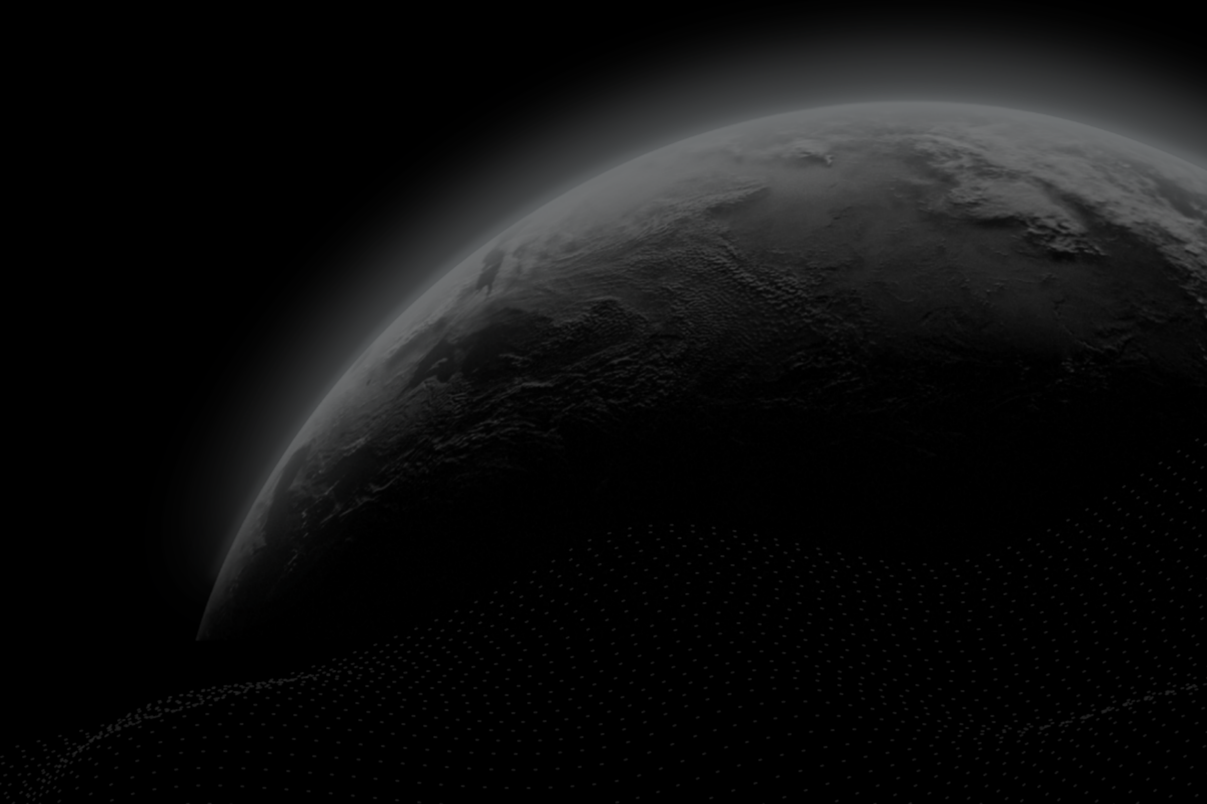
CERTIK

Security Assessment

Wido - Audit

CertiK Verified on Dec 13th, 2022



CertiK Verified on Dec 13th, 2022

Wido - Audit

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

Executive Summary

TYPES	ECOSYSTEM	METHODS
DeFi	Ethereum	Manual Review, Static Analysis
LANGUAGE	**TIMELINE**	**KEY COMPONENTS**
Solidity	Delivered on 12/13/2022	N/A

CODEBASE

https://github.com/widolabs/wido-contracts/tree/main/contracts

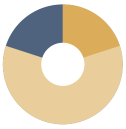
...View All

COMMITS

- e2066363fa777ec0a42b53bbf821a0081d9a71ca
- 6dbfae814a9f6881fd76ba5d741b584b31b73dde

...View All

Vulnerability Summary



5	5	0	0	0	0	0
Total Findings	Resolved	Mitigated	Partially Resolved	Acknowledged	Declined	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.
🟨	1	Medium	1 Resolved	Medium risks may not pose a direct risk to users' funds, but they can affect the overall functioning of a platform.
🟨	3	Minor	3 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.
🟦	1	Informational	1 Resolved	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 | WIDO - AUDIT

