0.7

0x Process Quality Review

Score: 94%

Overview

This is a 0x Protocol Process Quality Review completed on September 29th 2021. It was performed using the Process Review process (version 0.7.3) and is documented here. The review was performed by Nic of DeFiSafety. Check out our Telegram.

The final score of the review is **94%**, an awesome **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**
- **Here are the admin controls and strategies**

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.

Chain

This section indicates the blockchain used by this protocol.

> ✓ **Chain:** Ethereum

Guidance:

Ethereum
Binance Smart Chain
Polygon
Avalanche
Terra
Celo
Arbitrum
Solana

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 following questions:

1) Are the executing code addresses readily available? (%)
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)

1) Are the executing code addresses readily available? (%)

> ✓ **Answer:** 100%

They are available at website https://protocol.0x.org/en/latest/basics/addresses.html, as indicated in the Appendix.

Guidance:

100%	Clearly labelled and on website, docs or repo, quick to find
70%	Clearly labelled and on website, docs or repo but takes a bit of looking
40%	Addresses in mainnet.json, in discord or sub graph, etc
20%	Address found but labeling not clear or easy to find
0%	Executing addresses could not be found

2) Is the code actively being used? (%)

> **Answer:** 100%

Activity is over 10 transactions a day on contract *ZeroEx.sol*, as indicated in the Appendix.

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

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

> **Answer:** Yes

GitHub: https://github.com/0xProject/protocol

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

4) Is there a development history visible? (%)

> **Answer:** 100%

With 16,743 commits and 66 branches, 0x Protocol's main GitHub repository has a great development history.

This metric 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 30 commits

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

> ✓ **Answer:** Yes

Location: https://0x.org/about/team.

For a **"Yes"** in this question, the real names of some team members must be public on the website or other documentation (LinkedIn, etc). If the team is anonymous, then this question is a **"No"**.

Documentation

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

Required questions are;

6) Is there a whitepaper? (Y/N)
7) Are the basic software functions documented? (Y/N)
8) Does the software function documentation fully (100%) cover the deployed contracts? (%)
9) Are there sufficiently detailed comments for all functions within the deployed contract code (%)
10) Is it possible to trace from software documentation to the implementation in code (%)

6) Is there a whitepaper? (Y/N)

> ✓ **Answer:** Yes

Location: https://0x.org/docs/core-concepts#introduction.

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

> ✓ **Answer:** Yes

All of the 0x Protocol basic software functions are documented in https://protocol.0x.org/en/latest/basics/orders.html.

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

> **Answer:** 100%

0x's software documentation covers all of their deployed contracts in the "Basics", "Advanced", "Architecture", and "Tokenomics" sections.

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

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

> **Answer:** 58%

Code examples are in the Appendix. As per the SLOC, there is 58% 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.

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

> **Answer:** 100%

There is clear and explicit traceability between the documented 0x software functions and their implementation in the protocol's source code. This can be seen in the "Architecture" section of their documentation.

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

Testing

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

11) Full test suite (Covers all the deployed code) (%)
12) Code coverage (Covers all the deployed lines of code, or explains misses) (%)
13) Scripts and instructions to run the tests (Y/N)
14) Report of the results (%)
15) Formal Verification test done (%)
16) Stress Testing environment (%)

11) Is there a Full test suite? (%)

> ✓ **Answer:** 100%

Code examples are in the Appendix. As per the SLOC, there is 1635% testing to code (TtC).

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

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

> ✓ **Answer:** 80%

0x Protocol has 80% code coverage with coveralls at https://coveralls.io/github/0xProject/0x-monorepo?branch=development.

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 improved by adding tests that achieve full code coverage. A clear report and scripts in the software repository will guarantee a high score.

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

> **Answer:** Yes

Scripts/Instructions location: https://github.com/0xProject/protocol/blob/development/README.md.

14) Report of the results (%)

> **Answer:** 70%

There is a GitHub code coverage report available at https://coveralls.io/builds/31242763.

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.

15) Formal Verification test done (%)

> **Answer:** 0%

There is no evidence that a 0x Protocol Formal Verification test has been performed.

