88MPH process Quality Review

Score: 79%

This is a Process Quality Review of 88MPH completed on March 30, 2021. It was performed using the Process Review process (version 0.6.2) and is documented here. The review was performed by Lucas of DeFiSafety. Check out our Telegram.

The final score of the review is 79%, a pass. The breakdown of the scoring is in Scoring Appendix. For our purposes, a pass is 70%.

Summary of the Process

Very simply, the review looks for the following declarations from the developer's site. With these declarations, it is reasonable to trust the smart contracts.

- **Here are my smart contracts on the blockchain**
- **Here is the documentation that explains what my smart contracts do**
- **Here are the tests I ran to verify my smart contract**
- **Here are the audit(s) performed on my code by third party experts**

Disclaimer

This report is for informational purposes only and does not constitute investment advice of any kind, nor does it constitute an offer to provide investment advisory or other services. Nothing in this report shall be considered a solicitation or offer to buy or sell any security, token, future, option or other financial instrument or to offer or provide any investment advice or service to any person in any jurisdiction. Nothing contained in this report constitutes investment advice or offers any opinion with respect to the suitability of any security, and the views expressed in this report should not be taken as advice to buy, sell or hold any security. The information in this report should not be relied upon for the purpose of investing. In preparing the information contained in this report, we have not taken into account the investment needs, objectives and financial circumstances of any particular investor. This information has no regard to the specific investment objectives, financial situation and particular needs of any specific recipient of this information and investments discussed may not be suitable for all investors.

Any views expressed in this report by us were prepared based upon the information available to us at the time such views were written. The views expressed within this report are limited to DeFiSafety and the author and do not reflect those of any additional or third party and are strictly based upon DeFiSafety, its authors, interpretations and evaluation of relevant data. Changed or additional information could cause such views to change. All information is subject to possible correction. Information may quickly become unreliable for various reasons, including changes in market conditions or economic circumstances.

Code and Team

This section looks at the code deployed on the Mainnet that gets reviewed and its corresponding software repository. The document explaining these questions is here. This review will answer the questions;

1. Are the executing code addresses readily available? (Y/N)
2. Is the code actively being used? (%)
3. Is there a public software repository? (Y/N)
4. Is there a development history visible? (%)
5. Is the team public (not anonymous)? (Y/N)

Are the executing code addresses readily available? (Y/N)

> ✓ Answer: Yes

They are available at website https://88mph.app/docs/addresses/ as indicated in the Appendix.

Is the code actively being used? (%)

> ✓ Answer: 100%

Activity is 16 transactions a day on contract rewards.sol, as indicated in the Appendix.

Percentage Score Guidance

100%	More than 10 transactions a day
70%	More than 10 transactions a week
40%	More than 10 transactions a month
10%	Less than 10 transactions a month
0%	No activity

Is there a public software repository? (Y/N)

> ✓ Answer: Yes

GitHub: https://github.com/88mphapp/88mph-contracts

Is there a public software repository with the code at a minimum, but normally test and scripts also (Y/N). Even if the repo was created just to hold the files and has just 1 transaction, it gets a Yes. For teams with private repos, this answer is No.

Is there a development history visible? (%)

> ✓ Answer: 100%

With 161 commits and 10 branches, this is a healthy repo.

This checks if the software repository demonstrates a strong steady history. This is normally demonstrated by commits, branches and releases in a software repository. A healthy history demonstrates a history of more than a month (at a minimum).

Guidance:

100%	Any one of 100+ commits, 10+branches
70%	Any one of 70+ commits, 7+branches

50%	Any one of 50+ commits, 5+branches
30%	Any one of 30+ commits, 3+branches
0%	Less than 2 branches or less than 10 commits

Is the team public (not anonymous)? (Y/N)

> ✓ Answer: Yes

The names of the team can be seen on on their Medium Articles.

For a yes in this question the real names of some team members must be public on the website or other documentation. If the team is anonymous and then this question is a No.

Documentation

This section looks at the software documentation. The document explaining these questions is here.

