Blaize.Security

February 13th, 2023 / V. 1.0

VIEWPOINT LABS

TITLE DEED CEX

SMART CONTRACT AUDIT

TABLE OF CONTENTS

AUDIT RATING

Viewpoint Labs contract's source code was taken from the repository provided by the Viewpoint Labs team.

SCORE **9.7** /10

The scope of the project is Title Deeds CEX by **Viewpoint Labs** team

Repository:
https://github.com/viewpoint-labs/smart-contracts/tree/td-cex

td-cex branch

Intial commit (audited):

- 9149779f81fbf2f488e7b13509040711705ca067

Final commit (post-audit):

- e768133bd776626acd4b4d9dcb774a5efbf93c16

The scope of the project is **Viewpoint Labs** set of contracts during 2nd audit iteration:

landsale\contracts
TitleDeedRedeemer.sol
TitleDeedExchanger.sol

game\contracts
TokenBase.sol
BlueprintNFT.sol
ParcelNFT.sol

common\contracts\tokens\extensions
ERC721BatchRetriever.sol
ERC2981Royalty.sol
ERC4906.sol
MintableToken.sol
TokenMetadata.sol
TokenRoles.sol

common\contracts\tokens
ERC721Default.sol
ERC1155Default.sol

common\contracts\utils
CountersPlus.sol

TECHNICAL SUMMARY

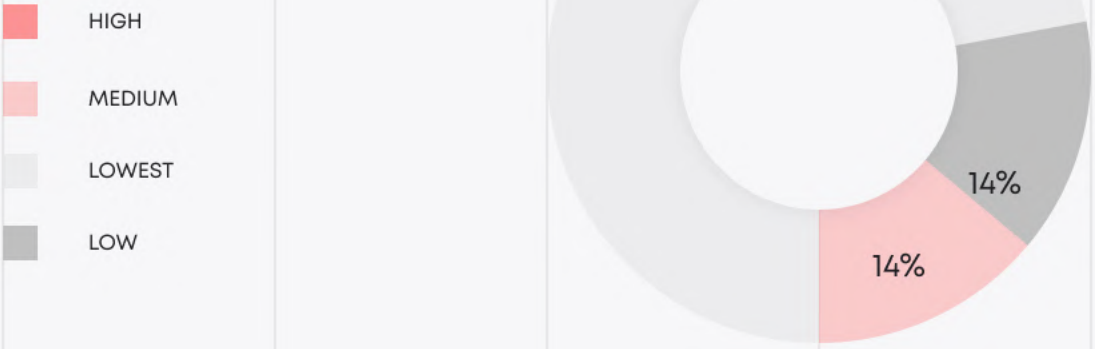
In this report, we consider the security of the contracts for Title Deeds CEX protocol. Our task is to find and describe security issues in the smart contracts of the platform. This report presents the findings of the security audit of **Title Deeds CEX** smart contracts conducted between **February 6th, 2023 - February 13th, 2023.**

Testable code

The testable code is 99%, which above the industry standard of 95%.

The scope of the audit includes the unit test coverage, that bases on the smart contracts code, documentation and requirements presented by the Viewpoint Labs team. Coverage is calculated based on the set of Hardhat framework tests and scripts from additional testing strategies. Though, in order to ensure a security of the contract Blaize.Security team recommends the Viewpoint Labs team put in place a bug bounty program to encourage further and active analysis of the smart contracts.

THE GRAPH OF VULNERABILITIES DISTRIBUTION:



The table below shows the number of found issues and their severity. A total of 7 problems were found. 7 issues were fixed or verified by the Viewpoint Labs team.

	FOUND	FIXED/VERIFIED
Critical	0	0
High	0	0
Medium	1	1
Low	1	1
Lowest	5	5

SEVERITY DEFINITION

Critical

A system contains several issues ranked as very serious and dangerous for users and the secure work of the system. Needs immediate improvements and further checking.

High

A system contains a couple of serious issues, which lead to unreliable work of the system and migh cause a huge information or financial leak. Needs immediate improvements and further checking.