16) Stress Testing environment (%)

Answer: 100%

There is evidence of 0x Protocol's testnet smart contract usage at https://0x.org/docs/guides/0x-cheat-sheet#mainnet-1-1.

Security

This section looks at the 3rd party software audits done. It is explained in this document. This section answers the following questions;

17) Did 3rd Party audits take place? (%)
18) Is the bounty value acceptably high?

17) Did 3rd Party audits take place? (%)

Answer: 100%

0x Protocol has had multiple audits performed on their various mainnet launches for their exchange and staking contracts. Multiple of these had been performed pre launch at https://protocol.0x.org/en/latest/additional/audits.html.

Guidance:

100% Multiple Audits performed before deployment and results public and implemented or not required
90% Single audit performed before deployment and results public and implemented or not required
70% Audit(s) performed after deployment and no changes required. Audit report is public

50% Audit(s) performed after deployment and changes needed but not implemented
20% No audit performed
0% Audit Performed after deployment, existence is public, report is not public and no improvements deployed OR smart contract address' not found, (where question 1 is 0%)

Deduct 25% if code is in a private repo and no note from auditors that audit is applicable to deployed code

18) Is the bounty value acceptably high (%)

Answer: 70%

0x Protocol's Bug Bounty program offers a maximum of 100k for the most critical of findings.

Guidance:

100% Bounty is 10% TVL or at least $1M AND active program (see below)
90% Bounty is 5% TVL or at least 500k AND active program
80% Bounty is 5% TVL or at least 500k
70% Bounty is 100k or over AND active program
60% Bounty is 100k or over
50% Bounty is 50k or over AND active program
40% Bounty is 50k or over
20% Bug bounty program bounty is less than 50k
0% No bug bounty program offered

An active program means that a third party (such as Immunefi) is actively driving hackers to the site. An inactive program would be static mentions on the docs.

Access Controls

This section covers the documentation of special access controls for a DeFi protocol. The admin access controls are the contracts that allow updating contracts or coefficients in the protocol. Since these contracts can allow the protocol admins to "change the rules", complete disclosure of capabilities is vital for user's transparency. It is explained in this document. The questions this section asks are as follow;

19) Can a user clearly and quickly find the status of the admin controls?
20) Is the information clear and complete?
21) Is the information in non-technical terms that pertain to the investments?
22) Is there Pause Control documentation including records of tests?

19) Can a user clearly and quickly find the status of the access controls (%)

> ✓ **Answer:** 100%

0x Protocol's access control information can easily be found in their "Core Concepts" section at https://0x.org/docs/core-concepts#a-non-custodial-exchange-protocol, as well as in their "Contracts" section at https://protocol.0x.org/en/latest/architecture/governor.html#managing-ownership.

Guidance:

100% Clearly labelled and on website, docs or repo, quick to find
70% Clearly labelled and on website, docs or repo but takes a bit of looking
40% Access control docs in multiple places and not well labelled
20% Access control docs in multiple places and not labelled
0% Admin Control information could not be found

20) Is the information clear and complete (%)

> **Answer:** 100%

All of the contracts that have access to user funds are immutable, and this is clearly explained at https://0x.org/docs/core-concepts#a-non-custodial-exchange-protocol.

Guidance:

All the contracts are immutable -- 100% OR

a) All contracts are clearly labelled as upgradeable (or not) -- 30% AND
b) The type of ownership is clearly indicated (OnlyOwner / MultiSig / Defined Roles) -- 30% AND
c) The capabilities for change in the contracts are described -- 30%

21) Is the information in non-technical terms that pertain to the investments (%)

> **Answer:** 100%

All of the contracts that have access to user funds are immutable, and this is clearly explained at https://0x.org/docs/core-concepts#a-non-custodial-exchange-protocol.

Guidance:

100% All the contracts are immutable
90% Description relates to investments safety and updates in clear, complete non-software l language
30% Description all in software specific language
0% No admin control information could not be found

22) Is there Pause Control documentation including records of tests (%)

> **Answer:** 80%

Pause Control is clearly explained at https://protocol.0x.org/en/latest/additional/emergency.html, but there is no evidence of regular tests.