Required questions are;

1. Is there a whitepaper? (Y/N)
2. Are the basic software functions documented? (Y/N)
3. Does the software function documentation fully (100%) cover the deployed contracts? (%)
4. Are there sufficiently detailed comments for all functions within the deployed contract code (%)
5. Is it possible to trace from software documentation to the implementation in codee (%)

Is there a whitepaper? (Y/N)

> ✓ Answer: Yes

Location: https://88mph.app/docs/

Are the basic software functions documented? (Y/N)

> ✓ Answer: Yes

They are well-documented on their Smart Contract Reference documentation.

Does the software function documentation fully (100%) cover the deployed contracts? (%)

> ✓ Answer: 80%

All major contracts are well-documented

Guidance:

100%	All contracts and functions documented
80%	Only the major functions documented
79-1%	Estimate of the level of software documentation
0%	No software documentation

Are there sufficiently detailed comments for all functions within the deployed contract code (%)

> ⚠ Answer: 0%

With a SLOC of under 20%, there is no or little useful commenting in the code.

Code examples are in the Appendix. As per the SLOC, there is 18% commenting to code (CtC).

The Comments to Code (CtC) ratio is the primary metric for this score.

Guidance:
100% CtC > 100 Useful comments consistently on all code
90-70% CtC > 70 Useful comment on most code
60-20% CtC > 20 Some useful commenting
0% CtC < 20 No useful commenting

How to improve this score

This score can improve by adding comments to the deployed code such that it comprehensively covers the code. For guidance, refer to the SecurEth Software Requirements.

Is it possible to trace from software documentation to the implementation in code (%)

> ⓘ Answer: 60%

Clear association between code and documents via non explicit traceability

Guidance:
100% - Clear explicit traceability between code and documentation at a requirement level for all code
60% - Clear association between code and documents via non explicit traceability
40% - Documentation lists all the functions and describes their functions
0% - No connection between documentation and code

How to improve this score

This score can improve by adding traceability from requirements to code such that it is clear where each requirement is coded. For reference, check the SecurEth guidelines on traceability.

Testing

This section looks at the software testing available. It is explained in this document. This section answers the following questions;

1. Full test suite (Covers all the deployed code) (%)
2. Code coverage (Covers all the deployed lines of code, or explains misses) (%)
3. Scripts and instructions to run the tests (Y/N)
4. Packaged with the deployed code (Y/N)
5. Report of the results (%)
6. Formal Verification test done (%)
7. Stress Testing environment (%)

Is there a Full test suite? (%)

> Answer: 40%

With a TtC ratio of 90%, this has a lower test base than preferred. Based on test results listed in the Quantstamp audit report, a score of 40%

This score is guided by the Test to Code ratio (TtC). Generally a good test to code ratio is over 100%. However the reviewers best judgement is the final deciding factor.

Guidance:
100% TtC > 120% Both unit and system test visible
80% TtC > 80% Both unit and system test visible
40% TtC < 80% Some tests visible
0% No tests obvious

How to improve this score

This score can improve by adding tests to fully cover the code. Document what is covered by traceability or test results in the software repository.

Code coverage (Covers all the deployed lines of code, or explains misses) (%)

> Answer: 42%

There is a coverage result in the Quantstamp audit report.

Guidance:
100% - Documented full coverage
99-51% - Value of test coverage from documented results
50% - No indication of code coverage but clearly there is a reasonably complete set of tests
30% - Some tests evident but not complete
0% - No test for coverage seen

How to improve this score

This score can improve by adding tests achieving full code coverage. A clear report and scripts in the software repository will guarantee a high score.

Scripts and instructions to run the tests (Y/N)

> ✓ Answer: Yes

Location:https://github.com/88mphapp/88mph-contracts

How to improve this score

Add the scripts to the repository and ensure they work. Ask an outsider to create the environment and run the tests. Improve the scripts and docs based on their feedback.

