SOLIDIFIED

Audit Report for Aztec - August 8, 2022

Summary

Audit Report prepared by Solidified covering the Aztec protocol.

The following report covers the **Subsidy Contract**.

Process and Delivery

Independent Solidified experts performed an unbiased and isolated audit of the code. The debrief on August 10 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: `80a8b3dbf1b2c5e302fc2613ec7588108269da34`

```
src
|-- aztec
|   -- Subsidy.sol
```

Intended Behavior

Smart contract responsible for managing subsidy for bridge interactions paid in ETH.

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	High	-
Test Coverage	High	-

SOLIDIFIED

Audit Report for Aztec - August 8, 2022

Issues Found

Issue #	Description	Severity	Status
1.	Reset of setGasUsageAndMinGasPerMinute can result in lost ETH	Minor	Resolved
2.	Dust or tiny leftover amounts in sub.available	Minor	Resolved
3.	MIN_SUBSIDY_VALUE might not be enough for expensive bridge interaction	Minor	Acknowledged
4.	It is not possible to change gasPerMinute of active subsidy	Note	
5.	Subsidizer needs to trust the bridge completely	Note	
6.	No withdraw or update/topUp for a active subsidy criteria	Note	
7	Additional Notes	Note	

SOLIDIFIED

Audit Report for Aztec - August 8, 2022

Critical Issues

No issues found

Major Issues

No issues found

Minor Issues

1. Reset of setGasUsageAndMinGasPerMinute can result in lost ETH

It is possible to call the `setGasUsageAndMinGasPerMinute` again and initialize it with different values.

The criteria could already be subsidized and the `available` would be reset to zero.

In that case the actual ETH amount would be stuck in the contract.

The correct criterial init in the Subsidy contract would be only once in the constructor. However to prevent potential incorrect usage from bridge implementation we recommend fixing it.

Recommendation
We would recommend not to call `setGasUsageAndMinGasPerMinute` again for an active criteria or not overwrite available with zero.

2. Dust or tiny leftover amounts in sub.available

It is not possible to topUp/update an active subsidy with more ETH until the `sub.available == 0.`

The actual subsidy ETH amount of an individual bridge call can't be calculated in advance because it depends on the `block.timestamp` and the gas price.

Therefore, it is very likely that the last subsidy bridge call can not subsidy the full `gasUsage`.

The last fully covered bridge call could result in a tiny leftover amount, which is not enough of an incentive to call the bridge the next time.

Recommendation
Allow to topUp the ETH amount of an active subsidy before available equals zero.

3. MIN_SUBSIDY_VALUE might not be enough for expensive bridge interaction

The `MIN_SUBSIDY_VALUE` should prevent front-running/blocking with a tiny subsidy amount for specific `criteria`.

Theoretically a single bridge interaction could be way more expensive due to high gas prices or a lot of gas usage.

In such a case the subsidy could be blocked with the `MIN_SUBSIDY_VALUE`.

Recommendation
Consider if the bridge should define the `MIN_SUBSIDY_VALUE` instead of a global variable for all bridges.

Informational Notes/Precaution

4. It is not possible to change gasPerMinute of active subsidy

The gasPerMinute rate might need multiple adjustments until it will result in the preferred interval between two bridge interactions.

Consider allowing the subsidizer to adjust the rate.

5. Subsidizer needs to trust the bridge completely

A subsidizer needs to trust the bridge implementation completely. Theoretically, a bridge could trigger the subsidy without a call from the Rollup processor or set an incorrect gasUsage.

Alternatively the `DefiBridgeProxy` could trigger the subsidy with the exact gas amount after a bridge call by using `msg.gas`

6. No withdraw or update/topUp for a active subsidy criteria

It is not possible to withdraw or increase the `ETH` amount for an active criteria. Only if the available amount equals zero.

This makes it not possible for multiple subsidizers to compensate for the same criteria.

If the only reason for that is the `gasPerMinute` rate defined by the first subsidy caller, potentially letting the bridge set the `gasPerMinute` rate instead of the minimum `gasPerMinute` rate could solve this problem.

7. Notes

- Adding a public view function to get the current subsidy for a given block.timestamp for a specific bridge criteria (for integrations)

- The contract and the struct have the same name in Solidity

- Consider a ratePerSec instead of ratePerMinute to avoid conversion or convert only once in `setGasUsageAndMinGasPerMinute`

Audit Report for Aztec - August 8, 2022

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