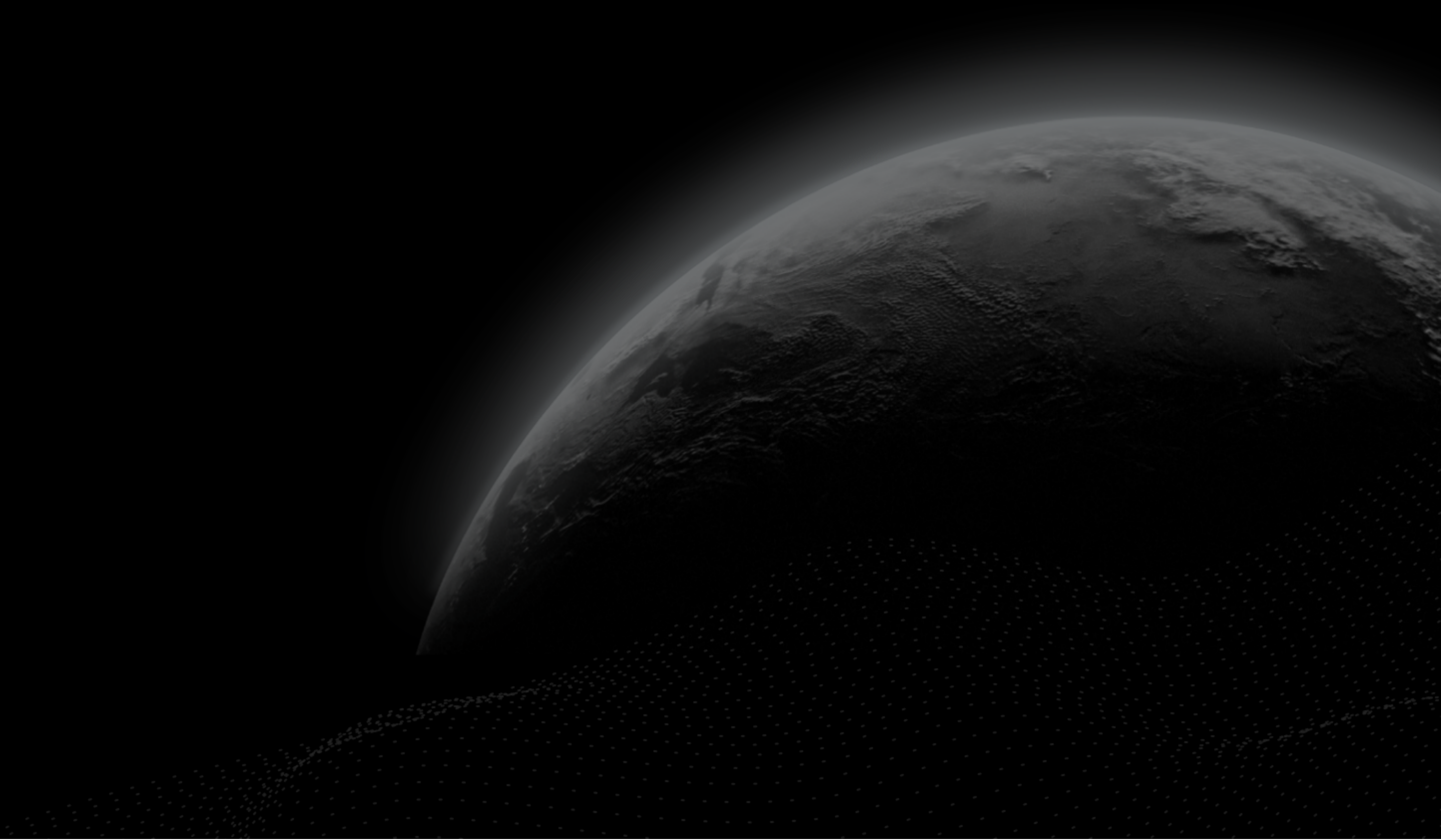
Security Assessment

# Venus - Unlist Market & Borrow Cap

CertiK Assessed on Apr 9th, 2024



CertiK Assessed on Apr 9th, 2024

# Venus - Unlist Market & Borrow Cap

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

## Executive Summary

TYPES
DeFi

ECOSYSTEM
Binance Smart Chain (BSC)

