ArthController Smart Contract Preliminary Audit Report

Project Synopsis

Project Name	**MahaDao Audit**
Platform	Ethereum, Solidity
Github Repo	https://github.com/MahaDAO/arthcoin-v2/blob/develop/contracts/Arth/ArthController.sol
Deployed Contract	Not Deployed
Total Duration	15 Days
Timeline of Audit	**15th April 2021 to 02nd April 2021**

Contract Details

Total Contract(s)	1
Name of Contract(s)	ArthController
Language	Solidity
Commit Hash	`d4d445c8e8fe9708ef04a94c09be2e961aa48105`

Contract Vulnerabilities Synopsis

Issues	Open Issues	Closed Issues
Critical Severity	1	0
Medium Severity	3	0
Low Severity	5	0
Informational	2	0
Total Found	11	0

Detailed Results

The contract has gone through several stages of the audit procedure that includes structural analysis, automated testing, manual code review, etc.

All the issues have been explained and discussed in detail below. Along with the explanation of the issue found during the audit, the recommended way to overcome the issue or improve the code quality has also been mentioned.

A. Contract Name: ArthPool

High Severity Issues

A.1 Contract uses Uninitialized State Variables

Line - 32, 43, 110, 395

Description:

The ArthController contract accesses some imperative State Variables that have never been initialized throughout the contract.

Mentioned below are those State Variables and the specific parts in the contract where they are accessed:

- **ARTH (***Line 32***)** is being accessed in the ***getARTHInfo*** function at Line 395 but was never initialized.

```
394             getARTHXPrice(), // ARTHX price.
395             ARTH.totalSupply(), // ARTH total supply
396             globalCollateralRatio, // Global collate
```

- ***controllerAddress (****Line 43****)*** is being used in the ***onlyByOwnerOrGovernance*** modifier at Line 110 but was never initialized.

```
106     modifier onlyByOwnerOrGovernance() {
107         require(
108             msg.sender == ownerAddress ||
109                 msg.sender == timelockAddress ||
110                 msg.sender == controllerAddress,
111             'ARTHController: FORBIDDEN'
112         );
113         _;
114     }
```

Recommendation:

The above-mentioned state variables must be initialized within the constructor before being used in any particular functions, to avoid any unwanted scenario during function executions.

Medium Severity Issues

A.2 Multiplication is being performed on the result of Division

Line no - 414-417

Explanation:

During the automated testing of the ArthController.sol contract, it was found that one of the functions in the contract is performing multiplication on the result of a Division.

Integer Divisions in Solidity might truncate. Moreover, performing division before multiplication might lead to loss of precision.

The following functions involve division before multiplication in the mentioned lines:

- ***_getOraclePrice***

```
414         uint256 eth2GMUPrice =
415             uint256(_ETHGMUPricer.getLatestPrice()).mul(_PRICE_PRECISION).div(
416                 uint256(10)**_ethGMUPricerDecimals
417             );
```

Recommendation:

Solidity doesn't encourage arithmetic operations that involve division before multiplication. Therefore the above-mentioned function should be checked once and redesigned if it does not lead to expected results.

A.3 Functions includes Costly Loops

Line no - 214, 365

Description:

The **ArthController contract** has ***for loops*** in some functions that includes state variables like **.length** of a non-memory array in the condition of the for loops.

```
213     // 'Delete' from the array by setting the address to 0x0
214     for (uint256 i = 0; i < arthPoolsArray.length; i++) {
215         if (arthPoolsArray[i] == poolAddress) {
216             arthPoolsArray[i] = address(0); // This will lea
217             break;
218         }
219     }
```

As a result, these state variables consume a lot more extra gas for every iteration of the loop.
The following functions in the contract includes such loops at the specified Line numbers:

- ***removePool*** at Line 214
- ***getGlobalCollateralValue*** at Line 365

Recommendation:
A better and effective approach would be to use a local variable instead of a state variable like ***.length*** in a loop.

