



USDV

Security Assessment

CertiK Assessed on Jun 5th, 2025





Certik Assessed on Jun 5th, 2025

USDV

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

Executive Summary

TYPES

ERC-20

ECOSYSTEM

Ethereum (ETH) | Tron (TRX)

METHODS

Manual Review, Static Analysis

LANGUAGE

Solidity

TIMELINE

Delivered on 06/05/2025

KEY COMPONENTS

N/A

CODEBASE

Client provided .sol file added to [https://github.com/CertiKProject/certik-audit-](https://github.com/CertiKProject/certik-audit-projects/tree/f97bc446f19084991995916cb368f3fbae4c9390/projects/us)[projects/tree/f97bc446f19084991995916cb368f3fbae4c9390/projects/us](https://github.com/CertiKProject/certik-audit-projects/tree/f97bc446f19084991995916cb368f3fbae4c9390/projects/us)[View All in Codebase Page](#)

Highlighted Centralization Risks

Contract upgradeability

Transfers can be paused

Privileged role can mint tokens

Has blacklist/whitelist

Vulnerability Summary



4

Total Findings

3

Resolved

0

Partially Resolved

1

Acknowledged

0

Declined

1 Centralization

1 Acknowledged

Centralization findings highlight privileged roles & functions and their capabilities, or instances where the project takes custody of users' assets.

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 may include logical errors that, under specific circumstances, could result in fund losses or loss of project control.

0 Medium

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

■ 0 Minor

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 | USDV

I **Summary**

[Executive Summary](#)

[Vulnerability Summary](#)

[Codebase](#)

[Audit Scope](#)

[Approach & Methods](#)

I **Review Notes**

[Overview](#)

[Deployment](#)

[v1](#)

[v2](#)

[v3](#)

[ETH Sepolia](#)

[Tron Nile](#)

[External Dependencies](#)

[Addresses](#)

[Privileged Functions](#)

I **Findings**

[USD-02 : Centralization Risks in USDV.sol](#)

[USD-03 : Missing Zero Address Validation](#)

[USD-04 : Inconsistent Solidity Versions](#)

[USD-05 : Missing Interface Implementation](#)

I **Appendix**

I **Disclaimer**

CODEBASE | USDV


Repository

Client provided .sol file added to <https://github.com/CertiKProject/certik-audit-projects/tree/f97bc446f19084991995916cb368f3fbae4c9390/projects/usdv>

Update code files were added to <https://github.com/CertiKProject/certik-audit-projects/tree/a5c10c1433b53caba4869fe5aad23d916ac3f3c4/projects/usdv>

AUDIT SCOPE | USDV

1 file audited ● 1 file without findings

ID	Repo	File	SHA256 Checksum
● USD	CertiKProject/certik-audit-projects	 contracts/USDV.sol	a61e8fa76d104cf9afab107d28a275ff24d97865fd98c94e700e98248d777028

APPROACH & METHODS | USDV

This report has been prepared for USDV to discover issues and vulnerabilities in the source code of the USDV 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 | USDV

Overview

The **USDV** is a ERC20 token project with minting and pause functionality. The files currently under review include:

- UDSV.sol

Deployment

The contract **USDV** has been deployed on ETH Sepolia and Tron Nile before the audit is started.

v1

ETH Sepolia

- Proxy contract address: [0x8e3e346F760EA6945b1b52A13181A537Ad974F8a](#)
- Implementation contract address: [0xEE04d66582AF39Be5c21E86F43b2D65AB9F8fb05](#)

Tron Nile

- Proxy contract address: [TJ6orrcYLTyYymzzaGDN32JAdyXbyXpCXL](#)
- Implementation contract address: [TEMcLfbg5Y7UBocM8B7yQ2rNyxDTX39mEm](#)

v2

The contract **USDV** has been redeployed on ETH Sepolia and Tron Nile to fix bugs after the audit at 06/05/2025

ETH Sepolia

- Proxy contract address: [0x8e3e346F760EA6945b1b52A13181A537Ad974F8a](#)
- Implementation contract address: [0x486957fafab7c62c7b78d593dd008a9c04b22397](#)

