



Security Assessment

# MultiChain Foundation - Aptos

CertiK Verified on Nov 30th, 2022





CertiK Verified on Nov 30th, 2022

## MultiChain Foundation - Aptos

The security assessment was prepared by CertiK, the leader in Web3.0 security.

### Executive Summary

#### TYPES

Cross-Chain Bridge

#### ECOSYSTEM

Aptos

#### METHODS

Manual Review, Static Analysis

#### LANGUAGE

Move

#### TIMELINE

Delivered on 11/30/2022

#### KEY COMPONENTS

N/A

#### CODEBASE

<https://github.com/anyswap/aptos-contract>[...View All](#)

#### COMMITTS

- 735bc5199169266c0ac3a9ad5c18fe1f82d063f1
- 3f1d93077d528d34f0bb5de716fd526805b9ac29

[...View All](#)

### Vulnerability Summary



6

Total Findings

4

Resolved

1

Mitigated

0

Partially Resolved

1

Acknowledged

0

Declined

0

Unresolved



0

Critical

Critical risks are those that impact the safe functioning of a platform and must be addressed before launch. Users should not invest in any project with outstanding critical risks.



0

Major

Major risks can include centralization issues and logical errors. Under specific circumstances, these major risks can lead to loss of funds and/or control of the project.



0

Medium

Medium risks may not pose a direct risk to users' funds, but they can affect the overall functioning of a platform.



2

Minor

2 Resolved



Minor risks can be any of the above, but on a smaller scale. They generally do not compromise the overall integrity of the project, but they may be less efficient than other solutions.



4

Informational

2 Resolved, 1 Mitigated, 1 Acknowledged



Informational errors are often recommendations to improve the style of the code or certain operations to fall within industry best practices. They usually do not affect the overall functioning of the code.

# TABLE OF CONTENTS | MULTICHAIN FOUNDATION - APTOS

## I **Summary**

[Executive Summary](#)

[Vulnerability Summary](#)

[Codebase](#)

[Audit Scope](#)

[Approach & Methods](#)

## I **Review Notes**

[Overview](#)

[Router](#)

[USDC](#)

[Resource and Account Relationship](#)

[External Dependencies](#)

[Privileged Roles](#)

## I **Findings**

[POO-01 : Entry Functions Should Not Have Return Values](#)

[POO-02 : Entry Functions Should Not Have Struct Inputs](#)

[GLOBAL-01 : Dependency on Key Management](#)

[735-01 : Potential Failure on Coin Distribution](#)

[POL-01 : Missing Validation on Underlying Coin Existence](#)

[USD-01 : Unused Variables](#)

## I **Appendix**

## I **Disclaimer**

# CODEBASE | MULTICHAIN FOUNDATION - APTOS

## Repository




<https://github.com/anyswap/aptos-contract>

## Commit

- 735bc5199169266c0ac3a9ad5c18fe1f82d063f1
- 3f1d93077d528d34f0bb5de716fd526805b9ac29

# AUDIT SCOPE | MULTICHAIN FOUNDATION - APTOS

3 files audited ● 1 file with Acknowledged findings ● 2 files with Resolved findings

ID	File	SHA256 Checksum
● USD	 mintable/sources/USDC.move	3faa46b4ed23cf467d1be855a4331cf0020b7b14edab34ffe40c6f023858c0a8
● POO	 router/sources/Pool.move	b023b8b5ee0558b3f174a1dadf9ee4d6445becd4323cc00c0c78cfdbb014f8e9
● ROU	 router/sources/Router.move	e9c026e168fceab68d655392611efd90407b9bdc402c64db3f0da31b593c3418

## APPROACH & METHODS | MULTICHAIN FOUNDATION - APTOS

This report has been prepared for MultiChain to discover issues and vulnerabilities in the source code of the MultiChain Foundation - Aptos project as well as any contract dependencies that were not part of an officially recognized library. A comprehensive examination has been performed, utilizing Manual Review and Static Analysis techniques.

