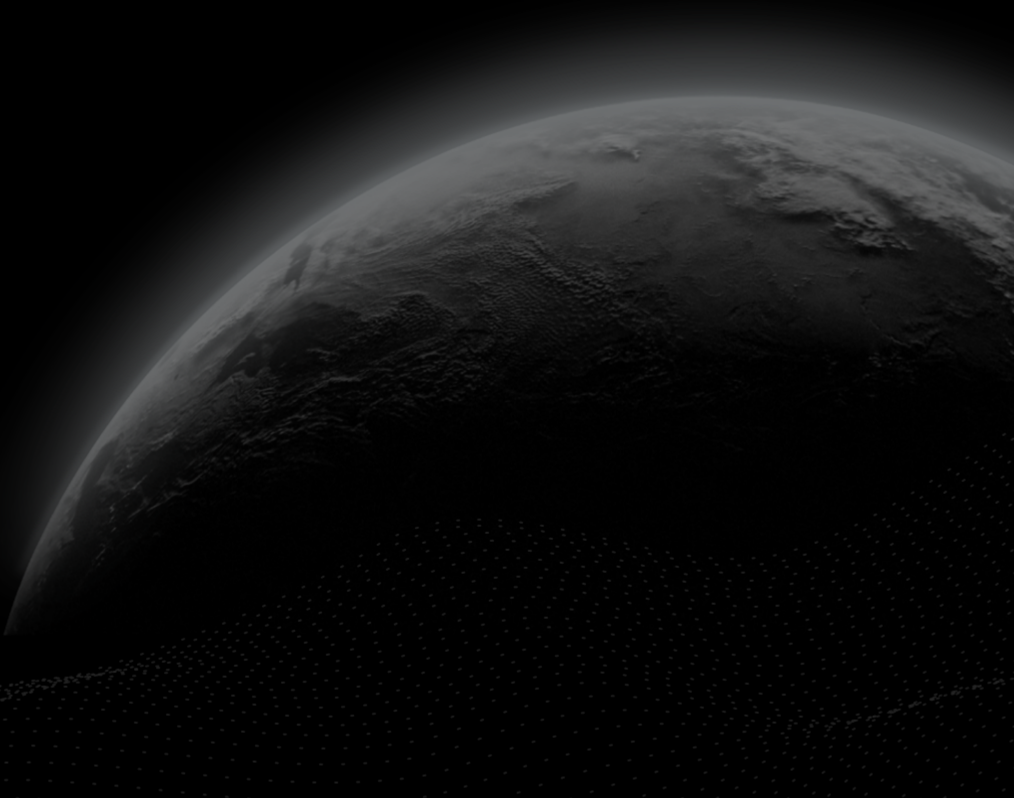
CERTIK

Security Assessment

FlyWallet

CertiK Verified on Mar 7th, 2023



CertiK Verified on Mar 7th, 2023

FlyWallet

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

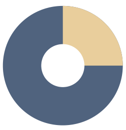
Executive Summary

TYPES	ECOSYSTEM	METHODS
DeFi, Wallet	Celo | Polygon	Manual Review, Static Analysis

LANGUAGE	TIMELINE	KEY COMPONENTS
Solidity	Delivered on 03/07/2023	N/A

CODEBASE	COMMITS
https://github.com/flywallet-io/TravelSaver/tree/main/contracts	d593d2a9a738062d724e6ce02de93eecfeb5e950
...View All	...View All

Vulnerability Summary



4	4	0	0	0	0	0
Total Findings	Resolved	Mitigated	Partially Resolved	Acknowledged	Declined	Unresolved

Count	Severity	Status	Description
0	Critical		Critical risks are those that impact the safe functioning of a platform and must be addressed before launch. Users should not invest in any project with outstanding critical risks.
0	Major		Major risks can include centralization issues and logical errors. Under specific circumstances, these major risks can lead to loss of funds and/or control of the project.
0	Medium		Medium risks may not pose a direct risk to users' funds, but they can affect the overall functioning of a platform.
1	Minor	1 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.
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 | FLYWALLET

Summary

- Executive Summary
- Vulnerability Summary
- Codebase
- Audit Scope
- Approach & Methods

Decentralization Efforts

- Description
- Recommendations
  - Short Term:
  - Long Term:
  - Permanent:
- Status/Alleviations

Findings

- TST-02 : Unchecked ERC-20 `transfer()`/`transferFrom()` Call
- TST-01 : Incompatibility with Deflationary Tokens
- TST-03 : External Call Inside Loop
- TST-05 : No check for operator Plan ID

Optimizations

- TST-04 : User-Defined Getters
- TST-07 : Redundant code

Appendix

Disclaimer

CODEBASE | FLYWALLET

Repository

https://github.com/flywallet-io/TravelSaver/tree/main/contracts

Commit

d593d2a9a738062d724e6ce02de93eecfeb5e950

AUDIT SCOPE | FLYWALLET

1 file audited • 1 file with Resolved findings

ID	File	SHA256 Checksum
• TST	contracts/TravelSaver.sol	da8ce62439369e9bcdb7630674bab455becf3e82cab434c2adcac0dda12c800e

APPROACH & METHODS | FLYWALLET

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

DECENTRALIZATION EFFORTS | FLYWALLET

Description

In the contract `TravelSaver` the account `operatorWallet` has authority over the function(s) shown as below.

