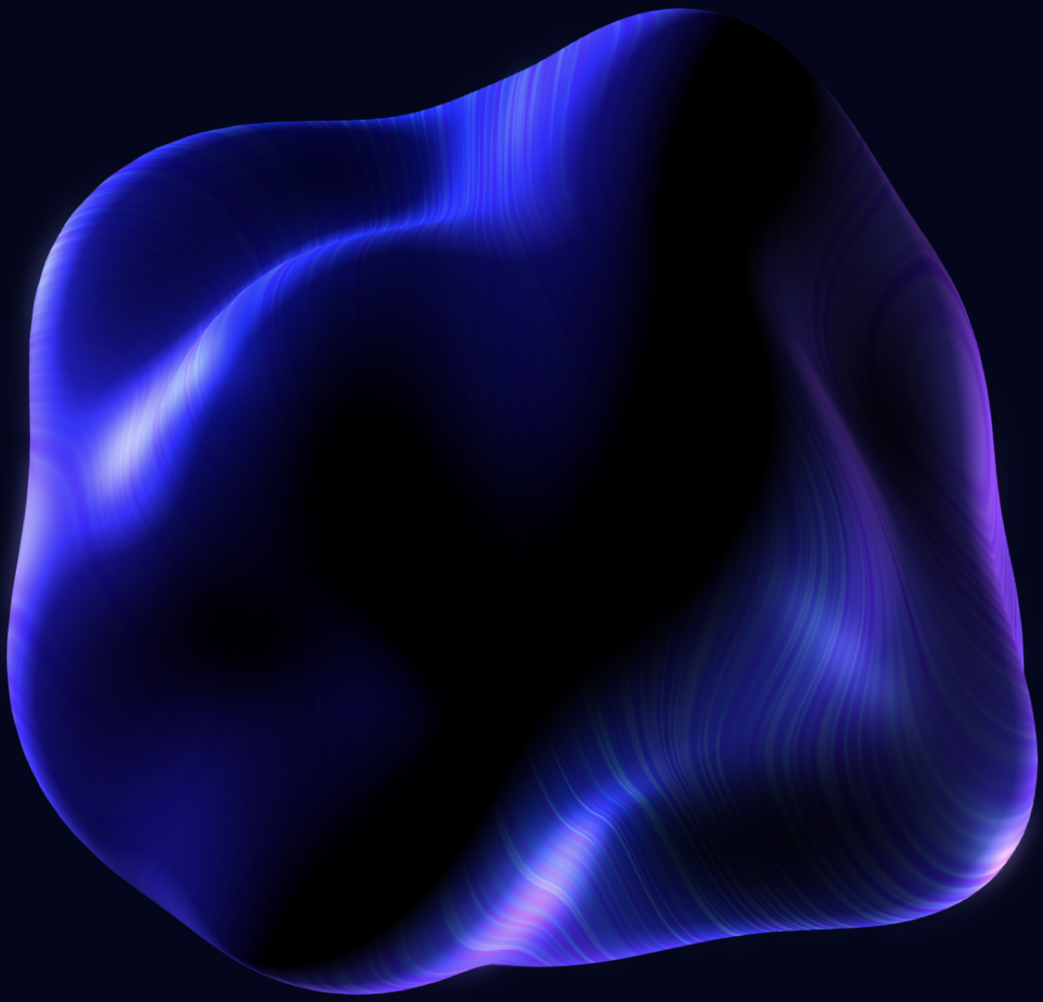
MUNDUS
SECURITY

Deployment Check



mundus.dev

@mundus_security

Mundus.dev

MUNDUS
SECURITY

MAHADAO ARTH deployment check

This document may contain confidential information about IT systems and the intellectual property of the Customer as well as information about potential vulnerabilities and methods of their exploitation.
The report containing confidential information can be used internally by the Customer, or it can be disclosed publicly after all vulnerabilities are fixed — upon a decision of the Customer.

Reference information

Name	MAHADAO ARTH
Language	Solidity
Chain	Ethereum mainnet
Website	https://mahadao.com/
Documentation	https://docs.arth.loans/
Reference repositories	https://github.com/MahaDAO/arth-core https://github.com/MahaDAO/arth-strategies https://github.com/MahaDAO/gmu-oracle-contracts https://github.com/MahaDAO/chainlink-keepers https://github.com/MahaDAO/token https://github.com/MahaDAO/flashloans-arth

Deployment check summary for MahaDAO ARTH

Codebase inconsistency issues

Type	Severity
In deployed contracts	**HIGH**
Smart contracts vs Git	**MEDIUM**
In dependencies of codebase	**MEDIUM**

Storage analysis

Type	# Issues
Found Total	20
Found Critical	7
Left as acknowledged after re-checks	0
Left as Critical after re-checks	0

Deployment check is expert review of the storage and codebase consistency of a deployed project*

Deployed Smart contracts and/or Git repos & Contract storage

Deployment check is especially important for projects with active development and regular updates to ensure that after all incremental updates, the set of contracts and their settings are consistent. It includes two rechecks.