Packaged with the deployed code (Y/N)

> ✓ Answer: Yes

Report of the results (%)

> ⚠ Answer: 0%

No test report evident

Guidance:
100% - Detailed test report as described below
70% - GitHub Code coverage report visible
0% - No test report evident

How to improve this score

Add a report with the results. The test scripts should generate the report or elements of it.

Formal Verification test done (%)

⚠ Answer: 0%

There is no evidence of formal verification present.

Stress Testing environment (%)

⚠ Answer: 0%

There are no published Kovan or Ropsten TestNet addresses published, and therefore verification of stress-testing is impossible.

Audits

✓ Answer: 100%

88MPH has been audited by 4 different groups. It is concerning that 3 of the reports indicate only superficial concerns but the Quantstamp report lists a number of concerning issues on minting access/admin controls.

PeckShield V2

PeckShield V3

QuantStamp

Certik ZC Bond Audit

Guidance:

1. Multiple Audits performed before deployment and results public and implemented or not required (100%)
2. Single audit performed before deployment and results public and implemented or not required (90%)
3. Audit(s) performed after deployment and no changes required. Audit report is public. (70%)
4. No audit performed (20%)
5. Audit Performed after deployment, existence is public, report is not public and no improvements deployed OR smart contract address' not found, question 1 (0%)

Appendices

Author Details

The author of this review is Rex of DeFi Safety.

Email : rex@defisafety.com Twitter : @defisafety

I started with Ethereum just before the DAO and that was a wonderful education. It showed the importance of code quality. The second Parity hack also showed the importance of good process. Here my aviation background offers some value. Aerospace knows how to make reliable code using quality processes.

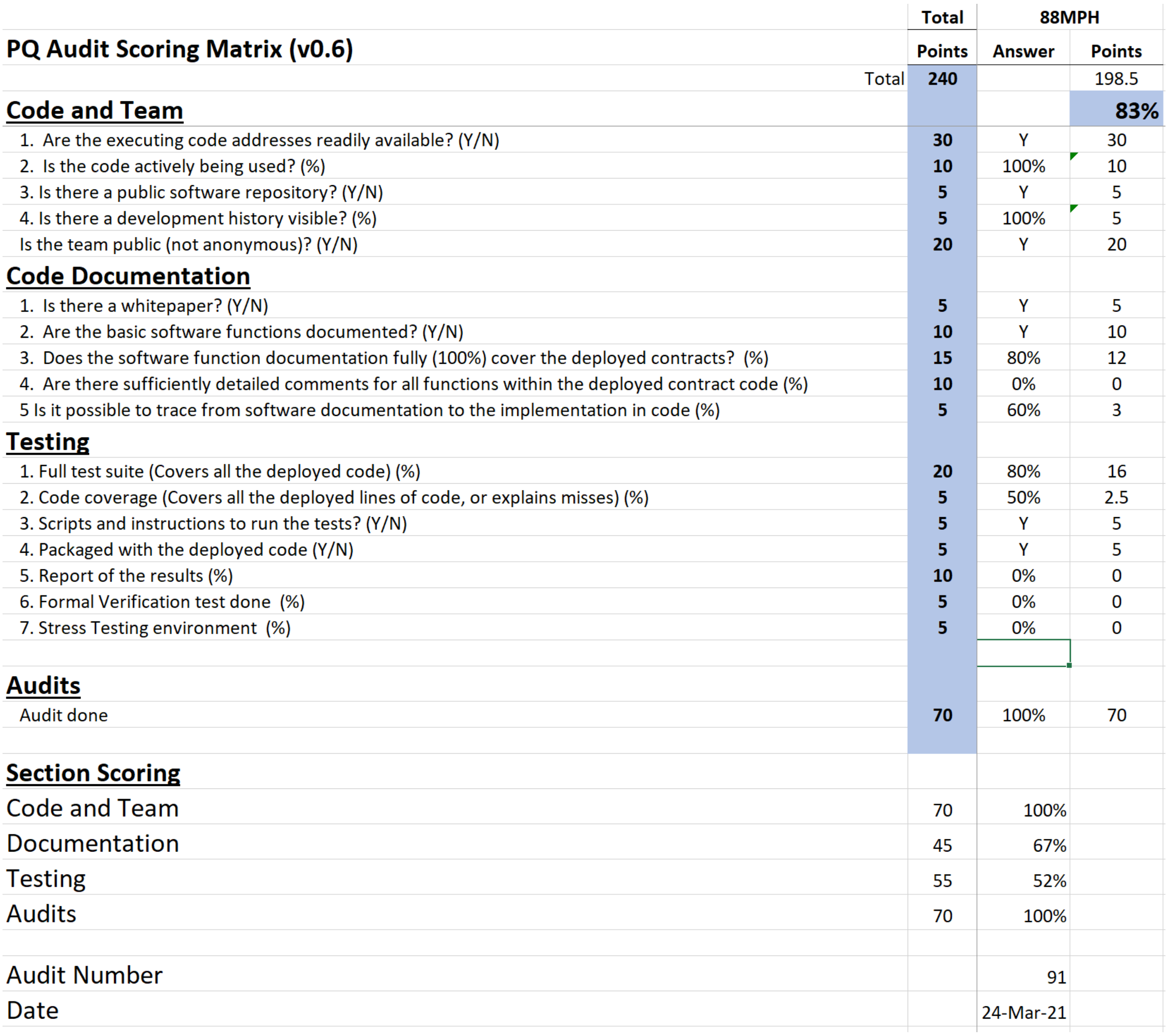
I was coaxed to go to EthDenver 2018 and there I started SecuEth.org with Bryant and Roman. We created guidelines on good processes for blockchain code development. We got

