# Trust Token

## Lending Platform

### Security Assessment

May 10th, 2021

[Preliminary Report]

# Disclaimer

CertiK reports are not, nor should be considered, an "endorsement" or "disapproval" of any particular project or team. These reports are not, nor should be considered, an indication of the economics or value of any "product" or "asset" created by any team or project that contracts CertiK to perform a security review.

CertiK Reports do not provide any warranty or guarantee regarding the absolute bug-free nature of the technology analyzed, nor do they provide any indication of the technologies proprietors, business, business model or legal compliance.

CertiK Reports should not be used in any way to make decisions around investment or involvement with any particular project. These reports in no way provide investment advice, nor should be leveraged as investment advice of any sort.

CertiK Reports represent an extensive auditing process intending to help our customers increase the quality of their code while reducing the high level of risk presented by cryptographic tokens and blockchain technology.

Blockchain technology and cryptographic assets present a high level of ongoing risk. CertiK's position is that each company and individual are responsible for their own due diligence and continuous security. CertiK's goal is to help reduce the attack vectors and the high level of variance associated with utilizing new and consistently changing technologies, and in no way claims any guarantee of security or functionality of the technology we agree to analyze.

## What is a CertiK report?

- A document describing in detail an in depth analysis of a particular piece(s) of source code provided to CertiK by a Client.
- An organized collection of testing results, analysis and inferences made about the structure, implementation and overall best practices of a particular piece of source code.
- Representation that a Client of CertiK has completed a round of auditing with the intention to increase the quality of the company/product's IT infrastructure and or source code.

# Overview

## Project Summary

| | |
|---|---|
| **Project Name** | Trust Token - Lending Platform |
| **Description** | A lending platform implementation with enhanced features. |
| **Platform** | Ethereum; Solidity, Yul |
| **Codebase** | GitHub Repository |
| **Commits** | 1. a96a83e6fd8511f9aad748a4a5194685a27d06f3 |

## Audit Summary

| | |
|---|---|
| **Delivery Date** | May 10th, 2021 |
| **Method of Audit** | Static Analysis, Manual Review |
| **Consultants Engaged** | 2 |
| **Timeline** | April 19th, 2021 - May 10th, 2021 |

## Vulnerability Summary

| | |
|---|---|
| **Total Issues** | 21 |
| **Total Critical** | 0 |
| **Total Major** | 1 |
| **Total Medium** | 2 |
| **Total Minor** | 2 |
| **Total Informational** | 16 |

