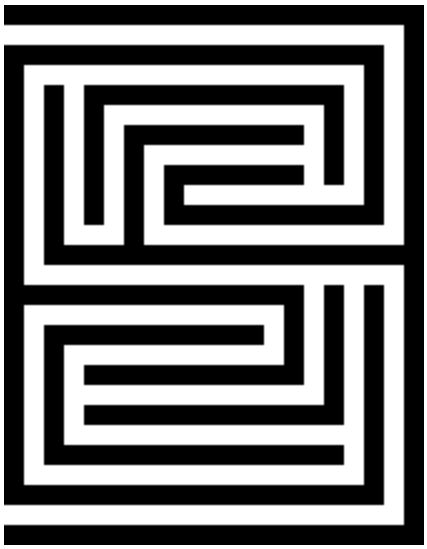

# Coin Token Audit Public Report

## PROJECT: Coin Token Audit

## December 2020


**Prepared For:**

Coin Team | Coin XYZ, Inc.

https://coindefi.org/

**Prepared By:**

Jonathan Haas | Bramah Systems, LLC.

jonathan@bramah.systems

# Table of Contents

# Coin Token Review

## Executive Summary

### Scope of Engagement

Bramah Systems, LLC was engaged in December of 2020 to perform a comprehensive security review of the Coin token smart contracts (specific contracts denoted within the appendix). Our review was conducted over a period of one day by both members of the Bramah Systems, LLC. executive staff.

Bramah Systems completed the assessment using manual, static and dynamic analysis techniques.

### Timeline

Review Commencement: December 28th, 2020

Report Delivery: December 29th, 2020

### Engagement Goals

The primary scope of the engagement was to evaluate and establish the overall security of the Coin protocol, with a specific focus on trading actions. In specific, the engagement sought to answer the following questions:

- Is it possible for an attacker to steal or freeze tokens?
- Does the Solidity code match the specification as provided?
- Is there a way to interfere with the contract mechanisms?
- Are the arithmetic calculations trustworthy?

### Contract Specification

Contract specification was provided in the form of code comments and functional unit tests. The contract is modeled after the ERC865 specification, a not yet standard EIP that allows for paying for transfers with tokens, instead of gas, within a singular transaction.

## Overall Assessment

Bramah Systems was engaged to evaluate and identify any potential security concerns within the codebase of the Coin Protocol. During the course of our engagement, Bramah Systems found relatively few instances wherein the team deviated materially from established best practices and procedures of secure software development within DLT, as our report details.

These aside, the team otherwise used **thoroughly** reviewed and vetted components and provided details as to the token structure, economics, and intent, which helped Bramah highlight any potential concerns with their approach.

# Disclaimer

As of the date of publication, the information provided in this report reflects the presently held, commercially reasonable understanding of Bramah Systems, LLC.'s knowledge of security patterns as they relate to the Coin Protocol, with the understanding that distributed ledger technologies ("DLT") remain under frequent and continual development, and resultantly carry with them unknown technical risks and flaws. The scope of the review provided herein is limited solely to items denoted within "Scope of Engagement" and contained within "Directory Structure". The report does NOT cover, review, or opine upon security considerations unique to the Solidity compiler, tools used in the development of the protocol, or distributed ledger technologies themselves, or to any other matters not specifically covered in this report.

The contents of this report must NOT be construed as investment advice or advice of any other kind. This report does NOT have any bearing upon the potential economics of the Coin protocol or any other relevant product, service or asset of Coin or otherwise. This report is not and should not be relied upon by Coin or any reader of this report as any form of financial, tax, legal, regulatory, or other advice.

To the full extent permissible by applicable law, Bramah Systems, LLC. disclaims all warranties, express or implied. The information in this report is provided "as is" without warranty, representation, or guarantee of any kind, including the accuracy of the information provided. Bramah Systems, LLC. makes no warranties, representations, or guarantees about the Coin Protocol. Use of this report and/or any of the information provided herein is at the users sole risk, and Bramah Systems, LLC. hereby disclaims, and each user of this report hereby waives, releases, and holds Bramah Systems, LLC. harmless from, any and all liability, damage, expense, or harm (actual, threatened, or claimed) from such use.

