegateCall(address,bytes) (node\_modules/@openzeppelin/contracts/utils/Address.sol#174-176) is never used and should be removed  
  Address.functionDelegateCall(address,bytes,bytes) (node\_modules/@openzeppelin/contracts/utils/Address.sol#174-176) is never used and should be removed  
  Address.functionStaticCall(address,bytes,string) (node\_modules/@openzeppelin/contracts/utils/Address.sol#157-166) is never used and should be removed  
  Address.isContract(address) (node\_modules/@openzeppelin/contracts/utils/Address.sol#42) is never used and should be removed  
  Address.verifyCallResult(bool,bytes) (node\_modules/@openzeppelin/contracts/utils/Address.sol#201-221) is never used and should be removed  
  Address.verifyCallResult(bool,bytes,Bytes) (node\_modules/@openzeppelin/contracts/utils/Address.sol#201-221) is never used and should be removed  
  Context.\_msgData() (node\_modules/@openzeppelin/contracts/proxy/Initializable.sol#23) is never used and should be removed  
  Context.\_msgData() (node\_modules/@openzeppelin/contracts/proxy/Initializable.sol#129-131) is never used and should be removed  
  Initializable.\_disableInitializers() (node\_modules/@openzeppelin/contracts/proxy/Initializable.sol#133-146) is never used and should be removed  
  Math.\_ceilDiv(uint256,uint256) (node\_modules/@openzeppelin/contracts/math/Math.sol#42) is never used and should be removed  
  Math.\_max(uint256,uint256) (node\_modules/@openzeppelin/contracts/math/Math.sol#10-13) is never used and should be removed  
  SafeCast.\_ceilDiv(uint256,uint256) (node\_modules/@openzeppelin/contracts/math/SafeCast.sol#206-209) is never used and should be removed  
  SafeCast.\_ceilDiv(uint256) (node\_modules/@openzeppelin/contracts/math/SafeCast.sol#236-240) is never used and should be removed  
  SafeCast.\_ceilDiv(uint256) (node\_modules/@openzeppelin/contracts/math/SafeCast.sol#270-273) is never used and should be removed  
  SafeCast.\_ceilDiv(uint256) (node\_modules/@openzeppelin/contracts/math/SafeCast.sol#270-273) is never used and should be removed  
  SafeCast.\_ceilDiv(int256) (node\_modules/@openzeppelin/contracts/math/SafeCast.sol#228-227) is never used and should be removed  
  SafeCast.\_ceilInt128(uint256) (node\_modules/@openzeppelin/contracts/math/SafeCast.sol#47-50) is never used and should be removed  
  SafeCast.\_ceilInt256(uint256) (node\_modules/@openzeppelin/contracts/math/SafeCast.sol#134-137) is never used and should be removed  
  SafeCast.\_ceilInt32(uint256) (node\_modules/@openzeppelin/contracts/math/SafeCast.sol#192-195) is never used and should be removed  
  SafeCast.\_ceilInt64(uint256) (node\_modules/@openzeppelin/contracts/math/SafeCast.sol#196-199) is never used and should be removed  
  SafeCast.\_ceilInt96(uint256) (node\_modules/@openzeppelin/contracts/math/SafeCast.sol#132-135) is never used and should be removed  
  SafeCast.\_ceilInt96(uint256) (node\_modules/@openzeppelin/contracts/math/SafeCast.sol#62-65) is never used and should be removed  
  ValueDecoder.toInt(bytes) (contract/dependencies/#BmB-network/lib/ValueDecoder.sol#3) is never used and should be removed  
  ValueDecoder.toInt(bytes) (contract/dependencies/#BmB-network/lib/ValueDecoder.sol#10) is never used and should be removed  
  Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#dead-code

Pragma version0.8.0 (node\_modules/@openzeppelin/contracts/proxy/Initializable.sol#4) allows old versions  
  Pragma version0.8.0 (node\_modules/@openzeppelin/contracts/token/ERC20/IERC20.sol#4) allows old versions  
  Pragma version0.8.0 (node\_modules/@openzeppelin/contracts/token/ERC20/extensions/IERC20Metadata.sol#4) allows old versions  
  Pragma version0.8.0 (node\_modules/@openzeppelin/contracts/math/Math.sol#4) allows old versions  
  Pragma version0.8.0 (node\_modules/#BmB-network/toolbox/dist/contracts/IChain.sol#2) allows old versions  
  Pragma version0.8.0 (node\_modules/#BmB-network/toolbox/dist/contracts/IRegistry.sol#2) allows old versions  
  Pragma version0.8.0 (contract/core/IPriceProvider.sol#3) necessitates a version too recent to be trusted. Consider deploying with 0.6.12/0.7.6/0.8.7  
  Pragma version0.8.0 (contract/core/UmbrellaPassportPriceProvider.sol#3) necessitates a version too recent to be trusted. Consider deploying with 0.6.12/0.7.6/0.8.7  
  Pragma version0.8.0 (contract/dependencies/#BmB-network/interfaces/IDatumReceiver.sol#3) necessitates a version too recent to be trusted. Consider deploying with 0.6.12/0.7.6/0.8.7  
  Pragma version0.8.0 (contract/dependencies/#BmB-network/lib/PassportStructs.sol#3) necessitates a version too recent to be trusted. Consider deploying with 0.6.12/0.7.6/0.8.7  
  Pragma version0.8.0 (contract/interfaces/IGovernable.sol#3) necessitates a version too recent to be trusted. Consider deploying with 0.6.12/0.7.6/0.8.7  
  Pragma version0.8.0 (contract/interfaces/core/IPriceProvider.sol#3) necessitates a version too recent to be trusted. Consider deploying with 0.6.12/0.7.6/0.8.7  
  Pragma version0.8.0 (contract/interfaces/core/IUmberllaPriceProvider.sol#3) necessitates a version too recent to be trusted. Consider deploying with 0.6.12/0.7.6/0.8.7  
  pragma0.8.0 is not recommended for deployment  
  Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#incorrect-versions-of-solidity

Low level call in Address.sendValue(address,uint256) (node\_modules/@openzeppelin/contracts/utils/Address.sol#65):  
  - (success) = recipient.call.value(amount) (node\_modules/@openzeppelin/contracts/utils/Address.sol#43)  
  - low level call in Address.functionCallWithValue(address,bytes,uint256,string) (node\_modules/@openzeppelin/contracts/utils/Address.sol#128-139):  
  - (success,returnData) = target.call.value(value)(data) (node\_modules/@openzeppelin/contracts/utils/Address.sol#137)  
  - low level call in Address.functionStaticCall(address,bytes,string) (node\_modules/@openzeppelin/contracts/utils/Address.sol#157-166):  
  - (success,returnData) = target.staticcall(data) (node\_modules/@openzeppelin/contracts/utils/Address.sol#164)  
  - low level call in Address.functionDelegateCall(address,bytes,bytes) (node\_modules/@openzeppelin/contracts/utils/Address.sol#184-193):  
  - (success,returnData) = target.delegatecall(data) (node\_modules/@openzeppelin/contracts/utils/Address.sol#191)

Function Governable.\_Governable\_init() (contracts/access/Governable.sol#18-22) is not in mixedCase  
  Parameter Governable.transferOwnership(address) \_proposedGovernor (contracts/access/Governable.sol#6) is not in mixedCase  
  Parameter ValueDecoder.toUnit(Bytes).bytes (contract/dependencies/#BmB-network/lib/ValueDecoder.sol#6) is not in mixedCase  
  Parameter ValueDecoder.toUnit(Bytes).bytes (contract/dependencies/#BmB-network/lib/ValueDecoder.sol#12) is not in mixedCase  
  Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#conformance-to-solidity-naming-conventions

**UmbrellaPriceProvider.sol**

UmbrellaPriceProvider.\_toKey(Bytes).i (contracts/core/UmbrellaPriceProvider.sol#64) is a local variable never initialized  
  Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#uninitialized-local-variables

Address.verifyCallResult(bool,Bytes,Bytes) (node\_modules/@openzeppelin/contracts/utils/Address.sol#201-211) uses assembly  
  - INLINE ASM (node\_modules/@openzeppelin/contracts/utils/Address.sol#210-211) uses assembly

Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#assembly-use

Different versions of Solidity is used:  
  - Version used: ['0.8.9', '>=0.6.8', '<=0.8.0', '\*\*0.8.1', '<>0.8.2']



```

Pragma version='0.8.2' (node_modules/@openzeppelin/contracts/proxy/utils/Initializable.sol#4) allows old versions
Pragma version='0.8.1' (node_modules/@openzeppelin/contracts/utils/Address.sol#4) allows old versions
Pragma version='0.8.0' (node_modules/@openzeppelin/contracts/Context.sol#4) allows old versions
Pragma version=0.8.5 (contracts/access/Governable.sol#3) necessitates a version too recent to be trusted. Consider deploying with 0.6.12/0.7.6/0.8.7
Pragma version=0.8.4 (contracts/access/Governable.sol#3) necessitates a version too recent to be trusted. Consider deploying with 0.6.12/0.7.6/0.8.7
Pragma version=0.8.3 (contracts/interface/IDeveloper.sol#3) necessitates a version too recent to be trusted. Consider deploying with 0.6.12/0.7.6/0.8.7
Pragma version=0.8.2 (contracts/interface/IForceProvider.sol#3) necessitates a version too recent to be trusted. Consider deploying with 0.6.12/0.7.6/0.8.7
Pragma version=0.8.1 (contracts/libraries/DataTypes.sol#3) necessitates a version too recent to be trusted. Consider deploying with 0.6.12/0.7.6/0.8.7
solc=0.8.5 is not recommended for deployment
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#incorrect-versions-of-solidity

Low level call in Address.sendValue(address,uint256) (node_modules/@openzeppelin/contracts/Address.sol#65):
  - (success,returndata) = target.call.value(value)(data) (node_modules/@openzeppelin/contracts/Address.sol#128-139)
Low level call in Address.functionCallWithValue(address,bytes,uint256,string) (node_modules/@openzeppelin/contracts/Address.sol#128-139):
  - (success,returndata) = target.callWithValue(value)(data) (node_modules/@openzeppelin/contracts/Address.sol#137)
Low level call in Address.functionDelegateCall(address,bytes,string) (node_modules/@openzeppelin/contracts/Address.sol#128-139):
  - (success,returndata) = target.delegatecall(data) (node_modules/@openzeppelin/contracts/Address.sol#157-166)
Low level call in Address.functionStaticCall(address,bytes,string) (node_modules/@openzeppelin/contracts/Address.sol#128-139):
  - (success,returndata) = target.staticcall(data) (node_modules/@openzeppelin/contracts/Address.sol#157-166)
Function Governance_Governable.init() (contracts/access/Governable.sol#4-2) is not in mixedCase
Parameter Governor.transferGovernorship(address) _proposedGovernor (contracts/access/Governable.sol#6) is not in mixedCase
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#conformance-to-solidity-naming-conventions

Variable IPriceProvidersAggregator.priceProviders(IForceProvider.priceProviders,contracts/interfaces/core/IPriceProvidersAggregator.sol#36) is too similar to IPriceProvidersAggregator.setPriceProvider(DataTypes.Providers.IPriceProv
der.priceProvider, (contracts/interface/core/IPriceProvidersAggregator.sol#10))
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#variable-names-are-too-similar