The auditing process pays special attention to the following considerations:

- Testing the smart contracts against both common and uncommon attack vectors.
- Assessing the codebase to ensure compliance with current best practices and industry standards.
- Ensuring contract logic meets the specifications and intentions of the client.
- Cross referencing contract structure and implementation against similar smart contracts produced by industry leaders.
- Thorough line-by-line manual review of the entire codebase by industry experts.

The security assessment resulted in findings that ranged from critical to informational. We recommend addressing these findings to ensure a high level of security standards and industry practices. We suggest recommendations that could better serve the project from the security perspective:

- Testing the smart contracts against both common and uncommon attack vectors;
- Enhance general coding practices for better structures of source codes;
- Add enough unit tests to cover the possible use cases;
- Provide more comments per each function for readability, especially contracts that are verified in public;
- Provide more transparency on privileged activities once the protocol is live.

# REVIEW NOTES | MULTICHAIN FOUNDATION - APTOS

## Overview

**MultiChain** is a cross-chain router protocol that allows users to transfer tokens between various blockchains. This audit is primarily focused on the bridge to the Aptos ecosystem.

The project consists of the following modules:

## Router

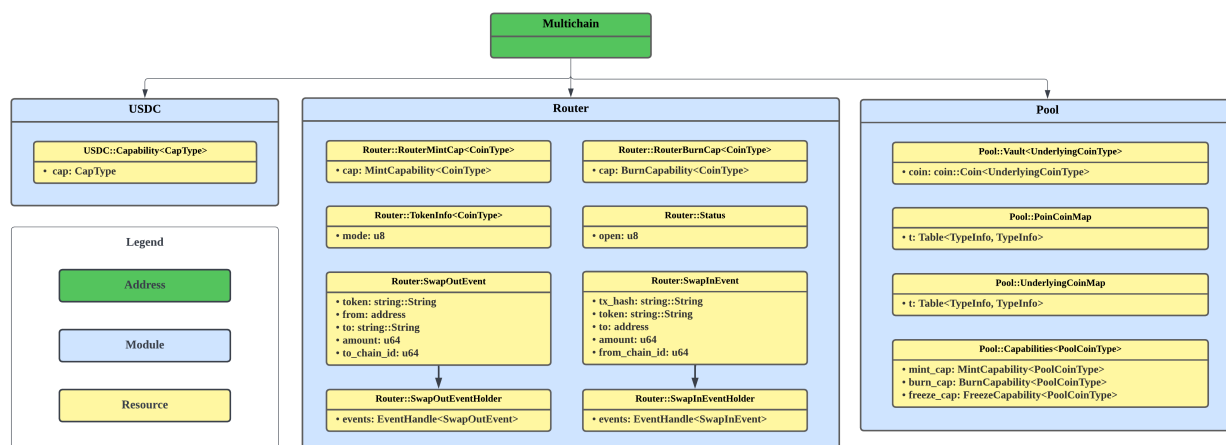
The router module maintains the main functionalities for the **MultiChain** project. It includes the liquidity pools used by the bridges of the protocol as well as the logic for transferring tokens into and out of the Aptos chain.

**Note:** Users should note that when transferring assets to a chain, it is possible for there to not be enough liquidity in the target chain to withdraw to the user. In this case, the protocol mints the user `any` tokens that can be redeemed for the actual token if there is ever enough liquidity in the target chain.

## USDC

The USDC component is an implementation of a coin on the Aptos chain.

## Resource and Account Relationship



## External Dependencies

Based on the current design, the project relies on Multichain's SMPC nodes (described [here](#)) to ensure that assets are transferred correctly across chains.

Furthermore, the project is developed using the Move language and runs on top of the Aptos blockchain. The vulnerabilities and updates of the language/Aptos client may also affect the project as a whole.