Medium

A system contains issues which may lead to mediumfinancial loss or users' private information leak. Needs immediate improvements and further checking.

Low

A system contains several risks ranked as relatively small with the low impact on the users' information and financial security. Needs improvements.

Lowest

A system does not contain any issue critical to the secure work of the system, yet is relevant for best

AUDITING STRATEGY AND TECHNIQUES APPLIED \ PROCEDURE

We have scanned this smart contract for commonly known and more specific vulnerabilities:

- Unsafe type inference;
- Timestamp Dependence;
- Reentrancy;
- Implicit visibility level;
- Gas Limit and Loops;
- Transaction-Ordering Dependence;
- Unchecked external call - Unchecked math;
- DoS with Block Gas Limit;
- DoS with (unexpected) Throw;
- Byte array vulnerabilities;
- Malicious libraries;
- Style guide violation;
- ERC20 API violation;
- Uninitialized state/storage/ local variables;
- Compile version not fixed.

Procedure

In our report we checked the contract with the following parameters:

- Whether the contract is secure;
- Whether the contract corresponds to the documentation;
- Whether the contract meets best practices in efficient use of gas, code readability;

Automated analysis:

Scanning contract by several public available automated analysis tools such as Mythril, Solhint, Slither and Smartdec. Manual verification of all the issues found with tools.

Manual audit:

Manual analysis of smart contracts for security vulnerabilities. Checking smart contract logic and comparing it with the one described in the documentation.

EXECUTIVE SUMMARY

Blaize Security reviewed the whole set of contracts within the scope provided by the Viewpoint Labs team. The protocol allows users to redeem their Title Deeds NFTs in the Ethereum network and receive Parcel and Blueprint NFTs in the BNB Chain network. The protocol overview section contains a detailed description.

