0.7

dYdX Layer 2 (v3) 0.7 Process Quality Review UPDATE

Score: 91%

Overview

This is a dYdX Layer 2 (v3) Process Quality Review completed on August 11th 2021. A dYdX Layer 1 review was originally performed in December 2020, and can be found here. 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 **91%**, 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**
- **Here are the admin controls and strategies**

Disclaimer

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

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://github.com/dydxprotocol/solo/blob/master/migrations/deployed.json, as indicated in the Appendix.

Note: Although the contracts are in a .json file, there is a clearly labelled and quick-to-find link to it in their main GitHub repository.

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 well over 10 transactions a day on contract *SoloMargin.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/dydxprotocol/solo.

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 500 commits and 2 branches, this is a robust software repository.

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://www.linkedin.com/company/dydx.

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://github.com/dydxprotocol/solo/blob/master/README.md or https://legacy-docs.dydx.exchange/#introduction.

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

> **Answer:** Yes

All of the basic software functions are documented at https://docs.dydx.exchange/#general (Layer 2) and https://legacy-docs.dydx.exchange/#introduction (Layer 1).

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

> **Answer:** 100%

All contracts and functions are documented in the dYdX documentation. They rigorously document their API, Solo Protocol, and Perpetual Protocol in their Layer 1 documentation, and also detailed API and Perpetual Contract information in their Layer 2 documentation.

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:** 54%

Code examples are in the Appendix. As per the SLOC, there is 54% commenting to code (CtC).

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

Note: Only the core contract, interface, and trader files were used in our calculation. Any third party, library, or mock files were not taken into consideration during this process.

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 all throughout the dYdX documentation due to the fact that they give

side-by-side examples on how the described code is implemented into code. Examples of this can be seen all throughout the layer 2 and layer 1 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 331% 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:** 100%

The dYdX solo, perpetual, and protocol repositories all have 100% code coverage from codecov. These are the core and updated contracts.

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

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

> ✓ **Answer:** Yes

Scrips/Instructions location: https://github.com/dydxprotocol/solo/tree/master/scripts, instructions can be found at https://github.com/dydxprotocol/solo/blob/master/README.md.

14) Report of the results (%)

> ⓘ **Answer:** 70%

The dYdX coveralls code coverage reports can be found at https://coveralls.io/github/dydxprotocol.

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 of a dYdX Formal Verification test in any of their documentation or in further web research.

16) Stress Testing environment (%)

> **Answer:** 100%

All of the dYdX "42" contract addresses in the deployed.json are Kovan testnet addresses that are still regularly being used.

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%

dYdX were first audited pre-launch by Zeppelin Solutions and Bramah Systems.

They were also audited by PeckShield for their layer 2 launch.

All reports can be found here and here.

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:** 50%

The dYdX Bug Bounty program offers up to 50k for the most critical of bug finds.

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%

There is a clearly-labelled "Governance" section on the dYdX website. Once there, you have buttons that either take you to the forums or the documentation (question mark button).

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:** 90%

a) Contracts are clearly labelled as upgradeable through the voting processes described in "Voting & Governance".

b) Defined roles are detailed in "Architecture" "Voting Process", as well as "Proposal Lifecycle"

c) Capabilities for change in contract can be found in "Parameters".

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:** 90%

All governance and access control information relates to investment safety and is described in user-friendly language. In addition, there is also a "Technical Overview" section for developers.

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:** 40%

dYdX mention a similar function to Pause Control with their Merkle-pauser executor: "*The Merkle-pauser executor can execute proposals that freeze the Merkle root, which is updated periodically with each user's cumulative reward balance, allowing new rewards to be distributed to users over time, in case the proposed root is incorrect or malicious.*" This can be found here