Deployment check protects against

- Errors in CI/CD, especially in large projects
- Potential attacks from people with access to the codebase
- Large number of human errors during updates of the project in the network
- Incorrect cross-references between smart contracts
- Relation to old versions of contracts
- Forgotten role members
- Uncorrected ownership

*Note that security audit is not the part of this review

Deployment check timeline

date	stage
14.01.2023	scope of work settled
15.01.2023 23.01.2023	initial check
24.01.2023	presentation to the client
15.02.2023 27.02.2023	first re-check
28.02.2023	presentation to the client
03.04.2023 09.04.2023	second re-check
10.04.2023	presentation to the client

Scope of work

contract	address
ActivePool	0xa443129308556ab06e69a98e1c39c81080e01530
ARTHFlashMinter	0xc4bbefdc3066b919cd1a6b5901241e11282e625d
ARTHValuecoin	0x8cc0f052fff7ead7f2edcccac895502e884a8a71
BorrowerOperations	0x4c50063f8238dea92c738f23221733a9a6c6888b
CollSurplusPool	0xbb719b2d7207e8b8b13ca4dc9c8b6201d79cf7e5
CommunityIssuance	0x61274cd1f801b097be7e5197b158999307893d2e
DefaultPool	0x47f747fd93eef25cc1e0b6d7a239289c7cfec212
ETHGMUOracle	0xc31adc9ae073a1f6a9ce5c41b32c18790ea667fe
ETHTroveStrategy *Proxy*	0xf3f261f54d83978061325 98dc2b6b5c00d6eb3ea 0xa9735e594624339f8fbc8a99c57c13c7b4e8bcac
GMUOracle	0x066a917fa2e1739ccfc306dc73ff78eeca8b6f29
Governance	0x91eb23b66beb3467998402ba50aa1c1a98811eb1
SortedTroves	0xd60d7a2a8344d4f635bf9ea9f8cd015a614c3659
StabilityPool	0x910f16455e5eb4605fe639e2846579c228eed3b5
StabilityPoolKeeper	0x5e98d3f8b5074b6389477fd88856f5209748caa7
TroveManager	0x8b1da95724b1e376ae49fdb67afe33fe41093af5
USDCCurveStrategy *Proxy*	0x9ff6629d08fddaec63b0d855b9c29acdf4dc14e4 0x5480e8beedb3eba5747a4a3aef0850a3759df9b4

Table of contents

Findings summary

Storage findings

contract	storage issues initial check	storage issues first re-check	storage issues second re-check
ActivePool	none	--	--
ARTHFlashMinter	--	found	none
ARTHValuecoin	found **1 critical**	none	--
BorrowerOperations	found	none	--
CollSurplusPool	none	--	--
CommunityIssuance	none	--	--
DefaultPool	none	--	--
ETHGMUOracle	none	--	--
ETHTroveStrategy	found **1 critical**	found	none
GMUOracle	found **1 critical**	none	--
Governance	none	--	--
SortedTroves	none	--	--
StabilityPool	found	none	--
StabilityPoolKeeper	none	--	--
TroveManager	found	none	--
USDCCurveStrategy	found **4 criticals**	found **4 criticals**	none

Source code findings

name	severity
Inconsistency between the same project files across contracts	High
Searching for the original commit in the client's repository	Medium
Analyzing the dependencies of the contracts	Medium

Deployment check: source code

This analysis aims to identify any differences or inconsistencies in the source code of the smart contracts. We perform the analysis in three steps:

1. Analyzing for inconsistency between source code files across deployed smart contracts (excluding well-known dependencies such as OpenZeppelin or Uniswap).
2. Looking for the original commit in the client's repository, which represents all source code of deployed smart contracts in the case of providing the client's git
3. Analyzing the dependencies of the contracts

See number of files statistics in section A1.

Inconsistency between the same project files across contracts (excluding dependencies)

The goal is to check for any differences and inconsistencies in the source code of the same parts of the contracts. We compare each pair of smart contracts in the scope of work (SoW). Files with the same name and relative path included (imported) in both contracts should have the same content.

#	contract-1	contract-2	path	comment
1	ETHGMUOracle	StabilityPoolKeeper	contracts/interfaces/IEpoch.sol	see diffs. 1-2 below
2	GMUOracle	StabilityPoolKeeper	contracts/interfaces/IEpoch.sol	see diffs. 1-2 below
3	ETHGMUOracle	USDCCurveStrategy	contracts/interfaces/IPriceFeed.sol	see diffs. 3-6 below
4	ETHGMUOracle	ETHTroveStrategy	contracts/interfaces/IPriceFeed.sol	see diffs. 3-6 below
5	GMUOracle	USDCCurveStrategy	contracts/interfaces/IPriceFeed.sol	see diffs. 3-6 below
6	GMUOracle	ETHTroveStrategy	contracts/interfaces/IPriceFeed.sol	see diffs. 3-6 below
7	GMUOracle	StabilityPoolKeeper	contracts/utils/Epoch.sol	see diff. 7 below
8	USDCCurveStrategy	ETHTroveStrategy	contracts/staking/StakingRewardsChild.sol	see diff. 8 below
9	GMUOracle	StabilityPoolKeeper	contracts/interfaces/KeeperCompatibleInterface.sol	cosmetics

Diffs. 1 -- 2: event EpochTriggered is added to IEpoch.sol in **ETHGMUOracle** and **GMUOracle** contracts, as opposed to **StabilityPoolKeeper**. Events do not directly affect the on-chain part of the protocol logic.

Diffs. 3 -- 6:

- event LastGoodPriceUpdated and function getDecimalPercision() are added to IPriceFeed.sol in **ETHGMUOracle** and **GMUOracle** contracts, as opposed to **USDCCurveStrategy** and **ETHTroveStrategy**.
  The getDecimalPercision methods returns constant TARGET_DIGITS, which is not used in **USDCCurveStrategy** and **ETHTroveStrategy** contracts, therefore these differences do not affect the protocol logic.
- function lastGoodPrice() is added to IPriceFeed.sol in **USDCCurveStrategy** and **ETHTroveStrategy** contracts, as opposed to **ETHGMUOracle** and **GMUOracle**. However, the lastGoodPrice method is never used in **USDCCurveStrategy** and **ETHTroveStrategy** contracts, therefore these differences do not affect the protocol logic.