Guidance:

100% All the contracts are immutable or no pause control needed and this is explained OR
100% Pause control(s) are clearly documented and there is records of at least one test within 3 months

80% Pause control(s) explained clearly but no evidence of regular tests

40%	Pause controls mentioned with no detail on capability or tests
0%	Pause control not documented or explained

How to improve this score:

Create a document that covers the items described above in plain language that investors can understand. An example is enclosed.

Appendices

Author Details

The author of this review is Rex of DeFi Safety.

Email : rex@defisafety.com Twitter : @defisafety

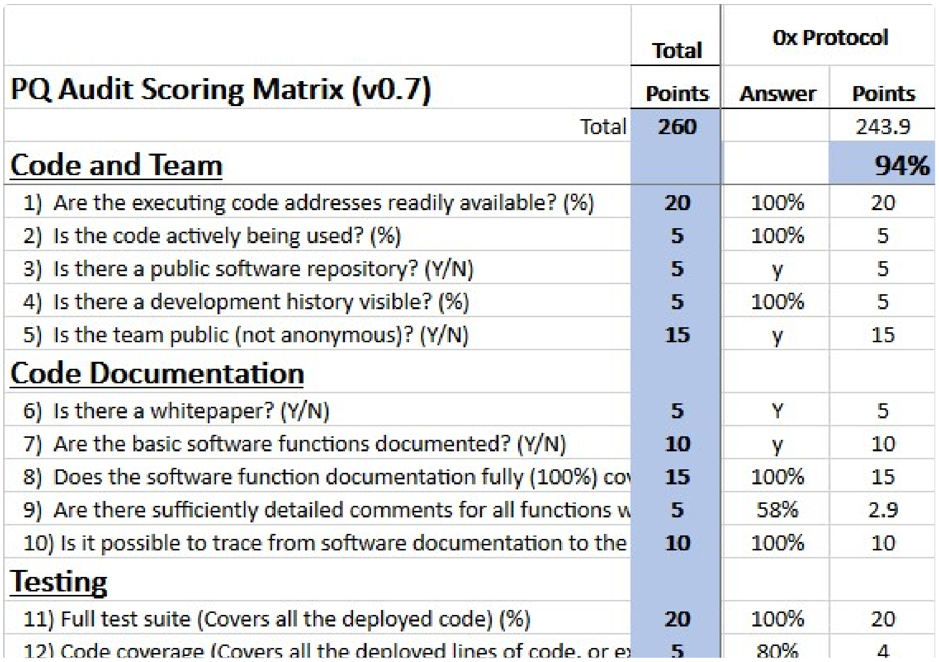
I started with Ethereum just before the DAO and that was a wonderful education. It showed the importance of code quality. The second Parity hack also showed the importance of good process. Here my aviation background offers some value. Aerospace knows how to make reliable code using quality processes.

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.

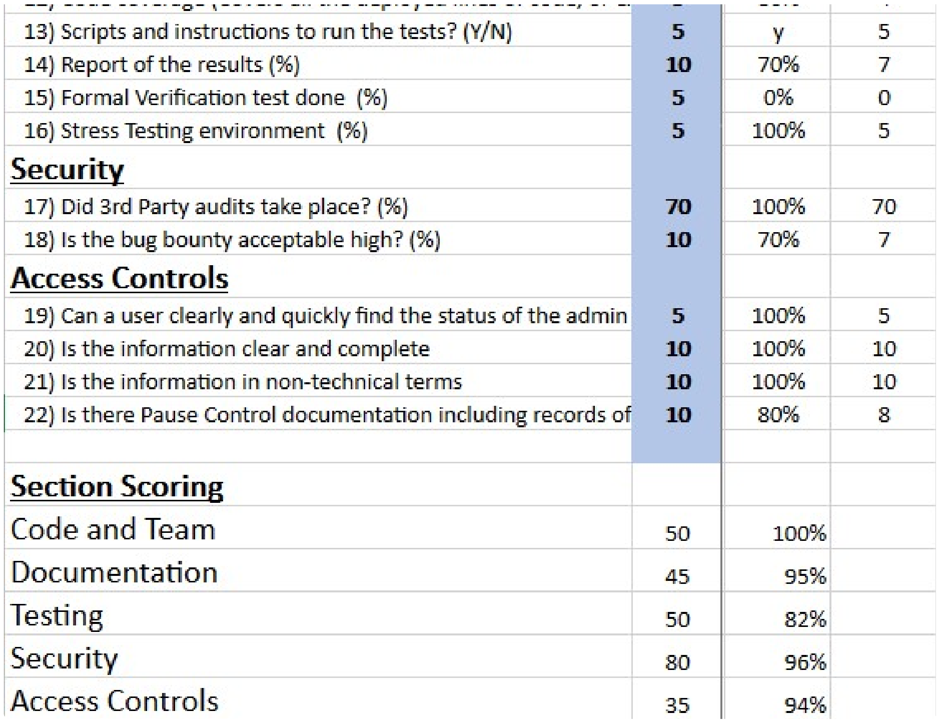
Process Quality Reviews are an extension of the SecurEth guidelines that will further increase the quality processes in Solidity and Vyper development.