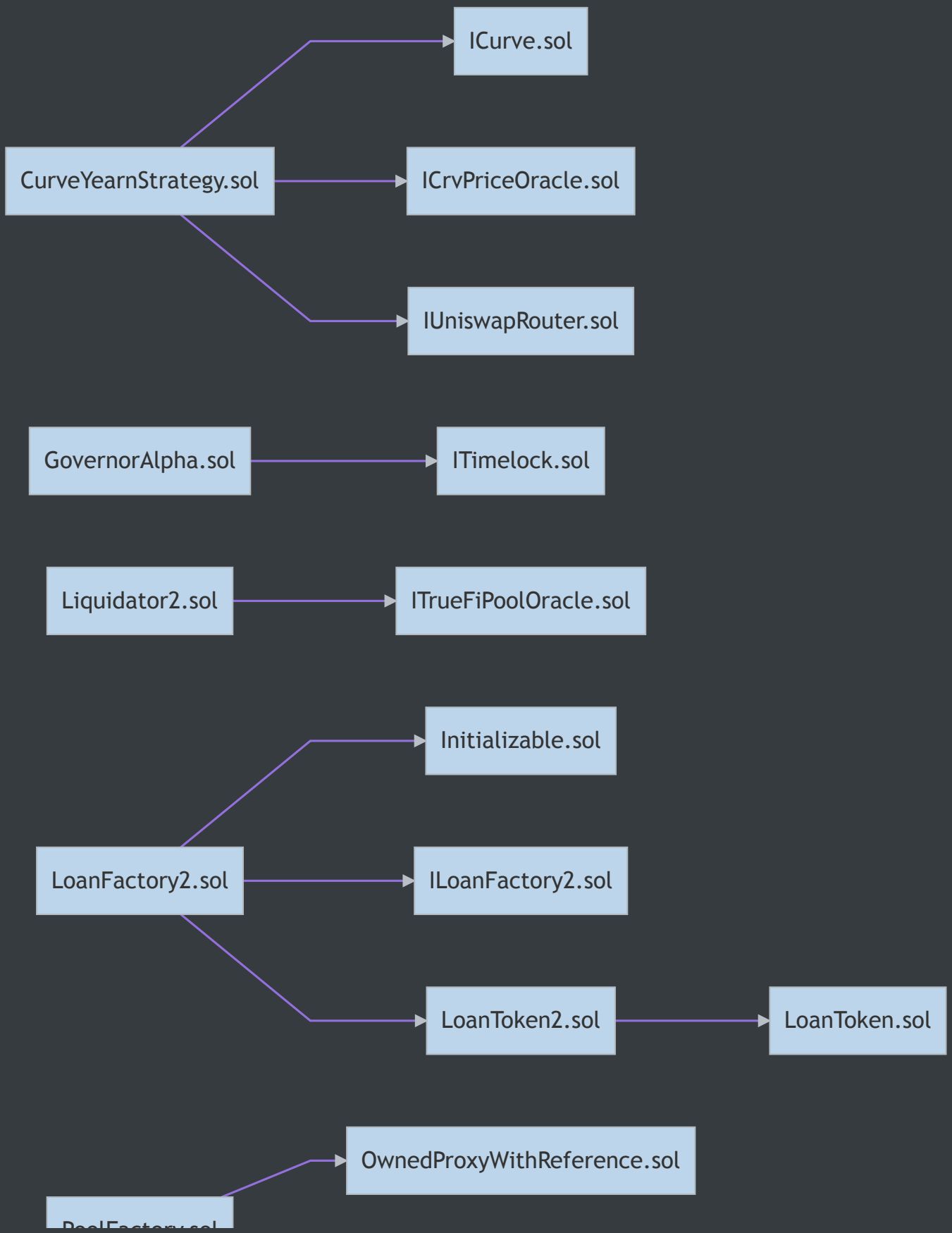
# Executive Summary

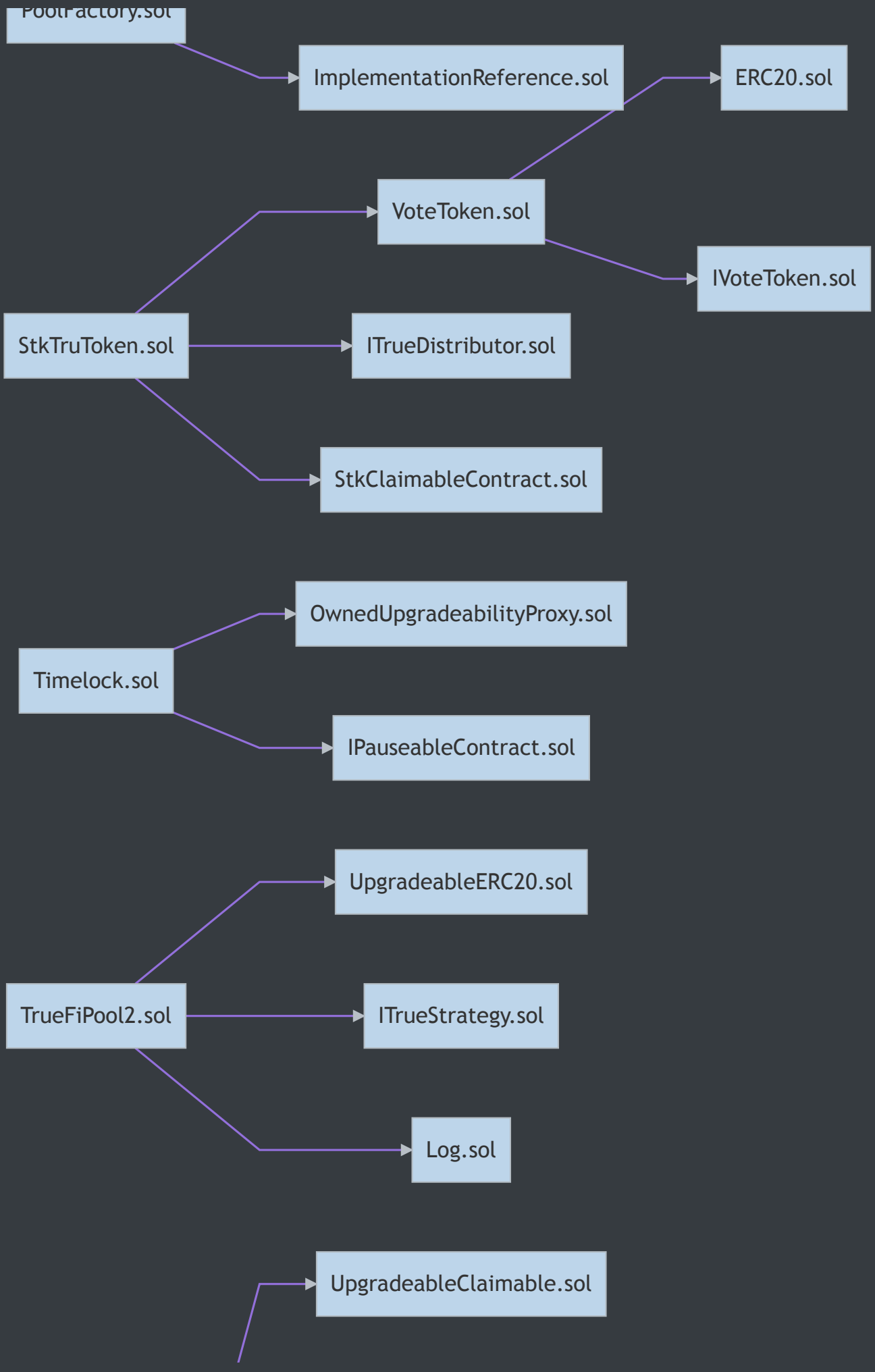
This section will represent the summary of the whole audit process once it has concluded.

