Code Security Assessment

Stader Liquid Token

Jan 7th, 2022

Table of Contents

Summary

This report has been prepared for Stader to discover issues and vulnerabilities in the source code of the Stader Liquid Token project as well as any contract dependencies that were not part of an officially recognized library. A comprehensive examination has been performed, utilizing Static Analysis and 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:

- 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.

Overview

Project Summary

Project Name	Stader Liquid Token
Platform	Terra
Language	Rust
Codebase	https://github.com/stader-labs/stader-liquid-token/
Commit	e2561633859cc846bea024854b61248dd6f28665

Audit Summary

Delivery Date	Jan 07, 2022
Audit Methodology	Static Analysis, Manual Review
Key Components	airdrops-registry, reward, staking

Vulnerability Summary

Vulnerability Level	Total	Pending	Declined	Acknowledged	Partially Resolved	Resolved
Critical	0	0	0	0	0	0
Major	1	0	0	1	0	0
Medium	0	0	0	0	0	0
Minor	0	0	0	0	0	0
Informational	5	0	0	4	0	1
Discussion	0	0	0	0	0	0

Audit Scope

ID	File	SHA256 Checksum
CKP	contracts/airdrops-registry/src/contract.rs	a12f946871267f6d3be2964177ff31a29f0dceb9c5137e284602748cb031caf0
ERR	contracts/airdrops-registry/src/error.rs	48f64958ab657cca1a4014ffe80e6c6acee5e2d5382fafa2e01d06bf14ff14c5
LIB	contracts/airdrops-registry/src/lib.rs	f19dd9f92dbdcfaf94e19491336606a46a930e767da1b49b0ce6c775020b061e
MSG	contracts/airdrops-registry/src/msg.rs	00bd80e0e0dc00234b10ccded608dcb2ade043344baf756f2660b40820a61786
STA	contracts/airdrops-registry/src/state.rs	e9f3c3ae908d876c8c56224e166d10f52bedbfb1e006e4deb25d3aba299b2053
CON	contracts/reward/src/contract.rs	4a317d7c767c636d874525acc4b39c7980269bf84b789668ee99120aae90ac3e
ERO	contracts/reward/src/error.rs	6dcaf0178dee860af8ae1d79f55d9a6ab87584df94430930c96f4238207650f0
LIS	contracts/reward/src/lib.rs	f19dd9f92dbdcfaf94e19491336606a46a930e767da1b49b0ce6c775020b061e
MSS	contracts/reward/src/msg.rs	8669bf3df9df74008618aa067251bc0ab0fcfbba7f579075bcafa015ad821809
STT	contracts/reward/src/state.rs	ce7d4028cf8f51dd455493d8056737762fd8a36a7c22642b9801952e05120ffa
COT	contracts/staking/src/contract.rs	a6ddf316bef4f34597073d2ca51f38042e68bf33528b8131bd18d944e9d5ba19
ERS	contracts/staking/src/error.rs	8ca6d9ccd6f3c43a7654291d4d7031a245c1a1e996e242d6cc83baaab22b04e2
HEL	contracts/staking/src/helpers.rs	96d78fb0be8b3efe5ed9edee6a39a6203789c347ab88cccc959217e909fe895a
LIR	contracts/staking/src/lib.rs	784398106541d077a9fa6c5f2c04547ebcd994a62d80dc1e2913293131afad68
MSR	contracts/staking/src/msg.rs	56d19f5bdbf8ab7c47546183cd0f9f052c590e019c548a7dc9d1b491304a3839
STE	contracts/staking/src/state.rs	7535c433e4c83e403bd3ad07d92f62009abf80dceee09aabc5f6e32fe69142fe

Review Notes

According to Stader Litepaper, Stader's V2 focuses the liquid staking token on the staked LUNA and adds cross-ecosystem integrations to power strategies on top of staking rewards and airdrops.

Stader aims to separate the base capital and the rewards with different contracts. This ensures that the base capital staked is always isolated from the interactions with other protocols.

Few Stader's core smart contracts:

1. Delegator Contract - Delegators' funds will be deposited and withdrawn from this contract.
2. Validator Contract - Stakes the delegator funds. Claims rewards and airdrops.
3. Pools Contract - Overseer of validator contract. Manages stake across each validator pool and supports multiple pools.
4. Strategies Contract - Leverages staking rewards and synthetic assets to interact/ integrate with other Defi/ Gaming protocols to amplify yields

Reference:

- Stader Litepaper

External Dependencies

There are a few depending injection contracts or addresses in the current project:

- `airdrop_contract_str`, `cw20_contract_str`, and `airdrop_token_str` for the contract `Airdrops-registry`.
- `staking_contract` and `reward_withdraw_contract` for the contract `Reward`.
- `airdrop_registry_contract`, `airdrop_withdrawal_contract`, `reward_contract`, `cw20_token_contract`, `val_addr`, `redel_addr`, `user_addr_str`, for the contract `Staking`.