```

## UsingStableCoinProvider.sol

```

Address.verifyCallResult(bool,bytes,string) (node_modules/@openzeppelin/contracts/utils/Address.sol#201-221) uses assembly
  - INLINE ASM (node_modules/@openzeppelin/contracts/utils/Address.sol#213-216)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#assembly-usage

Different versions of Solidity is used:
  - Version used: '0.8.0', '0.8.1', '0.8.2'
  - 0.8.2 (node_modules/@openzeppelin/contract/proxy/utils/Initializable.sol#4)
  - 0.8.1 (node_modules/@openzeppelin/contract/utils/Address.sol#4)
  - 0.8.0 (node_modules/@openzeppelin/contract/Context.sol#4)
  - 0.8.5 (contracts/access/Governable.sol#3)
  - 0.8.9 (contracts/features/UsingStableCoinProvider.sol#3)
  - 0.8.8 (contracts/features/UsingStableCoinProvider.sol#3)
  - 0.8.9 (contracts/interface/IStableCoinProvider.sol#3)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#different-pragma-directives-are-used

Address.functionCall(addresses,bytes) (node_modules/@openzeppelin/contracts/utils/Address.sol#85-97) is never used and should be removed
Address.functionCall(address,bytes,string) (node_modules/@openzeppelin/contracts/utils/Address.sol#95-101) is never used and should be removed
Address.functionCallWithValue(address,bytes,uint256) (node_modules/@openzeppelin/contracts/utils/Address.sol#128-139) is never used and should be removed
Address.functionDelegateCall(address,bytes) (node_modules/@openzeppelin/contracts/utils/Address.sol#174-176) is never used and should be removed
Address.functionDelegateCallWithValue(address,bytes,uint256,string) (node_modules/@openzeppelin/contracts/utils/Address.sol#180-193) is never used and should be removed
Address.functionStaticCall(address,bytes,string) (node_modules/@openzeppelin/contracts/utils/Address.sol#128-139) is never used and should be removed
Address.functionStaticCallWithValue(address,bytes,uint256) (node_modules/@openzeppelin/contracts/utils/Address.sol#157-166) is never used and should be removed
Address.functionStaticCall(address,bytes,string) (node_modules/@openzeppelin/contracts/utils/Address.sol#180-193) is never used and should be removed
Address.functionStaticCallWithValue(address,bytes) (node_modules/@openzeppelin/contracts/utils/Address.sol#157-166) is never used and should be removed
Address.functionStaticCall(address,bytes) (node_modules/@openzeppelin/contracts/utils/Address.sol#180-193) is never used and should be removed
Address.functionStaticCallWithValue(address,bytes,uint256) (node_modules/@openzeppelin/contracts/utils/Address.sol#201-221) is never used and should be removed
Address.functionStaticCallWithValue(address,bytes) (node_modules/@openzeppelin/contracts/utils/Address.sol#201-221) is never used and should be removed
Context._msgData() (node_modules/@openzeppelin/contracts/utils/Address.sol#213-216) is never used and should be removed
Governance_Governable.Governable() (contracts/access/Governable.sol#4-2) is not in mixedCase
Initializable_disableInitializers() (node_modules/@openzeppelin/contracts/proxy/utils/Initializable.sol#129-131) is never used and should be removed
Initializable_setInitializableVersion(uint32) (node_modules/@openzeppelin/contracts/proxy/utils/Initializable.sol#133-148) is never used and should be removed
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#read-code

Pragma version='0.8.2' (node_modules/@openzeppelin/contracts/proxy/utils/Initializable.sol#4) allows old versions
Pragma version='0.8.1' (node_modules/@openzeppelin/contracts/proxy/utils/Initializable.sol#4) allows old versions
Pragma version='0.8.0' (node_modules/@openzeppelin/contracts/proxy/utils/Initializable.sol#4) allows old versions
Pragma version=0.8.5 (contracts/access/Governable.sol#3) necessitates a version too recent to be trusted. Consider deploying with 0.6.12/0.7.6/0.8.7
Pragma version=0.8.4 (contracts/access/Governable.sol#3) necessitates a version too recent to be trusted. Consider deploying with 0.6.12/0.7.6/0.8.7
Pragma version=0.8.3 (contracts/interface/IDeveloper.sol#3) necessitates a version too recent to be trusted. Consider deploying with 0.6.12/0.7.6/0.8.7
Pragma version=0.8.2 (contracts/interface/IForceProvider.sol#3) necessitates a version too recent to be trusted. Consider deploying with 0.6.12/0.7.6/0.8.7
Pragma version=0.8.1 (contracts/libraries/DataTypes.sol#3) necessitates a version too recent to be trusted. Consider deploying with 0.6.12/0.7.6/0.8.7
solc=0.8.5 is not recommended for deployment
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#incorrect-versions-of-solidity

Low level call in Address.sendValue(address,uint256) (node_modules/@openzeppelin/contracts/Address.sol#65):
  - (success) = recipient.call.value(amount) (node_modules/@openzeppelin/contracts/Address.sol#128-139)
Low level call in Address.functionCallWithValue(address,bytes,uint256,string) (node_modules/@openzeppelin/contracts/Address.sol#128-139):
  - (success,returndata) = target.callWithValue(value)(data) (node_modules/@openzeppelin/contracts/Address.sol#137)
Low level call in Address.functionDelegateCall(address,bytes) (node_modules/@openzeppelin/contracts/Address.sol#128-139):
  - (success,returndata) = target.delegatecall(data) (node_modules/@openzeppelin/contracts/Address.sol#157-166)
Low level call in Address.functionStaticCall(address,bytes,string) (node_modules/@openzeppelin/contracts/Address.sol#128-139):
  - (success,returndata) = target.staticcall(data) (node_modules/@openzeppelin/contracts/Address.sol#157-166)
Function Governance_Governable.init() (contracts/access/Governable.sol#4-2) is not in mixedCase
Parameter Governor.transferGovernorship(address) _proposedGovernor (contracts/access/Governable.sol#6) is not in mixedCase
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#conformance-to-solidity-naming-conventions


```

## UsingStalePeriod.sol

```

UsingStalePeriod._priceSetSale(uint256,uint256) (contracts/features/UsingStalePeriod.sol#45-47) uses timestamp for comparisons
  Dangerous comparison: UsingStalePeriod > stalePeriod (contracts/features/UsingStalePeriod.sol#46)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#block-timestamp

Address.verifyCallResult(bool,bytes,string) (node_modules/@openzeppelin/contracts/utils/Address.sol#201-221) uses assembly
  - INLINE ASM (node_modules/@openzeppelin/contracts/utils/Address.sol#213-216)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#assembly-usage

Different versions of Solidity is used:
  - Version used: '0.8.9', '0.8.1', '0.8.2'
  - 0.8.2 (node_modules/@openzeppelin/contract/proxy/utils/Initializable.sol#4)
  - 0.8.1 (node_modules/@openzeppelin/contract/utils/Address.sol#4)
  - 0.8.0 (node_modules/@openzeppelin/contract/Context.sol#4)
  - 0.8.9 (contracts/access/Governable.sol#3)
  - 0.8.8 (contracts/features/UsingStalePeriod.sol#3)
  - 0.8.9 (contracts/interface/Governable.sol#3)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#different-pragma-directives-are-used

Address.functionCall(address,bytes) (node_modules/@openzeppelin/contracts/utils/Address.sol#85-97) is never used and should be removed
Address.functionCall(address,bytes,string) (node_modules/@openzeppelin/contracts/utils/Address.sol#95-101) is never used and should be removed
Address.functionCallWithValue(address,bytes,uint256) (node_modules/@openzeppelin/contracts/utils/Address.sol#128-139) is never used and should be removed
Address.functionDelegateCall(address,bytes) (node_modules/@openzeppelin/contracts/utils/Address.sol#174-176) is never used and should be removed
Address.functionDelegateCallWithValue(address,bytes,uint256) (node_modules/@openzeppelin/contracts/utils/Address.sol#180-193) is never used and should be removed
Address.functionStaticCall(address,bytes,bytes) (node_modules/@openzeppelin/contracts/utils/Address.sol#128-139) is never used and should be removed
Address.functionStaticCallWithValue(address,bytes,bytes) (node_modules/@openzeppelin/contracts/utils/Address.sol#157-166) is never used and should be removed
Address.functionStaticCall(address,bytes) (node_modules/@openzeppelin/contracts/utils/Address.sol#180-193) is never used and should be removed
Address.functionStaticCallWithValue(address,bytes) (node_modules/@openzeppelin/contracts/utils/Address.sol#201-221) is never used and should be removed
Address.functionStaticCallWithValue(address,bytes,uint256) (node_modules/@openzeppelin/contracts/utils/Address.sol#201-221) is never used and should be removed
Address.functionStaticCallWithValue(address,bytes) (node_modules/@openzeppelin/contracts/utils/Address.sol#201-221) is never used and should be removed
Governable_Governable.Governable() (contracts/access/Governable.sol#4-2) is never used and should be removed
InitializableDisable_disableInitializers() (node_modules/@openzeppelin/contracts/proxy/utils/Initializable.sol#129-131) is never used and should be removed
InitializableSetInitializableVersion(uint32) (node_modules/@openzeppelin/contracts/proxy/utils/Initializable.sol#133-148) is never used and should be removed
UsingStalePeriod._priceSetSale(uint256,uint256) (contracts/features/UsingStalePeriod.sol#45-47) is never used and should be removed
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#read-code

Pragma version='0.8.2' (node_modules/@openzeppelin/contracts/proxy/utils/Initializable.sol#4) allows old versions
Pragma version='0.8.1' (node_modules/@openzeppelin/contracts/proxy/utils/Initializable.sol#4) allows old versions
Pragma version=0.8.5 (contracts/access/Governable.sol#3) necessitates a version too recent to be trusted. Consider deploying with 0.6.12/0.7.6/0.8.7
Pragma version=0.8.4 (contracts/features/UsingStalePeriod.sol#3) necessitates a version too recent to be trusted. Consider deploying with 0.6.12/0.7.6/0.8.7
Pragma version=0.8.3 (contracts/interface/IDeveloper.sol#3) necessitates a version too recent to be trusted. Consider deploying with 0.6.12/0.7.6/0.8.7
Pragma version=0.8.2 (contracts/interface/IForceProvider.sol#3) necessitates a version too recent to be trusted. Consider deploying with 0.6.12/0.7.6/0.8.7
solc=0.8.5 is not recommended for deployment
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#incorrect-versions-of-solidity

Low level call in Address.sendValue(address,uint256) (node_modules/@openzeppelin/contracts/Address.sol#65):
  - (success) = recipient.call.value(amount) (node_modules/@openzeppelin/contracts/Address.sol#128-139)
Low level call in Address.functionCallWithValue(address,bytes,uint256,string) (node_modules/@openzeppelin/contracts/Address.sol#128-139):
  - (success,returndata) = target.callWithValue(value)(data) (node_modules/@openzeppelin/contracts/Address.sol#137)
Low level call in Address.functionDelegateCall(address,bytes) (node_modules/@openzeppelin/contracts/Address.sol#128-139):
  - (success,returndata) = target.delegatecall(data) (node_modules/@openzeppelin/contracts/Address.sol#157-166)
Low level call in Address.functionStaticCall(address,bytes,string) (node_modules/@openzeppelin/contracts/Address.sol#128-139):
  - (success,returndata) = target.staticcall(data) (node_modules/@openzeppelin/contracts/Address.sol#157-166)
Function Governance_Governable.init() (contracts/access/Governable.sol#4-2) is not in mixedCase
Parameter Governor.transferGovernorship(address) _proposedGovernor (contracts/access/Governable.sol#6) is not in mixedCase
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#conformance-to-solidity-naming-conventions