The above dependencies are not within the current audit scope and serve as a black box. Modules/contracts within the module are assumed to be valid and non-vulnerable actors in this audit and implement proper logic to collaborate with the current project and other modules.

## I Privileged Roles

As mentioned in Finding `GLOBAL-01`, the project maintains different privileged roles with functionality including but not limited to:

- maintaining the liquidity pools
- minting and burning of liquidity pool coins
- ensuring users receive the proper coins and amount on the Aptos chain
- pausing and unpausing the router

The advantage of the privileged role in the codebase is that the client reserves the ability to adjust the protocol according to the runtime required to best serve the community. It is also worth noting that the potential drawbacks of these functions should be clearly stated through the client's action/plan. Additionally, the project could have devastating consequences if the private keys of the privileged accounts are compromised.

Furthermore, the project's modules are upgradeable, meaning that features may be added or removed, possibly impacting the security of the project.

To improve the trustworthiness of the project, dynamic runtime updates in the project should be notified to the community. Furthermore, any plan to invoke the aforementioned functions should also be considered to move to the execution queue of a "timelock" contract.



## FINDINGS | MULTICHAIN FOUNDATION - APTOS



6

Total Findings

0

Critical

0

Major

0

Medium

2

Minor

4

Informational

This report has been prepared to discover issues and vulnerabilities for MultiChain Foundation - Aptos. Through this audit, we have uncovered 6 issues ranging from different severity levels. Utilizing the techniques of Manual Review & Static Analysis to complement rigorous manual code reviews, we discovered the following findings:

ID	Title	Category	Severity	Status
<a href="#">POO-01</a>	Entry Functions Should Not Have Return Values	Language Specific	Minor	● Resolved
<a href="#">POO-02</a>	Entry Functions Should Not Have Struct Inputs	Language Specific	Minor	● Resolved
<a href="#">GLOBAL-01</a>	Dependency On Key Management	External Dependency	Informational	● Mitigated
<a href="#">735-01</a>	Potential Failure On Coin Distribution	Logical Issue	Informational	● Resolved
<a href="#">POL-01</a>	Missing Validation On Underlying Coin Existence	Logical Issue	Informational	● Resolved
<a href="#">USD-01</a>	Unused Variables	Volatile Code	Informational	● Acknowledged

## POO-01 | ENTRY FUNCTIONS SHOULD NOT HAVE RETURN VALUES

Category	Severity	Location	Status
Language Specific	Minor	router/sources/Pool.move: 96, 155, 169	Resolved

### Description

In Move modules, `entry` functions should not have return values. However, in `Pool.move`, the following `entry` functions have return values:

- at L96, `query_underlying<PoolCoinType>(): (address, vector<u8>)`
- at L155, `withdrawByVault<CoinType>(account: &signer, amount: u64): coin::Coin<CoinType>`
- at L169, `vault<CoinType>(account: &signer): u64`

### Proof of Concept

We prepared a simple module and tried to interact with it

```
module TEST::test {  
  public entry fun test(): u64{  
    80  
  }  
}
```

Error Message:

```
{  
  "Error": "Simulation failed with status: Transaction Executed and Committed with  
Error INVALID_MAIN_FUNCTION_SIGNATURE"  
}
```

An entry function with a return value will allow other modules to interact with this function, but users will not be able to trigger the `entry` function with return values on the chain, for example, via the command line.

Please note that there is an open ticket for this issue, so it might lead to a compiler error in the future:

<https://github.com/aptos-labs/aptos-core/issues/2106>

### Recommendation

Recommend changing aforementioned functions to non-entry functions.

### Alleviation

**[MultiChain Foundation]:** The team resolved this issue by converting the aforementioned functions to `non-entry` in commit [3f1d93077d528d34f0bb5de716fd526805b9ac29](#).

