0.7

Akropolis Process Quality Review

Score: 76%

Overview

This is an Akropolis Process Quality Review completed on September 20th 2021. It was performed using the Process Review process (version 0.7.3) and is documented here. The review was performed by Nick of DeFiSafety. Check out our Telegram.

The final score of the review is **76%**, 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

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

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 in the README.md of the Delphi GitHub repository at https://github.com/akropolisio/delphi, 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, including internal, on contract *AdminUpgradeabilityProxy.sol* (Staking Contract), 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/akropolisio

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%

This protocol has a rich development history, with some 515 commits and 68 separate branches.

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://github.com/akropolisio/akropolis/graphs/contributors

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://whitepaper.io/document/634/akropolis-whitepaper

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

> **Answer:** Yes

This whitepaper details the basic software functions.

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

>

Answer: 100%

The software documentation is impressive and covers all deployed contracts.

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: 59%

Code examples are in the Appendix. As per the SLOC, there is 59% 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: 60%

Akropolis' code is clearly outlined in the docs and can be non explicitly traced in their github repository for each of their individual projects.

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:** 80%

Code examples are in the Appendix. As per the SLOC, there is 93% 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

How to improve this score:

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

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

> **Answer:** 50%

Documented code coverage could not be found, but there is clear evidence of robust testing in the TtC score.

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

Scrips/Instructions location: https://github.com/akropolisio/delphi/tree/release-1.0/scripts.

14) Report of the results (%)

> ⚠ **Answer:** 0%

No test report was found.

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%

No formal verification was found.

16) Stress Testing environment (%)

> ✓ **Answer:** 100%

Akropolis has been deployed in full on the Rinkeby testnet.

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%

Certik published two Akropolis Sparta audit reports before the mainnet launch.

MixBytes published a Delphi staking audit report after the mainnet launch.

MythX published a Delphi staking report after the mainnet launch.

Notes On Audit Results:

The MythX audit underlined countless low-severity issues in the Akropolis' Delphi smart contracts. However, none of them identify if they have been corrected or not, which is concerning.

All of the fixes underline in both Certik reports have been mitigated or completely fixed.

The MyxBytes audit underlined several medium and major issues. However, all of them have since been fixed by the Akropolis team.

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:** 20%

The bounty program offered by Akropolis has a maximum reward of $40,000.

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:** 0%

Governance information was found at https://wiki.akropolis.io/sparta/#akro-governance, however there are no details explicitly detailing what the admins have control over.

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:** 60%

a) Smart contracts are clearly labelled as upgradeable through the voting process.

b) Defined roles (Borrower, and LP) are outlined.

~~c) What can be changed in contracts is described, but not to what extent.~~

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%

How to improve this score:

Create a document that covers the items described above. An example is enclosed.

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

> ⚠ **Answer:** 30%

Governance information is written in a technical way that does not communicate to users about how/why their funds are safe.

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

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.

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

> ⚠ **Answer:** 0%

There is no pause control documentation.

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	Akropolis UPDATE	
PQ Audit Scoring Matrix (v0.7)			

PQ Audit Scoring Matrix (v0.7)	Points	Answer	Points
Total	260		198.45
Code and Team			76%
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	59%	2.95
10) Is it possible to trace from software documentation to the	10	60%	6
Testing			
11) Full test suite (Covers all the deployed code) (%)	20	80%	16
12) Code coverage (Covers all the deployed lines of code, or ex	5	50%	2.5
13) Scripts and instructions to run the tests? (Y/N)	5	Y	5
14) Report of the results (%)	10	0%	0
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	20%	2
Access Controls			
19) Can a user clearly and quickly find the status of the admin	5	0%	0
20) Is the information clear and complete	10	60%	6
21) Is the information in non-technical terms	10	30%	3
22) Is there Pause Control documentation including records of	10	0%	0
Section Scoring			
Code and Team	50	100%	
Documentation	45	87%	
Testing	50	57%	
Security	80	90%	
Access Controls	35	26%	

Executing Code Appendix

- CurveFiStablecoinStrategy `0x72e8F9aa2fa78Ce2eF7cbEc97cB5c8E696Ebe593`
- PoolToken for Vault CurveFi `0xD28a298fDe6Bb995A2a01293866916989e48507D`

