SOLIDIFIED

Audit Report for Aztec - May 11, 2022

Summary

Audit Report prepared by Solidified covering the Aztec protocol Ethereum Bridge contract for **Aave Bridge**.

The following report covers the **Aave Bridge**.

Process and Delivery

Three (3) independent Solidified experts performed an unbiased and isolated audit of the code. The debrief on 18 May 2022.

Audited Files

The source code has been supplied in the form of one public Github repository.

https://github.com/aztecProtocol/aztec-connect-bridges/

Commit Hash: `4a377651457e9ecf8c811e28b6a2570ef202f146`

```
src
|-- bridges
|   -- aave
      |-- AccountingToken.sol
      |-- imports
      |-- interfaces
      |-- lending
```

Intended Behavior

Smart contract responsible for depositing, managing and redeeming Defi interactions with the Aaave protocol by issuing an internal accounting token.

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 in 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	Medium	-
Test Coverage	High	-

Issues Found

Issue #	Description	Severity	Status
1	Notes on code improvement	Note	

Critical Issues

No issues found

Major Issues

No issues found

Minor Issues

No issues found

Informational Notes

1. Notes on code improvement

Simplify sanityConvert

The method `_sanityConvert` in `AaveLendingBridge.sol` checks the parameters of the convert method for their input and output types. Only valid parameters are allowed to call the convert method. The `_sanityConvert` can most likely be simplified if it only checks for valid configuration and otherwise revert.

Currently it is a mixture between what is allowed and what is forbidden.

Custom Error Types in finalise method
The `finalise` method is the only one which doesn't use custom error types.

Global Error Types for all bridges
All bridges could use the same error types for common errors. Like if the bridge is synchronous and the `finalise` method is not implemented.

Disclaimer

Solidified audit is not a security warranty, investment advice, or an endorsement of Aztec Protocol 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 from legal and financial liability.

Oak Security GmbH