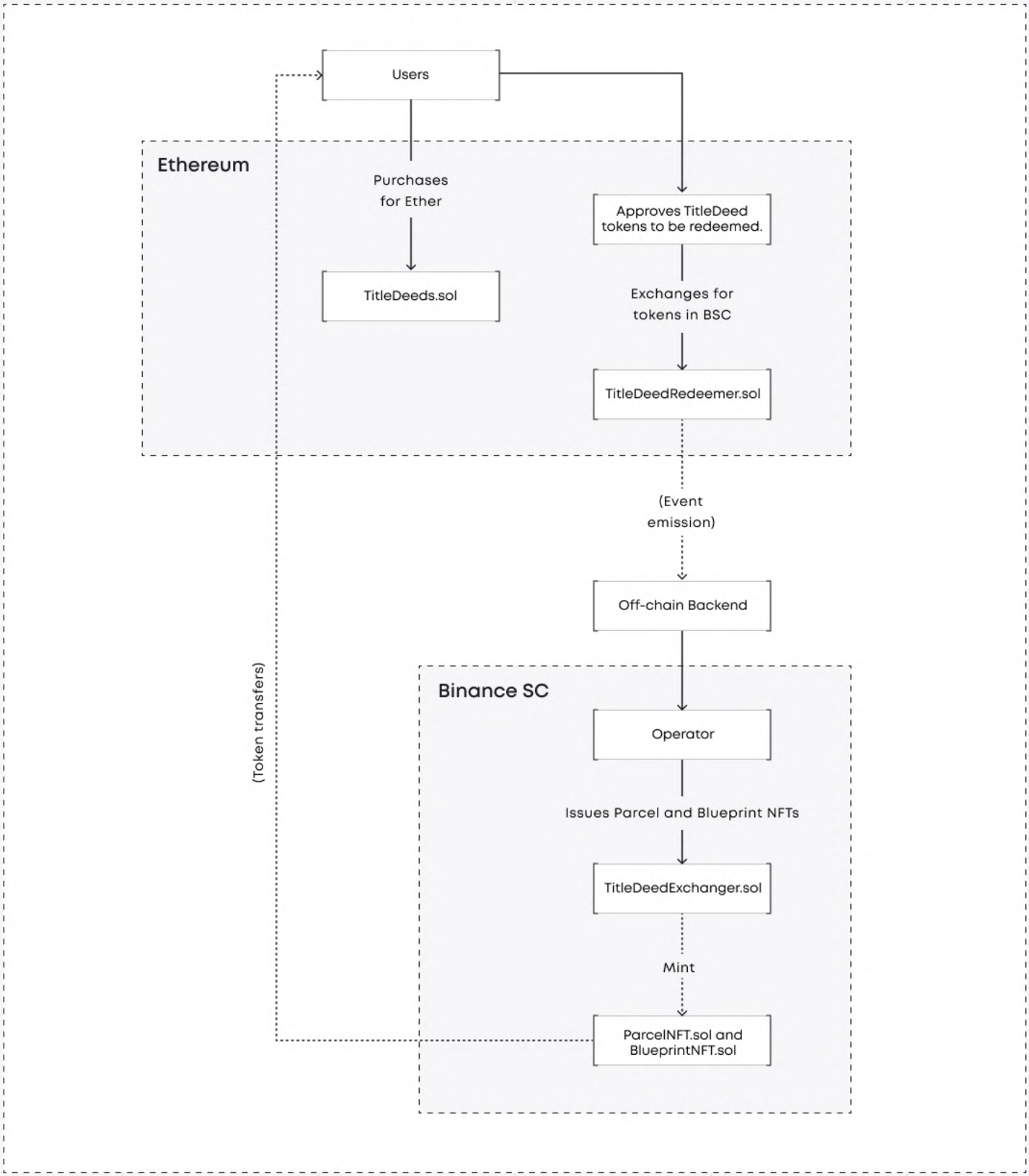
During the manual audit, the Blaize Security team analyzed contracts against the list of common vulnerabilities and internal checklists, checked the correspondence to the Solidity best practices (including code style and gas optimization), and validated the correspondence of the business logic of the protocol to the described one. The team found one medium-risk, one low-risk, and a few lowest-severity issues during the audit, and the Viewpoint Labs team successfully fixed all of them. The protocol also contains custom ERC721 and ERC1155, which extend a basic NFT functionality with role management, minting, royalty, metadata update notifications, and batchable retrieving of info about NFTs. The Blaize Security team also reviewed all of these implementations.

The overall security of smart contracts is high enough. Contracts are well-written and tested: Viewpoint Labs team prepared a solid unit test coverage. Nevertheless, the Blaize Security team prepared its own tests, including additional scenarios to validate the exchange process.