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

Scoring Appendix

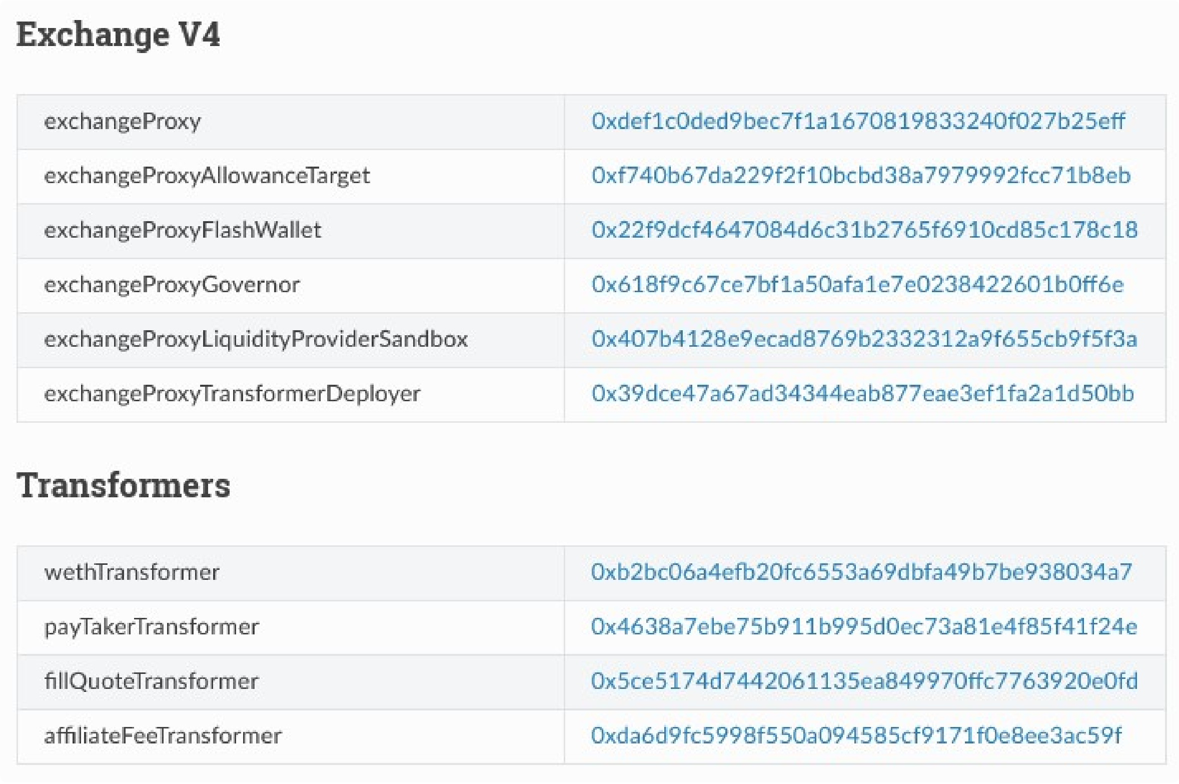


	Total	0x Protocol	
PQ Audit Scoring Matrix (v0.7)	Points	Answer	Points
Total	260		243.9
Code and Team			94%
1) Are the executing code addresses readily available? (%)	20	100%	20
2) Is the code actively being used? (%)	5	100%	5
3) Is there a public software repository? (Y/N)	5	y	5
4) Is there a development history visible? (%)	5	100%	5
5) Is the team public (not anonymous)? (Y/N)	15	y	15
Code Documentation			
6) Is there a whitepaper? (Y/N)	5	Y	5
7) Are the basic software functions documented? (Y/N)	10	y	10
8) Does the software function documentation fully (100%) cov	15	100%	15
9) Are there sufficiently detailed comments for all functions w	5	58%	2.9
10) Is it possible to trace from software documentation to the	10	100%	10
Testing			
11) Full test suite (Covers all the deployed code) (%)	20	100%	20
12) Code coverage (Covers all the deployed lines of code, or ex	5	80%	4



13) Scripts and instructions to run the tests? (Y/N)	**5**	y	5
14) Report of the results (%)	**10**	70%	7
