Worldland - audit

Security Assessment

CertiK Assessed on Feb 12th, 2026

CertiK Assessed on Feb 12th, 2026

Worldland - audit

The security assessment was prepared by CertiK.

Executive Summary

TYPES	ECOSYSTEM	METHODS
ERC-20	EVM Compatible	Manual Review, Static Analysis

LANGUAGE	TIMELINE
Solidity	Preliminary comments published on 02/10/2026 Final report published on 02/12/2026

Vulnerability Summary

8	2	0	6	0
Total Findings	Resolved	Partially Resolved	Acknowledged	Declined

🟧	2	Centralization	2 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.
🟨	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	4 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 | WORLDLAND - AUDIT

APPROACH & METHODS | WORLDLAND - AUDIT

This audit was conducted for Worldland to evaluate the security and correctness of the smart contracts associated with the Worldland - audit project. The assessment included a comprehensive review of the in-scope smart contracts. The audit was performed using a combination of Manual Review and Static Analysis.

The review process emphasized the following areas:

- Architecture review and threat modeling to understand systemic risks and identify design-level flaws.
- Identification of vulnerabilities through both common and edge-case attack vectors.
- Manual verification of contract logic to ensure alignment with intended design and business requirements.
- Dynamic testing to validate runtime behavior and assess execution risks.
- Assessment of code quality and maintainability, including adherence to current best practices and industry standards.

The audit resulted in findings categorized across multiple severity levels, from informational to critical. To enhance the project's security and long-term robustness, we recommend addressing the identified issues and considering the following general improvements:

- Improve code readability and maintainability by adopting a clean architectural pattern and modular design.
- Strengthen testing coverage, including unit and integration tests for key functionalities and edge cases.
- Maintain meaningful inline comments and documentations.
- Implement clear and transparent documentation for privileged roles and sensitive protocol operations.
- Regularly review and simulate contract behavior against newly emerging attack vectors.

REVIEW NOTES | WORLDLAND - AUDIT

Overview

Worldland is a project that includes its token and TGE contracts. The scope of this audit is as follows:

- MyToken.sol
- RevocableStairVesting.sol
- VestingWallet.sol
- VestingWalletCliff.sol
- VestingWalletStair.sol
- WorldLandStairVesting.sol

External Dependencies

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.

WorldLandNativeToken

- `foundation_wallet`

RevocableStairVesting

- `_revoker`
- `_treasury`

VestingWalletStair

- `owner`

VestingWallet

- `owner`

VestingWalletCliff

- `owner`

WorldLandStairVesting

- `owner`

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/
- vesting_contracts/access
- vesting_contracts/interfaces
- vesting_contracts/token
- vesting_contracts/utils

Privileged Functions

In the **Worldland** project, privileged roles are adopted to ensure the dynamic runtime updates of the project, which are specified in the **Centralization Risks** findings below.

The advantage of those privileged roles in the codebase is that the client reserves the ability to adjust the protocol according to the runtime requirements 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 a `Timelock` contract.

FINDINGS | WORLDLAND - AUDIT

8	0	2	0	0	2	4
Total Findings	Critical	Centralization	Major	Medium	Minor	Informational

This report has been prepared for Worldland to identify potential vulnerabilities and security issues within the reviewed codebase. During the course of the audit, a total of 8 issues were identified. Leveraging a combination of Manual Review & Static Analysis the following findings were uncovered:

ID	Title	Category	Severity	Status
WOA-02	**Initial Token Distribution**	**Centralization**	**Centralization**	● **Acknowledged**
WOA-03	**Centralization Related Risks**	**Centralization**	**Centralization**	● **Acknowledged**
WOA-04	Missing Zero Address Validation	Volatile Code	Minor	● Resolved
WOA-05	Step-Duration Validation Is Inconsistent With “First Step Unlocks Immediately”	Logical Issue	Minor	● Resolved
WOA-01	Discussion On Adding Tokens After Revocation In `RevocableStairVesting`	Logical Issue	Informational	● Acknowledged
WOA-06	Constructor Does Not Chain-Call The Inherited Contract Constructor	Logical Issue	Informational	● Acknowledged
WOA-07	`VestingWalletCliff` Releases All Tokens Accumulated Linearly During The Cliff Period At The Cliff	Logical Issue	Informational	● Acknowledged
WOA-08	Revocation Timestamp Can Be Set Without Revoking Funds	Coding Style	Informational	● Acknowledged

WOA-02 | Initial Token Distribution

Category	Severity	Location	Status
Centralization	● **Centralization**	**token_contracts/contracts/MyToken.sol (init-a3bdca): 11~12**	● **Acknowledged**

Description

All of the `WorldLandNativeToken` tokens are sent to the externally-owned account (EOA) address. This is a centralization risk because the owner(s) of the EOA can distribute tokens without obtaining the consensus of the community. Any compromise to these addresses may allow a hacker to steal and sell tokens on the market, resulting in severe damage to the project.