# Timeliness of Content

All content within this report is presented only as of the date published or indicated, to the commercially reasonable knowledge of Bramah Systems, LLC. as of such date, and may be superseded by subsequent events or for other reasons. The content contained within this report is subject to change without notice. Bramah Systems, LLC. does not guarantee or warrant the accuracy or timeliness of any of the content contained within this report, whether accessed through digital means or otherwise.

Bramah Systems, LLC. is not responsible for setting individual browser cache settings nor can it ensure any parties beyond those individuals directly listed within this report are receiving the most recent content as reasonably understood by Bramah Systems, LLC. as of the date this report is provided to such individuals.

# General Recommendations

## Best Practices & Solidity Development Guidelines

### Usage of deprecated constructors

The contract makes usage of multiple deprecated constructors, mostly notably the usage of "**throw**". While throw was more widely accepted in usage at the time of publication of the initial Coin token contract, even at this time it's usage was deprecated. Usage of assert, revert, or require is suggested. As pointed out within documentation by Consensys Diligence, the logical constraints of a **if...throw** statement can still be accomplished -- as presented with the following example: **if(msg.sender != owner) { throw; }**

- **if(msg.sender != owner) { revert(); }**
- **assert(msg.sender == owner);**
- **require(msg.sender == owner);**

**Resolution**: The team has chosen to keep the contracts in their current state for these items.

### Solidity version should be updated

The bulk of the protocol uses **pragma version^0.424**. This version of Solidity was released on May 16, 2018 and does not include many compiler optimizations and potential security considerations of later Solidity versions, and should be updated where possible.

**Resolution**: The team has chosen to keep the contracts in their current state for these items.

### Usage of tx.origin instead of msg.sender

The usage of **tx.origin** is heavily recommended against, due to a number of limitations it

places, most notable of which is that it does not allow for the **owner** to ever be a contract. Various members of the Ethereum foundation, including Vitalik, have explicitly suggested avoiding its usage.

**Resolution**: The team has chosen to keep the contracts in their current state for these items.

## Avoid hardcoded gas amounts

The usage of hardcoded magic numbers for gas amounts is heavily suggested against. As gas values can change over time for certain operations, and gas costs vary heavily, usage of hardcoded gas amounts is placed hand-in-hand with suggested discontinuation of **transfer** usage.

**Resolution**: The team has chosen to keep the contracts in their current state for these items.

## Compiler version not fixed

The pragma indicated within **CoinvestToken.sol** allows for compilation by version **0.4.24** of the Solidity compiler and above. This should be fixed (e.g. the **^** should be removed) to a specific version to prevent unanticipated changes in the Solidity language.

**Resolution**: The team has chosen to keep the contracts in their current state for these items.

## Sensitive changes should emit an event

Multiple sensitive changes (namely, transference of tokens in compliance with the ERC20 specification and removing tokens incorrectly sent to the smart contract address) should emit an event to indicate the action has occurred -- and to generally provide more robust audit logging. While not explicitly necessary, these audit logs allow for easier root cause analysis in the instance that an anomalous event occurs.

**Resolution**: The team has chosen to keep the contracts in their current state for these items.

## Usage of send and transfer considered against best-practice

Following EIP-1884, the usage of transfer and send is no longer suggested, due to changing gas costs in their usage. Use .call.value(...)("") instead.

**Resolution**: The team has chosen to keep the contracts in their current state for these items.

## Solidity style guide not followed

The token presently does not follow the Solidity style guide. Two areas in particular, function visibility and ordering along with capitalizing constants, present the most concern.

Functions should be grouped according to their visibility and ordered as follows:

- constructor
- fallback function (if exists)
- external
- public
- internal
- private

Constants, per present style-guide recommendation, should use capitalized snake-case.