EthFoundation funding to assist in their development.

Process Quality Reviews are an extension of the SecurEth guidelines that will further increase the quality processes in Solidity and Vyper development.

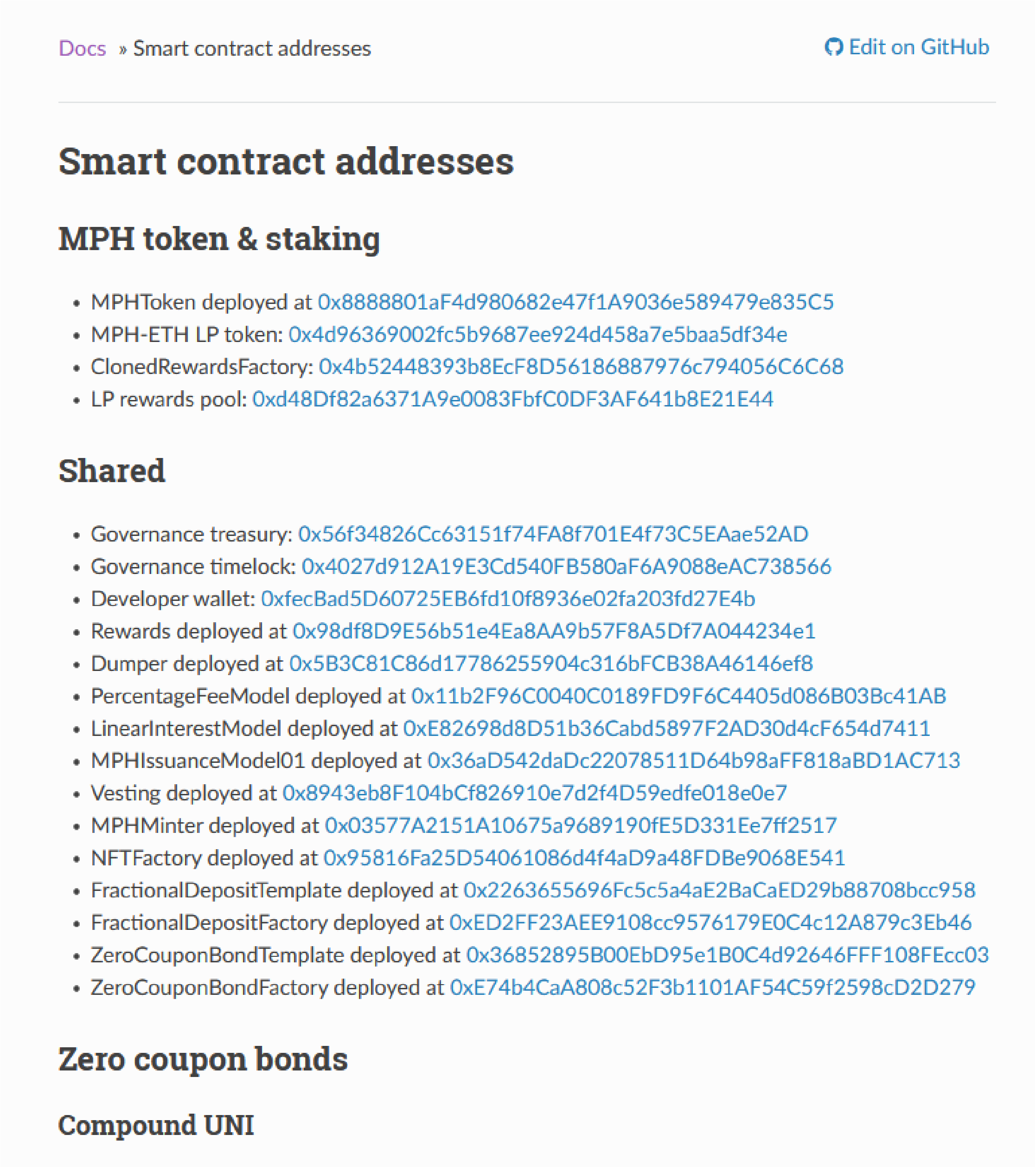
DeFiSafety is my full time gig and we are working on funding vehicles for a permanent staff.