Recommendation

It is recommended that the team be transparent regarding the initial token distribution process. The token distribution plan should be published in a public location that the community can access. The team should make efforts to restrict access to the private keys of the deployer account or EOAs. A multi-signature (⅔, ⅗) wallet can be used to prevent a single point of failure due to a private key compromise. Additionally, the team can lock up a portion of tokens, release them with a vesting schedule for long-term success, and deanonymize the project team with a third-party KYC provider to create greater accountability.

Alleviation

[Worldland, 02/12/2026]: The team acknowledged this issue and stated that they will address the issue in the future, which will not be included in this audit engagement.

[CertiK, 02/12/2026]: It is suggested to implement the aforementioned methods to avoid centralized failure. Also, CertiK strongly encourages the project team to periodically revisit the private key security management of all addresses related to centralized roles.

WOA-03 | Centralization Related Risks

Category	Severity	Location	Status
Centralization	● **Centralization**	**vesting_contracts/finance/RevocableStairVesting.sol (init-a3bdca): 22, 106, 129; vesting_contracts/finance/VestingWallet.sol (init-a3bdca): 36; vesting_contracts/finance/VestingWalletCliff.sol (init-a3bdca): 14; vesting_contracts/finance/VestingWalletStair.sol (init-a3bdca): 20; vesting_contracts/finance/WorldLandStairVesting.sol (init-a3bdca): 23**	● **Acknowledged**

Description

In the contract `RevocableStairVesting` , the role `revoker` has authority over the following functions:

- `revoke()`
- `revoke(address token)`

If a `revoker` account is compromised, an attacker could repeatedly revoke all unvested ETH/ERC20 balances and drain them into the treasury.

The `RevocableStairVesting` , `WorldLandStairVesting` , `VestingWalletStair` , `VestingWalletCliff` and `VestingWallet` contract inherit from `Ownable` . As a result, the contract owner also has authority over the following functions:

- `transferOwnership()`
- `renounceOwnership()`

If an `owner` account is compromised, an attacker could reassign or abandon ownership.

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

[Worldland, 02/12/2026]: The team acknowledged this issue and stated that they will address the issue in the future, which will not be included in this audit engagement.

[CertiK, 02/12/2026]: It is suggested to implement the aforementioned methods to avoid centralized failure. Also, CertiK strongly encourages the project team to periodically revisit the private key security management of all addresses related to centralized roles.

WOA-04 | Missing Zero Address Validation

Category	Severity	Location	Status
Volatile Code	● Minor	vesting_contracts/finance/RevocableStairVesting.sol (init-a3bdca): 71, 72	● 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

[Worldland, 02/12/2026]: The team heeded the advice and resolved the issue by adding zero address check in commit e2274f94ce9bda80ce3068d25e5c11c9495243dc

WOA-05 Step-Duration Validation Is Inconsistent With “First Step Unlocks Immediately”

Category	Severity	Location	Status
Logical Issue	● Minor	vesting_contracts/finance/VestingWalletStair.sol (init-a3bdca): 66~68, 121~123	● Resolved

Description

In `VestingWalletStair` , according to the logic of `_vestingSchedule` , since the first tranche of tokens is vested immediately when the cliff ends, token vesting actually occurs at the very beginning of the `_stepDuration` .

`VestingWalletStair.sol`

```
121         // Calculate how many complete steps have passed since cliff
122         // Add 1 so the first step vests immediately when cliff ends
123         uint64 stepsCompleted = (timeSinceCliff / _stepDuration) + 1;
```

Therefore, at the start of the last `_stepDuration` —that is, at the end of the previous `_stepDuration` —vesting has already been completed. As a result, in the example of a 1-year cliff plus 4 quarterly steps, vesting should already be completed at the end of the 3rd quarterly step.

However, the constructor check fully counts all vesting periods, meaning it assumes that vesting for a 1-year cliff plus 4 quarterly steps is completed only at the end of the 4th quarterly step. This leads to an inconsistency between the check and the intended design.

`VestingWalletStair.sol`

```
66          if (cliffSeconds + (stepDuration * numberOfSteps) > durationSec) {
67              revert InvalidStepConfiguration(stepDuration, numberOfSteps,
 durationSec);
68          }
```

Recommendation

It is recommended to modify the check to:

```
        if (cliffSeconds + (stepDuration * (numberOfSteps - 1)) > durationSec) {
            revert InvalidStepConfiguration(stepDuration, numberOfSteps,
durationSec);
        }
```

Alleviation

[Worldland, 02/12/2026]: The team heeded the advice and resolved the issue by modifying the check in commit e2274f94ce9bda80ce3068d25e5c11c9495243dc

WOA-01 | Discussion On Adding Tokens After Revocation In `RevocableStairVesting`

Category	Severity	Location	Status
Logical Issue	● Informational	vesting_contracts/finance/RevocableStairVesting.sol (init-a3bdca): 22	● Acknowledged

Description

After tokens are revoked, if new tokens are injected into the contract for any reason, these tokens will not be fully releasable. They will be releasable to the beneficiary only in proportion to the percentage of the total vesting period that had elapsed at the time of the first revocation, and the remaining portion would need to be withdrawn by calling `revoke()` again. We would like to confirm with the team whether this behavior is intended by design.

