Archimedes Finance

Smart Contract Security Audit

Prepared by: **Halborn**

Date of Engagement: **October 31st, 2022 - November 11th, 2022**

Visit: **Halborn.com**

DOCUMENT REVISION HISTORY

VERSION	MODIFICATION	DATE	AUTHOR
0.1	Document Creation	11/11/2022	Manuel García
0.2	Draft Review	11/14/2022	Kubilay Onur Gungor
0.3	Draft Review	11/14/2022	Gabi Urrutia
1.0	Remediation Plan	11/18/2022	Manuel García
1.1	Remediation Plan Review	11/21/2022	Kubilay Onur Gungor
1.2	Remediation Plan Review	11/21/2022	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
Kubilay Onur Gungor	Halborn	Kubilay.Gungor@halborn.com
Manuel García	Halborn	Manuel.Diaz@halborn.com

EXECUTIVE OVERVIEW

1.1 INTRODUCTION

Archimedes Finance engaged Halborn to conduct a security audit on their finance protocol beginning on October 31st, 2022 and ending on November 11th, 2022. The security assessment was scoped to the smart contracts provided in the Archimedes_Finance GitHub repository thisis-archimedes/Archimedes_Finance.

1.2 AUDIT SUMMARY

The team at Halborn was provided three weeks for the engagement and assigned one full-time security engineer to audit the security of the smart contract. The security engineers are blockchain and smart-contract security experts with advanced penetration testing, smart-contract hacking, and deep knowledge of multiple blockchain protocols.

The purpose of this audit to achieve the following:

- Ensure that all functions in the protocol smart contracts are intended.
- Identify potential security issues in Arcade bridge smart contracts.

In summary, Halborn identified many security risks that were mostly addressed by the Archimedes 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 bridge code and can quickly identify items

that do not follow security best practices. The following phases and associated tools were used throughout the term of 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 hotspots 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

The security assessment was scoped to the Solidity smart contracts of the audit branch

1. Solidity Smart Contracts
   (a) Repository: Archimedes Finance
   (b) Commit ID: 8df6cee5354480e5082c7cb753cfe6d52504b616
   (c) Fixed Commit ID: dc38d4936eacff44b2ae1b1feece4d41dbd2748e

Out-of-scope: External libraries and financial related attacks.

- The Proxy contracts that will be used by Archimedes Finance team to deploy the upgradeable contracts were not in the scope of this audit

2. ASSESSMENT SUMMARY & FINDINGS OVERVIEW

CRITICAL	HIGH	MEDIUM	LOW	INFORMATIONAL
0	0	1	3	6

LIKELIHOOD

IMPACT

(HAL-02) (HAL-03) (HAL-04)				
			(HAL-01)	
(HAL-06) (HAL-07) (HAL-08) (HAL-09) (HAL-10)	(HAL-05)			

SECURITY ANALYSIS	RISK LEVEL	REMEDIATION DATE
HAL01 - LACK OF ALLOWANCE CHECK ON LEVERAGE ENGINE CAN LEAD TO OUSD UNDERFLOW	Medium	SOLVED - 11/18/2022
HAL02 - LACK OF DISABLEINITIALIZERS CALL TO PREVENT UNINITIALIZED CONTRACTS	Low	SOLVED - 11/18/2022
HAL03 - LACK OF PARAMETER PRECISION	Low	NOT SOLVED
HAL04 - INCONSISTENT PARAMETER FORMATTING	Low	RISK ACCEPTED
HAL05 - LACK OF 0 ADDRESS CHECK	Informational	SOLVED - 11/18/2022
HAL06 - > 0 IS LESS EFFICIENT THAN != 0 FOR UINTS	Informational	SOLVED - 11/18/2022
HAL07 - REDUNDANT VARIABLE USAGE	Informational	PARTIALLY SOLVED
HAL08 - REDUNDANT FUNCTION CALLS AND CHECKS	Informational	PARTIALLY SOLVED
HAL09 - USE OF POSTFIX OPERATORS RATHER THAN PREFIX OPERATORS	Informational	SOLVED - 11/18/2022
HAL10 - FOR LOOP COULD BE REPLACED BY MULTIPLICATION	Informational	ACKNOWLEDGED

FINDINGS & TECH DETAILS

3.1 (HAL-01) HAL01 - OUSD BEHAVIOUR CAN LEAD TO UNDERFLOW WHEN OPENING A POSITION - MEDIUM

Description:

It has been observed that once a user tries to create any position, the Leverage Engine will try to transfer the OUSD principle to the Leverage Engine. However, due to OUSD behavior, trying to transfer more tokens than allowance will revert with an Underflow error message. Although in recent versions of solidity, this is no longer a critical issue as the value doesn't reset to the maximum value, it is still recommended to verify that the user has enough allowance before transferring the tokens, as the underflow error can be confusing to users.

Code Location:

Listing 1: contracts/LeverageEngine.sol (Line 99)

```
62 /* Non-privileged functions */
63
64 /// @dev deposit OUSD under NFT ID
65 ///
66 /// User sends OUSD to the contract.
67 /// We mint NFT, assign to msg.sender and do the leverage cycles
68 ///
69 /// @param ousdPrinciple the amount of OUSD sent to Archimedes
70 /// @param cycles How many leverage cycles to do
71 /// @param maxArchAmount max amount of Arch tokens to burn for
↳ position
72 function createLeveragedPosition(
73     uint256 ousdPrinciple,
74     uint256 cycles,
75     uint256 maxArchAmount
76 ) external nonReentrant whenNotPaused returns (uint256) {
77     // add some minor buffer to the arch we will use for the
↳ position
78     if (cycles == 0 || cycles > _parameterStore.
↳ getMaxNumberOfCycles()) {
```

```
79          revert("Invalid number of cycles");
80      }
81      // console.log("ousdPrinciple %s", ousdPrinciple);
82      if (ousdPrinciple < _parameterStore.getMinPositionCollateral()
↳ ) {
83          revert("Collateral lower then min");
84      }
85      // console.log("maxArchAmountBufferedDown %s", maxArchAmount);
86      uint256 maxArchAmountBufferedDown = maxArchAmount;
87      uint256 lvUSDAmount = _parameterStore.
↳ getAllowedLeverageForPositionWithArch(ousdPrinciple, cycles,
↳ maxArchAmountBufferedDown);
88      uint256 lvUSDAmountNeedForArguments = _parameterStore.
↳ getAllowedLeverageForPosition(ousdPrinciple, cycles);
89      /// check that user gave enough arch allowance for cycle-
↳ principle combo
90      require(lvUSDAmountNeedForArguments - 1 <= lvUSDAmount, "cant
↳ get enough lvUSD");
91      uint256 archNeededToBurn = (_parameterStore.
↳ calculateArchNeededForLeverage(lvUSDAmount) / 10000) * 10000; //
↳ minus 1000 wei
92      // console.log("archNeededToBurn %s", maxArchAmount);
93
94      require(archNeededToBurn <= maxArchAmountBufferedDown, "Not
↳ enough Arch given for Pos");
95      uint256 availableLev = _coordinator.getAvailableLeverage();
96      require(availableLev >= lvUSDAmount, "Not enough available
↳ lvUSD");
97      _burnArchTokenForPosition(msg.sender, archNeededToBurn);
98      uint256 positionTokenId = _positionToken.safeMint(msg.sender);
99      _ousd.safeTransferFrom(msg.sender, _addressCoordinator,
↳ ousdPrinciple);
100     _coordinator.depositCollateralUnderNFT(positionTokenId,
↳ ousdPrinciple);
101     _coordinator.getLeveragedOUSD(positionTokenId, lvUSDAmount);
102     uint256 psoitionExpireTime = _coordinator.
↳ getPositionExpireTime(positionTokenId);
103
104     emit PositionCreated(msg.sender, positionTokenId,
↳ ousdPrinciple, lvUSDAmount, archNeededToBurn, psoitionExpireTime);
105
106     return positionTokenId;
107 }
```

Proof Of Concept:

- The user calls the createLeveragedPosition() in the LeverageEngine contract having enough OUSD balance but not enough allowance.
- Transaction revert due to integer underflow.

Risk Level:

Likelihood - 4
Impact - 2

Recommendation:

Implementing a require statement validating that the user's allowance for the Leverage Engine is equal to or greater than the OUSD principle is recommended.

Remediation Plan:

SOLVED: The Archimedes Finance team fixed the issue by checking that the OUSD allowance is equal to or greater than the OUSD principle provided in the arguments.

3.2 (HAL-02) LACK OF DISABLEINITIALIZERS CALL TO PREVENT UNINITIALIZED CONTRACTS - LOW

Description:

Multiple contracts are using the `Initializable` module from OpenZeppelin. To prevent leaving an implementation contract uninitialized OpenZeppelin's documentation recommends adding the `_disableInitializers` function in the constructor to lock the contracts automatically when they are deployed:

Code Location:

Listing 2

```
/**
 * @dev Locks the contract, preventing any future reinitialization. This cannot be part of an initializer call.
 * Calling this in the constructor of a contract will prevent that contract from being initialized or reinitialized
 * to any version. It is recommended to use this to lock implementation contracts that are designed to be called
 * through proxies.
 */
function _disableInitializers() internal virtual {
    require(!_initializing, "Initializable: contract is initializing");
    if (_initialized < type(uint8).max) {
        _initialized = type(uint8).max;
        emit Initialized(type(uint8).max);
    }
}
```

Risk Level:

Likelihood - 1
Impact - 3

Recommendation:

Consider calling the _disableInitializers function in the TimeLockControllerUpgradeable contract constructor as well:

Listing 3

```
/// @custom:oz-upgrades-unsafe-allow constructor
constructor() {
    _disableInitializers();
}
```

Remediation Plan:

SOLVED: The Archimedes Finance team fixed the issue by calling the _disableInitializers function in the contract constructor.

3.3 (HAL-03) LACK OF PARAMETER PRECISION - LOW

Description:

It has been detected that some parameters used in the different contracts are set and used in a way that lets the minimum settable value at 1%.

Code Location:

Listing 4: ParameterStore.sol (Line 40)

```
34          _maxNumberOfCycles = 10;
35          _originationFeeRate = 5 ether / 100;
36          _globalCollateralRate = 90;
37          _rebaseFeeRate = 10 ether / 100; // meaning 10%
38          _treasuryAddress;
39          _curveGuardPercentage = 90;
40          _slippage = 2; // 2%;
41          _archToLevRatio = 1 ether; // meaning 1 arch is equal 1
↳ lvUSD
42          _curveMaxExchangeGuard = 50; // meaning we allow exchange
↳ with get 50% more then we expected
43          _treasuryAddress = address(0);
```

For example, _slippage is set to 2, implying that a maximum slippage of 2% is allowed. However, if this value needs to be modified in the future or if user-controlled slippage is implemented, it would only accept integer values, such as 0%, 1%, 2%, etc.

Risk Level:

Likelihood - 1
Impact - 3

Recommendation:

Consider increasing the digits in the % calculations to allow more flexibility when using these parameters.

Remediation Plan:

NOT SOLVED: The Archimedes Finance team did not solve the issue.

3.4 (HAL-04) INCONSISTENT PARAMETER FORMATTING - LOW

Description:

ParameterStore.sol contract stores global parameters used when managing positions, such as _maxNumberOfCycles, _originationFeeRate, etc.

However, it has been detected that parameters with a similar format or that are used in similar ways (like percentages, for example) are defined and used differently, which could cause unexpected contract behavior (or even worse situations such as fund loss) if they are mistakenly modified.

Code Location:

Listing 5: ParameterStore.sol (Lines 36,37)

```
34          _maxNumberOfCycles = 10;
35          _originationFeeRate = 5 ether / 100;
36          _globalCollateralRate = 90;
37          _rebaseFeeRate = 10 ether / 100; // meaning 10%
38          _treasuryAddress;
39          _curveGuardPercentage = 90;
40          _slippage = 2; // 2%;
41          _archToLevRatio = 1 ether; // meaning 1 arch is equal 1
↳ lvUSD
42          _curveMaxExchangeGuard = 50; // meaning we allow exchange
↳ with get 50% more then we expected
43          _treasuryAddress = address(0);
```

In this example can be seen how two different rates (with 90% and 10% values, respectively) are calculated differently.

Risk Level:

Likelihood - 1
Impact - 3

Recommendation:

It is strongly recommended to unify the parameter criteria in a way that allows contract administrators to set parameters of the same nature in the same way.

Remediation Plan:

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

3.5 (HAL-05) LACK OF 0 ADDRESS CHECK - INFORMATIONAL

Description:

When setting the dependencies for the contract through the setDependencies () functions, the 0 address is not being checked in any of the contracts.

Code Location:

Listing 6: contracts/LeverageEngine.sol

```
43 function setDependencies(
44     address addressCoordinator,
45     address addressPositionToken,
46     address addressParameterStore,
47     address addressArchToken,
48     address addressOUSD
49 ) external nonReentrant onlyAdmin {
50     _addressCoordinator = addressCoordinator;
51     _coordinator = ICoordinator(addressCoordinator);
52     _addressPositionToken = addressPositionToken;
53     _positionToken = PositionToken(addressPositionToken);
54     _addressParameterStore = addressParameterStore;
55     _parameterStore = ParameterStore(addressParameterStore);
56     _addressArchToken = addressArchToken;
57     _archToken = ArchToken(addressArchToken);
58     _addressOUSD = addressOUSD;
59     _ousd = IERC20Upgradeable(_addressOUSD);
60 }
61
```

Risk Level:

Likelihood - 2
Impact - 1

Recommendation:

When setting an address variable, always make sure the value is not zero.

Remediation Plan:

SOLVED: The Archimedes Finance team fixed the issue by adding a zero address check in the setDependencies() functions.

3.6 (HAL-06) > 0 IS LESS EFFICIENT THAN != 0 FOR UINTS - INFORMATIONAL

Description:

The use of `>` consumes more gas than `!=`. There are some cases where both can be used indistinctly, such as in unsigned integers where numbers can't be negative, and as such, there is only a need to check that a number is not 0.

Code Location:

CDPosition.sol
- Line 40: `require(_nftCDP[nftID].oUSDPrinciple > 0, "NFT ID must exist ");`

Coordinator.sol
- Line 143: `require(numberOfSharesInPosition > 0, "Position has no shares");`

ParameterStore.sol
- Line 63: `require(newMaxNumberOfCycles < 20 && newMaxNumberOfCycles > 0, "New max n of cycles out of range");`
- Line 81: `require(newGlobalCollateralRate <= 100 && newGlobalCollateralRate > 0, "New collateral rate out of range");`
- Line 87: `require(newMaxNumberOfCycles < 20 && newMaxNumberOfCycles > 0, "New max n of cycles out of range");`

Risk Level:

Likelihood - 1
Impact - 1

Recommendation:

Use `!=` instead of `>` in cases where both can be used.

Remediation Plan:

SOLVED: The Archimedes Finance team fixed the issue by replacing `>` with `!=`.

3.7 (HAL-07) REDUNDANT VARIABLE USAGE - INFORMATIONAL

Description:

Redundant variable assignments have been found through the code. These statements increase gas costs for contract deploying and interactions, and impact code readability.

Code Location:

In this example, maxArchAmountBufferedDown is assigned with the value from the maxArchAmount parameter instead of using the parameter variable directly.

Listing 7: contracts/LeverageEngine.sol (Line 72)

```
72 function createLeveragedPosition(
73     uint256 ousdPrinciple,
74     uint256 cycles,
75     uint256 maxArchAmount
76 ) external nonReentrant whenNotPaused returns (uint256) {
77     // add some minor buffer to the arch we will use for the
↳ position
78     if (cycles == 0 || cycles > _parameterStore.
↳ getMaxNumberOfCycles()) {
79         revert("Invalid number of cycles");
80     }
81     // console.log("ousdPrinciple %s", ousdPrinciple);
82     if (ousdPrinciple < _parameterStore.getMinPositionCollateral()
↳ ) {
83         revert("Collateral lower then min");
84     }
85     // console.log("maxArchAmountBufferedDown %s", maxArchAmount);
86     uint256 maxArchAmountBufferedDown = maxArchAmount;
```

Moreover, in this other example cyclePrinciple is assigned with the value from the principle parameter.

Listing 8: contracts/ParameterStore.sol (Line 171)

```
171 function getAllowedLeverageForPosition(uint256 principle, uint256
 ↳ numberOfCycles) public view returns (uint256) {
172     require(numberOfCycles <= _maxNumberOfCycles, "Cycles greater
 ↳ than max allowed");
173     uint256 leverageAmount = 0;
174     uint256 cyclePrinciple = principle;
```

Risk Level:

Likelihood - 1
Impact - 1

Recommendation:

Redundant variable assignments should be deleted to improve code readability and usability and reduce gas costs when deploying or interacting with the contract.

Remediation Plan:

PARTIALLY SOLVED: The Archimedes Finance team fixed the issue in some instances, but it has been partially solved.

3.8 (HAL-08) REDUNDANT FUNCTIONS CALLS AND CHECKS - INFORMATIONAL

Description:

While requirements are needed for proper usage and security of the contracts, some redundant validations have been seen.

Code Location:

In this instance, the getAllowedLeverageForPositionWithArch function will already call and return the getAllowedLeverageForPosition function return value. Moreover, this function will already revert if not enough ARCH is provided. Making the require statements in lines 90 and 94 redundant.

Listing 9: contracts/LeverageEngine.sol (Lines 87,88,90,94)

```
72 function createLeveragedPosition(
73         uint256 ousdPrinciple,
74         uint256 cycles,
75         uint256 maxArchAmount
76     ) external nonReentrant whenNotPaused returns (uint256) {
77         // add some minor buffer to the arch we will use for the
↳ position
78         if (cycles == 0 || cycles > _parameterStore.
↳ getMaxNumberOfCycles()) {
79             revert("Invalid number of cycles");
80         }
81         // console.log("ousdPrinciple %s", ousdPrinciple);
82         if (ousdPrinciple < _parameterStore.
↳ getMinPositionCollateral()) {
83             revert("Collateral lower then min");
84         }
85         // console.log("maxArchAmountBufferedDown %s",
↳ maxArchAmount);
86         uint256 maxArchAmountBufferedDown = maxArchAmount;
87         uint256 lvUSDAmount = _parameterStore.
↳ getAllowedLeverageForPositionWithArch(ousdPrinciple, cycles,
↳ maxArchAmountBufferedDown);
```

```
88          uint256 lvUSDAmountNeedForArguments = _parameterStore.
↳ getAllowedLeverageForPosition(ousdPrinciple, cycles);
89          /// check that user gave enough arch allowance for cycle-
↳ principle combo
90          require(lvUSDAmountNeedForArguments - 1 <= lvUSDAmount, "
↳ cant get enough lvUSD");
91          uint256 archNeededToBurn = (_parameterStore.
↳ calculateArchNeededForLeverage(lvUSDAmount) / 10000) * 10000; //
↳ minus 1000 wei
92          // console.log("archNeededToBurn %s", maxArchAmount);
93
94          require(archNeededToBurn <= maxArchAmountBufferedDown, "
↳ Not enough Arch given for Pos");
95          uint256 availableLev = _coordinator.getAvailableLeverage()
↳ ;
96          require(availableLev >= lvUSDAmount, "Not enough available
↳  lvUSD");
97          _burnArchTokenForPosition(msg.sender, archNeededToBurn);
98          uint256 positionTokenId = _positionToken.safeMint(msg.
↳ sender);
99          _ousd.safeTransferFrom(msg.sender, _addressCoordinator,
↳ ousdPrinciple);
100         _coordinator.depositCollateralUnderNFT(positionTokenId,
↳ ousdPrinciple);
101         _coordinator.getLeveragedOUSD(positionTokenId, lvUSDAmount
↳ );
102         uint256 psoitionExpireTime = _coordinator.
↳ getPositionExpireTime(positionTokenId);
103
104         emit PositionCreated(msg.sender, positionTokenId,
↳ ousdPrinciple, lvUSDAmount, archNeededToBurn, psoitionExpireTime);
105
106         return positionTokenId;
107     }
```

Risk Level:

Likelihood - 1

Impact - 1

Recommendation:

Avoid calling the same function twice unnecessarily.

Remediation Plan:

PARTIALLY SOLVED: The Archimedes Finance team fixed the issue in some instances, but it has been partially solved.

3.9 (HAL-09) USE OF POSTFIX OPERATORS RATHER THAN PREFIX OPERATORS - INFORMATIONAL

Description:

The use of postfix operators `i++` consume more gas than prefix operators `++i`.

Code Location:

ParameterStore.sol
- Line 176: `for (uint256 i = 0; i < numberOfCycles; ++i){`

PositionToken.sol
- Line 88: `for (uint256 i = 0; i < _addressToTokensOwnedMapping[from].length; i++){`

Risk Level:

Likelihood - 1
Impact - 1

Recommendation:

It is recommended to use prefix operators rather than postfix.

Remediation Plan:

SOLVED: The Archimedes Finance team fixed the issue.

3.10 (HAL-10) FOR LOOP COULD BE REPLACE BY MULTIPLICATION - INFORMATIONAL

Description:

For loops are especially useful in functions where some complex logic is involved in each loop. However, in cases where the for loop will only add a constant value to a variable, it is recommended to use multiplication instead, as it will save gas.

Code Location:

For example, the loop in the getAllowedLeverageForPosition function:

Listing 10: contracts/ParameterStore.sol (Line 179)

```
171 function getAllowedLeverageForPosition(uint256 principle, uint256
 ↳ numberOfCycles) public view returns (uint256) {
172     require(numberOfCycles <= _maxNumberOfCycles, "Cycles greater
 ↳ than max allowed");
173     uint256 leverageAmount = 0;
174     uint256 cyclePrinciple = principle;
175     // console.log("getAllowedLeverageForPosition principle %s,
 ↳ numberOfCycles %s", principle / 1 ether, numberOfCycles);
176     for (uint256 i = 0; i < numberOfCycles; ++i) {
177         // console.log("getAllowedLeverageForPosition looping on
 ↳ cycles");
178         cyclePrinciple = (cyclePrinciple * _globalCollateralRate)
 ↳ / 100;
179         leverageAmount += cyclePrinciple;
180     }
181     // console.log("getAllowedLeverageForPosition: leverageAmount
 ↳ %s", leverageAmount / 1 ether);
182     return leverageAmount;
183 }
184
```

Risk Level:

Likelihood - 1
Impact - 1

Recommendation:

It is recommended to replace this loop by a multiplication.

Remediation Plan:

ACKNOWLEDGE: The Archimedes Finance team acknowledged the issue.

AUTOMATED TESTING

4.1 STATIC ANALYSIS REPORT

Description:

Halborn used automated testing techniques to enhance coverage of certain areas of the scoped contract. Among the tools used was Slither, a Solidity static analysis framework. After Halborn verified all the contracts in the repository and was able to compile them correctly into their ABI and binary formats. 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:

- AccessController.sol

```
AddressUpgradeable.verifyCallResult(bool,bytes,string) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/AddressUpgradeable.sol#174-194) uses assembly
        - INLINE ASM (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/AddressUpgradeable.sol#186-189)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#assembly-usage

Different versions of Solidity are used:
        - Version used: ['0.8.13', '^0.8.0', '^0.8.1', '^0.8.2']
        - ^0.8.0 (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/access/AccessControlUpgradeable.sol#4)
        - ^0.8.0 (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/access/IAccessControlUpgradeable.sol#4)
        - ^0.8.2 (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/proxy/utils/Initializable.sol#4)
        - ^0.8.1 (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/AddressUpgradeable.sol#4)
        - ^0.8.0 (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/ContextUpgradeable.sol#4)
        - ^0.8.0 (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/StringsUpgradeable.sol#4)
        - ^0.8.0 (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/introspection/ERC165Upgradeable.sol#4)
        - ^0.8.0 (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/introspection/IERC165Upgradeable.sol#4)
        - 0.8.13 (jobs/archimedes/brownie/contracts/AccessController.sol#2)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#different-pragma-directives-are-used

AccessControlUpgradeable.__AccessControl_init() (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/access/AccessControlUpgradeable.sol#51-52) is never used and should be removed
AccessControlUpgradeable.__AccessControl_init_unchained() (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/access/AccessControlUpgradeable.sol#54-55) is never used and should be removed
AccessControlUpgradeable._setRoleAdmin(bytes32,bytes32) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/access/AccessControlUpgradeable.sol#220-224) is never used and should be removed
AccessControlUpgradeable._setupRole(bytes32,address) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/access/AccessControlUpgradeable.sol#213-213) is never used and should be removed
AccessController._requireAdmin() (jobs/archimedes/brownie/contracts/AccessController.sol#107-109) is never used and should be removed
AddressUpgradeable.functionCall(address,bytes) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/AddressUpgradeable.sol#85-87) is never used and should be removed
AddressUpgradeable.functionCall(address,bytes,string) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/AddressUpgradeable.sol#95-101) is never used and should be removed
AddressUpgradeable.functionCallWithValue(address,bytes,uint256) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/AddressUpgradeable.sol#114-120) is never used and should be removed
AddressUpgradeable.functionCallWithValue(address,bytes,uint256,string) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/AddressUpgradeable.sol#126-139) is never used and should be removed
AddressUpgradeable.functionStaticCall(address,bytes) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/AddressUpgradeable.sol#147-149) is never used and should be removed
AddressUpgradeable.functionStaticCall(address,bytes,string) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/AddressUpgradeable.sol#157-166) is never used and should be removed
AddressUpgradeable.isContract(address) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/AddressUpgradeable.sol#36-42) is never used and should be removed
AddressUpgradeable.sendValue(address,uint256) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/AddressUpgradeable.sol#60-65) is never used and should be removed
AddressUpgradeable.verifyCallResult(bool,bytes,string) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/AddressUpgradeable.sol#174-194) is never used and should be removed
ContextUpgradeable.__Context_init() (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/ContextUpgradeable.sol#18-19) is never used and should be removed
ContextUpgradeable.__Context_init_unchained() (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/ContextUpgradeable.sol#21-22) is never used and should be removed
ContextUpgradeable._msgData() (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/ContextUpgradeable.sol#27-29) is never used and should be removed
ERC165Upgradeable.__ERC165_init() (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/introspection/ERC165Upgradeable.sol#24-25) is never used and should be removed
ERC165Upgradeable.__ERC165_init_unchained() (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/introspection/ERC165Upgradeable.sol#27-28) is never used and should be removed
Initializable._disableInitializers() (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/proxy/utils/Initializable.sol#131-137) is never used and should be removed
StringsUpgradeable.toHexString(address) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/StringsUpgradeable.sol#72-74) is never used and should be removed
StringsUpgradeable.toHexString(uint256) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/StringsUpgradeable.sol#41-52) is never used and should be removed
StringsUpgradeable.toString(uint256) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/StringsUpgradeable.sol#16-36) is never used and should be removed
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#dead-code

Pragma version^0.8.0 (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/access/AccessControlUpgradeable.sol#4) allows old versions
Pragma version^0.8.0 (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/access/IAccessControlUpgradeable.sol#4) allows old versions
Pragma version^0.8.2 (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/proxy/utils/Initializable.sol#4) allows old versions
Pragma version^0.8.1 (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/AddressUpgradeable.sol#4) allows old versions
Pragma version^0.8.0 (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/ContextUpgradeable.sol#4) allows old versions
Pragma version^0.8.0 (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/StringsUpgradeable.sol#4) allows old versions
Pragma version^0.8.0 (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/introspection/ERC165Upgradeable.sol#4) allows old versions
Pragma version^0.8.0 (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/introspection/IERC165Upgradeable.sol#4) allows old versions
Pragma version0.8.13 (jobs/archimedes/brownie/contracts/AccessController.sol#2) necessitates a version too recent to be trusted. Consider deploying with 0.6.12/0.7.6/0.8.7
solc-0.8.13 is not recommended for deployment
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#incorrect-versions-of-solidity

Pragma version^0.8.0 (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/access/AccessControlUpgradeable.sol#4) allows old versions
Pragma version^0.8.0 (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/access/IAccessControlUpgradeable.sol#4) allows old versions
Pragma version^0.8.2 (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/proxy/utils/Initializable.sol#4) allows old versions
Pragma version^0.8.1 (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/AddressUpgradeable.sol#4) allows old versions
Pragma version^0.8.0 (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/ContextUpgradeable.sol#4) allows old versions
Pragma version^0.8.0 (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/StringsUpgradeable.sol#4) allows old versions
Pragma version^0.8.0 (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/introspection/ERC165Upgradeable.sol#4) allows old versions
Pragma version^0.8.0 (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/introspection/IERC165Upgradeable.sol#4) allows old versions
Pragma version0.8.13 (jobs/archimedes/brownie/contracts/AccessController.sol#2) necessitates a version too recent to be trusted. Consider deploying with 0.6.12/0.7.6/0.8.7
solc-0.8.13 is not recommended for deployment
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#incorrect-versions-of-solidity

Low level call in AddressUpgradeable.sendValue(address,uint256) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/AddressUpgradeable.sol#60-65):
        - (success) = recipient.call{value: amount}() (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/AddressUpgradeable.sol#63)
Low level call in AddressUpgradeable.functionCallWithValue(address,bytes,uint256,string) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/AddressUpgradeable.sol#120-139):
        - (success,returndata) = target.call{value: value}(data) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/AddressUpgradeable.sol#137)
Low level call in AddressUpgradeable.functionStaticCall(address,bytes,string) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/AddressUpgradeable.sol#157-166):
        - (success,returndata) = target.staticcall(data) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/AddressUpgradeable.sol#164)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#low-level-calls

Function AccessControlUpgradeable.__AccessControl_init() (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/access/AccessControlUpgradeable.sol#51-52) is not in mixedCase
Function AccessControlUpgradeable.__AccessControl_init_unchained() (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/access/AccessControlUpgradeable.sol#54-55) is not in mixedCase
Variable AccessControlUpgradeable.__gap (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/access/AccessControlUpgradeable.sol#259) is not in mixedCase
Function ContextUpgradeable.__Context_init() (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/ContextUpgradeable.sol#18-19) is not in mixedCase
Function ContextUpgradeable.__Context_init_unchained() (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/ContextUpgradeable.sol#21-22) is not in mixedCase
Variable ContextUpgradeable.__gap (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/ContextUpgradeable.sol#36) is not in mixedCase
Function ERC165Upgradeable.__ERC165_init() (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/introspection/ERC165Upgradeable.sol#24-25) is not in mixedCase
Function ERC165Upgradeable.__ERC165_init_unchained() (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/introspection/ERC165Upgradeable.sol#27-28) is not in mixedCase
Variable ERC165Upgradeable.__gap (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/introspection/ERC165Upgradeable.sol#41) is not in mixedCase
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#conformance-to-solidity-naming-conventions

AccessControlUpgradeable.__gap (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/access/AccessControlUpgradeable.sol#259) is never used in AccessController (jobs/archimedes/brownie/contracts/AccessController.sol#8-110)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#unused-state-variable

AccessControlUpgradeable.__gap (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/access/AccessControlUpgradeable.sol#259) is never used in AccessController (jobs/archimedes/brownie/contracts/AccessController.sol#8-110)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#unused-state-variable

grantRole(bytes32,address) should be declared external:
        - AccessControlUpgradeable.grantRole(bytes32,address) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/access/AccessControlUpgradeable.sol#150-152)
revokeRole(bytes32,address) should be declared external:
        - AccessControlUpgradeable.revokeRole(bytes32,address) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/access/AccessControlUpgradeable.sol#165-167)
renounceRole(bytes32,address) should be declared external:
        - AccessControlUpgradeable.renounceRole(bytes32,address) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/access/AccessControlUpgradeable.sol#185-189)
        - AccessController.renounceRole(bytes32,address) (jobs/archimedes/brownie/contracts/AccessController.sol#62-69)
setAdmin(address) should be declared external:
        - AccessController.setAdmin(address) (jobs/archimedes/brownie/contracts/AccessController.sol#41-47)
setGovernor(address) should be declared external:
        - AccessController.setGovernor(address) (jobs/archimedes/brownie/contracts/AccessController.sol#71-77)
setExecutive(address) should be declared external:
        - AccessController.setExecutive(address) (jobs/archimedes/brownie/contracts/AccessController.sol#79-85)
setGuardian(address) should be declared external:
        - AccessController.setGuardian(address) (jobs/archimedes/brownie/contracts/AccessController.sol#87-93)
getAddressExecutive() should be declared external:
        - AccessController.getAddressExecutive() (jobs/archimedes/brownie/contracts/AccessController.sol#95-97)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#public-function-that-could-be-declared-external
```