Diff. 7: `require(_startTime >= block.timestamp)` in constructor and `emit EpochTriggered();` in `modifier checkEpoch()` are added to `Epoch.sol` in **GMUOracle** contract, as opposed to **StabilityPoolKeeper**.
Both **StabilityPoolKeeper** and **GMUOracle** contracts inherit from `Epoch.sol`, which has the following `constructor(uint256 _period, uint256 _startTime, uint256 _startEpoch)`. However, the **GMUOracle** contract is deployed with `Epoch(86400, block.timestamp, 0)` parameters, which makes the `require(_startTime >= block.timestamp)` sanity check excessive. Therefore, these differences do not affect the protocol logic.

Diff. 8:

- `StakingRewardsChild.sol` inherits from `Ownable.sol` in **USDCCurveStrategy**, as opposed to inheriting from `Operator.sol` in **ETHTroveStrategy**. However `Operator.sol` itself inherits from `Ownable.sol`, therefore this difference does not affect the protocol logic.
- The `onlyOperator` modifier is removed from the `notifyRewardAmount()` method of `StakingRewardsChild.sol` file in **USDCCurveStrategy** contract, as opposed to **ETHTroveStrategy** contract. Any account is able to modify the `rewardRate` state variable of the **USDCCurveStrategy** contract.

Recommendation: add `onlyOwner` modifier to the `notifyRewardAmount` method of the **USDCCurveStrategy** contract.

Summary

The team found inconsistencies in source code files across contracts in the SoW. These inconsistencies indicate that the smart contracts were from different source code repositories. In the case of **USDCCurveStrategy** contract these inconsistencies lead to insufficient access control over one of its methods.

Severity: High.

Searching for the original commit in the client's repository

At this stage, we are looking for the original commit in the client's repository. In the best case, all contracts should be deployed from a single codebase revision to decrease the probability of inconsistency in the contract logic.

See exact way of figuring out this information in section A2.

arth-core

contracts	commit	# contracts

TroveManager
StabilityPool
SortedTroves
Governance
DefaultPool
CommunityIssuance
CollSurplusPool
BorrowerOperations
ActivePool
	latest (2023-01-21T07:20:52-05:00):
	894fa83adbea96f195c9ee3b44dcf14f7a889795
	
	earliest (2023-01-20T12:05:10+05:30):
	3ab82383f1f8a15bfbf3544654aa606385a01845
		9

Conclusion:

All of the contracts related to the arth-core repository are matched to a range of revisions, as shown in table arth-core.

arth-strategies

contracts	commit	# contracts
ETHTroveStrategy_Proxy **USDCCurveStrategy** *+ proxy*	single commit (2023-04-04T16:22:21+03:00): c7952ee2fab92a841c8bf1ed3fb2026cb55864b5	3

ETHTroveStrategy	latest (2023-02-28T12:06:47-05:00):
	77094007e6d26a7a5da304d8c05f6c1656e23950
	
	earliest (2023-01-30T03:08:19+05:30):
	899cece604a064f53e79e37782c2192f9029b41b
		1

Conclusion:

There is no original commit for all contracts in the SoW related to the arth-strategies repository. We found two different groups of contracts (see table arth-strategies) and matched them to two different code revisions.

chainlink-keepers

contracts	commit	# contracts

StabilityPoolKeeper	latest (2023-01-22T14:56:58+05:30):
	474ca343c5ce2983bc158cf85734e09ae9e5fb3b
	earliest (2023-01-21T16:35:06+05:30):
	7d4c9c1e19387d2bd5028613bba0c5d7ffdf104d
		1

flashloans-arth

contracts	commit	# contracts
ARTHFlashMinter	single commit (2022-06-20T01:28:31+04:00): 7b42a9d9541b833f832d7c4ef54146cd5951dc4b	1

gmu-oracle-contracts

contracts	commit	# contracts

ETHGMUOracle
GMUOracle
	latest (2023-02-07T15:41:42+05:30):
	95c12a54410a667433d5d036e9a53eeb1d017876
	earliest (2023-01-28T14:35:52+05:30):
	8c87b5835449f31083ac1e9f070f07339c79d3d2
		2

Conclusion:

All of the contracts related to the gmu-oracle-contracts repository are matched to a range of revisions, as shown in table gmu-oracle-contracts.

token

contracts	commit	# contracts

MahaToken
ARTHValuecoin
	latest (2023-01-28T19:48:48+05:30):
	c75610da6c48b3a79e0703a11852bb0ac2a58a0d
	
	earliest (2023-01-07T11:19:53+05:30):
	643761ba97e36a70c4c7e08ca26ff27853954b9f
		2

Conclusion:

Both contracts related to the token repository are matched to a range of revisions, as shown in table token.

Summary

The project is composed of six repositories, each containing a variety of contracts. We identified the original commit for each group of contracts in their respective repositories. However, we also evaluated the consistency of the contracts across all repositories. We noticed that differences in the codebase, especially in the interfaces of the deployed contracts from different repositories, suggest that the revisions from which they were deployed may have logical inconsistencies. Hence, we recommend conducting a thorough security audit to validate the observed inconsistencies among the deployed contracts.

Severity: **Medium**.