Tron Nile

- Proxy contract address: [TPYjmXgxazXBjpThatzP3vQvp4V63iRrL3](#)
- Implementation contract address: [TCR7d72NGiRHMaXzPfwit1bgTCVL8K9EJg](#)

v3

The contract **USDV** has been redeployed on ETH Sepolia and Tron Nile to refactor the `decreaseAllowance` and `increaseAllowance` functions at 06/05/2025

ETH Sepolia

- Proxy contract address: [0x4adf851343db2FC959c25a860C6f9dF2824e45eE](#)
- Implementation contract address: [0x570a408ea92f11105643c191a0f16b7728f8e689](#)

Tron Nile

- Proxy contract address: [TVbMcDt7us2b4U6NqogDZyth3W7HDC5Pmm](#)
- Implementation contract address: [TAzTmE6X4wh8rUdXb5YcFCLtxKiDbwBjS](#)

External Dependencies

In **USDV**, the module inherits or uses a few of the depending injection contracts or addresses to fulfill the need of its business logic. 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.

Addresses

The following addresses interact at some point with specified contracts, making them an external dependency. All of the following values are initialized either at deployment time or by specific functions in smart contracts.

USDV

- defaultAdmin
- pauser
- minter
- upgrader
- blacklistAdmin

We assume these contracts or addresses are valid and non-vulnerable actors and implementing proper logic to collaborate with the current project.

Also, the following library/contract are considered as the third-party dependencies:

- @openzeppelin/contracts-upgradeable/@v5.3.0
- @openzeppelin/contracts/@v5.3.0

Privileged Functions

In the **USDV** project, the privileged roles are adopted to ensure the dynamic runtime updates of the project, which are specified in the **Centralization** finding.

The advantage of those privileged roles 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 the potential drawbacks of these functions, which should be clearly stated through the client's action/plan. Additionally, if the private keys of the privileged accounts are compromised, it could lead to devastating consequences for the project.

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 the **Timelock** contract.

FINDINGS | USDV



4

Total Findings

0

Critical

1

Centralization

0

Major

0

Medium

0

Minor

3

Informational

This report has been prepared to discover issues and vulnerabilities for USDV. Through this audit, we have uncovered 4 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:

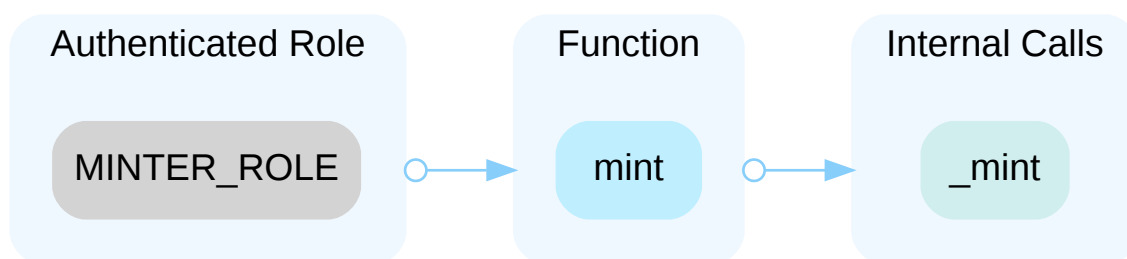
ID	Title	Category	Severity	Status
USD-02	Centralization Risks In USDV.Sol	Centralization	Centralization	● Acknowledged
USD-03	Missing Zero Address Validation	Volatile Code	Informational	● Resolved
USD-04	Inconsistent Solidity Versions	Language Version	Informational	● Resolved
USD-05	Missing Interface Implementations	Coding Issue	Informational	● Resolved