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

Privileged Functions

Airdrops-registry

In the contract `Airdrops-registry`, the role `manager` has the authority over the following function:

- `update_airdrop_registry()` to update airdrop registry.

The `Airdrops-registry` contract does not hold the user information and fund. The `Staking` contract interacts by providing registered airdrop contracts.

Considering both `contracts.airdrop_contract` and `contracts.cw20_contract` are provided by the `Airdrops-registry` contract and their implementations are unknown, the `Staking` contract might be affected if an incorrect `airdrop_contract` or `airdrop_withdrawal_contract` is mistakenly registered. For example, if `MerkleAirdropMsg::Claim` transfers tokens to the Staking contract, `Cw20ExecuteMsg::Transfer` does not transfer tokens to `airdrop_withdrawal_contract`, the airdrop would not be converted to users' rewards, and thus users' rewards might be affected. However, users' principal and claimed rewards are not affected.

According to the Stader Litepaper, both airdrop and reward contracts only hold users extra bonuses. The user principles are isolated from these contracts.

Reward

In the contract `Reward,` the role `staking_contract` has the authority over the following functions:

- `swap()` to swap tokens to LUNA;
- `transfer()` to transfer tokens to another contract.

The role `manager` has the authority over the following function:

- `update_config()` to update configurations.

As the UTC Time 17:00 December 29th, 2021, the auditors confirmed that the `staking_contract` of the Reward contract deployed at `terra1d8cjkwxvrw8cmpyja3p8luag2hxsktersc0wez` is `terra1xacqx447msqp46qmv8k2sq6v5jh9fdj37az898`, which is the Staking contract with the following `InitMsg`:

```
{
  "min_deposit": "1000",
  "max_deposit": "100000000000",
  "reward_contract": "terra1d8cjkwxvrw8cmpyja3p8luag2hxsktersc0wez",
  "airdrops_registry_contract": "terra1vq83s69rykjypyqcqhc7hsqzups6p9fwzu0wre",
  "airdrop_withdrawal_contract": "terra1dykk536s6yvtek4hjyst6tflu5r7jf4ycvy4r5",
  "protocol_fee_contract": "terra1dykk536s6yvtek4hjyst6tflu5r7jf4ycvy4r5",
  "protocol_deposit_fee": "0",
  "protocol_reward_fee": "0.1",
  "protocol_withdraw_fee": "0.003",
  "unbonding_period": 1814400,
  "undelegation_cooldown": 259200,
  "swap_cooldown": 3600,
```

```
  "reinvest_cooldown": 3600
}
```

Meanwhile, the `manager` role is set to `terra1wcywgnfkrgufd5cgeq436lcjskgx2yxs3yhfqg`, which is an externally owned account. The `manager` role only has authority over the `update_config()` function to update configurations, so the auditors agree that no *direct* manager privileged message hinders base funds movement.

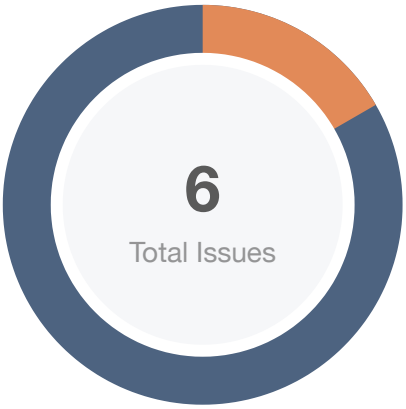
Staking

In the contract `Staking`, the role `manager` has the authority over the following functions:

- `update_config()` to update configurations;
- `add_validator()` to add a new validator;
- `remove_validator_from_pool()` to remove a validator;
- `rebalance_pool()` to rebalance the pool by redelegating between validators;
- `swap_rewards()` to execute the `Swap` message in the contract `config.reward_contract` without the restriction of cooldown;
- `reinvest()` to execute the `Transfer` message in the contract `config.reward_contract` and delegate the transferred tokens to a validator without the restriction of cooldown;
- `undelegate_stake()` to undelegated stakings without cooldown restriction.