METHODS
Manual Review, Static Analysis

LANGUAGE
Solidity

TIMELINE
Delivered on 04/09/2024

KEY COMPONENTS
N/A

CODEBASE
https://github.com/VenusProtocol/isolated-pools
https://github.com/VenusProtocol/venus-protocol
View All in Codebase Page

COMMITS
PR-349 base: 0b3a26bb23a359af6435f3d3b95a116bd1301a88
PR-438 base: 935292415bc22f79163581858c083a117f1743d3
PR-429 base: abb29cec0a15ae247f4846f4e2e5d47f2f139e88
View All in Codebase Page

## Vulnerability Summary

| Total Findings | Resolved | Mitigated | Partially Resolved | Acknowledged | Declined |
|---|---|---|---|---|---|
| 5 | 3 | 1 | 0 | 1 | 0 |

| | | | | |
|---|---|---|---|---|
| 🟥 | 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. |
| 🟧 | 1 | Major | 1 Mitigated | 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. |
| 🟨 | 1 | Minor | 1 Acknowledged | 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. |
| 🟦 | 3 | Informational | 3 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 | VENUS - UNLIST MARKET & BORROW CAP

# CODEBASE | VENUS - UNLIST MARKET & BORROW CAP

## Repository

https://github.com/VenusProtocol/isolated-pools

https://github.com/VenusProtocol/venus-protocol

## Commit

PR-349 base: 0b3a26bb23a359af6435f3d3b95a116bd1301a88

PR-438 base: 935292415bc22f79163581858c083a117f1743d3

PR-429 base: abb29cec0a15ae247f4846f4e2e5d47f2f139e88

PR-349 update 1: 144cb9761cc3da4215dda2240b68d939c2e586f7

PR-438 update 1: 086c073fa31df3cf34971b255843dff6232e8dd7

PR-429 update 1: 36ee37bd94b291a66685e633bba5c5136ce03a3c

# AUDIT SCOPE | VENUS - UNLIST MARKET & BORROW CAP

4 files audited ● 1 file with Acknowledged findings ● 1 file with Mitigated findings ● 1 file with Resolved findings

● 1 file without findings

| ID | Repo | File | SHA256 Checksum |
|---|---|---|---|
| ● CVP | VenusProtocol/isolated-pools | contracts/Comptroller.sol | eb69b991ff5e0378d3d7ceaed9de6d6df898b0bd3f3ae12fb80159255e56ab4e |
| ● MFD | VenusProtocol/venus-protocol | contracts/Comptroller/Diamond/facets/MarketFacet.sol | e8f585f0e7e036487492e19d4879b88bd48ee63ad483034931261f835558b489 |
| ● PFD | VenusProtocol/venus-protocol | contracts/Comptroller/Diamond/facets/PolicyFacet.sol | 0dafbe836692140fd32160941ce531ef64a96586d70657a3c993e381902a89cb |
| ● SFD | VenusProtocol/venus-protocol | contracts/Comptroller/Diamond/facets/SetterFacet.sol | 9e8986d8ca6b1c621b9db3a64e05a085c2c0efa43c3c94c7120c6f1fd7c3fd86 |

# APPROACH & METHODS

## VENUS - UNLIST MARKET & BORROW CAP

This report has been prepared for Venus to discover issues and vulnerabilities in the source code of the Venus - Unlist Market & Borrow Cap 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.

# SUMMARY | VENUS - UNLIST MARKET & BORROW CAP

This audit concerns the changes made in files outlined in:

- PR-429 until commit abb29cec0a15ae247f4846f4e2e5d47f2f139e88;
- PR-349 until commit 0b3a26bb23a359af6435f3d3b95a116bd1301a88;
- PR-438 until commit 935292415bc22f79163581858c083a117f1743d3.

PR-429 and PR-349 are designed to add functionality to unlist a market in the Venus-Protocol and Isolated-Pools respectively. Such an operation can only be performed if actions are paused, the borrow and supply caps are set to zero, and the collateral factor are set to zero.