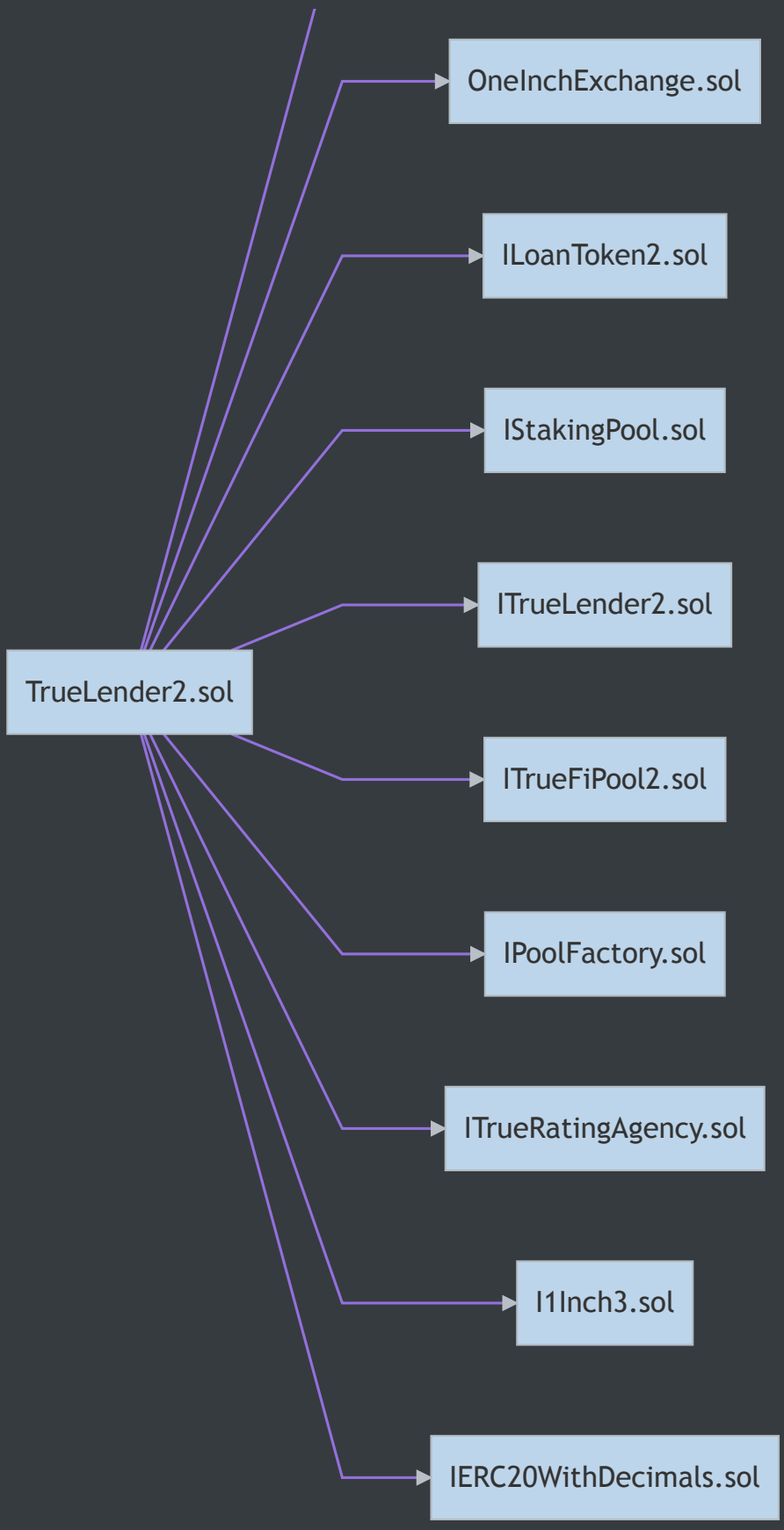
# Files In Scope

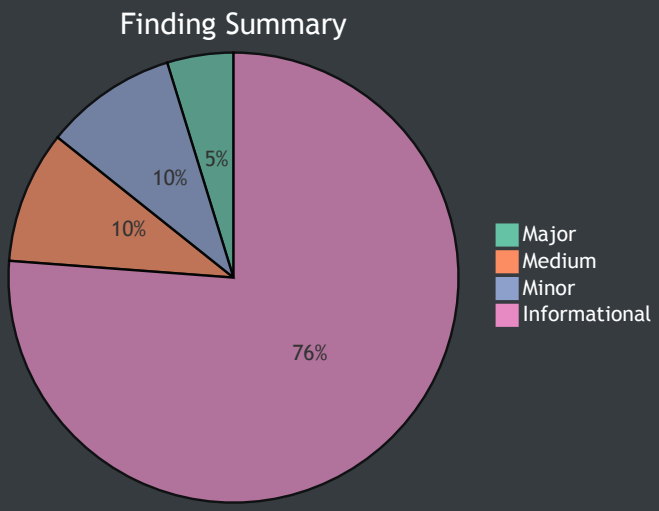
| ID | Contract | Location |
|---|---|---|
| GAA | GovernorAlpha.sol | contracts/governance/GovernorAlpha.sol |
| STT | StkTruToken.sol | contracts/governance/StkTruToken.sol |
| TIM | Timelock.sol | contracts/governance/Timelock.sol |
| VTN | VoteToken.sol | contracts/governance/VoteToken.sol |
| LIQ | Liquidator2.sol | contracts/truefi2/Liquidator2.sol |
| LF2 | LoanFactory2.sol | contracts/truefi2/LoanFactory2.sol |
| LT2 | LoanToken2.sol | contracts/truefi2/LoanToken2.sol |
| PFY | PoolFactory.sol | contracts/truefi2/PoolFactory.sol |
| TFP | TrueFiPool2.sol | contracts/truefi2/TrueFiPool2.sol |
| TL2 | TrueLender2.sol | contracts/truefi2/TrueLender2.sol |
| CYS | CurveYearnStrategy.sol | contracts/truefi2/strategies/CurveYearnStrategy.sol |

# File Dependency Graph




PoolFactory.sol
ImplementationReference.sol
ERC20.sol
VoteToken.sol
IVoteToken.sol
StkTruToken.sol
ITrueDistributor.sol
StkClaimableContract.sol
OwnedUpgradeabilityProxy.sol
Timelock.sol
IPauseableContract.sol
UpgradeableERC20.sol
TrueFiPool2.sol
ITrueStrategy.sol
Log.sol
UpgradeableClaimable.sol


OneInchExchange.sol
ILoanToken2.sol
IStakingPool.sol
ITrueLender2.sol
TrueLender2.sol
ITrueFiPool2.sol
IPoolFactory.sol
ITrueRatingAgency.sol
I1Inch3.sol
IERC20WithDecimals.sol