**Occurrences**:

Numerous

**Resolution**: The team has chosen to keep the contracts in their current state for these items.

# Specific Recommendations

## Unique to the Coin Protocol

While a number of considerations do exist as to how Coin may bolster the security of their ERC20 token, during the course of our audit we did not encounter any issues that would be specific to the Coin token itself, but rather would impact any smart contract developed

similarly.

# Toolset Warnings

## Unique to the Coin Protocol

### Overview

In addition to our manual review, our process involves utilizing static analysis and formal methods in order to perform additional verification of the presence of security vulnerabilities (or lack thereof). An additional part of this review phase consists of reviewing any automated unit testing frameworks that exist.

The following sections detail warnings generated by the automated tools and confirmation of false positives where applicable.

### Compilation Warnings

Our review, at request of Coin, covers the Solidity code (*.sol) as of commit **c8329b3f6501dff9f60fb9c16aa38741da5184fa** of the Coin token repository.

### Test Coverage

The contract repository features basic unit tests provided in the form of a TypeScript file that validates various functional stages of the smart contract.

### Static Analysis Coverage

The contract repository underwent heavy scrutiny with multiple static analysis agents, including:

- Securify
- MAIAN
- Mythril
- Oyente
- Slither

In each case, the team had either mitigated relevant concerns raised by each of these tools or

provided adequate justification for the risk (such as adhering to the ERC-20 token standard).

## Surya Coverage Report

| Contract | Type | Bases | | |
|---|---|---|---|---|
| L | Function Name | Visibility | Mutability | Modifiers |
| ApproveAndCallFallBack | Implementation | | | |
| L | receiveApproval | Public ❗ | 🛑 | NO ❗ |
| CoinToken | Implementation | Ownable | | |
| L | | Public ❗ | 🛑 | NO ❗ |
| L | receiveApproval | Public ❗ | 🛑 | NO ❗ |
| L | transfer | Public ❗ | 🛑 | NO ❗ |
| L | transferFrom | Public ❗ | 🛑 | NO ❗ |
| L | approve | Public ❗ | 🛑 | NO ❗ |
| L | increaseApproval | Public ❗ | 🛑 | NO ❗ |

| | | | | |
|---|---|---|---|---|
| └ | decreaseApproval | Public ❗ | 🛑 | NO ❗ |
| └ | approveAndCall | Public ❗ | 🛑 | NO ❗ |
| └ | _transfer | Internal 🔒 | 🛑 | |
| └ | _approve | Internal 🔒 | 🛑 | |
| └ | _increaseApproval | Internal 🔒 | 🛑 | |
| └ | _decreaseApproval | Internal 🔒 | 🛑 | |
| └ | transferPreSigned | Public ❗ | 🛑 | NO ❗ |
| └ | approvePreSigned | Public ❗ | 🛑 | NO ❗ |
| └ | increaseApprovalPreSigned | Public ❗ | 🛑 | NO ❗ |
| └ | decreaseApprovalPreSigned | Public ❗ | 🛑 | NO ❗ |
| └ | approveAndCallPreSigned | Public ❗ | 🛑 | NO ❗ |
| └ | revokeHash | Public ❗ | 🛑 | NO ❗ |
| └ | revokeHashPreSigned | Public ❗ | 🛑 | NO ❗ |

| | | | | |
|---|---|---|---|---|
| └ | getRevokeHash | Public ❗ | | NO ❗ |
| └ | recoverRevokeHash | Public ❗ | | NO ❗ |
| └ | getPreSignedHash | Public ❗ | | NO ❗ |
| └ | recoverPreSigned | Public ❗ | | NO ❗ |
| └ | getSignHash | Public ❗ | | NO ❗ |
| └ | ecrecoverFromSig | Public ❗ | | NO ❗ |
| └ | totalSupply | External ❗ | | NO ❗ |
| └ | balanceOf | External ❗ | | NO ❗ |
| └ | allowance | External ❗ | | NO ❗ |
| └ | tokenEscape | External ❗ | 🛑 | onlyOwner |