For instance,
local_variable = arthPoolsArray.length;
for (uint256 i = 0; i < local_variable; i++) {
if (arthPoolsArray[i] == poolAddress) {
arthPoolsArray[i] = address(0); // This will leave a null in the array and keep the indices the same.
break;
}

A.4 Modifiers created but never used

Line no - 93, 116

Description:
The ArthController contract includes some modifiers, at the above-mentioned lines, that have been created but never used throughout the contract.

```
116 ▾    modifier onlyByOwnerGovernanceOrPool() {
117 ▾        require(
118 ▾            msg.sender == ownerAddress ||
119                  msg.sender == timelockAddress ||
120                  arthPools[msg.sender] == true,
121              'ARTHController: FORBIDDEN'
122          );
123          _;
124      }
```

While this consumes additional space in the contract, it also adversely affects the gas optimization as well as the readability of the smart contract code.

Recommendation:
Adequate use of all State Variable, modifiers, mappings etc, must be ensured in the contract. If a particular modifier holds no significance it should be removed from the contract.

Low Severity Issues

A.5 External Visibility should be preferred

Explanation:
Those functions that are never called throughout the contract should be marked as ***external*** visibility instead of ***public*** visibility.
This will effectively result in Gas Optimization as well.

Therefore, the following function must be marked as **external** within the contract:

- ***setARTHXAddress***
- ***setPriceTarget***
- ***setRefreshCooldown***
- ***setETHGMUOracle***
- ***setARTHXETHOracle***
- ***setARTHETHOracle***
- ***toggleCollateralRatio***
- ***setMintingFee***
- ***setArthStep***
- ***setRedemptionFee***
- ***setOwner***
- ***setPriceBand***
- ***setTimelock***
- ***getGlobalCollateralRatio***
- ***getARTHInfo***

Recommendation:
If the public visibility of these functions is not intended, the visibility keyword must be modified to external.

A.6 Absence of Error messages in Require Statements

Line no - 89
Description:
In the ArthController contract, the ***modifier onlyCollateralRatioPauser*** includes a **require** statement that does not include an error message.

```
88      modifier onlyCollateralRatioPauser() {
89          require(hasRole(COLLATERAL_RATIO_PAUSER, msg.sender));
90          _;
91      }
```

While this makes it troublesome to detect the reason behind a particular function revert, it also reduces the readability of the code.

Recommendation:
Error Messages must be included in every require statement in the contract

A.7 Comparison to boolean Constant

Line no- 94, 120, 154,192, 206,

Description:
Boolean constants can directly be used in conditional statements or require statements.
Therefore, it's not considered a better practise to explicitly use **TRUE or FALSE** in the **require** statements.

```
186     function addPool(address poolAddress)
187         external
188         override
189         onlyByOwnerOrGovernance
190     {
191         require(
192             arthPools[poolAddress] == false,
193             'ARTHController: address present'
194         );
```

Recommendation:
The equality to boolean constants must be removed from the above-mentioned line.

A.8 No Event emission for crucial State Variable modification

Line no - 242, 291, 299, 323, 307

Description:

Functions that modify an imperative arithmetic state variable contract should emit an event after the modification.
However, during the automated testing it was found that the following functions modify some crucial arithmetic parameters like ***priceTarget, mintingFee, redemptionFee, priceBand etc,*** but doesn't emit any event afterwards:

- ***setPriceBand***

- ***setRedemptionFee***
- ***setArthStep***
- ***mintingFee***
- ***setPriceTarget***

Since there is no event emitted on updating these variables, it might be difficult to track it off-chain.

Recommendation:

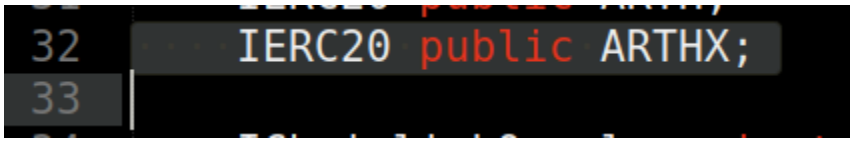
An event should be fired after changing crucial arithmetic state variables.

A.9 State Variable never used in the Contract

Line no - 32

Description:

ARTHX state variable has been initialized in the contract at Line 32 but never used throughout the contract.



Recommendation:
Effective use of all State Variables must be ensured in the contract. Unused variables should be removed from the contract.