## POO-02 | ENTRY FUNCTIONS SHOULD NOT HAVE STRUCT INPUTS

Category	Severity	Location	Status
Language Specific	Minor	router/sources/Pool.move: 146	Resolved

### Description

Functions with the `public entry` visibility can be called by a client, but generally, such functions that require a struct as an input cannot be called this way.

The following are the currently accepted inputs when using the Aptos CLI:

```
% aptos move run --help
aptos-move-run 0.3.9
Run a Move function

USAGE:
  aptos move run [OPTIONS] --function-id <FUNCTION_ID>

OPTIONS:
  --args <ARGS>...
    Arguments combined with their type separated by spaces.

    Supported types [u8, u64, u128, bool, hex, string, address, raw]

    Example: `address:0x1 bool:true u8:0`
```

The `public entry` function `depositByVault()` requires a `coin::Coin` struct as an input, but this cannot be called in a transaction.

Note that in the future, a compiler error may be introduced for this issue: <https://github.com/aptos-labs/aptos-core/issues/1886>.

### Recommendation

Recommend changing the function to not use the `entry` keyword.

### Alleviation

**[MultiChain Foundation]:** The team resolved this issue by converting the aforementioned functions to `non-entry` in commit [3f1d93077d528d34f0bb5de716fd526805b9ac29](https://github.com/multi-chain-foundation/multi-chain-foundation/commit/3f1d93077d528d34f0bb5de716fd526805b9ac29).

## GLOBAL-01 | DEPENDENCY ON KEY MANAGEMENT

Category	Severity	Location	Status
External Dependency	● Informational		● Mitigated

### Description

The project `multichain` highly depends on the role `@Multichain` to perform restricted functions shown below:

- `Multichain::Pool::register_coin<UnderlyingCoinType, PoolCoinType>()` will register an underlying coin and pool coin pair
- `Multichain::Pool::withdrawByVault<CoinType>()` will withdraw underlying assets from the vault
- `Multichain::Pool::mint_poolcoin<UnderlyingCoinType, PoolCoinType>()` will mint pool coins to a receiver
- `Multichain::Pool::burn_poolcoin<UnderlyingCoinType, PoolCoinType>()` will burn pool coins from a user
- `Multichain::Pool::vault<CoinType>()` will return the vault value
- `Multichain::Pool::copy_capabilities<CoinType>()` will return mint and burn capabilities based on the given coin type
- `Multichain::Router::set_coin<CoinType>()` will create/update `CoinInfo` for the given coin type
- `Multichain::Router::set_status()` will update the router status
- `Multichain::Router::approve_coin<CoinType>()` will create `RouterMintCap` and `RouterBurnCap` for `USDC` for the admin
- `Multichain::Router::set_poolcoin_cap<CoinType>()` will create `RouterMintCap` and `RouterBurnCap` for the admin
- `Multichain::Router::swapin<CoinType, PoolCoin>()` will issue assets to the user, based on the cross-chain event from other chains to Aptos. When the underlying coin is not enough, the pool coin will be minted to the user instead
- `Multichain::USDC::mint()` will mint coins to the receiver
- `Multichain::USDC::burn()` will burn coins from the target

Any compromise to the `@Multichain` account may allow a hacker to take advantage of the privileges above, such as manipulating coin distribution, withdrawing funds, disabling the router, etc.

Additionally, the program is upgradeable by default. Therefore, the module owner's (`@Multichain`) account should be carefully managed and avoid upgrading the modules into malicious ones.

Considering the critical role of `@Multichain`, the team needs to keep the private key safe. According to the [documentation](#) provided by the multichain team, the `@Multichain` role is managed by the SMPC network, which is out of the current audit scope.

## ■ Recommendation

We encourage the team to constantly monitor the status of the SMPC network and ensure the security of the operations to avoid errors.

## ■ Alleviation

**[MultiChain Foundation]:** The team acknowledged the finding and did not make any changes related to this finding. This issue is mitigated by the multichain SMPC network, which is a threshold-distributed signature algorithm based on secure multi-party computation (SMPC). With the network, the team is able to handle the signatures and assets safely and effectively.