Finding Summary
5%
10%
10%
76%
Major
Medium
Minor
Informational

# Manual Review Findings

| ID | Title | Type | Severity | Resolved |
|---|---|---|---|---|
| GAA-01 | Redundant Check Against Constant Value | Language Specific | Informational | ⓘ |
| STT-01 | Use of non safe transfer | Logical Issue | Minor | ⓘ |
| STT-02 | Visibility Specifiers Missing | Language Specific | Informational | ⓘ |
| STT-03 | Return Variable Declaration | Coding Style | Informational | ⓘ |
| STT-04 | Require With No Error Message | Language Specific | Informational | ⓘ |
| LT2-01 | Constant Variable Naming | Language Specific | Informational | ⓘ |
| LT2-02 | Wrong Comment | Logical Issue | Informational | ⓘ |
| LT2-03 | Magic Number | Coding Style | Informational | ⓘ |
| LT2-04 | Redundant Calculation | Coding Style | Informational | ⓘ |
| PFY-01 | Missing Zero Address Check | Language Specific | Informational | ⓘ |
| TFP-01 | Invalid Check | Language Specific | Major | ⓘ |
| TFP-02 | Inefficient Pattern | Logical Issue | Minor | ⓘ |
| TFP-03 | Magic Numbers | Coding Style | Informational | ⓘ |
| TFP-04 | Missing Zero Address Check | Language Specific | Informational | ⓘ |
| TFP-05 | Proper Representation | Language Specific | Informational | ⓘ |

| | | | | |
|---|---|---|---|---|
| TFP-06 | Missing Zero Address Check | Language Specific | Informational | |
| TL2-01 | Missing Check | Logical Issue | Medium | |
| TL2-02 | Code Design | Logical Issue | Medium | |
| TL2-03 | Visibility Specifiers Missing | Language Specific | Informational | |
| TL2-04 | Return Variable Declaration | Coding Style | Informational | |
| CYS-01 | Visibility Specifiers Missing | Language Specific | Informational | |

## GAA-01: Redundant Check Against Constant Value

| Type | Severity | Location |
| --- | --- | --- |
| Language Specific | Informational | GovernorAlpha.sol L361 |

### Description:

The code contains a check expressed against a constant value. `x == false` while x is a bool.

### Recommendation:

Consider removing the constant right part.

## STT-01: Use of non safe transfer

| Type | Severity | Location |
| --- | --- | --- |
| Logical Issue | Minor | StkTruToken.sol L478 |

### Description:

The code uses transfer while SafeERC20 is available.

### Recommendation:

Consider using the safe functionality from the SafeERC20 lib.

## STT-02: Visibility Specifiers Missing

| Type | Severity | Location |
| --- | --- | --- |
| Language Specific | Informational | StkTruToken.sol L25, L26, L58 |

### Description:

The linked variable declarations do not have a visibility specifier explicitly set.

### Recommendation:

Inconsistencies in the default visibility the Solidity compilers impose can cause issues in the functionality of the codebase. We advise that visibility specifiers for the linked variables are explicitly set.

## STT-03: Return Variable Declaration

| Type | Severity | Location |
| --- | --- | --- |
| Coding Style | Informational | StkTruToken.sol L594 |

### Description:

The linked function declarations contain explicitly named `return` variables that are degrading the readability of the code.

### Recommendation:

We advise that the linked variables are omitted from the declaration and introduced inside the functions scope.

## STT-04: Require With No Error Message

| Type | Severity | Location |
| --- | --- | --- |
| Language Specific | Informational | StkTruToken.sol L264, L334, L478 |

### Description:

The code contains require checks with no error messages.

### Recommendation:

Consider adding error messages.

## LT2-01: Constant Variable Naming

| Type | Severity | Location |
| --- | --- | --- |
| Language Specific | Informational | LoanToken2.sol L37 |

### Description:

The linked variable does not have a proper ALL_CAPS name deviating from solidity's standards.

### Recommendation:

Consider renaming the variable with ALL_CAPS.

## LT2-02: Wrong Comment

| Type | Severity | Location |
| --- | --- | --- |
| Logical Issue | Informational | LoanToken2.sol L434 |

### Description:

The function interest has commenting that does not proper represent the functionality.

### Recommendation:

Consider refactoring the comment.

## LT2-03: Magic Number

| Type | Severity | Location |
| --- | --- | --- |
| Coding Style | Informational | LoanToken2.sol L268, L439 |

### Description:

The code represents constantly a value of 10000.

### Recommendation:

Consider adding a immutable constant.

## LT2-04: Redundant Calculation

| Type | Severity | Location |
| --- | --- | --- |
| Coding Style | Informational | LoanToken2.sol L268 |

### Description:

The code redundant calculates interest variable while there is a interest function.

### Recommendation:

Consider using the interest function for the calculation.