AUTOMATED TESTING

AUTOMATED TESTING

- ArchToken.sol

```
Math.mulDiv(uint256,uint256,uint256) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts/utils/math/Math.sol#55-135) performs a multiplication on the result of a division:
        -denominator = denominator / twos (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts/utils/math/Math.sol#102)
        -inverse = (3 * denominator) ^ 2 (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts/utils/math/Math.sol#117)
Math.mulDiv(uint256,uint256,uint256) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts/utils/math/Math.sol#55-135) performs a multiplication on the result of a division:
        -denominator = denominator / twos (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts/utils/math/Math.sol#102)
        -inverse *= 2 - denominator * inverse (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts/utils/math/Math.sol#121)
Math.mulDiv(uint256,uint256,uint256) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts/utils/math/Math.sol#55-135) performs a multiplication on the result of a division:
        -denominator = denominator / twos (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts/utils/math/Math.sol#102)
        -inverse *= 2 - denominator * inverse (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts/utils/math/Math.sol#122)
Math.mulDiv(uint256,uint256,uint256) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts/utils/math/Math.sol#55-135) performs a multiplication on the result of a division:
        -denominator = denominator / twos (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts/utils/math/Math.sol#102)
        -inverse *= 2 - denominator * inverse (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts/utils/math/Math.sol#123)
Math.mulDiv(uint256,uint256,uint256) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts/utils/math/Math.sol#55-135) performs a multiplication on the result of a division:
        -denominator = denominator / twos (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts/utils/math/Math.sol#102)
        -inverse *= 2 - denominator * inverse (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts/utils/math/Math.sol#124)
Math.mulDiv(uint256,uint256,uint256) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts/utils/math/Math.sol#55-135) performs a multiplication on the result of a division:
        -denominator = denominator / twos (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts/utils/math/Math.sol#102)
        -inverse *= 2 - denominator * inverse (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts/utils/math/Math.sol#125)
Math.mulDiv(uint256,uint256,uint256) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts/utils/math/Math.sol#55-135) performs a multiplication on the result of a division:
        -denominator = denominator / twos (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts/utils/math/Math.sol#102)
        -inverse *= 2 - denominator * inverse (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts/utils/math/Math.sol#126)
Math.mulDiv(uint256,uint256,uint256) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts/utils/math/Math.sol#55-135) performs a multiplication on the result of a division:
        -prod0 = prod0 / twos (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts/utils/math/Math.sol#105)
        -result = prod0 * inverse (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts/utils/math/Math.sol#132)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#divide-before-multiply

ERC20Votes._writeCheckpoint(ERC20Votes.Checkpoint[],function(uint256,uint256) returns(uint256),uint256) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts/token/ERC20/extensions/ERC20Votes.sol#228-240) uses a dangerous strict equality:
        - pos > 0 && ckpts[pos - 1].fromBlock == block.number (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts/token/ERC20/extensions/ERC20Votes.sol#235)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#dangerous-strict-equalities

ERC20Votes._moveVotingPower(address,address,uint256).newWeight_scope_1 (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts/token/ERC20/extensions/ERC20Votes.sol#220) is a local variable never initialized
ERC20Votes._moveVotingPower(address,address,uint256).oldWeight_scope_0 (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts/token/ERC20/extensions/ERC20Votes.sol#220) is a local variable never initialized
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#uninitialized-local-variables

ERC20Permit.constructor(string).name (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts/token/ERC20/extensions/draft-ERC20Permit.sol#44) shadows:
        - ERC20.name() (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts/token/ERC20/ERC20.sol#62-64) (function)
        - IERC20Metadata.name() (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts/token/ERC20/extensions/IERC20Metadata.sol#17) (function)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#local-variable-shadowing

Variable 'ERC20Votes._moveVotingPower(address,address,uint256).newWeight (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts/token/ERC20/extensions/ERC20Votes.sol#215)' in ERC20Votes._moveVotingPower(address,address,uint256) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts/token/ERC20/extensions/ERC20Votes.sol#208-224) potentially used before declaration: (oldWeight,newWeight) = _writeCheckpoint(_checkpoints[dst],_add,amount) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts/token/ERC20/extensions/ERC20Votes.sol#220)
Variable 'ERC20Votes._moveVotingPower(address,address,uint256).oldWeight (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts/token/ERC20/extensions/ERC20Votes.sol#215)' in ERC20Votes._moveVotingPower(address,address,uint256) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts/token/ERC20/extensions/ERC20Votes.sol#208-224) potentially used before declaration: (oldWeight,newWeight) = _writeCheckpoint(_checkpoints[dst],_add,amount) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts/token/ERC20/extensions/ERC20Votes.sol#220)
Variable 'ECDSA.tryRecover(bytes32,bytes).r (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts/utils/cryptography/ECDSA.sol#62)' in ECDSA.tryRecover(bytes32,bytes) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts/utils/cryptography/ECDSA.sol#57-88) potentially used before declaration: r = mload(uint256)(signature + 0x20) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts/utils/cryptography/ECDSA.sol#81)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#pre-declaration-usage-of-local-variables

ERC20Votes.delegateBySig(address,uint256,uint256,uint8,bytes32,bytes32) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts/token/ERC20/extensions/ERC20Votes.sol#133-150) uses timestamp for comparisons
        Dangerous comparisons:
        - require(bool,string)(block.timestamp <= expiry,ERC20Votes: signature expired) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts/token/ERC20/extensions/ERC20Votes.sol#141)
ERC20Permit.permit(address,address,uint256,uint256,uint8,bytes32,bytes32) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts/token/ERC20/extensions/draft-ERC20Permit.sol#49-68) uses timestamp for comparisons
        Dangerous comparisons:
        - require(bool,string)(block.timestamp <= deadline,ERC20Permit: expired deadline) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts/token/ERC20/extensions/draft-ERC20Permit.sol#58)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#block-timestamp

ECDSA.tryRecover(bytes32,bytes) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts/utils/cryptography/ECDSA.sol#57-88) uses assembly
        - INLINE ASM (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts/utils/cryptography/ECDSA.sol#68-72)
        - INLINE ASM (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts/utils/cryptography/ECDSA.sol#80-83)
Math.mulDiv(uint256,uint256,uint256) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts/utils/math/Math.sol#55-135) uses assembly
        - INLINE ASM (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts/utils/math/Math.sol#66-70)
        - INLINE ASM (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts/utils/math/Math.sol#86-90)
        - INLINE ASM (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts/utils/math/Math.sol#100-109)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#assembly-usage

Different versions of Solidity are used:
        - Version used: ['^0.8.13', '^0.8.0']
        - ^0.8.0 (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts/access/AccessControl.sol#4)
        - ^0.8.0 (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts/access/IAccessControl.sol#4)
        - ^0.8.0 (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts/governance/utils/IVotes.sol#3)
        - ^0.8.0 (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts/token/ERC20/ERC20.sol#4)
        - ^0.8.0 (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts/token/ERC20/IERC20.sol#4)
        - ^0.8.0 (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts/token/ERC20/extensions/ERC20Votes.sol#4)
        - ^0.8.0 (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts/token/ERC20/extensions/IERC20Metadata.sol#4)
        - ^0.8.0 (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts/token/ERC20/extensions/draft-ERC20Permit.sol#4)
        - ^0.8.0 (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts/token/ERC20/extensions/draft-IERC20Permit.sol#4)
        - ^0.8.0 (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts/utils/Context.sol#4)
        - ^0.8.0 (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts/utils/Counters.sol#4)
        - ^0.8.0 (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts/utils/Strings.sol#4)
        - ^0.8.0 (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts/utils/cryptography/ECDSA.sol#4)
        - ^0.8.0 (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts/utils/cryptography/draft-EIP712.sol#4)
        - ^0.8.0 (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts/utils/introspection/ERC165.sol#4)
        - ^0.8.0 (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts/utils/introspection/IERC165.sol#4)
        - ^0.8.0 (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts/utils/math/Math.sol#4)
        - ^0.8.0 (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts/utils/math/SafeCast.sol#4)
        - 0.8.13 (jobs/archimedes/brownie/contracts/ArchToken.sol#2)
        - 0.8.13 (jobs/archimedes/brownie/contracts/BasicAccessController.sol#2)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#different-pragma-directives-are-used

AccessControl._setRoleAdmin(bytes32,bytes32) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts/access/AccessControl.sol#214-218) is never used and should be removed
AccessControl._setupRole(bytes32,address) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts/access/AccessControl.sol#205-207) is never used and should be removed
ArchToken._burn(address,uint256) (jobs/archimedes/brownie/contracts/ArchToken.sol#33-35) is never used and should be removed
BasicAccessController._requireAdmin() (jobs/archimedes/brownie/contracts/BasicAccessController.sol#67-69) is never used and should be removed
Context._msgData() (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts/utils/Context.sol#21-23) is never used and should be removed
Counters.decrement(Counters.Counter) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts/utils/Counters.sol#32-36) is never used and should be removed
Counters.reset(Counters.Counter) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts/utils/Counters.sol#40-42) is never used and should be removed
ECDSA.recover(bytes32,bytes) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts/utils/cryptography/ECDSA.sol#104-108) is never used and should be removed
ECDSA.recover(bytes32,bytes32,bytes32) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts/utils/cryptography/ECDSA.sol#132-140) is never used and should be removed
ECDSA.toEthSignedMessageHash(bytes) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts/utils/cryptography/ECDSA.sol#216-218) is never used and should be removed
ECDSA.toEthSignedMessageHash(bytes32) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts/utils/cryptography/ECDSA.sol#202-206) is never used and should be removed
ECDSA.tryRecover(bytes32,bytes) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts/utils/cryptography/ECDSA.sol#57-88) is never used and should be removed
ECDSA.tryRecover(bytes32,bytes32,bytes32) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts/utils/cryptography/ECDSA.sol#117-125) is never used and should be removed
ERC20._burn(address,uint256) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts/token/ERC20/ERC20.sol#280-295) is never used and should be removed
ERC20Votes._add(uint256,uint256) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts/token/ERC20/extensions/ERC20Votes.sol#242-244) is never used and should be removed
ERC20Votes._burn(address,uint256) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts/token/ERC20/extensions/ERC20Votes.sol#172-176) is never used and should be removed
ERC20Votes._subtract(uint256,uint256) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts/token/ERC20/extensions/ERC20Votes.sol#246-248) is never used and should be removed
Math.ceilDiv(uint256,uint256) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts/utils/math/Math.sol#45-48) is never used and should be removed
Math.max(uint256,uint256) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts/utils/math/Math.sol#19-21) is never used and should be removed
Math.min(uint256,uint256) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts/utils/math/Math.sol#26-28) is never used and should be removed
Math.mulDiv(uint256,uint256,uint256) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts/utils/math/Math.sol#55-135) is never used and should be removed
Math.mulDiv(uint256,uint256,uint256,Math.Rounding) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts/utils/math/Math.sol#140-151) is never used and should be removed
Math.sqrt(uint256) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts/utils/math/Math.sol#158-214) is never used and should be removed
Math.sqrt(uint256,Math.Rounding) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts/utils/math/Math.sol#219-225) is never used and should be removed
SafeCast.toInt104(int256) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts/utils/math/SafeCast.sol#900-903) is never used and should be removed
SafeCast.toInt112(int256) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts/utils/math/SafeCast.sol#882-885) is never used and should be removed
SafeCast.toInt120(int256) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts/utils/math/SafeCast.sol#864-867) is never used and should be removed
SafeCast.toInt128(int256) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts/utils/math/SafeCast.sol#846-849) is never used and should be removed
SafeCast.toInt136(int256) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts/utils/math/SafeCast.sol#828-831) is never used and should be removed
SafeCast.toInt144(int256) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts/utils/math/SafeCast.sol#810-813) is never used and should be removed
SafeCast.toInt152(int256) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts/utils/math/SafeCast.sol#792-795) is never used and should be removed
SafeCast.toInt16(int256) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts/utils/math/SafeCast.sol#1098-1101) is never used and should be removed
SafeCast.toInt160(int256) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts/utils/math/SafeCast.sol#774-777) is never used and should be removed
SafeCast.toInt168(int256) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts/utils/math/SafeCast.sol#756-759) is never used and should be removed
SafeCast.toInt176(int256) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts/utils/math/SafeCast.sol#738-741) is never used and should be removed
SafeCast.toInt184(int256) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts/utils/math/SafeCast.sol#720-723) is never used and should be removed
SafeCast.toInt192(int256) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts/utils/math/SafeCast.sol#702-705) is never used and should be removed
SafeCast.toInt200(int256) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts/utils/math/SafeCast.sol#684-687) is never used and should be removed
SafeCast.toInt208(int256) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts/utils/math/SafeCast.sol#666-669) is never used and should be removed
SafeCast.toInt216(int256) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts/utils/math/SafeCast.sol#648-651) is never used and should be removed
SafeCast.toInt224(int256) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts/utils/math/SafeCast.sol#630-633) is never used and should be removed
SafeCast.toInt232(int256) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts/utils/math/SafeCast.sol#612-615) is never used and should be removed
SafeCast.toInt24(int256) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts/utils/math/SafeCast.sol#1080-1083) is never used and should be removed
SafeCast.toInt240(int256) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts/utils/math/SafeCast.sol#594-597) is never used and should be removed
SafeCast.toInt248(int256) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts/utils/math/SafeCast.sol#576-579) is never used and should be removed
SafeCast.toInt256(uint256) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts/utils/math/SafeCast.sol#1130-1134) is never used and should be removed
SafeCast.toInt32(int256) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts/utils/math/SafeCast.sol#1062-1065) is never used and should be removed
SafeCast.toInt40(int256) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts/utils/math/SafeCast.sol#1044-1047) is never used and should be removed
SafeCast.toInt48(int256) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts/utils/math/SafeCast.sol#1026-1029) is never used and should be removed
SafeCast.toInt56(int256) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts/utils/math/SafeCast.sol#1008-1011) is never used and should be removed
SafeCast.toInt64(int256) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts/utils/math/SafeCast.sol#990-993) is never used and should be removed
SafeCast.toInt72(int256) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts/utils/math/SafeCast.sol#972-975) is never used and should be removed
SafeCast.toInt8(int256) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts/utils/math/SafeCast.sol#1116-1119) is never used and should be removed
SafeCast.toInt80(int256) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts/utils/math/SafeCast.sol#954-957) is never used and should be removed
SafeCast.toInt88(int256) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts/utils/math/SafeCast.sol#936-939) is never used and should be removed
SafeCast.toInt96(int256) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts/utils/math/SafeCast.sol#918-921) is never used and should be removed
SafeCast.toUint104(uint256) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts/utils/math/SafeCast.sol#340-343) is never used and should be removed
SafeCast.toUint112(uint256) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts/utils/math/SafeCast.sol#323-326) is never used and should be removed
SafeCast.toUint120(uint256) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts/utils/math/SafeCast.sol#306-309) is never used and should be removed
SafeCast.toUint128(uint256) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts/utils/math/SafeCast.sol#289-292) is never used and should be removed
SafeCast.toUint136(uint256) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts/utils/math/SafeCast.sol#272-275) is never used and should be removed
SafeCast.toUint144(uint256) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts/utils/math/SafeCast.sol#255-258) is never used and should be removed
SafeCast.toUint152(uint256) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts/utils/math/SafeCast.sol#238-241) is never used and should be removed
SafeCast.toUint16(uint256) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts/utils/math/SafeCast.sol#527-530) is never used and should be removed
SafeCast.toUint160(uint256) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts/utils/math/SafeCast.sol#221-224) is never used and should be removed
SafeCast.toUint168(uint256) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts/utils/math/SafeCast.sol#204-207) is never used and should be removed
SafeCast.toUint176(uint256) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts/utils/math/SafeCast.sol#187-190) is never used and should be removed
SafeCast.toUint184(uint256) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts/utils/math/SafeCast.sol#170-173) is never used and should be removed
SafeCast.toUint192(uint256) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts/utils/math/SafeCast.sol#153-156) is never used and should be removed
SafeCast.toUint200(uint256) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts/utils/math/SafeCast.sol#136-139) is never used and should be removed
SafeCast.toUint208(uint256) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts/utils/math/SafeCast.sol#119-122) is never used and should be removed
SafeCast.toUint216(uint256) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts/utils/math/SafeCast.sol#102-105) is never used and should be removed
SafeCast.toUint232(uint256) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts/utils/math/SafeCast.sol#68-71) is never used and should be removed
SafeCast.toUint24(uint256) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts/utils/math/SafeCast.sol#510-513) is never used and should be removed
SafeCast.toUint240(uint256) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts/utils/math/SafeCast.sol#51-54) is never used and should be removed
SafeCast.toUint248(uint256) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts/utils/math/SafeCast.sol#34-37) is never used and should be removed
SafeCast.toUint256(int256) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts/utils/math/SafeCast.sol#556-561) is never used and should be removed
SafeCast.toUint40(uint256) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts/utils/math/SafeCast.sol#476-479) is never used and should be removed
SafeCast.toUint48(uint256) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts/utils/math/SafeCast.sol#459-462) is never used and should be removed
SafeCast.toUint56(uint256) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts/utils/math/SafeCast.sol#442-445) is never used and should be removed
SafeCast.toUint64(uint256) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts/utils/math/SafeCast.sol#425-428) is never used and should be removed
SafeCast.toUint72(uint256) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts/utils/math/SafeCast.sol#408-411) is never used and should be removed
SafeCast.toUint8(uint256) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts/utils/math/SafeCast.sol#544-547) is never used and should be removed
SafeCast.toUint80(uint256) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts/utils/math/SafeCast.sol#391-394) is never used and should be removed
SafeCast.toUint88(uint256) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts/utils/math/SafeCast.sol#374-377) is never used and should be removed
SafeCast.toUint96(uint256) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts/utils/math/SafeCast.sol#357-360) is never used and should be removed
Strings.toHexString(address) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts/utils/Strings.sol#72-74) is never used and should be removed
Strings.toHexString(uint256) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts/utils/Strings.sol#41-52) is never used and should be removed
Strings.toString(uint256) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts/utils/Strings.sol#16-36) is never used and should be removed
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#dead-code
```

```
Pragma version^0.8.0 (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts/access/AccessControl.sol#4) allows old versions
Pragma version^0.8.0 (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts/access/IAccessControl.sol#4) allows old versions
Pragma version^0.8.0 (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts/governance/utils/IVotes.sol#3) allows old versions
Pragma version^0.8.0 (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts/token/ERC20/IERC20.sol#4) allows old versions
Pragma version^0.8.8 (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts/token/ERC20/IERC20.sol#4) allows old versions
Pragma version^0.8.0 (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts/token/ERC20/extensions/ERC20Votes.sol#4) allows old versions
Pragma version^0.8.0 (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts/token/ERC20/extensions/IERC20Metadata.sol#4) allows old versions
Pragma version^0.8.0 (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts/token/ERC20/extensions/draft-ERC20Permit.sol#4) allows old versions
Pragma version^0.8.0 (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts/token/ERC20/extensions/draft-IERC20Permit.sol#4) allows old versions
Pragma version^0.8.0 (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts/utils/Context.sol#4) allows old versions
Pragma version^0.8.0 (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts/utils/Counters.sol#4) allows old versions
Pragma version^0.8.0 (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts/utils/Strings.sol#4) allows old versions
Pragma version^0.8.0 (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts/utils/cryptography/ECDSA.sol#4) allows old versions
Pragma version^0.8.0 (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts/utils/cryptography/draft-EIP712.sol#4) allows old versions
Pragma version^0.8.0 (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts/utils/introspection/ERC165.sol#4) allows old versions
Pragma version^0.8.0 (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts/utils/introspection/IERC165.sol#4) allows old versions
Pragma version^0.8.0 (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts/utils/math/Math.sol#4) allows old versions
Pragma version^0.8.0 (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts/utils/math/SafeCast.sol#4) allows old versions
Pragma version0.8.13 (jobs/archimedes/brownie/contracts/ArchToken.sol#2) necessitates a version too recent to be trusted. Consider deploying with 0.6.12/0.7.6/0.8.7
Pragma version0.8.13 (jobs/archimedes/brownie/contracts/BasicAccessController.sol#2) necessitates a version too recent to be trusted. Consider deploying with 0.6.12/0.7.6/0.8.7
solc-0.8.13 is not recommended for deployment
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#incorrect-versions-of-solidity

Function ERC20Permit.DOMAIN_SEPARATOR() (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts/token/ERC20/extensions/draft-ERC20Permit.sol#81-83) is not in mixedCase
Variable ERC20Permit._PERMIT_TYPEHASH_DEPRECATED_SLOT (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts/token/ERC20/extensions/draft-ERC20Permit.sol#37) is not in mixedCase
Function IERC20Permit.DOMAIN_SEPARATOR() (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts/token/ERC20/extensions/draft-IERC20Permit.sol#59) is not in mixedCase
Variable EIP712._CACHED_DOMAIN_SEPARATOR (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts/utils/cryptography/draft-EIP712.sol#31) is not in mixedCase
Variable EIP712._CACHED_CHAIN_ID (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts/utils/cryptography/draft-EIP712.sol#32) is not in mixedCase
Variable EIP712._CACHED_THIS (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts/utils/cryptography/draft-EIP712.sol#33) is not in mixedCase
Variable EIP712._HASHED_NAME (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts/utils/cryptography/draft-EIP712.sol#35) is not in mixedCase
Variable EIP712._HASHED_VERSION (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts/utils/cryptography/draft-EIP712.sol#36) is not in mixedCase
Variable EIP712._TYPE_HASH (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts/utils/cryptography/draft-EIP712.sol#37) is not in mixedCase
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#conformance-to-solidity-naming-conventions

ArchToken.constructor(address) (jobs/archimedes/brownie/contracts/ArchToken.sol#15-18) uses literals with too many digits:
        - _mint(_addressTreasury,10000000000000000000000000) (jobs/archimedes/brownie/contracts/ArchToken.sol#16)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#too-many-digits

ERC20Permit._PERMIT_TYPEHASH_DEPRECATED_SLOT (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts/token/ERC20/extensions/draft-ERC20Permit.sol#37) is never used in ArchToken (jobs/archimedes/brownie/contracts/ArchToken.sol#14-36)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#unused-state-variable

ERC20Permit._PERMIT_TYPEHASH_DEPRECATED_SLOT (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts/token/ERC20/extensions/draft-ERC20Permit.sol#37) should be constant
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#state-variables-that-could-be-declared-constant
```

- BasicAccessController.sol

```
Different versions of Solidity are used:
        - Version used: ['0.8.13', '^0.8.0']
        - ^0.8.0 (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts/access/AccessControl.sol#4)
        - ^0.8.0 (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts/access/IAccessControl.sol#4)
        - ^0.8.0 (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts/utils/Context.sol#4)
        - ^0.8.0 (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts/utils/Strings.sol#4)
        - ^0.8.0 (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts/utils/introspection/ERC165.sol#4)
        - ^0.8.0 (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts/utils/introspection/IERC165.sol#4)
        - 0.8.13 (jobs/archimedes/brownie/contracts/BasicAccessController.sol#2)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#different-pragma-directives-are-used

AccessControl._setRoleAdmin(bytes32,bytes32) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts/access/AccessControl.sol#214-218) is never used and should be removed
AccessControl._setupRole(bytes32,address) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts/access/AccessControl.sol#205-207) is never used and should be removed
BasicAccessController._requireAdmin() (jobs/archimedes/brownie/contracts/BasicAccessController.sol#67-69) is never used and should be removed
Context._msgData() (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts/utils/Context.sol#21-23) is never used and should be removed
Strings.toHexString(address) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts/utils/Strings.sol#72-74) is never used and should be removed
Strings.toHexString(uint256) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts/utils/Strings.sol#41-52) is never used and should be removed
Strings.toString(uint256) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts/utils/Strings.sol#16-36) is never used and should be removed
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#dead-code

Pragma version^0.8.0 (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts/access/AccessControl.sol#4) allows old versions
Pragma version^0.8.0 (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts/access/IAccessControl.sol#4) allows old versions
Pragma version^0.8.0 (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts/utils/Context.sol#4) allows old versions
Pragma version^0.8.0 (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts/utils/Strings.sol#4) allows old versions
Pragma version^0.8.0 (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts/utils/introspection/ERC165.sol#4) allows old versions
Pragma version^0.8.0 (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts/utils/introspection/IERC165.sol#4) allows old versions
Pragma version0.8.13 (jobs/archimedes/brownie/contracts/BasicAccessController.sol#2) necessitates a version too recent to be trusted. Consider deploying with 0.6.12/0.7.6/0.8.7
solc-0.8.13 is not recommended for deployment
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#incorrect-versions-of-solidity

grantRole(bytes32,address) should be declared external:
        - AccessControl.grantRole(bytes32,address) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts/access/AccessControl.sol#144-146)
revokeRole(bytes32,address) should be declared external:
        - AccessControl.revokeRole(bytes32,address) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts/access/AccessControl.sol#159-161)
renounceRole(bytes32,address) should be declared external:
        - AccessControl.renounceRole(bytes32,address) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts/access/AccessControl.sol#179-183)
        - BasicAccessController.renounceRole(bytes32,address) (jobs/archimedes/brownie/contracts/BasicAccessController.sol#46-53)
setAdmin(address) should be declared external:
        - BasicAccessController.setAdmin(address) (jobs/archimedes/brownie/contracts/BasicAccessController.sol#25-31)
setMinter(address) should be declared external:
        - BasicAccessController.setMinter(address) (jobs/archimedes/brownie/contracts/BasicAccessController.sol#55-61)
getAddressMinter() should be declared external:
        - BasicAccessController.getAddressMinter() (jobs/archimedes/brownie/contracts/BasicAccessController.sol#63-65)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#public-function-that-could-be-declared-external
```

- CDPosition.sol

```
ERC1967UpgradeUpgradeable._functionDelegateCall(address,bytes) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/proxy/ERC1967/ERC1967UpgradeUpgradeable.sol#198-204) uses delegatecall to a input-controlled function id
        - (success,returndata) = target.delegatecall(data) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/proxy/ERC1967/ERC1967UpgradeUpgradeable.sol#202)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#controlled-delegatecall

Reentrancy in VaultOUSD.archimedesRedeem(uint256,address,address) (jobs/archimedes/brownie/contracts/VaultOUSD.sol#55-70):
        External calls:
        - _takeRebaseFees() (jobs/archimedes/brownie/contracts/VaultOUSD.sol#60)
                - _ousd.transfer(_paramStore.getTreasuryAddress(),feeToCollect) (jobs/archimedes/brownie/contracts/VaultOUSD.sol#138)
        - redeemedAmountInAssets = super.redeem(shares,receiver,owner) (jobs/archimedes/brownie/contracts/VaultOUSD.sol#61)
                - returndata = address(token).functionCall(data,SafeERC20: low-level call failed) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/token/ERC20/utils/SafeERC20Upgradeable.sol#110)
                - SafeERC20Upgradeable.safeTransfer(_asset,receiver,assets) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/token/ERC4626Upgradeable.sol#219)
                - (success,returndata) = target.call{value: value}(data) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/AddressUpgradeable.sol#137)
        External calls sending eth:
        - redeemedAmountInAssets = super.redeem(shares,receiver,owner) (jobs/archimedes/brownie/contracts/VaultOUSD.sol#61)
                - (success,returndata) = target.call{value: value}(data) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/AddressUpgradeable.sol#137)
        State variables written after the call(s):
        - _assetsHandledByArchimedes = 0 (jobs/archimedes/brownie/contracts/VaultOUSD.sol#71)
        - _assetsHandledByArchimedes = _assetsHandledByArchimedes - redeemedAmountInAssets (jobs/archimedes/brownie/contracts/VaultOUSD.sol#73)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#reentrancy-vulnerabilities

VaultOUSD._takeRebaseFees() (jobs/archimedes/brownie/contracts/VaultOUSD.sol#108-140) ignores return value by _ousd.transfer(_paramStore.getTreasuryAddress(),feeToCollect) (jobs/archimedes/brownie/contracts/VaultOUSD.sol#138)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#unchecked-transfer

CDPosition (jobs/archimedes/brownie/contracts/CDPosition.sol#17-199) is an upgradeable contract that does not protect its initiliaze functions: CDPosition.initialize() (jobs/archimedes/brownie/contracts/CDPosition.sol#177-186). Anyone can delete the contract with: UUPSUpgradeable.upgradeTo(address) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/proxy/utils/UUPSUpgradeable.sol#72-75)UUPSUpgradeable.upgradeToAndCall(address,bytes) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/proxy/utils/UUPSUpgradeable.sol#85-88)ParameterStore (jobs/archimedes/brownie/contracts/ParameterStore.sol#10-243) is an upgradeable contract that does not protect its initiliaze functions: ParameterStore.initialize() (jobs/archimedes/brownie/contracts/ParameterStore.sol#31-52). Anyone can delete the contract with: UUPSUpgradeable.upgradeTo(address) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/proxy/utils/UUPSUpgradeable.sol#72-75)UUPSUpgradeable.upgradeToAndCall(address,bytes) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/proxy/utils/UUPSUpgradeable.sol#85-88)VaultOUSD (jobs/archimedes/brownie/contracts/VaultOUSD.sol#19-140) is an upgradeable contract that does not protect its initiliaze functions: VaultOUSD.initialize(IERC20MetadataUpgradeable,string,string) (jobs/archimedes/brownie/contracts/VaultOUSD.sol#30-38). Anyone can delete the contract with: UUPSUpgradeable.upgradeTo(address) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/proxy/utils/UUPSUpgradeable.sol#72-75)UUPSUpgradeable.upgradeToAndCall(address,bytes) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/proxy/utils/UUPSUpgradeable.sol#85-88)Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#unprotected-upgradeable-contract

MathUpgradeable.mulDiv(uint256,uint256,uint256) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/math/MathUpgradeable.sol#55-135) performs a multiplication on the result of a division:
        -denominator = denominator / twos (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/math/MathUpgradeable.sol#102)
        -inverse = (3 * denominator) ^ 2 (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/math/MathUpgradeable.sol#117)
MathUpgradeable.mulDiv(uint256,uint256,uint256) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/math/MathUpgradeable.sol#55-135) performs a multiplication on the result of a division:
        -denominator = denominator / twos (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/math/MathUpgradeable.sol#102)
        -inverse *= 2 - denominator * inverse (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/math/MathUpgradeable.sol#121)
MathUpgradeable.mulDiv(uint256,uint256,uint256) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/math/MathUpgradeable.sol#55-135) performs a multiplication on the result of a division:
        -denominator = denominator / twos (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/math/MathUpgradeable.sol#102)
        -inverse *= 2 - denominator * inverse (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/math/MathUpgradeable.sol#122)
MathUpgradeable.mulDiv(uint256,uint256,uint256) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/math/MathUpgradeable.sol#55-135) performs a multiplication on the result of a division:
        -denominator = denominator / twos (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/math/MathUpgradeable.sol#102)
        -inverse *= 2 - denominator * inverse (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/math/MathUpgradeable.sol#123)
MathUpgradeable.mulDiv(uint256,uint256,uint256) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/math/MathUpgradeable.sol#55-135) performs a multiplication on the result of a division:
        -denominator = denominator / twos (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/math/MathUpgradeable.sol#102)
        -inverse *= 2 - denominator * inverse (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/math/MathUpgradeable.sol#124)
MathUpgradeable.mulDiv(uint256,uint256,uint256) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/math/MathUpgradeable.sol#55-135) performs a multiplication on the result of a division:
        -denominator = denominator / twos (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/math/MathUpgradeable.sol#102)
        -inverse *= 2 - denominator * inverse (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/math/MathUpgradeable.sol#125)
MathUpgradeable.mulDiv(uint256,uint256,uint256) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/math/MathUpgradeable.sol#55-135) performs a multiplication on the result of a division:
        -denominator = denominator / twos (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/math/MathUpgradeable.sol#102)
        -inverse *= 2 - denominator * inverse (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/math/MathUpgradeable.sol#126)
MathUpgradeable.mulDiv(uint256,uint256,uint256) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/math/MathUpgradeable.sol#55-135) performs a multiplication on the result of a division:
        -prod0 = prod0 / twos (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/math/MathUpgradeable.sol#105)
        -result = prod0 * inverse (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/math/MathUpgradeable.sol#132)
ParameterStore.calculateAverageAllowedVarAr(uint256) (jobs/archimedes/brownie/contracts/ParameterStore.sol#218-228) performs a multiplication on the result of a division:
        -(_arithToLevRatio / 100000000000000000) * archAmount (jobs/archimedes/brownie/contracts/ParameterStore.sol#219)
VaultOUSD._takeRebaseFees() (jobs/archimedes/brownie/contracts/VaultOUSD.sol#108-140) performs a multiplication on the result of a division:
        -unhandledRebasePayment = ((totalAssets() - _assetsHandledByArchimedes) / 100) * 100 (jobs/archimedes/brownie/contracts/VaultOUSD.sol#126)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#divide-before-multiply

CDPosition.canDeletePosition(uint256) (jobs/archimedes/brownie/contracts/CDPosition.sol#48-51) uses a dangerous strict equality:
        - require(bool,string)(_nftCDP[nftID].usdDeserved == 0,!vUSD borrowed must be zero!) (jobs/archimedes/brownie/contracts/CDPosition.sol#49)
CDPosition.nftIDMustNotExist(uint256) (jobs/archimedes/brownie/contracts/CDPosition.sol#43-46) uses a dangerous strict equality:
        - require(bool,string)(_nftCDP[nftID].oUSDPrinciple == 0,NFT ID must not exist!) (jobs/archimedes/brownie/contracts/CDPosition.sol#44)
ERC4626Upgradeable._convertToAssets(uint256,MathUpgradeable.Rounding) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/token/ERC20/extensions/ERC4626Upgradeable.sol#168-174) uses a dangerous strict equality:
        - (supply == 0) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/token/ERC20/extensions/ERC4626Upgradeable.sol#170-173)
ERC4626Upgradeable._convertToShares(uint256,MathUpgradeable.Rounding) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/token/ERC20/extensions/ERC4626Upgradeable.sol#157-163) uses a dangerous strict equality:
        - (assets == 0 || supply == 0) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/token/ERC20/extensions/ERC4626Upgradeable.sol#159-162)
ERC4626Upgradeable._isVaultCollateralized() (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/token/ERC20/extensions/ERC4626Upgradeable.sol#224-226) uses a dangerous strict equality:
        - totalAssets() > 0 || totalSupply() == 0 (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/token/ERC20/extensions/ERC4626Upgradeable.sol#225)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#dangerous-strict-equalities
```

```
Reentrancy in VaultOUSD.archimedesDeposit(uint256,address) (jobs/archimedes/brownie/contracts/VaultOUSD.sol#37-41):
        External calls:
        - _takeRebaseFees() (jobs/archimedes/brownie/contracts/VaultOUSD.sol#38)
                - _ousd.transfer(_paramStore.getTreasuryAddress(),feeToCollect) (jobs/archimedes/brownie/contracts/VaultOUSD.sol#138)
        State variables written after the call(s):
        - _assetsHandledByArchimedes += assets (jobs/archimedes/brownie/contracts/VaultOUSD.sol#39)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#reentrancy-vulnerabilities-1

ERC1967UpgradeUpgradeable._upgradeToAndCallUUPS(address,bytes,bool) (slot (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/proxy/ERC1967/ERC1967UpgradeUpgradeable.sol#98) is a local variable never initialized
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#uninitialized-local-variables

ERC1967UpgradeUpgradeable._upgradeToAndCallUUPS(address,bytes,bool) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/proxy/ERC1967/ERC1967UpgradeUpgradeable.sol#87-105) ignores return value by IERC1822ProxiableUpgradeable(newImplementation).p
roxiableUUID() (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/proxy/ERC1967/ERC1967UpgradeUpgradeable.sol#90-102)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#unused-return

VaultOUSD.initialize(IERC20MetadataUpgradeable,string,string).asset (jobs/archimedes/brownie/contracts/VaultOUSD.sol#89) shadows:
        - ERC4626Upgradeable.asset() (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/token/ERC20/extensions/ERC4626Upgradeable.sol#44-46) (function)
        - IERC4626Upgradeable.asset() (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/interfaces/IERC4626Upgradeable.sol#32) (function)
VaultOUSD.initialize(IERC20MetadataUpgradeable,string,string).name (jobs/archimedes/brownie/contracts/VaultOUSD.sol#90) shadows:
        - ERC20Upgradeable.name() (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/token/ERC20/ERC20Upgradeable.sol#67-69) (function)
        - IERC20MetadataUpgradeable.name() (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/token/ERC20/extensions/IERC20MetadataUpgradeable.sol#17) (function)
VaultOUSD.initialize(IERC20MetadataUpgradeable,string,string).symbol (jobs/archimedes/brownie/contracts/VaultOUSD.sol#91) shadows:
        - ERC20Upgradeable.symbol() (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/token/ERC20/ERC20Upgradeable.sol#75-77) (function)
        - IERC20MetadataUpgradeable.symbol() (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/token/ERC20/extensions/IERC20MetadataUpgradeable.sol#22) (function)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#local-variable-shadowing

VaultOUSD.archimedesDeposit(uint256,address) (jobs/archimedes/brownie/contracts/VaultOUSD.sol#37-41) should emit an event for:
        - _assetsHandledByArchimedes += assets (jobs/archimedes/brownie/contracts/VaultOUSD.sol#39)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#missing-events-arithmetic

CDPosition.setDependencies(address,address).addressVaultOUSD (jobs/archimedes/brownie/contracts/CDPosition.sol#188) lacks a zero-check on :
                - _addressVaultOUSD = addressVaultOUSD (jobs/archimedes/brownie/contracts/CDPosition.sol#189)
CDPosition.setDependencies(address,address).addressParameterStore (jobs/archimedes/brownie/contracts/CDPosition.sol#188) lacks a zero-check on :
                - _addressParameterStore = addressParameterStore (jobs/archimedes/brownie/contracts/CDPosition.sol#190)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#missing-zero-address-validation

Reentrancy in ERC4626Upgradeable._deposit(address,address,uint256,uint256) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/token/ERC20/extensions/ERC4626Upgradeable.sol#179-196):
        External calls:
        - SafeERC20Upgradeable.safeTransferFrom(_asset,caller,address(this),assets) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/token/ERC20/extensions/ERC4626Upgradeable.sol#192)
        State variables written after the call(s):
        - _mint(receiver,shares) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/token/ERC20/extensions/ERC4626Upgradeable.sol#193)
                - _balances[account] += amount (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/token/ERC20/ERC20Upgradeable.sol#268)
        - _mint(receiver,shares) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/token/ERC20/extensions/ERC4626Upgradeable.sol#193)
                - _totalSupply += amount (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/token/ERC20/ERC20Upgradeable.sol#267)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#reentrancy-vulnerabilities-2

Reentrancy in ERC4626Upgradeable._deposit(address,address,uint256,uint256) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/token/ERC20/extensions/ERC4626Upgradeable.sol#179-196):
        External calls:
        - SafeERC20Upgradeable.safeTransferFrom(_asset,caller,address(this),assets) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/token/ERC20/extensions/ERC4626Upgradeable.sol#192)
        Event emitted after the call(s):
        - Deposit(caller,receiver,assets,shares) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/token/ERC20/extensions/ERC4626Upgradeable.sol#195)
        - Transfer(address(0),account,amount) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/token/ERC20/ERC20Upgradeable.sol#269)
                - _mint(receiver,shares) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/token/ERC20/extensions/ERC4626Upgradeable.sol#193)
Reentrancy in ERC4626Upgradeable._withdraw(address,address,address,uint256,uint256) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/token/ERC20/extensions/ERC4626Upgradeable.sol#201-222):
        External calls:
        - SafeERC20Upgradeable.safeTransfer(_asset,receiver,assets) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/token/ERC20/extensions/ERC4626Upgradeable.sol#219)
        Event emitted after the call(s):
        - Withdraw(caller,receiver,owner,assets,shares) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/token/ERC20/extensions/ERC4626Upgradeable.sol#221)
Reentrancy in VaultOUSD.archimedesDeposit(uint256,address) (jobs/archimedes/brownie/contracts/VaultOUSD.sol#37-41):
        External calls:
        - _takeRebaseFees() (jobs/archimedes/brownie/contracts/VaultOUSD.sol#38)
                - _ousd.transfer(_paramStore.getTreasuryAddress(),feeToCollect) (jobs/archimedes/brownie/contracts/VaultOUSD.sol#138)
        - super.deposit(assets,receiver) (jobs/archimedes/brownie/contracts/VaultOUSD.sol#40)
                - SafeERC20Upgradeable.safeTransferFrom(_asset,caller,address(this),assets) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/token/ERC20/extensions/ERC4626Upgradeable.sol#192)
                - returndata = address(token).functionCall(data,SafeERC20: low-level call failed) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/token/ERC20/utils/SafeERC20Upgradeable.sol#110)
                - (success,returndata) = target.call{value: value}(data) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/AddressUpgradeable.sol#137)
        External calls sending eth:
        - super.deposit(assets,receiver) (jobs/archimedes/brownie/contracts/VaultOUSD.sol#40)
                - (success,returndata) = target.call{value: value}(data) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/AddressUpgradeable.sol#137)
        Event emitted after the call(s):
        - Deposit(caller,receiver,assets,shares) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/token/ERC20/extensions/ERC4626Upgradeable.sol#195)
                - super.deposit(assets,receiver) (jobs/archimedes/brownie/contracts/VaultOUSD.sol#40)
        - Transfer(address(0),account,amount) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/token/ERC20/ERC20Upgradeable.sol#269)
                - super.deposit(assets,receiver) (jobs/archimedes/brownie/contracts/VaultOUSD.sol#40)
Reentrancy in VaultOUSD.archimedesRedeem(uint256,address,address) (jobs/archimedes/brownie/contracts/VaultOUSD.sol#55-70):
        External calls:
        - _takeRebaseFees() (jobs/archimedes/brownie/contracts/VaultOUSD.sol#60)
                - _ousd.transfer(_paramStore.getTreasuryAddress(),feeToCollect) (jobs/archimedes/brownie/contracts/VaultOUSD.sol#138)
        - redeemedAmountInAssets = super.redeem(shares,receiver,owner) (jobs/archimedes/brownie/contracts/VaultOUSD.sol#61)
                - returndata = address(token).functionCall(data,SafeERC20: low-level call failed) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/token/ERC20/utils/SafeERC20Upgradeable.sol#118)
                - SafeERC20Upgradeable.safeTransfer(_asset,receiver,assets) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/token/ERC20/extensions/ERC4626Upgradeable.sol#219)
                - (success,returndata) = target.call{value: value}(data) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/AddressUpgradeable.sol#137)
        External calls sending eth:
        - redeemedAmountInAssets = super.redeem(shares,receiver,owner) (jobs/archimedes/brownie/contracts/VaultOUSD.sol#61)
                - (success,returndata) = target.call{value: value}(data) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/AddressUpgradeable.sol#137)
        Event emitted after the call(s):
        - Approval(owner,spender,amount) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/token/ERC20/ERC20Upgradeable.sol#324)
                - redeemedAmountInAssets = super.redeem(shares,receiver,owner) (jobs/archimedes/brownie/contracts/VaultOUSD.sol#61)
        - Transfer(account,address(0),amount) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/token/ERC20/ERC20Upgradeable.sol#297)
                - redeemedAmountInAssets = super.redeem(shares,receiver,owner) (jobs/archimedes/brownie/contracts/VaultOUSD.sol#61)
        - Withdraw(caller,receiver,owner,assets,shares) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/token/ERC20/extensions/ERC4626Upgradeable.sol#221)
                - redeemedAmountInAssets = super.redeem(shares,receiver,owner) (jobs/archimedes/brownie/contracts/VaultOUSD.sol#61)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#reentrancy-vulnerabilities-3

CDPosition.removeSharesFromPosition(uint256,uint256) (jobs/archimedes/brownie/contracts/CDPosition.sol#82-85) uses timestamp for comparisons
        Dangerous comparisons:
        - require(bool,string)(_nftCDP[nftID].shares >= shares,Shares exceed position balance) (jobs/archimedes/brownie/contracts/CDPosition.sol#83)
CDPosition.withdrawOUSDFromPosition(uint256,uint256) (jobs/archimedes/brownie/contracts/CDPosition.sol#116-119) uses timestamp for comparisons
        Dangerous comparisons:
        - require(bool,string)(getOUSDTotalIncludeInterest(nftID) >= oUSDAmountToWithdraw,Insufficient OUSD balance) (jobs/archimedes/brownie/contracts/CDPosition.sol#117)
CDPosition.getOUSDInterestEarned(uint256) (jobs/archimedes/brownie/contracts/CDPosition.sol#126-141) uses timestamp for comparisons
        Dangerous comparisons:
        - _nftCDP[nftID].oUSDTotalWithoutInterest > totalFundsFromPreviewRedeem (jobs/archimedes/brownie/contracts/CDPosition.sol#136)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#block-timestamp
CDPosition.removeSharesFromPosition(uint256,uint256) (jobs/archimedes/brownie/contracts/CDPosition.sol#82-85) uses timestamp for comparisons
        Dangerous comparisons:
        - require(bool,string)(_nftCDP[nftID].shares >= shares,Shares exceed position balance) (jobs/archimedes/brownie/contracts/CDPosition.sol#83)
CDPosition.withdrawOUSDFromPosition(uint256,uint256) (jobs/archimedes/brownie/contracts/CDPosition.sol#116-119) uses timestamp for comparisons
        Dangerous comparisons:
        - require(bool,string)(getOUSDTotalIncludeInterest(nftID) >= oUSDAmountToWithdraw,Insufficient OUSD balance) (jobs/archimedes/brownie/contracts/CDPosition.sol#117)
CDPosition.getOUSDInterestEarned(uint256) (jobs/archimedes/brownie/contracts/CDPosition.sol#126-141) uses timestamp for comparisons
        Dangerous comparisons:
        - _nftCDP[nftID].oUSDTotalWithoutInterest > totalFundsFromPreviewRedeem (jobs/archimedes/brownie/contracts/CDPosition.sol#136)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#block-timestamp

AddressUpgradeable.verifyCallResult(bool,bytes,string) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/AddressUpgradeable.sol#174-194) uses assembly
        - INLINE ASM (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/AddressUpgradeable.sol#186-189)
StorageSlotUpgradeable.getAddressSlot(bytes32) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/StorageSlotUpgradeable.sol#52-57) uses assembly
        - INLINE ASM (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/StorageSlotUpgradeable.sol#54-56)
StorageSlotUpgradeable.getBooleanSlot(bytes32) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/StorageSlotUpgradeable.sol#62-67) uses assembly
        - INLINE ASM (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/StorageSlotUpgradeable.sol#64-66)
StorageSlotUpgradeable.getBytes32Slot(bytes32) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/StorageSlotUpgradeable.sol#72-77) uses assembly
        - INLINE ASM (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/StorageSlotUpgradeable.sol#74-76)
StorageSlotUpgradeable.getUint256Slot(bytes32) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/StorageSlotUpgradeable.sol#82-87) uses assembly
        - INLINE ASM (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/StorageSlotUpgradeable.sol#84-86)
MathUpgradeable.mulDiv(uint256,uint256,uint256) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/math/MathUpgradeable.sol#55-135) uses assembly
        - INLINE ASM (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/math/MathUpgradeable.sol#66-70)
        - INLINE ASM (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/math/MathUpgradeable.sol#86-93)
        - INLINE ASM (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/math/MathUpgradeable.sol#100-109)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#assembly-usage

. . . . . .
Different versions of Solidity are used:
        - Version used: ['^0.8.13', '^0.8.0', '^0.8.1', '^0.8.2']
        - ^0.8.0 (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/access/AccessControlUpgradeable.sol#4)
        - ^0.8.0 (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/access/IAccessControlUpgradeable.sol#4)
        - ^0.8.0 (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/interfaces/IERC20MetadataUpgradeable.sol#4)
        - ^0.8.0 (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/interfaces/IERC4626Upgradeable.sol#4)
        - ^0.8.0 (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/interfaces/draft-IERC1822Upgradeable.sol#4)
        - ^0.8.2 (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/proxy/ERC1967/ERC1967UpgradeUpgradeable.sol#4)
        - ^0.8.0 (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/proxy/beacon/IBeaconUpgradeable.sol#4)
        - ^0.8.2 (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/proxy/utils/Initializable.sol#4)
        - ^0.8.0 (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/proxy/utils/UUPSUpgradeable.sol#4)
        - ^0.8.0 (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/security/ReentrancyGuardUpgradeable.sol#4)
        - ^0.8.0 (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/token/ERC20/ERC20Upgradeable.sol#4)
        - ^0.8.0 (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/token/ERC20/IERC20Upgradeable.sol#4)
        - ^0.8.0 (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/token/ERC20/extensions/ERC4626Upgradeable.sol#4)
        - ^0.8.0 (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/token/ERC20/extensions/IERC20MetadataUpgradeable.sol#4)
        - ^0.8.0 (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/token/ERC20/extensions/draft-IERC20PermitUpgradeable.sol#4)
        - ^0.8.0 (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/token/ERC20/utils/SafeERC20Upgradeable.sol#4)
        - ^0.8.1 (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/AddressUpgradeable.sol#4)
        - ^0.8.0 (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/ContextUpgradeable.sol#4)
        - ^0.8.0 (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/StorageSlotUpgradeable.sol#4)
        - ^0.8.0 (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/StringsUpgradeable.sol#4)
        - ^0.8.0 (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/introspection/ERC165Upgradeable.sol#4)
        - ^0.8.0 (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/introspection/IERC165Upgradeable.sol#4)
        - ^0.8.0 (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/math/MathUpgradeable.sol#4)
        - 0.8.13 (jobs/archimedes/brownie/contracts/AccessController.sol#2)
        - 0.8.13 (jobs/archimedes/brownie/contracts/CDPosition.sol#2)
        - 0.8.13 (jobs/archimedes/brownie/contracts/ParameterStore.sol#2)
        - 0.8.13 (jobs/archimedes/brownie/contracts/VaultOUSD.sol#2)
        - ^0.8.0 (jobs/archimedes/brownie/contracts/interfaces/IOUSD.sol#4)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#different-pragma-directives-are-used
```