## 735-01 | POTENTIAL FAILURE ON COIN DISTRIBUTION

Category	Severity	Location	Status
Logical Issue	● Informational	mintable/sources/USDC.move: 61; router/sources/Pool.move: 194; router/sources/Router.move: 143	● Resolved

### Description

The `mint_poolcoin` function allows the admin (`@Multichain`) to mint pool tokens to the receiver.

According to the logic in `aptos-framework/sources/coin.move`, the user should register the `CoinStore` for a specific coin type in order to receive assets of that coin type.

```
public fun deposit<CoinType>(account_addr: address, coin: Coin<CoinType>)
acquires CoinStore {
    assert!(
        is_account_registered<CoinType>(account_addr),
        error::not_found(ECOIN_STORE_NOT_PUBLISHED),
    );
    ...
}
```

If the user does not register the `CoinStore`, functionality like `mint_poolcoin` or `swapin` should fail. Similarly, the failure may happen when issuing any new token to the user, like mintable coins, pool coins, etc.

### Recommendation

Considering cross-chain transactions are not atomic, we would like to check with the team on how they plan to handle the situation of when a token transfer to the Aptos chain fails due to the user not having the associated `CoinStore` resource.

### Alleviation

**[MultiChain Foundation]:** In our design now, users who want to cross assets to Aptos must register both underlyingCoin and PoolCoin when they use multichain Dapp with Petra Wallet.

## **POL-01** | MISSING VALIDATION ON UNDERLYING COIN EXISTENCE

Category	Severity	Location	Status
Logical Issue	● Informational	Pool.move (af20bdd2e6fe4160f761c5cfc2c3d98aaa89a5ee): 44	● Resolved

### **I Description**

The function `register_coin` will allow the admin ( `@Multichain` ) to register a pair for an underlying coin and the corresponding pool coin.

The concern is that there is no validation ensuring that the given underlying coin has been initialized. If the coin has not been initialized, the pair will not be available for use.

### **I Recommendation**

Recommend adding a validation on the underlying coin to ensure the coin is already initialized, for example, using the

`coin::is_coin_initialized<>()` function.

### **I Alleviation**

**[MultiChain Foundation]:** The team resolved this issue by adding validation in commit [fbdf58bdf0dbbd8022b70761150a7ea4fecaa67](#).



## USD-01 | UNUSED VARIABLES

Category	Severity	Location	Status
Volatile Code	● Informational	mintable/sources/USDC.move: 11~15	● Acknowledged

### I Description

The variables `ERR_CAP_MISSED` and `ERR_CAP_EXISTS` are declared but never used.

### I Recommendation

Recommend removing the unused variables if it is not intended to be used.

### I Alleviation

**[MultiChain Foundation]:** The team acknowledged this issue and decided not to change the current codebase at this time.

## APPENDIX | MULTICHAIN FOUNDATION - APTOS

### Finding Categories

Categories	Description
Logical Issue	Logical Issue findings detail a fault in the logic of the linked code, such as an incorrect notion on how <code>block.timestamp</code> works.
Volatile Code	Volatile Code findings refer to segments of code that behave unexpectedly on certain edge cases that may result in a vulnerability.
Language Specific	Language Specific findings are issues that would only arise within Solidity, i.e. incorrect usage of <code>private</code> or <code>delete</code> .

### Checksum Calculation Method

The "Checksum" field in the "Audit Scope" section is calculated as the SHA-256 (Secure Hash Algorithm 2 with digest size of 256 bits) digest of the content of each file hosted in the listed source repository under the specified commit.

The result is hexadecimal encoded and is the same as the output of the Linux `sha256sum` command against the target file.

