Lucky Train

Security Assessment

CertiK Assessed on Sept 10th, 2025

CertiK Assessed on Sept 10th, 2025

Lucky Train

The security assessment was prepared by CertiK.

Executive Summary

TYPES	ECOSYSTEM	METHODS
Jetton	TON (TON)	Manual Review

LANGUAGE	TIMELINE
FunC	Preliminary comments published on 09/10/2025
	Final report published on 09/10/2025

Vulnerability Summary

8	7	1	0	0	0
Total Findings	Resolved	Multi-Sig	Partially Resolved	Acknowledged	Declined

1	Centralization	1 Multi-Sig	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.
1	Major	1 Resolved	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.
4	Minor	4 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.
2	Informational	2 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 | LUCKY TRAIN

CODEBASE | LUCKY TRAIN

Repository

https://gitlab.com/lucky-train/smart-contract/-/tree/9db7fa3d240c854cdc8c12d89ccaeba6c85e1a9c

https://gitlab.com/lucky-train/smart-contract/-/tree/2366e4307b4d3aeb062bf023782f694186671ba6/

https://gitlab.com/lucky-train/smart-contract/-/tree/82b8251eeee0e58c8b20eb7d056cdc727ff7cf4f/contracts/

https://gitlab.com/lucky-train/smart-contract/-/tree/b86451a91230bb02a6e8e8393ef816c2958150a3/contracts/

Commit

9db7fa3d240c854cdc8c12d89ccaeba6c85e1a9c

2366e4307b4d3aeb062bf023782f694186671ba6

82b8251eeee0e58c8b20eb7d056cdc727ff7cf4f

b86451a91230bb02a6e8e8393ef816c2958150a3

Audit Scope

The file in scope is listed in the appendix.

APPROACH & METHODS | LUCKY TRAIN

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

FINDINGS | LUCKY TRAIN

8	0	1	1	0	4	2
Total Findings	Critical	Centralization	Major	Medium	Minor	Informational

This report has been prepared for Lucky Train 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 the following findings were uncovered:

ID	Title	Category	Severity	Status
LUT-02	**Centralization Related Risks**	**Centralization**	**Centralization**	**2/3 Multi-Sig**
LUT-03	Incorrect Gas Handling Leads To Lost Stake	Incorrect Calculation	Major	Resolved
LUT-04	Missing `end_parse()` Validation Allows Unchecked References In `ticketParams`	Logical Issue	Minor	Resolved
LUT-05	Unsafe Handling Of Malformed `jetton::transfer_notification` Can Lead To Permanent Jetton Loss	Logical Issue	Minor	Resolved
LUT-06	Usage Of Same `op` For Different Messages Breaks Message Parsing	Coding Issue	Minor	Resolved
LUT-07	Incorrect Gas Handling In `op::withdrawTeam` Handler	Volatile Code	Minor	Resolved
LUT-08	Presence Of `TODO` Comments And Commented Code	Coding Style	Informational	Resolved
LUT-09	Typos	Coding Style	Informational	Resolved

LUT-02 | Centralization Related Risks

Category	Severity	Location	Status
Centralization	**Centralization**	**contracts/staking.fc (base): 7**	**2/3 Multi-Sig**

Description

In the contract `staking` the role `storage::owner` has authority over the functions

- op::debug::setall, op::debug::setcode
- op::withdrawTeam
- op::setDistributeData, op::setPaused, op::setManager, op::removeManager, op::setTicketParams, op::changeOwner
- op::setJettonWallet

Any compromise to the `storage::owner` account may allow the hacker to take advantage of this authority and

- extract all the fees
- change the contract code to extract all the locked funds
- change staking parameters
- before `op::setJettonWallet` message the `storage::owner` can send fake `op::jetton::transfer_notification` message and break contract invariants

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

[Lucky Train, 07/01/2025]: The team acknowledged the issue and decided to use https://multisig.ton.org/.

[CertiK, 09/10/2025]:

The team has deployed the staking contract at address EQB9M1I392BBx5SLTzv56GKwYLnhB9IrZAxjWg3BjI_LUCKY.