CODEBASE | WIDO - AUDIT

Repository

https://github.com/widolabs/wido-contracts/tree/main/contracts

Commit

- e2066363fa777ec0a42b53bbf821a0081d9a71ca
- 6dbfae814a9f6881fd76ba5d741b584b31b73dde

AUDIT SCOPE | WIDO - AUDIT

5 files audited ● 5 files without findings

ID	File	SHA256 Checksum
● IWD	contracts/interfaces/IWidoRouter.sol	05be14d8306ef3bec51482c9ae40bff58c6028251066cf9f30a0451addf5e09c
● ITM	contracts/interfaces/IWidoTokenManager.sol	f2fe1f711a3ffab4e12d0485cffc1ced22ea9f264581d5f2d3a1ac92776577f3
● WRH	contracts/WidoRouter.sol	7aa87bdfda6f51ed5a5ff762278bccaf61c763643327ff58bd35316d3371ea77
● WTM	contracts/WidoTokenManager.sol	ccfa756e84ed4216d6a781036fc6ef2cdd07ba40978161d4cb9f07e95c7e7664
● WZP	contracts/WidoZapUniswapV2Pool.sol	fb8af828ad923025ad9c3d486c17c5ddc2497973ee8d1e5058885aaf363541c0

APPROACH & METHODS | WIDO - AUDIT

This report has been prepared for Wido to discover issues and vulnerabilities in the source code of the Wido - Audit 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 | WIDO - AUDIT

Overview

Wido is a routing protocol that finds the best path to get from tokenA to tokenB. dApps and Protocols use Wido to accept deposits in any token, which leads to improved deposit conversion.

Features

Main functionalities

The project provides `executeOrder()` function to transform ERC20 token from `order.inputs` to `order.outputs` . The `minOutputAmount` parameters should be set as the minimum output user expected to receive.

The project also provides the `zapIn()` and `zapOut()` to add liquidity with one of the pool tokens. The `minOutputAmount` parameters should be set as the minimum output user expected to receive.

Due to this mechanism, if the `minOutputAmount` amount is properly set, it will prevent sandwich attacks during the token-swapping process.

Therefore, we recommend users who interact with those functions properly set the `minOutputAmount` parameters to avoid potential sandwich attacks.

No-withdrawable tokens are redistributed

The protocol will generate dust/leftover tokens during the swapping and liquidity adding process, those tokens will be redistributed in the next round of token swapping and liquidity adding process.

Third Party Dependencies

The scope of the audit treats third party entities as black boxes and assume their functional correctness. However, in the real world, third parties can be compromised and this may lead to lost or stolen assets.

The contract `WidoRouter.sol` and `WidoZapUniswapV2Pool.sol` are serving as the underlying entities to interact with third parties (e.g, tokens and accounts) mainly via the structs `Order` and `Step` .

Privileged Functions

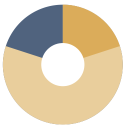
In the contract `WidoRouter.sol` , the role `onlyOwner` has authority over the following function:

- function `setBank()` to set a new `bank` address.

Any compromise to the `onlyOwner` account may allow a hacker to take advantage of this authority and modify the bank address without the consensus of community.

To improve the trustworthiness of the project, dynamic runtime updates in the project should be notified to the community. Any plan to invoke the aforementioned functions should be also considered to move to the execution queue of `Timelock` contract.

FINDINGS | WIDO - AUDIT



5	0	0	1	3	1
Total Findings	Critical	Major	Medium	Minor	Informational

This report has been prepared to discover issues and vulnerabilities for Wido - Audit. Through this audit, we have uncovered 5 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
WRB-01	Potential Bypass Of Fees	Logical Issue	Medium	● Resolved
WRB-02	Mistakenly Approve To Router Contract Could Lead To Fund Stolen	Logical Issue	Minor	● Resolved
WRB-03	Potential Leftovers In WidoRouter Contract	Logical Issue	Minor	● Resolved
WRB-04	Function `executeOrderWithSignature()` Not Verify The `route` Parameter	Logical Issue	Minor	● Resolved
WRB-05	Interacting With External Vulnerable Swap Contracts May Cause User's Loss	Logical Issue	Informational	● Resolved

