

Smart Contract

Security Assessment

**For Ares NFT
09 Sept 2022**



Ascendant

Ascendant

@ascendantproj
www.ascendant.finance



Ascendant

Table of Contents

3 Disclaimer

4 Executive Summary

5 Overview

6 Findings Summary & Legend

9 Manual Review

- Request For Comment
- Issue Checking Status
- Audit Findings
- Functional Test Status

21 Automated Review

- Unified Model Language

23 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	3
● Low	10
● Informational	62
Total	75

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 Ares NFT 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	ARES NFT
Platform	Ethereum
Language	Solidity

Contracts Assessed





Name	Location
AresNFT.sol	Not Published
ReentrancyGuard.sol	In AresNFT contract
Ownable.sol	In AresNFT contract
Context.sol	In AresNFT contract
Address.sol	In AresNFT contract
IERC721Receiver.sol	In AresNFT contract

Name	Location
IERC165.sol	In AresNFT contract
IERC2981.sol	In AresNFT contract
ERC165.sol	In AresNFT contract
IERC721.sol	In AresNFT contract
IERC721Metadata.sol	In AresNFT contract
Strings.sol	In AresNFT contract
ERC2981.sol	In AresNFT contract
Merkleproof.sol	In AresNFT contract
ERC721A.sol	In AresNFT contract
Payable.sol	In AresNFT contract

Findings Summary

Severity	Found
 High	0
 Medium	3
 Low	10
 Informational	62
Total	75

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

Request For Comment

Criteria	Comments	Status
7313 total NFTs	maxSupply/totalCollectionSize constant variable removed from ERC721A.sol	PASS
Reserved for the team and marketing: (1550/7313)	No comments	PASS
Mint price: 0.86 ETH - Whitelist //0.15 ETH - pre- sale public mint // 0.20 ETH - public mint	No comments	PASS

Review

Criteria	Comments	Status
Royalty for NFT owners: 10% (Secondary markets)	No comments	PASS
Contract name: "Ares-NFT"	No comments	PASS
Mint price: 0.86 ETH - Whitelist //0.15 ETH - pre- sale public mint // 0.20 ETH - public mint	No comments	PASS
Royalty for NFT owners: 10% (Secondary markets)	No comments	PASS
Contract name: "Ares-NFT"	No comments	PASS
Minting should save it in the user wallet and automatically update to Opensea (placeholder image until collection is revealed)	No comments	PASS
The ETH sales from minting to one wallet Address	No comments	PASS

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	Medium
Contract	AresNFT.sol
Description	Strict equivalency
Code Snippet	<pre>modifier correctValue(uint256 expectedValue) { require(expectedValue == msg.value, "Ether value sent is not correct"); _; }</pre>
Recommendation	<p>Mint functions require the value to strictly equal the price * mintAmount. Strict equivalency can sometimes lead to reverted transactions due to rounding issues.</p> <p>As a general rule, <code>msg.value >= price</code>, and you can include what the function should do if the value is greater.</p>
Status	ACKNOWLEDGED

Audit Findings

Severity	Medium x3
Contract	AresNFT.sol
Description	Checks-Effects-Interactions
Code Snippet	<pre>All mint functions: { require(mintsPerAddress[currentPhase][msg.sender] + quantity < publicAllowancePlusOne, "Exceeds allowance!"); _safeMint(msg.sender, quantity); mintsPerAddress[currentPhase][msg.sender] += quantity; }</pre>
Recommendation	<p>The mint functions currently all fail the checks-effects-interactions pattern, which requires that interactions (such as <code>_safeMint</code>) should come after state changes (updating the mapping), which exposes the function to Reentrancy attacks.</p> <p><code>mintPerAddress[currentPhase][msg.sender] += quantity</code> should either precede <code>_safeMint</code> or a <code>ReentrancyGuard</code> should be added to these functions.</p>
Status	ACKNOWLEDGED

Audit Findings

Severity	Low
Contract	AresNFT.sol
Description	Public baseURI
Code Snippet	<pre>string public baseURI;</pre>
Recommendation	<p>baseURI is public, which means a bad actor could potentially look up URI and steal NFTs pre-mint. It appears that the tokenURI function is meant to only show the placeholder URI until the baseURI is set.</p> <p>baseURI must never be set until the team is ready to reveal.</p>
Status	ACKNOWLEDGED

Audit Findings

Severity	Low
Contract	AresNFT.sol
Description	No setPrice functions
Code Snippet	None
Recommendation	<p>There are no setPrice functions, which means that after deployment, there will be no way to change prices if necessary. Furthermore, the nextPhase function hardcodes the price values. If plans change, the contract will need to be redeployed.</p> <p>Never hardcode values or leave out set functions that may be needed in the future.</p>
Status	ACKNOWLEDGED

Functional Test Status

Function Name	Type/Return Type	Score
_addPayee	private	PASS
_pendingPayment	private	PASS
payee	read/public	PASS
release	write/public	PASS
released	read/public	PASS
setRoyalties	write/external	PASS
totalReleased	read/public	PASS
totalShares	read/public	PASS
updatePayee	write/public	PASS
_setRoyalties	internal	PASS
owner	read/public	PASS
renounceOwnership	write/public	PASS
transferOwnership	write/public	PASS
transferFrom	write/public	PASS
increaseAllowance	write/public	PASS
decreaseAllowance	write/public	PASS
_transfer	internal	PASS
_mint	internal	PASS