To improve the trustworthiness of the project, dynamic runtime updates in the project should be notified to the community. Any plans to invoke the aforementioned functions should also be considered to move to the execution queue of the `Timelock` contract.

Findings



Critical	**0** (0.00%)
Major	**1** (16.67%)
Medium	**0** (0.00%)
Minor	**0** (0.00%)
Informational	**5** (83.33%)
Discussion	**0** (0.00%)

ID	Title	Category	Severity	Status
CON-01	Identical Blocks In `if-else` Statements	Control Flow	Informational	Acknowledged
COT-01	Centralization Risk	**Centralization / Privilege**	**Major**	Acknowledged
COT-02	Redundant Clone	Language Specific	Informational	Acknowledged
COT-03	Unnecessary Gas Consumption by `unwrap_or()`	Language Specific	Informational	Acknowledged
COT-04	Dependencies on CW20 Token Contract	Logical Issue	Informational	Resolved
HEL-01	Unnecessary Gas Consumption by `unwrap_or()`	Language Specific	Informational	Acknowledged

CON-01 | Identical Blocks In `if-else` Statements

Category	Severity	Location	Status
Control Flow	Informational	projects/stader-liquid-token/contracts/reward/src/contract.rs (b6b25fa): 135~141	Acknowledged

Description

The function `swap()` swaps the supported coins for `uluna` coins. The following code implements the same operation for `if` and `else if` parts:

```
135         if is_listed {
136             messages.push(create_swap_msg(coin, config.reward_denom.to_string()));
137         } else if query_exchange_rates(&deps, config.reward_denom.clone(), vec!
[coin.denom.clone()])
138             .is_ok()
139         {
140             messages.push(create_swap_msg(coin, config.reward_denom.to_string()));
141         }
```

Recommendation

It is recommended to merge the two conditions. For example,

```
135         if is_listed || query_exchange_rates(&deps, config.reward_denom.clone(),
vec![coin.denom.clone()]).is_ok() {
136             messages.push(create_swap_msg(coin, config.reward_denom.to_string()));
137         }
```

Alleviation

[**Stader Team**]: We choose not to address this as it has negligible gas implications and no performance/security implications.

COT-01 | Centralization Risk

Category	Severity	Location	Status
Centralization / Privilege	**Major**	projects/stader-liquid-token/contracts/staking/src/contract.rs (b6b25fa): 143, 203, 232, 288, 523, 546, 715	Acknowledged

Description

In the contract `Staking`, the role `manager` has the authority over the following functions:

- `update_config()` to update configurations;
- `add_validator()` to add a new validator;
- `remove_validator_from_pool()` to remove a validator;
- `rebalance_pool()` to rebalance the pool by redelegating between validators;
- `swap_rewards()` to execute the `Swap` message in the contract `config.reward_contract` without the restriction of cooldown;
- `reinvest()` to execute the `Transfer` message in the contract `config.reward_contract` and delegate the transferred tokens to a validator without the restriction of cooldown;
- `undelegate_stake()` to undelegate stakings without the restriction of cooldown.

Any compromise to the `manager` account may allow the hacker to take advantage of this and disrupt operations involving this contract.

Recommendation

We advise the client to carefully manage the `manager` account's private key to avoid any potential risks of being hacked. In general, we strongly recommend centralized privileges or roles in the protocol to 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 the different levels in terms of short-term and long-term:

- Time-lock with reasonable latency, e.g., 48 hours, for awareness on privileged operations;
- Assignment of privileged roles to multi-signature wallets to prevent a single point of failure due to the private key;
- Introduction of a DAO/governance/voting module to increase transparency and user involvement.

Alleviation

[**Stader Team**]: Stader uses a cw3 multi-sig contract to make the following contracts controlled by a multi-sig admin. The current setup used is a 3 on 5 multi-sig. I.e. three signatures needed on five eligible signatures. For the Staking contract - `terra1xacqx447msqp46qmv8k2sq6v5jh9fdj37az898`, the multi-sig admin deployed is `terra1alxgc922ylxp0lfk8vs7aqmc504430p9aum36m`.

Critical functions like `swap_rewards()`, `reinvest()` and `undelegate_stake()` are permissionless with a cooldown to prevent abuse.

