Assure DeFi®

THE VERIFICATION GOLD STANDARD

Security Assessment

DecenterAI

Date: 14/05/2025

Audit Status: PASS

Audit Edition: Standard

ASSURE DEFI®

THE VERIFICATION GOLD STANDARD

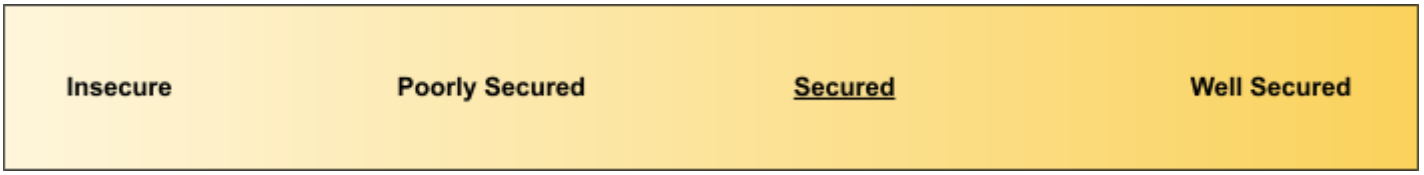
Risk Analysis

Vulnerability summary

Classification	Description
High	High-level vulnerabilities can result in the loss of assets or manipulation of data.
Medium	Medium-level vulnerabilities can be challenging to exploit, but they still have a considerable impact on smart contract execution, such as allowing public access to critical functions.
Low	Low-level vulnerabilities are primarily associated with outdated or unused code snippets that generally do not significantly impact execution, sometimes they can be ignored.
Informational	Informational vulnerabilities, code style violations, and informational statements do not affect smart contract execution and can typically be disregarded.

Executive Summary

According to the Assure assessment, the Customer's smart contract is **Secured.**



Scope

Target Code And Revision

For this audit, we performed research, investigation, and review of the DecenterAI contracts followed by issue reporting, along with mitigation and remediation instructions outlined in this report.

Target Code And Revision

Project	Assure
Language	Solidity
Codebase	https://etherscan.io/address/0x781dB9A4D8Ae055571e8796DD4423bc13CeE5dD6#writeContract
Audit Methodology	Static, Manual

Attacks made to the contract

In order to check for the security of the contract, we tested several attacks in order to make sure that the contract is secure and follows best practices.

Category	Item
Code review & Functional Review	• Compiler warnings. • Race conditions and Reentrancy. Cross-function race conditions. • Possible delays in data delivery. • Oracle calls. • Front running. • Timestamp dependence. • Integer Overflow and Underflow. • DoS with Revert. • DoS with block gas limit. • Methods execution permissions. • Economy model. • Private user data leaks. • Malicious Event log. • Scoping and Declarations. • Uninitialized storage pointers. • Arithmetic accuracy. • Design Logic. • Cross-function race conditions. • Safe Zeppelin module. • Fallback function security. • Overpowered functions / Owner privileges

AUDIT OVERVIEW

HIGH

1. ETH-Transfer DoS via sendETH [Fixed ✅]

Functions: sendETH(address payable recipient, uint256 amount), _transfer(...)

Issue: A failing low-level recipient.call{value:amount}("") in sendETH reverts the entire transfer transaction if the marketing wallet's fallback reverts, effectively halting all transfers once any swap occurs.

Recommendation: Use a gas-limited call that does not revert on failure and emit an event for failure.

Update: Ownership is renounced and marketing wallet can't revert because never would be a contract.

2. Large Automatic Swaps & Market Impact [Fixed ✅]

Functions: _transfer(address from, address to, uint256 amount)

Issue: Upon reaching the swap threshold, the contract swaps its entire balance, leading to significant slippage and making it a target for front-running (sandwich) attacks.

Recommendation: Cap each swap to a configurable multiple of swapTokensAtAmount (e.g., min(contractBalance, swapTokensAtAmount * 5)) and/or introduce randomized or time-based caps to break large swaps into smaller, less predictable chunks.

Update: per design, swap logic lives in _transfer for simplicity and no privileged setter exists; acceptable given protocol goals.