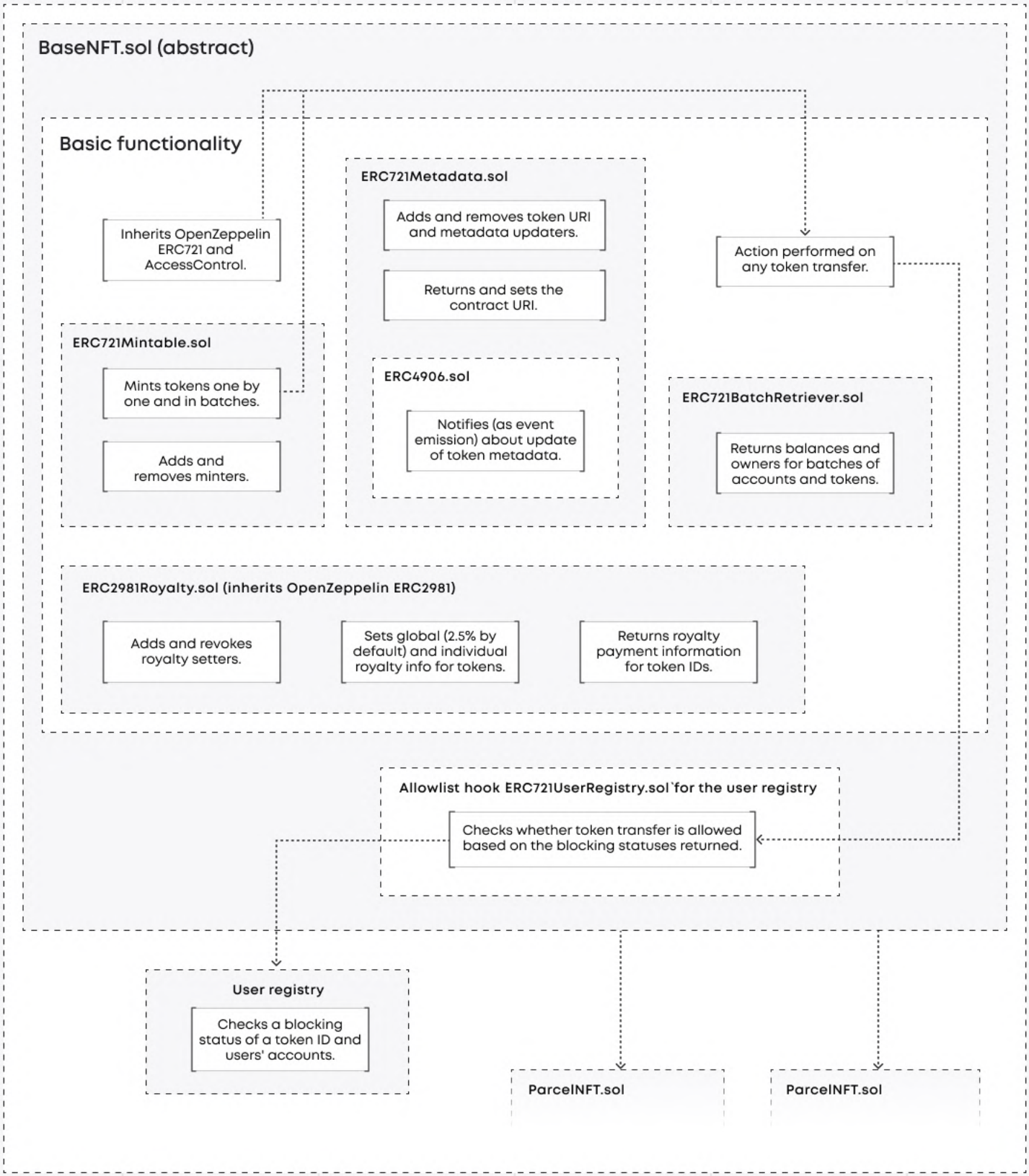
	RATING
Security	9.9
Gas usage and logic optimization	9.5
Code quality	9.6
Test coverage**	10
Total	9.7

**Contract has a native coverage, prepared by Viewpoint Labs team, though, Blaize Security has prepared their own set of unit tests and additional scenarios to cover the whole code.

PROTOCOL FLOW



BINANCE SMART CHAIN (BNB SC) MAINNET



BINANCE SMART CHAIN (BNB SC) MAINNET

ParcelNFT.sol

The ERC721 NF token represents a land parcel of San-Crypto virtual city for the Bluelight game.

This non-upgradeable contract has the ERC721 standard functionality, as well as token metadata with ERC4906 notifications, ERC2981 royalty info and admin minting.

This token (SCLP) can be exchanged for the corresponding purchased TitleDeed token on the Ethereum network via the back end.

Initialization

Name:
"San-Crypto Land Parcel",
Symbol: **"SCLP"**,
Royalty receiver address,
Contract URI,
Base token URI,
Owner address,
Minter address,
URI updater address,
Metadata updater address.

ParcelNFT.sol

The ERC1155 NF Blueprint token of San-Crypto virtual city for the Bluelight game.

This non-upgradeable contract has the ERC1155 standard functionality, as well as user burning, token metadata with ERC4906 notifications, ERC2981 royalty info and admin minting.

This token can be received as a bonus when a TitleDeed token is redeemed (exchanged for an SCLP token) with a probability determined on the back end side, e.g. 1%.

Initialization

Name:
"San-Crypto Blueprint",
Symbol: **"SCB"**,
Royalty receiver address,
Contract URI,
Base token URI,
Owner address,
Minter address,
URI updater address,
Metadata updater address.