The multisig address EQARY2MymBnGVU8Z1Yt1H0HrE3JDLN23W6jInKCN42WpjXCQ was deployed with:

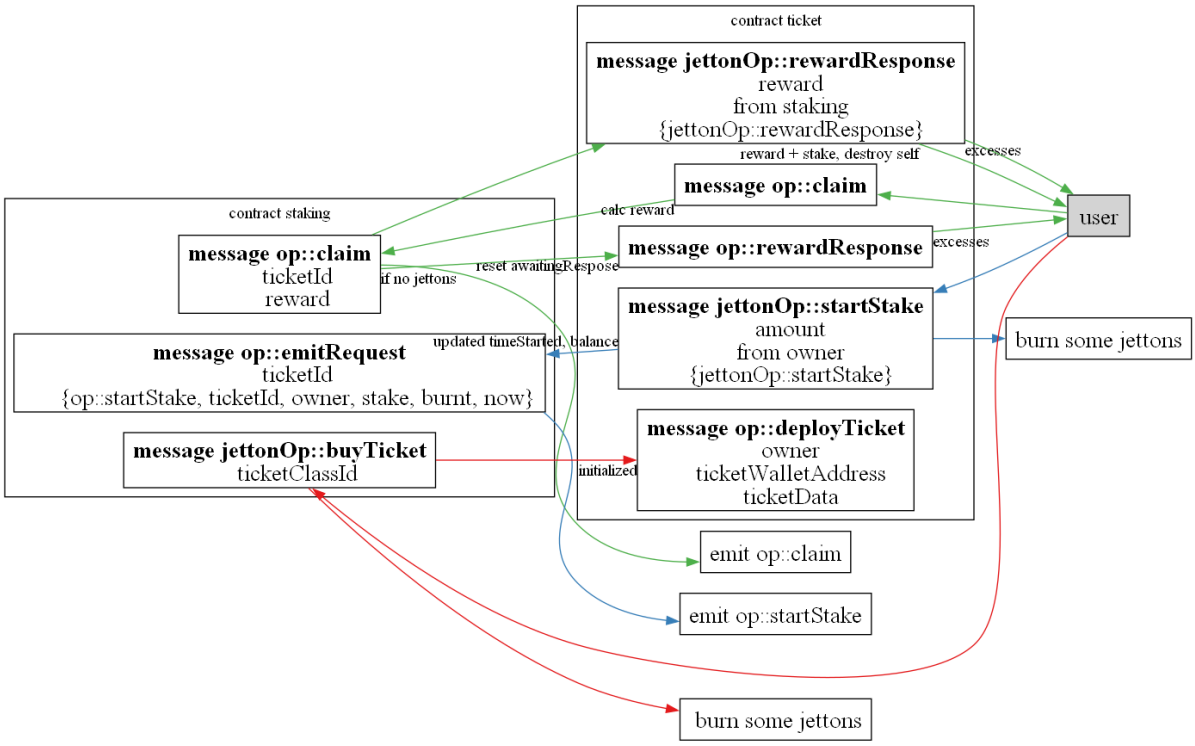
1. Threshold: 2/3
2. Signers: #1 — UQBfdHfP72dR6i6e7qQL4n3X-wBklH7S6NN8tjAVAgcSGxUA, #2 — UQBV45tJRLF9clr1ZnPmIFEIj8q0p3N0NfXwpT9MOH21D4Pu, #3 — UQBjIpXzToDtO1OkJMeR9ufezWreytHwuL4Kc-U2Sf_3aznX

At the transaction 15e6469369347a543305277f0d596d5334882d5bd7f6e3ef89422981bb597601, the owner of the staking contract was assigned to the multisig address EQARY2MymBnGVU8Z1Yt1H0HrE3JDLN23W6jInKCN42WpjXCQ.

LUT-03 | Incorrect Gas Handling Leads To Lost Stake

Category	Severity	Location	Status
Incorrect Calculation	Major	contracts/ticket.fc (base): 272	Resolved

Description




```
271 msg::send(
272     10000000,  ;; Fixed gas amount
273     storage::staking,
274     begin_cell()
275         .store_uint(op::claim, 32)
276         .store_uint(ctx::query_id, 64)
277         .store_uint(storage::ticketId, 64)
278         .store_coins(reward)
279         .end_cell(),
280     SEND_MODE_CARRY_ALL_REMAINING_MESSAGE_VALUE  ;; Forward all remaining
281 );
```

The code snippet shows a `msg::send()` operation that both specifies a fixed gas amount (10000000) and uses `SEND_MODE_CARRY_ALL_REMAINING_MESSAGE_VALUE`. This creates a contradiction in gas handling behavior.