Analyzing the dependencies of the contracts

The goal is to check the consistency of every dependency version and identify any changes across every dependency codebase.

contract	@openzeppelin
ETHTroveStrategy_Implementation	4.8.2
USDCCurveStrategy_Proxy	4.8.2
USDCCurveStrategy_Implementation	4.8.2
StabilityPoolKeeper	4.7.3
GMUOracle	4.5.0
ETHGMUOracle	4.5.0
ETHTroveStrategy_Proxy	4.5.0
ARTHFlashMinter	4.5.0
ARTHValuecoin	4.3.3

Summary

The contracts in the SoW use OpenZeppelin dependencies. Four different versions of the OpenZeppelin smart contracts are used in the SoW's contracts: 4.3.3, 4.5.0, 4.7.3 and 4.8.2. The security issues of using four different versions of OpenZeppelin should be investigated during the security audit of the deployed smart contracts code.

Severity: Medium

Deployment check: storage

We thoroughly examine both its public and private storage to ensure that there are no misconfigurations, especially:

1. Incorrect or outdated addresses to other smart contracts referenced in the scope of work (SoW) - this includes addresses stored in variables, mappings, and other data structures.
2. Any references to other smart contracts or externally owned accounts (EOAs) that may be incorrect or outdated.
3. Any incorrect protocol settings stored in variables or other data structures.
4. Misconfigurations related to the roles and permissions of the contract.
5. Governance issues that may impact the operation and business logic of the smart contract.

ARTHFlashMinter

NOTE: At the time of initial check (24.01.2023) this contract was not in the scope of audit.

issue #	issue type	initial check status	first re-check status	second re-check status
1	EOA	not discovered	acknowledged	fixed

1. `ecosystemFund = 0x6bfc9DB28f0A6d11a8d9d64c86026DDD2fad293B`: EOA
   - second re-check: fixed to `FeesSplitter (0x9032F1Bd0cc645Fde1b41941990dA85f265A7623)`

ARTHValuecoin

NOTE: critical findings are marked with ⚠.

issue #	issue type	initial check status	first re-check status	second re-check status
1 ⚠	EOA	acknowledged	fixed	--

1. `troveManagerAddresses(0xaefb39d1bc9f5f506730005ec96ff10b4ded8dda) -> true`: EOA
   - first re-check: fixed to `false`

USDCCurveStrategy

NOTE: At the time of first re-check (24.01.2023) the provided implementation address was 0x122f4530c2c8ed9a7dc4846a155579ede0e23ecb. Since then the MAHA team changed it to 0x9ff6629d08fddaec63b0d855b9c29acdf4dc14e4 and notified the Mundus Security team.

Critical findings are marked with ⚠.

issue #	issue type	initial check status	first re-check status	second re-check status
1 ⚠	compromised EOA	acknowledged	remains	fixed
2 ⚠	compromised EOA	acknowledged	remains	fixed
3 ⚠	compromised EOA	acknowledged	remains	fixed
4 ⚠	outdated contract	acknowledged	remains	fixed
5	out of scope contract	dismissed	--	--
6	out of scope contract	dismissed	--	--
7	out of scope contract	dismissed	--	--
8	out of scope contract	dismissed	--	--

1. treasury = 0x67c569F960C1Cc0B9a7979A851f5a67018c5A3b0: compromised EOA
   - second re-check: fixed to FeesSplitter (0x9032F1Bd0cc645Fde1b41941990dA85f265A7623)
2. _owner = 0x67c569F960C1Cc0B9a7979A851f5a67018c5A3b0: compromised EOA
   - second re-check: fixed to 0x0
3. _operator = 0x67c569F960C1Cc0B9a7979A851f5a67018c5A3b0: compromised EOA
   - second re-check: fixed by deploying a new implementation contract with no _operator storage slot
4. rewardsToken = 0xb4d930279552397bba2ee473229f89ec245bc365: outdated contract
   - second re-check: fixed to MahaToken (745407c86df8db893011912d3ab28e68b62e49b0)

5. liquidityPool = 0xb4018cb02e264c3fcfe0f21a1f5cfbcaaba9f61f: out of scope ARTH/USDC Curve.fi pool contract
6. lendingPool = 0x76f0c94ced5b48020bf0d7f3d0ceabc877744cb5: out of scope AAVE pool contract
7. varDebtArth = 0x93c457512aae663f36e555d2ad62e1dee9d91836: out of scope ARTH AAVE VariableDebtToken contract
8. lp = 0xdf34bad1d3b16c8f28c9cf95f15001949243a038: out of scope ARTH/USDC Curve.fi pool contract

ETHTroveStrategy

NOTE: At the time of initial check (24.01.2023) the provided implementation address was 0x94cdc9fbd3a4b8517e94ea9b5d2af509633e6dfa. Since then the MAHA team changed it first to 0xd08633560c51982cc5fd2c6a8c1c5ed9d798dcbe and then to 0xf3f261f54d83978061325 98dc2b6b5c00d6eb3ea and notified the Mundus Security team.

Critical findings are marked with ⚠.

issue #	issue type	initial check	first re-check	second re-check status
1	wrong value	acknowledged	fixed	--
2	EOA	acknowledged	remains	dismissed
3 ⚠	outdated contract	acknowledged	fixed	--
4	out of scope contract	dismissed	--	--

1. _IMPLEMENTATION_SLOT -> 0x66b18f8d058276a03fbfe955a9586dca8fb5ca59: wrong value
   - first re-check: fixed to current implementation (0xf3f261f54d83978061325 98dc2b6b5c00d6eb3ea)
2. _operator = 0x77cd66d59ac48a0E7CE54fF16D9235a5fffF335E: EOA
   - first re-check: changed to 0xd35D97b968704beE1898e46012a776bf51E3c06e: EOA
3. borrowerOperations = 0xd3761e54826837b8bbd6ef0a278d5b647b807583: outdated contract
   - first re-check: fixed to BorrowerOperations (0x4c50063f8238dea92c738f23221733a9a6c6888b)
4. pool = 0x76f0c94ced5b48020bf0d7f3d0ceabc877744cb5: out of scope contract (DAI, USDC, ARTH, MAHA AAVE pool)

GMUOracle

NOTE: At the time of initial check (24.01.2023) the provided contract address was 0x7EE5010Cbd5e499b7d66a7cbA2Ec3BdE5fca8e00. Since then the MAHA team changed it to 0x066A917fA2e1739ccfc306dc73ff78EECa8B6F29 and notified the Mundus Security team.