AUTOMATED TESTING

```
AccessControlUpgradeable.__AccessControl_init_unchained() (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/access/AccessControlUpgradeable.sol#54-55) is never used and should be removed
AccessControlUpgradeable._setRoleAdmin(bytes32,bytes32) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/access/AccessControlUpgradeable.sol#220-224) is never used and should be removed
AccessControlUpgradeable._setupRole(bytes32,address) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/access/AccessControlUpgradeable.sol#211-213) is never used and should be removed
AddressUpgradeable.functionCall(address,bytes) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/AddressUpgradeable.sol#85-87) is never used and should be removed
AddressUpgradeable.functionCallWithValue(address,bytes,uint256) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/AddressUpgradeable.sol#114-120) is never used and should be removed
AddressUpgradeable.functionStaticCall(address,bytes) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/AddressUpgradeable.sol#147-149) is never used and should be removed
AddressUpgradeable.functionStaticCall(address,bytes,string) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/AddressUpgradeable.sol#157-166) is never used and should be removed
AddressUpgradeable.sendValue(address,uint256) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/AddressUpgradeable.sol#60-65) is never used and should be removed
ContextUpgradeable.__Context_init() (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/ContextUpgradeable.sol#9-10) is never used and should be removed
ContextUpgradeable.__Context_init_unchained() (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/ContextUpgradeable.sol#21-22) is never used and should be removed
ContextUpgradeable._msgData() (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/ContextUpgradeable.sol#27-29) is never used and should be removed
ERC165Upgradeable.__ERC165_init() (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/introspection/ERC165Upgradeable.sol#24-25) is never used and should be removed
ERC165Upgradeable.__ERC165_init_unchained() (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/introspection/ERC165Upgradeable.sol#27-28) is never used and should be removed
ERC1967UpgradeUpgradeable.__ERC1967Upgrade_init() (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/proxy/ERC1967/ERC1967UpgradeUpgradeable.sol#21-22) is never used and should be removed
ERC1967UpgradeUpgradeable.__ERC1967Upgrade_init_unchained() (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/proxy/ERC1967/ERC1967UpgradeUpgradeable.sol#24-25) is never used and should be removed
ERC1967UpgradeUpgradeable._changeAdmin(address) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/proxy/ERC1967/ERC1967UpgradeUpgradeable.sol#139-142) is never used and should be removed
ERC1967UpgradeUpgradeable._getAdmin() (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/proxy/ERC1967/ERC1967UpgradeUpgradeable.sol#122-124) is never used and should be removed
ERC1967UpgradeUpgradeable._getBeacon() (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/proxy/ERC1967/ERC1967UpgradeUpgradeable.sol#158-160) is never used and should be removed
ERC1967UpgradeUpgradeable._setAdmin(address) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/proxy/ERC1967/ERC1967UpgradeUpgradeable.sol#129-132) is never used and should be removed
ERC1967UpgradeUpgradeable._setBeacon(address) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/proxy/ERC1967/ERC1967UpgradeUpgradeable.sol#165-172) is never used and should be removed
ERC1967UpgradeUpgradeable._upgradeBeaconToAndCall(address,bytes,bool) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/proxy/ERC1967/ERC1967UpgradeUpgradeable.sol#180-198) is never used and should be removed
Initializable._disableInitializers() (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/Initializable.sol#131-137) is never used and should be removed
MathUpgradeable.average(uint256,uint256) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/math/MathUpgradeable.sol#34-37) is never used and should be removed
MathUpgradeable.ceilDiv(uint256,uint256) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/math/MathUpgradeable.sol#45-48) is never used and should be removed
MathUpgradeable.max(uint256,uint256) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/math/MathUpgradeable.sol#19-21) is never used and should be removed
MathUpgradeable.min(uint256,uint256) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/math/MathUpgradeable.sol#26-28) is never used and should be removed
MathUpgradeable.sqrt(uint256) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/math/MathUpgradeable.sol#58-124) is never used and should be removed
MathUpgradeable.sqrt(uint256,MathUpgradeable.Rounding) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/math/MathUpgradeable.sol#210-225) is never used and should be removed
SafeERC20Upgradeable.safeApprove(IERC20Upgradeable,address,uint256) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/token/ERC20/utils/SafeERC20Upgradeable.sol#46-59) is never used and should be removed
SafeERC20Upgradeable.safeDecreaseAllowance(IERC20Upgradeable,address,uint256) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/token/ERC20/utils/SafeERC20Upgradeable.sol#78-81) is never used and should be removed
SafeERC20Upgradeable.safeIncreaseAllowance(IERC20Upgradeable,address,uint256) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/token/ERC20/utils/SafeERC20Upgradeable.sol#61-68) is never used and should be removed
SafeERC20Upgradeable.safePermit(IERC20PermitUpgradeable,address,address,uint256,uint256,uint8,bytes32,bytes32) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/token/ERC20/utils/SafeERC20Upgradeable.sol#83-97) is never used and should be removed
StorageSlotUpgradeable.getBytes32Slot(bytes32) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/StorageSlotUpgradeable.sol#72-77) is never used and should be removed
StorageSlotUpgradeable.getInt256Slot(bytes32) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/StorageSlotUpgradeable.sol#82-87) is never used and should be removed
StringsUpgradeable.toHexString(address) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/StringsUpgradeable.sol#72-74) is never used and should be removed
StringsUpgradeable.toHexString(uint256) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/StringsUpgradeable.sol#41-52) is never used and should be removed
StringsUpgradeable.toString(uint256) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/StringsUpgradeable.sol#16-36) is never used and should be removed
UUPSUpgradeable.__UUPSUpgradeable_init_unchained() (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/proxy/utils/UUPSUpgradeable.sol#26-27) is never used and should be removed
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#dead-code

Pragma version^0.8.0 (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/access/AccessControlUpgradeable.sol#4) allows old versions
Pragma version^0.8.0 (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/access/IAccessControlUpgradeable.sol#4) allows old versions
Pragma version^0.8.0 (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/interfaces/IERC20MetadataUpgradeable.sol#4) allows old versions
Pragma version^0.8.0 (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/interfaces/IERC4626Upgradeable.sol#4) allows old versions
Pragma version^0.8.0 (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/interfaces/draft-IERC1822Upgradeable.sol#4) allows old versions
Pragma version^0.8.2 (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/interfaces/draft-IERC1967Upgradeable.sol#4) allows old versions
Pragma version^0.8.0 (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/proxy/beacon/IBeaconUpgradeable.sol#4) allows old versions
Pragma version^0.8.2 (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/proxy/utils/Initializable.sol#4) allows old versions
Pragma version^0.8.0 (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/proxy/utils/UUPSUpgradeable.sol#4) allows old versions
Pragma version^0.8.0 (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/security/ReentrancyGuardUpgradeable.sol#4) allows old versions
Pragma version^0.8.0 (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/token/ERC20/ERC20Upgradeable.sol#4) allows old versions
Pragma version^0.8.0 (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/token/ERC20/IERC20Upgradeable.sol#4) allows old versions
Pragma version^0.8.0 (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/token/ERC20/extensions/ERC4626Upgradeable.sol#4) allows old versions
Pragma version^0.8.0 (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/token/ERC20/extensions/IERC20MetadataUpgradeable.sol#4) allows old versions
Pragma version^0.8.0 (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/token/ERC20/extensions/draft-IERC20PermitUpgradeable.sol#4) allows old versions
Pragma version^0.8.0 (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/token/ERC20/utils/SafeERC20Upgradeable.sol#4) allows old versions
Pragma version^0.8.1 (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/AddressUpgradeable.sol#4) allows old versions
Pragma version^0.8.0 (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/ContextUpgradeable.sol#4) allows old versions
Pragma version^0.8.0 (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/StorageSlotUpgradeable.sol#4) allows old versions
Pragma version^0.8.0 (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/StringsUpgradeable.sol#4) allows old versions
Pragma version^0.8.0 (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/introspection/ERC165Upgradeable.sol#4) allows old versions
Pragma version^0.8.0 (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/introspection/IERC165Upgradeable.sol#4) allows old versions
Pragma version^0.8.0 (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/math/MathUpgradeable.sol#4) allows old versions
Pragma version0.8.13 (jobs/archimedes/brownie/contracts/AccessControl.sol#2) necessitates a version too recent to be trusted. Consider deploying with 0.6.12/0.7.6/0.8.7
Pragma version0.8.13 (jobs/archimedes/brownie/contracts/CDPosition.sol#2) necessitates a version too recent to be trusted. Consider deploying with 0.6.12/0.7.6/0.8.7
Pragma version0.8.13 (jobs/archimedes/brownie/contracts/ParameterStore.sol#2) necessitates a version too recent to be trusted. Consider deploying with 0.6.12/0.7.6/0.8.7
Pragma version0.8.13 (jobs/archimedes/brownie/contracts/VaultOUSD.sol#2) 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 (jobs/archimedes/brownie/contracts/interfaces/IOUSD.sol#4) allows old versions
solc-0.8.13 is not recommended for deployment
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#incorrect-versions-of-solidity

Low level call in ERC1967UpgradeUpgradeable._functionDelegateCall(address,bytes) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/proxy/ERC1967/ERC1967UpgradeUpgradeable.sol#198-204):
        - (success,returndata) = target.delegatecall(data) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/proxy/ERC1967/ERC1967UpgradeUpgradeable.sol#202)
Low level call in AddressUpgradeable.sendValue(address,uint256) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/AddressUpgradeable.sol#60-65):
        - (success) = recipient.call{value: amount}() (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/AddressUpgradeable.sol#63)
Low level call in AddressUpgradeable.functionCallWithValue(address,bytes,uint256,string) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/AddressUpgradeable.sol#128-139):
        - (success,returndata) = target.call{value: value}(data) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/AddressUpgradeable.sol#137)
Low level call in AddressUpgradeable.functionStaticCall(address,bytes,string) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/AddressUpgradeable.sol#157-166):
        - (success,returndata) = target.staticcall(data) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/AddressUpgradeable.sol#164)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#low-level-calls

Function AccessControlUpgradeable.__AccessControl_init() (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/access/AccessControlUpgradeable.sol#51-52) is not in mixedCase
Function AccessControlUpgradeable.__AccessControl_init_unchained() (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/access/AccessControlUpgradeable.sol#54-55) is not in mixedCase
Variable AccessControlUpgradeable.__gap (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/access/AccessControlUpgradeable.sol#259) is not in mixedCase
Function ERC1967UpgradeUpgradeable.__ERC1967Upgrade_init() (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/proxy/ERC1967/ERC1967UpgradeUpgradeable.sol#21-22) is not in mixedCase
Function ERC1967UpgradeUpgradeable.__ERC1967Upgrade_init_unchained() (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/proxy/ERC1967/ERC1967UpgradeUpgradeable.sol#24-25) is not in mixedCase
Variable ERC1967UpgradeUpgradeable.__gap (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/proxy/ERC1967/ERC1967UpgradeUpgradeable.sol#211) is not in mixedCase
Function UUPSUpgradeable.__UUPSUpgradeable_init() (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/proxy/utils/UUPSUpgradeable.sol#23-24) is not in mixedCase
Function UUPSUpgradeable.__UUPSUpgradeable_init_unchained() (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/proxy/utils/UUPSUpgradeable.sol#26-27) is not in mixedCase
Variable UUPSUpgradeable.__gap (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/proxy/utils/UUPSUpgradeable.sol#107) is not in mixedCase
Variable UUPSUpgradeable.self (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/proxy/utils/UUPSUpgradeable.sol#29) is not in mixedCase
Function ReentrancyGuardUpgradeable.__ReentrancyGuard_init() (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/security/ReentrancyGuardUpgradeable.sol#48-42) is not in mixedCase
Function ReentrancyGuardUpgradeable.__ReentrancyGuard_init_unchained() (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/security/ReentrancyGuardUpgradeable.sol#44-46) is not in mixedCase
Variable ReentrancyGuardUpgradeable.__gap (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/security/ReentrancyGuardUpgradeable.sol#74) is not in mixedCase
Function ERC20Upgradeable.__ERC20_init(string,string) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/token/ERC20/ERC20Upgradeable.sol#55-57) is not in mixedCase
Function ERC20Upgradeable.__ERC20_init_unchained(string,string) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/token/ERC20/ERC20Upgradeable.sol#58-62) is not in mixedCase
Variable ERC20Upgradeable.__gap (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/token/ERC20/ERC20Upgradeable.sol#394) is not in mixedCase
Function ERC4626Upgradeable.__ERC4626_init(IERC20MetadataUpgradeable) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/token/ERC20/extensions/ERC4626Upgradeable.sol#35-37) is not in mixedCase
Function ERC4626Upgradeable.__ERC4626_init_unchained(IERC20MetadataUpgradeable) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/token/ERC20/extensions/ERC4626Upgradeable.sol#39-41) is not in mixedCase
Variable ERC4626Upgradeable.__gap (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/token/ERC20/extensions/ERC4626Upgradeable.sol#233) is not in mixedCase
Function IERC20PermitUpgradeable.DOMAIN_SEPARATOR() (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/token/ERC20/extensions/draft-IERC20PermitUpgradeable.sol#59) is not in mixedCase
Function ContextUpgradeable.__Context_init() (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/ContextUpgradeable.sol#18-19) is not in mixedCase
Function ContextUpgradeable.__Context_init_unchained() (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/ContextUpgradeable.sol#21-22) is not in mixedCase
Variable ContextUpgradeable.__gap (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/ContextUpgradeable.sol#36) is not in mixedCase
Function ERC165Upgradeable.__ERC165_init() (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/introspection/ERC165Upgradeable.sol#24-25) is not in mixedCase
Function ERC165Upgradeable.__ERC165_init_unchained() (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/introspection/ERC165Upgradeable.sol#27-28) is not in mixedCase
Variable ERC165Upgradeable.__gap (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/introspection/ERC165Upgradeable.sol#41) is not in mixedCase
Variable CDPosition._ntfCDP (jobs/archimedes/brownie/contracts/CDPosition.sol#30) is not in mixedCase
Variable CDPosition._addressVaultOUSD (jobs/archimedes/brownie/contracts/CDPosition.sol#32) is not in mixedCase
Variable CDPosition._addressParameterStore (jobs/archimedes/brownie/contracts/CDPosition.sol#33) is not in mixedCase
Variable CDPosition._vault (jobs/archimedes/brownie/contracts/CDPosition.sol#34) is not in mixedCase
Variable CDPosition._parameterStore (jobs/archimedes/brownie/contracts/CDPosition.sol#35) is not in mixedCase
Variable ParameterStore._maxNumberOfCycles (jobs/archimedes/brownie/contracts/ParameterStore.sol#14) is not in mixedCase
Variable ParameterStore._originationFeeRate (jobs/archimedes/brownie/contracts/ParameterStore.sol#15) is not in mixedCase
Variable ParameterStore._globalCollateralRate (jobs/archimedes/brownie/contracts/ParameterStore.sol#16) is not in mixedCase
Variable ParameterStore._rebaseFeeRate (jobs/archimedes/brownie/contracts/ParameterStore.sol#17) is not in mixedCase
Variable ParameterStore._treasuryAddress (jobs/archimedes/brownie/contracts/ParameterStore.sol#18) is not in mixedCase
Variable ParameterStore._curveGuardPercentage (jobs/archimedes/brownie/contracts/ParameterStore.sol#19) is not in mixedCase
Variable ParameterStore._liquidationDiscount (jobs/archimedes/brownie/contracts/ParameterStore.sol#20) is not in mixedCase
Variable ParameterStore._archToLevRatio (jobs/archimedes/brownie/contracts/ParameterStore.sol#22) is not in mixedCase
Variable ParameterStore._curveMaxExchangeGuard (jobs/archimedes/brownie/contracts/ParameterStore.sol#24) is not in mixedCase
Variable ParameterStore._minPositionCollateral (jobs/archimedes/brownie/contracts/ParameterStore.sol#25) is not in mixedCase
Variable ParameterStore._positionTimeToLiveInDays (jobs/archimedes/brownie/contracts/ParameterStore.sol#26) is not in mixedCase
Parameter VaultOUSD.setDependencies(address,address)._addressParameterStore (jobs/archimedes/brownie/contracts/VaultOUSD.sol#31) is not in mixedCase
Parameter VaultOUSD.setDependencies(address,address)._addressOUSD (jobs/archimedes/brownie/contracts/VaultOUSD.sol#31) is not in mixedCase
Variable VaultOUSD._paramStore (jobs/archimedes/brownie/contracts/VaultOUSD.sol#22) is not in mixedCase
Variable VaultOUSD._ousd (jobs/archimedes/brownie/contracts/VaultOUSD.sol#23) is not in mixedCase
Variable VaultOUSD._assetsHandledByArchimedes (jobs/archimedes/brownie/contracts/VaultOUSD.sol#25) is not in mixedCase
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#conformance-to-solidity-naming-conventions

ParameterStore.changeArchToLevRatio(uint256) (jobs/archimedes/brownie/contracts/ParameterStore.sol#99-103) uses literals with too many digits:
        - require(bool,string)(newArchToLevRatio < 500000000000000000000000 && newArchToLevRatio >= 10000000000000000,new ArchToLevRatio out of range) (jobs/archimedes/brownie/contracts/ParameterStore.sol#100)
ParameterStore.changeMinPositionCollateral(uint256) (jobs/archimedes/brownie/contracts/ParameterStore.sol#111-115) uses literals with too many digits:
        - require(bool,string)(newMinPositionCollateral < (1000000000000000000000000) && newMinPositionCollateral > (1000000000000000000),New min collateral out of range) (jobs/archimedes/brownie/contracts/ParameterStore.sol#112)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#too-many-digits

RentrancyGuardUpgradeable.__gap (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/security/ReentrancyGuardUpgradeable.sol#74) is never used in CDPosition (jobs/archimedes/brownie/contracts/CDPosition.sol#17-109)
UUPSUpgradeable.__gap (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/proxy/utils/UUPSUpgradeable.sol#107) is never used in ParameterStore (jobs/archimedes/brownie/contracts/ParameterStore.sol#10-243)
UUPSUpgradeable.__gap (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/proxy/utils/UUPSUpgradeable.sol#107) is never used in VaultOUSD (jobs/archimedes/brownie/contracts/VaultOUSD.sol#19-146)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#unused-state-variable

grantRole(bytes32,address) should be declared external:
        - AccessControlUpgradeable.grantRole(bytes32,address) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/access/AccessControlUpgradeable.sol#158-152)
revokeRole(bytes32,address) should be declared external:
        - AccessControlUpgradeable.revokeRole(bytes32,address) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/access/AccessControlUpgradeable.sol#165-167)
renounceRole(bytes32,address) should be declared external:
        - AccessControlUpgradeable.renounceRole(bytes32,address) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/access/AccessControlUpgradeable.sol#185-189)
        - AccessController.renounceRole(bytes32,address) (jobs/archimedes/brownie/contracts/AccessController.sol#62-69)
name() should be declared external:
        - ERC20Upgradeable.name() (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/token/ERC20/ERC20Upgradeable.sol#67-69)
symbol() should be declared external:
        - ERC20Upgradeable.symbol() (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/token/ERC20/ERC20Upgradeable.sol#75-77)
transfer(address,uint256) should be declared external:
        - ERC20Upgradeable.transfer(address,uint256) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/token/ERC20/ERC20Upgradeable.sol#118-122)
approve(address,uint256) should be declared external:
        - ERC20Upgradeable.approve(address,uint256) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/token/ERC20/ERC20Upgradeable.sol#141-145)
transferFrom(address,address,uint256) should be declared external:
        - ERC20Upgradeable.transferFrom(address,address,uint256) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/token/ERC20/ERC20Upgradeable.sol#163-172)
increaseAllowance(address,uint256) should be declared external:
        - ERC20Upgradeable.increaseAllowance(address,uint256) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/token/ERC20/ERC20Upgradeable.sol#186-190)
decreaseAllowance(address,uint256) should be declared external:
        - ERC20Upgradeable.decreaseAllowance(address,uint256) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/token/ERC20/ERC20Upgradeable.sol#206-215)
asset() should be declared external:
        - ERC4626Upgradeable.asset() (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/token/ERC20/extensions/ERC4626Upgradeable.sol#44-46)
convertToShares(uint256) should be declared external:
        - ERC4626Upgradeable.convertToShares(uint256) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/token/ERC20/extensions/ERC4626Upgradeable.sol#54-56)
convertToAssets(uint256) should be declared external:
        - ERC4626Upgradeable.convertToAssets(uint256) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/token/ERC20/extensions/ERC4626Upgradeable.sol#59-61)
mint(uint256,address) should be declared external:
        - ERC4626Upgradeable.mint(uint256,address) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/token/ERC20/extensions/ERC4626Upgradeable.sol#114-121)
withdraw(uint256,address,address) should be declared external:
        - ERC4626Upgradeable.withdraw(uint256,address,address) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/token/ERC20/extensions/ERC4626Upgradeable.sol#124-135)
setAdmin(address) should be declared external:
        - AccessController.setAdmin(address) (jobs/archimedes/brownie/contracts/AccessController.sol#41-47)
getAddressExecutive() should be declared external:
        - AccessController.getAddressExecutive() (jobs/archimedes/brownie/contracts/AccessController.sol#95-97)
initialize() should be declared external:
        - CDPosition.initialize() (jobs/archimedes/brownie/contracts/CDPosition.sol#177-186)
initialize() should be declared external:
        - ParameterStore.initialize() (jobs/archimedes/brownie/contracts/ParameterStore.sol#31-52)
initialize(IERC20MetadataUpgradeable,string,string) should be declared external:
        - VaultOUSD.initialize(IERC20MetadataUpgradeable,string,string) (jobs/archimedes/brownie/contracts/VaultOUSD.sol#88-106)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#public-function-that-could-be-declared-external
```