WRB-01 | POTENTIAL BYPASS OF FEES

Category	Severity	Location	Status
Logical Issue	● Medium	WidoRouter.sol (Wido Update - 11/24): 231, 250	● Resolved

Description

In the contract `WidoRouter`, function `executeOrder()` takes `feeBps` as input, which specifies how many fees to be charged.

```
    function executeOrder(
        Order calldata order,
        Step[] calldata route,
        uint256 feeBps,
        address partner
    )
```

Later, the variable `feeBps` will be passed to the function `_collectFees()` to calculate the `fee` that will be transferred to the `bank` address.

```
    function _collectFees(address fromToken, uint256 amount, uint256 feeBps) private
{
        require(feeBps <= 100, "Fee out of range");
        uint256 fee = (amount * feeBps) / 10000;
        if (fromToken == address(0)) {
            bank.safeTransferETH(fee);
        } else {
            ERC20(fromToken).safeTransfer(bank, fee);
        }
    }
```

However, there is no restriction for the input `feeBps` in the linked functions. Consequently, users can set the input `feeBps` to zero and thus bypassing the `require()` statement.

Recommendation

We would like to learn if it is the intended design. If it is not intended, instead of providing fee percentage from user input, it is recommended to store the fee percentage in a variable in contract, which can only be changed by owner address.

Alleviation

[**Wido**, 12/09/2022]: The team confirmed specifying fees per transaction is intended by design, including 0 fees.

[**CertiK**, 12/09/2022]: Taking 0 fees are intended design, and the Wido team fully understands it might cause some profit loss.

WRB-02 | MISTAKENLY APPROVE TO ROUTER CONTRACT COULD LEAD TO FUND STOLEN

Category	Severity	Location	Status
Logical Issue	● Minor	WidoRouter.sol (Wido Update - 11/24): 121	● Resolved

Description

In the `WidoRouter` contract, the low-level call is implemented to perform swap operation in a flexible way.

```
(bool success, bytes memory result) = step.targetAddress.call{value: value}
(editedSwapData);
```

As per the current design, users will approve funds to the WidoManager contract instead of the `WidoRouter` contract, which will prevent attackers from directly triggering the low-level call with `transferFrom()` invocation to drain users' assets.

Users should never approve the WidoRouter contract. If users approve `WidoRouter` contract by mistake, it could lead to a potential loss of funds.

Recommendation

In the short term, we recommend informing the community of the potential risk to prevent users from approving router contracts with an unlimited allowance. Additionally, it could be helpful to add a whitelist mechanism to ensure that only the legitimate `targetAddress` is allowed.

In the long term, the contract is recommended to be redesigned to avoid low-level calls. A suggested solution is to hardcode the intended interfaces/protocols in the contracts.

Alleviation

[**Wido**, 12/08/2022]: The team added a comment in the contract to notify users not to approve the `WidoRouter` contract to spend their tokens in the commit e2066363fa777ec0a42b53bbf821a0081d9a71ca.

[**CertiK**, 12/09/2022]: Wido team has adopted the short-term solution and added comments accordingly to warn users of the potential risk. Users should not directly approve routers to spend their tokens.

WRB-03 | POTENTIAL LEFTOVERS IN WIDOROUTER CONTRACT

Category	Severity	Location	Status
Logical Issue	● Minor	WidoRouter.sol (Wido Update - 11/24): 166	● Resolved

Description

In the `WidoRouter` contract, users can swap tokens via the `executeOrder()` function call with a designated input `Order` struct.

```
function executeOrder(
    Order calldata order,
    Step[] calldata route,
    uint256 feeBps,
    address partner
)
```

```
struct Order {
    OrderInput[] inputs;
    OrderOutput[] outputs;
    address user;
    uint32 nonce;
    uint32 expiration;
}
```

The `order.inputs` specifies the amount of `from` tokens to be swapped. The exact amount (`order.inputs`) tokens will be pulled from user's address during the `_executeOrder()` operation. After the swap, all the `toTokens` in the contract will be transferred to user's address.

However, due to the various swapping mechanisms of different protocols, there might be some potential `fromToken` leftovers in the `WidoRouter` contract during the swap.

For example, a user wants to swap 100 tokens. The exact 100 tokens will be pulled from the user's address to the `WidoRouter` contract. If the external swap protocol only takes 98 tokens for swapping, there will be two tokens left in the `WidoRouter` .