USD-02 | CENTRALIZATION RISKS IN USDV.SOL

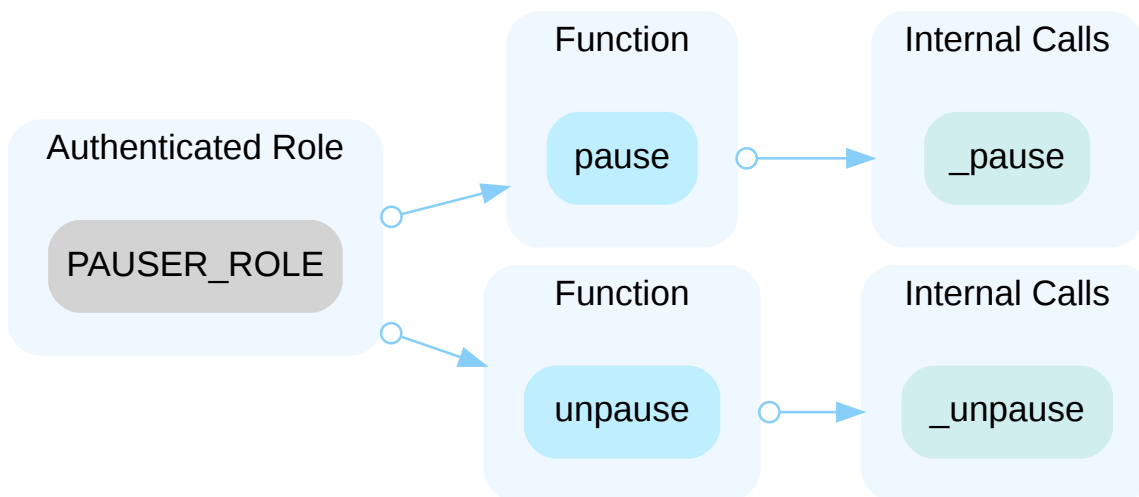
Category	Severity	Location	Status
Centralization	● Centralization	contracts/USDV.sol (0603): 47, 51, 55, 70	● Acknowledged

Description

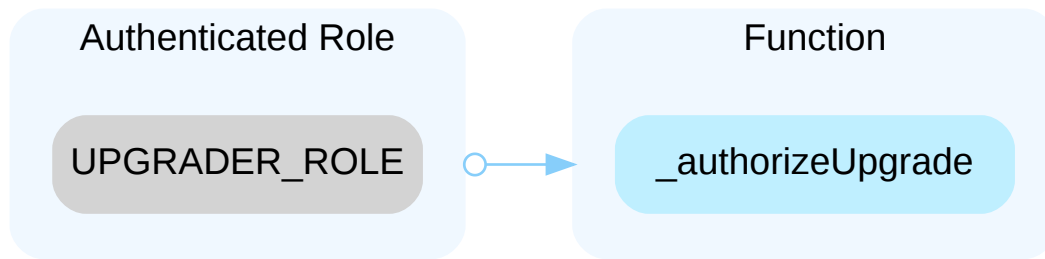
In the contract `USDV`, the role `MINTER_ROLE` has authority over the functions shown in the diagram below and can update the token balance of an arbitrary account without sanity restriction. Any compromise to the `MINTER_ROLE` account may allow the hacker to take advantage of this authority and manipulate users' balances.



In the contract `USDV`, the role `PAUSER_ROLE` has authority over the functions shown in the diagram below. Any compromise to the `PAUSER_ROLE` account may allow the hacker to take advantage of this authority and pause contract execution or unpause the contract.



In the contract `USDV`, the role `UPGRADER_ROLE` has authority over the functions shown in the diagram below. Any compromise to the `UPGRADER_ROLE` may allow a hacker to take advantage of this authority and change the implementation contract which is pointed by proxy and therefore execute potential malicious functionality in the implementation contract..



In the contract `AccessControlUpgradeable` inherited by `USDV`, the role `BLACKLIST_ADMIN_ROLE` and `DEFAULT_ADMIN_ROLE` has authority over the function below:

- `grantRole/revokeRole`

Any compromise to the `BLACKLIST_ADMIN_ROLE` and `DEFAULT_ADMIN_ROLE` may allow a hacker to take advantage of this authority and add/remove role members.

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 (2/3, 3/5) 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.