- Coordinator.sol

AUTOMATED TESTING

```
ERC1967UpgradeUpgradeable._functionDelegateCall(address,bytes) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/proxy/ERC1967/ERC1967UpgradeUpgradeable.sol#198-204) uses delegatecall to a input-controlled function id
        - (success,returndata) = target.delegatecall(data) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/proxy/ERC1967/ERC1967UpgradeUpgradeable.sol#202)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#controlled-delegatecall

Reentrancy in VaultOUSD.archimedesRedeem(uint256,address,address) (jobs/archimedes/brownie/contracts/VaultOUSD.sol#55-78):
        External calls:
        - _takeRebaseFees() (jobs/archimedes/brownie/contracts/VaultOUSD.sol#60)
                - _ousd.transfer(_paramStore.getTreasuryAddress(),feeToCollect) (jobs/archimedes/brownie/contracts/VaultOUSD.sol#138)
        - redeemedAmountInAssets = super.redeem(shares,receiver,owner) (jobs/archimedes/brownie/contracts/VaultOUSD.sol#61)
                - returndata = address(token).functionCall(data,SafeERC20: low-level call failed) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/token/ERC20/utils/SafeERC20Upgradeable.sol#110)
                - SafeERC20Upgradeable.safeTransfer(_asset,receiver,assets) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/token/ERC20/extensions/ERC4626Upgradeable.sol#219)
                - (success,returndata) = target.call{value: value}(data) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/AddressUpgradeable.sol#137)
        External calls sending eth:
        - redeemedAmountInAssets = super.redeem(shares,receiver,owner) (jobs/archimedes/brownie/contracts/VaultOUSD.sol#61)
                - (success,returndata) = target.call{value: value}(data) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/AddressUpgradeable.sol#137)
        State variables written after the call(s):
        - _assetsHandledByArchimedes = 0 (jobs/archimedes/brownie/contracts/VaultOUSD.sol#71)
        - _assetsHandledByArchimedes = _assetsHandledByArchimedes - redeemedAmountInAssets (jobs/archimedes/brownie/contracts/VaultOUSD.sol#73)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#reentrancy-vulnerabilities

Coordinator._borrowUnderNFT(uint256,uint256) (jobs/archimedes/brownie/contracts/Coordinator.sol#201-204) ignores return value by _lvUSD.transfer(_addressExchanger,_amount) (jobs/archimedes/brownie/contracts/Coordinator.sol#202)
VaultOUSD._takeRebaseFees() (jobs/archimedes/brownie/contracts/VaultOUSD.sol#108-140) ignores return value by _ousd.transfer(_paramStore.getTreasuryAddress(),feeToCollect) (jobs/archimedes/brownie/contracts/VaultOUSD.sol#138)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#unchecked-transfer

CDPosition (jobs/archimedes/brownie/contracts/CDPosition.sol#17-199) is an upgradeable contract that does not protect its initiliaze functions: CDPosition.initialize() (jobs/archimedes/brownie/contracts/CDPosition.sol#177-186). Anyone can delete the contract with: UUPSUpgradeable.upgradeTo(address) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/proxy/utils/UUPSUpgradeable.sol#72-75)UUPSUpgradeable.upgradeToAndCall(address,bytes) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/proxy/utils/UUPSUpgradeable.sol#85-88)Coordinator (jobs/archimedes/brownie/contracts/Coordinator.sol#22-226) is an upgradeable contract that does not protect its initiliaze functions: Coordinator.initialize() (jobs/archimedes/brownie/contracts/Coordinator.sol#101-130). Anyone can delete the contract with: UUPSUpgradeable.upgradeTo(address) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/proxy/utils/UUPSUpgradeable.sol#72-75)UUPSUpgradeable.upgradeToAndCall(address,bytes) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/proxy/utils/UUPSUpgradeable.sol#85-88)Exchanger (jobs/archimedes/brownie/contracts/Exchanger.sol#20-448) is an upgradeable contract that does not protect its initiliaze functions: Exchanger.initialize() (jobs/archimedes/brownie/contracts/Exchanger.sol#81-94). Anyone can delete the contract with: UUPSUpgradeable.upgradeTo(address) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/proxy/utils/UUPSUpgradeable.sol#72-75)UUPSUpgradeable.upgradeToAndCall(address,bytes) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/proxy/utils/UUPSUpgradeable.sol#85-88)ParameterStore (jobs/archimedes/brownie/contracts/ParameterStore.sol#10-243) is an upgradeable contract that does not protect its initiliaze functions: ParameterStore.initialize() (jobs/archimedes/brownie/contracts/ParameterStore.sol#31-52). Anyone can delete the contract with: UUPSUpgradeable.upgradeTo(address) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/proxy/utils/UUPSUpgradeable.sol#72-75)UUPSUpgradeable.upgradeToAndCall(address,bytes) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/proxy/utils/UUPSUpgradeable.sol#85-88)VaultOUSD (jobs/archimedes/brownie/contracts/VaultOUSD.sol#19-146) is an upgradeable contract that does not protect its initiliaze functions: VaultOUSD.initialize(IERC20MetadataUpgradeable,string,string) (jobs/archimedes/brownie/contracts/VaultOUSD.sol#88-106). Anyone can delete the contract with: UUPSUpgradeable.upgradeTo(address) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/proxy/utils/UUPSUpgradeable.sol#72-75)UUPSUpgradeable.upgradeToAndCall(address,bytes) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/proxy/utils/UUPSUpgradeable.sol#85-88)Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#unprotected-upgradeable-contract

MathUpgradeable.mulDiv(uint256,uint256,uint256) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/math/MathUpgradeable.sol#55-135) performs a multiplication on the result of a division:
        -denominator = denominator / twos (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/math/MathUpgradeable.sol#102)
        -inverse = (3 * denominator) ^ 2 (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/math/MathUpgradeable.sol#117)
MathUpgradeable.mulDiv(uint256,uint256,uint256) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/math/MathUpgradeable.sol#55-135) performs a multiplication on the result of a division:
        -denominator = denominator / twos (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/math/MathUpgradeable.sol#102)
        -inverse *= 2 - denominator * inverse (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/math/MathUpgradeable.sol#121)
MathUpgradeable.mulDiv(uint256,uint256,uint256) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/math/MathUpgradeable.sol#55-135) performs a multiplication on the result of a division:
        -denominator = denominator / twos (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/math/MathUpgradeable.sol#102)
        -inverse *= 2 - denominator * inverse (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/math/MathUpgradeable.sol#122)
MathUpgradeable.mulDiv(uint256,uint256,uint256) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/math/MathUpgradeable.sol#55-135) performs a multiplication on the result of a division:
        -denominator = denominator / twos (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/math/MathUpgradeable.sol#102)
        -inverse *= 2 - denominator * inverse (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/math/MathUpgradeable.sol#123)
MathUpgradeable.mulDiv(uint256,uint256,uint256) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/math/MathUpgradeable.sol#55-135) performs a multiplication on the result of a division:
        -denominator = denominator / twos (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/math/MathUpgradeable.sol#102)
        -inverse *= 2 - denominator * inverse (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/math/MathUpgradeable.sol#124)
MathUpgradeable.mulDiv(uint256,uint256,uint256) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/math/MathUpgradeable.sol#55-135) performs a multiplication on the result of a division:
        -denominator = denominator / twos (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/math/MathUpgradeable.sol#102)
        -inverse *= 2 - denominator * inverse (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/math/MathUpgradeable.sol#125)
MathUpgradeable.mulDiv(uint256,uint256,uint256) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/math/MathUpgradeable.sol#55-135) performs a multiplication on the result of a division:
        -denominator = denominator / twos (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/math/MathUpgradeable.sol#102)
        -inverse *= 2 - denominator * inverse (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/math/MathUpgradeable.sol#126)
MathUpgradeable.mulDiv(uint256,uint256,uint256) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/math/MathUpgradeable.sol#55-135) performs a multiplication on the result of a division:
        -prod0 = prod0 / twos (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/math/MathUpgradeable.sol#105)
        -result = prod0 * inverse (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/math/MathUpgradeable.sol#132)
ParameterStore.calculateLeverageAllowedForArch(uint256) (jobs/archimedes/brownie/contracts/ParameterStore.sol#218-220) performs a multiplication on the result of a division:
        -(_archTorcRatio / 1000000000000000000) * archAmount (jobs/archimedes/brownie/contracts/ParameterStore.sol#219)
VaultOUSD._takeRebaseFees() (jobs/archimedes/brownie/contracts/VaultOUSD.sol#108-140) performs a multiplication on the result of a division:
        -unhandledRebasePayment = ((totalAssets() - _assetsHandledByArchimedes) / 100) * 100 (jobs/archimedes/brownie/contracts/VaultOUSD.sol#126)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#divide-before-multiply

CDPosition.canDeletePosition(uint256) (jobs/archimedes/brownie/contracts/CDPosition.sol#48-51) uses a dangerous strict equality:
        - require(bool,string)(_nftCDP[nftID].lvUSDBorrowed == 0,lvUSD borrowed must be zero) (jobs/archimedes/brownie/contracts/CDPosition.sol#49)
CDPosition.nftIDMustNotExist(uint256) (jobs/archimedes/brownie/contracts/CDPosition.sol#43-46) uses a dangerous strict equality:
        - require(bool,string)(_nftCDP[nftID].oUSDPrinciple == 0,NFT ID must not exist) (jobs/archimedes/brownie/contracts/CDPosition.sol#44)
ERC4626Upgradeable._convertToAssets(uint256,MathUpgradeable.Rounding) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/token/ERC20/extensions/ERC4626Upgradeable.sol#168-174) uses a dangerous strict equality:
        - (supply == 0) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/token/ERC20/extensions/ERC4626Upgradeable.sol#170-173)
ERC4626Upgradeable._convertToShares(uint256,MathUpgradeable.Rounding) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/token/ERC20/extensions/ERC4626Upgradeable.sol#157-163) uses a dangerous strict equality:
        - (assets == 0 || supply == 0) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/token/ERC20/extensions/ERC4626Upgradeable.sol#159-162)
ERC4626Upgradeable._isVaultCollateralized() (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/token/ERC20/extensions/ERC4626Upgradeable.sol#224-226) uses a dangerous strict equality:
        - totalAssets() > 0 || totalSupply() == 0 (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/token/ERC20/extensions/ERC4626Upgradeable.sol#225)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#dangerous-strict-equalities

Reentrancy in VaultOUSD.archimedesDeposit(uint256,address) (jobs/archimedes/brownie/contracts/VaultOUSD.sol#37-41):
        External calls:
        - _takeRebaseFees() (jobs/archimedes/brownie/contracts/VaultOUSD.sol#38)
                - _ousd.transfer(_paramStore.getTreasuryAddress(),feeToCollect) (jobs/archimedes/brownie/contracts/VaultOUSD.sol#138)
        State variables written after the call(s):
        - _assetsHandledByArchimedes += assets (jobs/archimedes/brownie/contracts/VaultOUSD.sol#39)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#reentrancy-vulnerabilities-1

ERC1967UpgradeUpgradeable._upgradeToAndCallUUPS(address,bytes,bool).slot (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/proxy/ERC1967/ERC1967UpgradeUpgradeable.sol#98) is a local variable never initialized
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#uninitialized-local-variables

ERC1967UpgradeUpgradeable._upgradeToAndCallUUPS(address,bytes,bool) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/proxy/ERC1967/ERC1967UpgradeUpgradeable.sol#87-105) ignores return value by IERC1822ProxiableUpgradeable(newImplementation).proxiableUUID() (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/proxy/ERC1967/ERC1967UpgradeUpgradeable.sol#98-102)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#unused-return

VaultOUSD.initialize(IERC20MetadataUpgradeable,string,string).asset (jobs/archimedes/brownie/contracts/VaultOUSD.sol#89) shadows:
        - ERC4626Upgradeable.asset() (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/token/ERC20/extensions/ERC4626Upgradeable.sol#44-46) (function)
        - IERC4626Upgradeable.asset() (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/interfaces/IERC4626Upgradeable.sol#32) (function)
VaultOUSD.initialize(IERC20MetadataUpgradeable,string,string).name (jobs/archimedes/brownie/contracts/VaultOUSD.sol#90) shadows:
        - ERC20Upgradeable.name() (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/token/ERC20/ERC20Upgradeable.sol#67-69) (function)
        - IERC20MetadataUpgradeable.name() (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/token/ERC20/extensions/IERC20MetadataUpgradeable.sol#17) (function)
VaultOUSD.initialize(IERC20MetadataUpgradeable,string,string).symbol (jobs/archimedes/brownie/contracts/VaultOUSD.sol#91) shadows:
        - ERC20Upgradeable.symbol() (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/token/ERC20/ERC20Upgradeable.sol#75-77) (function)
        - IERC20MetadataUpgradeable.symbol() (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/token/ERC20/extensions/IERC20MetadataUpgradeable.sol#22) (function)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#local-variable-shadowing

VaultOUSD.archimedesDeposit(uint256,address) (jobs/archimedes/brownie/contracts/VaultOUSD.sol#37-41) should emit an event for:
        - _assetsHandledByArchimedes += assets (jobs/archimedes/brownie/contracts/VaultOUSD.sol#39)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#missing-events-arithmetic

CDPosition.setDependencies(address,address).addressVaultOUSD (jobs/archimedes/brownie/contracts/CDPosition.sol#188) lacks a zero-check on :
                - _addressVaultOUSD = addressVaultOUSD (jobs/archimedes/brownie/contracts/CDPosition.sol#189)
CDPosition.setDependencies(address,address).addressParameterStore (jobs/archimedes/brownie/contracts/CDPosition.sol#188) lacks a zero-check on :
                - _addressParameterStore = addressParameterStore (jobs/archimedes/brownie/contracts/CDPosition.sol#190)
Coordinator.setDependencies(address,address,address,address,address,address,address).addressLvUSD (jobs/archimedes/brownie/contracts/Coordinator.sol#39) lacks a zero-check on :
                - _addressLvUSD = addressLvUSD (jobs/archimedes/brownie/contracts/Coordinator.sol#47)
Coordinator.setDependencies(address,address,address,address,address,address,address).addressVaultOUSD (jobs/archimedes/brownie/contracts/Coordinator.sol#40) lacks a zero-check on :
                - _addressVaultOUSD = addressVaultOUSD (jobs/archimedes/brownie/contracts/Coordinator.sol#48)
Coordinator.setDependencies(address,address,address,address,address,address,address).addressCDP (jobs/archimedes/brownie/contracts/Coordinator.sol#41) lacks a zero-check on :
                - _addressCDP = addressCDP (jobs/archimedes/brownie/contracts/Coordinator.sol#49)
Coordinator.setDependencies(address,address,address,address,address,address,address).addressCDP (jobs/archimedes/brownie/contracts/Coordinator.sol#41) lacks a zero-check on :
                - _addressCDP = addressCDP (jobs/archimedes/brownie/contracts/Coordinator.sol#49)
Coordinator.setDependencies(address,address,address,address,address,address,address).addressOUSD (jobs/archimedes/brownie/contracts/Coordinator.sol#42) lacks a zero-check on :
                - _addressOUSD = addressOUSD (jobs/archimedes/brownie/contracts/Coordinator.sol#50)
Coordinator.setDependencies(address,address,address,address,address,address,address).addressExchanger (jobs/archimedes/brownie/contracts/Coordinator.sol#43) lacks a zero-check on :
                - _addressExchanger = addressExchanger (jobs/archimedes/brownie/contracts/Coordinator.sol#51)
Coordinator.setDependencies(address,address,address,address,address,address,address).addressPoolManager (jobs/archimedes/brownie/contracts/Coordinator.sol#45) lacks a zero-check on :
                - _addressPoolManager = addressPoolManager (jobs/archimedes/brownie/contracts/Coordinator.sol#52)
Exchanger.setDependencies(address,address,address,address,address,address,address).addressParameterStore (jobs/archimedes/brownie/contracts/Exchanger.sol#58) lacks a zero-check on :
                - _addressParameterStore = addressParameterStore (jobs/archimedes/brownie/contracts/Exchanger.sol#67)
Exchanger.setDependencies(address,address,address,address,address,address,address).addressCoordinator (jobs/archimedes/brownie/contracts/Exchanger.sol#59) lacks a zero-check on :
                - _addressCoordinator = addressCoordinator (jobs/archimedes/brownie/contracts/Exchanger.sol#68)
Exchanger.setDependencies(address,address,address,address,address,address,address).addressPoolLvUSD3CRV (jobs/archimedes/brownie/contracts/Exchanger.sol#63) lacks a zero-check on :
                - _addressPoolLvUSD3CRV = addressPoolLvUSD3CRV (jobs/archimedes/brownie/contracts/Exchanger.sol#69)
Exchanger.setDependencies(address,address,address,address,address,address,address).addressPoolOUSD3CRV (jobs/archimedes/brownie/contracts/Exchanger.sol#64) lacks a zero-check on :
                - _addressPoolOUSD3CRV = addressPoolOUSD3CRV (jobs/archimedes/brownie/contracts/Exchanger.sol#70)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#missing-zero-address-validation

Variable 'ERC1967UpgradeUpgradeable._upgradeToAndCallUUPS(address,bytes,bool).slot (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/proxy/ERC1967/ERC1967UpgradeUpgradeable.sol#98)' in ERC1967UpgradeUpgradeable._upgradeToAndCallUUPS(address,bytes,bool) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/proxy/ERC1967/ERC1967UpgradeUpgradeable.sol#87-105) potentially used before declaration: require(bool,string)(slot == _IMPLEMENTATION_SLOT,ERC1967Upgrade: unsupported proxiableUUID) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/proxy/ERC1967/ERC1967UpgradeUpgradeable.sol#99)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#pre-declaration-usage-of-local-variables

Reentrancy in ERC4626Upgradeable._deposit(address,address,uint256,uint256) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/token/ERC20/extensions/ERC4626Upgradeable.sol#179-196):
        External calls:
        - SafeERC20Upgradeable.safeTransferFrom(_asset,caller,address(this),assets) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/token/ERC20/extensions/ERC4626Upgradeable.sol#192)
        State variables written after the call(s):
        - _mint(receiver,shares) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/token/ERC20/extensions/ERC4626Upgradeable.sol#193)
                - _balances[account] += amount (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/token/ERC20/ERC20Upgradeable.sol#268)
        - _mint(receiver,shares) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/token/ERC20/extensions/ERC4626Upgradeable.sol#193)
                - _totalSupply += amount (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/token/ERC20/ERC20Upgradeable.sol#267)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#reentrancy-vulnerabilities-2
```

```
Reentrancy in ERC4626Upgradeable._deposit(address,address,uint256,uint256) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/token/ERC20/extensions/ERC4626Upgradeable.sol#179-196):
        External calls:
        - SafeERC20Upgradeable.safeTransferFrom(_asset,caller,address(this),assets) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/token/ERC20/extensions/ERC4626Upgradeable.sol#192)
        Event emitted after the call(s):
        - Deposit(caller,receiver,assets,shares) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/token/ERC20/extensions/ERC4626Upgradeable.sol#195)
        - Transfer(address(0),account,amount) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/token/ERC20/ERC20Upgradeable.sol#269)
            - _mint(receiver,shares) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/token/ERC20/extensions/ERC4626Upgradeable.sol#193)
Reentrancy in ERC4626Upgradeable._withdraw(address,address,address,uint256,uint256) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/token/ERC20/extensions/ERC4626Upgradeable.sol#201-222):
        External calls:
        - SafeERC20Upgradeable.safeTransfer(_asset,receiver,assets) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/token/ERC20/extensions/ERC4626Upgradeable.sol#219)
        Event emitted after the call(s):
        - Withdraw(caller,receiver,owner,assets,shares) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/token/ERC20/extensions/ERC4626Upgradeable.sol#221)
Reentrancy in VaultOUSD.archimedesDeposit(uint256,address) (jobs/archimedes/brownie/contracts/VaultOUSD.sol#37-41):
        External calls:
        - _takeRebaseFees() (jobs/archimedes/brownie/contracts/VaultOUSD.sol#38)
            - _ousd.transfer(_paramStore.getTreasuryAddress(),feeToCollect) (jobs/archimedes/brownie/contracts/VaultOUSD.sol#138)
        - super.deposit(assets,receiver) (jobs/archimedes/brownie/contracts/VaultOUSD.sol#40)
            - SafeERC20Upgradeable.safeTransferFrom(_asset,caller,address(this),assets) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/token/ERC20/extensions/ERC4626Upgradeable.sol#192)
            - returndata = address(token).functionCall(data,SafeERC20: low-level call failed) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/token/ERC20/utils/SafeERC20Upgradeable.sol#110)
            - (success,returndata) = target.call{value: value}(data) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/AddressUpgradeable.sol#137)
        External calls sending eth:
        - super.deposit(assets,receiver) (jobs/archimedes/brownie/contracts/VaultOUSD.sol#40)
            - (success,returndata) = target.call{value: value}(data) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/AddressUpgradeable.sol#137)
        Event emitted after the call(s):
        - Deposit(caller,receiver,assets,shares) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/token/ERC20/extensions/ERC4626Upgradeable.sol#195)
            - super.deposit(assets,receiver) (jobs/archimedes/brownie/contracts/VaultOUSD.sol#40)
        - Transfer(address(0),account,amount) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/token/ERC20/ERC20Upgradeable.sol#269)
            - super.deposit(assets,receiver) (jobs/archimedes/brownie/contracts/VaultOUSD.sol#40)
Reentrancy in VaultOUSD.archimedesRedeem(uint256,address,address) (jobs/archimedes/brownie/contracts/VaultOUSD.sol#55-78):
        External calls:
        - _takeRebaseFees() (jobs/archimedes/brownie/contracts/VaultOUSD.sol#60)
            - _ousd.transfer(_paramStore.getTreasuryAddress(),feeToCollect) (jobs/archimedes/brownie/contracts/VaultOUSD.sol#138)
        - redeemedAmountInAssets = super.redeem(shares,receiver,owner) (jobs/archimedes/brownie/contracts/VaultOUSD.sol#61)
            - returndata = address(token).functionCall(data,SafeERC20: low-level call failed) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/token/ERC20/utils/SafeERC20Upgradeable.sol#110)
            - SafeERC20Upgradeable.safeTransfer(_asset,receiver,assets) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/token/ERC20/extensions/ERC4626Upgradeable.sol#219)
            - (success,returndata) = target.call{value: value}(data) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/AddressUpgradeable.sol#137)
        External calls sending eth:
        - redeemedAmountInAssets = super.redeem(shares,receiver,owner) (jobs/archimedes/brownie/contracts/VaultOUSD.sol#61)
            - (success,returndata) = target.call{value: value}(data) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/AddressUpgradeable.sol#137)
        Event emitted after the call(s):
        - Approval(owner,spender,amount) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/token/ERC20/ERC20Upgradeable.sol#324)
            - redeemedAmountInAssets = super.redeem(shares,receiver,owner) (jobs/archimedes/brownie/contracts/VaultOUSD.sol#61)
        - Transfer(account,address(0),amount) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/token/ERC20/ERC20Upgradeable.sol#297)
            - redeemedAmountInAssets = super.redeem(shares,receiver,owner) (jobs/archimedes/brownie/contracts/VaultOUSD.sol#61)
        - Withdraw(caller,receiver,owner,assets,shares) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/token/ERC20/extensions/ERC4626Upgradeable.sol#221)
            - redeemedAmountInAssets = super.redeem(shares,receiver,owner) (jobs/archimedes/brownie/contracts/VaultOUSD.sol#61)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#reentrancy-vulnerabilities-3

CDPosition.removeSharesFromPosition(uint256,uint256) (jobs/archimedes/brownie/contracts/CDPosition.sol#82-85) uses timestamp for comparisons
        Dangerous comparisons:
        - require(bool,string)(_nftCDP[nftID].shares >= shares,Shares exceed position balance) (jobs/archimedes/brownie/contracts/CDPosition.sol#83)
CDPosition.withdrawOUSDFromPosition(uint256,uint256) (jobs/archimedes/brownie/contracts/CDPosition.sol#116-119) uses timestamp for comparisons
        Dangerous comparisons:
        - require(bool,string)(getOUSDTotalIncludeInterest(nftID) >= oUSDAmountToWithdraw,Insufficient OUSD balance) (jobs/archimedes/brownie/contracts/CDPosition.sol#117)
CDPosition.getOUSDInterestEarned(uint256) (jobs/archimedes/brownie/contracts/CDPosition.sol#126-141) uses timestamp for comparisons
        Dangerous comparisons:
        - _nftCDP[nftID].oUSDTotalWithoutInterest > totalFundsFromPreviewRedeem (jobs/archimedes/brownie/contracts/CDPosition.sol#136)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#block-timestamp

AddressUpgradeable.verifyCallResult(bool,bytes,string) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/AddressUpgradeable.sol#174-194) uses assembly
        - INLINE ASM (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/AddressUpgradeable.sol#186-189)
StorageSlotUpgradeable.getAddressSlot(bytes32) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/StorageSlotUpgradeable.sol#52-57) uses assembly
        - INLINE ASM (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/StorageSlotUpgradeable.sol#54-56)
StorageSlotUpgradeable.getBooleanSlot(bytes32) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/StorageSlotUpgradeable.sol#62-67) uses assembly
        - INLINE ASM (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/StorageSlotUpgradeable.sol#64-66)
StorageSlotUpgradeable.getBytes32Slot(bytes32) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/StorageSlotUpgradeable.sol#72-77) uses assembly
        - INLINE ASM (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/StorageSlotUpgradeable.sol#74-76)
StorageSlotUpgradeable.getUint256Slot(bytes32) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/StorageSlotUpgradeable.sol#82-87) uses assembly
        - INLINE ASM (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/StorageSlotUpgradeable.sol#84-86)
MathUpgradeable.mulDiv(uint256,uint256,uint256) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/math/MathUpgradeable.sol#55-135) uses assembly
        - INLINE ASM (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/math/MathUpgradeable.sol#66-70)
        - INLINE ASM (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/math/MathUpgradeable.sol#86-93)
        - INLINE ASM (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/math/MathUpgradeable.sol#100-109)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#assembly-usage

Different versions of Solidity are used:
        - Version used: ['^0.8.13', '^0.8.0', '^0.8.1', '^0.8.13', '^0.8.2']
        - ^0.8.0 (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/access/AccessControlUpgradeable.sol#4)
        - ^0.8.0 (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/access/IAccessControlUpgradeable.sol#4)
        - ^0.8.0 (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/interfaces/IERC20MetadataUpgradeable.sol#4)
        - ^0.8.0 (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/interfaces/IERC4626Upgradeable.sol#4)
        - ^0.8.0 (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/interfaces/draft-IERC1822Upgradeable.sol#4)
        - ^0.8.2 (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/proxy/ERC1967/ERC1967UpgradeUpgradeable.sol#4)
        - ^0.8.0 (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/proxy/beacon/IBeaconUpgradeable.sol#4)
        - ^0.8.2 (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/proxy/utils/Initializable.sol#4)
        - ^0.8.0 (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/proxy/utils/UUPSUpgradeable.sol#4)
        - ^0.8.0 (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/security/ReentrancyGuardUpgradeable.sol#4)
        - ^0.8.0 (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/token/ERC20/ERC20Upgradeable.sol#4)
        - ^0.8.0 (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/token/ERC20/IERC20Upgradeable.sol#4)
        - ^0.8.0 (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/token/ERC20/extensions/ERC4626Upgradeable.sol#4)
        - ^0.8.0 (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/token/ERC20/extensions/IERC20MetadataUpgradeable.sol#4)
        - ^0.8.0 (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/token/ERC20/extensions/draft-IERC20PermitUpgradeable.sol#4)
        - ^0.8.0 (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/token/ERC20/utils/SafeERC20Upgradeable.sol#4)
        - ^0.8.1 (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/AddressUpgradeable.sol#4)
        - ^0.8.0 (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/ContextUpgradeable.sol#4)
        - ^0.8.0 (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/StorageSlotUpgradeable.sol#4)
        - ^0.8.0 (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/StringsUpgradeable.sol#4)
        - ^0.8.0 (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/introspection/ERC165Upgradeable.sol#4)
        - ^0.8.0 (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/introspection/IERC165Upgradeable.sol#4)
        - ^0.8.0 (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/math/MathUpgradeable.sol#4)
        - 0.8.13 (jobs/archimedes/brownie/contracts/AccessController.sol#2)
        - 0.8.13 (jobs/archimedes/brownie/contracts/CDPosition.sol#2)
        - 0.8.13 (jobs/archimedes/brownie/contracts/Coordinator.sol#2)
        - 0.8.13 (jobs/archimedes/brownie/contracts/Exchanger.sol#2)
        - 0.8.13 (jobs/archimedes/brownie/contracts/ParameterStore.sol#2)
        - 0.8.13 (jobs/archimedes/brownie/contracts/VaultOUSD.sol#2)
        - 0.8.13 (jobs/archimedes/brownie/contracts/interfaces/ICoordinator.sol#2)
        - ^0.8.0 (jobs/archimedes/brownie/contracts/interfaces/ICurveFi.sol#2)
        - ^0.8.13 (jobs/archimedes/brownie/contracts/interfaces/IExchanger.sol#2)
        - ^0.8.0 (jobs/archimedes/brownie/contracts/interfaces/IOUSD.sol#4)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#different-pragma-directives-are-used

AccessControlUpgradeable.__AccessControl_init_unchained() (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/access/AccessControlUpgradeable.sol#54-55) is never used and should be removed
AccessControlUpgradeable._setRoleAdmin(bytes32,bytes32) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/access/AccessControlUpgradeable.sol#220-224) is never used and should be removed
AccessControlUpgradeable._setupRole(bytes32,address) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/access/AccessControlUpgradeable.sol#211-213) is never used and should be removed
AddressUpgradeable.functionCall(address,bytes) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/AddressUpgradeable.sol#85-87) is never used and should be removed
AddressUpgradeable.functionCallWithValue(address,bytes,uint256) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/AddressUpgradeable.sol#114-120) is never used and should be removed
AddressUpgradeable.functionStaticCall(address,bytes) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/AddressUpgradeable.sol#147-149) is never used and should be removed
AddressUpgradeable.functionStaticCall(address,bytes,string) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/AddressUpgradeable.sol#157-166) is never used and should be removed
AddressUpgradeable.sendValue(address,uint256) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/AddressUpgradeable.sol#60-65) is never used and should be removed
ContextUpgradeable.__Context_init() (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/ContextUpgradeable.sol#18-19) is never used and should be removed
ContextUpgradeable.__Context_init_unchained() (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/ContextUpgradeable.sol#21-22) is never used and should be removed
ContextUpgradeable._msgData() (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/ContextUpgradeable.sol#27-29) is never used and should be removed
Coordinator._checkEqualBalanceWithBuffer(uint256,uint256) (jobs/archimedes/brownie/contracts/Coordinator.sol#216-220) is never used and should be removed
ERC165Upgradeable.__ERC165_init() (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/introspection/ERC165Upgradeable.sol#24-25) is never used and should be removed
ERC165Upgradeable.__ERC165_init_unchained() (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/introspection/ERC165Upgradeable.sol#27-28) is never used and should be removed
ERC1967UpgradeUpgradeable.__ERC1967Upgrade_init() (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/proxy/ERC1967/ERC1967UpgradeUpgradeable.sol#21-22) is never used and should be removed
ERC1967UpgradeUpgradeable.__ERC1967Upgrade_init_unchained() (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/proxy/ERC1967/ERC1967UpgradeUpgradeable.sol#24-25) is never used and should be removed
ERC1967UpgradeUpgradeable._changeAdmin(address) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/proxy/ERC1967/ERC1967UpgradeUpgradeable.sol#139-142) is never used and should be removed
ERC1967UpgradeUpgradeable._getAdmin() (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/proxy/ERC1967/ERC1967UpgradeUpgradeable.sol#122-124) is never used and should be removed
ERC1967UpgradeUpgradeable._getBeacon() (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/proxy/ERC1967/ERC1967UpgradeUpgradeable.sol#158-160) is never used and should be removed
ERC1967UpgradeUpgradeable._setAdmin(address) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/proxy/ERC1967/ERC1967UpgradeUpgradeable.sol#129-132) is never used and should be removed
ERC1967UpgradeUpgradeable._setBeacon(address) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/proxy/ERC1967/ERC1967UpgradeUpgradeable.sol#165-172) is never used and should be removed
ERC1967UpgradeUpgradeable._upgradeBeaconToAndCall(address,bytes,bool) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/proxy/ERC1967/ERC1967UpgradeUpgradeable.sol#180-190) is never used and should
 be removed
Initializable._disableInitializers() (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/proxy/utils/Initializable.sol#131-137) is never used and should be removed
MathUpgradeable.average(uint256,uint256) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/math/MathUpgradeable.sol#34-37) is never used and should be removed
MathUpgradeable.ceilDiv(uint256,uint256) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/math/MathUpgradeable.sol#45-48) is never used and should be removed
MathUpgradeable.max(uint256,uint256) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/math/MathUpgradeable.sol#19-21) is never used and should be removed
MathUpgradeable.min(uint256,uint256) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/math/MathUpgradeable.sol#26-28) is never used and should be removed
MathUpgradeable.sqrt(uint256) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/math/MathUpgradeable.sol#158-214) is never used and should be removed
MathUpgradeable.sqrt(uint256,MathUpgradeable.Rounding) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/math/MathUpgradeable.sol#219-225) is never used and should be removed
SafeERC20Upgradeable.safeDecreaseAllowance(IERC20Upgradeable,address,uint256) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/token/ERC20/utils/SafeERC20Upgradeable.sol#70-81) is never used and s
hould be removed
SafeERC20Upgradeable.safePermit(IERC20PermitUpgradeable,address,address,uint256,uint256,uint8,bytes32,bytes32) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/token/ERC20/utils/SafeERC20Upgradeab
le.sol#83-97) is never used and should be removed
StorageSlotUpgradeable.getBytes32Slot(bytes32) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/StorageSlotUpgradeable.sol#72-77) is never used and should be removed
StorageSlotUpgradeable.getUint256Slot(bytes32) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/StorageSlotUpgradeable.sol#82-87) is never used and should be removed
StringsUpgradeable.toHexString(address) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/StringsUpgradeable.sol#72-74) is never used and should be removed
StringsUpgradeable.toHexString(uint256) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/StringsUpgradeable.sol#41-52) is never used and should be removed
StringsUpgradeable.toString(uint256) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/StringsUpgradeable.sol#16-36) is never used and should be removed
UUPSUpgradeable.__UUPSUpgradeable_init_unchained() (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/proxy/utils/UUPSUpgradeable.sol#26-27) is never used and should be removed
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#dead-code
```

```
Pragma version^0.8.0 (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/access/AccessControlUpgradeable.sol#4) allows old versions
Pragma version^0.8.0 (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/access/IAccessControlUpgradeable.sol#4) allows old versions
Pragma version^0.8.0 (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/interfaces/IERC20MetadataUpgradeable.sol#4) allows old versions
Pragma version^0.8.0 (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/interfaces/IERC4626Upgradeable.sol#4) allows old versions
Pragma version^0.8.0 (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/interfaces/draft-IERC1822Upgradeable.sol#4) allows old versions
Pragma version^0.8.2 (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/proxy/ERC1967/ERC1967UpgradeUpgradeable.sol#4) allows old versions
Pragma version^0.8.0 (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/proxy/beacon/IBeaconUpgradeable.sol#4) allows old versions
Pragma version^0.8.2 (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/proxy/utils/Initializable.sol#4) allows old versions
Pragma version^0.8.0 (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/proxy/utils/UUPSUpgradeable.sol#4) allows old versions
Pragma version^0.8.0 (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/security/ReentrancyGuardUpgradeable.sol#4) allows old versions
Pragma version^0.8.0 (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/token/ERC20/ERC20Upgradeable.sol#4) allows old versions
Pragma version^0.8.0 (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/token/ERC20/IERC20Upgradeable.sol#4) allows old versions
Pragma version^0.8.0 (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/token/ERC20/extensions/ERC4626Upgradeable.sol#4) allows old versions
Pragma version^0.8.0 (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/token/ERC20/extensions/IERC20MetadataUpgradeable.sol#4) allows old versions
Pragma version^0.8.0 (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/token/ERC20/extensions/draft-IERC20PermitUpgradeable.sol#4) allows old versions
Pragma version^0.8.0 (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/token/ERC20/utils/SafeERC20Upgradeable.sol#4) allows old versions
Pragma version^0.8.1 (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/AddressUpgradeable.sol#4) allows old versions
Pragma version^0.8.0 (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/ContextUpgradeable.sol#4) allows old versions
Pragma version^0.8.0 (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/StorageSlotUpgradeable.sol#4) allows old versions
Pragma version^0.8.0 (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/StringsUpgradeable.sol#4) allows old versions
Pragma version^0.8.0 (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/introspection/ERC165Upgradeable.sol#4) allows old versions
Pragma version^0.8.0 (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/introspection/IERC165Upgradeable.sol#4) allows old versions
Pragma version^0.8.0 (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/math/MathUpgradeable.sol#4) allows old versions
Pragma version0.8.13 (jobs/archimedes/brownie/contracts/AccessController.sol#2) necessitates a version too recent to be trusted. Consider deploying with 0.6.12/0.7.6/0.8.7
Pragma version0.8.13 (jobs/archimedes/brownie/contracts/CDPosition.sol#2) necessitates a version too recent to be trusted. Consider deploying with 0.6.12/0.7.6/0.8.7
Pragma version0.8.13 (jobs/archimedes/brownie/contracts/Coordinator.sol#2) necessitates a version too recent to be trusted. Consider deploying with 0.6.12/0.7.6/0.8.7
Pragma version0.8.13 (jobs/archimedes/brownie/contracts/Exchanger.sol#2) necessitates a version too recent to be trusted. Consider deploying with 0.6.12/0.7.6/0.8.7
Pragma version0.8.13 (jobs/archimedes/brownie/contracts/ParameterStore.sol#2) necessitates a version too recent to be trusted. Consider deploying with 0.6.12/0.7.6/0.8.7
Pragma version0.8.13 (jobs/archimedes/brownie/contracts/VaultOUSD.sol#2) necessitates a version too recent to be trusted. Consider deploying with 0.6.12/0.7.6/0.8.7
Pragma version0.8.13 (jobs/archimedes/brownie/contracts/interfaces/ICoordinator.sol#2) 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 (jobs/archimedes/brownie/contracts/interfaces/ICurveFi.sol#2) allows old versions
Pragma version^0.8.13 (jobs/archimedes/brownie/contracts/interfaces/IExchanger.sol#2) 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 (jobs/archimedes/brownie/contracts/interfaces/IOUSD.sol#4) allows old versions
solc-0.8.13 is not recommended for deployment
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#incorrect-versions-of-solidity

Low level call in ERC1967UpgradeUpgradeable._functionDelegateCall(address,bytes) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/proxy/ERC1967/ERC1967UpgradeUpgradeable.sol#198-204):
        - (success,returndata) = target.delegatecall(data) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/proxy/ERC1967/ERC1967UpgradeUpgradeable.sol#202)
Low level call in AddressUpgradeable.sendValue(address,uint256) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/AddressUpgradeable.sol#60-65):
        - (success) = recipient.call{value: amount}() (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/AddressUpgradeable.sol#63)
Low level call in AddressUpgradeable.functionCallWithValue(address,bytes,uint256,string) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/AddressUpgradeable.sol#128-139):
        - (success,returndata) = target.call{value: value}(data) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/AddressUpgradeable.sol#137)
Low level call in AddressUpgradeable.functionStaticCall(address,bytes,string) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/AddressUpgradeable.sol#157-166):
        - (success,returndata) = target.staticcall(data) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/AddressUpgradeable.sol#164)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#low-level-calls

Function AccessControlUpgradeable.__AccessControl_init() (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/access/AccessControlUpgradeable.sol#51-52) is not in mixedCase
Function AccessControlUpgradeable.__AccessControl_init_unchained() (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/access/AccessControlUpgradeable.sol#54-55) is not in mixedCase
Variable AccessControlUpgradeable.__gap (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/access/AccessControlUpgradeable.sol#259) is not in mixedCase
Function ERC1967UpgradeUpgradeable.__ERC1967Upgrade_init() (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/proxy/ERC1967/ERC1967UpgradeUpgradeable.sol#21-22) is not in mixedCase
Function ERC1967UpgradeUpgradeable.__ERC1967Upgrade_init_unchained() (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/proxy/ERC1967/ERC1967UpgradeUpgradeable.sol#24-25) is not in mixedCase
Variable ERC1967UpgradeUpgradeable.__gap (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/proxy/ERC1967/ERC1967UpgradeUpgradeable.sol#211) is not in mixedCase
Function UUPSUpgradeable.__UUPSUpgradeable_init() (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/proxy/utils/UUPSUpgradeable.sol#23-24) is not in mixedCase
Function UUPSUpgradeable.__UUPSUpgradeable_init_unchained() (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/proxy/utils/UUPSUpgradeable.sol#26-27) is not in mixedCase
Variable UUPSUpgradeable.__gap (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/proxy/utils/UUPSUpgradeable.sol#107) is not in mixedCase
Variable UUPSUpgradeable.__self (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/proxy/utils/UUPSUpgradeable.sol#29) is not in mixedCase
Function ReentrancyGuardUpgradeable.__ReentrancyGuard_init() (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/security/ReentrancyGuardUpgradeable.sol#40-42) is not in mixedCase
Function ReentrancyGuardUpgradeable.__ReentrancyGuard_init_unchained() (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/security/ReentrancyGuardUpgradeable.sol#44-46) is not in mixedCase
Variable ReentrancyGuardUpgradeable.__gap (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/security/ReentrancyGuardUpgradeable.sol#74) is not in mixedCase
Function ERC20Upgradeable.__ERC20_init(string,string) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/token/ERC20/ERC20Upgradeable.sol#55-57) is not in mixedCase
Function ERC20Upgradeable.__ERC20_init_unchained(string,string) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/token/ERC20/ERC20Upgradeable.sol#59-62) is not in mixedCase
Variable ERC20Upgradeable.__gap (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/token/ERC20/ERC20Upgradeable.sol#304) is not in mixedCase
Function ERC4626Upgradeable.__ERC4626_init(IERC20MetadataUpgradeable) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/token/ERC20/extensions/ERC4626Upgradeable.sol#35-37) is not in mixedCase
Function ERC4626Upgradeable.__ERC4626_init_unchained(IERC20MetadataUpgradeable) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/token/ERC20/extensions/ERC4626Upgradeable.sol#39-41) is not in mixe
dCase
Variable ERC4626Upgradeable.__gap (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/token/ERC20/extensions/ERC4626Upgradeable.sol#233) is not in mixedCase
Function IERC20PermitUpgradeable.DOMAIN_SEPARATOR() (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/token/ERC20/extensions/draft-IERC20PermitUpgradeable.sol#59) is not in mixedCase
Function ContextUpgradeable.__Context_init() (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/ContextUpgradeable.sol#18-19) is not in mixedCase
Function ContextUpgradeable.__Context_init_unchained() (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/ContextUpgradeable.sol#21-22) is not in mixedCase
Variable ContextUpgradeable.__gap (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/ContextUpgradeable.sol#36) is not in mixedCase
Function ERC165Upgradeable.__ERC165_init() (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/introspection/ERC165Upgradeable.sol#24-25) is not in mixedCase
Function ERC165Upgradeable.__ERC165_init_unchained() (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/introspection/ERC165Upgradeable.sol#27-28) is not in mixedCase
Variable ERC165Upgradeable.__gap (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/introspection/ERC165Upgradeable.sol#41) is not in mixedCase
Variable CDPosition._nftCDP (jobs/archimedes/brownie/contracts/CDPosition.sol#10) is not in mixedCase
Variable CDPosition._addressVaultOUSD (jobs/archimedes/brownie/contracts/CDPosition.sol#32) is not in mixedCase
Variable CDPosition._addressParameterStore (jobs/archimedes/brownie/contracts/CDPosition.sol#33) is not in mixedCase
Variable CDPosition._vault (jobs/archimedes/brownie/contracts/CDPosition.sol#34) is not in mixedCase
Variable CDPosition._parameterStore (jobs/archimedes/brownie/contracts/CDPosition.sol#35) is not in mixedCase
Parameter Coordinator.depositCollateralUnderNFT(uint256,uint256)._nftId (jobs/archimedes/brownie/contracts/Coordinator.sol#73) is not in mixedCase
Parameter Coordinator.depositCollateralUnderNFT(uint256,uint256)._amountInOUSD (jobs/archimedes/brownie/contracts/Coordinator.sol#73) is not in mixedCase
Parameter Coordinator.withdrawCollateralUnderNFT(uint256,uint256,address)._nftId (jobs/archimedes/brownie/contracts/Coordinator.sol#82) is not in mixedCase
Parameter Coordinator.withdrawCollateralUnderNFT(uint256,uint256,address)._amount (jobs/archimedes/brownie/contracts/Coordinator.sol#83) is not in mixedCase
Parameter Coordinator.withdrawCollateralUnderNFT(uint256,uint256,address)._to (jobs/archimedes/brownie/contracts/Coordinator.sol#84) is not in mixedCase
Parameter Coordinator.borrowUnderNFT(uint256,uint256)._nftId (jobs/archimedes/brownie/contracts/Coordinator.sol#89) is not in mixedCase
Parameter Coordinator.borrowUnderNFT(uint256,uint256)._amount (jobs/archimedes/brownie/contracts/Coordinator.sol#89) is not in mixedCase
Parameter Coordinator.repayUnderNFT(uint256,uint256)._nftId (jobs/archimedes/brownie/contracts/Coordinator.sol#93) is not in mixedCase
Parameter Coordinator.repayUnderNFT(uint256,uint256)._amountLvUSDToRepay (jobs/archimedes/brownie/contracts/Coordinator.sol#93) is not in mixedCase
Parameter Coordinator.getLeveragedOUSD(uint256,uint256)._nftId (jobs/archimedes/brownie/contracts/Coordinator.sol#97) is not in mixedCase
Parameter Coordinator.getLeveragedOUSD(uint256,uint256)._amountToLeverage (jobs/archimedes/brownie/contracts/Coordinator.sol#97) is not in mixedCase
Parameter Coordinator.unwindLeveragedOUSD(uint256,address)._nftId (jobs/archimedes/brownie/contracts/Coordinator.sol#123) is not in mixedCase
Parameter Coordinator.unwindLeveragedOUSD(uint256,address)._userAddress (jobs/archimedes/brownie/contracts/Coordinator.sol#123) is not in mixedCase
Parameter Coordinator.getPositionExpireTime(uint256)._nftId (jobs/archimedes/brownie/contracts/Coordinator.sol#167) is not in mixedCase
Variable Coordinator._addressLvUSD (jobs/archimedes/brownie/contracts/Coordinator.sol#24) is not in mixedCase
Variable Coordinator._addressVaultOUSD (jobs/archimedes/brownie/contracts/Coordinator.sol#25) is not in mixedCase
Variable Coordinator._addressCDP (jobs/archimedes/brownie/contracts/Coordinator.sol#26) is not in mixedCase
Variable Coordinator._addressOUSD (jobs/archimedes/brownie/contracts/Coordinator.sol#27) is not in mixedCase
Variable Coordinator._addressExchanger (jobs/archimedes/brownie/contracts/Coordinator.sol#28) is not in mixedCase
Variable Coordinator._addressPoolManager (jobs/archimedes/brownie/contracts/Coordinator.sol#29) is not in mixedCase
Variable Coordinator._vault (jobs/archimedes/brownie/contracts/Coordinator.sol#31) is not in mixedCase
grantRole(bytes32,address) should be declared external:
        - AccessControlUpgradeable.grantRole(bytes32,address) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/access/AccessControlUpgradeable.sol#150-152)
revokeRole(bytes32,address) should be declared external:
        - AccessControlUpgradeable.revokeRole(bytes32,address) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/access/AccessControlUpgradeable.sol#165-167)
renounceRole(bytes32,address) should be declared external:
        - AccessControlUpgradeable.renounceRole(bytes32,address) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/access/AccessControlUpgradeable.sol#185-189)
        - AccessController.renounceRole(bytes32,address) (jobs/archimedes/brownie/contracts/AccessController.sol#62-69)
name() should be declared external:
        - ERC20Upgradeable.name() (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/token/ERC20/ERC20Upgradeable.sol#67-69)
symbol() should be declared external:
        - ERC20Upgradeable.symbol() (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/token/ERC20/ERC20Upgradeable.sol#75-77)
transfer(address,uint256) should be declared external:
        - ERC20Upgradeable.transfer(address,uint256) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/token/ERC20/ERC20Upgradeable.sol#118-122)
approve(address,uint256) should be declared external:
        - ERC20Upgradeable.approve(address,uint256) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/token/ERC20/ERC20Upgradeable.sol#141-145)
transferFrom(address,address,uint256) should be declared external:
        - ERC20Upgradeable.transferFrom(address,address,uint256) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/token/ERC20/ERC20Upgradeable.sol#163-172)
increaseAllowance(address,uint256) should be declared external:
        - ERC20Upgradeable.increaseAllowance(address,uint256) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/token/ERC20/ERC20Upgradeable.sol#186-190)
decreaseAllowance(address,uint256) should be declared external:
        - ERC20Upgradeable.decreaseAllowance(address,uint256) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/token/ERC20/ERC20Upgradeable.sol#206-215)
asset() should be declared external:
        - ERC4626Upgradeable.asset() (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/token/ERC20/extensions/ERC4626Upgradeable.sol#44-46)
convertToShares(uint256) should be declared external:
        - ERC4626Upgradeable.convertToShares(uint256) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/token/ERC20/extensions/ERC4626Upgradeable.sol#54-56)
convertToAssets(uint256) should be declared external:
        - ERC4626Upgradeable.convertToAssets(uint256) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/token/ERC20/extensions/ERC4626Upgradeable.sol#59-61)
mint(uint256,address) should be declared external:
        - ERC4626Upgradeable.mint(uint256,address) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/token/ERC20/extensions/ERC4626Upgradeable.sol#114-121)
withdraw(uint256,address,address) should be declared external:
        - ERC4626Upgradeable.withdraw(uint256,address,address) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/token/ERC20/extensions/ERC4626Upgradeable.sol#124-135)
setAdmin(address) should be declared external:
        - AccessController.setAdmin(address) (jobs/archimedes/brownie/contracts/AccessController.sol#41-47)
getAddressExecutive() should be declared external:
        - AccessController.getAddressExecutive() (jobs/archimedes/brownie/contracts/AccessController.sol#95-97)
initialize() should be declared external:
        - CDPosition.initialize() (jobs/archimedes/brownie/contracts/CDPosition.sol#177-186)
initialize() should be declared external:
        - Coordinator.initialize() (jobs/archimedes/brownie/contracts/Coordinator.sol#181-190)
initialize() should be declared external:
        - Exchanger.initialize() (jobs/archimedes/brownie/contracts/Exchanger.sol#81-94)
initialize() should be declared external:
        - ParameterStore.initialize() (jobs/archimedes/brownie/contracts/ParameterStore.sol#31-52)
initialize(IERC20MetadataUpgradeable,string,string) should be declared external:
        - VaultOUSD.initialize(IERC20MetadataUpgradeable,string,string) (jobs/archimedes/brownie/contracts/VaultOUSD.sol#88-106)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#public-function-that-could-be-declared-external
```

- Exchanger.sol

AUTOMATED TESTING

```
ERC1967UpgradeUpgradeable._functionDelegateCall(address,bytes) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/proxy/ERC1967/ERC1967UpgradeUpgradeable.sol#198-204) uses delegatecall to a input-co
ntrolled function id
        - (success,returndata) = target.delegatecall(data) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/proxy/ERC1967/ERC1967UpgradeUpgradeable.sol#202)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#controlled-delegatecall

Exchanger (jobs/archimedes/brownie/contracts/Exchanger.sol#20-448) is an upgradeable contract that does not protect its initiliaze functions: Exchanger.initialize() (jobs/archimedes/brownie/contracts/Exchanger.sol#81-94). Anyone c
an delete the contract with: UUPSUpgradeable.upgradeTo(address) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/proxy/utils/UUPSUpgradeable.sol#72-75)UUPSUpgradeable.upgradeToAndCall(address,byte
s) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/proxy/utils/UUPSUpgradeable.sol#85-88)ParameterStore (jobs/archimedes/brownie/contracts/ParameterStore.sol#10-243) is an upgradeable contract th
at does not protect its initiliaze functions: ParameterStore.initialize() (jobs/archimedes/brownie/contracts/ParameterStore.sol#31-52). Anyone can delete the contract with: UUPSUpgradeable.upgradeTo(address) (.brownie/packages/Ope
nZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/proxy/utils/UUPSUpgradeable.sol#72-75)UUPSUpgradeable.upgradeToAndCall(address,bytes) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/
proxy/utils/UUPSUpgradeable.sol#85-88)Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#unprotected-upgradeable-contract

ParameterStore.calculateLeverageAllowedForArch(uint256) (jobs/archimedes/brownie/contracts/ParameterStore.sol#218-220) performs a multiplication on the result of a division:
        -(_archToLevRatio / 100000000000000000) * archAmount (jobs/archimedes/brownie/contracts/ParameterStore.sol#219)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#divide-before-multiply

ERC1967UpgradeUpgradeable._upgradeToAndCallUUPS(address,bytes,bool).slot (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/proxy/ERC1967/ERC1967UpgradeUpgradeable.sol#98) is a local variable never
initialized
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#uninitialized-local-variables

ERC1967UpgradeUpgradeable._upgradeToAndCallUUPS(address,bytes,bool) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/proxy/ERC1967/ERC1967UpgradeUpgradeable.sol#87-105) ignores return value by IER
C1822ProxiableUpgradeable(newImplementation).proxiableUUID() (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/proxy/ERC1967/ERC1967UpgradeUpgradeable.sol#98-102)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#unused-return

Exchanger.setDependencies(address,address,address,address,address,address).addressParameterStore (jobs/archimedes/brownie/contracts/Exchanger.sol#58) lacks a zero-check on :
        - _addressParameterStore = addressParameterStore (jobs/archimedes/brownie/contracts/Exchanger.sol#67)
Exchanger.setDependencies(address,address,address,address,address,address).addressCoordinator (jobs/archimedes/brownie/contracts/Exchanger.sol#59) lacks a zero-check on :
        - _addressCoordinator = addressCoordinator (jobs/archimedes/brownie/contracts/Exchanger.sol#68)
Exchanger.setDependencies(address,address,address,address,address,address).addressPoolLvUSD3CRV (jobs/archimedes/brownie/contracts/Exchanger.sol#63) lacks a zero-check on :
        - _addressPoolLvUSD3CRV = addressPoolLvUSD3CRV (jobs/archimedes/brownie/contracts/Exchanger.sol#69)
Exchanger.setDependencies(address,address,address,address,address,address).addressPoolUSD3CRV (jobs/archimedes/brownie/contracts/Exchanger.sol#64) lacks a zero-check on :
        - _addressPoolUSD3CRV = addressPoolUSD3CRV (jobs/archimedes/brownie/contracts/Exchanger.sol#70)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#missing-zero-address-validation

Variable 'ERC1967UpgradeUpgradeable._upgradeToAndCallUUPS(address,bytes,bool).slot (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/proxy/ERC1967/ERC1967UpgradeUpgradeable.sol#98)' in ERC1967Upgra
deUpgradeable._upgradeToAndCallUUPS(address,bytes,bool) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/proxy/ERC1967/ERC1967UpgradeUpgradeable.sol#87-105) potentially used before declaration: re
quire(bool,string)(slot == _IMPLEMENTATION_SLOT,ERC1967Upgrade: unsupported proxiableUUID) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/proxy/ERC1967/ERC1967UpgradeUpgradeable.sol#99)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#pre-declaration-usage-of-local-variables

AddressUpgradeable.verifyCallResult(bool,bytes,string) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/AddressUpgradeable.sol#174-194) uses assembly
        - INLINE ASM (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/AddressUpgradeable.sol#186-189)
StorageSlotUpgradeable.getAddressSlot(bytes32) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/StorageSlotUpgradeable.sol#52-57) uses assembly
        - INLINE ASM (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/StorageSlotUpgradeable.sol#54-56)
StorageSlotUpgradeable.getBooleanSlot(bytes32) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/StorageSlotUpgradeable.sol#62-67) uses assembly
        - INLINE ASM (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/StorageSlotUpgradeable.sol#64-66)
StorageSlotUpgradeable.getBytes32Slot(bytes32) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/StorageSlotUpgradeable.sol#72-77) uses assembly
        - INLINE ASM (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/StorageSlotUpgradeable.sol#74-76)
StorageSlotUpgradeable.getUint256Slot(bytes32) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/StorageSlotUpgradeable.sol#82-87) uses assembly
        - INLINE ASM (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/StorageSlotUpgradeable.sol#84-86)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#assembly-usage

Different versions of Solidity are used:
        - Version used: ['^0.8.13', '^0.8.0', '^0.8.1', '^0.8.13', '^0.8.2']
        - ^0.8.0 (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/access/AccessControlUpgradeable.sol#4)
        - ^0.8.0 (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/access/IAccessControlUpgradeable.sol#4)
        - ^0.8.0 (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/interfaces/draft-IERC1822Upgradeable.sol#4)
        - ^0.8.2 (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/proxy/ERC1967/ERC1967UpgradeUpgradeable.sol#4)
        - ^0.8.0 (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/proxy/beacon/IBeaconUpgradeable.sol#4)
        - ^0.8.2 (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/proxy/utils/Initializable.sol#4)
        - ^0.8.0 (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/proxy/utils/UUPSUpgradeable.sol#4)
        - ^0.8.0 (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/security/ReentrancyGuardUpgradeable.sol#4)
        - ^0.8.0 (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/token/ERC20/IERC20Upgradeable.sol#4)
        - ^0.8.0 (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/token/ERC20/extensions/draft-IERC20PermitUpgradeable.sol#4)
        - ^0.8.0 (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/token/ERC20/utils/SafeERC20Upgradeable.sol#4)
        - ^0.8.1 (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/AddressUpgradeable.sol#4)
        - ^0.8.0 (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/ContextUpgradeable.sol#4)
        - ^0.8.0 (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/StorageSlotUpgradeable.sol#4)
        - ^0.8.0 (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/StringsUpgradeable.sol#4)
        - ^0.8.0 (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/introspection/ERC165Upgradeable.sol#4)
        - ^0.8.0 (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/introspection/IERC165Upgradeable.sol#4)
        - 0.8.13 (jobs/archimedes/brownie/contracts/AccessController.sol#2)
        - 0.8.13 (jobs/archimedes/brownie/contracts/Exchanger.sol#2)
        - 0.8.13 (jobs/archimedes/brownie/contracts/ParameterStore.sol#2)
        - ^0.8.0 (jobs/archimedes/brownie/contracts/interfaces/ICurveFi.sol#2)
        - ^0.8.13 (jobs/archimedes/brownie/contracts/interfaces/IExchanger.sol#2)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#different-pragma-directives-are-used

AccessControlUpgradeable.__AccessControl_init_unchained() (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/access/AccessControlUpgradeable.sol#54-55) is never used and should be removed
AccessControlUpgradeable._setRoleAdmin(bytes32,bytes32) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/access/AccessControlUpgradeable.sol#220-224) is never used and should be removed
AccessControlUpgradeable._toString(uint256) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/access/AccessControlUpgradeable.sol#213-218) is never used and should be removed
AddressUpgradeable.functionCall(address,bytes) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/AddressUpgradeable.sol#85-87) is never used and should be removed
AddressUpgradeable.functionCallWithValue(address,bytes,uint256) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/AddressUpgradeable.sol#114-120) is never used and should be removed
AddressUpgradeable.functionStaticCall(address,bytes) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/AddressUpgradeable.sol#147-149) is never used and should be removed
AddressUpgradeable.functionStaticCall(address,bytes,string) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/AddressUpgradeable.sol#157-166) is never used and should be removed
AddressUpgradeable.sendValue(address,uint256) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/AddressUpgradeable.sol#60-65) is never used and should be removed
ContextUpgradeable.__Context_init() (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/ContextUpgradeable.sol#18-19) is never used and should be removed
ContextUpgradeable.__Context_init_unchained() (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/ContextUpgradeable.sol#21-22) is never used and should be removed
ContextUpgradeable._msgData() (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/ContextUpgradeable.sol#27-29) is never used and should be removed
ERC165Upgradeable.__ERC165_init() (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/introspection/ERC165Upgradeable.sol#24-25) is never used and should be removed
ERC165Upgradeable.__ERC165_init_unchained() (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/introspection/ERC165Upgradeable.sol#27-28) is never used and should be removed
ERC1967UpgradeUpgradeable.__ERC1967Upgrade_init() (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/proxy/ERC1967/ERC1967UpgradeUpgradeable.sol#21-22) is never used and should be removed
ERC1967UpgradeUpgradeable.__ERC1967Upgrade_init_unchained() (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/proxy/ERC1967/ERC1967UpgradeUpgradeable.sol#24-25) is never used and should be removed
ERC1967UpgradeUpgradeable._changeAdmin(address) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/proxy/ERC1967/ERC1967UpgradeUpgradeable.sol#139-142) is never used and should be removed
ERC1967UpgradeUpgradeable._getAdmin() (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/proxy/ERC1967/ERC1967UpgradeUpgradeable.sol#122-124) is never used and should be removed
ERC1967UpgradeUpgradeable._getBeacon() (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/proxy/ERC1967/ERC1967UpgradeUpgradeable.sol#158-160) is never used and should be removed
ERC1967UpgradeUpgradeable._setAdmin(address) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/proxy/ERC1967/ERC1967UpgradeUpgradeable.sol#129-132) is never used and should be removed
ERC1967UpgradeUpgradeable._setBeacon(address) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/proxy/ERC1967/ERC1967UpgradeUpgradeable.sol#165-172) is never used and should be removed
ERC1967UpgradeUpgradeable._upgradeBeaconToAndCall(address,bytes,bool) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/proxy/ERC1967/ERC1967UpgradeUpgradeable.sol#180-190) is never used and should
 be removed
Pragma version^0.8.0 (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/access/AccessControlUpgradeable.sol#4) allows old versions
Pragma version^0.8.0 (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/access/IAccessControlUpgradeable.sol#4) allows old versions
Pragma version^0.8.0 (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/interfaces/draft-IERC1822Upgradeable.sol#4) allows old versions
Pragma version^0.8.2 (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/proxy/ERC1967/ERC1967UpgradeUpgradeable.sol#4) allows old versions
Pragma version^0.8.0 (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/proxy/beacon/IBeaconUpgradeable.sol#4) allows old versions
Pragma version^0.8.2 (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/proxy/utils/Initializable.sol#4) allows old versions
Pragma version^0.8.0 (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/proxy/utils/UUPSUpgradeable.sol#4) allows old versions
Pragma version^0.8.0 (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/security/ReentrancyGuardUpgradeable.sol#4) allows old versions
Pragma version^0.8.0 (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/token/ERC20/IERC20Upgradeable.sol#4) allows old versions
Pragma version^0.8.0 (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/token/ERC20/extensions/draft-IERC20PermitUpgradeable.sol#4) allows old versions
Pragma version^0.8.0 (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/token/ERC20/utils/SafeERC20Upgradeable.sol#4) allows old versions
Pragma version^0.8.1 (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/AddressUpgradeable.sol#4) allows old versions
Pragma version^0.8.0 (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/ContextUpgradeable.sol#4) allows old versions
Pragma version^0.8.0 (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/StorageSlotUpgradeable.sol#4) allows old versions
Pragma version^0.8.0 (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/StringsUpgradeable.sol#4) allows old versions
Pragma version^0.8.0 (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/introspection/ERC165Upgradeable.sol#4) allows old versions
Pragma version^0.8.0 (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/introspection/IERC165Upgradeable.sol#4) allows old versions
Pragma version0.8.13 (jobs/archimedes/brownie/contracts/AccessController.sol#2) necessitates a version too recent to be trusted. Consider deploying with 0.6.12/0.7.6/0.8.7
Pragma version0.8.13 (jobs/archimedes/brownie/contracts/Exchanger.sol#2) necessitates a version too recent to be trusted. Consider deploying with 0.6.12/0.7.6/0.8.7
Pragma version0.8.13 (jobs/archimedes/brownie/contracts/ParameterStore.sol#2) 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 (jobs/archimedes/brownie/contracts/interfaces/ICurveFi.sol#2) allows old versions
Pragma version^0.8.13 (jobs/archimedes/brownie/contracts/interfaces/IExchanger.sol#2) necessitates a version too recent to be trusted. Consider deploying with 0.6.12/0.7.6/0.8.7
solc-0.8.13 is not recommended for deployment
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#incorrect-versions-of-solidity

Low level call in ERC1967UpgradeUpgradeable._functionDelegateCall(address,bytes) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/proxy/ERC1967/ERC1967UpgradeUpgradeable.sol#198-204):
        - (success,returndata) = target.delegatecall(data) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/proxy/ERC1967/ERC1967UpgradeUpgradeable.sol#202)
Low level call in AddressUpgradeable.sendValue(address,uint256) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/AddressUpgradeable.sol#60-65):
        - (success) = recipient.call{value: amount}() (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/AddressUpgradeable.sol#63)
Low level call in AddressUpgradeable.functionCallWithValue(address,bytes,uint256,string) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/AddressUpgradeable.sol#128-139):
        - (success,returndata) = target.call{value: value}(data) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/AddressUpgradeable.sol#137)
Low level call in AddressUpgradeable.functionStaticCall(address,bytes,string) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/AddressUpgradeable.sol#157-166):
        - (success,returndata) = target.staticcall(data) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/AddressUpgradeable.sol#164)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#low-level-calls

Function AccessControlUpgradeable.__AccessControl_init() (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/access/AccessControlUpgradeable.sol#51-52) is not in MixedCase
Function AccessControlUpgradeable.__AccessControl_init_unchained() (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/access/AccessControlUpgradeable.sol#54-55) is not in MixedCase
Variable AccessControlUpgradeable.__gap (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/access/AccessControlUpgradeable.sol#259) is not in MixedCase
Function ERC1967UpgradeUpgradeable.__ERC1967Upgrade_init() (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/proxy/ERC1967/ERC1967UpgradeUpgradeable.sol#21-22) is not in MixedCase
Function ERC1967UpgradeUpgradeable.__ERC1967Upgrade_init_unchained() (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/proxy/ERC1967/ERC1967UpgradeUpgradeable.sol#24-25) is not in MixedCase
Variable ERC1967UpgradeUpgradeable.__gap (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/proxy/ERC1967/ERC1967UpgradeUpgradeable.sol#211) is not in MixedCase
Function UUPSUpgradeable.__UUPSUpgradeable_init() (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/proxy/utils/UUPSUpgradeable.sol#23-24) is not in MixedCase
Function UUPSUpgradeable.__UUPSUpgradeable_init_unchained() (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/proxy/utils/UUPSUpgradeable.sol#26-27) is not in MixedCase
Variable UUPSUpgradeable.__gap (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/proxy/utils/UUPSUpgradeable.sol#107) is not in MixedCase
Variable UUPSUpgradeable.__self (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/proxy/utils/UUPSUpgradeable.sol#29) is not in MixedCase
Function ReentrancyGuardUpgradeable.__ReentrancyGuard_init() (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/security/ReentrancyGuardUpgradeable.sol#40-42) is not in MixedCase
Function ReentrancyGuardUpgradeable.__ReentrancyGuard_init_unchained() (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/security/ReentrancyGuardUpgradeable.sol#44-46) is not in MixedCase
Variable ReentrancyGuardUpgradeable.__gap (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/security/ReentrancyGuardUpgradeable.sol#74) is not in MixedCase
Function IERC20PermitUpgradeable.DOMAIN_SEPARATOR() (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/token/ERC20/extensions/draft-IERC20PermitUpgradeable.sol#59) is not in MixedCase
Function ContextUpgradeable.__Context_init() (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/ContextUpgradeable.sol#18-19) is not in MixedCase
Function ContextUpgradeable.__Context_init_unchained() (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/ContextUpgradeable.sol#21-22) is not in MixedCase
Variable ContextUpgradeable.__gap (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/ContextUpgradeable.sol#36) is not in MixedCase
Function ERC165Upgradeable.__ERC165_init() (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/introspection/ERC165Upgradeable.sol#24-25) is not in MixedCase
```

```
Function ICurveFiCurve.remove_liquidity(uint256,uint256[2],address) (jobs/archimedes/brownie/contracts/interfaces/ICurveFi.sol#122-126) is not in mixedCase
Parameter ICurveFiCurve.remove_liquidity(uint256,uint256[2],address)._burn_amount (jobs/archimedes/brownie/contracts/interfaces/ICurveFi.sol#123) is not in mixedCase
Parameter ICurveFiCurve.remove_liquidity(uint256,uint256[2],address)._min_amounts (jobs/archimedes/brownie/contracts/interfaces/ICurveFi.sol#124) is not in mixedCase
Function ICurveFiCurve.remove_liquidity_imbalance(uint256[2],uint256) (jobs/archimedes/brownie/contracts/interfaces/ICurveFi.sol#128) is not in mixedCase
Parameter ICurveFiCurve.remove_liquidity_imbalance(uint256[2],uint256)._max_burn_amount (jobs/archimedes/brownie/contracts/interfaces/ICurveFi.sol#128) is not in mixedCase
Function ICurveFiCurve.remove_liquidity_imbalance(uint256[2],uint256,address) (jobs/archimedes/brownie/contracts/interfaces/ICurveFi.sol#130-134) is not in mixedCase
Parameter ICurveFiCurve.remove_liquidity_imbalance(uint256[2],uint256,address)._max_burn_amount (jobs/archimedes/brownie/contracts/interfaces/ICurveFi.sol#132) is not in mixedCase
Function ICurveFiCurve.calc_withdraw_one_coin(uint256,int128) (jobs/archimedes/brownie/contracts/interfaces/ICurveFi.sol#136) is not in mixedCase
Parameter ICurveFiCurve.calc_withdraw_one_coin(uint256,int128)._burn_amount (jobs/archimedes/brownie/contracts/interfaces/ICurveFi.sol#136) is not in mixedCase
Function ICurveFiCurve.calc_withdraw_one_coin(uint256,int128,bool) (jobs/archimedes/brownie/contracts/interfaces/ICurveFi.sol#138-142) is not in mixedCase
Parameter ICurveFiCurve.calc_withdraw_one_coin(uint256,int128,bool)._burn_amount (jobs/archimedes/brownie/contracts/interfaces/ICurveFi.sol#139) is not in mixedCase
Function ICurveFiCurve.remove_liquidity_one_coin(uint256,int128,uint256) (jobs/archimedes/brownie/contracts/interfaces/ICurveFi.sol#144-148) is not in mixedCase
Parameter ICurveFiCurve.remove_liquidity_one_coin(uint256,int128,uint256)._burn_amount (jobs/archimedes/brownie/contracts/interfaces/ICurveFi.sol#145) is not in mixedCase
Parameter ICurveFiCurve.remove_liquidity_one_coin(uint256,int128,uint256)._min_received (jobs/archimedes/brownie/contracts/interfaces/ICurveFi.sol#147) is not in mixedCase
Function ICurveFiCurve.remove_liquidity_one_coin(uint256,int128,uint256,address) (jobs/archimedes/brownie/contracts/interfaces/ICurveFi.sol#150-155) is not in mixedCase
Parameter ICurveFiCurve.remove_liquidity_one_coin(uint256,int128,uint256,address)._burn_amount (jobs/archimedes/brownie/contracts/interfaces/ICurveFi.sol#151) is not in mixedCase
Parameter ICurveFiCurve.remove_liquidity_one_coin(uint256,int128,uint256,address)._min_received (jobs/archimedes/brownie/contracts/interfaces/ICurveFi.sol#153) is not in mixedCase
Function ICurveFiCurve.ramp_A(uint256,uint256) (jobs/archimedes/brownie/contracts/interfaces/ICurveFi.sol#157) is not in mixedCase
Parameter ICurveFiCurve.ramp_A(uint256,uint256)._future_A (jobs/archimedes/brownie/contracts/interfaces/ICurveFi.sol#157) is not in mixedCase
Parameter ICurveFiCurve.ramp_A(uint256,uint256)._future_time (jobs/archimedes/brownie/contracts/interfaces/ICurveFi.sol#157) is not in mixedCase
Function ICurveFiCurve.stop_ramp_A() (jobs/archimedes/brownie/contracts/interfaces/ICurveFi.sol#159) is not in mixedCase
Function ICurveFiCurve.admin_balances(uint256) (jobs/archimedes/brownie/contracts/interfaces/ICurveFi.sol#161) is not in mixedCase
Function ICurveFiCurve.withdraw_admin_fees() (jobs/archimedes/brownie/contracts/interfaces/ICurveFi.sol#163) is not in mixedCase
Function ICurveFiCurve.block_timestamp_last() (jobs/archimedes/brownie/contracts/interfaces/ICurveFi.sol#173) is not in mixedCase
Function ICurveFiCurve.initial_A() (jobs/archimedes/brownie/contracts/interfaces/ICurveFi.sol#175) is not in mixedCase
Function ICurveFiCurve.future_A() (jobs/archimedes/brownie/contracts/interfaces/ICurveFi.sol#177) is not in mixedCase
Function ICurveFiCurve.initial_A_time() (jobs/archimedes/brownie/contracts/interfaces/ICurveFi.sol#179) is not in mixedCase
Function ICurveFiCurve.future_A_time() (jobs/archimedes/brownie/contracts/interfaces/ICurveFi.sol#181) is not in mixedCase
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#conformance-to-solidity-naming-conventions

ParameterStore.changeArchToLevRatio(uint256) (jobs/archimedes/brownie/contracts/ParameterStore.sol#99-103) uses literals with too many digits:
        - require(bool,string)(newArchToLevRatio < 500000000000000000000000 && newArchToLevRatio >= 10000000000000000,new ArchToLevRatio out of range) (jobs/archimedes/brownie/contracts/ParameterStore.sol#100)
ParameterStore.changeMinPositionCollateral(uint256) (jobs/archimedes/brownie/contracts/ParameterStore.sol#111-115) uses literals with too many digits:
        - require(bool,string)(newMinPositionCollateral < (1000000000000000000000000) && newMinPositionCollateral > (10000000000000000),New min colleateral out of range) (jobs/archimedes/brownie/contracts/ParameterStore.sol#112)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#too-many-digits

UUPSUpgradeable.__gap (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/proxy/utils/UUPSUpgradeable.sol#107) is never used in Exchanger (jobs/archimedes/brownie/contracts/Exchanger.sol#20-448)
UUPSUpgradeable.__gap (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/proxy/utils/UUPSUpgradeable.sol#107) is never used in ParameterStore (jobs/archimedes/brownie/contracts/ParameterStore.sol#10-243)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#unused-state-variable

grantRole(bytes32,address) should be declared external:
        - AccessControlUpgradeable.grantRole(bytes32,address) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/access/AccessControlUpgradeable.sol#150-152)
revokeRole(bytes32,address) should be declared external:
        - AccessControlUpgradeable.revokeRole(bytes32,address) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/access/AccessControlUpgradeable.sol#165-167)
renounceRole(bytes32,address) should be declared external:
        - AccessControlUpgradeable.renounceRole(bytes32,address) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/access/AccessControlUpgradeable.sol#185-189)
        - AccessController.renounceRole(bytes32,address) (jobs/archimedes/brownie/contracts/AccessController.sol#62-69)
setAdmin(address) should be declared external:
        - AccessController.setAdmin(address) (jobs/archimedes/brownie/contracts/AccessController.sol#41-47)
getAddressExecutive() should be declared external:
        - AccessController.getAddressExecutive() (jobs/archimedes/brownie/contracts/AccessController.sol#95-97)
initialize() should be declared external:
        - Exchanger.initialize() (jobs/archimedes/brownie/contracts/Exchanger.sol#81-94)
initialize() should be declared external:
        - ParameterStore.initialize() (jobs/archimedes/brownie/contracts/ParameterStore.sol#31-52)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#public-function-that-could-be-declared-external
```