Consequently, other users could exploit the low-level call to transfer the leftovers to their own addresses.

It is also worth mentioning that if the swapping is performed via multiple paths, there could be multiple token leftovers inside the contract.

Recommendation

Considering there might be also `fromToken` leftovers during the swap, it is recommended to return those leftovers to users.

Alleviation

[**Wido**, 12/08/2022]: We expect the dust leftover to be minimal and infrequent which does not justify additional gas cost in each transaction for dust management.

[**CertiK**, 12/09/2022]: Wido understands the issue and confirms it is the intended design. As the dust might be small, the impact is limited for each transaction. However, the dust might accumulate due to the contract interactions.

WRB-04 | FUNCTION `executeOrderWithSignature()` NOT VERIFY THE `route` PARAMETER

Category	Severity	Location	Status
Logical Issue	● Minor	WidoRouter.sol (Wido Update - 11/24): 267~268	● Resolved

Description

In the function `executeOrderWithSignature()` , users can sign and delegate the transaction to another party.

```
    function executeOrderWithSignature(
        Order calldata order,
        Step[] calldata route,
        uint8 v,
        bytes32 r,
        bytes32 s,
        uint256 feeBps,
        address partner
    ) external override nonReentrant {
        require(verifyOrder(order, v, r, s), "Invalid order");
        nonces[order.user]++;
        _executeOrder(order, route, order.user, feeBps);
        emit FulfilledOrder(order, msg.sender, order.user, feeBps, partner);
    }
```

However, the contract only verifies the `Order` struct signed by the users and does not validate input `route` . In this case, the attacker can spoof a malicious `route` payload, which calls a malicious contract. Therefore, users might suffer unexpected loss.

Due to the protection on `minOutputAmount` , which enforces the minimum output amount from the swap, otherwise, it will revert.

```
  if (toTokenBalance < order.minOutputAmount) revert
SlippageTooHigh(order.minOutputAmount, toTokenBalance);
```

However, users might mistakenly set the `minOutputAmount,` thus creating a chance for attackers to steal their funds.

Recommendation

Recommend adding checks to verify the `route` parameter and to ensure that the users have correctly signed the `route` path. Also, we would encourage setting a default range for the `minOutputAmount` to ensure the output are within expectation.

Alleviation

[**Wido**, 12/08/2022]: This is an intended design as it would allow market makers to compete and fulfill user orders. The team expects users to verify and set appropriate `minOutputAmount` as they would otherwise be subjected to MEV attacks.

[**CertiK**, 12/09/2022]: It is true that if the value `minOutputAmount` is correctly set, it will prevent users' loss due to the protection and reverting of the transaction. Users must carefully choose the input `minOutputAmount` to avoid attacks.

WRB-05 | INTERACTING WITH EXTERNAL VULNERABLE SWAP CONTRACTS MAY CAUSE USER'S LOSS

Category	Severity	Location	Status
Logical Issue	● Informational	WidoRouter.sol (Wido Update - 11/24): 121	● Resolved

Description

The Wido project allows user to call arbitrary addresses to swap their tokens, which is specified in the `route` calldata.

```
    function _executeSteps(Step[] calldata route) private {
```

The concern is that if a vulnerable swap contract is used during the call (i.e., in `route` input), it could lead to users' asset loss.

For example, a user wants to swap tokens in a certain `route`, however, one of the swap contract in the `route` is vulnerable and drains 10% of the amount from user's input amount. Therefore, after swapping, the user to will lose 10% of the tokens during this swap.

Recommendation

It is recommended to whitelist the swapping contracts that are allowed to be used to avoid using vulnerable swapping contracts.

Alleviation

[**Wido**, 12/08/2022]: The parameter `minOutputAmount` is part of the order that prevents users against MEV attacks and guarantees a maximum slippage. Users can verify and set expectations on the amount of tokens they want to receive. If the `minOutputAmount` is set to the appropriate value, even an external vulnerable swap contract cannot result in user token loss.

[**CertiK**, 12/09/2022]: It is true that if the value `minOutputAmount` is correctly set, it will prevent users' loss due to the protection and reverting of the transaction. Users must carefully choose the input `minOutputAmount` to avoid attacks.

APPENDIX | WIDO - AUDIT

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 block.timestamp works.

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.