```

# AUTOMATED TESTING

## UniswapV2LikeLpTokenOracle.sol

```
UniswapV2LikeExchange._getAmountsIn(uint256,address[]),amounts (contracts/swapper/UniswapV2LikeExchange.sol#14C) is a local variable never initialized  
UniswapV2LikeExchange._getAmountsOut(uint256,address[]),amounts (contracts/swapper/UniswapV2LikeExchange.sol#152) is a local variable never initialized  
Reference: https://github.com/cryto/sithter/wiki/Detector-Documentation#uninitialized-local-variables  
  
UniswapV2LikeExchange._swapExactInput(bytes,uint256,uint256,address) (contracts/swapper/UniswapV2LikeExchange.sol#146) ignores return value by _tokenIn.approve(address(router),type() uint256.max) (contracts/swapper/UniswapV2LikeExchange.sol#181)[7]  
UniswapV2LikeExchange._swapExactOutput(bytes,uint256,uint256,address,address) (contracts/swapper/UniswapV2LikeExchange.sol#152-165) ignores return value by _tokenIn.approve(address(router),type() uint256.max) (contracts/swapper/UniswapV2LikeExchange.sol#181)[34]  
UniswapV2LikeExchange._getAmountsOut(uint256,address[],) (contracts/swapper/UniswapV2LikeExchange.sol#151-155) ignores return value by router.getAmountsOut(amountIn_path) (contracts/swapper/UniswapV2LikeExchange.sol#152-154)  
Reference: https://github.com/cryto/sithter/wiki/Detector-Documentation#ensure-return  
  
UniswapV2LikeExchange._swapExactInput(bytes,uint256,uint256,address) (contracts/swapper/UniswapV2LikeExchange.sol#156) lacks a zero-check on:  
    - _amountIn.amounts = amount (contracts/swapper/UniswapV2LikeExchange.sol#156)[89]  
UniswapV2LikeExchange._updateWhitelistedToken(address) (contracts/swapper/UniswapV2LikeExchange.sol#158) lacks a zero-check on:  
    - _whitelist = whitelist (contracts/swapper/UniswapV2LikeExchange.sol#158)[86]  
Reference: https://github.com/cryto/sithter/wiki/Detector-Documentation#zero-address-validation  
  
Variable UniswapV2LikeExchange._getAmountsOut(uint256,address[],amounts) (contracts/swapper/UniswapV2LikeExchange.sol#152) is in UniswapV2LikeExchange._getAmountsOut(uint256,address[],) (contracts/swapper/UniswapV2LikeExchange.sol#151-155)  
possibly used before declaration (contracts/swapper/UniswapV2LikeExchange.sol#152)[10]  
Variable UniswapV2LikeExchange._getAmountsOut(uint256,address[],amounts) (contracts/swapper/UniswapV2LikeExchange.sol#162) is in UniswapV2LikeExchange._getAmountsOut(uint256,address[],) (contracts/swapper/UniswapV2LikeExchange.sol#161-165) p  
ossibly used before declaration (contracts/swapper/UniswapV2LikeExchange.sol#162)[10]  
Reference: https://github.com/cryto/sithter/wiki/Detector-Documentation#pre-declaration-usage-of-local-variables  
  
UniswapV2LikeExchange._swapExactOutput(bytes,uint256,uint256,address,addreses) (contracts/swapper/UniswapV2LikeExchange.sol#145) uses timestamp for comparisons  
    - _remainingAmount > 0 (contracts/swapper/UniswapV2LikeExchange.sol#142)  
Reference: https://github.com/cryto/sithter/wiki/Detector-Documentation#block-timestamp  
  
Address.verifyCallResult(bool,bytes,string) (node_modules/Bopenseppin/contracts/utils/Address.sol#201-221) uses assembly  
    - INLINE ASM (node_modules/Bopenseppin/contracts/utils/Address.sol#219-216)  
Reference: https://github.com/cryto/sithter/wiki/Detector-Documentation#assembly-useage  
  
Different versions of Solidity is used:  
Version used: 0.8.4,  
    - 0.8.4 (node_modules/Bopenseppin/contracts/proxy/Util.sol#14)  
    - 0.8.4 (node_modules/Bopenseppin/contracts/token/ERC20.sol#14)  
    - 0.8.4 (node_modules/Bopenseppin/contracts/token/ERC20/SafeERC20.sol#14)  
    - 0.8.4 (node_modules/Bopenseppin/contracts/token/ERC20/IERC20.sol#14)  
    - 0.8.4 (node_modules/Bopenseppin/contracts/token/ERC20/IERC20.sol#15-19)  
    - 0.8.4 (node_modules/Bopenseppin/contracts/token/ERC20/SafeERC20.sol#14)  
    - 0.8.4 (node_modules/Bunswap/v2-periphery/contracts/interfaces/IUniswapV2Router01.sol#8)  
    - 0.8.4 (node_modules/Bunswap/v2-periphery/contracts/interfaces/IUniswapV2Router01.sol#16)  
    - 0.8.5 (node_modules/Governable.sol#3)  
    - 0.8.5 (node_modules/Governable.sol#13)  
    - 0.8.5 (node_modules/Governable.sol#33)  
    - 0.8.5 (node_modules/Governable.sol#43)  
Reference: https://github.com/cryto/sithter/wiki/Detector-Documentation#different-pragma-directives-are-used  
  
Address.functionCall(address,bytes) (node_modules/Bopenseppin/contracts/utils/Address.sol#85-97) is never used and should be removed  
Address.functionCallWithValue(address,bytes,uint256) (node_modules/Bopenseppin/contracts/utils/Address.sol#114-120) is never used and should be removed  
Address.functionDelegatecall(address,bytes,bytecode) (node_modules/Bopenseppin/contracts/utils/Address.sol#114-120) is never used and should be removed  
Address.functionStaticCall(address,bytes) (node_modules/Bopenseppin/contracts/utils/Address.sol#114-119) is never used and should be removed  
Address.functionCreate(bytes,bytes,bytecode) (node_modules/Bopenseppin/contracts/utils/Address.sol#114-119) is never used and should be removed  
Address.functionCreateWithValue(address,uint256) (node_modules/Bopenseppin/contracts/utils/Address.sol#114-119) is never used and should be removed  
Context._msgData (node_modules/Bopenseppin/contracts/utils/Context.sol#21-23) is never used and should be removed  
Governable._governableInit(Governable_init) (node_modules/Bopenseppin/contracts/Governable.sol#12-13) is never used and should be removed  
Initializable._init(Initializable_init) (node_modules/Bopenseppin/contracts/proxy/Util.sol#126-131) is never used and should be removed  
Initializable._initialized(Initialized_init) (node_modules/Bopenseppin/contracts/proxy/Util.sol#133-140) is never used and should be removed  
Initializable._isInitialized(Initialized_isInitialized) (node_modules/Bopenseppin/contracts/proxy/Util.sol#133-140) is never used and should be removed  
Initializable._safeTransferFromSafeERC20(address, address, address, uint256) (node_modules/Bopenseppin/contracts/token/ERC20/SafeERC20.sol#16-67) is never used and should be removed  
SafeERC20._safeIncreaseAllowance(IERC20,address,uint256) (node_modules/Bopenseppin/contracts/token/ERC20/SafeERC20.sol#16-67) is never used and should be removed  
SafeERC20._safeTransferFrom(IERC20,address,address,address,uint256) (node_modules/Bopenseppin/contracts/token/ERC20/SafeERC20.sol#16-67) is never used and should be removed  
Reference: https://github.com/cryto/sithter/wiki/Detector-Documentation#read-code  
  
Pragma version<0.8.2 (node_modules/Bopenseppin/contracts/proxy/Util.sol#113) allows old versions  
Pragma version<0.8.3 (node_modules/Bopenseppin/contracts/proxy/Util.sol#114) allows old versions  
Pragma version<0.8.0 (node_modules/Bopenseppin/contracts/token/ERC20.sol#14) allows old versions  
Pragma version<0.8.1 (node_modules/Bopenseppin/contracts/token/Address.sol#14) allows old versions  
Pragma version<0.8.0 (node_modules/Bopenseppin/contracts/utils/Address.sol#14) allows old versions  
Pragma version<0.8.2 (node modules/Bunswap/v2-periphery/contracts/interfaces/IUniswapV2Router01.sol#11) allows old versions  
Pragma version<0.6.2 (node_modules/Bunswap/v2-periphery/contracts/interfaces/IUniswapV2Router02.sol#11) allows old versions  
Pragma version<0.8.4 (node_modules/Bunswap/v2-periphery/ContractsInterface.sol#13) necessitates a version too recent to be trusted. Consider deploying with 0.6.12/0.7.6/0.8.7  
Pragma version<0.8.3 (node_modules/Bunswap/v2-periphery/ContractsInterface.sol#13) necessitates a version too recent to be trusted. Consider deploying with 0.6.12/0.7.6/0.8.7  
Pragma version<0.8.9 (node_modules/swapper/UniswapV2LikeExchange.sol#8) uses lesser than  
Pragma version<0.8.9 (node_modules/swapper/UniswapV2LikeExchange.sol#8) necessitates a version too recent to be trusted. Consider deploying with 0.6.12/0.7.6/0.8.7  
salt (node_modules/Bunswap/v2-core/ContractsInterface.sol#16) is never used and should be removed  
Reference: https://github.com/cryto/sithter/wiki/Detector-Documentation#incorrect-versions-of-solidity  
  
Low level call in Address.sendValue(address,uint256)  
    - (success) call.value(amount) (node_modules/Bopenseppin/contracts/utils/Address.sol#60-65);  
Low level call in Address.functionCallWithValue(address,bytes,uint256, string) (node_modules/Bopenseppin/contracts/utils/Address.sol#126-139):  
    - (success, returnData) target.staticcall(data) (node_modules/Bopenseppin/contracts/proxy/Util.sol#157-161);  
Low level call in Address.functionStaticCall(address,bytes,bytecode) (node_modules/Bopenseppin/contracts/proxy/Util.sol#157-161):  
    - (success, returnData) target.staticcall(data) (node_modules/Bopenseppin/contracts/proxy/Util.sol#157-161);  
Reference: https://github.com/cryto/sithter/wiki/Detector-Documentation#low-level-calls  
  
Function UniswapV2Router01._wethToken() (node_modules/Bunswap/v2-periphery/contracts/interfaces/IUniswapV2Router01.sol#5) is not in mixedCase  
Function Governable._governableInit() (Contracts/access/Governable.sol#85-92) is not in mixedCase  
Parameter Governable.transferOwnership(address),_proposedGovernor (Contracts/access/Governable.sol#86) is not in mixedCase  
Reference: https://github.com/cryto/sithter/wiki/Detector-Documentation#conformance-to-solidity-naming-conventions  
  
Variable UniswapV2Router01._address((address,address,uint256,uint256,uint256,uint256,uint256,uint256),amountsDesired) (node_modules/Bunswap/v2-periphery/ContractsInterface/IUniswapV2Router01.sol#10) is too similar to IUniswapV2Router01._address((address,address,uint256,uint256,uint256,uint256,uint256,uint256),amountOutB) (contract s/swapper/UniswapV2LikeExchange.sol#159)  
Reference: https://github.com/cryto/sithter/wiki/Detector-Documentation#variable-names-are-too-similar
```