| | | | | |
|---|---|---|---|---|
| SafeMathLib | Library | | | |
| └ | mul | Internal 🔒 | | |

| | | | | |
|---|---|---|---|---|
| └ | div | Internal 🔒 | | |
| └ | sub | Internal 🔒 | | |
| └ | add | Internal 🔒 | | |

| Ownable | Implementation | | | |
|---|---|---|---|---|
| └ | | Public ❗ | 🛑 | NO ❗ |
| └ | transferOwnership | Public ❗ | 🛑 | onlyOwner |
| └ | transferCoinvest | External ❗ | 🛑 | onlyCoinvest |
| └ | alterAdmin | External ❗ | 🛑 | onlyCoinvest |

| Bank | Implementation | Ownable | | |
|---|---|---|---|---|
| └ | | Public ❗ | 🛑 | NO ❗ |
| └ | transfer | External ❗ | 🛑 | NO ❗ |

| | | | | |
|---|---|---|---|---|
| └ | changeInvestment | External ❗ | 🛑 | onlyOwner |
| └ | tokenEscape | External ❗ | 🛑 | onlyCoinvest |

| ERC20Interface | Implementation | | | |
|---|---|---|---|---|
| └ | totalSupply | Public ❗ | | NO ❗ |
| └ | balanceOf | Public ❗ | | NO ❗ |
| └ | allowance | Public ❗ | | NO ❗ |
| └ | transfer | Public ❗ | 🛑 | NO ❗ |
| └ | approve | Public ❗ | 🛑 | NO ❗ |
| └ | transferFrom | Public ❗ | 🛑 | NO ❗ |

| PostAuditTest | Implementation | Ownable | | |
|---|---|---|---|---|
| └ | | Public ❗ | 💵 | NO ❗ |
| └ | | External ❗ | 💵 | onlyAdmin |
| └ | receiveApproval | Public ❗ | 🛑 | NO ❗ |

| | | | | |
|---|---|---|---|---|
| └ | buy | Public ❗ | 💵 | notPaused onlySenderOrToken |
| └ | sell | Public ❗ | 💵 | notPaused onlySenderOrToken |
| └ | finalizeBuy | Internal 🔒 | 🛑 | |
| └ | finalizeSell | Internal 🔒 | 🛑 | |
| └ | addCrypto | Public ❗ | 🛑 | onlyOwner |
| └ | addTrades | External ❗ | 🛑 | onlyCoinvest |
| └ | alterPause | External ❗ | 🛑 | onlyOwner |
| └ | changeContracts | External ❗ | 🛑 | onlyOwner |
| └ | changeUrls | External ❗ | 🛑 | onlyOwner |
| └ | getPrices | Public ❗ | 🛑 | NO ❗ |

| | | | | |
|---|---|---|---|---|
| └ | __callback | Public ❗ | 🛑 | NO ❗ |
| └ | craftUrl | Public ❗ | | NO ❗ |
| └ | decodePrices | Public ❗ | | NO ❗ |
| └ | calculateValue | Public ❗ | | NO ❗ |
| └ | bitConv | Internal 🔒 | | |
| └ | bitRec | Public ❗ | | NO ❗ |
| └ | changeGas | External ❗ | 🛑 | onlyOwner |
| └ | tokenEscape | External ❗ | 🛑 | onlyCoinvest |
| └ | parseInt | Internal 🔒 | | |
| └ | parseInt | Internal 🔒 | | |

| | | | | |
|---|---|---|---|---|
| strings | Library | | | |
| └ | memcpy | Private 🔐 | | |

| | | | | |
|---|---|---|---|---|
| └ | toSlice | Internal 🔒 | | |
| └ | len | Internal 🔒 | | |
| └ | toSliceB32 | Internal 🔒 | | |
| └ | copy | Internal 🔒 | | |
| └ | toString | Internal 🔒 | | |
| └ | len | Internal 🔒 | | |
| └ | empty | Internal 🔒 | | |
| └ | compare | Internal 🔒 | | |
| └ | equals | Internal 🔒 | | |
| └ | nextRune | Internal 🔒 | | |
| └ | nextRune | Internal 🔒 | | |