- LeverageEngine.sol

```
ERC1967UpgradeUpgradeable._functionDelegateCall(address,bytes) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/proxy/ERC1967/ERC1967UpgradeUpgradeable.sol#198-204) uses delegatecall to a input-co
ntrolled function id
        - (success,returndata) = target.delegatecall(data) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/proxy/ERC1967/ERC1967UpgradeUpgradeable.sol#202)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#controlled-delegatecall

LeverageEngine._burnArchTokenForPosition(address,uint256) (jobs/archimedes/brownie/contracts/LeverageEngine.sol#135-138) ignores return value by _archToken.transferFrom(sender,treasuryAddress,archAmount) (jobs/archimedes/brownie/c
ontracts/LeverageEngine.sol#137)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#unchecked-transfer

LeverageEngine (jobs/archimedes/brownie/contracts/LeverageEngine.sol#17-156) is an upgradeable contract that does not protect its initiliaze functions: LeverageEngine.initialize() (jobs/archimedes/brownie/contracts/LeverageEngine.
sol#123-132). Anyone can delete the contract with: UUPSUpgradeable.upgradeTo(address) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/proxy/utils/UUPSUpgradeable.sol#72-75)UUPSUpgradeable.upgrade
ToAndCall(address,bytes) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/proxy/utils/UUPSUpgradeable.sol#85-88)ParameterStore (jobs/archimedes/brownie/contracts/ParameterStore.sol#10-243) is an u
pgradeable contract that does not protect its initiliaze functions: ParameterStore.initialize() (jobs/archimedes/brownie/contracts/ParameterStore.sol#31-52). Anyone can delete the contract with: UUPSUpgradeable.upgradeTo(address)
(.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/proxy/utils/UUPSUpgradeable.sol#72-75)UUPSUpgradeable.upgradeToAndCall(address,bytes) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/
contracts-upgradeable/proxy/utils/UUPSUpgradeable.sol#85-88)PositionToken (jobs/archimedes/brownie/contracts/PositionToken.sol#15-132) is an upgradeable contract that does not protect its initiliaze functions: PositionToken.initia
lize() (jobs/archimedes/brownie/contracts/PositionToken.sol#47-59). Anyone can delete the contract with: UUPSUpgradeable.upgradeTo(address) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/proxy/u
tils/UUPSUpgradeable.sol#72-75)UUPSUpgradeable.upgradeToAndCall(address,bytes) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/proxy/utils/UUPSUpgradeable.sol#85-88)Reference: https://github.com/
crytic/slither/wiki/Detector-Documentation#unprotected-upgradeable-contract

Math.mulDiv(uint256,uint256,uint256) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts/utils/math/Math.sol#55-135) performs a multiplication on the result of a division:
        -denominator = denominator / twos (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts/utils/math/Math.sol#102)
        -inverse = (3 * denominator) ^ 2 (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts/utils/math/Math.sol#117)
Math.mulDiv(uint256,uint256,uint256) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts/utils/math/Math.sol#55-135) performs a multiplication on the result of a division:
        -denominator = denominator / twos (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts/utils/math/Math.sol#102)
        -inverse *= 2 - denominator * inverse (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts/utils/math/Math.sol#121)
Math.mulDiv(uint256,uint256,uint256) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts/utils/math/Math.sol#55-135) performs a multiplication on the result of a division:
        -denominator = denominator / twos (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts/utils/math/Math.sol#102)
        -inverse *= 2 - denominator * inverse (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts/utils/math/Math.sol#122)
Math.mulDiv(uint256,uint256,uint256) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts/utils/math/Math.sol#55-135) performs a multiplication on the result of a division:
        -denominator = denominator / twos (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts/utils/math/Math.sol#102)
        -inverse *= 2 - denominator * inverse (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts/utils/math/Math.sol#123)
Math.mulDiv(uint256,uint256,uint256) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts/utils/math/Math.sol#55-135) performs a multiplication on the result of a division:
        -denominator = denominator / twos (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts/utils/math/Math.sol#102)
        -inverse *= 2 - denominator * inverse (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts/utils/math/Math.sol#124)
Math.mulDiv(uint256,uint256,uint256) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts/utils/math/Math.sol#55-135) performs a multiplication on the result of a division:
        -denominator = denominator / twos (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts/utils/math/Math.sol#102)
        -inverse *= 2 - denominator * inverse (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts/utils/math/Math.sol#125)
Math.mulDiv(uint256,uint256,uint256) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts/utils/math/Math.sol#55-135) performs a multiplication on the result of a division:
        -denominator = denominator / twos (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts/utils/math/Math.sol#102)
        -inverse *= 2 - denominator * inverse (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts/utils/math/Math.sol#126)
Math.mulDiv(uint256,uint256,uint256) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts/utils/math/Math.sol#55-135) performs a multiplication on the result of a division:
        -prod0 = prod0 / twos (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts/utils/math/Math.sol#105)
        -result = prod0 * inverse (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts/utils/math/Math.sol#132)
LeverageEngine.createLeveragedPosition(uint256,uint256,uint256) (jobs/archimedes/brownie/contracts/LeverageEngine.sol#72-107) performs a multiplication on the result of a division:
        -archNeededToBurn = (_parameterStore.calculateArchNeededForLeverage(lvUSDAmount) / 10000) * 10000 (jobs/archimedes/brownie/contracts/LeverageEngine.sol#91)
ParameterStore.calculateLeverageAllowedForArch(uint256) (jobs/archimedes/brownie/contracts/ParameterStore.sol#218-228) performs a multiplication on the result of a division:
        -(_archToLevRatio / 10000000000000000) * archAmount (jobs/archimedes/brownie/contracts/ParameterStore.sol#219)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#divide-before-multiply
```

```
ERC1967UpgradeUpgradeable._upgradeToAndCallUUPS(address,bytes,bool) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/proxy/ERC1967/ERC1967UpgradeUpgradeable.sol#87-105) ignores return value by IER
C1822ProxiableUpgradeable(newImplementation).proxiableUUID() (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/proxy/ERC1967/ERC1967UpgradeUpgradeable.sol#98-102)
ERC721Upgradeable._checkOnERC721Received(address,address,uint256,bytes) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/token/ERC721/ERC721Upgradeable.sol#399-421) ignores return value by IERC721
ReceiverUpgradeable(to).onERC721Received(_msgSender(),from,tokenId,data) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/token/ERC721/ERC721Upgradeable.sol#406-417)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#unused-return

ERC20Permit.constructor(string).name (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts/token/ERC20/extensions/draft-ERC20Permit.sol#44) shadows:
        - ERC20.name() (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts/token/ERC20/ERC20.sol#62-64) (function)
        - IERC20Metadata.name() (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts/token/ERC20/extensions/IERC20Metadata.sol#17) (function)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#local-variable-shadowing

LeverageEngine.setDependencies(address,address,address,address,address).addressCoordinator (jobs/archimedes/brownie/contracts/LeverageEngine.sol#44) lacks a zero-check on :
        - _addressCoordinator = addressCoordinator (jobs/archimedes/brownie/contracts/LeverageEngine.sol#50)
LeverageEngine.setDependencies(address,address,address,address,address).addressPositionToken (jobs/archimedes/brownie/contracts/LeverageEngine.sol#45) lacks a zero-check on :
        - _addressPositionToken = addressPositionToken (jobs/archimedes/brownie/contracts/LeverageEngine.sol#52)
LeverageEngine.setDependencies(address,address,address,address,address).addressParameterStore (jobs/archimedes/brownie/contracts/LeverageEngine.sol#46) lacks a zero-check on :
        - addressParameterStore = addressParameterStore (jobs/archimedes/brownie/contracts/LeverageEngine.sol#54)
LeverageEngine.setDependencies(address,address,address,address,address).addressArchToken (jobs/archimedes/brownie/contracts/LeverageEngine.sol#47) lacks a zero-check on :
        - _addressArchToken = addressArchToken (jobs/archimedes/brownie/contracts/LeverageEngine.sol#56)
LeverageEngine.setDependencies(address,address,address,address,address).addressOUSD (jobs/archimedes/brownie/contracts/LeverageEngine.sol#48) lacks a zero-check on :
        - _addressOUSD = addressOUSD (jobs/archimedes/brownie/contracts/LeverageEngine.sol#58)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#missing-zero-address-validation

Variable 'ERC1967UpgradeUpgradeable._upgradeToAndCallUUPS(address,bytes,bool).slot (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/proxy/ERC1967/ERC1967UpgradeUpgradeable.sol#98)' in ERC1967Upgra
deUpgradeable._upgradeToAndCallUUPS(address,bytes,bool) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/proxy/ERC1967/ERC1967UpgradeUpgradeable.sol#87-105) potentially used before declaration: re
quire(bool,string)(slot == _IMPLEMENTATION_SLOT,ERC1967Upgrade: unsupported proxiableUUID) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/proxy/ERC1967/ERC1967UpgradeUpgradeable.sol#99)
Variable 'ERC721Upgradeable._checkOnERC721Received(address,address,uint256,bytes).retval (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/token/ERC721/ERC721Upgradeable.sol#406)' in ERC721Upgradea
ble._checkOnERC721Received(address,address,uint256,bytes) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/token/ERC721/ERC721Upgradeable.sol#399-421) potentially used before declaration: retval =
= IERC721ReceiverUpgradeable.onERC721Received.selector (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/token/ERC721/ERC721Upgradeable.sol#407)
Variable 'ERC721Upgradeable._checkOnERC721Received(address,address,uint256,bytes).reason (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/token/ERC721/ERC721Upgradeable.sol#408)' in ERC721Upgradea
ble._checkOnERC721Received(address,address,uint256,bytes) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/token/ERC721/ERC721Upgradeable.sol#399-421) potentially used before declaration: reason.l
ength == 0 (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/token/ERC721/ERC721Upgradeable.sol#409)
Variable 'ERC721Upgradeable._checkOnERC721Received(address,address,uint256,bytes).reason (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/token/ERC721/ERC721Upgradeable.sol#408)' in ERC721Upgradea
ble._checkOnERC721Received(address,address,uint256,bytes) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/token/ERC721/ERC721Upgradeable.sol#399-421) potentially used before declaration: revert(u
int256,uint256)(32 + reason,mload(uint256)(reason)) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/token/ERC721/ERC721Upgradeable.sol#414)
Variable 'ERC20Votes._moveVotingPower(address,address,uint256).newWeight (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts/token/ERC20/extensions/ERC20Votes.sol#215)' in ERC20Votes._moveVotingPower(address,add
ress,uint256) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts/token/ERC20/extensions/ERC20Votes.sol#208-224) potentially used before declaration: (oldWeight,newWeight) = _writeCheckpoint(_checkpoints[dst],_a
dd,amount) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts/token/ERC20/extensions/ERC20Votes.sol#220)
Variable 'ERC20Votes._moveVotingPower(address,address,uint256).oldWeight (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts/token/ERC20/extensions/ERC20Votes.sol#215)' in ERC20Votes._moveVotingPower(address,add
ress,uint256) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts/token/ERC20/extensions/ERC20Votes.sol#208-224) potentially used before declaration: (oldWeight,newWeight) = _writeCheckpoint(_checkpoints[dst],_a
dd,amount) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts/token/ERC20/extensions/ERC20Votes.sol#220)
Variable 'ECDSA.tryRecover(bytes32,bytes).r (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts/utils/cryptography/ECDSA.sol#62)' in ECDSA.tryRecover(bytes32,bytes) (.brownie/packages/OpenZeppelin/openzeppelin-c
ontracts@4.7.0/contracts/utils/cryptography/ECDSA.sol#57-88) potentially used before declaration: r = mload(uint256)(signature + 0x20) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts/utils/cryptography/ECDSA
.sol#81)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#pre-declaration-usage-of-local-variables

Reentrancy in PositionToken.safeMint(address) (jobs/archimedes/brownie/contracts/PositionToken.sol#34-41):
        External calls:
        - _safeMint(to,positionTokenId) (jobs/archimedes/brownie/contracts/PositionToken.sol#37)
                - IERC721ReceiverUpgradeable(to).onERC721Received(_msgSender(),from,tokenId,data) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/token/ERC721/ERC721Upgradeable.sol#406-417)
        State variables written after the call(s):
        - _setApprovalForAll(to,getAddressExecutive(),true) (jobs/archimedes/brownie/contracts/PositionToken.sol#38)
                - _operatorApprovals[owner][operator] = approved (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/token/ERC721/ERC721Upgradeable.sol#378)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#reentrancy-vulnerabilities-2

Reentrancy in LeverageEngine.createLeveragedPosition(uint256,uint256,uint256) (jobs/archimedes/brownie/contracts/LeverageEngine.sol#72-107):
        External calls:
        - _burnArchTokenForPosition(msg.sender,archNeededToBurn) (jobs/archimedes/brownie/contracts/LeverageEngine.sol#97)
                - _archToken.transferFrom(sender,treasuryAddress,archAmount) (jobs/archimedes/brownie/contracts/LeverageEngine.sol#137)
        - positionTokenId = _positionToken.safeMint(msg.sender) (jobs/archimedes/brownie/contracts/LeverageEngine.sol#98)
        - _ousd.safeTransferFrom(msg.sender,_addressCoordinator,ousdPrinciple) (jobs/archimedes/brownie/contracts/LeverageEngine.sol#99)
        - _coordinator.depositCollateralUnderNFT(positionTokenId,ousdPrinciple) (jobs/archimedes/brownie/contracts/LeverageEngine.sol#100)
        - _coordinator.getLeveragedOUSD(positionTokenId,lvlUSDAmount) (jobs/archimedes/brownie/contracts/LeverageEngine.sol#101)
        Event emitted after the call(s):
        - PositionCreated(msg.sender,positionTokenId,ousdPrinciple,lvlUSDAmount,archNeededToBurn,psoitionExpireTime) (jobs/archimedes/brownie/contracts/LeverageEngine.sol#104)
Reentrancy in PositionToken.safeMint(address) (jobs/archimedes/brownie/contracts/PositionToken.sol#34-41):
        External calls:
        - _safeMint(to,positionTokenId) (jobs/archimedes/brownie/contracts/PositionToken.sol#37)
                - IERC721ReceiverUpgradeable(to).onERC721Received(_msgSender(),from,tokenId,data) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/token/ERC721/ERC721Upgradeable.sol#406-417)
        Event emitted after the call(s):
        - ApprovalForAll(owner,operator,approved) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/token/ERC721/ERC721Upgradeable.sol#379)
                - _setApprovalForAll(to,getAddressExecutive(),true) (jobs/archimedes/brownie/contracts/PositionToken.sol#38)
        - NFTCreated(positionTokenId,to) (jobs/archimedes/brownie/contracts/PositionToken.sol#39)
Reentrancy in LeverageEngine.unwindLeveragedPosition(uint256) (jobs/archimedes/brownie/contracts/LeverageEngine.sol#116-121):
        External calls:
        - _positionToken.burn(positionTokenId) (jobs/archimedes/brownie/contracts/LeverageEngine.sol#118)
        - positionWindfall = _coordinator.unwindLeveragedOUSD(positionTokenId,msg.sender) (jobs/archimedes/brownie/contracts/LeverageEngine.sol#119)
        Event emitted after the call(s):
        - PositionUnwind(msg.sender,positionTokenId,positionWindfall) (jobs/archimedes/brownie/contracts/LeverageEngine.sol#120)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#reentrancy-vulnerabilities-3

ERC20Votes.delegateBySig(address,uint256,uint256,uint8,bytes32,bytes32) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts/token/ERC20/extensions/ERC20Votes.sol#133-150) uses timestamp for comparisons
        Dangerous comparisons:
        - require(bool,string)(block.timestamp <= expiry,ERC20Votes: signature expired) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts/token/ERC20/extensions/ERC20Votes.sol#141)
ERC20Permit.permit(address,address,uint256,uint256,uint8,bytes32,bytes32) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts/token/ERC20/extensions/draft-ERC20Permit.sol#49-68) uses timestamp for comparisons
        Dangerous comparisons:
        - require(bool,string)(block.timestamp <= deadline,ERC20Permit: expired deadline) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts/token/ERC20/extensions/draft-ERC20Permit.sol#58)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#block-timestamp

ERC721Upgradeable._checkOnERC721Received(address,address,uint256,bytes) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/token/ERC721/ERC721Upgradeable.sol#399-421) uses assembly
        - INLINE ASM (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/token/ERC721/ERC721Upgradeable.sol#413-415)
AddressUpgradeable.verifyCallResult(bool,bytes,string) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/AddressUpgradeable.sol#174-194) uses assembly
        - INLINE ASM (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/AddressUpgradeable.sol#186-189)
StorageSlotUpgradeable.getAddressSlot(bytes32) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/StorageSlotUpgradeable.sol#52-57) uses assembly
        - INLINE ASM (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/StorageSlotUpgradeable.sol#54-56)
StorageSlotUpgradeable.getBooleanSlot(bytes32) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/StorageSlotUpgradeable.sol#62-67) uses assembly
        - INLINE ASM (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/StorageSlotUpgradeable.sol#64-66)
StorageSlotUpgradeable.getBytes32Slot(bytes32) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/StorageSlotUpgradeable.sol#72-77) uses assembly
        - INLINE ASM (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/StorageSlotUpgradeable.sol#74-76)
StorageSlotUpgradeable.getUint256Slot(bytes32) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/StorageSlotUpgradeable.sol#82-87) uses assembly
        - INLINE ASM (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/StorageSlotUpgradeable.sol#84-86)
ECDSA.tryRecover(bytes32,bytes) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts/utils/cryptography/ECDSA.sol#57-88) uses assembly
        - INLINE ASM (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts/utils/cryptography/ECDSA.sol#68-72)
        - INLINE ASM (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts/utils/cryptography/ECDSA.sol#80-83)
Math.mulDiv(uint256,uint256,uint256) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts/utils/math/Math.sol#55-135) uses assembly
        - INLINE ASM (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts/utils/math/Math.sol#66-70)
        - INLINE ASM (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts/utils/math/Math.sol#86-93)
        - INLINE ASM (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts/utils/math/Math.sol#100-109)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#assembly-usage

Different versions of Solidity are used:
        - Version used: ['0.8.13', '^0.8.0', '^0.8.1', '^0.8.2']
        - ^0.8.0 (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/access/AccessControlUpgradeable.sol#4)
        - ^0.8.0 (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/access/IAccessControlUpgradeable.sol#4)
        - ^0.8.0 (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/interfaces/draft-IERC1822Upgradeable.sol#4)
        - ^0.8.2 (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/proxy/ERC1967/ERC1967UpgradeUpgradeable.sol#4)
        - ^0.8.0 (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/proxy/beacon/IBeaconUpgradeable.sol#4)
        - ^0.8.2 (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/proxy/utils/Initializable.sol#4)
        - ^0.8.0 (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/proxy/utils/UUPSUpgradeable.sol#4)
        - ^0.8.0 (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/security/PausableUpgradeable.sol#4)
        - ^0.8.0 (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/security/ReentrancyGuardUpgradeable.sol#4)
        - ^0.8.0 (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/token/ERC20/IERC20Upgradeable.sol#4)
        - ^0.8.0 (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/token/ERC20/extensions/draft-IERC20PermitUpgradeable.sol#4)
        - ^0.8.0 (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/token/ERC20/utils/SafeERC20Upgradeable.sol#4)
        - ^0.8.0 (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/token/ERC721/ERC721Upgradeable.sol#4)
        - ^0.8.0 (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/token/ERC721/IERC721ReceiverUpgradeable.sol#4)
        - ^0.8.0 (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/token/ERC721/IERC721Upgradeable.sol#4)
        - ^0.8.0 (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/token/ERC721/extensions/ERC721BurnableUpgradeable.sol#4)
        - ^0.8.0 (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/token/ERC721/extensions/ERC721EnumerableUpgradeable.sol#4)
        - ^0.8.0 (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/token/ERC721/extensions/IERC721EnumerableUpgradeable.sol#4)
        - ^0.8.0 (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/token/ERC721/extensions/IERC721MetadataUpgradeable.sol#4)
        - ^0.8.1 (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/AddressUpgradeable.sol#4)
        - ^0.8.0 (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/ContextUpgradeable.sol#4)
        - ^0.8.0 (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/CountersUpgradeable.sol#4)
        - ^0.8.0 (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/StorageSlotUpgradeable.sol#4)
        - ^0.8.0 (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/StringsUpgradeable.sol#4)
        - ^0.8.0 (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/introspection/ERC165Upgradeable.sol#4)
        - ^0.8.0 (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/introspection/IERC165Upgradeable.sol#4)
        - ^0.8.0 (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts/access/AccessControl.sol#4)
        - ^0.8.0 (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts/access/IAccessControl.sol#4)
        - ^0.8.0 (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts/governance/utils/IVotes.sol#3)
        - ^0.8.0 (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts/token/ERC20/ERC20.sol#4)
        - ^0.8.0 (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts/token/ERC20/IERC20.sol#4)
        - ^0.8.0 (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts/token/ERC20/extensions/ERC20Votes.sol#4)
        - ^0.8.0 (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts/token/ERC20/extensions/IERC20Metadata.sol#4)
        - ^0.8.0 (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts/token/ERC20/extensions/draft-ERC20Permit.sol#4)
```

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

AccessControl._setRoleAdmin(bytes32,bytes32) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts/access/AccessControl.sol#214-218) is never used and should be removed
AccessControl._setupRole(bytes32,address) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts/access/AccessControl.sol#205-207) is never used and should be removed
AccessControlUpgradeable.__AccessControl_init_unchained() (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/access/AccessControlUpgradeable.sol#54-55) is never used and should be removed
AccessControlUpgradeable._setRoleAdmin(bytes32,bytes32) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/access/AccessControlUpgradeable.sol#220-224) is never used and should be removed
AccessControlUpgradeable._setupRole(bytes32,address) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/access/AccessControlUpgradeable.sol#211-213) is never used and should be removed
AddressUpgradeable.functionCall(address,bytes) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/AddressUpgradeable.sol#85-87) is never used and should be removed
AddressUpgradeable.functionCallWithValue(address,bytes,uint256) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/AddressUpgradeable.sol#114-120) is never used and should be removed
AddressUpgradeable.functionStaticCall(address,bytes) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/AddressUpgradeable.sol#147-149) is never used and should be removed
AddressUpgradeable.functionStaticCall(address,bytes,string) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/AddressUpgradeable.sol#157-166) is never used and should be removed
AddressUpgradeable.sendValue(address,uint256) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/AddressUpgradeable.sol#60-65) is never used and should be removed
ArchToken._burn(address,uint256) (jobs/archimedes/brownie/contracts/ArchToken.sol#33-35) is never used and should be removed
BasicAccessController._requireAdmin() (jobs/archimedes/brownie/contracts/BasicAccessController.sol#67-69) is never used and should be removed
Context._msgData() (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts/utils/Context.sol#21-23) is never used and should be removed
ContextUpgradeable.__Context_init() (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/ContextUpgradeable.sol#18-19) is never used and should be removed
ContextUpgradeable.__Context_init_unchained() (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/ContextUpgradeable.sol#21-22) is never used and should be removed
ContextUpgradeable._msgData() (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/ContextUpgradeable.sol#27-29) is never used and should be removed
Counters.decrement(Counters.Counter) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts/utils/Counters.sol#32-38) is never used and should be removed
Counters.reset(Counters.Counter) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts/utils/Counters.sol#40-42) is never used and should be removed
CountersUpgradeable.decrement(CountersUpgradeable.Counter) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/CountersUpgradeable.sol#32-38) is never used and should be removed
CountersUpgradeable.reset(CountersUpgradeable.Counter) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/CountersUpgradeable.sol#40-42) is never used and should be removed
ECDSA.recover(bytes32,bytes) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts/utils/cryptography/ECDSA.sol#104-108) is never used and should be removed
ECDSA.recover(bytes32,bytes32,bytes32) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts/utils/cryptography/ECDSA.sol#132-140) is never used and should be removed
ECDSA.toEthSignedMessageHash(bytes) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts/utils/cryptography/ECDSA.sol#216-218) is never used and should be removed
ECDSA.toEthSignedMessageHash(bytes32) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts/utils/cryptography/ECDSA.sol#202-206) is never used and should be removed
ECDSA.tryRecover(bytes32,bytes) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts/utils/cryptography/ECDSA.sol#57-88) is never used and should be removed
ECDSA.tryRecover(bytes32,bytes32,bytes32) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts/utils/cryptography/ECDSA.sol#117-125) is never used and should be removed
ERC165Upgradeable.__ERC165_init() (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/introspection/ERC165Upgradeable.sol#24-25) is never used and should be removed
ERC165Upgradeable.__ERC165_init_unchained() (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/introspection/ERC165Upgradeable.sol#27-28) is never used and should be removed
ERC1967UpgradeUpgradeable.__ERC1967Upgrade_init() (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/proxy/ERC1967/ERC1967UpgradeUpgradeable.sol#21-22) is never used and should be removed
ERC1967UpgradeUpgradeable.__ERC1967Upgrade_init_unchained() (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/proxy/ERC1967/ERC1967UpgradeUpgradeable.sol#24-25) is never used and should be removed
ERC1967UpgradeUpgradeable._changeAdmin(address) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/proxy/ERC1967/ERC1967UpgradeUpgradeable.sol#139-142) is never used and should be removed
ERC1967UpgradeUpgradeable._getAdmin() (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/proxy/ERC1967/ERC1967UpgradeUpgradeable.sol#122-124) is never used and should be removed
ERC1967UpgradeUpgradeable._getBeacon() (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/proxy/ERC1967/ERC1967UpgradeUpgradeable.sol#158-160) is never used and should be removed
ERC1967UpgradeUpgradeable._setAdmin(address) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/proxy/ERC1967/ERC1967UpgradeUpgradeable.sol#129-132) is never used and should be removed
ERC1967UpgradeUpgradeable._setBeacon(address) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/proxy/ERC1967/ERC1967UpgradeUpgradeable.sol#165-172) is never used and should be removed
ERC1967UpgradeUpgradeable._upgradeBeaconToAndCall(address,bytes,bool) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/proxy/ERC1967/ERC1967UpgradeUpgradeable.sol#188-190) is never used and should be removed
ERC20._burn(address,uint256) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts/token/ERC20/ERC20.sol#280-295) is never used and should be removed
ERC20Votes._add(uint256,uint256) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts/token/ERC20/extensions/ERC20Votes.sol#242-244) is never used and should be removed
ERC20Votes._burn(address,uint256) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts/token/ERC20/extensions/ERC20Votes.sol#172-176) is never used and should be removed
ERC20Votes._subtract(uint256,uint256) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts/token/ERC20/extensions/ERC20Votes.sol#246-248) is never used and should be removed
ERC721BurnableUpgradeable.__ERC721Burnable_init_unchained() (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/token/ERC721/extensions/ERC721BurnableUpgradeable.sol#18-19) is never used and should be removed
ERC721EnumerableUpgradeable.__ERC721Enumerable_init_unchained() (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/token/ERC721/extensions/ERC721EnumerableUpgradeable.sol#19-20) is never used and should be removed
Initializable._disableInitializers() (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/proxy/utils/Initializable.sol#131-137) is never used and should be removed
Math.ceilDiv(uint256,uint256) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts/utils/math/Math.sol#45-48) is never used and should be removed
Math.max(uint256,uint256) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts/utils/math/Math.sol#19-21) is never used and should be removed
Math.min(uint256,uint256) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts/utils/math/Math.sol#26-28) is never used and should be removed
Math.mulDiv(uint256,uint256,uint256) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts/utils/math/Math.sol#55-135) is never used and should be removed
Math.mulDiv(uint256,uint256,uint256,Math.Rounding) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts/utils/math/Math.sol#140-151) is never used and should be removed
SafeCast.toUint160(uint256) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts/utils/math/SafeCast.sol#221-224) is never used and should be removed
SafeCast.toUint168(uint256) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts/utils/math/SafeCast.sol#204-207) is never used and should be removed
SafeCast.toUint176(uint256) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts/utils/math/SafeCast.sol#187-190) is never used and should be removed
SafeCast.toUint184(uint256) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts/utils/math/SafeCast.sol#170-173) is never used and should be removed
SafeCast.toUint192(uint256) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts/utils/math/SafeCast.sol#153-156) is never used and should be removed
SafeCast.toUint200(uint256) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts/utils/math/SafeCast.sol#136-139) is never used and should be removed
SafeCast.toUint208(uint256) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts/utils/math/SafeCast.sol#119-122) is never used and should be removed
SafeCast.toUint216(uint256) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts/utils/math/SafeCast.sol#102-105) is never used and should be removed
SafeCast.toUint232(uint256) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts/utils/math/SafeCast.sol#68-71) is never used and should be removed
SafeCast.toUint24(uint256) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts/utils/math/SafeCast.sol#510-513) is never used and should be removed
SafeCast.toUint240(uint256) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts/utils/math/SafeCast.sol#51-54) is never used and should be removed
SafeCast.toUint248(uint256) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts/utils/math/SafeCast.sol#34-37) is never used and should be removed
SafeCast.toUint256(int256) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts/utils/math/SafeCast.sol#558-561) is never used and should be removed
SafeCast.toUint40(uint256) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts/utils/math/SafeCast.sol#476-479) is never used and should be removed
SafeCast.toUint48(uint256) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts/utils/math/SafeCast.sol#459-462) is never used and should be removed
SafeCast.toUint56(uint256) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts/utils/math/SafeCast.sol#442-445) is never used and should be removed
SafeCast.toUint64(uint256) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts/utils/math/SafeCast.sol#425-428) is never used and should be removed
SafeCast.toUint72(uint256) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts/utils/math/SafeCast.sol#408-411) is never used and should be removed
SafeCast.toUint8(uint256) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts/utils/math/SafeCast.sol#544-547) is never used and should be removed
SafeCast.toUint80(uint256) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts/utils/math/SafeCast.sol#391-394) is never used and should be removed
SafeCast.toUint88(uint256) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts/utils/math/SafeCast.sol#374-377) is never used and should be removed
SafeCast.toUint96(uint256) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts/utils/math/SafeCast.sol#357-360) is never used and should be removed
SafeERC20Upgradeable.safeApprove(IERC20Upgradeable,address,uint256) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/token/ERC20/utils/SafeERC20Upgradeable.sol#46-59) is never used and should be removed
SafeERC20Upgradeable.safeDecreaseAllowance(IERC20Upgradeable,address,uint256) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/token/ERC20/utils/SafeERC20Upgradeable.sol#70-81) is never used and should be removed
SafeERC20Upgradeable.safeIncreaseAllowance(IERC20Upgradeable,address,uint256) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/token/ERC20/utils/SafeERC20Upgradeable.sol#61-68) is never used and should be removed
SafeERC20Upgradeable.safePermit(IERC20PermitUpgradeable,address,address,uint256,uint256,uint8,bytes32,bytes32) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/token/ERC20/utils/SafeERC20Upgradeable.sol#83-97) is never used and should be removed
SafeERC20Upgradeable.safeTransfer(IERC20Upgradeable,address,uint256) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/token/ERC20/utils/SafeERC20Upgradeable.sol#22-28) is never used and should be removed
StorageSlotUpgradeable.getBytes32Slot(bytes32) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/StorageSlotUpgradeable.sol#72-77) is never used and should be removed
StorageSlotUpgradeable.getUint256Slot(bytes32) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/StorageSlotUpgradeable.sol#82-87) is never used and should be removed
Strings.toHexString(address) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts/utils/Strings.sol#72-74) is never used and should be removed
Strings.toHexString(uint256) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts/utils/Strings.sol#41-52) is never used and should be removed
Strings.toString(uint256) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts/utils/Strings.sol#16-36) is never used and should be removed
StringsUpgradeable.toHexString(address) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/StringsUpgradeable.sol#72-74) is never used and should be removed
StringsUpgradeable.toHexString(uint256) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/StringsUpgradeable.sol#41-52) is never used and should be removed
UUPSUpgradeable.__UUPSUpgradeable_init_unchained() (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/proxy/utils/UUPSUpgradeable.sol#26-27) is never used and should be removed
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#dead-code

Pragma version^0.8.0 (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/access/AccessControlUpgradeable.sol#4) allows old versions
Pragma version^0.8.0 (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/access/IAccessControlUpgradeable.sol#4) allows old versions
Pragma version^0.8.0 (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/interfaces/draft-IERC1822Upgradeable.sol#4) allows old versions
Pragma version^0.8.2 (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/proxy/ERC1967/ERC1967UpgradeUpgradeable.sol#4) allows old versions
Pragma version^0.8.0 (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/proxy/beacon/IBeaconUpgradeable.sol#4) allows old versions
Pragma version^0.8.2 (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/proxy/utils/Initializable.sol#4) allows old versions
Pragma version^0.8.0 (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/proxy/utils/UUPSUpgradeable.sol#4) allows old versions
Pragma version^0.8.0 (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/security/PausableUpgradeable.sol#4) allows old versions
Pragma version^0.8.0 (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/security/ReentrancyGuardUpgradeable.sol#4) allows old versions
Pragma version^0.8.0 (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/token/ERC20/IERC20Upgradeable.sol#4) allows old versions
Pragma version^0.8.0 (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/token/ERC20/extensions/draft-IERC20PermitUpgradeable.sol#4) allows old versions
Pragma version^0.8.0 (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/token/ERC20/utils/SafeERC20Upgradeable.sol#4) allows old versions
Pragma version^0.8.0 (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/token/ERC721/IERC721Upgradeable.sol#4) allows old versions
Pragma version^0.8.0 (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/token/ERC721/IERC721ReceiverUpgradeable.sol#4) allows old versions
Pragma version^0.8.0 (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/token/ERC721/IERC721Upgradeable.sol#4) allows old versions
```

AUTOMATED TESTING