Inherits the OpenZeppelin ERC1155Burnable extension that implements token burning for users (their owners).

mints

TitleDeedExchanger.sol

This contract is responsible for minting property and bonus NFTs (e.g. Parcel and Blueprint NFTs) by the back end, after burning the respective TitleDeed tokens on the Ethereum network.

Bonus tokens are minted at a chance determined by the back end.

That is, it is the process of exchanging TitleDeed tokens from the Ethereum chain for Parcel and Blueprint tokens to the Binance chain via the back end.

Initialization

Property ERC721Mintable` NFT address (e.g. ParcelNFT),

Bonus ERC1155Mintable` NFT address (e.g. BlueprintNFT),

Operator address.

Sets the operator.

mints

Issues one or a batch of property and bonus NFTs (for users that redeems).

transfers to

Users

addresses match

The back end uses the event to exchange tokens.

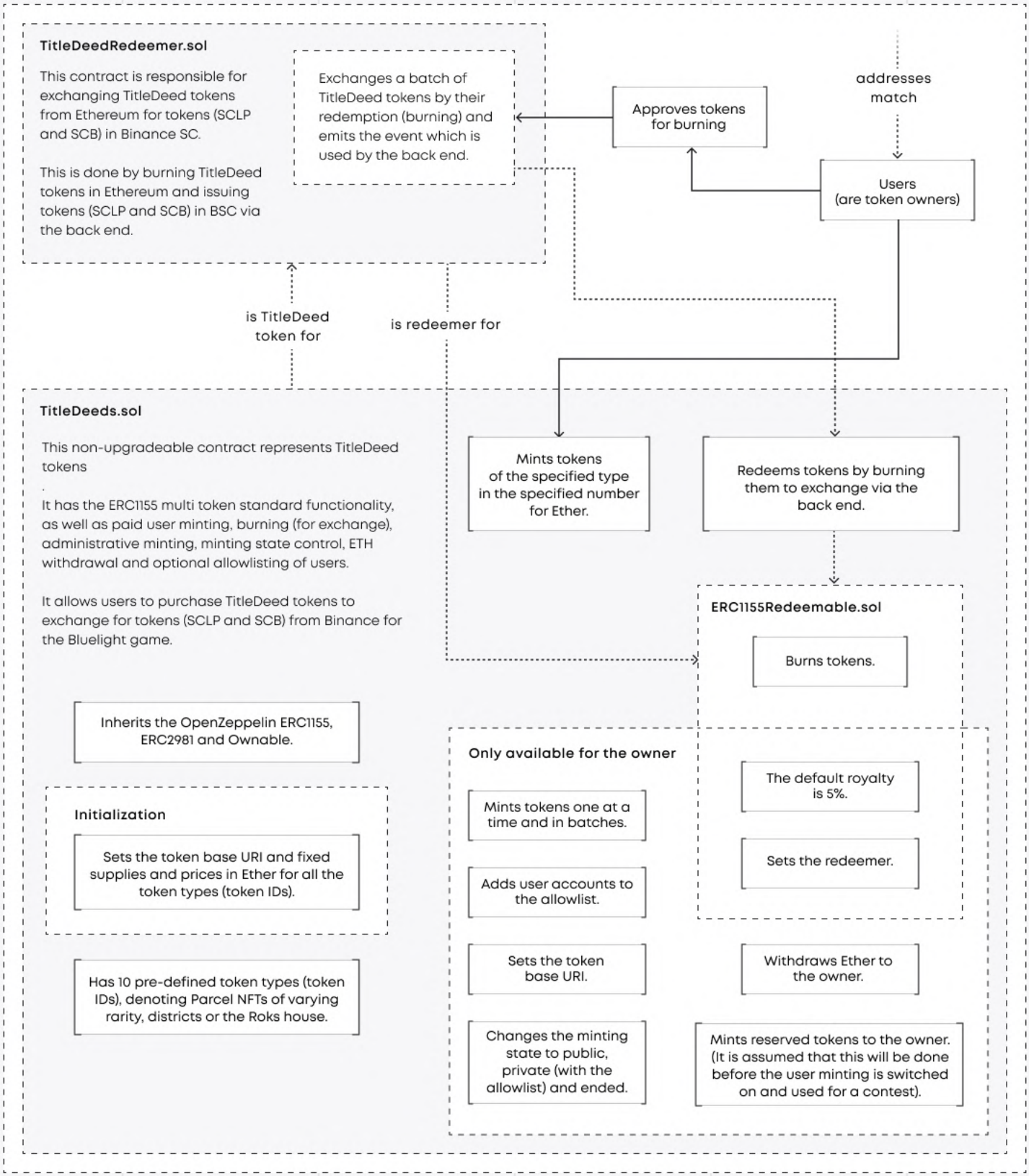
Off-chain

Back end

Operator (or owner)

Mints after redemption of TitleDeed tokens in Ethereum.

ETHEREUM MAINNET



COMPLETE ANALYSIS

MEDIUM-1	✔ Resolved

Missing validation of zero amounts when exchanging tokens.

TitleDeedRedeemer.sol: exchangeBatch().
In the case of the function call with zero amounts in `_amounts`, the tokens will not be burnt. However, `redeemIndexer` will be incremented for nothing, and the method will emit the false event. Since it is unknown how the dApp uses emitted events, it is difficult to properly define the issue's severity. However, since the function is public and affects the storage variables, auditors marked it as medium.

Recommendation:

Add a check for zero amounts.

Post-audit:

Validation was added

LOW-1	✔ Resolved

Function parameters are not validated.

TitleDeedExchanger.sol: setOperator().
TitleDeeds.sol: setAllowlisted().
Address parameters of setters are missing zero-address validation.

Recommendation:

Validate function parameters.

Post-audit:

Validation was added

LOWEST-1	✔ Resolved

Unnecessary gas spendings.

- TitleDeedRedeemer.sol: exchangeBatch().

It is a batchable function, where the storage variable `redeemIndexer`is incremented by one at a time instead of a single-time increment.

Recommendation:

Combine function `exchangeBatch()`with `_redeem().`

- TitleDeedRedeemer.sol: constructor().
  Check for the zero address after setting the state variable.

Recommendation:

Use a pre-write check.

LOWEST-2	✔ Resolved

Lack of events.

There are no events in the functions that modify the storage. It is recommended to emit events on every change of storage variable to keep track of historical changes. The following functions should emit such an event:

- TitleDeedExchanger.sol: issueWithBonus(), issueBatchWithBonus(). [About a nonce].

Recommendation:

Add events to conveniently track state changes.

LOWEST-3	✔ Resolved

Usage of deprecated method.

ERC721Roles.sol: _setupRoles(), addOwner().

ERC721Roles implements granting of roles (primary roles' setup or owner adding) with execution via the _setupRole() method, which (according to OpenZeppelin) was deprecated, in favor of _grantRole().

Though _setupRole() method calls itself a _grantRole() and nothing else. The general recommendation is to use a method suggested by OpenZeppelin in case they ever decide to remove the deprecated function.

Recommendation:

Use _grantRole() instead of _setupRole().

LOWEST-4	✔ Resolved

Unused imports.

ParcelNFT.sol.

ERC721Default`and ERC721UserRegistry`are imported, however they are never used in code.

Recommendation:

Remove unnecessary imports.

LOWEST-5	✔ Resolved

ERC1155Base is missing setters for blacklist

ERC1155Redeemable.sol: _redeemFrom()

ERC1155Base.sol

Contract misses setters for blacklist and blacklist status, however a similar contract, ERC721Base, has setters for these values.

Recommendation:

Add setters OR verify that they are unnecessary.