Critical findings are marked with ⚠.

issue #	**issue type**	**initial check status**	**first re-check status**	**second re-check status**
1 ⚠	compromised EOA	acknowledged	fixed	--
2	out of scope contract	dismissed	--	--

1. _operator = 0x67c569F960C1Cc0B9a7979A851f5a67018c5A3b0: compromised EOA
    - first re-check: fixed by deploying a new contract with no _operator storage slot
2. _oracle = 0x4c517D4e2C851CA76d7eC94B805269Df0f2201De: out of scope Liquity Pricefeed contract
    - first re-check: new version of contract does not contain _oracle storage slot

BorrowerOperations

issue #	issue type	initial check status	first re-check status	second re-check status
1	out of scope contract	dismissed	--	--

1. `_owner = 0x67002ECB9934312DF2aE28fE522C72c775e952BE`: out of scope GasPool contract

TroveManager

issue #	issue type	initial check status	first re-check status	second re-check status
1	out of scope contract	dismissed	--	--

1. `gasPoolAddress = 0x67002ECB9934312DF2aE28fE522C72c775e952BE`: out of scope GasPool contract

StabilityPool

issue #	issue type	initial check status	first re-check status	second re-check status
1	out of scope contract	dismissed	--	--
2	wrong value	dismissed	--	--

1. `_owner = 0x67002ECB9934312DF2aE28fE522C72c775e952BE`: out of scope GasPool contract
2. `defaultPool = 0x0`: unset storage variable

Disclaimers

Mundus disclaimer

The smart contracts given for audit have been analyzed in accordance with the best industry practices at the date of this report, in relation to cybersecurity vulnerabilities and issues in smart contract source code, the details of which are disclosed in this report (Source Code); the Source Code compilation, deployment, and functionality (performing the intended functions).

The audit makes no statements or warranties on the security of the code. It also cannot be considered as a sufficient assessment regarding the utility and safety of the code, bug-free status, or any other statements of the contract. While we have done our best in conducting the analysis and producing this report, it is important to note that you should not rely on this report only – we recommend proceeding with several independent audits and a public bug bounty program to ensure the security of smart contracts.

Technical disclaimers

Smart contracts are deployed and executed on a blockchain platform. The platform, its programming language, and other software related to the smart contract can have vulnerabilities that can lead to hacks. Thus, the audit can't guarantee the explicit security of the audited smart contracts.

Appendix

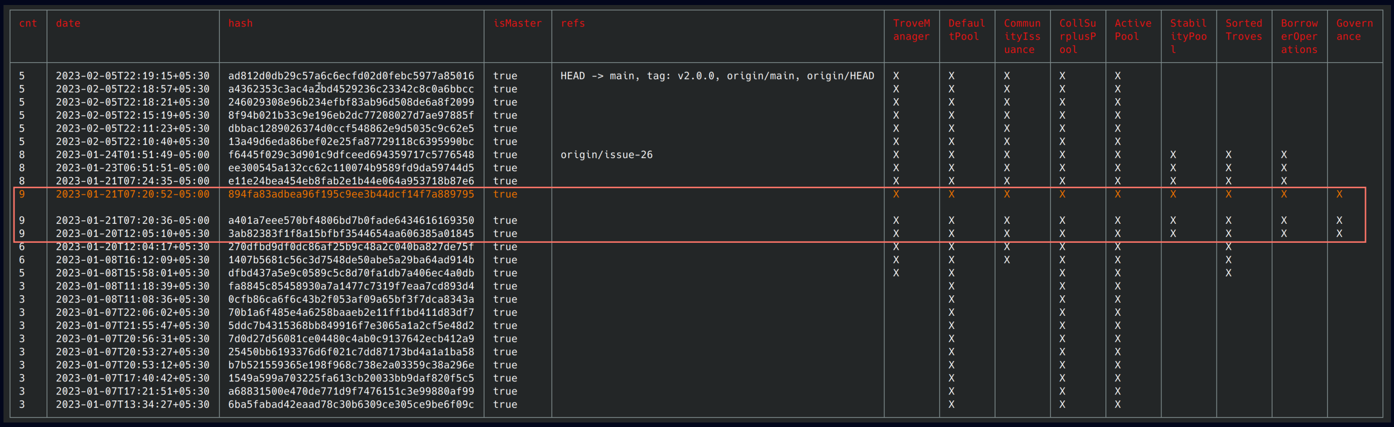
A1. Statistics among contracts

contract	# project files	# @openzeppelin files
ActivePool	6	0
ARTHFlashMinter	3	6
USDCCurveStrategy_Implementation	7	8
USDCCurveStrategy_Proxy	1	7
ARTHValuecoin	1	10
BorrowerOperations	22	0
CollSurplusPool	5	0
CommunityIssuance	11	0
DefaultPool	6	0
ETHGMUOracle	4	1
ETHTroveStrategy_Implementation	8	7
ETHTroveStrategy_Proxy	1	7
GMUOracle	6	4
Governance	10	0
SortedTroves	15	0
StabilityPoolKeeper	5	5
StabilityPool	23	0
TroveManager	21	0