- function claimTravelPlan(): allows to transfer ERC20 tokens from specific TravelPlan to the operators wallet

Any compromise to the privileged account may allow the hacker to take advantage of this authority.

Recommendations

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.

Status/Alleviations

`[CertiK]` : The team heeded our advice and deployed the operatorWallet as a multi-sign wallet.

1.Celo

TravelSaver contract:

https://explorer.celo.org/mainnet/address/0x46c4F585B1948f21E733C5e08e55330de22f9119/read-contract#address-tabs

operatorWallet:

https://explorer.celo.org/mainnet/address/0x2e7997BaF30435d70b5a2EC3eA334975b16C5204/contracts#address-tabs

⅔ signers:

mainnet:0xaBB8f1cf22488eDf86aBA09557e372CEf44B2aD9
mainnet:0xD8891D6DF73C084F2b92c74a86Beb65eBF831F3C
mainnet:0x5A5c853d313070907884206375a1dea7F1871842

2.Polygon

TravelSaver contract:

https://polygonscan.com/address/0x207856B02b264b7C60fdE304658d683184254330#code

operatorWallet:

https://polygonscan.com/address/0x383bc9eae0dfaec56d10a12baf23603701a4a004#code

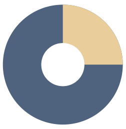
⅔ signers:

matic:0xaBB8f1cf22488eDf86aBA09557e372CEf44B2aD9 matic:0x8D1eD48beecC201ada45Da98D35918733833cf04
matic:0xCD3f903924ad0438DbBeB614eD526E3C4332A4d4

`[FlyWallet]` : We acknowledge the case of the operator wallet private keys being compromised hence contracts operator wallet address provided in the constructor will be a gnosis multisig with a at least 2 x hardware wallets to minimize such a risk.

Once the user made a claim, hence funds were transferred out of the contract, it is the operator's responsibility to either provide the flight booking or process a manual refund by customer service. Opeator then will convert funds into fiat in order to make associated flight provider fees.

FINDINGS | FLYWALLET



4	0	0	0	1	3
Total Findings	Critical	Major	Medium	Minor	Informational

This report has been prepared to discover issues and vulnerabilities for FlyWallet . 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
TST-02	Unchecked ERC-20 `transfer()` / `transferFrom()` Call	Volatile Code	Minor	Resolved
TST-01	Incompatibility With Deflationary Tokens	Logical Issue	Informational	Resolved
TST-03	External Call Inside Loop	Control Flow	Informational	Resolved
TST-05	No Check For Operator Plan ID	Logical Issue	Informational	Resolved

TST-02 | UNCHECKED ERC-20 `transfer()` / `transferFrom()` CALL

Category	Severity	Location	Status
Volatile Code	Minor	contracts/TravelSaver.sol: 307, 327, 475	Resolved

Description

The return value of the transfer()/transferFrom() call is not checked.

```
307         token.transferFrom(msg.sender, address(this), amount);
```

```
327         token.transfer(operatorWallet, value);
```

```
475         token.transferFrom(caller, address(this), amount);
```

Recommendation

Since some ERC-20 tokens return no values and others return a `bool` value, they should be handled with care. We advise using the OpenZeppelin's `SafeERC20.sol` implementation to interact with the `transfer()` and `transferFrom()` functions of external ERC-20 tokens. The OpenZeppelin implementation checks for the existence of a return value and reverts if `false` is returned, making it compatible with all ERC-20 token implementations.

Alleviation

The team heeded our advice and resolved the issue in commit 7d67a623bb4d4bfab36584c177bfe9e284abaf24.

TST-01 | INCOMPATIBILITY WITH DEFLATIONARY TOKENS

Category	Severity	Location	Status
Logical Issue	Informational	contracts/TravelSaver.sol: 305, 307, 326, 327, 472, 475	Resolved

Description

When transferring deflationary ERC20 tokens, the input amount may not be equal to the received amount due to the charged transaction fee. For example, if a user sends 100 deflationary tokens (with a 10% transaction fee), only 90 tokens actually arrived to the contract. However, a failure to discount such fees may allow the same user to withdraw 100 tokens from the contract, which causes the contract to lose 10 tokens in such a transaction.