MEDIUM

1. Misnamed & Bypassable Max-Wallet Limit [Acknowledge ✅]

Functions: _transfer(address from, address to, uint256 amount)

Issue: Variable maxWalletPercent = 20 is applied as totalSupply * 20 / 1000 (2%), not 20%.

The limit only applies on buys from uniswapV2Pair and can be bypassed via non-pair transfers.

Recommendation: Rename to maxWalletPermille or adjust divisor to /100 for percent semantics.

Enforce the wallet cap on all inbound transfers

LOW

1. Centralized Fee Control [Fixed ✅]

Functions: updateFees(uint256 _buyFee, uint256 _sellFee)

Issue: The owner can unilaterally change buy/sell fees (up to 25%) at any time, potentially trapping or penalizing holders without warning.

Recommendation: Implement a governance timelock (e.g., 48 hours) before fee changes take effect and emit an event FeesUpdated(uint256 oldBuyFee, uint256 newBuyFee, uint256 oldSellFee, uint256 newSellFee).

Update: Ownership renounced.

INFORMATIONAL

1. Missing Transparency & Gas Optimizations [Acknowledge ✅]

Functions: Various (_transfer, updateFees, swap logic)

Issue: Lack of events for key on-chain actions (SwapExecuted, ETHSendFailed, MaxWalletUpdated).

Minor gas inefficiencies (repeated state reads, use of full uint256 for small config values).

Recommendation: Emit diagnostic events for swaps, fee changes, and wallet-limit updates.

Cache state variables in memory within hot functions and consider using smaller integer types (uint8/uint16) for config parameters.

Technical Findings Summary

Findings

Vulnerability Level	Total	Pending	Not Apply	Acknowledged	Partially Fixed	Fixed
High	2					2
Medium	1			1		
Low	1					1
Informational	1			1		

Assessment Results

Score Results

Review	Score
Global Score	**85/100**
Assure KYC	Not completed
Audit Score	85/100

The Following Score System Has been Added to this page to help understand the value of the audit, the maximum score is 100, however to attain that value the project must pass and provide all the data needed for the assessment. Our Passing Score has been changed to 84 Points for a higher standard, if a project does not attain 85% is an automatic failure. Read our notes and final assessment below. The Global Score is a combination of the evaluations obtained between having or not having KYC and the type of contract audited together with its manual audit.

Audit PASS

Following our comprehensive security audit of the token contract for the DecenterAI project, the project did meet the necessary criteria required to pass the security audit

Disclaimer

Assure Defi has conducted an independent security assessment to verify the integrity of and highlight any vulnerabilities or errors, intentional or unintentional, that may be present in the reviewed code for the scope of this assessment. This report does not constitute agreement, acceptance, or advocating for the Project, and users relying on this report should not consider this as having any merit for financial DecenterAI in any shape, form, or nature. The contracts audited do not account for any economic developments that the Project in question may pursue, and the veracity of the findings thus presented in this report relate solely to the proficiency, competence, aptitude, and discretion of our independent auditors, who make no guarantees nor assurance that the contracts are entirely free of exploits, bugs, vulnerabilities or deprecation of technologies.

All information provided in this report does not constitute financial or investment DecenterAI, nor should it be used to signal that any person reading this report should invest their funds without sufficient individual due diligence, regardless of the findings presented. Information is provided ‘as is, and Assure Defi is under no covenant to audit completeness, accuracy, or solidity of the contracts. In no event will Assure Defi or its partners, employees, agents, or parties related to the provision of this audit report be liable to any parties for, or lack thereof, decisions or actions with regards to the information provided in this audit report.

The assessment DecenterAIs provided by Assure Defi are subject to dependencies and are under continuing development. You agree that your access or use, including but not limited to any DecenterAIs, reports, and materials, will be at your sole risk on an as-is, where-is, and as-available basis. Cryptographic tokens are emergent technologies with high levels of technical risk and uncertainty. The assessment reports could include false positives, negatives, and unpredictable results. The DecenterAIs may access, and depend upon, multiple layers of third parties.