A2. Original commit in the client's repository

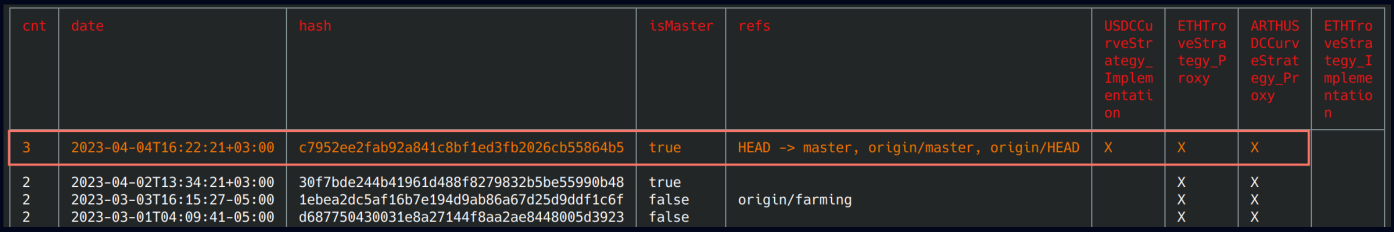
This section contains output screenshots of the internal tool. We use it to determine suitable set of revisions in contracts' repos.

arth-core

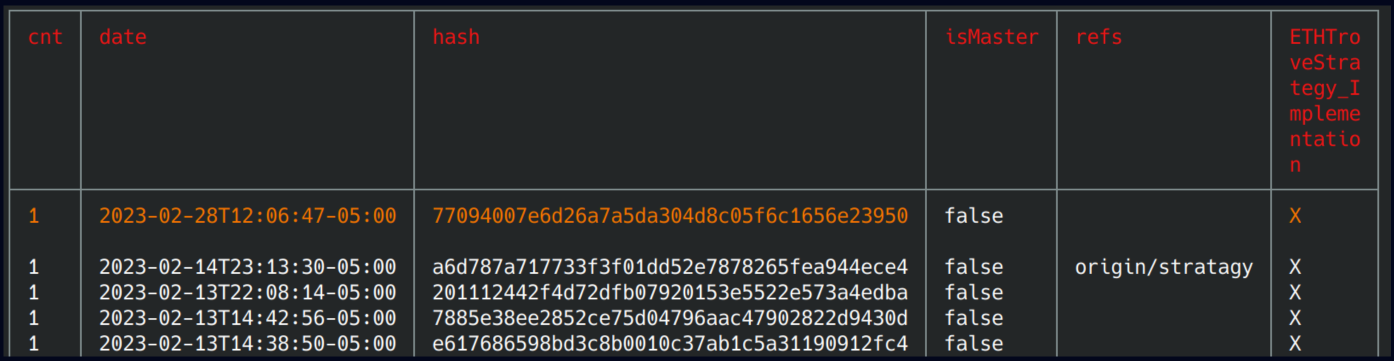


cnt	date	hash	isMaster	refs	TroveManager	DefaultPool	CommunityIssuance	CollSurplusPool	ActivePool	StabilityPool	SortedTroves	BorrowerOperations	Governance
5	2023-02-05T22:19:15+05:30	ad812d0db29c57a6c6ecfd02d0febc5977a85016	true	HEAD -> main, tag: v2.0.0, origin/main, origin/HEAD	X	X	X	X	X				
5	2023-02-05T22:18:57+05:30	a4362353c3ac4a2bd4529236c23342c8c0a6bbcc	true		X	X	X	X	X				
5	2023-02-05T22:18:21+05:30	246029308e96b234efbf83ab96d508de6a8f2099	true		X	X	X	X	X				
5	2023-02-05T22:15:19+05:30	8f94b021b33c9e196eb2dc77208027d7ae97885f	true		X	X	X	X	X				
5	2023-02-05T22:11:23+05:30	dbbac1289026374d0ccf548862e9d5035c9c62e5	true		X	X	X	X	X				
5	2023-02-05T22:10:40+05:30	13a49d6eda86bef02e25fa87729118c6395990bc	true		X	X	X	X	X				
8	2023-01-24T01:51:49-05:00	f6445f029c3d901c9dfceed694359717c5776548	true	origin/issue-26	X	X	X	X	X	X	X	X	
8	2023-01-23T06:51:51-05:00	ee300545a132cc62c110074b9589fd9da59744d5	true		X	X	X	X	X	X	X	X	
8	2023-01-21T07:24:35-05:00	e11e24bea454eb8fab2e1b44e064a953718b87e6	true		X	X	X	X	X	X	X	X	
9	2023-01-21T07:20:52-05:00	894fa83adbea96f195c9ee3b44dcf14f7a889795	true		X	X	X	X	X	X	X	X	X
9	2023-01-21T07:20:36-05:00	a401a7eee570bf4806bd7b0fade6434616169350	true		X	X	X	X	X	X	X	X	X
9	2023-01-20T12:05:10+05:30	3ab82383f1f8a15bfbf3544654aa606385a01845	true		X	X	X	X	X	X	X	X	X
6	2023-01-20T12:04:17+05:30	270dfbd9df0dc86af25b9c48a2c040ba827de75f	true		X	X	X	X	X		X		
6	2023-01-08T16:12:09+05:30	1407b5681c56c3d7548de50abe5a29ba64ad914b	true		X	X	X	X	X		X		
5	2023-01-08T15:58:01+05:30	dfbd437a5e9c0589c5c8d70fa1db7a406ec4a0db	true		X	X		X	X		X		
3	2023-01-08T11:18:39+05:30	fa8845c85458930a7a1477c7319f7eaa7cd893d4	true			X		X	X				
3	2023-01-08T11:08:36+05:30	0cfb86ca6f6c43b2f053af09a65bf3f7dca8343a	true			X		X	X				
3	2023-01-07T22:06:02+05:30	70b1a6f485e4a6258baaeb2e11ff1bd411d83df7	true			X		X	X				
3	2023-01-07T21:55:47+05:30	5ddc7b4315368bb849916f7e3065a1a2cf5e48d2	true			X		X	X				
3	2023-01-07T20:56:31+05:30	7d0d27d56081ce04480c4ab0c9137642ecb412a9	true			X		X	X				
3	2023-01-07T20:53:27+05:30	25450bb6193376d6f021c7dd87173bd4a1a1ba58	true			X		X	X				
3	2023-01-07T20:53:12+05:30	b7b521559365e198f968c738e2a03359c38a296e	true			X		X	X				
3	2023-01-07T17:40:42+05:30	1549a599a703225fa613cb20033bb9daf820f5c5	true			X		X	X				
3	2023-01-07T17:21:51+05:30	a68831500e470de771d9f7476151c3e99880af99	true			X		X	X				
3	2023-01-07T13:34:27+05:30	6ba5fabad42eaad78c30b6309ce305ce9be6f09c	true			X		X	X				