## VspNetMainnetOracle.sol

```
UsingStalePeriod._priceFuncIndex(uint256,uint256) (contracts/features/UsingStalePeriod.sol#4-47) uses timestamp for comparisons  
    - block.timestamp - timeLastUpdate > stalePeriod (contracts/features/UsingStalePeriod.sol#46)  
VspNetMainnetOracle._getPriceInUSD(address) (contracts/features/priceToken/vspNetMainnetOracle.sol#42-46) uses timestamp for comparisons  
    - (bool,string)_priceIndexInUSD(address) (contracts/features/priceToken/vspNetMainnetOracle.sol#42-46)[2] uses timestamp for comparison  
    - (bool,string)_priceIndexInUSD(address) (contracts/features/priceToken/vspNetMainnetOracle.sol#42-46)[3] uses timestamp for comparison  
    - (bool,string)_priceIndexInUSD(address) (contracts/features/priceToken/vspNetMainnetOracle.sol#42-46)[4] uses timestamp for comparison  
    - (bool,string)_priceIndexInUSD(address) (contracts/features/priceToken/vspNetMainnetOracle.sol#42-46)[5] uses timestamp for comparison  
Reference: https://github.com/cryto/sithter/wiki/Detector-Documentation#block-timestamp  
  
Address.verifyCallResult(bool,bytes,string) (node_modules/Bopenseppin/contracts/utils/Address.sol#201-221) uses assembly  
    - INLINE ASM (node_modules/Bopenseppin/contracts/utils/Address.sol#219-216)  
Reference: https://github.com/cryto/sithter/wiki/Detector-Documentation#assembly-useage  
  
Different versions of Solidity is used:  
Version used: 0.8.4,  
    - 0.8.2 (node modules/Bopenseppin/contracts/proxy/Util.sol#14)  
    - 0.8.1 (node modules/Bopenseppin/contracts/token/Address.sol#14)  
    - 0.8.0 (node modules/Bunswap/v2-core/ContractsInterface.sol#14)  
    - 0.8.4 (node modules/Bunswap/v2-core/ContractsInterface.sol#14)[4] uses timestamp for comparisons  
    - >=0.5.0 (node modules/Bunswap/v2-core/ContractsInterface/IUniswapV2Pair.sol#1)  
    - 0.8.9 (node modules/Governable.sol#3)  
    - 0.8.5 (node features/UsingProviderAggregator.sol#3)  
    - 0.8.9 (node features/UsingTabletCoinProvider.sol#3)  
    - 0.8.9 (node features/UsingTabletPriceProvider.sol#3)  
    - 0.8.9 (contracts/interface/Governable.sol#3)  
    - 0.8.9 (contracts/interface/Code/IPriceProvider.sol#3)  
    - 0.8.9 (contracts/interface/Code/IUniswapV2Pair.sol#3)  
    - 0.8.9 (contracts/interface/Code/IUniswapV2Pair.sol#3)[4] uses timestamp for comparisons  
    - 0.8.9 (contracts/interface/Code/IUniswapV2PriceProvider.sol#3)  
    - 0.8.9 (contracts/interface/Code/IUniswapV2PriceProvider.sol#3)[4] uses timestamp for comparisons  
    - 0.8.9 (node libraries/BytesTypes.sol#3)  
    - 0.8.9 (node libraries/BytesTypes.sol#3)[2] uses timestamp for comparisons  
Reference: https://github.com/cryto/sithter/wiki/Detector-Documentation#different-pragma-directives-are-used  
  
Address.functionCall(address,bytes) (node_modules/Bopenseppin/contracts/utils/Address.sol#85-87) is never used and should be removed  
Address.functionCallWithValue(address,bytes, string) (node_modules/Bopenseppin/contracts/utils/Address.sol#114-121) is never used and should be removed  
Address.functionCallWithValue(address,bytes,uint256) (node_modules/Bopenseppin/contracts/utils/Address.sol#114-120) is never used and should be removed  
Address.functionDeleteCall(address,bytes, string) (node_modules/Bopenseppin/contracts/utils/Address.sol#114-119) is never used and should be removed  
Address.functionDeleteCall(address,bytes,uint256) (node_modules/Bopenseppin/contracts/utils/Address.sol#114-119) is never used and should be removed  
Address.functionStaticCall(address,bytes,bytecode) (node_modules/Bopenseppin/contracts/utils/Address.sol#114-119) is never used and should be removed  
Address.functionCreate(address,bytes,bytecode) (node_modules/Bopenseppin/contracts/utils/Address.sol#114-119) is never used and should be removed  
Address.isContract(address) (node_modules/Bopenseppin/contracts/utils/Address.sol#43-45) is never used and should be removed  
Address.sendValue(address,uint256) (node_modules/Bopenseppin/contracts/utils/Address.sol#10-14) is never used and should be removed  
Context._msgData (node modules/Bopenseppin/contracts/utils/Context.sol#21-23) is never used and should be removed  
Governable._governableInit() (Contracts/access/Governable.sol#38-42) is never used and should be removed
```