15) Formal Verification test done (%)	**5**	0%	0
16) Stress Testing environment (%)	**5**	100%	5
Security			
17) Did 3rd Party audits take place? (%)	**70**	100%	70
18) Is the bug bounty acceptable high? (%)	**10**	70%	7
Access Controls			
19) Can a user clearly and quickly find the status of the admin	**5**	100%	5
20) Is the information clear and complete	**10**	100%	10
21) Is the information in non-technical terms	**10**	100%	10
22) Is there Pause Control documentation including records of	**10**	80%	8
Section Scoring			
Code and Team	50	100%	
Documentation	45	95%	
Testing	50	82%	
Security	80	96%	
Access Controls	35	94%	

Executing Code Appendix



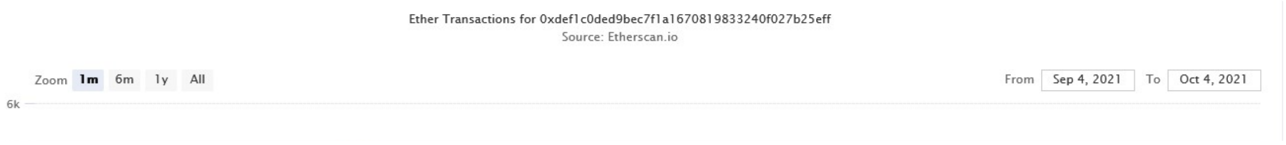
Exchange V4

exchangeProxy	0xdef1c0ded9bec7f1a1670819833240f027b25eff
exchangeProxyAllowanceTarget	0xf740b67da229f2f10bcbd38a7979992fcc71b8eb
exchangeProxyFlashWallet	0x22f9dcf4647084d6c31b2765f6910cd85c178c18
exchangeProxyGovernor	0x618f9c67ce7bf1a50afa1e7e0238422601b0ff6e
exchangeProxyLiquidityProviderSandbox	0x407b4128e9ecad8769b2332312a9f655cb9f5f3a
exchangeProxyTransformerDeployer	0x39dce47a67ad34344eab877eae3ef1fa2a1d50bb

