Audit report for Digital Reserve Currency (DRC)

1. Summary

DRC has engaged Decenter in the period starting on November 11th 2020 and ending on November 13th 2020 to assess and audit their platform's Solidity smart contracts. This document describes the issues discovered during the audit. Decenter's assessment is focused primarily on code review of the contract, with an emphasis on security, gas usage optimization and overall code quality.

2. Audit

2.1 Authenticity

The audited smart contract can be found on the Ethereum mainnet at https://etherscan.io/address/0xa150db9b1fa65b44799d4dd949d922c0a33ee606#code.

2.2 Scope

This audit covered only the smart contract already deployed on the Ethereum mainnet mentioned in the previous section.

2.3 Legend

🔴 High priority issues

🟠 Medium priority issues

🟡 Minor issues and optimisations

🟢 Notes and recommendations

3. Issues Found

---

1. Max allowance is not reduced by subsequent transfers

File name: DigitalReserveCurrency.sol;

In case a user provides a maximum (2^256) allowance to any `_spender` In this implementation of the ERC20 standard, the allowance will subsequently not be reduced with any future transfers.

This is not an issue in itself, just something that should be noted and known while designing the user interface.

2. Potential approval frontrun

File name: DigitalReserveCurrency.sol;

The ERC20 standard has a known attack vector for the `approve` function where the approved `_spender` can frontrun an approval update when the user tries to reduce it, for example, from 1000 to 500. The `_spender` can potentially frontrun that change and first withdraw 1000 and then another 500 after the approval update goes through.

To prevent such an attack vector, developers should make sure to create the user interface in such a way that they set the allowance first to 0 before setting it to another value for the same spender.

3. Older version of Solidity used

The latest stable version of Solidity at the time of writing this audit is 0.7.4, while the contract is currently using an older 0.4.26 version. We recommend updating to the latest stable release as it’s good practice to keep up with the latest developments in the Solidity language.

4. Audit

The audit makes no statements or warranties about utility of the code, safety of the code, suitability of business model, or any other statements about fitness of the contracts to purpose or their bug-free status.

5. Closing Summary

It is the conclusion of this review that the codebase does not contain any critical or other issues. The DRC token smart contract is well written and adheres to the ERC20 standard. Additionally, architecture of the DRC token creates a permanently fixed supply of the token and has no admin controls, nor it is upgradable in any other way, as intended by design. Aside from the remarks in this audit, the security of the DRC token smart contract is sufficient given the assumed requirements.