## ChainlinkAndFallbacksOracle.sol

```

InitializationUtil((uint256)(0)) (node_modules/@openzeppelin/contracts/proxy/utils/Initializable.sol#129-139) is never used and should be removed
Initializable, setInitializedVersion(uint) (node_modules/@openzeppelin/contracts/proxy/utils/Initializable.sol#133-145) is never used and should be removed
Math.average(uint256,uint256) (node_modules/@openzeppelin/contracts/math/Math.sol#28-31) is never used and should be removed
Math.ceilDivide(uint256,uint256) (node_modules/@openzeppelin/contracts/math/Math.sol#39-43) is never used and should be removed
Math.min(uint256,uint256) (node_modules/@openzeppelin/contracts/math/Math.sol#144-147) is never used and should be removed
Reference: https://github.com/crytic/slither/wikidoc/Detector-Documentation#used-code

Pragma version 0.8.2 (node_modules/@openzeppelin/contracts/proxy/utils/Initializable.sol#4) allows old versions
Pragma version 0.8.1 (node_modules/@openzeppelin/contracts/proxy/utils/Address.sol#4) allows old versions
Pragma version 0.8.0 (node_modules/@openzeppelin/contracts/utils/Context.sol#4) allows old versions
Pragma version 0.8.0 (node_modules/@openzeppelin/contracts/utils/Context.sol#4) allows old versions
Pragma version 0.8.0 (node_modules/@uniswap/v2-core/contracts/interfaces/IUniswapV2Pair.sol#8) allows old versions
Pragma version 0.8.0 (contracts/access/Governable.sol#9) necessitates a version too recent to be trusted. Consider deploying with 0.6.12/0.7.6/0.8.7
Pragma version 0.8.0 (contracts/features/UsingPancakeAggregator.sol#3) necessitates a version too recent to be trusted. Consider deploying with 0.6.12/0.7.6/0.8.7
Pragma version 0.8.0 (contracts/features/UsingSafeCast.sol#9) necessitates a version too recent to be trusted. Consider deploying with 0.6.12/0.7.6/0.8.7
Pragma version 0.8.0 (contracts/interfaces/core/IFreeProvider.sol#3) necessitates a version too recent to be trusted. Consider deploying with 0.6.12/0.7.6/0.8.7
Pragma version 0.8.0 (contracts/interfaces/core/IUniswapV2LikeFreeProvider.sol#3) necessitates a version too recent to be trusted. Consider deploying with 0.6.12/0.7.6/0.8.7
Pragma version 0.8.0 (contracts/interfaces/core/IUniswapV2LikeFreeProvider.sol#3) necessitates a version too recent to be trusted. Consider deploying with 0.6.12/0.7.6/0.8.7
Pragma version 0.8.0 (contracts/interfaces/core/ILockableContractProvider.sol#3) necessitates a version too recent to be trusted. Consider deploying with 0.6.12/0.7.6/0.8.7
Pragma version 0.8.0 (contracts/interfaces/core/IUniswapV2LikeOracle.sol#3) necessitates a version too recent to be trusted. Consider deploying with 0.6.12/0.7.6/0.8.7
Pragma version 0.8.0 (contracts/interfaces/core/IUniswapV2Pair.sol#3) necessitates a version too recent to be trusted. Consider deploying with 0.6.12/0.7.6/0.8.7
Pragma version 0.8.0 (contracts/libraries/Types.sol#8) necessitates a version too recent to be trusted. Consider deploying with 0.6.12/0.7.6/0.8.7
Pragma version 0.8.0 (contracts/pricing/FixedPriceToken.sol#3) necessitates a version too recent to be trusted. Consider deploying with 0.6.12/0.7.6/0.8.7
Pragma version 0.8.0 (contracts/pricing/PriceOracle.sol#3) necessitates a version too recent to be trusted. Consider deploying with 0.6.12/0.7.6/0.8.7
Reference: https://github.com/crytic/slither/wikidoc/Detector-Documentation#incorrect-versions-of-solidity

Low level call in Address.add(value,address,uint256) (node_modules/@openzeppelin/contracts/utils/Address.sol#60-65):
- (success) = recipient.call.value(amount) (node_modules/@openzeppelin/contracts/utils/Address.sol#60)
Low level call in Address.functionCallWithValue(address,bytes,uint256,string) (node_modules/@openzeppelin/contracts/utils/Address.sol#128-139):
- (success) = target.functionCallWithValue(data,calldata) (node_modules/@openzeppelin/contracts/utils/Address.sol#137)
Low level call in Address.functionStaticCall(addresses,bytes,string) (node_modules/@openzeppelin/contracts/utils/Address.sol#157-166):
- (success,returnData) = target.staticcall(data) (node_modules/@openzeppelin/contracts/utils/Address.sol#164)
Low level call in Address.functionStaticReturnData(addresses,bytes) (node_modules/@openzeppelin/contracts/utils/Address.sol#184-193):
- (success,returnData) = target.staticcall(data) (node_modules/@openzeppelin/contracts/utils/Address.sol#192)
Reference: https://github.com/crytic/slither/wikidoc/Detector-Documentation#low-level-calls

Function IUniswapV2Pair.DOMAIN_SEPARATOR() (node_modules/@uniswap/v2-core/contracts/interfaces/IUniswapV2Pair.sol#18) is not in mixedCase
Function IUniswapV2Pair.PERMIT_TYPEHASH() (node_modules/@uniswap/v2-core/contracts/interfaces/IUniswapV2Pair.sol#18) is not in mixedCase
Function IUniswapV2Pair.unsafeTransferFrom(address,address,uint256) (node_modules/@uniswap/v2-core/contracts/interfaces/IUniswapV2Pair.sol#36) is not in mixedCase
Function Governorable.Governable_init!() (contracts/access/Governable.sol#175-182) is not in mixedCase
Parameter Governorable.transferGovernorship(address_) proposedGovernor (contracts/access/Governable.sol#86) is not in mixedCase
Parameter VpMinOracle.VpMinOracle_init!() (contracts/pricing/FixedPriceToken.sol#1942) is not in mixedCase
Reference: https://github.com/crytic/slither/wikidoc/Detector-Documentation#variable-name-conventions

Variable IFixPriceProviderAggregator.priceProviders(IFixPriceProvider provider_) priceProviders (contracts/interfaces/core/IFixPriceProviderAggregator.sol#89) is too similar to IFixPriceProviderAggregator.setPriceProvider(IFixPriceProvider,IFixPriceProvider)
Reference: https://github.com/crytic/slither/wikidoc/Detector-Documentation#variable-names-are-too-similar

Variables IFixPriceProviderAggregator.priceProviders(IFixPriceProvider provider_) priceProviders (contracts/interfaces/core/IFixPriceProviderAggregator.sol#107)
Reference: https://github.com/crytic/slither/wikidoc/Detector-Documentation#variable-names-are-too-similar

ChainlinkAndFallbacksOracle.getPriceInUSD(DataTypes.Provider,address) _priceInUSD (contracts/periphery/ChainlinkAndFallbacksOracle.sol#217) is a local variable never initialized
ChainlinkAndFallbacksOracle.getQuote(DataTypes.Provider,address,uint256) _quote (contracts/periphery/ChainlinkAndFallbacksOracle.sol#218) is a local variable never initialized
ChainlinkAndFallbacksOracle.getQuoteInUSD(DataTypes.Provider,address,uint256) _quoteInUSD (contracts/periphery/ChainlinkAndFallbacksOracle.sol#219) is a local variable never initialized
ChainlinkAndFallbacksOracle.quote(DataTypes.Provider,address,uint256) _lastUpdatedAt (contracts/periphery/ChainlinkAndFallbacksOracle.sol#237) is a local variable never initialized
ChainlinkAndFallbacksOracle.quote(DataTypes.Provider,address,uint256) _lastUpdatedAt (contracts/periphery/ChainlinkAndFallbacksOracle.sol#237) is a local variable never initialized
ChainlinkAndFallbacksOracle.quoteToEth(DataTypes.Provider,address,uint256) _amountOut (contracts/periphery/ChainlinkAndFallbacksOracle.sol#234) is a local variable never initialized
ChainlinkAndFallbacksOracle.quoteToEth(DataTypes.Provider,address,uint256) _lastUpdatedAt (contracts/periphery/ChainlinkAndFallbacksOracle.sol#235) is a local variable never initialized
Reference: https://github.com/crytic/slither/wikidoc/Detector-Documentation#initializes-local-variables

ChainlinkAndFallbacksOracle.getLastPrice(DataTypes.Provider,address) getLastPrice (contracts/periphery/ChainlinkAndFallbacksOracle.sol#223) ignores return value by providersAggregator.getPriceInUSD(provider_,token_) (contracts/periphery/ChainlinkAndFallbacksOracle.sol#223)
ChainlinkAndFallbacksOracle.quote(DataTypes.Provider,address,uint256) quote (contracts/periphery/ChainlinkAndFallbacksOracle.sol#229-242) ignores return value by providersAggregator.quote(provider_,tokenIn_,tokenOut_,amountIn_) (contracts/periphery/ChainlinkAndFallbacksOracle.sol#235)
ChainlinkAndFallbacksOracle.quote(DataTypes.Provider,address,uint256) quote (contracts/periphery/ChainlinkAndFallbacksOracle.sol#248-260) ignores return value by providersAggregator.quote(tokenToEth(provider_,token_,amountIn_)) (contracts/periphery/ChainlinkAndFallbacksOracle.sol#251)
ChainlinkAndFallbacksOracle.quote(DataTypes.Provider,address,uint256) quote (contracts/periphery/ChainlinkAndFallbacksOracle.sol#266-278) ignores return value by providersAggregator.quote(tokenToEth(provider_,token_,amountIn_)) (contracts/periphery/ChainlinkAndFallbacksOracle.sol#270)
Reference: https://github.com/crytic/slither/wikidoc/Detector-Documentation#unused-return

Variables ChainlinkAndFallbacksOracle.quote(DataTypes.Provider,address) quote (contracts/periphery/ChainlinkAndFallbacksOracle.sol#221-233) potentially used before declaration priceInUSD = _priceInUSD (contracts/periphery/ChainlinkAndFallbacksOracle.sol#220)
Variables ChainlinkAndFallbacksOracle.quote(DataTypes.Provider,address) _lastUpdatedAt (contracts/periphery/ChainlinkAndFallbacksOracle.sol#219) in ChainlinkAndFallbacksOracle.getQuoteInUSD(DataTypes.Provider,address) (contracts/periphery/ChainlinkAndFallbacksOracle.sol#218) potentially used before declaration lastUpdatedAt = _lastUpdatedAt (contracts/periphery/ChainlinkAndFallbacksOracle.sol#237) in ChainlinkAndFallbacksOracle.quote(DataTypes.Provider,address,uint256) (contracts/periphery/ChainlinkAndFallbacksOracle.sol#235)
Variables ChainlinkAndFallbacksOracle.quote(DataTypes.Provider,address) _lastUpdatedAt (contracts/periphery/ChainlinkAndFallbacksOracle.sol#219-242) potentially used before declaration amountOut = _amountOut (contracts/periphery/ChainlinkAndFallbacksOracle.sol#237) in ChainlinkAndFallbacksOracle.quote(DataTypes.Provider,address,uint256) (contracts/periphery/ChainlinkAndFallbacksOracle.sol#235)
Variables ChainlinkAndFallbacksOracle.quote(DataTypes.Provider,address) quote (contracts/periphery/ChainlinkAndFallbacksOracle.sol#234) in ChainlinkAndFallbacksOracle.quote(DataTypes.Provider,address,uint256) (contracts/periphery/ChainlinkAndFallbacksOracle.sol#235)
Variables ChainlinkAndFallbacksOracle.quote(DataTypes.Provider,address) quote (contracts/periphery/ChainlinkAndFallbacksOracle.sol#235) in ChainlinkAndFallbacksOracle.quote(DataTypes.Provider,address,uint256) (contracts/periphery/ChainlinkAndFallbacksOracle.sol#236)
Variables ChainlinkAndFallbacksOracle.quote(DataTypes.Provider,address) quoteTokenToUSD (contracts/periphery/ChainlinkAndFallbacksOracle.sol#235) in ChainlinkAndFallbacksOracle.quote(DataTypes.Provider,address,uint256) (contracts/periphery/ChainlinkAndFallbacksOracle.sol#236)
Variables ChainlinkAndFallbacksOracle.quote(DataTypes.Provider,address) quoteTokenToUSD (contracts/periphery/ChainlinkAndFallbacksOracle.sol#235-240) in ChainlinkAndFallbacksOracle.quote(DataTypes.Provider,address,uint256) (contracts/periphery/ChainlinkAndFallbacksOracle.sol#236)
Variables ChainlinkAndFallbacksOracle.quote(DataTypes.Provider,address) quoteTokenToUSD (contracts/periphery/ChainlinkAndFallbacksOracle.sol#248-260) in ChainlinkAndFallbacksOracle.quote(DataTypes.Provider,address,uint256) (contracts/periphery/ChainlinkAndFallbacksOracle.sol#251)
Variables ChainlinkAndFallbacksOracle.quote(DataTypes.Provider,address) quoteTokenToUSD (contracts/periphery/ChainlinkAndFallbacksOracle.sol#266-278) in ChainlinkAndFallbacksOracle.quote(DataTypes.Provider,address,uint256) (contracts/periphery/ChainlinkAndFallbacksOracle.sol#270)
Reference: https://github.com/crytic/slither/wikidoc/Detector-Documentation#uses-local-variables

UsingStatefulRef! _priceUpdatedAt(uint256,uint256) (node_modules/@openzeppelin/contracts/UsingStatefulRef.sol#47) uses timestamp for comparisons
- block.timestamp - timeOffsetUpdate - stalePeriod, (contracts/features/UsingStatefulRef.sol#46)
ChainlinkAndFallbacksOracle.getPriceInUSD(address) (contracts/periphery/ChainlinkAndFallbacksOracle.sol#41) uses timestamp for comparisons
Dangerous comparison: _priceIsStale >= priceTolerance (contracts/periphery/ChainlinkAndFallbacksOracle.sol#44)
Dangerous comparison: _priceIsStale >= priceTolerance _lastUpdatedAt (contracts/periphery/ChainlinkAndFallbacksOracle.sol#45)
Variables _priceIsStale >= priceTolerance _lastUpdatedAt (contracts/periphery/ChainlinkAndFallbacksOracle.sol#45)
- _priceIsOK = amountOut >= 0 && _priceIsStale >= priceTolerance _lastUpdatedAt (contracts/periphery/ChainlinkAndFallbacksOracle.sol#58)
- require(bool,string)|_priceIsOK,fallback=_failed! (contracts/periphery/ChainlinkAndFallbacksOracle.sol#60)
- !priceIsOK = _priceIsOK & !priceIsOK (contracts/periphery/ChainlinkAndFallbacksOracle.sol#61)
- !priceIsOK & !_priceIsOK (contracts/periphery/ChainlinkAndFallbacksOracle.sol#62)
- !priceIsOK || _priceIsOK (contracts/periphery/ChainlinkAndFallbacksOracle.sol#63)
- !priceIsOK || !_priceIsOK (contracts/periphery/ChainlinkAndFallbacksOracle.sol#64)
- !priceIsOK & _priceIsOK (contracts/periphery/ChainlinkAndFallbacksOracle.sol#65)
- !priceIsOK || _priceIsOK & !_priceIsOK (contracts/periphery/ChainlinkAndFallbacksOracle.sol#66)
Variables ChainlinkAndFallbacksOracle.quote(address,address,uint256) quote (contracts/periphery/ChainlinkAndFallbacksOracle.sol#128-135) uses timestamp for comparisons
Dangerous comparison: _priceIsStale >= priceTolerance (contracts/periphery/ChainlinkAndFallbacksOracle.sol#134)
- _priceIsOK = amountOut >= 0 && _priceIsStale >= priceTolerance _lastUpdatedAt (contracts/periphery/ChainlinkAndFallbacksOracle.sol#142)
- require(bool,string)|_priceIsOK,fallback=_failed! (contracts/periphery/ChainlinkAndFallbacksOracle.sol#144)
- !priceIsOK = amountOut >= 0 && _priceIsStale >= priceTolerance _lastUpdatedAt (contracts/periphery/ChainlinkAndFallbacksOracle.sol#152)
- !priceIsOK & _priceIsOK (contracts/periphery/ChainlinkAndFallbacksOracle.sol#153)
- !priceIsOK || _priceIsOK & !_priceIsOK (contracts/periphery/ChainlinkAndFallbacksOracle.sol#155)
- !priceIsOK || !_priceIsOK (contracts/periphery/ChainlinkAndFallbacksOracle.sol#156)
- require(bool,string)|_priceIsOK,fallback=_failed! (contracts/periphery/ChainlinkAndFallbacksOracle.sol#160)
Reference: https://github.com/crytic/slither/wikidoc/Detector-Documentation#block-timeStamp

Address.verifyCallResult(bool,bytes,string) (node_modules/@openzeppelin/contracts/utils/Address.sol#201-221) uses assembly
- INLINE ASSEMBLY (node_modules/@openzeppelin/contracts/utils/Address.sol#218)
Reference: https://github.com/crytic/slither/wikidoc/Detector-Documentation#assembly-inusage

Different versions of Solidity is used:
- 0.8.0, "0.8.0", "0.8.1", "0.8.2"
- 0.8.2 (node_modules/@openzeppelin/contracts/proxy/utils/Initializable.sol#4)
- 0.8.2 (node_modules/@openzeppelin/contract/utils/Address.sol#4)
- 0.8.3 (contracts/access/Governable.sol#3)
- 0.8.3 (contracts/features/UsingNativeExecution.sol#3)
- 0.8.5 (contracts/features/UsingPancakeAggregator.sol#3)
- 0.8.5 (contracts/features/UsingSafeCast.sol#3)
- 0.8.6 (contracts/interfaces/core/IFreeProvider.sol#3)
- 0.8.6 (contracts/interfaces/core/IUniswapV2LikeFreeProvider.sol#3)
- 0.8.6 (contracts/interfaces/core/IUniswapV2LikeOracle.sol#3)
- 0.8.6 (contracts/pricing/FixedPriceToken.sol#3)
- 0.8.6 (contracts/pricing/PriceOracle.sol#3)
- 0.8.6 (contracts/pricing/PriceOracle.sol#3)
- 0.8.7 (contracts/libraries/Types.sol#3)
- 0.8.7 (contracts/interfaces/core/IUniswapV2Pair.sol#3)
- 0.8.7 (contracts/interfaces/core/IUniswapV2LikePair.sol#3)
- 0.8.7 (contracts/interfaces/core/IUniswapV2LikeToken.sol#3)
Reference: https://github.com/crytic/slither/wikidoc/Detector-Documentation#different-pragma-directives-are-used

Address.functionCall(address,bytes) (node_modules/@openzeppelin/Contracts/utils/Address.sol#85-91) is never used and should be removed
Address.functionCall(address,bytes,string) (node_modules/@openzeppelin/Contracts/utils/Address.sol#90-101) is never used and should be removed
Address.functionCallWithValue(address,bytes,uint256) (node_modules/@openzeppelin/Contracts/utils/Address.sol#114-120) is never used and should be removed
Address.functionDelegateCall(address,bytes,bytes) (node_modules/@openzeppelin/Contracts/utils/Address.sol#128-134) is never used and should be removed
Address.functionDelegateCall(address,bytes,string) (node_modules/@openzeppelin/Contracts/utils/Address.sol#147-153) is never used and should be removed
Address.functionDelegateCall(address,bytes,bytes) (node_modules/@openzeppelin/Contracts/utils/Address.sol#147-174) is never used and should be removed
Address.functionDelegateCall(address,bytes,string) (node_modules/@openzeppelin/Contracts/utils/Address.sol#147-176) is never used and should be removed
Address.functionDynamicCall(address,bytes,bytes) (node_modules/@openzeppelin/Contracts/utils/Address.sol#184-188) is never used and should be removed
Address.functionDynamicCall(address,bytes,string) (node_modules/@openzeppelin/Contracts/utils/Address.sol#184-193) is never used and should be removed
Address.functionStaticCall(address,bytes,bytes) (node_modules/@openzeppelin/Contracts/utils/Address.sol#194-203) is never used and should be removed
Address.functionStaticCall(address,bytes,string) (node_modules/@openzeppelin/Contracts/utils/Address.sol#202-223) is never used and should be removed