| | | | | |
|---|---|---|---|---|
| └ | ord | Internal 🔒 | | |
| └ | keccak | Internal 🔒 | | |
| └ | startsWith | Internal 🔒 | | |
| └ | beyond | Internal 🔒 | | |
| └ | endsWith | Internal 🔒 | | |
| └ | until | Internal 🔒 | | |
| └ | findPtr | Private 🔐 | | |
| └ | rfindPtr | Private 🔐 | | |
| └ | find | Internal 🔒 | | |
| └ | rfind | Internal 🔒 | | |
| └ | split | Internal 🔒 | | |

| | | | | |
|---|---|---|---|---|
| └ | split | Internal 🔒 | | |
| └ | rsplit | Internal 🔒 | | |
| └ | rsplit | Internal 🔒 | | |
| └ | count | Internal 🔒 | | |
| └ | contains | Internal 🔒 | | |
| └ | concat | Internal 🔒 | | |
| └ | join | Internal 🔒 | | |

| UserData | Implementation | Ownable | | |
|---|---|---|---|---|
| └ | | Public ❗ | 🛑 | NO ❗ |
| └ | modifyHoldings | External ❗ | 🛑 | NO ❗ |
| └ | returnHoldings | External ❗ | | NO ❗ |

| | | | | |
|---|---|---|---|---|
| └ | changeInvestment | External ❗ | 🛑 | onlyOwner |
| └ | tokenEscape | External ❗ | 🛑 | onlyCoinvest |

| | | | | |
|---|---|---|---|---|
| ApproveAndCallFallBack | Implementation | | | |
| └ | receiveApproval | Public ❗ | 🛑 | NO ❗ |

| | | | | |
|---|---|---|---|---|
| CashToken | Implementation | Ownable | | |
| └ | | Public ❗ | 🛑 | NO ❗ |
| └ | receiveApproval | Public ❗ | 🛑 | NO ❗ |
| └ | transfer | Public ❗ | 🛑 | NO ❗ |
| └ | transferFrom | Public ❗ | 🛑 | NO ❗ |
| └ | approve | Public ❗ | 🛑 | NO ❗ |
| └ | increaseApproval | Public ❗ | 🛑 | NO ❗ |
| └ | decreaseApproval | Public ❗ | 🛑 | NO ❗ |
| └ | approveAndCall | Public ❗ | 🛑 | NO ❗ |

| | | | | |
|---|---|---|---|---|
| L | _transfer | Internal 🔒 | 🛑 | |
| L | _approve | Internal 🔒 | 🛑 | |
| L | _increaseApproval | Internal 🔒 | 🛑 | |
| L | _decreaseApproval | Internal 🔒 | 🛑 | |
| L | transferPreSigned | Public ❗ | 🛑 | NO ❗ |
| L | approvePreSigned | Public ❗ | 🛑 | NO ❗ |
| L | increaseApprovalPreSigned | Public ❗ | 🛑 | NO ❗ |
| L | decreaseApprovalPreSigned | Public ❗ | 🛑 | NO ❗ |
| L | approveAndCallPreSigned | Public ❗ | 🛑 | NO ❗ |
| L | revokeHash | Public ❗ | 🛑 | NO ❗ |
| L | revokeHashPreSigned | Public ❗ | 🛑 | NO ❗ |
| L | getRevokeHash | Public ❗ | | NO ❗ |
| L | recoverRevokeHash | Public ❗ | | NO ❗ |

| | | | | |
|---|---|---|---|---|
| └ | getPreSignedHash | Public ❗ | | NO ❗ |
| └ | recoverPreSigned | Public ❗ | | NO ❗ |
| └ | getSignHash | Public ❗ | | NO ❗ |
| └ | ecrecoverFromSig | Public ❗ | | NO ❗ |
| └ | getNonce | External ❗ | | NO ❗ |
| └ | totalSupply | External ❗ | | NO ❗ |
| └ | balanceOf | External ❗ | | NO ❗ |
| └ | allowance | External ❗ | | NO ❗ |
| └ | mint | External ❗ | 🛑 | onlyOwner |
| └ | burn | External ❗ | 🛑 | onlyOwner |
| └ | tokenEscape | External ❗ | 🛑 | onlyOwner |