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

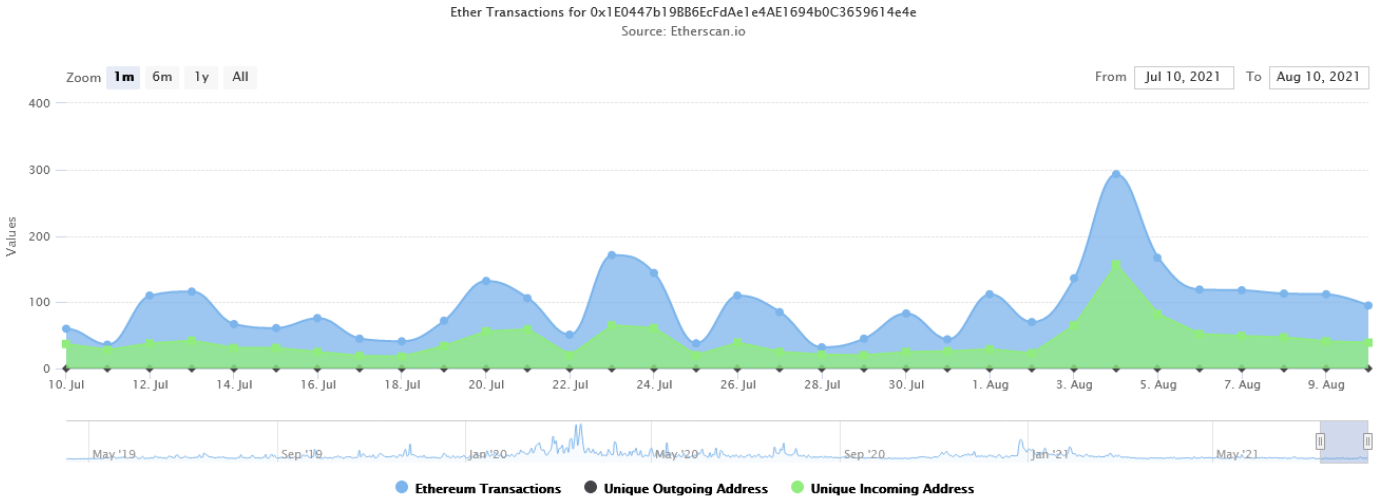
	Total	dYdX UPDATE	
PQ Audit Scoring Matrix (v0.7)	**Points**	**Answer**	**Points**
Total	**260**		236.7
Code and Team			**91%**
1) Are the executing code addresses readily available? (%)	**20**	100%	20
2) Is the code actively being used? (%)	**5**	100%	5
3) Is there a public software repository? (Y/N)	**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%) cover the deployed contracts? (%)	**15**	100%	15

9) Are there sufficiently detailed comments for all functions within the deployed contract code (%)	**5**	54%	2.7
10) Is it possible to trace from software documentation to the implementation in code (%)	**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 explains misses) (%)	**5**	100%	5
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**	50%	5
Access Controls			
19) Can a user clearly and quickly find the status of the admin controls	**5**	100%	5
20) Is the information clear and complete	**10**	90%	9
21) Is the information in non-technical terms	**10**	90%	9
22) Is there Pause Control documentation including records of tests	**10**	40%	4
Section Scoring			
Code and Team	50	100%	
Documentation	45	95%	
Testing	50	84%	
Security	80	94%	
Access Controls	35	77%	

Executing Code Appendix


```
"SoloMargin": {
    "1": {
        "links": {
            "AdminImpl": "0x8a6629fEba4196E0A61B8E8C94D4905e525bc055",
            "OperationImpl": "0x56E7d4520ABFECf10b38368b00723d9BD3c21ee1"
        },
        "address": "0x1E0447b19BB6EcFdAe1e4AE1694b0C3659614e4e",
        "transactionHash": "0x5d824b179e39313f45da503b1d75c1c7ce5287646a45ebe42431d50141fd451a"
    },
    "42": {
        "links": {
            "AdminImpl": "0x078D400C1a723b792de8afE240bf039b7069ac39",
            "OperationImpl": "0xcfa0451D8D1F08504Fa44a6f72D00F455b858a1d"
        },
        "address": "0x4EC3570cADaAEE08Ae384779B0f3A45EF85289DE",
        "transactionHash": "0xb949f10eabe6b7e66c7304a3e9d300375638e82932c5c1e723da3530ace277e3"
    }
},
"PayableProxyForSoloMargin": {
    "1": {
        "links": {},
        "address": "0xa8b39829cE2246f89B31C013b8Cde15506Fb9A76",
        "transactionHash": "0x5463436a4480bf13d75a7fea9055b7c616bd15cc5873a068ec489e738fb4cd0f"
    },
    "42": {
        "links": {},
        "address": "0x2b005Af087bb14e3887a8572807095404796105A",
        "transactionHash": "0x02038a963724d897729f3f2962cbaf0302f64311b4ce342d1598998befc0d717"
```