```







## ChainlinkFeedPriceProvider.sol

[Address.verifyCallResult\(bool,bytes,string\)](#) (`node_modules/openzeppelin/contracts/utils/Address.sol#201-221`) uses assembly  
 - INLINE ASM (`node_modules/crytic-solver/wasm/Detector-Documentation#assembly-useage`  
 Reference: <https://github.com/crytic/solver/wiki/Detector-Documentation#assembly-useage>

Different versions of Solidity is used:  
 Version used: [0.8.0, "0.8.9"]  
 Version used: ["0.8.0", "0.8.9"]  
 Version used: [0.8.0]  
 Version used: [0.8.0, "0.8.9"]  
 Version used: [0.8.0, "0.8.10"]  
 Version used: [0.8.0, "0.8.11"]  
 Version used: [0.8.0, "0.8.12"]  
 Version used: [0.8.0, "0.8.13"]  
 Version used: [0.8.0, "0.8.14"]  
 Version used: [0.8.0, "0.8.15"]  
 Version used: [0.8.0, "0.8.16"]  
 Version used: [0.8.0, "0.8.17"]  
 Version used: [0.8.0, "0.8.18"]  
 Version used: [0.8.0, "0.8.19"]  
 Version used: [0.8.0, "0.8.20"]  
 Version used: [0.8.0, "0.8.21"]  
 Version used: [0.8.0, "0.8.22"]  
 Version used: [0.8.0, "0.8.23"]  
 Version used: [0.8.0, "0.8.24"]  
 Version used: [0.8.0, "0.8.25"]  
 Version used: [0.8.0, "0.8.26"]  
 Version used: [0.8.0, "0.8.27"]  
 Version used: [0.8.0, "0.8.28"]  
 Version used: [0.8.0, "0.8.29"]  
 Version used: [0.8.0, "0.8.30"]  
 Version used: [0.8.0, "0.8.31"]  
 Version used: [0.8.0, "0.8.32"]  
 Version used: [0.8.0, "0.8.33"]  
 Version used: [0.8.0, "0.8.34"]  
 Version used: [0.8.0, "0.8.35"]  
 Version used: [0.8.0, "0.8.36"]  
 Version used: [0.8.0, "0.8.37"]  
 Version used: [0.8.0, "0.8.38"]  
 Version used: [0.8.0, "0.8.39"]  
 Version used: [0.8.0, "0.8.40"]  
 Reference: <https://github.com/crytic/solver/wasm/Detector-Documentation#incorrect-pragma-directives-are-used>

Address.functionCall(address, bytes, string) (`node_modules/openzeppelin/contracts/utils/Address.sol#85-97`) is never used and should be removed  
 Address.functionCall(address, bytes, string) (`node_modules/openzeppelin/contracts/utils/Address.sol#101-103`) is never used and should be removed  
 Address.functionCallWithValue(address, bytes, uint32) (`node_modules/openzeppelin/contracts/utils/Address.sol#114-120`) is never used and should be removed  
 Address.functionDelegateCall(address, bytes, string) (`node_modules/openzeppelin/contracts/utils/Address.sol#124-130`) is never used and should be removed  
 Address.functionDelegateCall(address, bytes) (`node_modules/openzeppelin/contracts/utils/Address.sol#134-135`) is never used and should be removed  
 Address.functionStaticCall(address, bytes, string) (`node_modules/openzeppelin/contracts/utils/Address.sol#143-149`) is never used and should be removed  
 Address.functionStaticCall(address) (`node_modules/openzeppelin/contracts/utils/Address.sol#43`) is never used and should be removed  
 Address.isContract(address) (`node_modules/openzeppelin/contracts/utils/Address.sol#201-221`) is never used and should be removed  
 Address.verifyCallResult(bool,bytes,string) (`node_modules/openzeppelin/contracts/utils/Address.sol#228-230`) is never used and should be removed  
 Context, msgData) (`node_modules/openzeppelin/contracts/utils/Context.sol#20`) is never used and should be removed  
 Context, msgData)(bool,openzeppelin/contracts/utils/Context.sol#21-22) is never used and should be removed  
 Context, msgData)(address,openzeppelin/contracts/utils/Context.sol#103-105) is never used and should be removed  
 Initializable, disableInitializers) (`node_modules/openzeppelin/contracts/proxy/utils/Initializable.sol#129-131`) is never used and should be removed  
 Initializable, setInitializableDefault(uint32) (`node_modules/openzeppelin/contracts/proxy/utils/Initializable.sol#132-149`) is never used and should be removed  
 Math,abs(uint256,uint256) (`node_modules/openzeppelin/contracts/math/Math.sol#10-13`) is never used and should be removed  
 Math,ceil(uint256,uint256) (`node_modules/openzeppelin/contracts/math/Math.sol#17-22`) is never used and should be removed  
 SafeCast,toInt256(int256) (`node_modules/openzeppelin/contracts/math/MathCast.sol#206-209`) is never used and should be removed  
 SafeCast,toInt32(int256) (`node_modules/openzeppelin/contracts/math/MathCast.sol#193-202`) is never used and should be removed  
 SafeCast,toInt256(uint256) (`node_modules/openzeppelin/contracts/math/MathCast.sol#174-192`) is never used and should be removed  
 SafeCast,toInt32(uint256) (`node_modules/openzeppelin/contracts/math/MathCast.sol#155-173`) is never used and should be removed  
 SafeCast,toInt256(uint32) (`node_modules/openzeppelin/contracts/math/MathCast.sol#122-140`) is never used and should be removed  
 SafeCast,toInt32(uint32) (`node_modules/openzeppelin/contracts/math/MathCast.sol#92-110`) is never used and should be removed  
 SafeCast,toInt256(uint256,Math) (`node_modules/openzeppelin/contracts/math/MathCast.sol#47-50`) is never used and should be removed  
 SafeCast,toInt32(uint256,Math) (`node_modules/openzeppelin/contracts/math/MathCast.sol#28-46`) is never used and should be removed  
 SafeCast,toInt256(uint32,Math) (`node_modules/openzeppelin/contracts/math/MathCast.sol#32-39`) is never used and should be removed  
 SafeCast,toInt32(uint32,Math) (`node_modules/openzeppelin/contracts/math/MathCast.sol#92-95`) is never used and should be removed  
 SafeCast,toInt32(int256,Math) (`node_modules/openzeppelin/contracts/math/MathCast.sol#106-114`) is never used and should be removed  
 SafeCast,toInt32(uint256,Math) (`node_modules/openzeppelin/contracts/math/MathCast.sol#107-115`) is never used and should be removed  
 SafeCast,toInt32(uint32,Math) (`node_modules/openzeppelin/contracts/math/MathCast.sol#122-125`) is never used and should be removed  
 SafeCast,toInt32(uint32,Math) (`node_modules/openzeppelin/contracts/math/MathCast.sol#62-65`) is never used and should be removed  
 Reference: <https://github.com/crytic/solver/wasm/Detector-Documentation#incorrect-pragma-directives-are-used>

Fragma version="0.8.0" (`node_modules/chainlink/contracts/svc/v0.8/interfaces/AggregatorInterface.sol#12`) allows old versions  
 Frama version="0.8.0" (`node_modules/chainlink/contracts/svc/v0.8/interfaces/FeedRegistryInterface.sol#12`) allows old versions  
 Frama version="0.8.0" (`node_modules/chainlink/contracts/svc/v0.8/interfaces/FeedRegistryInterface.sol#12`) allows old versions  
 Frama version="0.8.0" (`node_modules/chainlink/contracts/svc/v0.8/interfaces/FeedRegistryInterface.sol#12`) allows old versions  
 Frama version="0.8.2" (`node_modules/openzeppelin/contracts/proxy/utils/Initializable.sol#18`) allows old versions  
 Frama version="0.8.2" (`node_modules/openzeppelin/contracts/proxy/utils/Initializable.sol#18`) allows old versions  
 Frama version="0.8.4" (`node_modules/openzeppelin/contracts/proxy/Proxy.sol#18`) allows old versions  
 Frama version="0.8.6" (`node_modules/openzeppelin/contracts/token/ERC20/extensions/IERC20Metadata.sol#8`) allows old versions  
 Frama version="0.8.1" (`node_modules/openzeppelin/contracts/token/Address.sol#18`) allows old versions  
 Frama version="0.8.0" (`node_modules/openzeppelin/contracts/token/Address.sol#18`) allows old versions  
 Frama version="0.8.0" (`node_modules/openzeppelin/contracts/token/Address.sol#18`) allows old versions  
 Frama version="0.8.0" (`node_modules/chainlink/contracts/pricefeed/ChainlinkFeedPriceProvider.sol#8`) necessitates a version too recent to be trusted. Consider deploying with 0.6.12/0.7.6/0.8.7  
 Frama version="0.8.9" (`node_modules/chainlink/contracts/pricefeed/ChainlinkFeedPriceProvider.sol#8`) necessitates a version too recent to be trusted. Consider deploying with 0.6.12/0.7.6/0.8.7  
 Frama version="0.8.2" (`node_modules/chainlink/contracts/pricefeed/ChainlinkFeedPriceProvider.sol#8`) necessitates a version too recent to be trusted. Consider deploying with 0.6.12/0.7.6/0.8.7  
 Frama version="0.8.9" (`node_modules/chainlink/contracts/pricefeed/ChainlinkFeedPriceProvider.sol#8`) necessitates a version too recent to be trusted. Consider deploying with 0.6.12/0.7.6/0.8.7  
 Frama version="0.8.0" (`node_modules/chainlink/contracts/pricefeed/ChainlinkFeedPriceProvider.sol#8`) necessitates a version too recent to be trusted. Consider deploying with 0.6.12/0.7.6/0.8.7  
 Frama version="0.8.0" (`node_modules/chainlink/contracts/pricefeed/ChainlinkFeedPriceProvider.sol#8`) necessitates a version too recent to be trusted. Consider deploying with 0.6.12/0.7.6/0.8.7  
 Reference: <https://github.com/crytic/solver/wasm/Detector-Documentation#incorrect-versions-of-solidity>