Recommendation

It is recommended to consider allowing all tokens held by the contract to become fully reclaimable once `revoke()` has been executed.

Alleviation

[Worldland, 02/12/2026]: The RevokableStairVestingContract, after being revoked, is not intended to be used at all.

Although it would be a better design to make it possible to reclaim all of the tokens after revoke is called.

Since we are short on time I won't make nay changes for the current version.

WOA-06 | Constructor Does Not Chain-Call The Inherited Contract Constructor

Category	Severity	Location	Status
Logical Issue	● Informational	vesting_contracts/finance/VestingWalletCliff.sol (init-a3bdca): 26~31; vesting_contracts/finance/VestingWalletStair.sol (init-a3bdca): 50~73	● Acknowledged

Description

The constructor does not chain-call the constructor of the inherited `VestingWallet` contract. As a result, `duration()` and `start()` in `VestingWalletCliff` and `VestingWalletStair` remain set to 0. If either of these contracts is inherited without calling the `VestingWallet` constructor, `duration()` and `start()` will be incorrectly initialized.

Recommendation

If necessary, the `VestingWallet` constructor should be invoked in `VestingWalletStair` and `VestingWalletCliff` , and the constructors of `WorldLandStairVesting` and `RevocableStairVesting` should be updated accordingly to accommodate this change.

Alleviation

[Worldland, 02/11/2026]: The team acknowledged the issue and decided not to implement the recommended change in the current engagement.

WOA-07 | `VestingWalletCliff` Releases All Tokens Accumulated Linearly During The Cliff Period At The Cliff

Category	Severity	Location	Status
Logical Issue	● Informational	vesting_contracts/finance/VestingWalletCliff.sol (init-a3bdca): 52	● Acknowledged

Description

`VestingWalletCliff` does not release any tokens before the cliff. However, once the cliff is reached, it directly releases all tokens that were linearly accumulated during the cliff period, rather than starting linear release from zero.

Recommendation

We would like to remind the team to be aware of this behavior when using this contract.

Alleviation

[Worldland, 02/11/2026]: The team acknowledged the issue and decided not to implement the recommended change in the current engagement.

WOA-08 | Revocation Timestamp Can Be Set Without Revoking Funds

Category	Severity	Location	Status
Coding Style	● Informational	vesting_contracts/finance/RevocableStairVesting.sol (init-a3bdca): 108~110	● Acknowledged

Description

The `revoke()` functions are meant to take back any unvested ETH or tokens and send them to the treasury, while leaving already vested amounts to the beneficiary.

However, the revocation timestamp is set even when no funds are actually revoked, which freezes the vesting time without moving any assets.

Recommendation

Only update the revocation timestamp when `returnable > 0`, or explicitly handle the zero-amount case to avoid freezing the vesting schedule.

Alleviation

[Worldland, 02/11/2026]: The team acknowledged the issue and decided not to implement the recommended change in the current engagement.

APPENDIX | WORLDLAND - AUDIT

Audit Scope

cryptoecc/WorldLand_Contracts
token_contracts/contracts/MyToken.sol
vesting_contracts/finance/RevocableStairVesting.sol
vesting_contracts/finance/VestingWallet.sol
vesting_contracts/finance/VestingWalletCliff.sol
vesting_contracts/finance/VestingWalletStair.sol
vesting_contracts/finance/WorldLandStairVesting.sol
token_contracts/contracts/MyToken.sol
vesting_contracts/finance/RevocableStairVesting.sol
vesting_contracts/finance/VestingWallet.sol
vesting_contracts/finance/VestingWalletCliff.sol
vesting_contracts/finance/VestingWalletStair.sol
vesting_contracts/finance/WorldLandStairVesting.sol

Finding Categories

Categories	Description
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.
Volatile Code	Volatile Code findings refer to segments of code that behave unexpectedly on certain edge cases and may result in vulnerabilities.
Logical Issue	Logical Issue findings indicate general implementation issues related to the program logic.

Categories	Description
Centralization	Centralization findings detail the design choices of designating privileged roles or other centralized controls over the code.

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 Web3 Journey

Founded in 2017 by leading academics in the field of Computer Science from both Yale and Columbia University, CertiK is the largest 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.