PR-438 is designed to update how the protocol handles when the borrow cap is 0. Previously a borrow cap of 0 corresponded to unlimited borrowing, where this PR changes this behavior so that a borrow cap of 0 corresponds to borrowing being disallowed.

Note that any centralization risks present in the existing codebase before these PRs were not considered in this audit and only those added in these PRs are addressed in the audit. We recommend all users carefully review the centralization risks, much of which can be found in our previous audits which can be found here: https://skynet.certik.com/projects/venus.

# FINDINGS | VENUS - UNLIST MARKET & BORROW CAP

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

This report has been prepared to discover issues and vulnerabilities for Venus - Unlist Market & Borrow Cap. 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 |
|---|---|---|---|---|
| **VPB-02** | **Centralization Related Risks** | **Centralization** | **Major** | ● **Mitigated** |
| CVP-02 | File Allows Solidity Version That Is Susceptible To An Assembly Optimizer Bug | Language Version | Minor | ● Acknowledged |
| VPB-01 | Missing Checks When Unlisting Market | Design Issue | Informational | ● Resolved |
| VPB-04 | Discrepancy Between Use Of Borrow Cap In Core Vs. Isolated Pools | Coding Style | Informational | ● Resolved |
| VPB-05 | Typos And Inconsistencies | Inconsistency | Informational | ● Resolved |

# VPB-02 | CENTRALIZATION RELATED RISKS

| Category | Severity | Location | Status |
|---|---|---|---|
| **Centralization** | ● **Major** | **contracts/Comptroller.sol (PR349-Base): 211~212; contracts/Comptroller/Diamond/facets/MarketFacet.sol (PR429-Base): 142~143** | ● **Mitigated** |

## Description

Note that any centralization risks present in the existing codebase before the PR's in scope of this audit were not considered. Only those added to the in-scope PRs are addressed. We recommend all users carefully review the centralization risks, much of which can be found in our previous audits, which can be found here: https://skynet.certik.com/projects/venus.

### PR429 MarketFacet

In the contract `MarketFacet` the `DEFAULT_ADMIN_ROLE` of the `AccessControlManager` can grant addresses the privilege to call the function `unlistMarket()` . Any compromise to the `DEFAULT_ADMIN_ROLE` or accounts granted this privilege may allow the hacker to take advantage of this authority and unlist legitimate markets, only if they also have control over pausing all necessary action states and updating borrow caps, supply caps, and collateral factors to 0.

### PR349 Comptroller

In the contract `Comptroller` , the `DEFAULT_ADMIN_ROLE` of the `AccessControlManager` can grant addresses the privilege to call the function `unlistMarket()` . Any compromise to the `DEFAULT_ADMIN_ROLE` or accounts granted this privilege may allow the hacker to take advantage of this authority and unlist legitimate markets, only if they also have control over pausing all necessary action states and updating borrow caps, supply caps, and collateral factors to 0.

## Recommendation

The risk describes the current project design and potentially makes iterations to improve in the security operation and level of decentralization, which in most cases cannot be resolved entirely at the present stage. We advise the client to carefully manage the privileged account's private key to avoid any potential risks of being hacked. In general, we strongly recommend centralized privileges or roles in the protocol be improved via a decentralized mechanism or smart-contract-based accounts with enhanced security practices, e.g., multisignature wallets. Indicatively, here are some feasible suggestions that would also mitigate the potential risk at a different level in terms of short-term, long-term and permanent:

### Short Term:

Timelock and Multi sign (⅔, ⅗) combination *mitigate* by delaying the sensitive operation and avoiding a single point of key management failure.

- Time-lock with reasonable latency, e.g., 48 hours, for awareness on privileged operations;
  AND
- Assignment of privileged roles to multi-signature wallets to prevent a single point of failure due to the private key compromised;
  AND
- A medium/blog link for sharing the timelock contract and multi-signers addresses information with the public audience.

### Long Term:

Timelock and DAO, the combination, *mitigate* by applying decentralization and transparency.