[**CertiK**]: As the UTC Time 17:20 December 29th, 2021, the auditors confirmed that the admin of the Staking contract deployed at `terra1xacqx447msqp46qmv8k2sq6v5jh9fdj37az898` is `terra1alxgc922ylxp0lfk8vs7aqmc504430p9aum36m`, which a multi-sign contract with the following `InitMsg`:

```
{
  "max_voting_period": {
    "time": 604800
  },
  "required_weight": 3,
  "voters": [
    {
      "addr": "terra1mdr46s7nvftlchyruh6rags08qqtt470vceu4v",
      "weight": 1
    },
    {
      "addr": "terra14xgv060e0v0ww2ljelhy5598vwlmkwv8v4w6lq",
      "weight": 1
    },
    {
      "addr": "terra1z0wlt2affs5l7ljpzlsnk55w5x384v2te3wdq8",
      "weight": 1
    },
    {
      "addr": "terra139cugulh83wqq52aj7nt9vt4ss5ltu0lwh0d5g",
      "weight": 1
    },
    {
      "addr": "terra1dwylleadzwtz08hvesutuvmpxcw8yamtj2a5w8",
      "weight": 1
    }
  ]
}
```

Meanwhile, the `manager` role is set to `terra1wcywgnfkrgufd5cgeq436lcjskgx2yxs3yhfqg`, which is an externally owned account. The auditors agree that the functions `swap_rewards()`, `reinvest()` and `undelegate_stake()` are permissionless with a cooldown, while `update_config()`, `add_validator()`,

`remove_validator_from_pool()` and `rebalance_pool()` do not move funds into or out of the system *directly*. it should, however, still be noted that changing the contract parameters might affect the project's operation. For example, if `undelegation_cooldown` is set mistakenly to a huge number, users might not be able to trigger `undelegate_stake()` by themselves and wait for the manager to undelegate the stake.

[**Stader Team**]: There could be changes on Terra/Cosmos blockchain properties such as *Proposal to increase MaxEntries from 7 to 100*. We do not want cooldowns to be immutable to protect user funds long term.

The same reason applies to the redelegation of funds between validators (limits) and removing a validator. These are operations that cannot be made permissionless. Because if we do, it can abuse, and user funds would be locked.

The same is the case with `swap_rewards()`, it is entirely possible to abuse swap rewards every block and essentially convert all rewards to 0Luna.

Without a configurable cooldown, we might end up in a situation where attackers can continuously swap rewards and decrease Luna available in rewards to a minimal amount.

In short, we have ensured that mutable properties are set only to protect user funds from blockchain changes and to serve use cases of our clientele.

For example, some of our funds will need to swap our reward logic with a new strategy. Stader has built their contracts to support exchanges, institutions, and custodians, etc.

[**CertiK**]: The auditors understand the tradeoff between the centralization and performance restricted by external dependencies such as the blockchain properties in Terra/Cosmos protocol.

For example, to work well with the undelegation entries limitations on the Terra, the mechanism of queue-undelegated is introduced into the system, and the variable `undelegation_cooldown` is used to allow the undelegation to trigger by non-privileged roles. Besides `undelegation_cooldown`, the Stader team introduced `swap_cooldown` and `reinvest_cooldown` to allow users to trigger `swap_rewards()` and `reinvest()` without manager's operations after cooldowns. All these designs introduce, the Stader team would like to provide a better user experience, business scalability, and ecosystem building for the whole Stader community.

Reference: Stader Litepaper.

COT-02 | Redundant Clone

Category	Severity	Location	Status
Language Specific	Informational	projects/stader-liquid-token/contracts/staking/src/contract.rs (b6b25fa): 49, 93	Acknowledged

Description

The following variables are dropped without further use, so calling `clone()` is not necessary:

- `info.sender()` (Line 49)
- `env` (Line 93)

Recommendation

It is recommended to remove the redundant `clone()`s.

Alleviation

The Stader team heeded our advice and decided not to change the current codebase.

COT-03 | Unnecessary Gas Consumption by `unwrap_or()`

Category	Severity	Location	Status
Language Specific	Informational	projects/stader-liquid-token/contracts/staking/src/contract.rs (b6b25fa): 364, 370, 472, 572, 631, 796, 852, 905, 918	Acknowledged

Description

The following code uses `unwrap_or()` followed by a function call:

- `let mut val_meta = x.unwrap_or(VMeta::new());` (Line 364, 631)
- `unwrap_or(Uint128::zero());` (Line 370, 472, 572, 796, 852, 905, 918)

The functions in `unwrap_or()` will always be called and thus cause unnecessary gas consumption.