```
Function AccessControlUpgradeable.__AccessControl_init() (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/access/AccessControlUpgradeable.sol#51-52) is not in mixedCase
Function AccessControlUpgradeable.__AccessControl_init_unchained() (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/access/AccessControlUpgradeable.sol#54-55) is not in mixedCase
Variable AccessControlUpgradeable.__gap (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/access/AccessControlUpgradeable.sol#259) is not in mixedCase
Function ERC1967UpgradeUpgradeable.__ERC1967Upgrade_init() (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/proxy/ERC1967/ERC1967UpgradeUpgradeable.sol#21-22) is not in mixedCase
Function ERC1967UpgradeUpgradeable.__ERC1967Upgrade_init_unchained() (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/proxy/ERC1967/ERC1967UpgradeUpgradeable.sol#24-25) is not in mixedCase
Variable ERC1967UpgradeUpgradeable.__gap (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/proxy/ERC1967/ERC1967UpgradeUpgradeable.sol#211) is not in mixedCase
Function UUPSUpgradeable.__UUPSUpgradeable_init() (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/proxy/utils/UUPSUpgradeable.sol#23-24) is not in mixedCase
Function UUPSUpgradeable.__UUPSUpgradeable_init_unchained() (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/proxy/utils/UUPSUpgradeable.sol#26-27) is not in mixedCase
Variable UUPSUpgradeable.__gap (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/proxy/utils/UUPSUpgradeable.sol#107) is not in mixedCase
Variable UUPSUpgradeable.__self (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/proxy/utils/UUPSUpgradeable.sol#29) is not in mixedCase
Function PausableUpgradeable.__Pausable_init() (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/security/PausableUpgradeable.sol#34-36) is not in mixedCase
Function PausableUpgradeable.__Pausable_init_unchained() (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/security/PausableUpgradeable.sol#38-40) is not in mixedCase
Variable PausableUpgradeable.__gap (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/security/PausableUpgradeable.sol#116) is not in mixedCase
Function ReentrancyGuardUpgradeable.__ReentrancyGuard_init() (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/security/ReentrancyGuardUpgradeable.sol#40-42) is not in mixedCase
Function ReentrancyGuardUpgradeable.__ReentrancyGuard_init_unchained() (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/security/ReentrancyGuardUpgradeable.sol#44-46) is not in mixedCase
Variable ReentrancyGuardUpgradeable.__gap (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/security/ReentrancyGuardUpgradeable.sol#74) is not in mixedCase
Function IERC20PermitUpgradeable.DOMAIN_SEPARATOR() (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/token/ERC20/extensions/draft-IERC20PermitUpgradeable.sol#59) is not in mixedCase
Function ERC721Upgradeable.__ERC721_init(string,string) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/token/ERC721/ERC721Upgradeable.sol#45-47) is not in mixedCase
Function ERC721Upgradeable.__ERC721_init_unchained(string,string) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/token/ERC721/ERC721Upgradeable.sol#49-52) is not in mixedCase
Variable ERC721Upgradeable.__gap (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/token/ERC721/ERC721Upgradeable.sol#465) is not in mixedCase
Function ERC721BurnableUpgradeable.__ERC721Burnable_init() (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/token/ERC721/extensions/ERC721BurnableUpgradeable.sol#15-16) is not in mixedCase
Function ERC721BurnableUpgradeable.__ERC721Burnable_init_unchained() (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/token/ERC721/extensions/ERC721BurnableUpgradeable.sol#18-19) is not in mixedCa
se
Variable ERC721BurnableUpgradeable.__gap (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/token/ERC721/extensions/ERC721BurnableUpgradeable.sol#38) is not in mixedCase
Function ERC721EnumerableUpgradeable.__ERC721Enumerable_init() (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/token/ERC721/extensions/ERC721EnumerableUpgradeable.sol#16-17) is not in mixedCase
Function ERC721EnumerableUpgradeable.__ERC721Enumerable_init_unchained() (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/token/ERC721/extensions/ERC721EnumerableUpgradeable.sol#19-20) is not in m
ixedCase
Variable ERC721EnumerableUpgradeable.__gap (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/token/ERC721/extensions/ERC721EnumerableUpgradeable.sol#175) is not in mixedCase
Function ContextUpgradeable.__Context_init() (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/ContextUpgradeable.sol#18-19) is not in mixedCase
Function ContextUpgradeable.__Context_init_unchained() (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/ContextUpgradeable.sol#21-22) is not in mixedCase
Variable ContextUpgradeable.__gap (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/ContextUpgradeable.sol#36) is not in mixedCase
Function ERC165Upgradeable.__ERC165_init() (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/introspection/ERC165Upgradeable.sol#24-25) is not in mixedCase
Function ERC165Upgradeable.__ERC165_init_unchained() (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/introspection/ERC165Upgradeable.sol#27-28) is not in mixedCase
Variable ERC165Upgradeable.__gap (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/introspection/ERC165Upgradeable.sol#41) is not in mixedCase
Function ERC20Permit.DOMAIN_SEPARATOR() (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts/token/ERC20/extensions/draft-ERC20Permit.sol#81-83) is not in mixedCase
Variable ERC20Permit._PERMIT_TYPEHASH_DEPRECATED_SLOT (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts/token/ERC20/extensions/draft-ERC20Permit.sol#37) is not in mixedCase
Function IERC20Permit.DOMAIN_SEPARATOR() (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts/token/ERC20/extensions/draft-IERC20Permit.sol#59) is not in mixedCase
Variable EIP712._CACHED_DOMAIN_SEPARATOR (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts/utils/cryptography/draft-EIP712.sol#31) is not in mixedCase
Variable EIP712._CACHED_CHAIN_ID (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts/utils/cryptography/draft-EIP712.sol#32) is not in mixedCase
Variable EIP712._CACHED_THIS (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts/utils/cryptography/draft-EIP712.sol#33) is not in mixedCase
Variable EIP712._HASHED_NAME (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts/utils/cryptography/draft-EIP712.sol#35) is not in mixedCase
Variable EIP712._HASHED_VERSION (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts/utils/cryptography/draft-EIP712.sol#36) is not in mixedCase
Variable EIP712._TYPE_HASH (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts/utils/cryptography/draft-EIP712.sol#37) is not in mixedCase
Variable LeverageEngine._addressCoordinator (jobs/archimedes/brownie/contracts/LeverageEngine.sol#20) is not in mixedCase
Variable LeverageEngine._addressPositionToken (jobs/archimedes/brownie/contracts/LeverageEngine.sol#21) is not in mixedCase
Variable LeverageEngine._addressParameterStore (jobs/archimedes/brownie/contracts/LeverageEngine.sol#22) is not in mixedCase
Variable LeverageEngine._addressArchToken (jobs/archimedes/brownie/contracts/LeverageEngine.sol#23) is not in mixedCase
Variable LeverageEngine._addressOUSD (jobs/archimedes/brownie/contracts/LeverageEngine.sol#24) is not in mixedCase
Variable LeverageEngine._coordinator (jobs/archimedes/brownie/contracts/LeverageEngine.sol#26) is not in mixedCase
Variable LeverageEngine._positionToken (jobs/archimedes/brownie/contracts/LeverageEngine.sol#27) is not in mixedCase
Variable LeverageEngine._parameterStore (jobs/archimedes/brownie/contracts/LeverageEngine.sol#28) is not in mixedCase
Variable LeverageEngine._archToken (jobs/archimedes/brownie/contracts/LeverageEngine.sol#29) is not in mixedCase
Variable LeverageEngine._ousd (jobs/archimedes/brownie/contracts/LeverageEngine.sol#30) is not in mixedCase
Variable ParameterStore._maxNumberOfCycles (jobs/archimedes/brownie/contracts/ParameterStore.sol#14) is not in mixedCase
Variable ParameterStore._originationFeeRate (jobs/archimedes/brownie/contracts/ParameterStore.sol#15) is not in mixedCase
Variable ParameterStore._globalCollateralRate (jobs/archimedes/brownie/contracts/ParameterStore.sol#16) is not in mixedCase
Variable ParameterStore._rebaseFeeRate (jobs/archimedes/brownie/contracts/ParameterStore.sol#17) is not in mixedCase
Variable ParameterStore._treasuryAddress (jobs/archimedes/brownie/contracts/ParameterStore.sol#18) is not in mixedCase
Variable ParameterStore._curveGuardPercentage (jobs/archimedes/brownie/contracts/ParameterStore.sol#19) is not in mixedCase

ArchToken.constructor(address) (jobs/archimedes/brownie/contracts/ArchToken.sol#15-18) uses literals with too many digits:
        - _mint(_addressTreasury,1000000000000000000000000) (jobs/archimedes/brownie/contracts/ArchToken.sol#16)
ParameterStore.changeArchToLevRatio(uint256) (jobs/archimedes/brownie/contracts/ParameterStore.sol#99-103) uses literals with too many digits:
        - require(bool,string)(newArchToLevRatio < 500000000000000000000000 && newArchToLevRatio >= 1000000000000000000,new ArchToLevRatio out of range) (jobs/archimedes/brownie/contracts/ParameterStore.sol#100)
ParameterStore.changeMinPositionCollateral(uint256) (jobs/archimedes/brownie/contracts/ParameterStore.sol#111-115) uses literals with too many digits:
        - require(bool,string)(newMinPositionCollateral < (1000000000000000000000000) && newMinPositionCollateral > (1000000000000000000),New min collateral out of range) (jobs/archimedes/brownie/contracts/ParameterStore.sol#112)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#too-many-digits

ERC20Permit._PERMIT_TYPEHASH_DEPRECATED_SLOT (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts/token/ERC20/extensions/draft-ERC20Permit.sol#37) is never used in ArchToken (jobs/archimedes/brownie/contracts/Arc
hToken.sol#14-36)
PausableUpgradeable.__gap (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/security/PausableUpgradeable.sol#116) is never used in LeverageEngine (jobs/archimedes/brownie/contracts/LeverageEngine.s
ol#17-156)
UUPSUpgradeable.__gap (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/proxy/utils/UUPSUpgradeable.sol#107) is never used in ParameterStore (jobs/archimedes/brownie/contracts/ParameterStore.sol#10
-243)
UUPSUpgradeable.__gap (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/proxy/utils/UUPSUpgradeable.sol#107) is never used in PositionToken (jobs/archimedes/brownie/contracts/PositionToken.sol#15-1
32)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#unused-state-variable

ERC20Permit._PERMIT_TYPEHASH_DEPRECATED_SLOT (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts/token/ERC20/extensions/draft-ERC20Permit.sol#37) should be constant
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#state-variables-that-could-be-declared-constant

grantRole(bytes32,address) should be declared external:
        - AccessControlUpgradeable.grantRole(bytes32,address) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/access/AccessControlUpgradeable.sol#150-152)
revokeRole(bytes32,address) should be declared external:
        - AccessControlUpgradeable.revokeRole(bytes32,address) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/access/AccessControlUpgradeable.sol#165-167)
renounceRole(bytes32,address) should be declared external:
        - AccessControlUpgradeable.renounceRole(bytes32,address) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/access/AccessControlUpgradeable.sol#185-189)
        - AccessController.renounceRole(bytes32,address) (jobs/archimedes/brownie/contracts/AccessController.sol#62-69)
name() should be declared external:
        - ERC721Upgradeable.name() (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/token/ERC721/ERC721Upgradeable.sol#84-86)
symbol() should be declared external:
        - ERC721Upgradeable.symbol() (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/token/ERC721/ERC721Upgradeable.sol#91-93)
tokenURI(uint256) should be declared external:
        - ERC721Upgradeable.tokenURI(uint256) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/token/ERC721/ERC721Upgradeable.sol#98-103)
approve(address,uint256) should be declared external:
        - ERC721Upgradeable.approve(address,uint256) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/token/ERC721/ERC721Upgradeable.sol#117-127)
setApprovalForAll(address,bool) should be declared external:
        - ERC721Upgradeable.setApprovalForAll(address,bool) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/token/ERC721/ERC721Upgradeable.sol#141-143)
transferFrom(address,address,uint256) should be declared external:
        - ERC721Upgradeable.transferFrom(address,address,uint256) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/token/ERC721/ERC721Upgradeable.sol#155-164)
safeTransferFrom(address,address,uint256) should be declared external:
        - ERC721Upgradeable.safeTransferFrom(address,address,uint256) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/token/ERC721/ERC721Upgradeable.sol#169-175)
tokenOfOwnerByIndex(address,uint256) should be declared external:
        - ERC721EnumerableUpgradeable.tokenOfOwnerByIndex(address,uint256) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/token/ERC721/extensions/ERC721EnumerableUpgradeable.sol#43-46)
tokenByIndex(uint256) should be declared external:
        - ERC721EnumerableUpgradeable.tokenByIndex(uint256) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/token/ERC721/extensions/ERC721EnumerableUpgradeable.sol#58-61)
setAdmin(address) should be declared external:
        - AccessController.setAdmin(address) (jobs/archimedes/brownie/contracts/AccessController.sol#41-47)
initialize() should be declared external:
        - LeverageEngine.initialize() (jobs/archimedes/brownie/contracts/LeverageEngine.sol#123-132)
initialize() should be declared external:
        - ParameterStore.initialize() (jobs/archimedes/brownie/contracts/ParameterStore.sol#31-52)
initialize() should be declared external:
```

- LvUSDToken.sol

AUTOMATED TESTING

```
LvUSDToken.setMintDestination(address).mintDestination (jobs/archimedes/brownie/contracts/LvUSDToken.sol#29) lacks a zero-check on :
        - _mintingDestination = mintDestination (jobs/archimedes/brownie/contracts/LvUSDToken.sol#30)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#missing-zero-address-validation

Different versions of Solidity are used:
        - Version used: ['0.8.13', '^0.8.0']
        - ^0.8.0 (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts/access/AccessControl.sol#4)
        - ^0.8.0 (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts/access/IAccessControl.sol#4)
        - ^0.8.0 (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts/token/ERC20/ERC20.sol#4)
        - ^0.8.0 (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts/token/ERC20/IERC20.sol#4)
        - ^0.8.0 (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts/token/ERC20/extensions/IERC20Metadata.sol#4)
        - ^0.8.0 (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts/utils/Context.sol#4)
        - ^0.8.0 (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts/utils/Strings.sol#4)
        - ^0.8.0 (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts/utils/introspection/ERC165.sol#4)
        - ^0.8.0 (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts/utils/introspection/IERC165.sol#4)
        - 0.8.13 (jobs/archimedes/brownie/contracts/BasicAccessController.sol#2)
        - 0.8.13 (jobs/archimedes/brownie/contracts/LvUSDToken.sol#2)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#different-pragma-directives-are-used

AccessControl._setRoleAdmin(bytes32,bytes32) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts/access/AccessControl.sol#214-218) is never used and should be removed
AccessControl._setupRole(bytes32,address) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts/access/AccessControl.sol#205-207) is never used and should be removed
BasicAccessController._requireAdmin() (jobs/archimedes/brownie/contracts/BasicAccessController.sol#67-69) is never used and should be removed
Context._msgData() (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts/utils/Context.sol#21-23) is never used and should be removed
ERC20._burn(address,uint256) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts/token/ERC20/ERC20.sol#280-295) is never used and should be removed
Strings.toHexString(address) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts/utils/Strings.sol#72-74) is never used and should be removed
Strings.toHexString(uint256) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts/utils/Strings.sol#41-52) is never used and should be removed
Strings.toString(uint256) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts/utils/Strings.sol#16-36) is never used and should be removed
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#dead-code

Pragma version^0.8.0 (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts/access/AccessControl.sol#4) allows old versions
Pragma version^0.8.0 (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts/access/IAccessControl.sol#4) allows old versions
Pragma version^0.8.0 (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts/token/ERC20/ERC20.sol#4) allows old versions
Pragma version^0.8.0 (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts/token/ERC20/IERC20.sol#4) allows old versions
Pragma version^0.8.0 (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts/token/ERC20/extensions/IERC20Metadata.sol#4) allows old versions
Pragma version^0.8.0 (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts/utils/Context.sol#4) allows old versions
Pragma version^0.8.0 (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts/utils/Strings.sol#4) allows old versions
Pragma version^0.8.0 (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts/utils/introspection/ERC165.sol#4) allows old versions
Pragma version^0.8.0 (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts/utils/introspection/IERC165.sol#4) allows old versions
Pragma version0.8.13 (jobs/archimedes/brownie/contracts/BasicAccessController.sol#2) necessitates a version too recent to be trusted. Consider deploying with 0.6.12/0.7.6/0.8.7
Pragma version0.8.13 (jobs/archimedes/brownie/contracts/LvUSDToken.sol#2) necessitates a version too recent to be trusted. Consider deploying with 0.6.12/0.7.6/0.8.7
solc-0.8.13 is not recommended for deployment
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#incorrect-versions-of-solidity

Variable LvUSDToken._mintingDestination (jobs/archimedes/brownie/contracts/LvUSDToken.sol#10) is not in mixedCase
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#conformance-to-solidity-naming-conventions

Variable LvUSDToken._mintingDestination (jobs/archimedes/brownie/contracts/LvUSDToken.sol#10) is not in mixedCase
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#conformance-to-solidity-naming-conventions

grantRole(bytes32,address) should be declared external:
        - AccessControl.grantRole(bytes32,address) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts/access/AccessControl.sol#144-146)
revokeRole(bytes32,address) should be declared external:
        - AccessControl.revokeRole(bytes32,address) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts/access/AccessControl.sol#159-161)
renounceRole(bytes32,address) should be declared external:
        - AccessControl.renounceRole(bytes32,address) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts/access/AccessControl.sol#179-183)
        - BasicAccessController.renounceRole(bytes32,address) (jobs/archimedes/brownie/contracts/BasicAccessController.sol#46-53)
name() should be declared external:
        - ERC20.name() (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts/token/ERC20/ERC20.sol#62-64)
symbol() should be declared external:
        - ERC20.symbol() (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts/token/ERC20/ERC20.sol#70-72)
decimals() should be declared external:
        - ERC20.decimals() (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts/token/ERC20/ERC20.sol#87-89)
totalSupply() should be declared external:
        - ERC20.totalSupply() (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts/token/ERC20/ERC20.sol#94-96)
balanceOf(address) should be declared external:
        - ERC20.balanceOf(address) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts/token/ERC20/ERC20.sol#101-103)
transfer(address,uint256) should be declared external:
        - ERC20.transfer(address,uint256) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts/token/ERC20/ERC20.sol#113-117)
approve(address,uint256) should be declared external:
        - ERC20.approve(address,uint256) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts/token/ERC20/ERC20.sol#136-140)
transferFrom(address,address,uint256) should be declared external:
        - ERC20.transferFrom(address,address,uint256) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts/token/ERC20/ERC20.sol#158-167)
increaseAllowance(address,uint256) should be declared external:
        - ERC20.increaseAllowance(address,uint256) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts/token/ERC20/ERC20.sol#181-185)
decreaseAllowance(address,uint256) should be declared external:
        - ERC20.decreaseAllowance(address,uint256) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts/token/ERC20/ERC20.sol#201-210)
setAdmin(address) should be declared external:
        - BasicAccessController.setAdmin(address) (jobs/archimedes/brownie/contracts/BasicAccessController.sol#25-31)
getAddressMinter() should be declared external:
        - BasicAccessController.getAddressMinter() (jobs/archimedes/brownie/contracts/BasicAccessController.sol#63-65)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#public-function-that-could-be-declared-external
```

- ParameterStore.sol

```
ERC1967UpgradeUpgradeable._functionDelegateCall(address,bytes) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/proxy/ERC1967/ERC1967UpgradeUpgradeable.sol#198-204) uses delegatecall to a input-co
ntrolled function id
        - (success,returndata) = target.delegatecall(data) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/proxy/ERC1967/ERC1967UpgradeUpgradeable.sol#202)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#controlled-delegatecall

ParameterStore (jobs/archimedes/brownie/contracts/ParameterStore.sol#10-243) is an upgradeable contract that does not protect its initiliaze functions: ParameterStore.initialize() (jobs/archimedes/brownie/contracts/ParameterStore.
sol#31-52). Anyone can delete the contract with: UUPSUpgradeable.upgradeTo(address) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/proxy/utils/UUPSUpgradeable.sol#72-75)UUPSUpgradeable.upgradeTo
AndCall(address,bytes) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/proxy/utils/UUPSUpgradeable.sol#85-88)Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#unprotected-u
pgradeable-contract

ParameterStore.calculateLeverageAllowedForArch(uint256) (jobs/archimedes/brownie/contracts/ParameterStore.sol#210-220) performs a multiplication on the result of a division:
        -(_archToLevRatio / 1000000000000000000) * archAmount (jobs/archimedes/brownie/contracts/ParameterStore.sol#219)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#divide-before-multiply

ERC1967UpgradeUpgradeable._upgradeToAndCallUUPS(address,bytes,bool).slot (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/proxy/ERC1967/ERC1967UpgradeUpgradeable.sol#98) is a local variable never
initialized
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#uninitialized-local-variables

ERC1967UpgradeUpgradeable._upgradeToAndCallUUPS(address,bytes,bool) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/proxy/ERC1967/ERC1967UpgradeUpgradeable.sol#87-105) ignores return value by IER
C1822ProxiableUpgradeable(newImplementation).proxiableUUID() (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/proxy/ERC1967/ERC1967UpgradeUpgradeable.sol#98-102)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#unused-return

Variable 'ERC1967UpgradeUpgradeable._upgradeToAndCallUUPS(address,bytes,bool).slot (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/proxy/ERC1967/ERC1967UpgradeUpgradeable.sol#98)' in ERC1967Upgra
deUpgradeable._upgradeToAndCallUUPS(address,bytes,bool) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/proxy/ERC1967/ERC1967UpgradeUpgradeable.sol#87-105) potentially used before declaration: re
quire(bool,string)(slot == _IMPLEMENTATION_SLOT,ERC1967Upgrade: unsupported proxiableUUID) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/proxy/ERC1967/ERC1967UpgradeUpgradeable.sol#99)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#pre-declaration-usage-of-local-variables

AddressUpgradeable.verifyCallResult(bool,bytes,string) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/AddressUpgradeable.sol#174-194) uses assembly
        - INLINE ASM (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/AddressUpgradeable.sol#186-189)
StorageSlotUpgradeable.getAddressSlot(bytes32) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/StorageSlotUpgradeable.sol#52-57) uses assembly
        - INLINE ASM (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/StorageSlotUpgradeable.sol#54-56)
StorageSlotUpgradeable.getBooleanSlot(bytes32) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/StorageSlotUpgradeable.sol#62-67) uses assembly
        - INLINE ASM (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/StorageSlotUpgradeable.sol#64-66)
StorageSlotUpgradeable.getBytes32Slot(bytes32) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/StorageSlotUpgradeable.sol#72-77) uses assembly
        - INLINE ASM (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/StorageSlotUpgradeable.sol#74-76)
StorageSlotUpgradeable.getUint256Slot(bytes32) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/StorageSlotUpgradeable.sol#82-87) uses assembly
        - INLINE ASM (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/StorageSlotUpgradeable.sol#84-86)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#assembly-usage
```

```
Different versions of Solidity are used:
        - Version used: ['0.8.13', '^0.8.0', '^0.8.1', '^0.8.2']
        - ^0.8.0 (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/access/AccessControlUpgradeable.sol#4)
        - ^0.8.0 (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/access/IAccessControlUpgradeable.sol#4)
        - ^0.8.0 (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/interfaces/draft-IERC1822Upgradeable.sol#4)
        - ^0.8.2 (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/proxy/ERC1967/ERC1967UpgradeUpgradeable.sol#4)
        - ^0.8.0 (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/proxy/beacon/IBeaconUpgradeable.sol#4)
        - ^0.8.2 (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/proxy/utils/Initializable.sol#4)
        - ^0.8.0 (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/proxy/utils/UUPSUpgradeable.sol#4)
        - ^0.8.1 (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/AddressUpgradeable.sol#4)
        - ^0.8.0 (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/ContextUpgradeable.sol#4)
        - ^0.8.0 (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/StorageSlotUpgradeable.sol#4)
        - ^0.8.0 (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/StringsUpgradeable.sol#4)
        - ^0.8.0 (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/introspection/ERC165Upgradeable.sol#4)
        - ^0.8.0 (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/introspection/IERC165Upgradeable.sol#4)
        - 0.8.13 (jobs/archimedes/brownie/contracts/AccessController.sol#2)
        - 0.8.13 (jobs/archimedes/brownie/contracts/ParameterStore.sol#2)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#different-pragma-directives-are-used

AccessControlUpgradeable.__AccessControl_init_unchained() (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/access/AccessControlUpgradeable.sol#54-55) is never used and should be removed
AccessControlUpgradeable._setRoleAdmin(bytes32,bytes32) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/access/AccessControlUpgradeable.sol#220-224) is never used and should be removed
AccessControlUpgradeable._setupRole(bytes32,address) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/access/AccessControlUpgradeable.sol#211-213) is never used and should be removed
AddressUpgradeable.functionCall(address,bytes) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/AddressUpgradeable.sol#85-87) is never used and should be removed
AddressUpgradeable.functionCall(address,bytes,string) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/AddressUpgradeable.sol#95-101) is never used and should be removed
AddressUpgradeable.functionCallWithValue(address,bytes,uint256) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/AddressUpgradeable.sol#114-120) is never used and should be removed
AddressUpgradeable.functionCallWithValue(address,bytes,uint256,string) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/AddressUpgradeable.sol#128-139) is never used and should be removed
AddressUpgradeable.functionStaticCall(address,bytes) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/AddressUpgradeable.sol#147-149) is never used and should be removed
AddressUpgradeable.functionStaticCall(address,bytes,string) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/AddressUpgradeable.sol#157-166) is never used and should be removed
AddressUpgradeable.sendValue(address,uint256) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/AddressUpgradeable.sol#60-65) is never used and should be removed
ContextUpgradeable.__Context_init() (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/ContextUpgradeable.sol#18-19) is never used and should be removed
ContextUpgradeable.__Context_init_unchained() (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/ContextUpgradeable.sol#21-22) is never used and should be removed
ContextUpgradeable._msgData() (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/ContextUpgradeable.sol#27-29) is never used and should be removed
ERC165Upgradeable.__ERC165_init() (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/introspection/ERC165Upgradeable.sol#24-25) is never used and should be removed
ERC165Upgradeable.__ERC165_init_unchained() (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/introspection/ERC165Upgradeable.sol#27-28) is never used and should be removed
ERC1967UpgradeUpgradeable.__ERC1967Upgrade_init() (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/proxy/ERC1967/ERC1967UpgradeUpgradeable.sol#21-22) is never used and should be removed
ERC1967UpgradeUpgradeable.__ERC1967Upgrade_init_unchained() (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/proxy/ERC1967/ERC1967UpgradeUpgradeable.sol#24-25) is never used and should be removed
ERC1967UpgradeUpgradeable._changeAdmin(address) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/proxy/ERC1967/ERC1967UpgradeUpgradeable.sol#139-142) is never used and should be removed
ERC1967UpgradeUpgradeable._getAdmin() (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/proxy/ERC1967/ERC1967UpgradeUpgradeable.sol#122-124) is never used and should be removed
ERC1967UpgradeUpgradeable._getBeacon() (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/proxy/ERC1967/ERC1967UpgradeUpgradeable.sol#158-160) is never used and should be removed
ERC1967UpgradeUpgradeable._setAdmin(address) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/proxy/ERC1967/ERC1967UpgradeUpgradeable.sol#129-132) is never used and should be removed
ERC1967UpgradeUpgradeable._setBeacon(address) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/proxy/ERC1967/ERC1967UpgradeUpgradeable.sol#165-172) is never used and should be removed
ERC1967UpgradeUpgradeable._upgradeBeaconToAndCall(address,bytes,bool) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/proxy/ERC1967/ERC1967UpgradeUpgradeable.sol#180-190) is never used and should
be removed
Initializable._disableInitializers() (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/proxy/utils/Initializable.sol#131-137) is never used and should be removed
StorageSlotUpgradeable.getBytes32Slot(bytes32) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/StorageSlotUpgradeable.sol#72-77) is never used and should be removed
StorageSlotUpgradeable.getUint256Slot(bytes32) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/StorageSlotUpgradeable.sol#82-87) is never used and should be removed
StringsUpgradeable.toHexString(address) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/StringsUpgradeable.sol#72-74) is never used and should be removed
StringsUpgradeable.toHexString(uint256) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/StringsUpgradeable.sol#41-52) is never used and should be removed
StringsUpgradeable.toString(uint256) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/StringsUpgradeable.sol#16-36) is never used and should be removed
UUPSUpgradeable.__UUPSUpgradeable_init_unchained() (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/proxy/utils/UUPSUpgradeable.sol#26-27) is never used and should be removed
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#dead-code

Pragma version^0.8.0 (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/access/AccessControlUpgradeable.sol#4) allows old versions
Pragma version^0.8.0 (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/access/IAccessControlUpgradeable.sol#4) allows old versions
Pragma version^0.8.0 (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/interfaces/draft-IERC1822Upgradeable.sol#4) allows old versions
Pragma version^0.8.2 (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/proxy/ERC1967/ERC1967UpgradeUpgradeable.sol#4) allows old versions
Pragma version^0.8.0 (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/proxy/beacon/IBeaconUpgradeable.sol#4) allows old versions
Pragma version^0.8.2 (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/proxy/utils/Initializable.sol#4) allows old versions
Pragma version^0.8.0 (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/proxy/utils/UUPSUpgradeable.sol#4) allows old versions
Pragma version^0.8.1 (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/AddressUpgradeable.sol#4) allows old versions
Pragma version^0.8.0 (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/ContextUpgradeable.sol#4) allows old versions
Pragma version^0.8.0 (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/StorageSlotUpgradeable.sol#4) allows old versions
Pragma version^0.8.0 (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/StringsUpgradeable.sol#4) allows old versions
Pragma version^0.8.0 (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/introspection/ERC165Upgradeable.sol#4) allows old versions
Pragma version^0.8.0 (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/introspection/IERC165Upgradeable.sol#4) allows old versions
Pragma version0.8.13 (jobs/archimedes/brownie/contracts/AccessController.sol#2) necessitates a version too recent to be trusted. Consider deploying with 0.6.12/0.7.6/0.8.7
Pragma version0.8.13 (jobs/archimedes/brownie/contracts/ParameterStore.sol#2) necessitates a version too recent to be trusted. Consider deploying with 0.6.12/0.7.6/0.8.7
solc-0.8.13 is not recommended for deployment
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#incorrect-versions-of-solidity

Low level call in ERC1967UpgradeUpgradeable._functionDelegateCall(address,bytes) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/proxy/ERC1967/ERC1967UpgradeUpgradeable.sol#198-204):
        - (success,returndata) = target.delegatecall(data) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/proxy/ERC1967/ERC1967UpgradeUpgradeable.sol#202)
Low level call in AddressUpgradeable.sendValue(address,uint256) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/AddressUpgradeable.sol#60-65):
        - (success) = recipient.call{value: amount}() (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/AddressUpgradeable.sol#63)
Low level call in AddressUpgradeable.functionCallWithValue(address,bytes,uint256,string) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/AddressUpgradeable.sol#128-139):
        - (success,returndata) = target.call{value: value}(data) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/AddressUpgradeable.sol#137)
Low level call in AddressUpgradeable.functionStaticCall(address,bytes,string) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/AddressUpgradeable.sol#157-166):
        - (success,returndata) = target.staticcall(data) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/AddressUpgradeable.sol#164)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#low-level-calls

Function AccessControlUpgradeable.__AccessControl_init() (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/access/AccessControlUpgradeable.sol#51-52) is not in mixedCase
Function AccessControlUpgradeable.__AccessControl_init_unchained() (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/access/AccessControlUpgradeable.sol#54-55) is not in mixedCase
Variable AccessControlUpgradeable.__gap (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/access/AccessControlUpgradeable.sol#259) is not in mixedCase
Function ERC1967UpgradeUpgradeable.__ERC1967Upgrade_init() (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/proxy/ERC1967/ERC1967UpgradeUpgradeable.sol#21-22) is not in mixedCase
Function ERC1967UpgradeUpgradeable.__ERC1967Upgrade_init_unchained() (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/proxy/ERC1967/ERC1967UpgradeUpgradeable.sol#24-25) is not in mixedCase
Variable ERC1967UpgradeUpgradeable.__gap (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/proxy/ERC1967/ERC1967UpgradeUpgradeable.sol#211) is not in mixedCase
Function UUPSUpgradeable.__UUPSUpgradeable_init() (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/proxy/utils/UUPSUpgradeable.sol#23-24) is not in mixedCase
Function UUPSUpgradeable.__UUPSUpgradeable_init_unchained() (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/proxy/utils/UUPSUpgradeable.sol#26-27) is not in mixedCase
Variable UUPSUpgradeable.__gap (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/proxy/utils/UUPSUpgradeable.sol#107) is not in mixedCase
Variable UUPSUpgradeable.__self (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/proxy/utils/UUPSUpgradeable.sol#29) is not in mixedCase
Function ContextUpgradeable.__Context_init() (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/ContextUpgradeable.sol#18-19) is not in mixedCase
Function ContextUpgradeable.__Context_init_unchained() (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/ContextUpgradeable.sol#21-22) is not in mixedCase
Variable ContextUpgradeable.__gap (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/ContextUpgradeable.sol#36) is not in mixedCase
Function ERC165Upgradeable.__ERC165_init() (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/introspection/ERC165Upgradeable.sol#24-25) is not in mixedCase
Function ERC165Upgradeable.__ERC165_init_unchained() (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/introspection/ERC165Upgradeable.sol#27-28) is not in mixedCase
Variable ERC165Upgradeable.__gap (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/introspection/ERC165Upgradeable.sol#41) is not in mixedCase
Variable ParameterStore._maxNumberOfCycles (jobs/archimedes/brownie/contracts/ParameterStore.sol#14) is not in mixedCase
Variable ParameterStore._originationFeeRate (jobs/archimedes/brownie/contracts/ParameterStore.sol#15) is not in mixedCase
Variable ParameterStore._globalCollateralRate (jobs/archimedes/brownie/contracts/ParameterStore.sol#16) is not in mixedCase
Variable ParameterStore._rebaseFeeRate (jobs/archimedes/brownie/contracts/ParameterStore.sol#17) is not in mixedCase
Variable ParameterStore._treasuryAddress (jobs/archimedes/brownie/contracts/ParameterStore.sol#18) is not in mixedCase
Variable ParameterStore._curveGuardPercentage (jobs/archimedes/brownie/contracts/ParameterStore.sol#19) is not in mixedCase
Variable ParameterStore._slippage (jobs/archimedes/brownie/contracts/ParameterStore.sol#20) is not in mixedCase
Variable ParameterStore._archToLevRatio (jobs/archimedes/brownie/contracts/ParameterStore.sol#22) is not in mixedCase
Variable ParameterStore._curveMaxExchangeGuard (jobs/archimedes/brownie/contracts/ParameterStore.sol#24) is not in mixedCase
Variable ParameterStore._minPositionCollateral (jobs/archimedes/brownie/contracts/ParameterStore.sol#25) is not in mixedCase
Variable ParameterStore._positionTimeToLiveInDays (jobs/archimedes/brownie/contracts/ParameterStore.sol#26) is not in mixedCase
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#conformance-to-solidity-naming-conventions
ParameterStore.changeArchToLevRatio(uint256) (jobs/archimedes/brownie/contracts/ParameterStore.sol#99-103) uses literals with too many digits:
        - require(bool,string)(newArchToLevRatio < 500000000000000000000000 && newArchToLevRatio >= 100000000000000000,new ArchToLevRatio out of range) (jobs/archimedes/brownie/contracts/ParameterStore.sol#100)
ParameterStore.changeMinPositionCollateral(uint256) (jobs/archimedes/brownie/contracts/ParameterStore.sol#111-115) uses literals with too many digits:
        - require(bool,string)(newMinPositionCollateral < (1000000000000000000000000) && newMinPositionCollateral > (100000000000000000),New min collateral out of range) (jobs/archimedes/brownie/contracts/ParameterStore.sol#112)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#too-many-digits

UUPSUpgradeable.__gap (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/proxy/utils/UUPSUpgradeable.sol#107) is never used in ParameterStore (jobs/archimedes/brownie/contracts/ParameterStore.sol#10
-243)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#unused-state-variable

grantRole(bytes32,address) should be declared external:
        - AccessControlUpgradeable.grantRole(bytes32,address) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/access/AccessControlUpgradeable.sol#150-152)
revokeRole(bytes32,address) should be declared external:
        - AccessControlUpgradeable.revokeRole(bytes32,address) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/access/AccessControlUpgradeable.sol#165-167)
renounceRole(bytes32,address) should be declared external:
        - AccessControlUpgradeable.renounceRole(bytes32,address) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/access/AccessControlUpgradeable.sol#185-189)
        - AccessController.renounceRole(bytes32,address) (jobs/archimedes/brownie/contracts/AccessController.sol#62-69)
setAdmin(address) should be declared external:
        - AccessController.setAdmin(address) (jobs/archimedes/brownie/contracts/AccessController.sol#41-47)
getAddressExecutive() should be declared external:
        - AccessController.getAddressExecutive() (jobs/archimedes/brownie/contracts/AccessController.sol#95-97)
initialize() should be declared external:
        - ParameterStore.initialize() (jobs/archimedes/brownie/contracts/ParameterStore.sol#31-52)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#public-function-that-could-be-declared-external
```

- PoolManager.sol

AUTOMATED TESTING

```
ERC1967UpgradeUpgradeable._functionDelegateCall(address,bytes) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/proxy/ERC1967/ERC1967UpgradeUpgradeable.sol#198-204) uses delegatecall to a input-co
ntrolled function id
        - (success,returndata) = target.delegatecall(data) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/proxy/ERC1967/ERC1967UpgradeUpgradeable.sol#202)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#controlled-delegatecall

ParameterStore (jobs/archimedes/brownie/contracts/ParameterStore.sol#10-243) is an upgradeable contract that does not protect its initilaze functions: ParameterStore.initialize() (jobs/archimedes/brownie/contracts/ParameterStore.
sol#31-52). Anyone can delete the contract with: UUPSUpgradeable.upgradeTo(address) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/proxy/utils/UUPSUpgradeable.sol#72-75)UUPSUpgradeable.upgradeTo
AndCall(address,bytes) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/proxy/utils/UUPSUpgradeable.sol#85-88)PoolManager (jobs/archimedes/brownie/contracts/PoolManager.sol#15-86) is an upgradeabl
e contract that does not protect its initilaze functions: PoolManager.initialize() (jobs/archimedes/brownie/contracts/PoolManager.sol#26-35). Anyone can delete the contract with: UUPSUpgradeable.upgradeTo(address) (.brownie/packa
ges/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/proxy/utils/UUPSUpgradeable.sol#72-75)UUPSUpgradeable.upgradeToAndCall(address,bytes) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgra
deable/proxy/utils/UUPSUpgradeable.sol#85-88)Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#unprotected-upgradeable-contract

ParameterStore.calculateLeverageAllowedForArch(uint256) (jobs/archimedes/brownie/contracts/ParameterStore.sol#218-220) performs a multiplication on the result of a division:
        - _archToLevRatio / 10000000000000000) * archAmount (jobs/archimedes/brownie/contracts/ParameterStore.sol#219)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#divide-before-multiply

ERC1967UpgradeUpgradeable._upgradeToAndCallUUPS(address,bytes,bool).slot (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/proxy/ERC1967/ERC1967UpgradeUpgradeable.sol#98) is a local variable never
initialized
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#uninitialized-local-variables

ERC1967UpgradeUpgradeable._upgradeToAndCallUUPS(address,bytes,bool) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/proxy/ERC1967/ERC1967UpgradeUpgradeable.sol#87-105) ignores return value by IER
C1822ProxiableUpgradeable(newImplementation).proxiableUUID() (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/proxy/ERC1967/ERC1967UpgradeUpgradeable.sol#98-102)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#unused-return

PoolManager.setDependencies(address,address,address,address,address).addressParameterStore (jobs/archimedes/brownie/contracts/PoolManager.sol#46) lacks a zero-check on :
        - _addressParameterStore = addressParameterStore (jobs/archimedes/brownie/contracts/PoolManager.sol#53)
PoolManager.setDependencies(address,address,address,address,address).addressCoordinator (jobs/archimedes/brownie/contracts/PoolManager.sol#47) lacks a zero-check on :
        - _addressCoordinator = addressCoordinator (jobs/archimedes/brownie/contracts/PoolManager.sol#54)
PoolManager.setDependencies(address,address,address,address,address).addressPoolLvUSD3CRV (jobs/archimedes/brownie/contracts/PoolManager.sol#50) lacks a zero-check on :
        - _addressPoolLvUSD3CRV = addressPoolLvUSD3CRV (jobs/archimedes/brownie/contracts/PoolManager.sol#55)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#missing-zero-address-validation

Variable 'ERC1967UpgradeUpgradeable._upgradeToAndCallUUPS(address,bytes,bool).slot (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/proxy/ERC1967/ERC1967UpgradeUpgradeable.sol#98)' in ERC1967Upgra
deUpgradeable._upgradeToAndCallUUPS(address,bytes,bool) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/proxy/ERC1967/ERC1967UpgradeUpgradeable.sol#87-105) potentially used before declaration: re
quire(bool,string)(slot == _IMPLEMENTATION_SLOT,ERC1967Upgrade: unsupported proxiableUUID) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/proxy/ERC1967/ERC1967UpgradeUpgradeable.sol#99)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#pre-declaration-usage-of-local-variables

AddressUpgradeable.verifyCallResult(bool,bytes,string) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/AddressUpgradeable.sol#174-194) uses assembly
        - INLINE ASM (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/AddressUpgradeable.sol#186-189)
StorageSlotUpgradeable.getAddressSlot(bytes32) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/StorageSlotUpgradeable.sol#52-57) uses assembly
        - INLINE ASM (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/StorageSlotUpgradeable.sol#54-56)
StorageSlotUpgradeable.getBooleanSlot(bytes32) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/StorageSlotUpgradeable.sol#62-67) uses assembly
        - INLINE ASM (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/StorageSlotUpgradeable.sol#64-66)
StorageSlotUpgradeable.getBytes32Slot(bytes32) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/StorageSlotUpgradeable.sol#72-77) uses assembly
        - INLINE ASM (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/StorageSlotUpgradeable.sol#74-76)
StorageSlotUpgradeable.getUint256Slot(bytes32) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/StorageSlotUpgradeable.sol#82-87) uses assembly
        - INLINE ASM (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/StorageSlotUpgradeable.sol#84-86)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#assembly-usage

Different versions of Solidity are used:
        - Version used: ['0.8.13', '^0.8.0', '^0.8.1', '^0.8.2']
        - ^0.8.0 (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/access/AccessControlUpgradeable.sol#4)
        - ^0.8.0 (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/access/IAccessControlUpgradeable.sol#4)
        - ^0.8.0 (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/interfaces/draft-IERC1822Upgradeable.sol#4)
        - ^0.8.2 (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/proxy/ERC1967/ERC1967UpgradeUpgradeable.sol#4)
        - ^0.8.0 (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/proxy/beacon/IBeaconUpgradeable.sol#4)
        - ^0.8.2 (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/proxy/utils/Initializable.sol#4)
        - ^0.8.0 (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/proxy/utils/UUPSUpgradeable.sol#4)
        - ^0.8.0 (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/security/ReentrancyGuardUpgradeable.sol#4)
        - ^0.8.0 (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/token/ERC20/IERC20Upgradeable.sol#4)
        - ^0.8.0 (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/token/ERC20/extensions/draft-IERC20PermitUpgradeable.sol#4)
        - ^0.8.0 (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/token/ERC20/utils/SafeERC20Upgradeable.sol#4)
        - ^0.8.0 (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/AddressUpgradeable.sol#4)
        - ^0.8.0 (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/ContextUpgradeable.sol#4)
        - ^0.8.0 (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/StorageSlotUpgradeable.sol#4)
        - ^0.8.0 (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/StringsUpgradeable.sol#4)
        - ^0.8.0 (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/introspection/ERC165Upgradeable.sol#4)
        - ^0.8.0 (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/introspection/IERC165Upgradeable.sol#4)
        - 0.8.13 (jobs/archimedes/brownie/contracts/AccessController.sol#2)
        - 0.8.13 (jobs/archimedes/brownie/contracts/ParameterStore.sol#2)
        - 0.8.13 (jobs/archimedes/brownie/contracts/PoolManager.sol#2)
        - ^0.8.0 (jobs/archimedes/brownie/contracts/interfaces/ICurveFi.sol#2)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#different-pragma-directives-are-used