Post-audit:

Setters were added.

		landsale\contracts TitleDeedRedeemer.sol TitleDeedExchanger.sol
✔	Re-entrancy	Pass
✔	Access Management Hierarchy	Pass
✔	Arithmetic Over/Under Flows	Pass
✔	Delegatecall Unexpected Ether	Pass
✔	Default Public Visibility	Pass
✔	Hidden Malicious Code	Pass
✔	Entropy Illusion (Lack of Randomness)	Pass
✔	External Contract Referencing	Pass
✔	Short Address/ Parameter Attack	Pass
✔	Unchecked CALL Return Values	Pass
✔	Race Conditions / Front Running	Pass
✔	General Denial Of Service (DOS)	Pass
✔	Uninitialized Storage Pointers	Pass
✔	Floating Points and Precision	Pass
✔	Tx.Origin Authentication	Pass
✔	Signatures Replay	Pass
✔	Pool Asset Security (backdoors in the underlying ERC-20)	Pass

common\contracts\utils
CountersPlus.sol

✔ Re-entrancy	Pass
✔ Access Management Hierarchy	Pass
✔ Arithmetic Over/Under Flows	Pass
✔ Delegatecall Unexpected Ether	Pass
✔ Default Public Visibility	Pass
✔ Hidden Malicious Code	Pass
✔ Entropy Illusion (Lack of Randomness)	Pass
✔ External Contract Referencing	Pass
✔ Short Address/ Parameter Attack	Pass
✔ Unchecked CALL Return Values	Pass
✔ Race Conditions / Front Running	Pass
✔ General Denial Of Service (DOS)	Pass
✔ Uninitialized Storage Pointers	Pass
✔ Floating Points and Precision	Pass
✔ Tx.Origin Authentication	Pass
✔ Signatures Replay	Pass
✔ Pool Asset Security (backdoors in the underlying ERC-20)	Pass

	game\contracts TokenBase.sol BlueprintNFT.sol ParcelNFT.sol
✔ Re-entrancy	Pass
✔ Access Management Hierarchy	Pass
✔ Arithmetic Over/Under Flows	Pass
✔ Delegatecall Unexpected Ether	Pass
✔ Default Public Visibility	Pass
✔ Hidden Malicious Code	Pass
✔ Entropy Illusion (Lack of Randomness)	Pass
✔ External Contract Referencing	Pass
✔ Short Address/ Parameter Attack	Pass
✔ Unchecked CALL Return Values	Pass
✔ Race Conditions / Front Running	Pass
✔ General Denial Of Service (DOS)	Pass
✔ Uninitialized Storage Pointers	Pass
✔ Floating Points and Precision	Pass
✔ Tx.Origin Authentication	Pass
✔ Signatures Replay	Pass
✔ Pool Asset Security (backdoors in the underlying ERC-20)	Pass

		common\contracts\tokens\extensions ERC721BatchRetriever.sol ERC2981Royalty.sol ERC4906.sol
✔	Re-entrancy	Pass
✔	Access Management Hierarchy	Pass
✔	Arithmetic Over/Under Flows	Pass
✔	Delegatecall Unexpected Ether	Pass
✔	Default Public Visibility	Pass
✔	Hidden Malicious Code	Pass
✔	Entropy Illusion (Lack of Randomness)	Pass
✔	External Contract Referencing	Pass
✔	Short Address/ Parameter Attack	Pass
✔	Unchecked CALL Return Values	Pass
✔	Race Conditions / Front Running	Pass
✔	General Denial Of Service (DOS)	Pass
✔	Uninitialized Storage Pointers	Pass
✔	Floating Points and Precision	Pass
✔	Tx.Origin Authentication	Pass
✔	Signatures Replay	Pass
✔	Pool Asset Security (backdoors in the underlying ERC-20)	Pass

