

Smart Contract

Security Assessment

**For Insurance.sol
11 Sept 2022**



Ascendant

Ascendant

@ascendantproj
www.ascendant.finance



Ascendant

Table of Contents

3 Disclaimer

4 Executive Summary

5 Overview

7 Findings Summary & Legend

8 Manual Review

- Issue Checking Status
- Audit Findings
- Functional Test Status

16 Automated Review

- Omitted Results
- Unified Model Language

19 Conclusion

DISCLAIMER

Ascendant Finance (“Ascendant”) has conducted an independent audit to verify the integrity of and highlight any vulnerabilities or errors, intentional or unintentional, that may be present in the codes that were provided for the scope of this audit. This audit report does not constitute agreement, acceptance or advocacy for the Project that was audited, and users relying on this audit report should not consider this as having any merit for financial advice in any shape, form or nature. The contracts audited do not account for any economic developments that may be pursued by the Project in question, and that 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 completely free of exploits, bugs, vulnerabilities or deprecation of technologies. Further, this audit report shall not be disclosed nor transmitted to any persons or parties on any objective, goal or justification without due written assent, acquiescence or approval by Ascendant.

All information provided in this report does not constitute financial or investment advice, nor should it be used to signal that any persons reading this report should invest their funds without sufficient individual due diligence regardless of the findings presented in this report. Information is provided ‘as is’, and Ascendant is under no covenant to the completeness, accuracy or solidity of the contracts audited. In no event will Ascendant 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 and/or actions with regards to the information provided in this audit report.

Cryptocurrencies and any technologies by extension directly or indirectly related to cryptocurrencies are highly volatile and speculative by nature. All reasonable due diligence and safeguards may yet be insufficient, and users should exercise considerable caution when participating in any shape or form in this nascent industry.

The audit report has made all reasonable attempts to provide clear and articulate recommendations to the Project team with respect to the rectification, amendment and/or revision of any highlighted issues, vulnerabilities or exploits within the contracts provided. It is the sole responsibility of the Project team to sufficiently test and perform checks, ensuring that the contracts are functioning as intended, specifically that the functions therein contained within said contracts have the desired intended effects, functionalities and outcomes of the Project team. Ascendant retains full rights over all intellectual property (including expertise and new attack or exploit vectors) discovered during the audit process. Ascendant is therefore allowed and expected to re-use this knowledge in subsequent audits and to inform existing projects that may have similar vulnerabilities. Ascendant may, at its discretion, claim bug bounties from third-parties while doing so.

Executive Summary

Severity	Found
● High	0
● Medium	2
● Low	5
● Informational	67
Total	74

We performed an independent technical audit to identify Smart Contracts uncertainties. This shall protect the code from illegitimate authorization attempts or external & internal threats of any type. This also ensures end-to-end proofing of the contract from frauds. The audit was performed semi-manually. We analyzed the Smart Contracts code line-by-line and used an automation tool to report any suspicious code.

The following tools were used:

- Truffle
- Remix IDE
- Slither

Overview

This report has been prepared for Insurance on the Ethereum network. Ascendant provides a user-centered examination of the smart contracts to look for vulnerabilities, logic errors or other issues from both an internal and external perspective.

Summary





Project Name	Insurance.sol
Platform	Ethereum
Language	Solidity

Contracts Assessed





Name	Location
Insurance.sol	Not Published
IERC20PermitUpgradeable.sol	In Insurance Contract
OwnableUpgradeable.sol	In Insurance Contract
PausableUpgradeable.sol	In Insurance Contract
IERC20Upgradeable.sol	In Insurance Contract
ReentrancyGuardUpgradeable.sol	In Insurance Contract

Name	Location
IERC20MetadataUpgradeable.sol	In Insurance Contract
IERC20PermitUpgradeable.sol	In Insurance Contract
AddressUpgradeable.sol	In Insurance Contract
SafeERC20Upgradeable.sol	In Insurance Contract
Initializable.sol	In Insurance Contract
ContextUpgradeable.sol	In Insurance Contract

Findings Summary

Severity	Found
 High	0
 Medium	0
 Low	7
 Informational	67
Total	74

Classification of Issues

 High	Exploits, vulnerabilities or errors that will certainly or probabilistically lead towards loss of funds, control, or impairment of the contract and its functions. Issues under this classification are recommended to be fixed with utmost urgency.
 Medium	Bugs or issues that may be subject to exploit, though their impact is somewhat limited. Issues under this classification are recommended to be fixed as soon as possible.
 Low	Effects are minimal in isolation and do not pose a significant danger to the project or its users. Issues under this classification are recommended to be fixed nonetheless.
 Informational	Consistency, syntax or style best practices, Generally pose a negligible level of risk, if any.