Informational

A.10 Coding Style Issues in the Contract

Explanation:
Code readability of a Smart Contract is largely influenced by the Coding Style issues and in some specific scenarios may lead to bugs in the future.

During the automated testing, it was found that the ArthController contract had quite a few code style issues.

```
Parameter ArthController.setOwner(address)._ownerAddress (contracts/Arth/flat_ArthControl.sol#1206) is not in mixedCase
Parameter ArthController.setPriceBand(uint256)._priceBand (contracts/Arth/flat_ArthControl.sol#1214) is not in mixedCase
Variable ArthController.ARTH (contracts/Arth/flat_ArthControl.sol#922) is not in mixedCase
Variable ArthController.ARTHX (contracts/Arth/flat_ArthControl.sol#923) is not in mixedCase
Variable ArthController._ETHGMUPricer (contracts/Arth/flat_ArthControl.sol#925) is not in mixedCase
Variable ArthController._ARTHETHOracle (contracts/Arth/flat_ArthControl.sol#926) is not in mixedCase
Variable ArthController._ARTHXETHOracle (contracts/Arth/flat_ArthControl.sol#927) is not in mixedCase
Variable ArthController.DEFAULT_ADMIN_ADDRESS (contracts/Arth/flat_ArthControl.sol#938) is not in mixedCase
```

Recommendation:

Therefore, it is highly recommended to fix the issues like naming convention, indentation, and code layout issues in a smart contract.

A.11 NatSpec Annotations must be included

Description:

The smart contracts do not include the NatSpec annotations adequately.

Recommendation:

Cover by NatSpec all Contract methods.