Reference: https://thoreum-finance.medium.com/what-exploit-happened-today-for-gocerberus-and-garuda-also-for-lokum-ybear-piggy-caramelswap-3943ee23a39f

```
307         token.transferFrom(msg.sender, address(this), amount);
```

- Transferring tokens by `amount`.

```
305         plan.contributedAmount += amount;
```

- The `amount` appears to be used for bookkeeping purposes without compensating the potential transfer fees.

```
327         token.transfer(operatorWallet, value);
```

- Transferring tokens by `value`.

```
326         plan.contributedAmount -= value;
```

- The `value` appears to be used for bookkeeping purposes without compensating the potential transfer fees.

```
475         token.transferFrom(caller, address(this), amount);
```

- Transferring tokens by `amount`.

```
472         plan.contributedAmount += amount;
```

- The `amount` appears to be used for bookkeeping purposes without compensating the potential transfer fees.

Recommendation

We advise the client to regulate the set of tokens supported and add necessary mitigation mechanisms to keep track of accurate balances if there is a need to support deflationary tokens.

Alleviation

The team acknowledged this issue and they stated this is by design. Deflationary tokens will not be accepted when deploying the contract, only mainstream stable coins will be accepted.

TST-03 | EXTERNAL CALL INSIDE LOOP

Category	Severity	Location	Status
Control Flow	Informational	contracts/TravelSaver.sol: 475, 499~500	Resolved

Description

External calls are made inside a *for* loop. This might lead to a denial-of-service attack. If any of the calls fail, it will cause the entire loop to revert.

```
475         token.transferFrom(caller, address(this), amount);
```

```
499             token.balanceOf(sender) >= amountToTransfer &&
500             token.allowance(sender, address(this)) >= amountToTransfer
```

```
410     function runIntervals(uint256[] memory IDs) external {
411         for (uint256 i = 0; i < IDs.length; i++) {
412             _fulfillPaymentPlanInterval(IDs[i]);
413         }
414     }
```

Recommendation

We recommend using the pull-over-push strategy for external calls.

Alleviation

The team heeded our advice and resolved the issue in commit 61c6fd3215577322de9825b7b9ed37069db8997e.

TST-05 | NO CHECK FOR OPERATOR PLAN ID

Category	Severity	Location	Status
Logical Issue	Informational	contracts/TravelSaver.sol: 268	Resolved

Description

In the function `createTravelPlan()`, there is no check whether the `operatorPlanID` has been used or not. This brings a question whether the contract allows to create multiple TravelPlan for same operatorPlanID?

Recommendation

We would like to confirm with the client whether the current implementation aligns with the original project design.

Alleviation

`[FlyWallet ]`: The current implementation aligns with the original project design allowing operators to create unique and multiple TravelPlans as operatorPlanID is an optional ID referencing operators themselves.

OPTIMIZATIONS | FLYWALLET

ID	Title	Category	Severity	Status
TST-04	User-Defined Getters	Gas Optimization	Optimization	Resolved
TST-07	Redundant Code	Coding Style	Optimization	Resolved

TST-04 | USER-DEFINED GETTERS

Category	Severity	Location	Status
Gas Optimization	Optimization	contracts/TravelSaver.sol: 203~209, 216~222	Resolved

Description

The linked functions are equivalent to the compiler-generated getter functions for the respective variables.

Recommendation

We advise that the linked variables are instead declared as `public` as compiler-generated getter functions are less prone to error and much more maintainable than manually written ones.

Alleviation

The team heeded our advice and resolved the issue in commit 251daf2cf67846542fe41d74bc7099e60ebcebb5.

TST-07 | REDUNDANT CODE

Category	Severity	Location	Status
Coding Style	Optimization	contracts/TravelSaver.sol: 352, 365	Resolved

Description

The `totalAmount` value has been calculated on Line 352 once and stored in the variable `totalToTransfer`, it's not necessary to calculate it again.

Recommendation

We recommend to use the variable `totalToTransfer` to avoid redundant calculations.

Alleviation

The team heeded our advice and resolved the issue in commit aac0ec94cd640e5275e129a5d51306984e1566f4.

APPENDIX | FLYWALLET

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.
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.
Control Flow	Control Flow findings concern the access control imposed on functions, such as owner-only functions being invoke-able by anyone under certain circumstances.
Volatile Code	Volatile Code findings refer to segments of code that behave unexpectedly on certain edge cases that may result in a vulnerability.
Coding Style	Coding Style findings usually do not affect the generated byte-code but rather comment on how to make the codebase more legible and, as a result, easily maintainable.

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.