Manual Review



Ascendant

Issues Checking Status

Issue Description	Checking Status
Compiler errors	PASS
Race conditions and Reentrancy. Cross-function race conditions.	PASS
Possible delays in data delivery.	PASS
Oracle calls.	PASS
Front running.	PASS
Timestamp dependence.	PASS
Integer Overflow and Underflow.	PASS
DoS with Revert.	PASS
DoS with block gas limit.	PASS
Methods execution permissions.	PASS
Economy model of the contract.	PASS
The impact of the exchange rate on the logic.	PASS
Private user data leaks.	PASS
Malicious Event log.	PASS
Scoping and Declarations.	PASS
Uninitialized storage pointers.	PASS

Arithmetic accuracy.	PASS
Design Logic.	PASS
Cross-function race conditions.	PASS
Safe Open Zeppelin contracts implementation and usage.	PASS
Fallback function security.	PASS

Audit Findings

Severity	Low
Contract	Insurance.sol
Description	Checks-Effects-Interactions
Code Snippet	<pre>IERC20MetadataUpgradeable(token). safeTransferFrom(msg.sender, treasury, tokenAmount); // If renewing insurance, old amount needs to be added. projects[pid].availableAmount = project.availableAmount + userInsurance.amountInsured - userAmount; userInsurance.validTill = validTill; userInsurance.amountInsured = userAmount; userInsurance.amountClaimable = userAmount;</pre>
Recommendation	<p>safeTransferFrom is called before updating state changes, which breaks the Checks-Effects-Interactions pattern and is a reentrancy risk. It is noted that the function includes a Reentrancy Guard to mitigate this, and so the issue threat has been downgraded accordingly. No further action is necessary.</p>
Status	

Audit Findings

Severity	Low
Contract	Insurance.sol
Description	Checks-Effects-Interactions
Code Snippet	<pre>IERC20MetadataUpgradeable(tokenAddress).safeTransfer(msg.sender, claimTokenAmount); userInsurances[pid] [msg.sender].claimed = userInsurance.amountClaimable;</pre>
Recommendation	<p>safeTransferFrom is called before updating state changes, which breaks the Checks-Effects-Interactions pattern and is a reentrancy risk. It is noted that the function includes a Reentrancy Guard to mitigate this, and so the issue threat has been downgraded accordingly. No further action is necessary.</p>
Status	

Severity	Informational
Contract	Insurance.sol
Description	Mismatched pragma versions
Code Snippet	<p>Different versions of Solidity are used:</p> <ul style="list-style-type: none"> - Version used: ['^0.8.0', '^0.8.1', '^0.8.2', '^0.8.13'] - ^0.8.0 (insurance.sol#5) - ^0.8.0 (insurance.sol#67) - ^0.8.0 (insurance.sol#158) - ^0.8.1 (insurance.sol#186) - ^0.8.0 (insurance.sol#412) - ^0.8.2 (insurance.sol#563) - ^0.8.0 (insurance.sol#706) - ^0.8.0 (insurance.sol#782) - ^0.8.0 (insurance.sol#819) - ^0.8.0 (insurance.sol#935) - ^0.8.13 (insurance.sol#1033)
Recommendation	Use one Solidity version.
Status	

Functional Test Status

Function Name	Type/Return Type	Score
owner	read/public	PASS
renounceOwnership	write/public	PASS
transferOwnership	write/public	PASS
msgSender	internal	PASS
_msgData	internal	PASS
name	read/public	PASS
symbol	read/public	PASS
decimals	read/public	PASS
totalSupply	read/public	PASS
balanceOf	read/public	PASS
transfer	write/public	PASS
allowance	read/public	PASS
approve	write/public	PASS
transferFrom	write/public	PASS
paused	read/public	PASS
addProject	write/external	PASS
adminWithdraw	write/external	PASS
changeProjectAllowClaim	write/external	PASS

changeProjectIsActive	write/external	PASS
changeProjectLockAmount	write/external	PASS
changeProjectMaxDays	write/external	PASS
changeProjectMaxPerUser	write/external	PASS
changeProjectName	write/external	PASS
changeProjectPremium	write/external	PASS
claimInsurance	write/external	PASS
getProjects	read/external	PASS
pause	write/external	PASS
initialize	write/external	PASS
setAllowedTokens	write/external	PASS
setAuthorizedContract	write/external	PASS
setTreasury	write/external	PASS
setUserInsurance	write/external	PASS
setUserInsuranceClaimable	write/external	PASS
takeInsuranceCover	write/external	PASS
unpause	write/external	PASS