arth-strategies

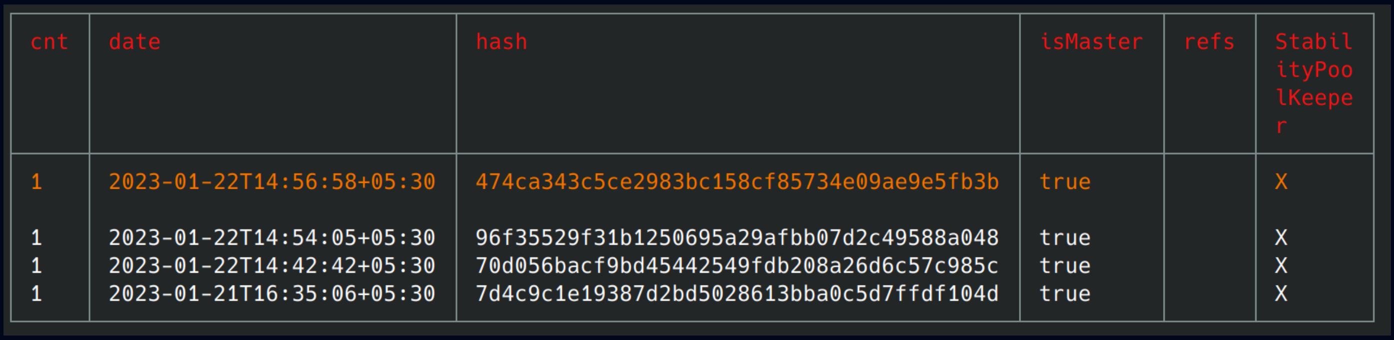


cnt	date	hash	isMaster	refs	USDCCurveStrategy_Implementation	ETHTroveStrategy_Proxy	ARTHUSDCCurveStrategy_Proxy	ETHTroveStrategy_Implementation
3	2023-04-04T16:22:21+03:00	c7952ee2fab92a841c8bf1ed3fb2026cb55864b5	true	HEAD -> master, origin/master, origin/HEAD	X	X	X	
2	2023-04-02T13:34:21+03:00	30f7bde244b41961d488f8279832b5be55990b48	true			X	X	
2	2023-03-03T16:15:27-05:00	1ebea2dc5af16b7e194d9ab86a67d25d9ddf1c6f	false	origin/farming		X	X	
2	2023-03-01T04:09:41-05:00	d687750430031e8a27144f8aa2ae8448005d3923	false			X	X	



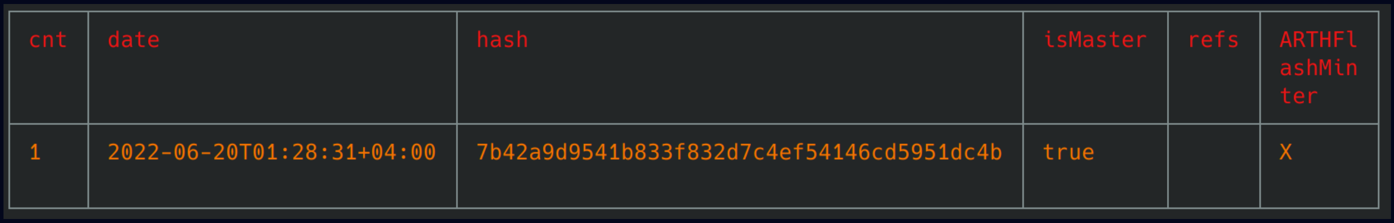
cnt	date	hash	isMaster	refs	ETHTroveStrategy_Implementation
1	2023-02-28T12:06:47-05:00	77094007e6d26a7a5da304d8c05f6c1656e23950	false		X
1	2023-02-14T23:13:30-05:00	a6d787a717733f3f01dd52e7878265fea944ece4	false	origin/stratagy	X
1	2023-02-13T22:08:14-05:00	201112442f4d72dfb07920153e5522e573a4edba	false		X
1	2023-02-13T14:42:56-05:00	7885e38ee2852ce75d04796aac47902822d9430d	false		X
1	2023-02-13T14:38:50-05:00	e617686598bd3c8b0010c37ab1c5a31190912fc4	false		X