Scoring Appendix



	Total	88MPH	
PQ Audit Scoring Matrix (v0.6)	**Points**	**Answer**	**Points**
Total	**240**		198.5
Code and Team			**83%**
1. Are the executing code addresses readily available? (Y/N)	**30**	Y	30
2. Is the code actively being used? (%)	**10**	100%	10
3. Is there a public software repository? (Y/N)	**5**	Y	5
4. Is there a development history visible? (%)	**5**	100%	5
Is the team public (not anonymous)? (Y/N)	**20**	Y	20
Code Documentation			
1. Is there a whitepaper? (Y/N)	**5**	Y	5
2. Are the basic software functions documented? (Y/N)	**10**	Y	10
3. Does the software function documentation fully (100%) cover the deployed contracts? (%)	**15**	80%	12
4. Are there sufficiently detailed comments for all functions within the deployed contract code (%)	**10**	0%	0
5 Is it possible to trace from software documentation to the implementation in code (%)	**5**	60%	3
Testing			
1. Full test suite (Covers all the deployed code) (%)	**20**	80%	16
2. Code coverage (Covers all the deployed lines of code, or explains misses) (%)	**5**	50%	2.5
3. Scripts and instructions to run the tests? (Y/N)	**5**	Y	5
4. Packaged with the deployed code (Y/N)	**5**	Y	5
5. Report of the results (%)	**10**	0%	0
6. Formal Verification test done (%)	**5**	0%	0
7. Stress Testing environment (%)	**5**	0%	0
Audits			
Audit done	**70**	100%	70
Section Scoring			
Code and Team	70	100%	
Documentation	45	67%	
Testing	55	52%	
Audits	70	100%	
Audit Number		91	
Date		24-Mar-21	

Executing Code Appendix



Docs » Smart contract addresses

Edit on GitHub

Smart contract addresses

MPH token & staking

- MPHToken deployed at 0x8888801aF4d980682e47f1A9036e589479e835C5
- MPH-ETH LP token: 0x4d96369002fc5b9687ee924d458a7e5baa5df34e
- ClonedRewardsFactory: 0x4b52448393b8EcF8D56186887976c794056C6C68
- LP rewards pool: 0xd48Df82a6371A9e0083FbfC0DF3AF641b8E21E44