Code Used Appendix



Example Code Appendix

```
/**
 * @title Admin
 * @author dYdX
 *
 * Public functions that allow the privileged owner address to manage Solo
 */
contract Admin is
    State,
    Ownable,
    ReentrancyGuard
{
    // ============ Token Functions ============

    /**
     * Withdraw an ERC20 token for which there is an associated market. Only excess tokens
     * withdrawn. The number of excess tokens is calculated by taking the current number o
     * held in Solo, adding the number of tokens owed to Solo by borrowers, and subtractin
     * number of tokens owed to suppliers by Solo.
     */
    function ownerWithdrawExcessTokens(
        uint256 marketId,
        address recipient
    )
        public
        onlyOwner
        nonReentrant
        returns (uint256)
    {
        return AdminImpl.ownerWithdrawExcessTokens(
            g_state,
```

```
            marketId,
            recipient
        );
    }

    /**
     * Withdraw an ERC20 token for which there is no associated market.
     */
    function ownerWithdrawUnsupportedTokens(
        address token,
        address recipient
    )
        public
        onlyOwner
        nonReentrant
        returns (uint256)
    {
        return AdminImpl.ownerWithdrawUnsupportedTokens(
            g_state,
            token,
            recipient
        );
    }

    // ============ Market Functions ============

    /**
     * Add a new market to Solo. Must be for a previously-unsupported ERC20 token.
     */
    function ownerAddMarket(
        address token,
        IPriceOracle priceOracle,
        IInterestSetter interestSetter,
        Decimal.D256 memory marginPremium,
        Decimal.D256 memory spreadPremium
    )
        public
        onlyOwner
        nonReentrant
    {
        AdminImpl.ownerAddMarket(
            g_state,
            token,
            priceOracle,
            interestSetter,
            marginPremium,
            spreadPremium
        );
    }

    /**
     * Set (or unset) the status of a market to "closing". The borrowedValue of a market ca
     * increase while its status is "closing".
```

```
     */
    function ownerSetIsClosing(
        uint256 marketId,
        bool isClosing
    )
        public
        onlyOwner
        nonReentrant
    {
        AdminImpl.ownerSetIsClosing(
            g_state,
            marketId,
            isClosing
        );
    }

    /**
     * Set the price oracle for a market.
     */
    function ownerSetPriceOracle(
        uint256 marketId,
        IPriceOracle priceOracle
    )
        public
        onlyOwner
        nonReentrant
    {
        AdminImpl.ownerSetPriceOracle(
            g_state,
            marketId,
            priceOracle
        );
    }

    /**
     * Set the interest-setter for a market.
     */
    function ownerSetInterestSetter(
        uint256 marketId,
        IInterestSetter interestSetter
    )
        public
        onlyOwner
        nonReentrant
    {
        AdminImpl.ownerSetInterestSetter(
            g_state,
            marketId,
            interestSetter
        );
    }

    /**
```

```
137      * Set a premium on the minimum margin-ratio for a market. This makes it so that any po
138      * that include this market require a higher collateralization to avoid being liquidate
139      */
140     function ownerSetMarginPremium(
141         uint256 marketId,
142         Decimal.D256 memory marginPremium
143     )
144         public
145         onlyOwner
146         nonReentrant
147     {
148         AdminImpl.ownerSetMarginPremium(
149             g_state,
150             marketId,
151             marginPremium
152         );
153     }
154
155     /**
156      * Set a premium on the liquidation spread for a market. This makes it so that any liqu
157      * that include this market have a higher spread than the global default.
158      */
159     function ownerSetSpreadPremium(
160         uint256 marketId,
161         Decimal.D256 memory spreadPremium
162     )
163         public
164         onlyOwner
165         nonReentrant
166     {
167         AdminImpl.ownerSetSpreadPremium(
168             g_state,
169             marketId,
170             spreadPremium
171         );
172     }
```

SLOC Appendix

Solidity Contracts

Language	Files	Lines	Blanks	Comments	Code	Complex
Solidity	24	4534	486	1415	2633	146

Comments to Code 1415/2633 = 54%

Javascript Tests

Language	Files	Lines	Blanks	Comments	Code	Complex
JavaScript	49	11029	1350	956	8723	399

Tests to Code 8723/2633 = 331%