Function Name	Type/Return Type	Score
_msgData	internal	PASS
_msgSender	internal	PASS
_baseURI	internal	PASS
_checkContractOnERC721Received	private	PASS
_exists	internal	PASS
_getAux	internal	PASS
_numberMinted	internal	PASS
_numberBurned	internal	PASS
_ownershipOf	internal	PASS
_safeMint	internal	PASS
_setAux	internal	PASS
_startTokenId	internal	PASS
_totalMinted	internal	PASS
ownerOf	read/public	PASS
balanceOf	read/public	PASS
getApproved	write/public	PASS
isApprovedForAll	read/public	PASS
safeTransferFrom	write/public	PASS

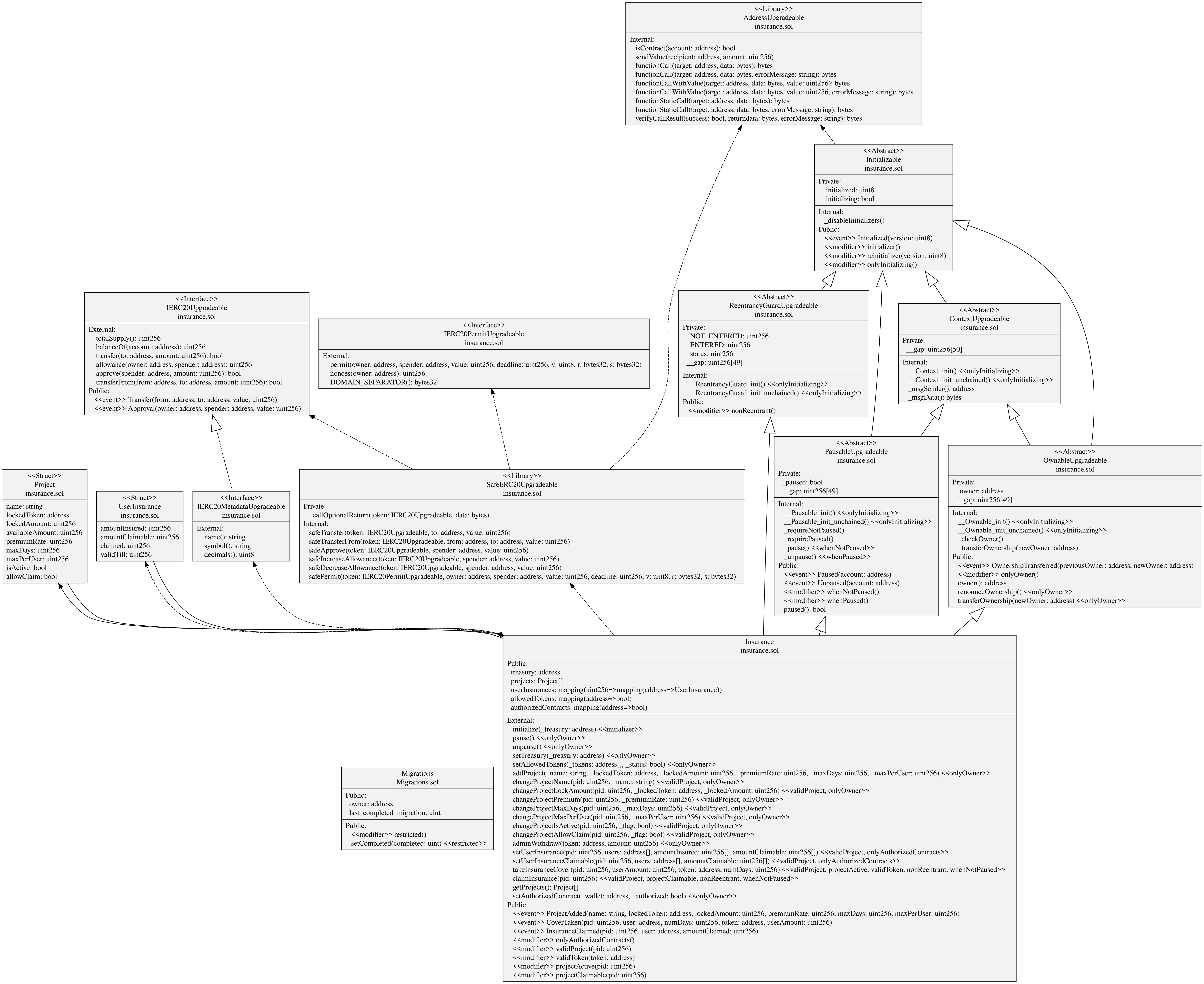
Function Name	Type/Return Type	Score
name	read/public	PASS
_withdraw	internal	PASS
accountBalance	read/public	PASS
getMintsPerAddress	read/external	PASS
mintPresale	write/external	PASS
mintPublic	write/external	PASS
mintReserved	write/external	PASS
mintWhitelist	write/external	PASS
nextPhase	write/external	PASS
ownedSince	read/public	PASS
setContractState	write/external	PASS
setBaseURI	write/external	PASS
setMaxPresale	write/external	PASS
setMaxSale	write/external	PASS
setMaxWhitelist	write/external	PASS
setMerkleRoot	write/external	PASS
setPlaceholderURI	write/external	PASS
setPresaleAllowance	write/external	PASS

_burn	internal	PASS
_approve	internal	PASS
_spendAllowance	internal	PASS
_beforeTokenTransfers	internal	PASS
_afterTokenTransfer	internal	PASS
setPublicAllowance	write/external	PASS
setRecipient	write/external	PASS
supportsInterface	read/external	PASS
tokenURI	read/public	PASS
verify	read/public	PASS
withdraw	write/public	PASS

Automated Review



Ascendant



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