## DISCLAIMER | CERTIK

This report is subject to the terms and conditions (including without limitation, description of services, confidentiality, disclaimer and limitation of liability) set forth in the Services Agreement, or the scope of services, and terms and conditions provided to you ("Customer" or the "Company") in connection with the Agreement. This report provided in connection with the Services set forth in the Agreement shall be used by the Company only to the extent permitted under the terms and conditions set forth in the Agreement. This report may not be transmitted, disclosed, referred to or relied upon by any person for any purposes, nor may copies be delivered to any other person other than the Company, without CertiK's prior written consent in each instance.

This report is not, nor should be considered, an "endorsement" or "disapproval" of any particular project or team. This report is 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 assessment. This report does 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.

This report should not be used in any way to make decisions around investment or involvement with any particular project. This report in no way provides investment advice, nor should be leveraged as investment advice of any sort. This report represents an extensive assessing 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.

The assessment services provided by CertiK is subject to dependencies and under continuing development. You agree that your access and/or use, including but not limited to any services, reports, and materials, will be at your sole risk on an as-is, where-is, and as-available basis. Cryptographic tokens are emergent technologies and carry with them high levels of technical risk and uncertainty. The assessment reports could include false positives, false negatives, and other unpredictable results. The services may access, and depend upon, multiple layers of third-parties.

ALL SERVICES, THE LABELS, THE ASSESSMENT REPORT, WORK PRODUCT, OR OTHER MATERIALS, OR ANY PRODUCTS OR RESULTS OF THE USE THEREOF ARE PROVIDED "AS IS" AND "AS AVAILABLE" AND WITH ALL FAULTS AND DEFECTS WITHOUT WARRANTY OF ANY KIND. TO THE MAXIMUM EXTENT PERMITTED UNDER APPLICABLE LAW, CERTIK HEREBY DISCLAIMS ALL WARRANTIES, WHETHER EXPRESS, IMPLIED, STATUTORY, OR OTHERWISE WITH RESPECT TO THE SERVICES, ASSESSMENT REPORT, OR OTHER MATERIALS. WITHOUT LIMITING THE FOREGOING, CERTIK SPECIFICALLY DISCLAIMS ALL IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, TITLE AND NON-INFRINGEMENT, AND ALL WARRANTIES ARISING FROM COURSE OF DEALING, USAGE, OR TRADE PRACTICE. WITHOUT LIMITING THE FOREGOING, CERTIK MAKES NO WARRANTY OF ANY KIND THAT THE SERVICES, THE LABELS, THE ASSESSMENT REPORT, WORK PRODUCT, OR OTHER MATERIALS, OR ANY PRODUCTS OR RESULTS OF THE USE THEREOF, WILL MEET CUSTOMER'S OR ANY OTHER PERSON'S REQUIREMENTS, ACHIEVE ANY INTENDED RESULT, BE COMPATIBLE OR WORK WITH ANY SOFTWARE, SYSTEM, OR OTHER SERVICES, OR BE SECURE, ACCURATE, COMPLETE, FREE OF HARMFUL CODE, OR ERROR-FREE. WITHOUT LIMITATION TO THE

FOREGOING, CERTIK PROVIDES NO WARRANTY OR UNDERTAKING, AND MAKES NO REPRESENTATION OF ANY KIND THAT THE SERVICE WILL MEET CUSTOMER'S REQUIREMENTS, ACHIEVE ANY INTENDED RESULTS, BE COMPATIBLE OR WORK WITH ANY OTHER SOFTWARE, APPLICATIONS, SYSTEMS OR SERVICES, OPERATE WITHOUT INTERRUPTION, MEET ANY PERFORMANCE OR RELIABILITY STANDARDS OR BE ERROR FREE OR THAT ANY ERRORS OR DEFECTS CAN OR WILL BE CORRECTED.