AccessControlUpgradeable.__AccessControl_init_unchained() (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/access/AccessControlUpgradeable.sol#54-55) is never used and should be removed
AccessControlUpgradeable._setRoleAdmin(bytes32,bytes32) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/access/AccessControlUpgradeable.sol#220-224) is never used and should be removed
AccessControlUpgradeable._setupRole(bytes32,address) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/access/AccessControlUpgradeable.sol#211-213) is never used and should be removed
AddressUpgradeable.functionCall(address,bytes) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/AddressUpgradeable.sol#55-57) is never used and should be removed
AddressUpgradeable.functionCallWithValue(address,bytes,uint256) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/AddressUpgradeable.sol#114-120) is never used and should be removed
AddressUpgradeable.functionStaticCall(address,bytes) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/AddressUpgradeable.sol#147-149) is never used and should be removed
AddressUpgradeable.functionStaticCall(address,bytes,string) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/AddressUpgradeable.sol#157-166) is never used and should be removed
AddressUpgradeable.sendValue(address,uint256) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/AddressUpgradeable.sol#60-65) is never used and should be removed
ContextUpgradeable.__Context_init() (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/ContextUpgradeable.sol#18-19) is never used and should be removed
ContextUpgradeable.__Context_init_unchained() (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/ContextUpgradeable.sol#21-22) is never used and should be removed
ContextUpgradeable._msgData() (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/ContextUpgradeable.sol#27-29) is never used and should be removed
ERC165Upgradeable.__ERC165_init() (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/introspection/ERC165Upgradeable.sol#24-25) is never used and should be removed
ERC165Upgradeable.__ERC165_init_unchained() (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/introspection/ERC165Upgradeable.sol#27-28) is never used and should be removed
ERC1967UpgradeUpgradeable.__ERC1967Upgrade_init() (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/proxy/ERC1967/ERC1967UpgradeUpgradeable.sol#21-22) is never used and should be removed
ERC1967UpgradeUpgradeable.__ERC1967Upgrade_init_unchained() (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/proxy/ERC1967/ERC1967UpgradeUpgradeable.sol#24-25) is never used and should be removed
ERC1967UpgradeUpgradeable._changeAdmin(address) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/proxy/ERC1967/ERC1967UpgradeUpgradeable.sol#139-142) is never used and should be removed
ERC1967UpgradeUpgradeable._getAdmin() (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/proxy/ERC1967/ERC1967UpgradeUpgradeable.sol#122-124) is never used and should be removed
ERC1967UpgradeUpgradeable._getBeacon() (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/proxy/ERC1967/ERC1967UpgradeUpgradeable.sol#158-160) is never used and should be removed
ERC1967UpgradeUpgradeable._setAdmin(address) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/proxy/ERC1967/ERC1967UpgradeUpgradeable.sol#128-132) is never used and should be removed
ERC1967UpgradeUpgradeable._setBeacon(address) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/proxy/ERC1967/ERC1967UpgradeUpgradeable.sol#165-172) is never used and should be removed
ERC1967UpgradeUpgradeable._upgradeBeaconToAndCall(address,bytes,bool) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/proxy/ERC1967/ERC1967UpgradeUpgradeable.sol#180-190) is never used and should
be removed
Initializable._disableInitializers() (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/proxy/utils/Initializable.sol#131-137) is never used and should be removed
SafeERC20Upgradeable.safeDecreaseAllowance(IERC20Upgradeable,address,uint256) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/token/ERC20/utils/SafeERC20Upgradeable.sol#70-81) is never used and s
hould be removed
SafeERC20Upgradeable.safeIncreaseAllowance(IERC20Upgradeable,address,uint256) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/token/ERC20/utils/SafeERC20Upgradeable.sol#61-68) is never used and s
hould be removed
SafeERC20Upgradeable.safePermit(IERC20PermitUpgradeable,address,address,uint256,uint256,uint8,bytes32,bytes32) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/token/ERC20/utils/SafeERC20Upgradeab
le.sol#83-97) is never used and should be removed

Pragma version^0.8.0 (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/access/AccessControlUpgradeable.sol#4) allows old versions
Pragma version^0.8.0 (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/access/IAccessControlUpgradeable.sol#4) allows old versions
Pragma version^0.8.0 (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/interfaces/draft-IERC1822Upgradeable.sol#4) allows old versions
Pragma version^0.8.2 (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/proxy/ERC1967/ERC1967UpgradeUpgradeable.sol#4) allows old versions
Pragma version^0.8.0 (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/proxy/beacon/IBeaconUpgradeable.sol#4) allows old versions
Pragma version^0.8.2 (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/proxy/utils/Initializable.sol#4) allows old versions
Pragma version^0.8.0 (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/proxy/utils/UUPSUpgradeable.sol#4) allows old versions
Pragma version^0.8.0 (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/security/ReentrancyGuardUpgradeable.sol#4) allows old versions
Pragma version^0.8.0 (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/token/ERC20/IERC20Upgradeable.sol#4) allows old versions
Pragma version^0.8.0 (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/token/ERC20/extensions/draft-IERC20PermitUpgradeable.sol#4) allows old versions
Pragma version^0.8.0 (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/token/ERC20/utils/SafeERC20Upgradeable.sol#4) allows old versions
Pragma version^0.8.1 (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/AddressUpgradeable.sol#4) allows old versions
Pragma version^0.8.0 (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/ContextUpgradeable.sol#4) allows old versions
Pragma version^0.8.0 (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/StorageSlotUpgradeable.sol#4) allows old versions
Pragma version^0.8.0 (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/StringsUpgradeable.sol#4) allows old versions
Pragma version^0.8.0 (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/introspection/ERC165Upgradeable.sol#4) allows old versions
Pragma version^0.8.0 (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/introspection/IERC165Upgradeable.sol#4) allows old versions
Pragma version0.8.13 (jobs/archimedes/brownie/contracts/AccessController.sol#2) necessitates a version too recent to be trusted. Consider deploying with 0.6.12/0.7.6/0.8.7
Pragma version0.8.13 (jobs/archimedes/brownie/contracts/ParameterStore.sol#2) necessitates a version too recent to be trusted. Consider deploying with 0.6.12/0.7.6/0.8.7
Pragma version0.8.13 (jobs/archimedes/brownie/contracts/PoolManager.sol#2) 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 (jobs/archimedes/brownie/contracts/interfaces/ICurveFi.sol#2) allows old versions
solc-0.8.13 is not recommended for deployment
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#incorrect-versions-of-solidity

Low level call in ERC1967UpgradeUpgradeable._functionDelegateCall(address,bytes) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/proxy/ERC1967/ERC1967UpgradeUpgradeable.sol#198-204):
        - (success,returndata) = target.delegatecall(data) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/proxy/ERC1967/ERC1967UpgradeUpgradeable.sol#202)
Low level call in AddressUpgradeable.sendValue(address,uint256) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/AddressUpgradeable.sol#60-65):
        - (success) = recipient.call{value: amount}() (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/AddressUpgradeable.sol#63)
Low level call in AddressUpgradeable.functionCallWithValue(address,bytes,uint256,string) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/AddressUpgradeable.sol#128-139):
        - (success,returndata) = target.call{value: value}(data) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/AddressUpgradeable.sol#137)
Low level call in AddressUpgradeable.functionStaticCall(address,bytes,string) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/AddressUpgradeable.sol#157-166):
        - (success,returndata) = target.staticcall(data) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/AddressUpgradeable.sol#164)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#low-level-calls

Function AccessControlUpgradeable.__AccessControl_init() (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/access/AccessControlUpgradeable.sol#51-52) is not in mixedCase
Function AccessControlUpgradeable.__AccessControl_init_unchained() (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/access/AccessControlUpgradeable.sol#54-55) is not in mixedCase
Variable AccessControlUpgradeable.__gap (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/access/AccessControlUpgradeable.sol#259) is not in mixedCase
Function ERC1967UpgradeUpgradeable.__ERC1967Upgrade_init() (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/proxy/ERC1967/ERC1967UpgradeUpgradeable.sol#21-22) is not in mixedCase
Function ERC1967UpgradeUpgradeable.__ERC1967Upgrade_init_unchained() (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/proxy/ERC1967/ERC1967UpgradeUpgradeable.sol#24-25) is not in mixedCase
Variable ERC1967UpgradeUpgradeable.__gap (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/proxy/ERC1967/ERC1967UpgradeUpgradeable.sol#211) is not in mixedCase
Function UUPSUpgradeable.__UUPSUpgradeable_init() (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/proxy/utils/UUPSUpgradeable.sol#23-24) is not in mixedCase
Function UUPSUpgradeable.__UUPSUpgradeable_init_unchained() (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/proxy/utils/UUPSUpgradeable.sol#26-27) is not in mixedCase
Variable UUPSUpgradeable.__gap (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/proxy/utils/UUPSUpgradeable.sol#107) is not in mixedCase
Variable UUPSUpgradeable.__self (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/proxy/utils/UUPSUpgradeable.sol#29) is not in mixedCase
Function ReentrancyGuardUpgradeable.__ReentrancyGuard_init() (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/security/ReentrancyGuardUpgradeable.sol#40-42) is not in mixedCase
Function ReentrancyGuardUpgradeable.__ReentrancyGuard_init_unchained() (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/security/ReentrancyGuardUpgradeable.sol#44-46) is not in mixedCase
Variable ReentrancyGuardUpgradeable.__gap (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/security/ReentrancyGuardUpgradeable.sol#74) is not in mixedCase
Function IERC20PermitUpgradeable.DOMAIN_SEPARATOR() (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/token/ERC20/extensions/draft-IERC20PermitUpgradeable.sol#59) is not in mixedCase
Function ContextUpgradeable.__Context_init() (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/ContextUpgradeable.sol#18-19) is not in mixedCase
Function ContextUpgradeable.__Context_init_unchained() (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/ContextUpgradeable.sol#21-22) is not in mixedCase
Variable ContextUpgradeable.__gap (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/ContextUpgradeable.sol#36) is not in mixedCase
Function ERC165Upgradeable.__ERC165_init() (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/introspection/ERC165Upgradeable.sol#24-25) is not in mixedCase
Function ERC165Upgradeable.__ERC165_init_unchained() (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/introspection/ERC165Upgradeable.sol#27-28) is not in mixedCase
Variable ERC165Upgradeable.__gap (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/introspection/ERC165Upgradeable.sol#41) is not in mixedCase
Variable ParameterStore._maxNumberOfCycles (jobs/archimedes/brownie/contracts/ParameterStore.sol#14) is not in mixedCase
Variable ParameterStore._originationFeeRate (jobs/archimedes/brownie/contracts/ParameterStore.sol#15) is not in mixedCase
Variable ParameterStore._globalCollateralRate (jobs/archimedes/brownie/contracts/ParameterStore.sol#16) is not in mixedCase
Variable ParameterStore.rebaseFeeRate (jobs/archimedes/brownie/contracts/ParameterStore.sol#17) is not in mixedCase
```

```
Function ICurveFiCurve.remove_liquidity(uint256,uint256[2],address) (jobs/archimedes/brownie/contracts/interfaces/ICurveFi.sol#122-126) is not in mixedCase
Parameter ICurveFiCurve.remove_liquidity(uint256,uint256[2],address)._burn_amount (jobs/archimedes/brownie/contracts/interfaces/ICurveFi.sol#123) is not in mixedCase
Parameter ICurveFiCurve.remove_liquidity(uint256,uint256[2],address)._min_amounts (jobs/archimedes/brownie/contracts/interfaces/ICurveFi.sol#124) is not in mixedCase
Function ICurveFiCurve.remove_liquidity_imbalance(uint256[2],uint256) (jobs/archimedes/brownie/contracts/interfaces/ICurveFi.sol#128) is not in mixedCase
Parameter ICurveFiCurve.remove_liquidity_imbalance(uint256[2],uint256)._max_burn_amount (jobs/archimedes/brownie/contracts/interfaces/ICurveFi.sol#128) is not in mixedCase
Function ICurveFiCurve.remove_liquidity_imbalance(uint256[2],uint256,address) (jobs/archimedes/brownie/contracts/interfaces/ICurveFi.sol#130-134) is not in mixedCase
Parameter ICurveFiCurve.remove_liquidity_imbalance(uint256[2],uint256,address)._max_burn_amount (jobs/archimedes/brownie/contracts/interfaces/ICurveFi.sol#132) is not in mixedCase
Function ICurveFiCurve.calc_withdraw_one_coin(uint256,int128) (jobs/archimedes/brownie/contracts/interfaces/ICurveFi.sol#136) is not in mixedCase
Parameter ICurveFiCurve.calc_withdraw_one_coin(uint256,int128)._burn_amount (jobs/archimedes/brownie/contracts/interfaces/ICurveFi.sol#136) is not in mixedCase
Function ICurveFiCurve.calc_withdraw_one_coin(uint256,int128,bool) (jobs/archimedes/brownie/contracts/interfaces/ICurveFi.sol#138-142) is not in mixedCase
Parameter ICurveFiCurve.calc_withdraw_one_coin(uint256,int128,bool)._burn_amount (jobs/archimedes/brownie/contracts/interfaces/ICurveFi.sol#139) is not in mixedCase
Function ICurveFiCurve.remove_liquidity_one_coin(uint256,int128,uint256) (jobs/archimedes/brownie/contracts/interfaces/ICurveFi.sol#144-148) is not in mixedCase
Parameter ICurveFiCurve.remove_liquidity_one_coin(uint256,int128,uint256)._burn_amount (jobs/archimedes/brownie/contracts/interfaces/ICurveFi.sol#145) is not in mixedCase
Parameter ICurveFiCurve.remove_liquidity_one_coin(uint256,int128,uint256)._min_received (jobs/archimedes/brownie/contracts/interfaces/ICurveFi.sol#147) is not in mixedCase
Function ICurveFiCurve.remove_liquidity_one_coin(uint256,int128,uint256,address) (jobs/archimedes/brownie/contracts/interfaces/ICurveFi.sol#150-155) is not in mixedCase
Parameter ICurveFiCurve.remove_liquidity_one_coin(uint256,int128,uint256,address)._burn_amount (jobs/archimedes/brownie/contracts/interfaces/ICurveFi.sol#151) is not in mixedCase
Parameter ICurveFiCurve.remove_liquidity_one_coin(uint256,int128,uint256,address)._min_received (jobs/archimedes/brownie/contracts/interfaces/ICurveFi.sol#153) is not in mixedCase
Function ICurveFiCurve.ramp_A(uint256,uint256) (jobs/archimedes/brownie/contracts/interfaces/ICurveFi.sol#157) is not in mixedCase
Parameter ICurveFiCurve.ramp_A(uint256,uint256)._future_A (jobs/archimedes/brownie/contracts/interfaces/ICurveFi.sol#157) is not in mixedCase
Parameter ICurveFiCurve.ramp_A(uint256,uint256)._future_time (jobs/archimedes/brownie/contracts/interfaces/ICurveFi.sol#157) is not in mixedCase
Function ICurveFiCurve.stop_ramp_A() (jobs/archimedes/brownie/contracts/interfaces/ICurveFi.sol#159) is not in mixedCase
Function ICurveFiCurve.admin_balances(uint256) (jobs/archimedes/brownie/contracts/interfaces/ICurveFi.sol#161) is not in mixedCase
Function ICurveFiCurve.withdraw_admin_fees() (jobs/archimedes/brownie/contracts/interfaces/ICurveFi.sol#163) is not in mixedCase
Function ICurveFiCurve.block_timestamp_last() (jobs/archimedes/brownie/contracts/interfaces/ICurveFi.sol#173) is not in mixedCase
Function ICurveFiCurve.initial_A() (jobs/archimedes/brownie/contracts/interfaces/ICurveFi.sol#175) is not in mixedCase
Function ICurveFiCurve.future_A() (jobs/archimedes/brownie/contracts/interfaces/ICurveFi.sol#177) is not in mixedCase
Function ICurveFiCurve.initial_A_time() (jobs/archimedes/brownie/contracts/interfaces/ICurveFi.sol#179) is not in mixedCase
Function ICurveFiCurve.future_A_time() (jobs/archimedes/brownie/contracts/interfaces/ICurveFi.sol#181) is not in mixedCase
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#conformance-to-solidity-naming-conventions

ParameterStore.changeArchToLevRatio(uint256) (jobs/archimedes/brownie/contracts/ParameterStore.sol#99-103) uses literals with too many digits:
        - require(bool,string)(newArchToLevRatio < 500000000000000000000000 && newArchToLevRatio >= 10000000000000000000,new ArchToLevRatio out of range) (jobs/archimedes/brownie/contracts/ParameterStore.sol#100)
ParameterStore.changeMinPositionCollateral(uint256) (jobs/archimedes/brownie/contracts/ParameterStore.sol#111-115) uses literals with too many digits:
        - require(bool,string)(newMinPositionCollateral < (100000000000000000000000000) && newMinPositionCollateral > (10000000000000000000),New min colleateral out of range) (jobs/archimedes/brownie/contracts/ParameterStore.sol#112)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#too-many-digits

UUPSUpgradeable.__gap (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/proxy/utils/UUPSUpgradeable.sol#107) is never used in ParameterStore (jobs/archimedes/brownie/contracts/ParameterStore.sol#10
-243)
UUPSUpgradeable.__gap (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/proxy/utils/UUPSUpgradeable.sol#107) is never used in PoolManager (jobs/archimedes/brownie/contracts/PoolManager.sol#15-86)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#unused-state-variable

grantRole(bytes32,address) should be declared external:
        - AccessControlUpgradeable.grantRole(bytes32,address) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/access/AccessControlUpgradeable.sol#150-152)
revokeRole(bytes32,address) should be declared external:
        - AccessControlUpgradeable.revokeRole(bytes32,address) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/access/AccessControlUpgradeable.sol#165-167)
renounceRole(bytes32,address) should be declared external:
        - AccessControlUpgradeable.renounceRole(bytes32,address) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/access/AccessControlUpgradeable.sol#185-189)
        - AccessController.renounceRole(bytes32,address) (jobs/archimedes/brownie/contracts/AccessController.sol#62-69)
setAdmin(address) should be declared external:
        - AccessController.setAdmin(address) (jobs/archimedes/brownie/contracts/AccessController.sol#41-47)
getAddressExecutive() should be declared external:
        - AccessController.getAddressExecutive() (jobs/archimedes/brownie/contracts/AccessController.sol#95-97)
initialize() should be declared external:
        - ParameterStore.initialize() (jobs/archimedes/brownie/contracts/ParameterStore.sol#31-52)
initialize() should be declared external:
        - PoolManager.initialize() (jobs/archimedes/brownie/contracts/PoolManager.sol#26-35)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#public-function-that-could-be-declared-external
```

- PositionToken.sol

```
ERC1967UpgradeUpgradeable._functionDelegateCall(address,bytes) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/proxy/ERC1967/ERC1967UpgradeUpgradeable.sol#198-204) uses delegatecall to a input-co
ntrolled function id
        - (success,returndata) = target.delegatecall(data) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/proxy/ERC1967/ERC1967UpgradeUpgradeable.sol#202)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#controlled-delegatecall

PositionToken (jobs/archimedes/brownie/contracts/PositionToken.sol#15-132) is an upgradeable contract that does not protect its initiliaze functions: PositionToken.initialize() (jobs/archimedes/brownie/contracts/PositionToken.sol#
47-59). Anyone can delete the contract with: UUPSUpgradeable.upgradeTo(address) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/proxy/utils/UUPSUpgradeable.sol#72-75)UUPSUpgradeable.upgradeToAndC
all(address,bytes) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/proxy/utils/UUPSUpgradeable.sol#85-88)Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#unprotected-upgra
deable-contract

ERC1967UpgradeUpgradeable._upgradeToAndCallUUPS(address,bytes,bool).slot (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/proxy/ERC1967/ERC1967UpgradeUpgradeable.sol#98) is a local variable never
initialized
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#uninitialized-local-variables

ERC1967UpgradeUpgradeable._upgradeToAndCallUUPS(address,bytes,bool) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/proxy/ERC1967/ERC1967UpgradeUpgradeable.sol#87-105) ignores return value by IER
C1822ProxiableUpgradeable(newImplementation).proxiableUUID() (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/proxy/ERC1967/ERC1967UpgradeUpgradeable.sol#98-102)
ERC721Upgradeable._checkOnERC721Received(address,address,uint256,bytes) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/token/ERC721/ERC721Upgradeable.sol#399-421) ignores return value by IERC721
ReceiverUpgradeable(to).onERC721Received(_msgSender(),from,tokenId,data) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/token/ERC721/ERC721Upgradeable.sol#406-417)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#unused-return

Variable 'ERC1967UpgradeUpgradeable._upgradeToAndCallUUPS(address,bytes,bool).slot (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/proxy/ERC1967/ERC1967UpgradeUpgradeable.sol#98)' in ERC1967Upgra
deUpgradeable._upgradeToAndCallUUPS(address,bytes,bool) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/proxy/ERC1967/ERC1967UpgradeUpgradeable.sol#87-105) potentially used before declaration: re
quire(bool,string)(slot == _IMPLEMENTATION_SLOT,ERC1967Upgrade: unsupported proxiableUUID) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/proxy/ERC1967/ERC1967UpgradeUpgradeable.sol#99)
Variable 'ERC721Upgradeable._checkOnERC721Received(address,address,uint256,bytes).retval (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/token/ERC721/ERC721Upgradeable.sol#406)' in ERC721Upgradea
ble._checkOnERC721Received(address,address,uint256,bytes) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/token/ERC721/ERC721Upgradeable.sol#399-421) potentially used before declaration: retval =
= IERC721ReceiverUpgradeable.onERC721Received.selector (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/token/ERC721/ERC721Upgradeable.sol#407)
Variable 'ERC721Upgradeable._checkOnERC721Received(address,address,uint256,bytes).reason (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/token/ERC721/ERC721Upgradeable.sol#408)' in ERC721Upgradea
ble._checkOnERC721Received(address,address,uint256,bytes) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/token/ERC721/ERC721Upgradeable.sol#399-421) potentially used before declaration: reason.l
ength == 0 (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/token/ERC721/ERC721Upgradeable.sol#409)
Variable 'ERC721Upgradeable._checkOnERC721Received(address,address,uint256,bytes).reason (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/token/ERC721/ERC721Upgradeable.sol#408)' in ERC721Upgradea
ble._checkOnERC721Received(address,address,uint256,bytes) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/token/ERC721/ERC721Upgradeable.sol#399-421) potentially used before declaration: revert(u
int256,uint256)(32 + reason,mload(uint256)(reason)) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/token/ERC721/ERC721Upgradeable.sol#414)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#pre-declaration-usage-of-local-variables

Reentrancy in PositionToken.safeMint(address) (jobs/archimedes/brownie/contracts/PositionToken.sol#34-41):
        External calls:
        - _safeMint(to,positionTokenId) (jobs/archimedes/brownie/contracts/PositionToken.sol#37)
                - IERC721ReceiverUpgradeable(to).onERC721Received(_msgSender(),from,tokenId,data) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/token/ERC721/ERC721Upgradeable.sol#406-417)
        State variables written after the call(s):
        - _setApprovalForAll(to,getAddressExecutive(),true) (jobs/archimedes/brownie/contracts/PositionToken.sol#38)
                - _operatorApprovals[owner][operator] = approved (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/token/ERC721/ERC721Upgradeable.sol#378)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#reentrancy-vulnerabilities-2

Reentrancy in PositionToken.safeMint(address) (jobs/archimedes/brownie/contracts/PositionToken.sol#34-41):
        External calls:
        - _safeMint(to,positionTokenId) (jobs/archimedes/brownie/contracts/PositionToken.sol#37)
                - IERC721ReceiverUpgradeable(to).onERC721Received(_msgSender(),from,tokenId,data) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/token/ERC721/ERC721Upgradeable.sol#406-417)
        Event emitted after the call(s):
        - ApprovalForAll(owner,operator,approved) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/token/ERC721/ERC721Upgradeable.sol#379)
                - _setApprovalForAll(to,getAddressExecutive(),true) (jobs/archimedes/brownie/contracts/PositionToken.sol#38)
        - NFTCreated(positionTokenId,to) (jobs/archimedes/brownie/contracts/PositionToken.sol#39)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#reentrancy-vulnerabilities-3

ERC721Upgradeable._checkOnERC721Received(address,address,uint256,bytes) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/token/ERC721/ERC721Upgradeable.sol#399-421) uses assembly
        - INLINE ASM (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/token/ERC721/ERC721Upgradeable.sol#413-415)
AddressUpgradeable.verifyCallResult(bool,bytes,string) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/AddressUpgradeable.sol#174-194) uses assembly
        - INLINE ASM (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/AddressUpgradeable.sol#186-189)
StorageSlotUpgradeable.getAddressSlot(bytes32) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/StorageSlotUpgradeable.sol#52-57) uses assembly
        - INLINE ASM (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/StorageSlotUpgradeable.sol#54-56)
StorageSlotUpgradeable.getBooleanSlot(bytes32) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/StorageSlotUpgradeable.sol#62-67) uses assembly
        - INLINE ASM (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/StorageSlotUpgradeable.sol#64-66)
StorageSlotUpgradeable.getBytes32Slot(bytes32) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/StorageSlotUpgradeable.sol#72-77) uses assembly
        - INLINE ASM (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/StorageSlotUpgradeable.sol#74-76)
StorageSlotUpgradeable.getUint256Slot(bytes32) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/StorageSlotUpgradeable.sol#82-87) uses assembly
        - INLINE ASM (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/StorageSlotUpgradeable.sol#84-86)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#assembly-usage

Different versions of Solidity are used:
        - Version used: ['0.8.13', '^0.8.0', '^0.8.1', '^0.8.2']
        - ^0.8.0 (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/access/AccessControlUpgradeable.sol#4)
        - ^0.8.0 (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/access/IAccessControlUpgradeable.sol#4)
        - ^0.8.0 (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/interfaces/draft-IERC1822Upgradeable.sol#4)
        - ^0.8.2 (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/proxy/ERC1967/ERC1967UpgradeUpgradeable.sol#4)
        - ^0.8.0 (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/proxy/beacon/IBeaconUpgradeable.sol#4)
        - ^0.8.2 (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/proxy/utils/Initializable.sol#4)
        - ^0.8.0 (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/proxy/utils/UUPSUpgradeable.sol#4)
        - ^0.8.0 (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/security/ReentrancyGuardUpgradeable.sol#4)
        - ^0.8.0 (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/token/ERC721/ERC721Upgradeable.sol#4)
        - ^0.8.0 (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/token/ERC721/IERC721ReceiverUpgradeable.sol#4)
        - ^0.8.0 (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/token/ERC721/IERC721Upgradeable.sol#4)
        - ^0.8.0 (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/token/ERC721/extensions/ERC721BurnableUpgradeable.sol#4)
        - ^0.8.0 (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/token/ERC721/extensions/ERC721EnumerableUpgradeable.sol#4)
        - ^0.8.0 (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/token/ERC721/extensions/IERC721EnumerableUpgradeable.sol#4)
        - ^0.8.0 (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/token/ERC721/extensions/IERC721MetadataUpgradeable.sol#4)
        - ^0.8.1 (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/AddressUpgradeable.sol#4)
        - ^0.8.0 (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/ContextUpgradeable.sol#4)
        - ^0.8.0 (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/CountersUpgradeable.sol#4)
        - ^0.8.0 (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/StorageSlotUpgradeable.sol#4)
        - ^0.8.0 (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/StringsUpgradeable.sol#4)
        - ^0.8.0 (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/introspection/ERC165Upgradeable.sol#4)
        - ^0.8.0 (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/introspection/IERC165Upgradeable.sol#4)
        - 0.8.13 (jobs/archimedes/brownie/contracts/AccessController.sol#2)
        - 0.8.13 (jobs/archimedes/brownie/contracts/PositionToken.sol#2)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#different-pragma-directives-are-used
```

```
AccessControlUpgradeable.__AccessControl_init_unchained() (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/access/AccessControlUpgradeable.sol#54-55) is never used and should be removed
AccessControlUpgradeable._setRoleAdmin(bytes32,bytes32) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/access/AccessControlUpgradeable.sol#220-224) is never used and should be removed
AccessControlUpgradeable._setupRole(bytes32,address) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/access/AccessControlUpgradeable.sol#211-213) is never used and should be removed
AddressUpgradeable.functionCall(address,bytes) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/AddressUpgradeable.sol#85-87) is never used and should be removed
AddressUpgradeable.functionCall(address,bytes,string) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/AddressUpgradeable.sol#95-101) is never used and should be removed
AddressUpgradeable.functionCallWithValue(address,bytes,uint256) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/AddressUpgradeable.sol#114-120) is never used and should be removed
AddressUpgradeable.functionCallWithValue(address,bytes,uint256,string) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/AddressUpgradeable.sol#128-139) is never used and should be removed
AddressUpgradeable.functionStaticCall(address,bytes) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/AddressUpgradeable.sol#147-149) is never used and should be removed
AddressUpgradeable.functionStaticCall(address,bytes,string) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/AddressUpgradeable.sol#157-166) is never used and should be removed
AddressUpgradeable.sendValue(address,uint256) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/AddressUpgradeable.sol#60-65) is never used and should be removed
ContextUpgradeable.__Context_init() (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/ContextUpgradeable.sol#18-19) is never used and should be removed
ContextUpgradeable.__Context_init_unchained() (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/ContextUpgradeable.sol#21-22) is never used and should be removed
ContextUpgradeable._msgData() (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/ContextUpgradeable.sol#27-29) is never used and should be removed
CountersUpgradeable.decrement(CountersUpgradeable.Counter) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/CountersUpgradeable.sol#32-38) is never used and should be removed
CountersUpgradeable.reset(CountersUpgradeable.Counter) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/CountersUpgradeable.sol#40-42) is never used and should be removed
ERC165Upgradeable.__ERC165_init() (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/introspection/ERC165Upgradeable.sol#24-25) is never used and should be removed
ERC165Upgradeable.__ERC165_init_unchained() (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/introspection/ERC165Upgradeable.sol#27-28) is never used and should be removed
ERC1967UpgradeUpgradeable.__ERC1967Upgrade_init() (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/proxy/ERC1967/ERC1967UpgradeUpgradeable.sol#21-22) is never used and should be removed
ERC1967UpgradeUpgradeable.__ERC1967Upgrade_init_unchained() (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/proxy/ERC1967/ERC1967UpgradeUpgradeable.sol#24-25) is never used and should be removed
ERC1967UpgradeUpgradeable._changeAdmin(address) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/proxy/ERC1967/ERC1967UpgradeUpgradeable.sol#139-142) is never used and should be removed
ERC1967UpgradeUpgradeable._getAdmin() (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/proxy/ERC1967/ERC1967UpgradeUpgradeable.sol#122-124) is never used and should be removed
ERC1967UpgradeUpgradeable._getBeacon() (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/proxy/ERC1967/ERC1967UpgradeUpgradeable.sol#158-160) is never used and should be removed
ERC1967UpgradeUpgradeable._setAdmin(address) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/proxy/ERC1967/ERC1967UpgradeUpgradeable.sol#129-132) is never used and should be removed
ERC1967UpgradeUpgradeable._setBeacon(address) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/proxy/ERC1967/ERC1967UpgradeUpgradeable.sol#165-172) is never used and should be removed
ERC1967UpgradeUpgradeable._upgradeBeaconToAndCall(address,bytes,bool) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/proxy/ERC1967/ERC1967UpgradeUpgradeable.sol#180-190) is never used and should be removed
ERC721BurnableUpgradeable.__ERC721Burnable_init_unchained() (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/token/ERC721/extensions/ERC721BurnableUpgradeable.sol#18-19) is never used and should be removed
ERC721EnumerableUpgradeable.__ERC721Enumerable_init_unchained() (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/token/ERC721/extensions/ERC721EnumerableUpgradeable.sol#19-20) is never used and should be removed
Initializable._disableInitializers() (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/proxy/utils/Initializable.sol#131-137) is never used and should be removed
StorageSlotUpgradeable.getBytes32Slot(bytes32) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/StorageSlotUpgradeable.sol#72-77) is never used and should be removed
StorageSlotUpgradeable.getUint256Slot(bytes32) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/StorageSlotUpgradeable.sol#82-87) is never used and should be removed
StringsUpgradeable.toHexString(address) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/StringsUpgradeable.sol#72-74) is never used and should be removed
StringsUpgradeable.toHexString(uint256) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/StringsUpgradeable.sol#41-52) is never used and should be removed
UUPSUpgradeable.__UUPSUpgradeable_init_unchained() (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/proxy/utils/UUPSUpgradeable.sol#26-27) is never used and should be removed
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#dead-code

Pragma version^0.8.0 (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/access/AccessControlUpgradeable.sol#4) allows old versions
Pragma version^0.8.0 (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/access/IAccessControlUpgradeable.sol#4) allows old versions
Pragma version^0.8.0 (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/interfaces/draft-IERC1822Upgradeable.sol#4) allows old versions
Pragma version^0.8.2 (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/proxy/ERC1967/ERC1967UpgradeUpgradeable.sol#4) allows old versions
Pragma version^0.8.0 (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/proxy/beacon/IBeaconUpgradeable.sol#4) allows old versions
Pragma version^0.8.2 (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/proxy/utils/Initializable.sol#4) allows old versions
Pragma version^0.8.0 (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/proxy/utils/UUPSUpgradeable.sol#4) allows old versions
Pragma version^0.8.0 (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/security/ReentrancyGuardUpgradeable.sol#4) allows old versions
Pragma version^0.8.0 (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/token/ERC721/ERC721Upgradeable.sol#4) allows old versions
Pragma version^0.8.0 (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/token/ERC721/IERC721Receiver.sol#4) allows old versions
Pragma version^0.8.0 (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/token/ERC721/IERC721Upgradeable.sol#4) allows old versions
Pragma version^0.8.0 (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/token/ERC721/extensions/ERC721BurnableUpgradeable.sol#4) allows old versions
Pragma version^0.8.0 (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/token/ERC721/extensions/ERC721EnumerableUpgradeable.sol#4) allows old versions
Pragma version^0.8.0 (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/token/ERC721/extensions/IERC721EnumerableUpgradeable.sol#4) allows old versions
Pragma version^0.8.0 (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/token/ERC721/extensions/IERC721MetadataUpgradeable.sol#4) allows old versions
Pragma version^0.8.1 (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/AddressUpgradeable.sol#4) allows old versions
Pragma version^0.8.0 (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/ContextUpgradeable.sol#4) allows old versions
Pragma version^0.8.0 (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/CountersUpgradeable.sol#4) allows old versions
Pragma version^0.8.0 (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/StorageSlotUpgradeable.sol#4) allows old versions
Pragma version^0.8.0 (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/StringsUpgradeable.sol#4) allows old versions
Pragma version^0.8.0 (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/introspection/ERC165Upgradeable.sol#4) allows old versions
Pragma version^0.8.0 (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/introspection/IERC165Upgradeable.sol#4) allows old versions
Pragma version^0.8.13 (jobs/archimedes/brownie/contracts/AccessController.sol#2) necessitates a version too recent to be trusted. Consider deploying with 0.6.12/0.7.6/0.8.7
Pragma version^0.8.13 (jobs/archimedes/brownie/contracts/PositionToken.sol#2) necessitates a version too recent to be trusted. Consider deploying with 0.6.12/0.7.6/0.8.7
solc-0.8.13 is not recommended for deployment
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#incorrect-versions-of-solidity

Low level call in ERC1967UpgradeUpgradeable._functionDelegateCall(address,bytes) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/proxy/ERC1967/ERC1967UpgradeUpgradeable.sol#198-204):
        - (success,returndata) = target.delegatecall(data) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/proxy/ERC1967/ERC1967UpgradeUpgradeable.sol#202)
Low level call in AddressUpgradeable.sendValue(address,uint256) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/AddressUpgradeable.sol#60-65):
        - (success) = recipient.call{value: amount}() (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/AddressUpgradeable.sol#63)
Low level call in AddressUpgradeable.functionCallWithValue(address,bytes,uint256,string) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/AddressUpgradeable.sol#128-139):
        - (success,returndata) = target.call{value: value}(data) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/AddressUpgradeable.sol#137)
Low level call in AddressUpgradeable.functionStaticCall(address,bytes,string) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/AddressUpgradeable.sol#157-166):
        - (success,returndata) = target.staticcall(data) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/AddressUpgradeable.sol#164)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#low-level-calls

Function AccessControlUpgradeable.__AccessControl_init() (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/access/AccessControlUpgradeable.sol#51-52) is not in mixedCase
Function AccessControlUpgradeable.__AccessControl_init_unchained() (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/access/AccessControlUpgradeable.sol#54-55) is not in mixedCase
Variable AccessControlUpgradeable.__gap (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/access/AccessControlUpgradeable.sol#250) is not in mixedCase
Function ERC1967UpgradeUpgradeable.__ERC1967Upgrade_init() (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/proxy/ERC1967/ERC1967UpgradeUpgradeable.sol#21-22) is not in mixedCase
Function ERC1967UpgradeUpgradeable.__ERC1967Upgrade_init_unchained() (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/proxy/ERC1967/ERC1967UpgradeUpgradeable.sol#24-25) is not in mixedCase
Variable ERC1967UpgradeUpgradeable.__gap (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/proxy/ERC1967/ERC1967UpgradeUpgradeable.sol#211) is not in mixedCase
Function UUPSUpgradeable.__UUPSUpgradeable_init() (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/ContextUpgradeable.sol#4) allows old versions
Function UUPSUpgradeable.__UUPSUpgradeable_init_unchained() (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/proxy/utils/UUPSUpgradeable.sol#26-27) is not in mixedCase
Variable UUPSUpgradeable.__gap (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/proxy/utils/UUPSUpgradeable.sol#107) is not in mixedCase
Variable UUPSUpgradeable.__self (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/proxy/utils/UUPSUpgradeable.sol#20) is not in mixedCase
Function ReentrancyGuardUpgradeable.__ReentrancyGuard_init() (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/security/ReentrancyGuardUpgradeable.sol#40-42) is not in mixedCase
Function ReentrancyGuardUpgradeable.__ReentrancyGuard_init_unchained() (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/security/ReentrancyGuardUpgradeable.sol#44-46) is not in mixedCase
Variable ReentrancyGuardUpgradeable.__gap (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/security/ReentrancyGuardUpgradeable.sol#74) is not in mixedCase
Function ERC721Upgradeable.__ERC721_init(string,string) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/token/ERC721/ERC721Upgradeable.sol#45-47) is not in mixedCase
Function ERC721Upgradeable.__ERC721_init_unchained(string,string) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/token/ERC721/ERC721Upgradeable.sol#49-52) is not in mixedCase
Variable ERC721Upgradeable.__gap (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/token/ERC721/ERC721Upgradeable.sol#465) is not in mixedCase
Function ERC721BurnableUpgradeable.__ERC721Burnable_init() (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/token/ERC721/extensions/ERC721BurnableUpgradeable.sol#15-16) is not in mixedCase
Function ERC721BurnableUpgradeable.__ERC721Burnable_init_unchained() (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/token/ERC721/extensions/ERC721BurnableUpgradeable.sol#18-19) is not in mixedCase
Variable ERC721BurnableUpgradeable.__gap (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/token/ERC721/extensions/ERC721BurnableUpgradeable.sol#38) is not in mixedCase
Function ERC721EnumerableUpgradeable.__ERC721Enumerable_init() (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/token/ERC721/extensions/ERC721EnumerableUpgradeable.sol#16-17) is not in mixedCase
Function ERC721EnumerableUpgradeable.__ERC721Enumerable_init_unchained() (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/token/ERC721/extensions/ERC721EnumerableUpgradeable.sol#19-20) is not in mixedCase
Variable ERC721EnumerableUpgradeable.__gap (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/token/ERC721/extensions/ERC721EnumerableUpgradeable.sol#175) is not in mixedCase
Function ContextUpgradeable.__Context_init() (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/ContextUpgradeable.sol#18-19) is not in mixedCase
Function ContextUpgradeable.__Context_init_unchained() (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/ContextUpgradeable.sol#21-22) is not in mixedCase
Variable ContextUpgradeable.__gap (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/ContextUpgradeable.sol#36) is not in mixedCase
Function ERC165Upgradeable.__ERC165_init() (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/introspection/ERC165Upgradeable.sol#24-25) is not in mixedCase
Function ERC165Upgradeable.__ERC165_init_unchained() (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/introspection/ERC165Upgradeable.sol#27-28) is not in mixedCase
Variable ERC165Upgradeable.__gap (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/introspection/ERC165Upgradeable.sol#41) is not in mixedCase
Variable PositionToken._addressToTokensOwnedMapping (jobs/archimedes/brownie/contracts/PositionToken.sol#31) is not in mixedCase
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#conformance-to-solidity-naming-conventions

UUPSUpgradeable.__gap (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/proxy/utils/UUPSUpgradeable.sol#107) is never used in PositionToken (jobs/archimedes/brownie/contracts/PositionToken.sol#15-132)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#unused-state-variable

UUPSUpgradeable.__gap (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/proxy/utils/UUPSUpgradeable.sol#107) is never used in PositionToken (jobs/archimedes/brownie/contracts/PositionToken.sol#15-132)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#unused-state-variable

grantRole(bytes32,address) should be declared external:
        - AccessControlUpgradeable.grantRole(bytes32,address) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/access/AccessControlUpgradeable.sol#150-152)
revokeRole(bytes32,address) should be declared external:
        - AccessControlUpgradeable.revokeRole(bytes32,address) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/access/AccessControlUpgradeable.sol#165-167)
renounceRole(bytes32,address) should be declared external:
        - AccessControlUpgradeable.renounceRole(bytes32,address) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/access/AccessControlUpgradeable.sol#185-189)
        - AccessController.renounceRole(bytes32,address) (jobs/archimedes/brownie/contracts/AccessController.sol#62-69)
name() should be declared external:
        - ERC721Upgradeable.name() (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/token/ERC721/ERC721Upgradeable.sol#84-86)
symbol() should be declared external:
        - ERC721Upgradeable.symbol() (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/token/ERC721/ERC721Upgradeable.sol#91-93)
tokenURI(uint256) should be declared external:
        - ERC721Upgradeable.tokenURI(uint256) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/token/ERC721/ERC721Upgradeable.sol#98-103)
approve(address,uint256) should be declared external:
        - ERC721Upgradeable.approve(address,uint256) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/token/ERC721/ERC721Upgradeable.sol#117-127)
setApprovalForAll(address,bool) should be declared external:
        - ERC721Upgradeable.setApprovalForAll(address,bool) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/token/ERC721/ERC721Upgradeable.sol#141-143)
transferFrom(address,address,uint256) should be declared external:
        - ERC721Upgradeable.transferFrom(address,address,uint256) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/token/ERC721/ERC721Upgradeable.sol#155-164)
safeTransferFrom(address,address,uint256) should be declared external:
        - ERC721Upgradeable.safeTransferFrom(address,address,uint256) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/token/ERC721/ERC721Upgradeable.sol#169-175)
tokenOfOwnerByIndex(address,uint256) should be declared external:
        - ERC721EnumerableUpgradeable.tokenOfOwnerByIndex(address,uint256) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/token/ERC721/extensions/ERC721EnumerableUpgradeable.sol#43-46)
tokenByIndex(uint256) should be declared external:
        - ERC721EnumerableUpgradeable.tokenByIndex(uint256) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/token/ERC721/extensions/ERC721EnumerableUpgradeable.sol#58-61)
setAdmin(address) should be declared external:
        - AccessController.setAdmin(address) (jobs/archimedes/brownie/contracts/AccessController.sol#41-47)
initialize() should be declared external:
        - PositionToken.initialize() (jobs/archimedes/brownie/contracts/PositionToken.sol#47-59)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#public-function-that-could-be-declared-external
```

- `VaultOUSD.sol`

AUTOMATED TESTING