Recommendation

It is recommended to use `unwrap_or_else()` to avoid unnecessary gas consumption.

Alleviation

[**Stader Team**]: Because the gas optimization here is minor, we are choosing to address this issue at a later point of time. We see no significant gains in tx fees for user or security vulnerability with the recommended change.

COT-04 | Dependencies on CW20 Token Contract

Category	Severity	Location	Status
Logical Issue	Informational	projects/stader-liquid-token/contracts/staking/src/contract.rs (b6b25fa): 154	Resolved

Description

The Staking contract uses a CW20 token contract to represent the number of shares users stake. It should be guaranteed that the CW20 contract

- does not have minters other than the Staking contract;
- does not have tokens minted other than the Staking contract.

Moreover, according to the implementation, the `Receive` message can only be executed by the token contract. Considering token transfers do not happen in the function `receive_cw20()` or `queue_undelegation()` while token burns happen in the function `undelegate_stake()`. We assume the transfer logic is implemented in the CW20 token contract.

We would like to confirm with the Stader team that the logic implemented in the CW20 token contract collaborates with the Staking contract.

Alleviation

[**Stader Team**]: The token contract receives the cw20 and transfers them to the staking contract. This behavior on the already deployed contracts can be checked:

- CW20 - `terra17y9qkl8dfkeg4py7n0g5407emqnemc3yqk5rup`
- Staking contract - `terra1xacqx447msqp46qmv8k2sq6v5jh9fdj37az898`
- Airdrop registry - `terra1vq83s69rykjypyqcqhc7hsqzups6p9fwzu0wre`
- Reward contract - `terra1d8cjkwxvrw8cmpyja3p8luag2hxsktersc0wez`

[**CertiK**]: As the UTC Time 18:30 December 29th 2021, the auditors confirmed that the Stader LunaX Token contract deployed at `terra17y9qkl8dfkeg4py7n0g5407emqnemc3yqk5rup` has the following `InitMsg`:

```
{
  "name": "Stader LunaX Token",
  "symbol": "LunaX",
  "decimals": 6,
  "initial_balances": [],
  "mint": {
```

```
    "minter": "terra1xacqx447msqp46qmv8k2sq6v5jh9fdj37az898"
  }
}
```

Its minter is `terra1xacqx447msqp46qmv8k2sq6v5jh9fdj37az898`, which is the Staking contract with the following `InitMsg`:

```
{
  "min_deposit": "1000",
  "max_deposit": "100000000000",
  "reward_contract": "terra1d8cjkwxvrw8cmpyja3p8luag2hxsktersc0wez",
  "airdrops_registry_contract": "terra1vq83s69rykjypyqcqhc7hsqzups6p9fwzu0wre",
  "airdrop_withdrawal_contract": "terra1dykk536s6yvtek4hjyst6tflu5r7jf4ycvy4r5",
  "protocol_fee_contract": "terra1dykk536s6yvtek4hjyst6tflu5r7jf4ycvy4r5",
  "protocol_deposit_fee": "0",
  "protocol_reward_fee": "0.1",
  "protocol_withdraw_fee": "0.003",
  "unbonding_period": 1814400,
  "undelegation_cooldown": 259200,
  "swap_cooldown": 3600,
  "reinvest_cooldown": 3600
}
```

HEL-01 | Unnecessary Gas Consumption by `unwrap_or()`

Category	Severity	Location	Status
Language Specific	Informational	projects/stader-liquid-token/contracts/staking/src/helpers.rs (b6b25fa): 162, 175~176, 188~189	Acknowledged

Description

The following code uses `unwrap_or` followed by a function call:

- `let mut val_meta = x.unwrap_or(VMeta::new());` (Line 162, 175, 188)
- `unwrap_or(Uint128::zero());` (Line 176, 189)

The functions in `unwrap_or()` will always be called and thus cause unnecessary gas consumption.

Recommendation

It is recommended to use `unwrap_or_else()` to avoid unnecessary gas consumption.

Alleviation

[**Stader Team**]: Because the gas optimization here is minor, we are choosing to address this issue at a later point of time. We see no significant gains in tx fees for user or security vulnerability with the recommended change.

Appendix

Finding Categories

Centralization / Privilege

Centralization / Privilege findings refer to either feature logic or implementation of components that act against the nature of decentralization, such as explicit ownership or specialized access roles in combination with a mechanism to relocate funds.

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.

Language Specific

Language Specific findings are issues that would only arise within Solidity, i.e. incorrect usage of private or delete.

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

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.

About

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.