I Alleviation

[USDV, 06/05/2025]: The team acknowledged the issue and plan to implement a multisignature management scheme similar to USDT's approach, utilizing Safe multi-signature wallets on the mainnet to govern all privileged roles. The specific configuration is as follows:

1. `DEFAULT_ADMIN_ROLE`

Multisig mechanism: 5/7 threshold

2. Other Privileged Roles (`MINTER/PAUSER/UPGRADER`)

Multisig mechanism: 3/5 threshold Independent signer pool: different with `DEFAULT_ADMIN`

3. Transparency Measures

Public multisig contract address Real-time publication of multisig transactions

4. Disaster Recovery

All signer addresses secured in cold wallets 3 backup management zones Private key sharding compliant with Shamir Secret Sharing (SSS) standard

USD-03 | MISSING ZERO ADDRESS VALIDATION

Category	Severity	Location	Status
Volatile Code	● Informational	contracts/USDV.sol (0603): 28~45	● Resolved

Description

The cited address input is missing a check that it is not `address(0)`.

Recommendation

We recommend adding a check the passed-in address is not `address(0)` to prevent unexpected errors.

Alleviation

[USDV, 06/05/2025]: The team heeded the advice and resolved the issue by adding checks. The new USDV contract is deployed at [ETH Sepolia](#) and [Tron Nile](#).

USD-04 | INCONSISTENT SOLIDITY VERSIONS

Category	Severity	Location	Status
Language Version	● Informational	contracts/USDV.sol (0603): 3	● Resolved

Description

The codebase contains multiple Solidity versions, which can lead to unexpected behavior, potential vulnerabilities, difficulties in maintaining the code, and inconsistencies in the execution of the smart contract. Using different versions may also result in increased complexity during code auditing, as different security features and bug fixes are present in different versions of the compiler.

Versions used: `^0.8.20`, `^0.8.22`, `^0.8.27`

`^0.8.27` is used in contracts/USDV.sol file.

```
3 pragma solidity ^0.8.27;
```

Recommendation

It is recommended to standardize on a single, up-to-date Solidity version throughout the codebase to ensure consistent behavior, benefit from the latest security features, and improve maintainability.

Alleviation

[USDV, 06/05/2025]: The team heeded the advice and resolved the issue by standardized the Solidity version during compilation:

1. Ethereum chain: Using Hardhat with
Solidity 0.8.28
2. TRON chain: Using TronBox with
Solidity 0.8.23.

The new USDV contract is deployed at [ETH Sepolia](#) and [Tron Nile](#).

USD-05 | MISSING INTERFACE IMPLEMENTATION

Category	Severity	Location	Status
Coding Issue	● Informational	contracts/USDV.sol (0603): 16~112	● Resolved

Description

The contract `USDV` does not implement the interface `IBeacon`, which reduces readability.

`USDV` implements the interface `IBeacon`, but does not inherit from it.

```
16 contract USDV is Initializable, ERC20Upgradeable, ERC20BurnableUpgradeable,  
    ERC20PausableUpgradeable, AccessControlEnumerableUpgradeable,  
    ERC20PermitUpgradeable, UUPSUpgradeable {  
17 //...  
18     function implementation() external view returns (address) {  
19         return ERC1967Utils.getImplementation();  
20     }  
21 }
```

```
9 interface IBeacon {
```

Recommendation

It is advised to implement the missing interface in the contract to ensure proper functionality and increase readability.

Alleviation

[USDV, 06/05/2025]: The team heeded the advice and resolved the issue by implementing `IBeacon` interface. The new USDV contract is deployed at [ETH Sepolia](#) and [Tron Nile](#).

APPENDIX | USDV

Finding Categories

Categories	Description
Language Version	Language Version findings indicate that the code uses certain compiler versions or language features with known security issues.
Coding Issue	Coding Issue findings are about general code quality including, but not limited to, coding mistakes, compile errors, and performance issues.
Volatile Code	Volatile Code findings refer to segments of code that behave unexpectedly on certain edge cases and may result in vulnerabilities.
Centralization	Centralization findings detail the design choices of designating privileged roles or other centralized controls over the 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.

Elevating Your Entire **Web3** Journey

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.