The `SEND_MODE_CARRY_ALL_REMAINING_MESSAGE_VALUE` flag indicates that all remaining message value should be forwarded, making the explicit gas amount redundant. The operation will spend the contract own balance and can fail in case of lack of gas.

`op::buyTicket` handler in staking contract requires 0.045 tons of gas. However, it

1. sends 0.020 with `op::deployTicket`
2. spends tons for 2 messages forwarding and 1 event emitting
3. spends tons for `op::buyTicket` processing
4. sends 0.031 with `op::burn`

Burn operation returns 0.025 tons as excesses, however, staking contract should not rely on that, since the jetton can have custom burning logic.

`op::claim` handler in ticket contract requires 0.100 tons of gas. However,

1. it sends 0.106 with `op::claim` to staking contract (more, than received)
2. staking contract transfers reward jettons with 0.096 tons attached, with **only 0.036 tons** with `op::rewardResponse` `transfer_notification`
3. the excesses from this sending are returned to staking contract. It is expected to be returned to the ticket owner.
4. 0.036 tons with `op::rewardResponse` `transfer_notification` is not enough to process the message by ticket contract, since it sends jettons to the user with 0.045 tons attached.
5. As a result, the claim process can fail with not enough tons leaving the ticket contract with `storage::awaitingRespose` set.

Scenario

To reproduce the failed claim scenario:

1. Edit in tests/StakingE2E.spec.ts the scenario 'should buy ticket and claim in one year'
2. At L820 use `blockchain.now = blockchain.now! + 1.5*oneYear + 3600`
3. The actual result: `jettonOp::rewardResponse` handler in ticket contract keeps all jettons and stays in `awaitingRespose` state.

Recommendation

Remove the explicit gas amount (10000000) when using `SEND_MODE_CARRY_ALL_REMAINING_MESSAGE_VALUE` to align with the expected behavior of forwarding all remaining value.

Carefully check all the gas requirements.

LUT-04 Missing `end_parse()` Validation Allows Unchecked References In `ticketParams`

Category	Severity	Location	Status
Logical Issue	Minor	contracts/staking.fc (base): 455~458	Resolved

Description

```
456  if (ticketParams.slice_bits() > 0) {
457      throw(err::staking::invalidTicketData);
458  }
```

The current validation only checks for remaining bits in `ticketParams` using `slice_bits()`, but fails to verify that the slice contains no references. This could allow malformed data with references to pass validation.

Using `end_parse()` would properly validate that:

1. No bits remain unparsed
2. No cell references remain unparsed
3. The entire ticket data structure has been properly consumed

Recommendation

Replace the `slice_bits()` check with `end_parse()`.

LUT-05 Unsafe Handling Of Malformed `jetton::transfer_notification` Can Lead To Permanent Jetton Loss

Category	Severity	Location	Status
Logical Issue	Minor	contracts/staking.fc (base): 131; contracts/ticket.fc (base): 95	Resolved

Description

The system fails to properly handle cases where the `jetton::transfer_notification` is in an incorrect format. When parsing fails due to malformed input, jettons become permanently locked as the contract doesn't implement a recovery mechanism for this edge case.

Specifically the jettons are locked in cases:

1. If the `forward_payload` is empty.
2. If jettons are of unexpected type (not from `storage::jettonWallet`).
3. If `ctx::jettonOp` has unexpected value.
4. If jettons arrive not from `storage::owner` / `storage::staking`.
5. If `storage::timeStarted` is in unexpected state.

Recommendation

Use try-catch pattern to handle parsing failures gracefully and return jettons back.