chainlink-keepers



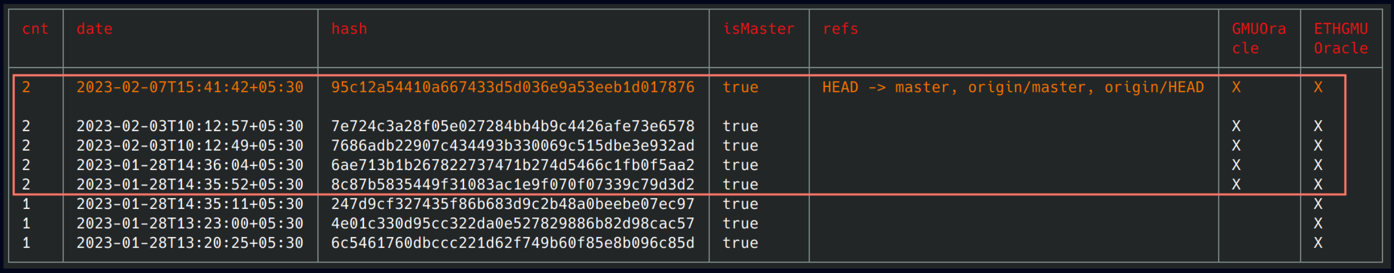
cnt	date	hash	isMaster	refs	StabilityPoolKeeper
1	2023-01-22T14:56:58+05:30	474ca343c5ce2983bc158cf85734e09ae9e5fb3b	true		X
1	2023-01-22T14:54:05+05:30	96f35529f31b1250695a29afbb07d2c49588a048	true		X
1	2023-01-22T14:42:42+05:30	70d056bacf9bd45442549fdb208a26d6c57c985c	true		X
1	2023-01-21T16:35:06+05:30	7d4c9c1e19387d2bd5028613bba0c5d7ffdf104d	true		X

flashloans-arth



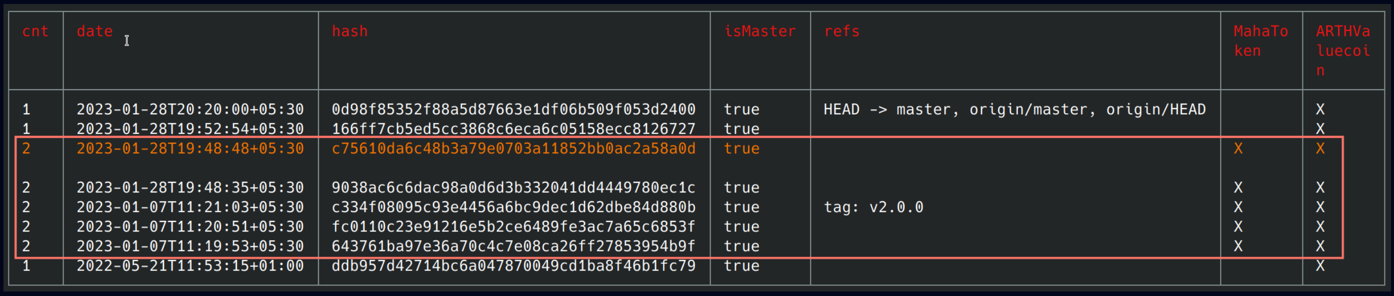
cnt	date	hash	isMaster	refs	ARTHFlashMinter
1	2022-06-20T01:28:31+04:00	7b42a9d9541b833f832d7c4ef54146cd5951dc4b	true		X

gmu-oracle-contracts



cnt	date	hash	isMaster	refs	GMUOracle	ETHGMUOracle
2	2023-02-07T15:41:42+05:30	95c12a54410a667433d5d036e9a53eeb1d017876	true	HEAD -> master, origin/master, origin/HEAD	X	X
2	2023-02-03T10:12:57+05:30	7e724c3a28f05e027284bb4b9c4426afe73e6578	true		X	X
2	2023-02-03T10:12:49+05:30	7686adb22907c434493b330069c515dbe3e932ad	true		X	X
2	2023-01-28T14:36:04+05:30	6ae713b1b267822737471b274d5466c1fb0f5aa2	true		X	X
2	2023-01-28T14:35:52+05:30	8c87b5835449f31083ac1e9f070f07339c79d3d2	true		X	X
1	2023-01-28T14:35:11+05:30	247d9cf327435f86b683d9c2b48a0beebe07ec97	true			X
1	2023-01-28T13:23:00+05:30	4e01c330d95cc322da0e527829886b82d98cac57	true			X
1	2023-01-28T13:20:25+05:30	6c5461760dbccc221d62f749b60f85e8b096c85d	true			X

token



cnt	date	hash	isMaster	refs	MahaToken	ARTHValuecoin
1	2023-01-28T20:20:00+05:30	0d98f85352f88a5d87663e1df06b509f053d2400	true	HEAD -> master, origin/master, origin/HEAD		X
1	2023-01-28T19:52:54+05:30	166ff7cb5ed5cc3868c6eca6c05158ecc8126727	true			X
2	2023-01-28T19:48:48+05:30	c75610da6c48b3a79e0703a11852bb0ac2a58a0d	true		X	X
2	2023-01-28T19:48:35+05:30	9038ac6c6dac98a0d6d3b332041dd4449780ec1c	true		X	X
2	2023-01-07T11:21:03+05:30	c334f08095c93e4456a6bc9dec1d62dbe84d880b	true	tag: v2.0.0	X	X
2	2023-01-07T11:20:51+05:30	fc0110c23e91216e5b2ce6489fe3ac7a65c6853f	true		X	X
2	2023-01-07T11:19:53+05:30	643761ba97e36a70c4c7e08ca26ff27853954b9f	true		X	X
1	2022-05-21T11:53:15+01:00	ddb957d42714bc6a047870049cd1ba8f46b1fc79	true			X