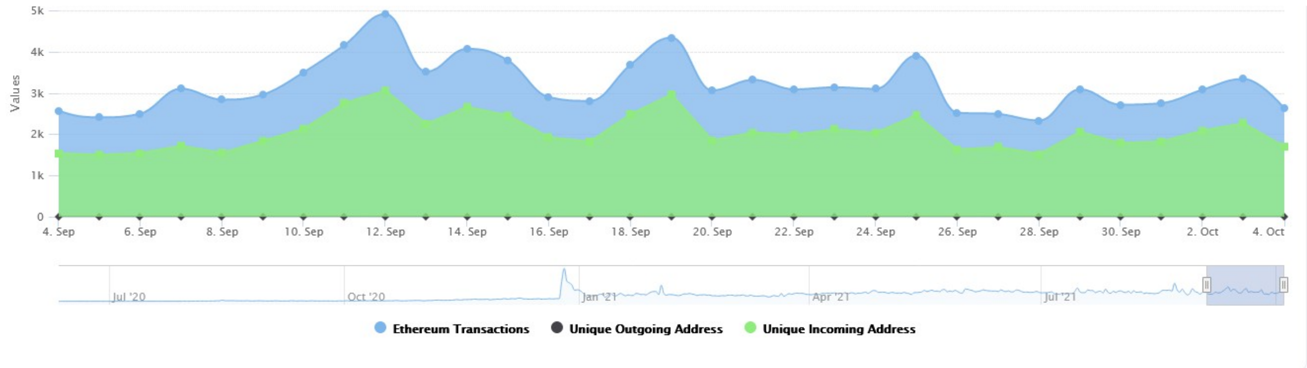
Transformers

wethTransformer	0xb2bc06a4efb20fc6553a69dbfa49b7be938034a7
payTakerTransformer	0x4638a7ebe75b911b995d0ec73a81e4f85f41f24e
fillQuoteTransformer	0x5ce5174d7442061135ea849970ffc7763920e0fd
affiliateFeeTransformer	0xda6d9fc5998f550a094585cf9171f0e8ee3ac59f

Code Used Appendix



Ether Transactions for 0xdef1c0ded9bec7f1a1670819833240f027b25eff
Source: Etherscan.io



Example Code Appendix

```
/// @dev An extensible proxy contract that serves as a universal entry point for
///      interacting with the 0x protocol.
contract ZeroEx {
    // solhint-disable separate-by-one-line-in-contract,indent,var-name-mixedcase
    using LibBytesV06 for bytes;

    /// @dev Construct this contract and register the `BootstrapFeature` feature.
    ///      After constructing this contract, `bootstrap()` should be called
    ///      by `bootstrap()` to seed the initial feature set.
    /// @param bootstrapper Who can call `bootstrap()`.
    constructor(address bootstrapper) public {
        // Temporarily create and register the bootstrap feature.
        // It will deregister itself after `bootstrap()` has been called.
        BootstrapFeature bootstrap = new BootstrapFeature(bootstrapper);
        LibProxyStorage.getStorage().impls[bootstrap.bootstrap.selector] =
            address(bootstrap);
    }

    // solhint-disable state-visibility

    /// @dev Forwards calls to the appropriate implementation contract.
    fallback() external payable {
        bytes4 selector = msg.data.readBytes4(0);
        address impl = getFunctionImplementation(selector);
        if (impl == address(0)) {
            _revertWithData(LibProxyRichErrors.NotImplementedError(selector));
        }

        (bool success, bytes memory resultData) = impl.delegatecall(msg.data);
        if (!success) {
            _revertWithData(resultData);
        }
        _returnWithData(resultData);
    }

    /// @dev Fallback for just receiving ether.
    receive() external payable {}

    // solhint-enable state-visibility

    /// @dev Get the implementation contract of a registered function.
    /// @param selector The function selector.
    /// @return impl The implementation contract address.
    function getFunctionImplementation(bytes4 selector)
        public
        view
        returns (address impl)
    {
        return LibProxyStorage.getStorage().impls[selector];
    }

    /// @dev Revert with arbitrary bytes.
    /// @param data Revert data.
    function _revertWithData(bytes memory data) private pure {
        assembly { revert(add(data, 32), mload(data)) }
    }

    /// @dev Return with arbitrary bytes.
    /// @param data Return data.
    function _returnWithData(bytes memory data) private pure {
        assembly { return(add(data, 32), mload(data)) }
    }
}
```

SLOC Appendix

Solidity Contracts

Language	Files	Lines	Blanks	Comments	Code	Complex
Solidity	11	1755	181	575	999	81

Comments to Code 575/999 = 58%

Javascript Tests

Language	Files	Lines	Blanks	Comments	Code	Complex
TypeScript	68	17814	1154	324	16336	569

Tests to Code 16336/999 = 1635%