TENSET

Smart Contract Audit

FlipIt

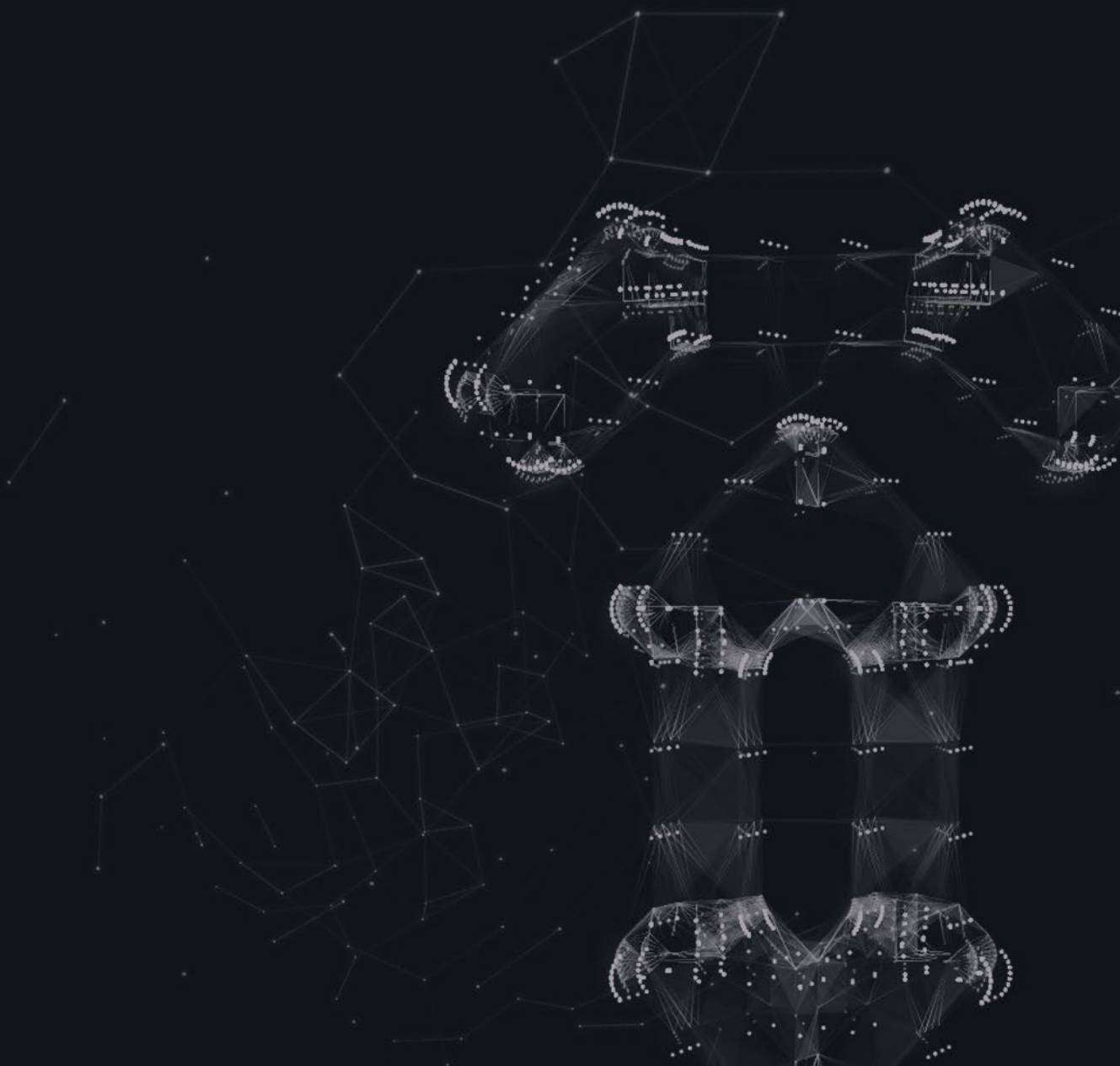

Table of Contents

Summary

SMART CONTRACT
0x1d745b9c04aa2946a4e16862cfe4dd2469a28fa1

Project Summary

LANGUAGE	CODEBASE	COMMITS
Solidity	Undisclosed	cbe93434417fcad06fc70b2a589c8ceed9384b84

Audit Scope

ID	File
FI	contracts/FlipIt.sol

Findings

Audit Overview

- 0 Critical
- 0 High
- 0 Medium
- 3 Low
- 0 Informational

Issues

Severity	Found	Resolved	Partially Fixed	Acknowledged
Low	3	3	0	0
Total	**3**	**3**	**0**	**0**

FI-01.
Floating Pragma

Low impact Resolved

Description

Contracts should be deployed with the same compiler version and flags that they have been tested with thoroughly. Locking the pragma helps to ensure that contracts do not accidentally get deployed using, for example, an outdated compiler version that might introduce bugs that affect the contract system negatively.

Recommendation

Lock the pragma version.

Comments

The team resolved this issue.

FI-02.
Missing event

Low impact Resolved

Description

Function **updateSupervisedTransfersEndAt** does not emit an event, so it is difficult to track changes in the value of **supervisedTransfersEndAt** off-chain.

Recommendation

Event should be emitted indicating **supervisedTransfersEndAt** change.

Comments

The team resolved this issue.

FI-03.
Frequently used storage variable

Low impact Resolved

Description

Function **updateSupervisedTransfersEndAt** reads the variable **supervisedTransfersEndsAt** from storage twice.

Recommendation

Catch frequently used storage variables in memory/stack, converting multiple **SLOAD** into **1 SLOAD**.

Comments

The team resolved this issue.

Guidelines

FI-01.
Public variable could be declared as internal

Gas optimalization Applied

Recommendation

All state variables that will not be used for interaction by external clients should be internal to save gas.

Comments

The team applied this guideline.

FI-02.
Missing comments

Readability Applied

Recommendation

The purpose of all state variables, events, errors, functions, arguments and other important information should be documented with NatSpec.

Comments

The team applied this guideline.

Disclaimer

The smart contracts given for audit have been analyzed by the best industry practices at the date of this report, with cybersecurity vulnerabilities and issues in smart contract source code, the details of which are disclosed in this report (Source Code); the Source Code compilation, deployment, and functionality (performing the intended functions).

The report contains no statements or warranties on the identification of all vulnerabilities and security of the code. The report covers the code submitted to and reviewed, so it may not be relevant after any modifications. Do not consider this report as a final and sufficient assessment regarding the utility and safety of the code, bug-free status, or any other contract statements.

While we have done our best in conducting the analysis and producing this report, it is important to note that you should not rely on this report only — we recommend proceeding with several independent audits and a public bug bounty program to ensure the security of smart contracts. English is the original language of the report. The Consultant is not responsible for the correctness of the translated versions.

Smart contracts are deployed and executed on a blockchain platform. The platform, its programming language, and other software related to the smart contract can have vulnerabilities that can lead to hacks. Thus, Consultant cannot guarantee the explicit security of the audited smart contracts.