# AUTOMATED TESTING

```
Math.average(uint256,uint256) (node_modules/@openzeppelin/contracts/utils/math/Math.sol#20-31) is never used and should be removed
MathceilDiv(uint256,uint256) (node_modules/@openzeppelin/Contracts/utils/math/Math.sol#39-42) is never used and should be removed
Math.max(uint256,uint256) (node_modules/@openzeppelin/contracts/utils/math/Math.sol#13-15) is never used and should be removed
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#dead-code

Frama version<0.8.0 (node_modules/@openzeppelin/contracts/token/ERC20/IERC20.sol#19) allows old versions
Frama version<0.8.0 (node_modules/@openzeppelin/contracts/token/ERC20/IERC20.sol#20) allows old versions
Frama version<0.8.0 (node_modules/@openzeppelin/contracts/token/ERC20/IERC20.sol#21) allows old versions
Frama version<0.8.0 (contracts/core/PriceProvider.sol#13) necessitates a version too recent to be trusted. Consider deploying with 0.6.12/0.7.6/0.8.7
Frama Version<0.8.0 (contracts/core/PriceProvider.sol#14) necessitates a version too recent to be trusted. Consider deploying with 0.4.12/0.7.6/0.8.7
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#incorrect-versions-of-solidity

PriceProvider (contracts/core/PriceProvider.sol#12-52) does not implement functions:
- PriceProvider.getPriceUSD(address) (contracts/core/PriceProvider.sol#16)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#unimplemented-functions

UniswapV3Exchange.sol

UniswapV3Exchange.swapExactOutput(bytes,uint256,address,_tokenInAddress) (contracts/swapper/UniswapV3Exchange.sol#130) is a local variable never initialized
UniswapV3Exchange._getAmountsOut(uint256,bytes,_amount) (contracts/swapper/UniswapV3Exchange.sol#94) is a local variable never initialized
max(amountOut,0) (contracts/swapper/UniswapV3Exchange.sol#119) is a local variable never initialized
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#uninitialized-local-variables

UniswapV3Exchange._getAmountsIn(uint256,bytes) (contracts/swapper/UniswapV3Exchange.sol#87) ignores return value by _QUOTE.quoteExactOutput(path,_amountOut) (contracts/swapper/UniswapV3Exchange.sol#84-96)
UniswapV3Exchange._getAmountsOut(uint256,bytes,_amount) (contracts/swapper/UniswapV3Exchange.sol#93-97) ignores return value by _QUOTE.quoteExactInputPath(_path,_amountIn) (contracts/swapper/UniswapV3Exchange.sol#84-96)
UniswapV3Exchange._swapExactInput(bytes,uint256,address) (contracts/swapper/UniswapV3Exchange.sol#100-120) ignores return value by _tokenIn.approve(address(UNIS_ROUTER),type())(uint256).max) (contracts/swapper/UniswapV3Exchange.sol#111)
UniswapV3Exchange._swapExactOutput(bytes,uint256,address,address) (contracts/swapper/UniswapV3Exchange.sol#123-159) ignores return value by _tokenIn.approve(address(UNIS_ROUTER),type())(uint256).max) (contracts/swapper/UniswapV3Exchange.sol#130)
Range.sol#11
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#unused-return

UniswapV3Exchange.constructor(address)_wethlike (contracts/swapper/UniswapV3Exchange.sol#86) lacks a zero-check on :
- range(_amount) (contracts/swapper/UniswapV3Exchange.sol#129)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#range-address-validation

Variables "UniswapV3Exchange._getAmountsIn(uint256,bytes)_amountOut" (contracts/swapper/UniswapV3Exchange.sol#88) in UniswapV3Exchange._getAmountsIn(uint256,bytes) (contracts/swapper/UniswapV3Exchange.sol#83-87) potentially used before declaration
Variable "UniswapV3Exchange._getAmountsOut(uint256,bytes)_amountOut" (contracts/swapper/UniswapV3Exchange.sol#94" in UniswapV3Exchange._getAmountsOut(uint256,bytes) (contracts/swapper/UniswapV3Exchange.sol#93-97) potentially used before declaration
- _amountOut (contracts/swapper/UniswapV3Exchange.sol#19)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#rs-declaration-usage-of-local-variables

UniswapV3Exchange._swapExactInput(bytes,uint256,uint256,address)_address (contracts/swapper/UniswapV3Exchange.sol#123-159) uses timestamp for comparisons
- _remainingTimestamp > 0 (contracts/swapper/UniswapV3Exchange.sol#156)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#block-limiteramp

Address.isContract(address) (node_modules/@openzeppelin/contracts/v3-4.2-solc-0.7/utils/Address.sol#26-35) uses assembly
- INLINE ASM (node_modules/@openzeppelin/contracts/v3-4.2-solc-0.7/utils/Address.sol#33)
Address.isContract(address) (node_modules/@openzeppelin/contracts/v3-4.2-solc-0.7/utils/Address.sol#71-108) uses assembly
- INLINE ASM (node_modules/@openzeppelin/contracts/v3-4.2-solc-0.7/utils/Address.sol#108)
BytesLib.slice(bytes,uint256) (node_modules/@uniswap/v3-periphery/contracts/libraries/BytesLib.sol#17-76) uses assembly
BytesLib.toAddress(bytes,uint256) (node_modules/@uniswap/v3-periphery/contracts/libraries/BytesLib.sol#178-88) uses assembly
- INLINE ASM (node_modules/@uniswap/v3-periphery/contracts/libraries/BytesLib.sol#57)
BytesLib.toAddress(bytes,uint256)_toAddress (node_modules/@uniswap/v3-periphery/contracts/libraries/BytesLib.sol#100-100) uses assembly
- INLINE ASM (node_modules/@uniswap/v3-periphery/contracts/libraries/BytesLib.sol#97)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#assembly-usege

Different versions of Solidity is used:
- Versions used: {'<=0.8.14', '>=0.5.0', '>=0.6.0', '>=0.7.6', '>=0.8.0'}
- >=0.6.0.sol#3 (node_modules/@openzeppelin/contracts/v3-4.2-solc-0.7/utils/Address.sol#26-35)
- >=0.6.0.sol#3 (node_modules/@openzeppelin/contracts/v3-4.2-solc-0.7/utils/Address.sol#71-108)
- >=0.6.0.sol#3 (node_modules/@openzeppelin/contracts/v3-4.2-solc-0.7/utils/Address.sol#108)
- >=0.6.0.sol#3 (node_modules/@openzeppelin/contracts/v3-4.2-solc-0.7/utils/Address.sol#129)
- >=0.6.0.sol#3 (node_modules/@openzeppelin/contracts/v3-4.2-solc-0.7/utils/Address.sol#156)
- >=0.6.0.sol#3 (node_modules/@uniswap/v3-periphery/contracts/libraries/BytesLib.sol#17-76)
- >=0.6.0.sol#3 (node_modules/@uniswap/v3-periphery/contracts/libraries/BytesLib.sol#178-88)
- >=0.6.0.sol#3 (node_modules/@uniswap/v3-periphery/contracts/libraries/BytesLib.sol#57)
- >=0.6.0.sol#3 (node_modules/@uniswap/v3-periphery/contracts/libraries/BytesLib.sol#100-100)
- >=0.6.0.sol#3 (node_modules/@uniswap/v3-periphery/contracts/libraries/BytesLib.sol#97)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#different-pragma-directives-are-used

Address.functionCall(address,bytes) (node_modules/@openzeppelin/contracts/v3-4.2-solc-0.7/utils/Address.sol#79-81) is never used and should be removed
Address.functionCallWithValue(address,bytes,uint256) (node_modules/@openzeppelin/contracts/v3-4.2-solc-0.7/utils/Address.sol#104-106) is never used and should be removed
Address.functionDelegateCall(address,bytes) (node_modules/@openzeppelin/contracts/v3-4.2-solc-0.7/utils/Address.sol#113-116) is never used and should be removed
Address.functionStaticCall(address,bytes) (node_modules/@openzeppelin/contracts/v3-4.2-solc-0.7/utils/Address.sol#129-131) is never used and should be removed
Address.functionStaticCall(address,bytes,string) (node_modules/@openzeppelin/contracts/v3-4.2-solc-0.7/utils/Address.sol#139-145) is never used and should be removed
Address.functionStaticCall(address,bytes,bytes) (node_modules/@openzeppelin/contracts/v3-4.2-solc-0.7/utils/Address.sol#151-153) is never used and should be removed
Path.hasMultiplePaths (node_modules/@uniswap/v3-periphery/contracts/libraries/Path.sol#25-27) is never used and should be removed
Path.hasPathTo (node_modules/@uniswap/v3-periphery/contracts/libraries/Path.sol#101-103) is never used and should be removed
Path.isPathValid (node_modules/@uniswap/v3-periphery/contracts/libraries/Path.sol#104-106) is never used and should be removed
SafeERC20.safeApprove(IERC20,address,uint256) (node_modules/@openzeppelin/contracts/v3-4.2-solc-0.7/token/ERC20/SafeERC20.sol#12-21) is never used and should be removed
SafeERC20.safeIncreaseAllowance(IERC20,address,uint256) (node_modules/@openzeppelin/contracts/v3-4.2-solc-0.7/token/ERC20/SafeERC20.sol#13-30) is never used and should be removed
SafeERC20.safeDecreaseAllowance(IERC20,address,uint256) (node_modules/@openzeppelin/contracts/v3-4.2-solc-0.7/token/ERC20/SafeERC20.sol#14-21) is never used and should be removed
SafeERC20.safeTransferFrom(IERC20,address,address,uint256) (node_modules/@openzeppelin/contracts/v3-4.2-solc-0.7/token/ERC20/SafeERC20.sol#15-20) is never used and should be removed
SafeMath.add(uint256,uint256) (node_modules/@openzeppelin/contracts/math/SafeMath.sol#18-19) is never used and should be removed
SafeMath.add(uint256,uint256)_add (node_modules/@openzeppelin/contracts/math/SafeMath.sol#19)
SafeMath.div(uint256,uint256,string) (node_modules/@openzeppelin/contracts/math/SafeMath.sol#190-193) is never used and should be removed
SafeMath.mod(uint256,uint256) (node_modules/@openzeppelin/contracts/math/SafeMath.sol#190-193)
SafeMath.mul(uint256,uint256) (node_modules/@openzeppelin/contracts/math/SafeMath.sol#116-121) is never used and should be removed
SafeMath.sub(uint256,uint256) (node_modules/@openzeppelin/contracts/math/SafeMath.sol#101-108)
SafeMath.tryAdd(uint256,uint256) (node_modules/@openzeppelin/contracts/math/SafeMath.sol#24-25)
SafeMath.tryDiv(uint256,uint256) (node_modules/@openzeppelin/contracts/math/SafeMath.sol#47-49)
SafeMath.tryMod(uint256,uint256) (node_modules/@openzeppelin/contracts/math/SafeMath.sol#40-43) is never used and should be removed
SafeMath.tryMul(uint256,uint256) (node_modules/@openzeppelin/contracts/math/SafeMath.sol#44-45)
SafeMath.trySub(uint256,uint256) (node_modules/@openzeppelin/contracts/math/SafeMath.sol#46-47) is never used and should be removed
SafeMath.trySub(uint256,uint256)_trySub (node_modules/@openzeppelin/contracts/math/SafeMath.sol#50-51) is never used and should be removed
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#read-code

Frama version<0.6.0.sol#0 (node_modules/@openzeppelin/contracts/v3-4.2-solc-0.7/math/SafeMath.sol#3) is too complex
Frama version<0.6.0.sol#0 (node_modules/@openzeppelin/contracts/v3-4.2-solc-0.7/math/SafeMath.sol#11) is too complex
Frama version<0.6.0.sol#0 (node_modules/@openzeppelin/contracts/v3-4.2-solc-0.7/math/SafeMath.sol#12) is too complex
Frama version<0.6.0.sol#0 (node_modules/@uniswap/v3-core/contracts/interface/callback/IUniswapV3SwapCallback.sol#2) allows old versions
Frama version<0.6.0.sol#0 (node_modules/@uniswap/v3-periphery/contracts/libraries/FETH.sol#2) allows old versions
Frama version<0.6.0.sol#0 (node_modules/@uniswap/v3-periphery/contracts/libraries/FETH.sol#12) allows old versions
Frama version<0.6.0.sol#0 (node_modules/@uniswap/v3-periphery/contracts/libraries/FETH.sol#14) uses less than solc=0.6.0 is not recommended for deployment
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#incorrect-versions-of-solidity

Low level call in Address.sendValue(address,uint256) (node_modules/@openzeppelin/contracts/v3-4.2-solc-0.7/utils/Address.sol#53-59):
- (success, recipient) (node_modules/@openzeppelin/contracts/v3-4.2-solc-0.7/utils/Address.sol#57)
Low level call in Address.sendValue(address,uint256)_sendValue (node_modules/@openzeppelin/contracts/v3-4.2-solc-0.7/utils/Address.sol#59-61):
- (success, recipientData) (target.callValue(value)(data)) (node_modules/@openzeppelin/contracts/v3-4.2-solc-0.7/utils/Address.sol#59)
Low level call in Address.functionStaticCall(address,bytes,string) (node_modules/@openzeppelin/contracts/v3-4.2-solc-0.7/utils/Address.sol#163-169):
- (success, recipientData) (target.delegatecall(data)) (node_modules/@openzeppelin/contracts/v3-4.2-solc-0.7/utils/Address.sol#167)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#low-level-calls

Parameter BytesLib.slice(uint256,uint256) (node_modules/@uniswap/v3-periphery/contracts/libraries/BytesLib.sol#13) is not in mixedCase
Parameter BytesLib.slice(bytes,uint256,_bytes) (node_modules/@uniswap/v3-periphery/contracts/libraries/BytesLib.sol#13)
Parameter BytesLib.slice(bytes,uint256,_bytes)_start (node_modules/@uniswap/v3-periphery/contracts/libraries/BytesLib.sol#14)
Parameter BytesLib.toAddress(bytes,uint256)_bytes (node_modules/@uniswap/v3-periphery/contracts/libraries/BytesLib.sol#78) is not in mixedCase
Parameter BytesLib.toAddress(bytes,uint256,_bytes) (node_modules/@uniswap/v3-periphery/contracts/libraries/BytesLib.sol#78)
Parameter BytesLib.toAddress(bytes,uint256)_start (node_modules/@uniswap/v3-periphery/contracts/libraries/BytesLib.sol#80)
Parameter BytesLib.toAddress(bytes,uint256,_start)_start (node_modules/@uniswap/v3-periphery/contracts/libraries/BytesLib.sol#80)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#conformance-to-solidity-naming-conventions

Variable UniswapV3SwapCallback.uniswapV3SwapCallback (node_modules/@uniswap/v3-core/contracts/interface/callback/IUniswapV3SwapCallback.sol#17) is too similar to IDUniswapV3SwapCallback.uniswapV3SwapCallback
checkInUint256(uint256,bytes)_amountDelta (node_modules/@uniswap/v3-core/contracts/interface/callback/IUniswapV3SwapCallback.sol#18)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#too-similar

ByteLib.toAddress(bytes,uint256) (node_modules/@uniswap/v3-periphery/contracts/libraries/BytesLib.sol#78-88) uses literals with too many digits:
- _recipient = mixedUint256(256) ("bytes" + 0x00... + "start") / 0xFFFFFFFF00000000000000000000000000000000 (node_modules/@uniswap/v3-periphery/contracts/libraries/BytesLib.sol#84)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#too-many-digits

UniswapV3Exchange.defaultPoolId (contracts/swapper/UniswapV3Exchange.sol#20) should be constant
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#state-variables-that-could-be-declared-constant
```