- Time-lock with reasonable latency, e.g., 48 hours, for awareness on privileged operations;
  AND
- Introduction of a DAO/governance/voting module to increase transparency and user involvement.
  AND
- A medium/blog link for sharing the timelock contract, multi-signers addresses, and DAO information with the public audience.

### Permanent:

Renouncing the ownership or removing the function can be considered *fully resolved*.

- Renounce the ownership and never claim back the privileged roles.
  OR
- Remove the risky functionality.

## Alleviation

`[Venus, 03/21/2024]` : "In both cases, we'll use the AccessControlManager (ACM) deployed at 0x4788629abc6cfca10f9f969efdeaa1cf70c23555.

In this ACM, only 0x939bd8d64c0a9583a7dcea9933f7b21697ab6396 (Normal Timelock) has the DEFAULT_ADMIN_ROLE. And this contract is a Timelock contract used during the Venus Improvement Proposals."

# CVP-02 | FILE ALLOWS SOLIDITY VERSION THAT IS SUSCEPTIBLE TO AN ASSEMBLY OPTIMIZER BUG

| Category | Severity | Location | Status |
|---|---|---|---|
| Language Version | ● Minor | contracts/Comptroller.sol (PR349-Base): 1384~1385 | ● Acknowledged |

## Description

In solidity versions 0.8.13 and 0.8.14, there is an optimizer bug where, if the use of a variable is in a separate `assembly` block from the block in which it was stored, the `mstore` operation is optimized out, leading to uninitialized memory. The code currently does not have such a pattern of execution, but it does use `mstore` s in `assembly` blocks, so it is a risk for future changes.

## Recommendation

We recommend ensuring that this bug is not introduced in future changes, by either ensuring it in your workflow or changing to a solidity version where this bug does not exist.

## Alleviation

`[Venus, 03/19/2024]` : "Issue acknowledged. I won't make any changes for the current version."

# VPB-01 | MISSING CHECKS WHEN UNLISTING MARKET

| Category | Severity | Location | Status |
|---|---|---|---|
| Design Issue | ● Informational | contracts/Comptroller.sol (PR349-Base): 220~225; contracts/Comptroller/Diamond/facets/MarketFacet.sol (PR429-Base): 151~156 | ● Resolved |

## Description

The possible actions to pause within a Comptroller are:

```
enum Action {
    MINT,
    REDEEM,
    BORROW,
    REPAY,
    SEIZE,
    LIQUIDATE,
    TRANSFER,
    ENTER_MARKET,
    EXIT_MARKET
}
```

When a market is unlisted, it is only ensured that the actions of `BORROW` , `MINT` , `REDEEM` , `REPAY` , `ENTER_MARKET` , and `LIQUIDATE` have been paused. It is not checked that the actions of `SEIZE` , `TRANSFER` , or `EXIT_MARKET` have been paused, even though these actions should still be paused. As a consequence, these three actions will be available for continued use, right up until a market has been unlisted, even if all other actions have been paused. For example, a seizure may still be initiated by another listed market.

Could you please provide more information on the decision to exclude these actions from being confirmed to be paused before unlisting the market? This allows such actions to never be set to paused, even after a market is unlisted, although the three actions would still be disabled from other checks made.

## Recommendation

We recommend providing more information on the decision to exclude these actions from being confirmed to be paused before unlisting the market. This allows such actions to never be set to paused, even after a market is unlisted, although the three actions would still be disabled from other checks made.

## Alleviation

`[CertiK, 03/21/2024]` : The client made changes resolving the finding in commits 144cb9761cc3da4215dda2240b68d939c2e586f7 and 36ee37bd94b291a66685e633bba5c5136ce03a3c.

# VPB-04 | DISCREPANCY BETWEEN USE OF BORROW CAP IN CORE VS. ISOLATED POOLS

| Category | Severity | Location | Status |
|---|---|---|---|
| Coding Style | ● Informational | contracts/Comptroller.sol (PR349-Base): 517~519; contracts/Comptroller/Diamond/facets/PolicyFacet.sol (PR438-Base): 134~135 | ● Resolved |

## Description

There is a difference in how the borrow cap is used between the Isolated Pools Comptroller and the Core pool Comptroller.