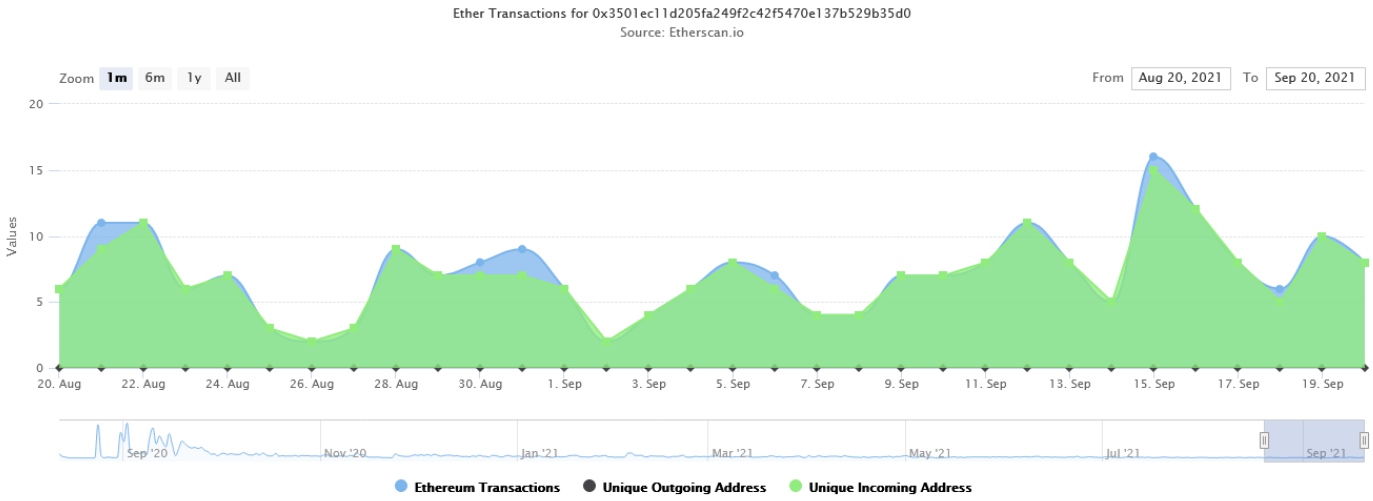
Rewards

- RewardVestingModule `0x2A9dcb9d79Aba0CC64565A87c9d20D11D1f33a07`
- RewardDistributionModule `0x84056675382c851cf42FAAFCeC3FCa90E21AE645`

Savings & Investing

- SavingsModule `0x73fC3038B4cD8FfD07482b92a52Ea806505e5748`
- InvestingModule `0xF311b1258d0F245b85090e4Fb01f2277cB2328aD`
- StakingPool `0x3501Ec11d205fa249f2C42f5470e137b529b35D0`
- StakingPoolAdel `0x1A547c3dd03c39Fb2b5aEaFC524033879bD28F13`

Code Used Appendix



Example Code Appendix

```
pragma solidity ^0.5.12;

import "../common/Base.sol";
import "../interfaces/core/CoreInterface.sol";
import "../utils/AddressMap.sol";

contract Pool is Base, CoreInterface {

    /* Short description */
    string  public name;
    string  public description;
    address public founder;

```

```
    /* Modules map */
    AddressMap.Data modules;

    using AddressList for AddressList.Data;
    using AddressMap for AddressMap.Data;

    /* Module constant mapping */
    mapping(bytes32 => bool) public is_constant;

    /**
     * @dev Contract ABI storage
     *      the contract interface contains source URI
     */
    mapping(address => string) public abiOf;

    function initialize() public initializer {
        Base.initialize();
        founder = _msgSender();
    }

    function setMetadata(string memory _name, string  memory _description) public onlyOwne
        name = _name;
        description = _description;
    }

    /**
     * @dev Set new module for given name
     * @param _name infrastructure node name
     * @param _module infrastructure node address
     * @param _constant have a `true` value when you create permanent name of module
     */
    function set(string memory _name, address _module, bool _constant) public onlyOwner {

        require(!isConstant(_name), "Pool: module address can not be replaced");

        // Notify
        if (modules.get(_name) != ZERO_ADDRESS)
            emit ModuleReplaced(_name, modules.get(_name), _module);
        else
            emit ModuleAdded(_name, _module);

        // Set module in the map
        modules.set(_name, _module);

        // Register constant flag
        is_constant[keccak256(abi.encodePacked(_name))] = _constant;
    }

     /**
     * @dev Remove module by name
     * @param _name module name
     */
    function remove(string memory _name)  public onlyOwner {
```

```
67          require(!isConstant(_name), "Pool: module can not be removed");
68
69          // Notify
70          emit ModuleRemoved(_name, modules.get(_name));
71
72          // Remove module
73          modules.remove(_name);
74      }
75
76      /**
77       * @dev Fast module exist check
78       * @param _module is a module address
79       * @return `true` wnen core contains module
80       */
81      function contains(address _module) public view returns (bool)
82      {
83          return modules.items.contains(_module);
84      }
85
86      /**
87       * @dev Modules counter
88       * @return count of modules in core
89       */
90      function size() public view returns (uint)
91      {
92          return modules.size();
93      }
94
95      /**
96       * @dev Check for module have permanent name
97       * @param _name is a module name
98       * @return `true` when module have permanent name
99       */
100     function isConstant(string memory _name) public view returns (bool)
101     {
102         return is_constant[keccak256(abi.encodePacked(_name))];
103     }
104
105     /**
106      * @dev Get module by name
107      * @param _name is module name
108      * @return module address
109      */
110     function get(string memory _name) public view returns (address)
111     {
112         return modules.get(_name);
113     }
114
115     /**
116      * @dev Get module name by address
117      * @param _module is a module address
118      * @return module name
119      */
```

```
120     function getName(address _module) public view returns (string memory)
121     {
122         return modules.keyOf[_module];
123     }
124 
125     /**
126      * @dev Get first module
127      * @return first address
128      */
129     function first() public view returns (address)
130     {
131         return modules.items.head;
132     }
133 
134     /**
135      * @dev Get next module
136      * @param _current is an current address
137      * @return next address
138      */
139     function next(address _current) public view returns (address)
140     {
141         return modules.items.next(_current);
142     }
143 
144 }
```

SLOC Appendix

Solidity Contracts

Language	Files	Lines	Blanks	Comments	Code	Complex
Solidity	12	41	420	846	1441	181

Comments to Code 846/1441 = 59%

Javascript Tests

Language	Files	Lines	Blanks	Comments	Code	Complex
JavaScript	5	1552	167	38	1347	5

Tests to Code 1347/1441 = 93%