LUT-06 Usage Of Same `op` For Different Messages Breaks Message Parsing

Category	Severity	Location	Status
Coding Issue	Minor	contracts/ticket.fc (base): 169, 192, 275	Resolved

Description

The code uses the same `op` s (like `op::rewardResponse`) for different message types with different formats. This creates ambiguity in message handling and prevents proper automatic parsing by blockchain explorers and indexers.

Each message type should have a unique `op` to ensure clear differentiation and reliable parsing. Currently, using the same `op` for multiple purposes makes it impossible for external systems to distinguish between different message types programmatically.

```
191    begin_cell()
192        .store_uint(op::jetton::excesses, 32)  ;; Operation
193        .end_cell(),
```

According to TEP-074 standard `op::jetton::excesses` message is supposed to contain `query_id` field.

Recommendation

Assign unique operation IDs to each distinct message type. For example:

- `op::rewardResponseOk`
- `op::rewardResponseFail`

LUT-07 | Incorrect Gas Handling In `op::withdrawTeam` Handler

Category	Severity	Location	Status
Volatile Code	Minor	contracts/staking.fc (base): 291	Resolved

Description

`op::withdrawTeam` handler of staking contract sends `op::jetton::transfer` message with 0.042 tons attached. But it doesn't check if enough gas was provided. As a result, own contract balance can be spent.

Recommendation

We recommend using `SEND_MODE_CARRY_ALL_REMAINING_MESSAGE_VALUE` or ensuring enough gas was provided.

LUT-08 | Presence Of `TODO` Comments And Commented Code

Category	Severity	Location	Status
Coding Style	Informational	contracts/imports/messages.fc (base): 41; contracts/imports/utils.fc (base): 35~36; contracts/staking.fc (base): 114, 327, 514; contracts/ticket.fc (base): 181~182, 217	Resolved

Description

The code contains `TODO` comments (`;; TODO: calculate gas`), commented-out code, and debug functionality. This indicates unfinished work and should be addressed before deployment.

```
35  ;; global int storage::isInited;           ;; 2 is inited
36  ;; global int storage::isStarted;          ;; 2 is started
```

"2" is supposed to be "1". `isStarted` is supposed to be `awaitingResponse` .

```
181                 1,                      ;; Query ID
```

"Query ID" is supposed to be "Forward TON amount".

Recommendation

Ensure all development artifacts are cleaned up before production deployment

LUT-09 | Typos

Category	Severity	Location	Status
Coding Style	Informational	contracts/ticket.fc (base): 13	Resolved

Description

The variable name `awaitingRespose` is supposed to be `awaitingResponse`.

Recommendation

We recommend fixing the typos.

OPTIMIZATIONS | LUCKY TRAIN

ID	Title	Category	Severity	Status
LUT-01	Constants Can Be Used Instead Of `PUSHINT`	Gas Optimization	Optimization	Resolved

LUT-01 | Constants Can Be Used Instead Of `PUSHINT`

Category	Severity	Location	Status
Gas Optimization	Optimization	contracts/imports/utils.fc (base): 3; contracts/jtnimports/constants.fc (base): 2~14	Resolved

Description

```
3  int workchain() asm "0 PUSHINT";
```

According to the documentation, numeric constants are substituted during compilation, so all optimization and pre-computations performed during the compilation are successfully performed (unlike the old method of defining constants via inline asm `PUSHINT` s).

Recommendation

We recommend declaring the constants.

APPENDIX | LUCKY TRAIN

Audit Scope

lucky-train/smart-contract
contracts/imports/messages.fc
contracts/imports/utils.fc
contracts/jtnimports/constants.fc
contracts/staking.fc
contracts/ticket.fc
contracts/imports/constants.fc
contracts/imports/stdlib.fc
contracts/jtnimports/discovery-params.fc
contracts/jtnimports/jetton-utils.fc
contracts/jtnimports/op-codes.fc
contracts/jtnimports/params.fc
contracts/jtnimports/stdlib.fc
contracts/jtnimports/utils.fc
contracts/xbrc20token.fc

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.

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.
Coding Issue	Coding Issue findings are about general code quality including, but not limited to, coding mistakes, compile errors, and performance issues.
Incorrect Calculation	Incorrect Calculation findings are about issues in numeric computation such as rounding errors, overflows, out-of-bounds and any computation that is not intended.
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.
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.