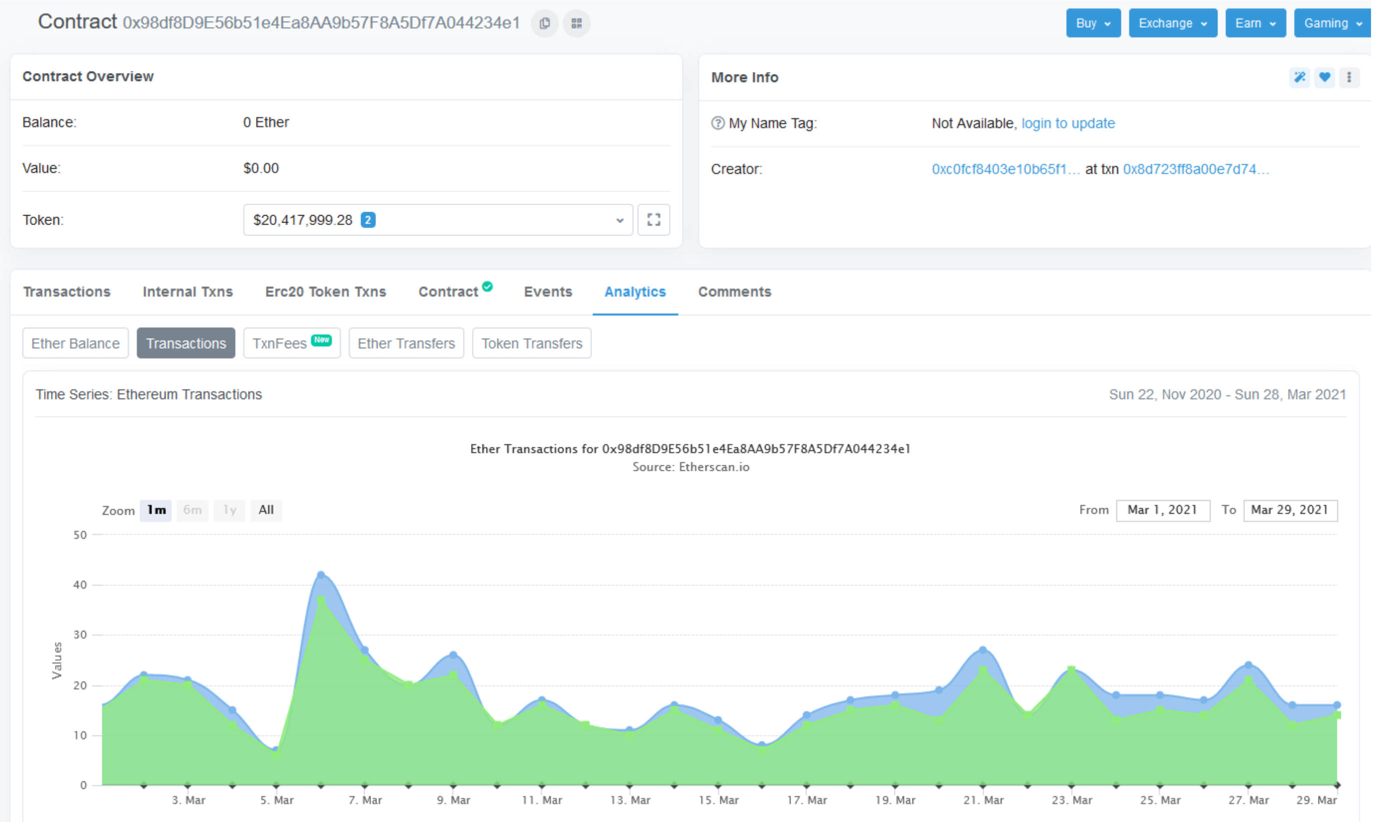
Shared

- Governance treasury: 0x56f34826Cc63151f74FA8f701E4f73C5EAae52AD
- Governance timelock: 0x4027d912A19E3Cd540FB580aF6A9088eAC738566
- Developer wallet: 0xfecBad5D60725EB6fd10f8936e02fa203fd27E4b
- Rewards deployed at 0x98df8D9E56b51e4Ea8AA9b57F8A5Df7A044234e1
- Dumper deployed at 0x5B3C81C86d17786255904c316bFCB38A46146ef8
- PercentageFeeModel deployed at 0x11b2F96C0040C0189FD9F6C4405d086B03Bc41AB
- LinearInterestModel deployed at 0xE82698d8D51b36Cabd5897F2AD30d4cF654d7411
- MPHIssuanceModel01 deployed at 0x36aD542daDc22078511D64b98aFF818aBD1AC713
- Vesting deployed at 0x8943eb8F104bCf826910e7d2f4D59edfe018e0e7
- MPHMinter deployed at 0x03577A2151A10675a9689190fE5D331Ee7ff2517
- NFTFactory deployed at 0x95816Fa25D54061086d4f4aD9a48FDBe9068E541
- FractionalDepositTemplate deployed at 0x2263655696Fc5c5a4aE2BaCaED29b88708bcc958
- FractionalDepositFactory deployed at 0xED2FF23AEE9108cc9576179E0C4c12A879c3Eb46
- ZeroCouponBondTemplate deployed at 0x36852895B00EbD95e1B0C4d92646FFF108FEcc03
- ZeroCouponBondFactory deployed at 0xE74b4CaA808c52F3b1101AF54C59f2598cD2D279

Zero coupon bonds

Compound UNI

Code Used Appendix



Contract 0x98df8D9E56b51e4Ea8AA9b57F8A5Df7A044234e1

Contract Overview

Balance: 0 Ether

Value: $0.00

Token: $20,417,999.28 2

More Info

My Name Tag: Not Available, login to update

Creator: 0xc0fcf8403e10b65f1... at txn 0x8d723ff8a00e7d74...

Time Series: Ethereum Transactions

Sun 22, Nov 2020 - Sun 28, Mar 2021

Ether Transactions for 0x98df8D9E56b51e4Ea8AA9b57F8A5Df7A044234e1
Source: Etherscan.io

Example Code Appendix

```

pragma solidity ^0.5.3;

/// @title Proxy - Generic proxy contract allows to execute all transaction