- Some unused returns were correctly flagged by Slither.
- The weak PRNG flagged by Slither is a false positive.
- No major issues found by Slither.

## 4.2 AUTOMATED SECURITY SCAN

### Description:

Halborn used automated security scanners to assist with detection of well-known security issues and to identify low-hanging fruits on the targets for this engagement. Among the tools used was MythX, a security analysis service for Ethereum smart contracts. MythX performed a scan on the smart contracts and sent the compiled results to the analyzers to locate any vulnerabilities.

### MythX results:

#### Governable.sol

No issues found by MythX.

#### ChainlinkArbitrumPriceProvider.sol

Report for contracts/core/ChainlinkArbitrumPriceProvider.sol <a href="https://dashboard.mythx.io/#/console/analyses/d22980c1-95c3-4c59-9a67-4e83fffa9f38d">https://dashboard.mythx.io/#/console/analyses/d22980c1-95c3-4c59-9a67-4e83fffa9f38d</a>			
Line	SWC Title	Severity	Short Description
10	(SWC-123) Requirement Violation	Low	Requirement violation.

#### ChainlinkAvalanchePriceProvider.sol

Report for contracts/core/ChainlinkAvalanchePriceProvider.sol <a href="https://dashboard.mythx.io/#/console/analyses/d690ffbe-f5dd-417b-8f45-9218d72d0957">https://dashboard.mythx.io/#/console/analyses/d690ffbe-f5dd-417b-8f45-9218d72d0957</a>			
Line	SWC Title	Severity	Short Description
10	(SWC-123) Requirement Violation	Low	Requirement violation.

#### ChainlinkMainnetPriceProvider.sol

No issues found by MythX.

#### ChainlinkPolygonPriceProvider.sol

Report for contracts/core/ChainlinkPolygonPriceProvider.sol <a href="https://dashboard.mythx.io/#/console/analyses/fb18196d-cf2a-442a-bfba-8db934f64646">https://dashboard.mythx.io/#/console/analyses/fb18196d-cf2a-442a-bfba-8db934f64646</a>			
Line	SWC Title	Severity	Short Description
10	(SWC-123) Requirement Violation	Low	Requirement violation.

#### ChainlinkPriceProvider.sol

Report for contracts/core/ChainlinkPriceProvider.sol <a href="https://dashboard.mythx.io/#/console/analyses/3df55a56-9594-47af-a044-4d7800c70187">https://dashboard.mythx.io/#/console/analyses/3df55a56-9594-47af-a044-4d7800c70187</a>			
Line	SWC Title	Severity	Short Description
16	(SWC-123) Requirement Violation	Low	Requirement violation.
86	(SWC-123) Requirement Violation	Low	Requirement violation.

**PriceProvidersAggregator.sol**

No issues found by MythX.

**UniswapV2LikePriceProvider.sol**

Report for contracts/core/UniswapV2LikePriceProvider.sol  
<https://dashboard.mythx.io/#/console/analyses/c399d6c2-2540-4ef4-9def-96207b0ecab3>

Line	SWC Title	Severity	Short Description
199	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "-" discovered
206	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "/" discovered
206	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "-" discovered
209	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "/" discovered
209	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "-" discovered

**UniswapV3PriceProvider.sol**

No issues found by MythX.

**Oracle.sol**

No issues found by MythX.

**SwapperOracle.sol**

Report for contracts/periphery/SwapperOracle.sol  
<https://dashboard.mythx.io/#/console/analyses/61c53b2b-714d-4360-80d8-205fc0453e51>

Line	SWC Title	Severity	Short Description
100	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "***" discovered
100	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "/" discovered
100	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "-" discovered
101	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "***" discovered
101	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "/" discovered
101	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "-" discovered
164	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "-" discovered

**Swapper.sol**

Report for contracts/swapper/Swapper.sol  
<https://dashboard.mythx.io/#/console/analyses/d66039ec-898c-4a17-b3bf-d290a3b39ec9>

Line	SWC Title	Severity	Short Description
73	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "++" discovered
90	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "/" discovered
90	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "+" discovered
90	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "***" discovered
111	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "++" discovered
128	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "***" discovered
128	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "-" discovered
128	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "/" discovered

**UniswapV2LikeExchange.sol**

Report for contracts/swapper/UniswapV2LikeExchange.sol  
<https://dashboard.mythx.io/#/console/analyses/0e1bb03e-5302-4b4e-bc9e-58ba657dc87b>

Line	SWC Title	Severity	Short Description
14	(SWC-123) Requirement Violation	Low	Requirement violation.

**CTokenOracle.sol**

Report for contracts/periphery/synth/CTokenOracle.sol  
<https://dashboard.mythx.io/#/console/analyses/40cb14a4-395e-4b66-8226-5f3bal679b46>

Line	SWC Title	Severity	Short Description
13	(SWC-123) Requirement Violation	Low	Requirement violation.
38	(SWC-123) Requirement Violation	Low	Requirement violation.

**CurveLpTokenOracle.sol**

No issues found by MythX.

**SynthDefaultOracle.sol**

Report for contracts/periphery/synth/SynthDefaultOracle.sol  
<https://dashboard.mythx.io/#/console/analyses/0517de04-ed66-4b35-8ddd-7d62dfde5e57>

Line	SWC Title	Severity	Short Description
93	(SWC-116) Timestamp Dependence	Low	A control flow decision is made based on The block.timestamp environment variable.

**UniswapV3CrossPoolOracle.sol**

No issues found by MythX.

**StableCoinProvider.sol**

Report for contracts/core/StableCoinProvider.sol  
<https://dashboard.mythx.io/#/console/analyses/9cd1a1c9-3682-4fld-975b-7a060b01de3e>

Line	SWC Title	Severity	Short Description
20	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation *** discovered

**UmbrellaPassportPriceProvider.sol**

Report for contracts/core/UmbrellaPassportPriceProvider.sol  
<https://dashboard.mythx.io/#/console/analyses/a3682c33-97cd-49d2-b79b-717840b5609f>

Line	SWC Title	Severity	Short Description
58	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "+" discovered
65	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "-" discovered
65	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "/" discovered
65	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "***" discovered
66	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "://" discovered
66	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "##" discovered
66	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "-*" discovered

**UmbrellaPriceProvider.sol**

Report for contracts/core/UmbrellaPriceProvider.sol  
<https://dashboard.mythx.io/#/console/analyses/2c01c5e8-041d-454c-81f2-6702abblc56f>

Line	SWC Title	Severity	Short Description
64	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "++" discovered
65	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "+" discovered
65	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "-" discovered
66	(SWC-110) Assert Violation	Unknown	Out of bounds array access

**UsingMaxDeviation.sol**

No issues found by MythX.

**UsingProvidersAggregator.sol**

No issues found by MythX.

**UsingStableCoinProvider.sol**

No issues found by MythX.

**UsingStalePeriod.sol**

No issues found by MythX.

**UniswapV2LikeLpTokenOracle.sol**

Report for contracts/periphery/tokens/UniswapV2LikeLpTokenOracle.sol  
<https://dashboard.mythx.io/#/console/analyses/5175b984-df83-41a9-b5fb-a82694521d4f>

Line	SWC Title	Severity	Short Description
45	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation *** discovered
46	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation *** discovered
48	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation /* discovered
48	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation *** discovered

**VspMainnetOracle.sol**

No issues found by MythX.

**ChainlinkAndFallbacksOracle.sol**

No issues found by MythX.

**SynthOracle.sol**

No issues found by MythX.

**ChainlinkFeedPriceProvider.sol**

Report for contracts/core/ChainlinkFeedPriceProvider.sol  
<https://dashboard.mythx.io/#/console/analyses/130075da-bf07-4fe2-a37a-8dad00c956fd>

Line	SWC Title	Severity	Short Description
14	(SWC-123) Requirement Violation	Low	Requirement violation.

**FluxPriceProvider.sol**

Report for contracts/core/fluxPriceProvider.sol  
<https://dashboard.mythx.io/#/console/analyses/3dc1f12d-9a11-4f0a-a3ec-dceaf41b4e35>

Line	SWC Title	Severity	Short Description
16	(SWC-123) Requirement Violation	Low	Requirement violation.
81	(SWC-123) Requirement Violation	Low	Requirement violation.

**PriceProvider.sol**

Report for contracts/core/ChainlinkPriceProvider.sol  
<https://dashboard.mythx.io/#/console/analyses/130075da-bf07-4fe2-a37a-8dad00c956fd>

Line	SWC Title	Severity	Short Description
53	(SWC-123) Requirement Violation	Low	Requirement violation.

**UniswapV3Exchange.sol**

No issues found by MythX.

- The requirement violations and assert violations are all false positives.

- Integer Overflows and Underflows flagged by MythX are false positives, as those contracts are using Solidity ^0.8.0 version. After the Solidity version 0.8.0 Arithmetic operations revert to underflow and overflow by default.
- No major issues found by MythX.

THANK YOU FOR CHOOSING  
HALBORN