WITHOUT LIMITING THE FOREGOING, NEITHER CERTIK NOR ANY OF CERTIK'S AGENTS MAKES ANY REPRESENTATION OR WARRANTY OF ANY KIND, EXPRESS OR IMPLIED AS TO THE ACCURACY, RELIABILITY, OR CURRENCY OF ANY INFORMATION OR CONTENT PROVIDED THROUGH THE SERVICE. CERTIK WILL ASSUME NO LIABILITY OR RESPONSIBILITY FOR (I) ANY ERRORS, MISTAKES, OR INACCURACIES OF CONTENT AND MATERIALS OR FOR ANY LOSS OR DAMAGE OF ANY KIND INCURRED AS A RESULT OF THE USE OF ANY CONTENT, OR (II) ANY PERSONAL INJURY OR PROPERTY DAMAGE, OF ANY NATURE WHATSOEVER, RESULTING FROM CUSTOMER'S ACCESS TO OR USE OF THE SERVICES, ASSESSMENT REPORT, OR OTHER MATERIALS.

ALL THIRD-PARTY MATERIALS ARE PROVIDED "AS IS" AND ANY REPRESENTATION OR WARRANTY OF OR CONCERNING ANY THIRD-PARTY MATERIALS IS STRICTLY BETWEEN CUSTOMER AND THE THIRD-PARTY OWNER OR DISTRIBUTOR OF THE THIRD-PARTY MATERIALS.

THE SERVICES, ASSESSMENT REPORT, AND ANY OTHER MATERIALS HEREUNDER ARE SOLELY PROVIDED TO CUSTOMER AND MAY NOT BE RELIED ON BY ANY OTHER PERSON OR FOR ANY PURPOSE NOT SPECIFICALLY IDENTIFIED IN THIS AGREEMENT, NOR MAY COPIES BE DELIVERED TO, ANY OTHER PERSON WITHOUT CERTIK'S PRIOR WRITTEN CONSENT IN EACH INSTANCE.

NO THIRD PARTY OR ANYONE ACTING ON BEHALF OF ANY THEREOF, SHALL BE A THIRD PARTY OR OTHER BENEFICIARY OF SUCH SERVICES, ASSESSMENT REPORT, AND ANY ACCOMPANYING MATERIALS AND NO SUCH THIRD PARTY SHALL HAVE ANY RIGHTS OF CONTRIBUTION AGAINST CERTIK WITH RESPECT TO SUCH SERVICES, ASSESSMENT REPORT, AND ANY ACCOMPANYING MATERIALS.

THE REPRESENTATIONS AND WARRANTIES OF CERTIK CONTAINED IN THIS AGREEMENT ARE SOLELY FOR THE BENEFIT OF CUSTOMER. ACCORDINGLY, NO THIRD PARTY OR ANYONE ACTING ON BEHALF OF ANY THEREOF, SHALL BE A THIRD PARTY OR OTHER BENEFICIARY OF SUCH REPRESENTATIONS AND WARRANTIES AND NO SUCH THIRD PARTY SHALL HAVE ANY RIGHTS OF CONTRIBUTION AGAINST CERTIK WITH RESPECT TO SUCH REPRESENTATIONS OR WARRANTIES OR ANY MATTER SUBJECT TO OR RESULTING IN INDEMNIFICATION UNDER THIS AGREEMENT OR OTHERWISE.

FOR AVOIDANCE OF DOUBT, THE SERVICES, INCLUDING ANY ASSOCIATED ASSESSMENT REPORTS OR MATERIALS, SHALL NOT BE CONSIDERED OR RELIED UPON AS ANY FORM OF FINANCIAL, TAX, LEGAL, REGULATORY, OR OTHER ADVICE.

# CertiK | Securing the Web3 World

Founded in 2017 by leading academics in the field of Computer Science from both Yale and Columbia University, CertiK is a leading blockchain security company that serves to verify the security and correctness of smart contracts and blockchain-based protocols. Through the utilization of our world-class technical expertise, alongside our proprietary, innovative tech, we're able to support the success of our clients with best-in-class security, all whilst realizing our overarching vision; provable trust for all throughout all facets of blockchain.