AutoMated Test Results

```
ArthController._getOraclePrice(ArthController.PriceChoice) (contracts/Arth/flat_ArthControl.sol#1300-1326) performs a multiplication on the result of a division
       -eth2GMUPrice = uint256(_ETHGMUPricer.getLatestPrice()).mul(_PRICE_PRECISION).div(uint256(10) ** _ethGMUPricerDecimals) (contracts/Arth/flat_ArthControl
)
       -eth2GMUPrice.mul(_PRICE_PRECISION).div(priceVsETH) (contracts/Arth/flat_ArthControl.sol#1325)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#divide-before-multiply
```

```
ArthController.setOwner(address) (contracts/Arth/flat_ArthControl.sol#1206-1212) should emit an event for:
        - ownerAddress = _ownerAddress (contracts/Arth/flat_ArthControl.sol#1211)
        - ownerAddress = _ownerAddress (contracts/Arth/flat_ArthControl.sol#1211)
ArthController.setTimelock(address) (contracts/Arth/flat_ArthControl.sol#1222-1228) should emit an event for:
        - timelockAddress = newTimelock (contracts/Arth/flat_ArthControl.sol#1227)
        - timelockAddress = newTimelock (contracts/Arth/flat_ArthControl.sol#1227)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#missing-events-access-control
INFO:Detectors:
ArthController.setPriceTarget(uint256) (contracts/Arth/flat_ArthControl.sol#1133-1139) should emit an event for:
        - priceTarget = newPriceTarget (contracts/Arth/flat_ArthControl.sol#1138)
ArthController.setRefreshCooldown(uint256) (contracts/Arth/flat_ArthControl.sol#1141-1147) should emit an event for:
        - refreshCooldown = newCooldown (contracts/Arth/flat_ArthControl.sol#1146)
ArthController.setETHGMUOracle(address) (contracts/Arth/flat_ArthControl.sol#1149-1157) should emit an event for:
        - _ethGMUPricerDecimals = _ETHGMUPricer.getDecimals() (contracts/Arth/flat_ArthControl.sol#1156)
ArthController.setMintingFee(uint256) (contracts/Arth/flat_ArthControl.sol#1182-1188) should emit an event for:
        - mintingFee = fee (contracts/Arth/flat_ArthControl.sol#1187)
```

```
ArthController.constructor(address,address)._creatorAddress (contracts/Arth/flat_ArthControl.sol#1021) lacks a zero-check on :
              - creatorAddress = _creatorAddress (contracts/Arth/flat_ArthControl.sol#1022)
              - ownerAddress = _creatorAddress (contracts/Arth/flat_ArthControl.sol#1025)
ArthController.constructor(address,address)._timelockAddress (contracts/Arth/flat_ArthControl.sol#1021) lacks a zero-check on :
              - timelockAddress = _timelockAddress (contracts/Arth/flat_ArthControl.sol#1023)
ArthController.setARTHXAddress(address)._arthxAddress (contracts/Arth/flat_ArthControl.sol#1125) lacks a zero-check on :
              - arthxAddress = _arthxAddress (contracts/Arth/flat_ArthControl.sol#1130)
ArthController.setETHGMUOracle(address)._ethGMUConsumerAddress (contracts/Arth/flat_ArthControl.sol#1149) lacks a zero-check on :
              - ethGMUConsumerAddress = _ethGMUConsumerAddress (contracts/Arth/flat_ArthControl.sol#1154)
ArthController.setARTHXETHOracle(address,address)._arthxOracleAddress (contracts/Arth/flat_ArthControl.sol#1160) lacks a zero-check on :
              - arthxETHOracleAddress = _arthxOracleAddress (contracts/Arth/flat_ArthControl.sol#1163)
ArthController.setARTHXETHOracle(address,address)._wethAddress (contracts/Arth/flat_ArthControl.sol#1161) lacks a zero-check on :
              - wethAddress = _wethAddress (contracts/Arth/flat_ArthControl.sol#1165)
ArthController.setARTHETHOracle(address,address)._arthOracleAddress (contracts/Arth/flat_ArthControl.sol#1168) lacks a zero-check on :
              - arthETHOracleAddress = _arthOracleAddress (contracts/Arth/flat_ArthControl.sol#1173)
ArthController.setARTHETHOracle(address,address)._wethAddress (contracts/Arth/flat_ArthControl.sol#1168) lacks a zero-check on :
              - wethAddress = _wethAddress (contracts/Arth/flat_ArthControl.sol#1175)
```

```
ArthController.constructor(address,address) (contracts/Arth/flat_ArthControl.sol#1021-1037) uses literals with too many digits:
        - priceTarget = 1000000 (contracts/Arth/flat_ArthControl.sol#1034)
ArthController.constructor(address,address) (contracts/Arth/flat_ArthControl.sol#1021-1037) uses literals with too many digits:
        - globalCollateralRatio = 1000000 (contracts/Arth/flat_ArthControl.sol#1036)
ArthController.refreshCollateralRatio() (contracts/Arth/flat_ArthControl.sol#1043-1073) uses literals with too many digits:
        - globalCollateralRatio.add(arthStep) >= 1000000 (contracts/Arth/flat_ArthControl.sol#1065)
ArthController.refreshCollateralRatio() (contracts/Arth/flat_ArthControl.sol#1043-1073) uses literals with too many digits:
        - globalCollateralRatio = 1000000 (contracts/Arth/flat_ArthControl.sol#1066)
```

```
toggleCollateralRatio() should be declared external:
        - ArthController.toggleCollateralRatio() (contracts/Arth/flat_ArthControl.sol#1178-1180)
setMintingFee(uint256) should be declared external:
        - ArthController.setMintingFee(uint256) (contracts/Arth/flat_ArthControl.sol#1182-1188)
setArthStep(uint256) should be declared external:
        - ArthController.setArthStep(uint256) (contracts/Arth/flat_ArthControl.sol#1190-1196)
setRedemptionFee(uint256) should be declared external:
        - ArthController.setRedemptionFee(uint256) (contracts/Arth/flat_ArthControl.sol#1198-1204)
setOwner(address) should be declared external:
        - ArthController.setOwner(address) (contracts/Arth/flat_ArthControl.sol#1206-1212)
setPriceBand(uint256) should be declared external:
        - ArthController.setPriceBand(uint256) (contracts/Arth/flat_ArthControl.sol#1214-1220)
setTimelock(address) should be declared external:
        - ArthController.setTimelock(address) (contracts/Arth/flat_ArthControl.sol#1222-1228)
getGlobalCollateralRatio() should be declared external:
        - ArthController.getGlobalCollateralRatio() (contracts/Arth/flat_ArthControl.sol#1249-1251)
getARTHInfo() should be declared external:
```