## PFY-01: Missing Zero Address Check

| Type | Severity | Location |
| --- | --- | --- |
| Language Specific | Informational | PoolFactory.sol L139 |

### Description:

The function setTrueLender does not check against a zero address.

### Recommendation:

Consider implementing a check.

## TFP-01: Invalid Check

| Type | Severity | Location |
| --- | --- | --- |
| Language Specific | Major | TrueFiPool2.sol L336, L378 |

### Description:

The code contains a check against the `tx.origin`.

### Recommendation:

Consider refactoring the code taking under consideration the upcoming hard fork of ethereum and the changes on `tx.origin`.

## TFP-02: Inefficient Pattern

| Type | Severity | Location |
| --- | --- | --- |
| Logical Issue | Minor | TrueFiPool2.sol L557 |

### Description:

The code performs a redundant allocation to check and return the same value.

### Recommendation:

Consider checking the value directly returning early and allocating later if the check was successful.

## TFP-03: Magic Numbers

| Type | Severity | Location |
| --- | --- | --- |
| Coding Style | Informational | TrueFiPool2.sol L313, L332, L382, L406, L423, L430 |

### Description:

The code represents constantly values that could be declared as constants.

### Recommendation:

Consider adding a immutable constant representation for those values.

## TFP-04: Missing Zero Address Check

| Type | Severity | Location |
| --- | --- | --- |
| Language Specific | Informational | TrueFiPool2.sol L322 |

### Description:

The function setBeneficiary does not check against a zero address.

### Recommendation:

Consider implementing a check.

## TFP-05: Proper Representation

| Type | Severity | Location |
| --- | --- | --- |
| Language Specific | Informational | TrueFiPool2.sol L414 |

### Description:

The code represents a formula but deviates from the usage of the api.

### Recommendation:

Consider refactoring the code TODO.

## TFP-06: Missing Zero Address Check

| Type | Severity | Location |
| --- | --- | --- |
| Language Specific | Informational | TrueFiPool2.sol L525 |

### Description:

The function setOracle does not check against a zero address.

### Recommendation:

Consider implementing a check.

## TL2-01: Missing Check

| Type | Severity | Location |
| --- | --- | --- |
| Logical Issue | Medium | TrueLender2.sol L189 |

### Description:

The function setLoansLimit does not check if the newLoansLimit is not bigger that the previous limit.

### Recommendation:

Consider implementing a check to ensure newLoansLimit > oldLoansLimit.

## TL2-02: Code Design

| Type | Severity | Location |
| --- | --- | --- |
| Logical Issue | Medium | TrueLender2.sol L386 |

### Description:

The function distribute does not check if msg.sender is the intended one and additionally calls functionality that is commented as helper test function in L423.

### Recommendation:

Consider providing a rationale.

## TL2-03: Visibility Specifiers Missing

| Type | Severity | Location |
| --- | --- | --- |
| Language Specific | Informational | TrueLender2.sol L41 |

### Description:

The linked variable declarations do not have a visibility specifier explicitly set.

### Recommendation:

Inconsistencies in the default visibility the Solidity compilers impose can cause issues in the functionality of the codebase. We advise that visibility specifiers for the linked variables are explicitly set.

## TL2-04: Return Variable Declaration

| Type | Severity | Location |
| --- | --- | --- |
| Coding Style | Informational | TrueLender2.sol L222, L310, L339 |

### Description:

The linked function declarations contain explicitly named `return` variables that are degrading the readability of the code.

### Recommendation:

We advise that the linked variables are omitted from the declaration and introduced inside the functions scope.

## CYS-01: Visibility Specifiers Missing

| Type | Severity | Location |
| --- | --- | --- |
| Language Specific | Informational | CurveYearnStrategy.sol L29, L31 |

### Description:

The linked variable declarations do not have a visibility specifier explicitly set.

### Recommendation:

Inconsistencies in the default visibility the Solidity compilers impose can cause issues in the functionality of the codebase. We advise that visibility specifiers for the linked variables are explicitly set.

# Appendix

## Finding Categories

### Logical Issue

Logical Issue findings are exhibits that detail a fault in the logic of the linked code, such as an incorrect notion on how `block.timestamp` works.

### Language Specific

Language Specific findings are issues that would only arise within Solidity, i.e. incorrect usage of `private` or `delete`.

### Coding Style

Coding Style findings usually do not affect the generated byte-code and comment on how to make the codebase more legible and as a result easily maintainable.