| | | | | |
|---|---|---|---|---|
| TokenSwap | Implementation | Ownable | | |

| | | | | |
|---|---|---|---|---|
| └ | | Public ❗ | 🛑 | NO ❗ |
| └ | tokenFallback | External ❗ | 🛑 | NO ❗ |
| └ | receiveApproval | Public ❗ | 🛑 | NO ❗ |
| └ | tokenEscape | External ❗ | 🛑 | onlyCoinvest |

| ERC20Interface | Implementation | | | |
|---|---|---|---|---|
| └ | totalSupply | Public ❗ | | NO ❗ |
| └ | balanceOf | Public ❗ | | NO ❗ |
| └ | allowance | Public ❗ | | NO ❗ |
| └ | transfer | Public ❗ | 🛑 | NO ❗ |
| └ | approve | Public ❗ | 🛑 | NO ❗ |
| └ | transferFrom | Public ❗ | 🛑 | NO ❗ |

| TestApproveAndCallFallBack | Implementation | ERC20Interface | | |
|---|---|---|---|---|
| └ | receiveApproval | Public ❗ | 🛑 | NO ❗ |

| Investment | Implementation | Ownable, usingOraclize | | |
|---|---|---|---|---|
| └ | | Public ❗ | 💵 | NO ❗ |
| └ | | External ❗ | 💵 | onlyAdmin |
| └ | receiveApproval | Public ❗ | 🛑 | NO ❗ |
| └ | buy | Public ❗ | 💵 | notPaused onlySenderOrToken |
| └ | sell | Public ❗ | 💵 | notPaused onlySenderOrToken |
| └ | finalizeBuy | Internal 🔒 | 🛑 | |
| └ | finalizeSell | Internal 🔒 | 🛑 | |
| └ | addCrypto | Public ❗ | 🛑 | onlyOwner |

| | | | | |
|---|---|---|---|---|
| └ | addTrades | External ❗ | 🛑 | onlyAdmin |
| └ | changeVars | External ❗ | 🛑 | onlyOwner |
| └ | changeGas | External ❗ | 🛑 | onlyAdmin |
| └ | getPrices | Internal 🔒 | 🛑 | |
| └ | __callback | Public ❗ | 🛑 | NO ❗ |
| └ | craftUrl | Public ❗ | | NO ❗ |
| └ | tokenEscape | External ❗ | 🛑 | coinvestOrOwner |

| | | | | |
|---|---|---|---|---|
| InvestLib | Library | | | |
| └ | calculateValue | Public ❗ | | NO ❗ |
| └ | bitConv | Internal 🔒 | | |
| └ | bitRec | Public ❗ | | NO ❗ |
| └ | decodePrices | Public ❗ | | NO ❗ |

| | | | | |
|---|---|---|---|---|
| L | parseInt | Internal 🔒 | 🛑 | |
| L | parseInt | Internal 🔒 | 🛑 | |

**Legend**

| Symbol | Meaning |
|---|---|
| 🛑 | Function can modify state |
| 💵 | Function is payable |

# Directory Structure

At time of review, the directory structure of the Coin smart contracts repository appeared as it does below. Our review, at request of Coin, covers the Solidity code (*.sol) as of commit hash **c8329b3f6501dff9f60fb9c16aa38741da5184fa.**

```
├── Bank.sol
├── CashToken.sol
├── Coinvest223.sol
├── CoinvestToken.sol
├── ERC20Interface.sol
├── Investment.sol
├── Ownable.sol
├── PostAuditTest.sol
├── SafeMathLib.sol
├── Strings.sol
├── TestApproveAndCall.sol
├── TokenSwap.sol
```

```
└─── UserData.sol

0 directories, 13 files
```