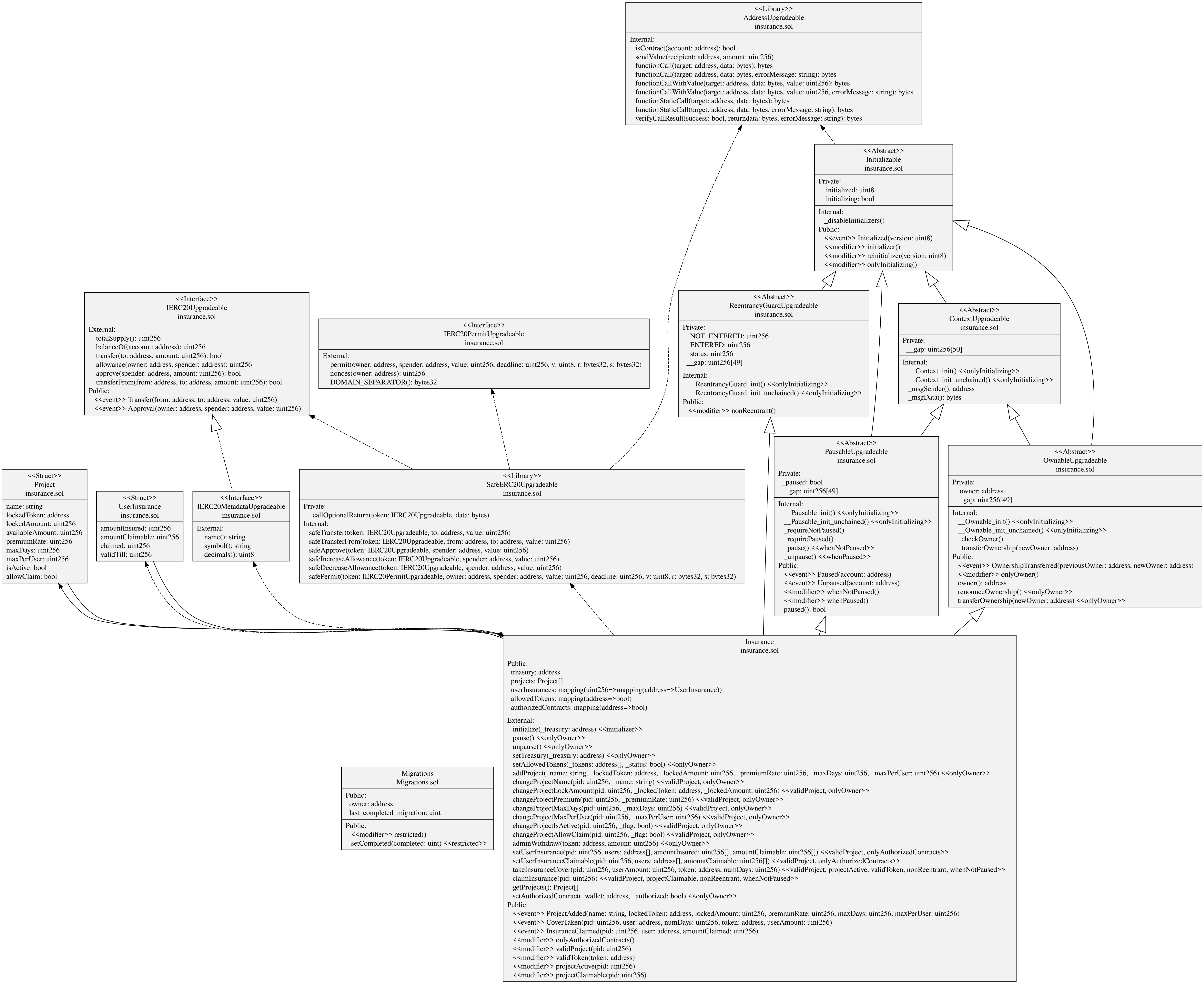
Automated Review



Ascendant

Omitted Results

Note: Any issues that have been omitted from this report have been deemed by the reviewing team as irrelevant, inapplicable, and/or negligible to the proper functioning of this contract. Thus, any omitted issues can be safely ignored.



Conclusion

The smart contracts reviewed in this audit contain no critical severity issues and all Medium to Low issues have either been corrected or acknowledged.

Please check the disclaimer above and note, the audit makes no statements or warranties on business model, investment attractiveness or code sustainability. The report is provided for the only contract mentioned in the report and does not include any other potential contracts deployed by Owner.



Ascendant

Ascendant

@ascendantproj

www.ascendant.finance



Ascendant