In the Isolated Pools Comptroller, the function `preBorrowHook()` requires that `nextTotalBorrows` is less or equal to the `borrowCap` . Within the Core pool Comptroller, function `borrowAllowed()` requires that `nextTotalBorrows` is strictly less than the `borrowCap` .

## Recommendation

We recommend reviewing the discrepancy and deciding whether one of the codebases should be updated to be consistent with the other.

## Alleviation

`[CertiK, 03/21/2024]` : The client made changes resolving the finding in commit f283bd3712f6bd38e0d753f55706b5e481da4161.

# VPB-05 | TYPOS AND INCONSISTENCIES

| Category | Severity | Location | Status |
|---|---|---|---|
| Inconsistency | ● Informational | contracts/Comptroller.sol (PR349-Base): 206, 207; contracts/Comptroller/ComptrollerStorage.sol (PR438-Base): 185; contracts/Comptroller/Diamond/facets/MarketFacet.sol (PR429-Base): 137 | ● Resolved |

## Description

### PR-429 MarketFacet

- In the comments above `unlistMarket()` it has "Unlist an market" instead of "Unlist a market".

### PR-349 Comptroller

- In the comments above `unlistMarket()` it has "Unlist an market" instead of "Unlist a market".
- In the comments above `unlistMarket()` it has "@dev Pauses all actions, sets borrow/supply caps to 0 and sets collateral factor to 0." However, the function checks these and does not set them.

### PR4-39 ComptrollerStorage

- The comments above `borrowCaps` states "Defaults to zero which corresponds to unlimited borrowing.", however, the functionality has changed so that zero corresponds to borrowing being disallowed.

## Recommendation

We recommend fixing the typos and inconsistencies mentioned above.

## Alleviation

`[CertiK, 03/21/2024]` : The client made changes resolving the finding in commits

- f038e3edbfae79b2c5fa9eafbf8f7f2b1767cec7
- 52236a321e8be625f9aeb14568c1357e4273f48c
- 086c073fa31df3cf34971b255843dff6232e8dd7

# OPTIMIZATIONS | VENUS - UNLIST MARKET & BORROW CAP

| ID | Title | Category | Severity | Status |
|---|---|---|---|---|
| CVP-01 | Consider Using Custom Errors | Gas Optimization | Optimization | ● Resolved |

# CVP-01 | CONSIDER USING CUSTOM ERRORS

| Category | Severity | Location | Status |
|---|---|---|---|
| Gas Optimization | ● Optimization | contracts/Comptroller.sol (PR349-Base): 220~230 | ● Resolved |

## Description

From Solidity `v0.8.4` , there are more gas-efficient ways to explain to users why an operation failed than through strings. Using custom errors can significantly reduce the size of the deployed bytecode and reduce the gas cost when calls revert.

## Recommendation

We recommend considering the use of custom errors to reduce gas costs. In addition, throughout the codebase custom errors and string errors are used, can you please clarify the convention being followed for when custom vs. string errors are used.

## Alleviation

`[CertiK, 03/21/2024]` : The client made changes resolving the finding in commit 9fd7543e8dd541f4a29c8cf76ef574ec573a3f7a.

# APPENDIX | VENUS - UNLIST MARKET & BORROW CAP

## Finding Categories

| Categories | Description |
|---|---|
| Gas Optimization | Gas Optimization findings do not affect the functionality of the code but generate different, more optimal EVM opcodes resulting in a reduction on the total gas cost of a transaction. |
| Coding Style | Coding Style findings may not affect code behavior, but indicate areas where coding practices can be improved to make the code more understandable and maintainable. |
| Language Version | Language Version findings indicate that the code uses certain compiler versions or language features with known security issues. |
| Inconsistency | Inconsistency findings refer to different parts of code that are not consistent or code that does not behave according to its specification. |
| Centralization | Centralization findings detail the design choices of designating privileged roles or other centralized controls over the code. |
| Design Issue | Design Issue findings indicate general issues at the design level beyond program logic that are not covered by other finding categories. |

## 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.