SOLIDIFIED

Audit Report for Cerealia - June 28, 2021

Summary

Audit Report prepared by Solidified covering the CEREAL token smart contracts.

Process and Delivery

Two (2) independent Solidified experts performed an unbiased and isolated audit of the code. The debrief on 17 June 2021.

Audited Files

The source code has been supplied in the form of a GitHub repository:

https://github.com/Cerealia/CEREAL-Utility-Token

Final Commit number: `f0fd17ba7f01c430f5aa9d437871e5bf7e9b1d65`

Intended Behavior

The smart contract implements a standard compliant ERC-20 token with a total supply of 500 million minted to the contract deployer.

Code Complexity and Test Coverage

Smart contract audits are an important step to improve the security of smart contracts and can find many issues. However, auditing complex codebases has its limits and a remaining risk is present (see disclaimer).

Users of a smart contract system should exercise caution. In order to help with the evaluation of the remaining risk, we provide a measure of the following key indicators: **code complexity**, **code readability**, **level of documentation**, and **test coverage**.

Note, that high complexity or lower test coverage does equate to a higher risk. Certain bugs are more easily detected in unit testing than a security audit and vice versa. It is, therefore, more likely that undetected issues remain if the test coverage is low or non-existent.

Criteria	Status	Comment
Code complexity	Low	-
Code readability and clarity	High	-
Level of Documentation	High	-
Test Coverage	High	The codebase is based entirely on OpenZeppelin's ERC-20 implementation and is therefore covered by OpenZeppelin's original testsuite.

Issues Found

Solidified found that the CEREAL token contract contains no critical issues, no major issuesfollowing, no minor issues, in addition to 2 informational notes.

We recommend all issues are amended, while the notes are up to the team's discretion, as they refer to best practices.

Issue #	Description	Severity	Status
1	Owner field in token contract may lead to confusion	Note	Resolved
2	Consider using the pragma range as the used OpenZeppelin library	Note	Resolved

SOLIDIFIED

Audit Report for Cerealia - June 28, 2021

Critical Issues

No critical issues have been found.

Major Issues

No major issues have been found.

Minor Issues

No minor issues have been found.

Informational Notes

1. Owner field in token contract may lead to confusion

The `cereal.sol` contract defines an owner variable, which is only used in the constructor to mint the total supply and is assigned to `msg.sender`.
However, the contract does not inherit from the `Ownable` contract, nor does it implement any owner-specific operations. Conventionally, the variable name owner is used for this context, so using it here may give users the impression that specific admin rights exist.

Recommendation
Since `owner` is not used outside the constructor it should be sufficient to directly mint to `msg.sender`.

2. Consider using the pragma range as the used OpenZeppelin library

The token allows all compiler versions greater than or equal to 0.8.0 by declaring the following pragma statement:

```
pragma solidity >=0.8.0;
```

This means that every future release of the Solidity compiler will compile this code. However, new major versions may have incompatible syntax changes.

Recommendation
Consider changing the declaration to match the OpenZeppelin code used, to limit the compiler version to only 0.8.x versions:

```
pragma solidity ^0.8.0;
```

Disclaimer

Solidified audit is not a security warranty, investment advice, or an endorsement of Cerealia or its products. This audit does not provide a security or correctness guarantee of the audited smart contract. Securing smart contracts is a multistep process, therefore running a bug bounty program as a complement to this audit is strongly recommended.

The individual audit reports are anonymized and combined during a debrief process, in order to provide an unbiased delivery and protect the auditors of Solidified platform from legal and financial liability.

Solidified Technologies Inc.