```
ERC1967UpgradeUpgradeable._functionDelegateCall(address,bytes) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/proxy/ERC1967/ERC1967UpgradeUpgradeable.sol#198-204) uses delegatecall to a input-co
ntrolled function id
        - (success,returndata) = target.delegatecall(data) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/proxy/ERC1967/ERC1967UpgradeUpgradeable.sol#202)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#controlled-delegatecall

Reentrancy in VaultOUSD.archimedesRedeem(uint256,address,address) (jobs/archimedes/brownie/contracts/VaultOUSD.sol#55-78):
        External calls:
        - _takeRebaseFees() (jobs/archimedes/brownie/contracts/VaultOUSD.sol#60)
                - _ousd.transfer(_paramStore.getTreasuryAddress(),feeToCollect) (jobs/archimedes/brownie/contracts/VaultOUSD.sol#138)
        - redeemedAmountInAssets = super.redeem(shares,receiver,owner) (jobs/archimedes/brownie/contracts/VaultOUSD.sol#61)
                - returndata = address(token).functionCall(data,SafeERC20: low-level call failed) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/token/ERC20/utils/SafeERC20Upgradeable.sol#110)
                - SafeERC20Upgradeable.safeTransfer(_asset,receiver,assets) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/token/ERC20/extensions/ERC4626Upgradeable.sol#219)
                - (success,returndata) = target.call{value: value}(data) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/AddressUpgradeable.sol#137)
        External calls sending eth:
        - redeemedAmountInAssets = super.redeem(shares,receiver,owner) (jobs/archimedes/brownie/contracts/VaultOUSD.sol#61)
                - (success,returndata) = target.call{value: value}(data) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/AddressUpgradeable.sol#137)
        State variables written after the call(s):
        - _assetsHandledByArchimedes = 0 (jobs/archimedes/brownie/contracts/VaultOUSD.sol#71)
        - _assetsHandledByArchimedes = _assetsHandledByArchimedes - redeemedAmountInAssets (jobs/archimedes/brownie/contracts/VaultOUSD.sol#73)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#reentrancy-vulnerabilities

VaultOUSD._takeRebaseFees() (jobs/archimedes/brownie/contracts/VaultOUSD.sol#108-140) ignores return value by _ousd.transfer(_paramStore.getTreasuryAddress(),feeToCollect) (jobs/archimedes/brownie/contracts/VaultOUSD.sol#138)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#unchecked-transfer

ParameterStore (jobs/archimedes/brownie/contracts/ParameterStore.sol#10-243) is an upgradeable contract that does not protect its initilaze functions: ParameterStore.initialize() (jobs/archimedes/brownie/contracts/ParameterStore.
sol#31-52). Anyone can delete the contract with: UUPSUpgradeable.upgradeTo(address) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/proxy/utils/UUPSUpgradeable.sol#72-75)UUPSUpgradeable.upgradeTo
AndCall(address,bytes) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/proxy/utils/UUPSUpgradeable.sol#85-88)VaultOUSD (jobs/archimedes/brownie/contracts/VaultOUSD.sol#19-146) is an upgradeable c
ontract that does not protect its initilaze functions: VaultOUSD.initialize(IERC20MetadataUpgradeable,string,string) (jobs/archimedes/brownie/contracts/VaultOUSD.sol#88-106). Anyone can delete the contract with: UUPSUpgradeable.u
pgradeTo(address) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/proxy/utils/UUPSUpgradeable.sol#72-75)UUPSUpgradeable.upgradeToAndCall(address,bytes) (.brownie/packages/OpenZeppelin/openzeppeli
n-contracts@4.7.0/contracts-upgradeable/proxy/utils/UUPSUpgradeable.sol#85-88)Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#unprotected-upgradeable-contract

MathUpgradeable.mulDiv(uint256,uint256,uint256) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/math/MathUpgradeable.sol#55-135) performs a multiplication on the result of a division:
        -denominator = denominator / twos (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/math/MathUpgradeable.sol#102)
        -inverse = (3 * denominator) ^ 2 (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/math/MathUpgradeable.sol#117)
MathUpgradeable.mulDiv(uint256,uint256,uint256) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/math/MathUpgradeable.sol#55-135) performs a multiplication on the result of a division:
        -denominator = denominator / twos (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/math/MathUpgradeable.sol#102)
        -inverse *= 2 - denominator * inverse (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/math/MathUpgradeable.sol#121)
MathUpgradeable.mulDiv(uint256,uint256,uint256) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/math/MathUpgradeable.sol#55-135) performs a multiplication on the result of a division:
        -denominator = denominator / twos (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/math/MathUpgradeable.sol#102)
        -inverse *= 2 - denominator * inverse (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/math/MathUpgradeable.sol#122)
MathUpgradeable.mulDiv(uint256,uint256,uint256) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/math/MathUpgradeable.sol#55-135) performs a multiplication on the result of a division:
        -denominator = denominator / twos (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/math/MathUpgradeable.sol#102)
        -inverse *= 2 - denominator * inverse (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/math/MathUpgradeable.sol#123)
MathUpgradeable.mulDiv(uint256,uint256,uint256) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/math/MathUpgradeable.sol#55-135) performs a multiplication on the result of a division:
        -denominator = denominator / twos (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/math/MathUpgradeable.sol#102)
        -inverse *= 2 - denominator * inverse (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/math/MathUpgradeable.sol#124)
MathUpgradeable.mulDiv(uint256,uint256,uint256) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/math/MathUpgradeable.sol#55-135) performs a multiplication on the result of a division:
        -denominator = denominator / twos (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/math/MathUpgradeable.sol#102)
        -inverse *= 2 - denominator * inverse (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/math/MathUpgradeable.sol#125)
MathUpgradeable.mulDiv(uint256,uint256,uint256) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/math/MathUpgradeable.sol#55-135) performs a multiplication on the result of a division:
        -denominator = denominator / twos (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/math/MathUpgradeable.sol#102)
        -inverse *= 2 - denominator * inverse (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/math/MathUpgradeable.sol#126)
MathUpgradeable.mulDiv(uint256,uint256,uint256) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/math/MathUpgradeable.sol#55-135) performs a multiplication on the result of a division:
        -prod0 = prod0 / twos (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/math/MathUpgradeable.sol#105)
        -result = prod0 * inverse (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/math/MathUpgradeable.sol#132)
ParameterStore.calculateLeverageAllowedForArch(uint256) (jobs/archimedes/brownie/contracts/ParameterStore.sol#218-220) performs a multiplication on the result of a division:
        -(_archToLevRatio / 1000000000000000000) * archAmount (jobs/archimedes/brownie/contracts/ParameterStore.sol#219)
VaultOUSD._takeRebaseFees() (jobs/archimedes/brownie/contracts/VaultOUSD.sol#108-140) performs a multiplication on the result of a division:
        -unhandledRebasePayment = ((totalAssets() - _assetsHandledByArchimedes) / 100) * 100 (jobs/archimedes/brownie/contracts/VaultOUSD.sol#126)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#divide-before-multiply

ERC4626Upgradeable._convertToAssets(uint256,MathUpgradeable.Rounding) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/token/ERC20/extensions/ERC4626Upgradeable.sol#168-174) uses a dangerous stric
t equality:
        - (supply == 0) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/token/ERC20/extensions/ERC4626Upgradeable.sol#170-173)
ERC4626Upgradeable._convertToShares(uint256,MathUpgradeable.Rounding) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/token/ERC20/extensions/ERC4626Upgradeable.sol#157-163) uses a dangerous stric
t equality:
        - (assets == 0 || supply == 0) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/token/ERC20/extensions/ERC4626Upgradeable.sol#159-162)
ERC4626Upgradeable._isVaultCollateralized() (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/token/ERC20/extensions/ERC4626Upgradeable.sol#224-226) uses a dangerous strict equality:
        - totalAssets() > 0 || totalSupply() == 0 (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/token/ERC20/extensions/ERC4626Upgradeable.sol#225)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#dangerous-strict-equalities

Reentrancy in VaultOUSD.archimedesDeposit(uint256,address) (jobs/archimedes/brownie/contracts/VaultOUSD.sol#37-41):
        External calls:
        - _takeRebaseFees() (jobs/archimedes/brownie/contracts/VaultOUSD.sol#38)
                - _ousd.transfer(_paramStore.getTreasuryAddress(),feeToCollect) (jobs/archimedes/brownie/contracts/VaultOUSD.sol#138)
        State variables written after the call(s):
        - _assetsHandledByArchimedes += assets (jobs/archimedes/brownie/contracts/VaultOUSD.sol#39)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#reentrancy-vulnerabilities-1

ERC1967UpgradeUpgradeable._upgradeToAndCallUUPS(address,bytes,bool).slot (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/proxy/ERC1967/ERC1967UpgradeUpgradeable.sol#98) is a local variable never
initialized
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#uninitialized-local-variables

ERC1967UpgradeUpgradeable._upgradeToAndCallUUPS(address,bytes,bool) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/proxy/ERC1967/ERC1967UpgradeUpgradeable.sol#87-105) ignores return value by IER
C1822Proxiable(newImplementation).proxiableUUID() (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/proxy/ERC1967/ERC1967UpgradeUpgradeable.sol#98-102)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#unused-return

VaultOUSD.initialize(IERC20MetadataUpgradeable,string,string).asset (jobs/archimedes/brownie/contracts/VaultOUSD.sol#89) shadows:
        - ERC4626Upgradeable.asset() (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/token/ERC20/extensions/ERC4626Upgradeable.sol#44-46) (function)
        - IERC4626Upgradeable.asset() (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/interfaces/IERC4626Upgradeable.sol#32) (function)
VaultOUSD.initialize(IERC20MetadataUpgradeable,string,string).name (jobs/archimedes/brownie/contracts/VaultOUSD.sol#90) shadows:
        - ERC20Upgradeable.name() (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/token/ERC20/ERC20Upgradeable.sol#67-69) (function)
        - IERC20MetadataUpgradeable.name() (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/token/ERC20/extensions/IERC20MetadataUpgradeable.sol#17) (function)
VaultOUSD.initialize(IERC20MetadataUpgradeable,string,string).symbol (jobs/archimedes/brownie/contracts/VaultOUSD.sol#91) shadows:
        - ERC20Upgradeable.symbol() (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/token/ERC20/ERC20Upgradeable.sol#75-77) (function)
        - IERC20MetadataUpgradeable.symbol() (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/token/ERC20/extensions/IERC20MetadataUpgradeable.sol#22) (function)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#local-variable-shadowing

VaultOUSD.archimedesDeposit(uint256,address) (jobs/archimedes/brownie/contracts/VaultOUSD.sol#37-41) should emit an event for:
        - _assetsHandledByArchimedes += assets (jobs/archimedes/brownie/contracts/VaultOUSD.sol#39)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#missing-events-arithmetic

Variable 'ERC1967UpgradeUpgradeable._upgradeToAndCallUUPS(address,bytes,bool).slot (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/proxy/ERC1967/ERC1967UpgradeUpgradeable.sol#98)' in ERC1967Upgra
deUpgradeable._upgradeToAndCallUUPS(address,bytes,bool) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/proxy/ERC1967/ERC1967UpgradeUpgradeable.sol#87-105) potentially used before declaration: re
quire(bool,string)(slot == _IMPLEMENTATION_SLOT,ERC1967Upgrade: unsupported proxiableUUID) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/proxy/ERC1967/ERC1967UpgradeUpgradeable.sol#99)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#pre-declaration-usage-of-local-variables

Reentrancy in ERC4626Upgradeable._deposit(address,address,uint256,uint256) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/token/ERC20/extensions/ERC4626Upgradeable.sol#179-196):
        External calls:
        - SafeERC20Upgradeable.safeTransferFrom(_asset,caller,address(this),assets) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/token/ERC20/extensions/ERC4626Upgradeable.sol#192)
        State variables written after the call(s):
        - _mint(receiver,shares) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/token/ERC20/extensions/ERC4626Upgradeable.sol#193)
                - _balances[account] += amount (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/token/ERC20/ERC20Upgradeable.sol#268)
        - _mint(receiver,shares) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/token/ERC20/extensions/ERC4626Upgradeable.sol#193)
                - _totalSupply += amount (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/token/ERC20/ERC20Upgradeable.sol#267)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#reentrancy-vulnerabilities-2

Reentrancy in ERC4626Upgradeable._deposit(address,address,uint256,uint256) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/token/ERC20/extensions/ERC4626Upgradeable.sol#179-196):
        External calls:
        - SafeERC20Upgradeable.safeTransferFrom(_asset,caller,address(this),assets) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/token/ERC20/extensions/ERC4626Upgradeable.sol#192)
        Event emitted after the call(s):
        - Deposit(caller,receiver,assets,shares) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/token/ERC20/extensions/ERC4626Upgradeable.sol#195)
        - Transfer(address(0),account,amount) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/token/ERC20/ERC20Upgradeable.sol#269)
                - _mint(receiver,shares) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/token/ERC20/extensions/ERC4626Upgradeable.sol#193)
Reentrancy in ERC4626Upgradeable._withdraw(address,address,address,uint256,uint256) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/token/ERC20/extensions/ERC4626Upgradeable.sol#201-222):
        External calls:
        - SafeERC20Upgradeable.safeTransfer(_asset,receiver,assets) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/token/ERC20/extensions/ERC4626Upgradeable.sol#219)
        Event emitted after the call(s):
        - Withdraw(caller,receiver,owner,assets,shares) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/token/ERC20/extensions/ERC4626Upgradeable.sol#221)
Reentrancy in VaultOUSD.archimedesDeposit(uint256,address) (jobs/archimedes/brownie/contracts/VaultOUSD.sol#37-41):
        External calls:
        - _takeRebaseFees() (jobs/archimedes/brownie/contracts/VaultOUSD.sol#38)
                - _ousd.transfer(_paramStore.getTreasuryAddress(),feeToCollect) (jobs/archimedes/brownie/contracts/VaultOUSD.sol#138)
        - super.deposit(assets,receiver) (jobs/archimedes/brownie/contracts/VaultOUSD.sol#40)
                - SafeERC20Upgradeable.safeTransferFrom(_asset,caller,address(this),assets) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/token/ERC20/extensions/ERC4626Upgradeable.sol#192)
                - returndata = address(token).functionCall(data,SafeERC20: low-level call failed) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/token/ERC20/utils/SafeERC20Upgradeable.sol#110)
                - (success,returndata) = target.call{value: value}(data) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/AddressUpgradeable.sol#137)
        External calls sending eth:
        - super.deposit(assets,receiver) (jobs/archimedes/brownie/contracts/VaultOUSD.sol#40)
                - (success,returndata) = target.call{value: value}(data) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/AddressUpgradeable.sol#137)
        Event emitted after the call(s):
        - Deposit(caller,receiver,assets,shares) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/token/ERC20/extensions/ERC4626Upgradeable.sol#195)
                - super.deposit(assets,receiver) (jobs/archimedes/brownie/contracts/VaultOUSD.sol#40)
        - Transfer(address(0),account,amount) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/token/ERC20/ERC20Upgradeable.sol#269)
                - super.deposit(assets,receiver) (jobs/archimedes/brownie/contracts/VaultOUSD.sol#40)
```

```
Reentrancy in VaultOUSD.archimedesRedeem(uint256,address,address) (jobs/archimedes/brownie/contracts/VaultOUSD.sol#55-78):
        External calls:
        - _takeRebaseFees() (jobs/archimedes/brownie/contracts/VaultOUSD.sol#60)
                - _ousd.transfer(_paramStore.getTreasuryAddress(),feeToCollect) (jobs/archimedes/brownie/contracts/VaultOUSD.sol#138)
        - redeemedAmountInAssets = super.redeem(shares,receiver,owner) (jobs/archimedes/brownie/contracts/VaultOUSD.sol#61)
                - returndata = address(token).functionCall(data,SafeERC20: low-level call failed) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/token/ERC20/utils/SafeERC20Upgradeable.sol#110)
                - SafeERC20Upgradeable.safeTransfer(_asset,receiver,assets) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/token/ERC20/extensions/ERC4626Upgradeable.sol#219)
                - (success,returndata) = target.call{value: value}(data) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/AddressUpgradeable.sol#137)
        External calls sending eth:
        - redeemedAmountInAssets = super.redeem(shares,receiver,owner) (jobs/archimedes/brownie/contracts/VaultOUSD.sol#61)
                - (success,returndata) = target.call{value: value}(data) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/AddressUpgradeable.sol#137)
        Event emitted after the call(s):
        - Approval(owner,spender,amount) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/token/ERC20/ERC20Upgradeable.sol#324)
                - redeemedAmountInAssets = super.redeem(shares,receiver,owner) (jobs/archimedes/brownie/contracts/VaultOUSD.sol#61)
        - Transfer(account,address(0),amount) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/token/ERC20/ERC20Upgradeable.sol#297)
                - redeemedAmountInAssets = super.redeem(shares,receiver,owner) (jobs/archimedes/brownie/contracts/VaultOUSD.sol#61)
        - Withdraw(caller,receiver,owner,assets,shares) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/token/ERC20/extensions/ERC4626Upgradeable.sol#221)
                - redeemedAmountInAssets = super.redeem(shares,receiver,owner) (jobs/archimedes/brownie/contracts/VaultOUSD.sol#61)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#reentrancy-vulnerabilities-3

AddressUpgradeable.verifyCallResult(bool,bytes,string) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/AddressUpgradeable.sol#174-194) uses assembly
        - INLINE ASM (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/AddressUpgradeable.sol#186-189)
StorageSlotUpgradeable.getAddressSlot(bytes32) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/StorageSlotUpgradeable.sol#52-57) uses assembly
        - INLINE ASM (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/StorageSlotUpgradeable.sol#54-56)
StorageSlotUpgradeable.getBooleanSlot(bytes32) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/StorageSlotUpgradeable.sol#62-67) uses assembly
        - INLINE ASM (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/StorageSlotUpgradeable.sol#64-66)
StorageSlotUpgradeable.getBytes32Slot(bytes32) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/StorageSlotUpgradeable.sol#72-77) uses assembly
        - INLINE ASM (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/StorageSlotUpgradeable.sol#74-76)
StorageSlotUpgradeable.getUint256Slot(bytes32) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/StorageSlotUpgradeable.sol#82-87) uses assembly
        - INLINE ASM (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/StorageSlotUpgradeable.sol#84-86)
MathUpgradeable.mulDiv(uint256,uint256,uint256) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/math/MathUpgradeable.sol#55-135) uses assembly
        - INLINE ASM (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/math/MathUpgradeable.sol#66-70)
        - INLINE ASM (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/math/MathUpgradeable.sol#86-93)
        - INLINE ASM (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/math/MathUpgradeable.sol#100-109)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#assembly-usage

Different versions of Solidity are used:
        - Version used: ['0.8.13', '^0.8.0', '^0.8.1', '^0.8.2']
        - ^0.8.0 (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/access/AccessControlUpgradeable.sol#4)
        - ^0.8.0 (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/access/IAccessControlUpgradeable.sol#4)
        - ^0.8.0 (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/interfaces/IERC20MetadataUpgradeable.sol#4)
        - ^0.8.0 (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/interfaces/IERC4626Upgradeable.sol#4)
        - ^0.8.0 (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/interfaces/draft-IERC1822Upgradeable.sol#4)
        - ^0.8.2 (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/proxy/ERC1967/ERC1967UpgradeUpgradeable.sol#4)
        - ^0.8.0 (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/proxy/beacon/IBeaconUpgradeable.sol#4)
        - ^0.8.2 (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/proxy/utils/Initializable.sol#4)
        - ^0.8.0 (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/proxy/utils/UUPSUpgradeable.sol#4)
        - ^0.8.0 (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/security/ReentrancyGuardUpgradeable.sol#4)
        - ^0.8.0 (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/token/ERC20/ERC20Upgradeable.sol#4)
        - ^0.8.0 (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/token/ERC20/IERC20Upgradeable.sol#4)
        - ^0.8.0 (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/token/ERC20/extensions/ERC4626Upgradeable.sol#4)
        - ^0.8.0 (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/token/ERC20/extensions/IERC20MetadataUpgradeable.sol#4)
        - ^0.8.0 (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/token/ERC20/extensions/draft-IERC20PermitUpgradeable.sol#4)
AccessControlUpgradeable.__AccessControl_init_unchained() (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/access/AccessControlUpgradeable.sol#54-55) is never used and should be removed
AccessControlUpgradeable._setRoleAdmin(bytes32,bytes32) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/access/AccessControlUpgradeable.sol#220-224) is never used and should be removed
AccessControlUpgradeable._setupRole(bytes32,address) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/access/AccessControlUpgradeable.sol#211-213) is never used and should be removed
AddressUpgradeable.functionCall(address,bytes) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/AddressUpgradeable.sol#85-87) is never used and should be removed
AddressUpgradeable.functionCallWithValue(address,bytes,uint256) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/AddressUpgradeable.sol#114-120) is never used and should be removed
AddressUpgradeable.functionStaticCall(address,bytes) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/AddressUpgradeable.sol#147-149) is never used and should be removed
AddressUpgradeable.functionStaticCall(address,bytes,string) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/AddressUpgradeable.sol#157-166) is never used and should be removed
AddressUpgradeable.sendValue(address,uint256) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/AddressUpgradeable.sol#60-65) is never used and should be removed
ContextUpgradeable.__Context_init() (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/ContextUpgradeable.sol#18-19) is never used and should be removed
ContextUpgradeable.__Context_init_unchained() (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/ContextUpgradeable.sol#21-22) is never used and should be removed
ContextUpgradeable._msgData() (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/ContextUpgradeable.sol#27-29) is never used and should be removed
ERC165Upgradeable.__ERC165_init() (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/introspection/ERC165Upgradeable.sol#24-25) is never used and should be removed
ERC165Upgradeable.__ERC165_init_unchained() (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/introspection/ERC165Upgradeable.sol#27-28) is never used and should be removed
ERC1967UpgradeUpgradeable.__ERC1967Upgrade_init() (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/proxy/ERC1967/ERC1967UpgradeUpgradeable.sol#21-22) is never used and should be removed
ERC1967UpgradeUpgradeable.__ERC1967Upgrade_init_unchained() (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/proxy/ERC1967/ERC1967UpgradeUpgradeable.sol#24-25) is never used and should be removed
ERC1967UpgradeUpgradeable._changeAdmin(address) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/proxy/ERC1967/ERC1967UpgradeUpgradeable.sol#139-142) is never used and should be removed
ERC1967UpgradeUpgradeable._getAdmin() (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/proxy/ERC1967/ERC1967UpgradeUpgradeable.sol#122-124) is never used and should be removed
ERC1967UpgradeUpgradeable._getBeacon() (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/proxy/ERC1967/ERC1967UpgradeUpgradeable.sol#158-160) is never used and should be removed
ERC1967UpgradeUpgradeable._setAdmin(address) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/proxy/ERC1967/ERC1967UpgradeUpgradeable.sol#129-132) is never used and should be removed
ERC1967UpgradeUpgradeable._setBeacon(address) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/proxy/ERC1967/ERC1967UpgradeUpgradeable.sol#165-172) is never used and should be removed
ERC1967UpgradeUpgradeable._upgradeBeaconToAndCall(address,bytes,bool) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/proxy/ERC1967/ERC1967UpgradeUpgradeable.sol#180-190) is never used and should be removed
Initializable._disableInitializers() (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/proxy/utils/Initializable.sol#131-137) is never used and should be removed
MathUpgradeable.average(uint256,uint256) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/math/MathUpgradeable.sol#34-37) is never used and should be removed
MathUpgradeable.ceilDiv(uint256,uint256) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/math/MathUpgradeable.sol#45-48) is never used and should be removed
MathUpgradeable.max(uint256,uint256) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/math/MathUpgradeable.sol#19-21) is never used and should be removed
MathUpgradeable.min(uint256,uint256) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/math/MathUpgradeable.sol#26-28) is never used and should be removed
MathUpgradeable.sqrt(uint256) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/math/MathUpgradeable.sol#158-214) is never used and should be removed
MathUpgradeable.sqrt(uint256,MathUpgradeable.Rounding) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/math/MathUpgradeable.sol#219-225) is never used and should be removed
SafeERC20Upgradeable.safeApprove(IERC20Upgradeable,address,uint256) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/token/ERC20/utils/SafeERC20Upgradeable.sol#46-59) is never used and should be removed
SafeERC20Upgradeable.safeDecreaseAllowance(IERC20Upgradeable,address,uint256) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/token/ERC20/utils/SafeERC20Upgradeable.sol#70-81) is never used and should be removed
SafeERC20Upgradeable.safeIncreaseAllowance(IERC20Upgradeable,address,uint256) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/token/ERC20/utils/SafeERC20Upgradeable.sol#61-68) is never used and should be removed
SafeERC20Upgradeable.safePermit(IERC20PermitUpgradeable,address,address,uint256,uint256,uint8,bytes32,bytes32) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/token/ERC20/utils/SafeERC20Upgradeable.sol#83-97) is never used and should be removed
StorageSlotUpgradeable.getBytes32Slot(bytes32) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/StorageSlotUpgradeable.sol#72-77) is never used and should be removed
StorageSlotUpgradeable.getUint256Slot(bytes32) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/StorageSlotUpgradeable.sol#82-87) is never used and should be removed
StringsUpgradeable.toHexString(address) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/StringsUpgradeable.sol#72-74) is never used and should be removed
StringsUpgradeable.toHexString(uint256) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/StringsUpgradeable.sol#41-52) is never used and should be removed
StringsUpgradeable.toString(uint256) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/StringsUpgradeable.sol#16-36) is never used and should be removed
UUPSUpgradeable.__UUPSUpgradeable_init_unchained() (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/proxy/utils/UUPSUpgradeable.sol#26-27) is never used and should be removed
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#dead-code

Pragma version^0.8.0 (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/access/AccessControlUpgradeable.sol#4) allows old versions
Pragma version^0.8.0 (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/access/IAccessControlUpgradeable.sol#4) allows old versions
Pragma version^0.8.0 (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/interfaces/IERC20MetadataUpgradeable.sol#4) allows old versions
Pragma version^0.8.0 (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/interfaces/IERC4626Upgradeable.sol#4) allows old versions
Pragma version^0.8.0 (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/interfaces/draft-IERC1822Upgradeable.sol#4) allows old versions
Pragma version^0.8.2 (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/proxy/ERC1967/ERC1967UpgradeUpgradeable.sol#4) allows old versions
Pragma version^0.8.0 (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/proxy/beacon/IBeaconUpgradeable.sol#4) allows old versions
Pragma version^0.8.2 (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/proxy/utils/Initializable.sol#4) allows old versions
Pragma version^0.8.0 (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/proxy/utils/UUPSUpgradeable.sol#4) allows old versions
Pragma version^0.8.0 (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/security/ReentrancyGuardUpgradeable.sol#4) allows old versions
Pragma version^0.8.0 (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/token/ERC20/ERC20Upgradeable.sol#4) allows old versions
Pragma version^0.8.0 (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/token/ERC20/IERC20Upgradeable.sol#4) allows old versions
Pragma version^0.8.0 (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/token/ERC20/extensions/ERC4626Upgradeable.sol#4) allows old versions
Pragma version^0.8.0 (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/token/ERC20/extensions/IERC20MetadataUpgradeable.sol#4) allows old versions
Pragma version^0.8.0 (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/token/ERC20/extensions/draft-IERC20PermitUpgradeable.sol#4) allows old versions
Pragma version^0.8.0 (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/token/ERC20/utils/SafeERC20Upgradeable.sol#4) allows old versions
Pragma version^0.8.1 (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/AddressUpgradeable.sol#4) allows old versions
Pragma version^0.8.0 (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/ContextUpgradeable.sol#4) allows old versions
Pragma version^0.8.0 (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/StorageSlotUpgradeable.sol#4) allows old versions
Pragma version^0.8.0 (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/StringsUpgradeable.sol#4) allows old versions
Pragma version^0.8.0 (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/introspection/ERC165Upgradeable.sol#4) allows old versions
Pragma version^0.8.0 (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/introspection/IERC165Upgradeable.sol#4) allows old versions
Pragma version^0.8.0 (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/math/MathUpgradeable.sol#4) allows old versions
Pragma version^0.8.13 (jobs/archimedes/brownie/contracts/AccessController.sol#2) necessitates a version too recent to be trusted. Consider deploying with 0.6.12/0.7.6/0.8.7
Pragma version^0.8.13 (jobs/archimedes/brownie/contracts/ParameterStore.sol#2) necessitates a version too recent to be trusted. Consider deploying with 0.6.12/0.7.6/0.8.7
Pragma version^0.8.13 (jobs/archimedes/brownie/contracts/VaultOUSD.sol#2) 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 (jobs/archimedes/brownie/contracts/interfaces/IOUSD.sol#4) allows old versions
solc-0.8.13 is not recommended for deployment
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#incorrect-versions-of-solidity

Low level call in ERC1967UpgradeUpgradeable._functionDelegateCall(address,bytes) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/proxy/ERC1967/ERC1967UpgradeUpgradeable.sol#198-204):
        - (success,returndata) = target.delegatecall(data) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/proxy/ERC1967/ERC1967UpgradeUpgradeable.sol#202)
Low level call in AddressUpgradeable.sendValue(address,uint256) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/AddressUpgradeable.sol#60-65):
        - (success) = recipient.call{value: amount}() (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/AddressUpgradeable.sol#63)
Low level call in AddressUpgradeable.functionCallWithValue(address,bytes,uint256,string) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/AddressUpgradeable.sol#128-139):
        - (success,returndata) = target.call{value: value}(data) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/AddressUpgradeable.sol#137)
Low level call in AddressUpgradeable.functionStaticCall(address,bytes,string) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/AddressUpgradeable.sol#157-166):
        - (success,returndata) = target.staticcall(data) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/AddressUpgradeable.sol#164)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#low-level-calls
```

AUTOMATED TESTING

```
Low level call in ERC1967UpgradeUpgradeable._functionDelegateCall(address,bytes) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/proxy/ERC1967/ERC1967UpgradeUpgradeable.sol#198-204):
        - (success,returndata) = target.delegatecall(data) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/proxy/ERC1967/ERC1967UpgradeUpgradeable.sol#202)
Low level call in AddressUpgradeable.sendValue(address,uint256) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/AddressUpgradeable.sol#60-65):
        - (success) = recipient.call{value: amount}() (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/AddressUpgradeable.sol#63)
Low level call in AddressUpgradeable.functionCallWithValue(address,bytes,uint256,string) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/AddressUpgradeable.sol#128-139):
        - (success,returndata) = target.call{value: value}(data) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/AddressUpgradeable.sol#137)
Low level call in AddressUpgradeable.functionStaticCall(address,bytes,string) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/AddressUpgradeable.sol#157-166):
        - (success,returndata) = target.staticcall(data) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/AddressUpgradeable.sol#164)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#low-level-calls

Function AccessControlUpgradeable.__AccessControl_init() (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/access/AccessControlUpgradeable.sol#51-52) is not in mixedCase
Function AccessControlUpgradeable.__AccessControl_init_unchained() (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/access/AccessControlUpgradeable.sol#54-55) is not in mixedCase
Variable AccessControlUpgradeable.__gap (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/access/AccessControlUpgradeable.sol#259) is not in mixedCase
Function ERC1967UpgradeUpgradeable._ERC1967Upgrade_init() (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/proxy/ERC1967/ERC1967UpgradeUpgradeable.sol#21-22) is not in mixedCase
Function ERC1967UpgradeUpgradeable._ERC1967Upgrade_init_unchained() (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/proxy/ERC1967/ERC1967UpgradeUpgradeable.sol#24-25) is not in mixedCase
Variable ERC1967UpgradeUpgradeable.__gap (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/proxy/ERC1967/ERC1967UpgradeUpgradeable.sol#211) is not in mixedCase
Function UUPSUpgradeable.__UUPSUpgradeable_init() (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/proxy/utils/UUPSUpgradeable.sol#23-24) is not in mixedCase
Function UUPSUpgradeable.__UUPSUpgradeable_init_unchained() (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/proxy/utils/UUPSUpgradeable.sol#26-27) is not in mixedCase
Variable UUPSUpgradeable.__gap (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/proxy/utils/UUPSUpgradeable.sol#107) is not in mixedCase
Variable UUPSUpgradeable.__self (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/proxy/utils/UUPSUpgradeable.sol#29) is not in mixedCase
Function ReentrancyGuardUpgradeable.__ReentrancyGuard_init() (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/security/ReentrancyGuardUpgradeable.sol#40-42) is not in mixedCase
Function ReentrancyGuardUpgradeable.__ReentrancyGuard_init_unchained() (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/security/ReentrancyGuardUpgradeable.sol#44-46) is not in mixedCase
Variable ReentrancyGuardUpgradeable.__gap (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/security/ReentrancyGuardUpgradeable.sol#74) is not in mixedCase
Function ERC20Upgradeable.__ERC20_init(string,string) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/token/ERC20/ERC20Upgradeable.sol#55-57) is not in mixedCase
Function ERC20Upgradeable.__ERC20_init_unchained(string,string) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/token/ERC20/ERC20Upgradeable.sol#59-62) is not in mixedCase
Variable ERC20Upgradeable.__gap (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/token/ERC20/ERC20Upgradeable.sol#394) is not in mixedCase
Function ERC4626Upgradeable.__ERC4626_init(IERC20MetadataUpgradeable) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/token/ERC20/extensions/ERC4626Upgradeable.sol#35-37) is not in mixedCase
Function ERC4626Upgradeable.__ERC4626_init_unchained(IERC20MetadataUpgradeable) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/token/ERC20/extensions/ERC4626Upgradeable.sol#39-41) is not in mixedCase
Variable ERC4626Upgradeable.__gap (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/token/ERC20/extensions/ERC4626Upgradeable.sol#233) is not in mixedCase
Function IERC20PermitUpgradeable.DOMAIN_SEPARATOR() (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/token/ERC20/extensions/draft-IERC20PermitUpgradeable.sol#59) is not in mixedCase
Function ContextUpgradeable.__Context_init() (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/ContextUpgradeable.sol#18-19) is not in mixedCase
Function ContextUpgradeable.__Context_init_unchained() (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/ContextUpgradeable.sol#21-22) is not in mixedCase
Variable ContextUpgradeable.__gap (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/ContextUpgradeable.sol#36) is not in mixedCase
Function ERC165Upgradeable.__ERC165_init() (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/introspection/ERC165Upgradeable.sol#24-25) is not in mixedCase
Function ERC165Upgradeable.__ERC165_init_unchained() (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/introspection/ERC165Upgradeable.sol#27-28) is not in mixedCase
Variable ERC165Upgradeable.__gap (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/utils/introspection/ERC165Upgradeable.sol#41) is not in mixedCase
Variable ParameterStore._maxNumberOfCycles (jobs/archimedes/brownie/contracts/ParameterStore.sol#14) is not in mixedCase
Variable ParameterStore._originationFeeRate (jobs/archimedes/brownie/contracts/ParameterStore.sol#15) is not in mixedCase
Variable ParameterStore._globalCollateralRate (jobs/archimedes/brownie/contracts/ParameterStore.sol#16) is not in mixedCase
Variable ParameterStore._rebaseFeeRate (jobs/archimedes/brownie/contracts/ParameterStore.sol#17) is not in mixedCase
Variable ParameterStore._treasuryAddress (jobs/archimedes/brownie/contracts/ParameterStore.sol#18) is not in mixedCase
Variable ParameterStore._curveGuardPercentage (jobs/archimedes/brownie/contracts/ParameterStore.sol#19) is not in mixedCase
Variable ParameterStore._slippage (jobs/archimedes/brownie/contracts/ParameterStore.sol#20) is not in mixedCase
Variable ParameterStore._archToLevRatio (jobs/archimedes/brownie/contracts/ParameterStore.sol#22) is not in mixedCase
Variable ParameterStore._curveMaxExchangeGuard (jobs/archimedes/brownie/contracts/ParameterStore.sol#24) is not in mixedCase
Variable ParameterStore._minPositionCollateral (jobs/archimedes/brownie/contracts/ParameterStore.sol#25) is not in mixedCase
Variable ParameterStore._positionTimeToLiveInDays (jobs/archimedes/brownie/contracts/ParameterStore.sol#26) is not in mixedCase
Parameter VaultOUSD.setDependencies(address,address)._addressParamStore (jobs/archimedes/brownie/contracts/VaultOUSD.sol#31) is not in mixedCase
Parameter VaultOUSD.setDependencies(address,address)._addressOUSD (jobs/archimedes/brownie/contracts/VaultOUSD.sol#31) is not in mixedCase
Variable VaultOUSD._paramStore (jobs/archimedes/brownie/contracts/VaultOUSD.sol#22) is not in mixedCase
Variable VaultOUSD._ousd (jobs/archimedes/brownie/contracts/VaultOUSD.sol#23) is not in mixedCase
Variable VaultOUSD._assetsHandledByArchimedes (jobs/archimedes/brownie/contracts/VaultOUSD.sol#25) is not in mixedCase
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#conformance-to-solidity-naming-conventions

ParameterStore.changeArchToLevRatio(uint256) (jobs/archimedes/brownie/contracts/ParameterStore.sol#99-103) uses literals with too many digits:
        - require(bool,string)(newArchToLevRatio < 500000000000000000000000 && newArchToLevRatio >= 10000000000000000,new ArchToLevRatio out of range) (jobs/archimedes/brownie/contracts/ParameterStore.sol#100)
ParameterStore.changeMinPositionCollateral(uint256) (jobs/archimedes/brownie/contracts/ParameterStore.sol#111-115) uses literals with too many digits:
        - require(bool,string)(newMinPositionCollateral < (1000000000000000000000000) && newMinPositionCollateral > (10000000000000000000),New min collateral out of range) (jobs/archimedes/brownie/contracts/ParameterStore.sol#112)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#too-many-digits

UUPSUpgradeable.__gap (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/proxy/utils/UUPSUpgradeable.sol#107) is never used in ParameterStore (jobs/archimedes/brownie/contracts/ParameterStore.sol#10
-243)
UUPSUpgradeable.__gap (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/proxy/utils/UUPSUpgradeable.sol#107) is never used in VaultOUSD (jobs/archimedes/brownie/contracts/VaultOUSD.sol#19-146)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#unused-state-variable

grantRole(bytes32,address) should be declared external:
        - AccessControlUpgradeable.grantRole(bytes32,address) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/access/AccessControlUpgradeable.sol#150-152)
revokeRole(bytes32,address) should be declared external:
        - AccessControlUpgradeable.revokeRole(bytes32,address) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/access/AccessControlUpgradeable.sol#165-167)
renounceRole(bytes32,address) should be declared external:
        - AccessControlUpgradeable.renounceRole(bytes32,address) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/access/AccessControlUpgradeable.sol#185-189)
        - AccessController.renounceRole(bytes32,address) (jobs/archimedes/brownie/contracts/AccessController.sol#62-69)
name() should be declared external:
        - ERC20Upgradeable.name() (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/token/ERC20/ERC20Upgradeable.sol#67-69)
symbol() should be declared external:
        - ERC20Upgradeable.symbol() (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/token/ERC20/ERC20Upgradeable.sol#75-77)
transfer(address,uint256) should be declared external:
        - ERC20Upgradeable.transfer(address,uint256) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/token/ERC20/ERC20Upgradeable.sol#118-122)
approve(address,uint256) should be declared external:
        - ERC20Upgradeable.approve(address,uint256) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/token/ERC20/ERC20Upgradeable.sol#141-145)
transferFrom(address,address,uint256) should be declared external:
        - ERC20Upgradeable.transferFrom(address,address,uint256) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/token/ERC20/ERC20Upgradeable.sol#163-172)
increaseAllowance(address,uint256) should be declared external:
        - ERC20Upgradeable.increaseAllowance(address,uint256) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/token/ERC20/ERC20Upgradeable.sol#186-190)
decreaseAllowance(address,uint256) should be declared external:
        - ERC20Upgradeable.decreaseAllowance(address,uint256) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/token/ERC20/ERC20Upgradeable.sol#206-215)
asset() should be declared external:
        - ERC4626Upgradeable.asset() (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/token/ERC20/extensions/ERC4626Upgradeable.sol#44-46)
convertToShares(uint256) should be declared external:
        - ERC4626Upgradeable.convertToShares(uint256) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/token/ERC20/extensions/ERC4626Upgradeable.sol#54-56)
convertToAssets(uint256) should be declared external:
        - ERC4626Upgradeable.convertToAssets(uint256) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/token/ERC20/extensions/ERC4626Upgradeable.sol#59-61)
mint(uint256,address) should be declared external:
        - ERC4626Upgradeable.mint(uint256,address) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/token/ERC20/extensions/ERC4626Upgradeable.sol#114-121)
withdraw(uint256,address,address) should be declared external:
        - ERC4626Upgradeable.withdraw(uint256,address,address) (.brownie/packages/OpenZeppelin/openzeppelin-contracts@4.7.0/contracts-upgradeable/token/ERC20/extensions/ERC4626Upgradeable.sol#124-135)
setAdmin(address) should be declared external:
        - AccessController.setAdmin(address) (jobs/archimedes/brownie/contracts/AccessController.sol#41-47)
getAddressExecutive() should be declared external:
        - AccessController.getAddressExecutive() (jobs/archimedes/brownie/contracts/AccessController.sol#95-97)
initialize() should be declared external:
        - ParameterStore.initialize() (jobs/archimedes/brownie/contracts/ParameterStore.sol#31-52)
initialize(IERC20MetadataUpgradeable,string,string) should be declared external:
        - VaultOUSD.initialize(IERC20MetadataUpgradeable,string,string) (jobs/archimedes/brownie/contracts/VaultOUSD.sol#88-106)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#public-function-that-could-be-declared-external
```

Issues found by slither are either reported above or false positives.

THANK YOU FOR CHOOSING

// HALBORN