/// @author Stefan George - <stefan@gnosis.io>
/// @author Richard Meissner - <richard@gnosis.io>
contract Proxy {

    // masterCopy always needs to be first declared variable, to ensure tha
    // To reduce deployment costs this variable is internal and needs to be
    address internal masterCopy;

    /// @dev Constructor function sets address of master copy contract.
    /// @param _masterCopy Master copy address.
    constructor(address _masterCopy)
        public
    {
        require(_masterCopy != address(0), "Invalid master copy address pro
        masterCopy = _masterCopy;
    }

    /// @dev Fallback function forwards all transactions and returns all re
    function ()
        external
        payable
    {
        // solium-disable-next-line security/no-inline-assembly
        assembly {
            let masterCopy := and(sload(0), 0xffffffffffffffffffffffffffffff
            // 0xa619486e == keccak("masterCopy()"). The value is right padd
            if eq(calldataload(0), 0xa619486e000000000000000000000000000000
                mstore(0, masterCopy)
                return(0, 0x20)
            }
            calldatacopy(0, 0, calldatasize())
            let success := delegatecall(gas, masterCopy, 0, calldatasize(),
            returndatacopy(0, 0, returndatasize())
            if eq(success, 0) { revert(0, returndatasize()) }
            return(0, returndatasize())
        }
    }
}
```

SLOC Appendix

Solidity Contracts

Language	Files	Lines	Blanks	Comments	Code	Complexity
Solidity	38	4991	649	677	3665	211

Comments to Code 677/3665 = 18%

Javascript Tests

Language	Files	Lines	Blanks	Comments	Code	Complexity
JavaScript	9	4962	920	735	3307	65

Tests to Code 3307/3665 = 90%