		common\contracts\tokens\extensions MintableToken.sol TokenMetadata.sol TokenRoles.sol
✔	Re-entrancy	Pass
✔	Access Management Hierarchy	Pass
✔	Arithmetic Over/Under Flows	Pass
✔	Delegatecall Unexpected Ether	Pass
✔	Default Public Visibility	Pass
✔	Hidden Malicious Code	Pass
✔	Entropy Illusion (Lack of Randomness)	Pass
✔	External Contract Referencing	Pass
✔	Short Address/ Parameter Attack	Pass
✔	Unchecked CALL Return Values	Pass
✔	Race Conditions / Front Running	Pass
✔	General Denial Of Service (DOS)	Pass
✔	Uninitialized Storage Pointers	Pass
✔	Floating Points and Precision	Pass
✔	Tx.Origin Authentication	Pass
✔	Signatures Replay	Pass
✔	Pool Asset Security (backdoors in the underlying ERC-20)	Pass

		common\contracts\tokens ERC721Default.sol ERC1155Default.sol
✔	Re-entrancy	Pass
✔	Access Management Hierarchy	Pass
✔	Arithmetic Over/Under Flows	Pass
✔	Delegatecall Unexpected Ether	Pass
✔	Default Public Visibility	Pass
✔	Hidden Malicious Code	Pass
✔	Entropy Illusion (Lack of Randomness)	Pass
✔	External Contract Referencing	Pass
✔	Short Address/ Parameter Attack	Pass
✔	Unchecked CALL Return Values	Pass
✔	Race Conditions / Front Running	Pass
✔	General Denial Of Service (DOS)	Pass
✔	Uninitialized Storage Pointers	Pass
✔	Floating Points and Precision	Pass
✔	Tx.Origin Authentication	Pass
✔	Signatures Replay	Pass
✔	Pool Asset Security (backdoors in the underlying ERC-20)	Pass

CODE COVERAGE AND TEST RESULTS FOR ALL FILES, PREPARED BY VIEWPOINT LABS TEAM

Contract: CountersPlus

- ✓ starts at zero

incBy

- ✓ increase the counter by 1
- ✓ increase the counter by 5
- ✓ increase the counter by amount multiple times (61ms)
- ✓ reverts on overflow (88ms)

decBy

- ✓ decrease the counter by 1 (71ms)
- ✓ decrease the counter by 5
- ✓ decrease the counter by amount multiple times (50ms)
- ✓ reverts on underflow

Contract: ERC1155Redeemable

- ✓ should have the owner as redeemer by default
- ✓ should be able to set the redeemer
- ✓ should be able to set the redeemer to 0x0
- ✓ should revert if redeeming by non-redeemer (48ms)
- ✓ should revert if trying redeem more than balance

Contract: TitleDeed Exchanger

- ✓ should deploy

Configuration

- ✓ should have a title deeds contract
- ✓ should have a parcel nft contract

exchange

- ✓ should exchange a title deed for a parcel nft (63ms)

TitleDeedsExchanger

- ✓ Owner must be able to change operator
- ✓ invalid TitleDeeds address (53ms)
- ✓ issueWithBonus (75ms)
- ✓ issueBatchWithBonus (85ms)

Contract: TitleDeeds

- ✓ should support ERC2981 interface

Token

- ✓ Should return correct uri for each token id (51ms)
- ✓ Should be able to set base uri (41ms)
- ✓ All prices are set
- ✓ Should have positive supply
- ✓ FT should have supply more than 1
- ✓ District title deed should have supply 1
- ✓ Should return correct token max supply

Title Deeds Reserve

- ✓ Should be able to reserve title deeds
- ✓ should revert if called by non-owner
- ✓ should revert when trying to reserve title deeds when already reserved
- ✓ should have Rok's House minted to the owner
- ✓ should have 10 common parcels
- ✓ should have 3 rare parcels
- ✓ should have 1 epic parcel
- ✓ should be able to transfer reserved title deeds (103ms)

Allowlist

- ✓ Should be able to add an address to the allowlist
- ✓ Should be able to remove an address from the allowlist
- ✓ Should be able to add multiple addresses to the allowlist
- ✓ Should be able to remove multiple addresses from the allowlist

Mint

- ✓ Should have inactive mint state by default
- ✓ Should change mint state to private
- ✓ Should change mint state to public
- ✓ Should change mint state to ended
- ✓ Should be able to mint a token
- ✓ Should be able to mint multiple tokens (44ms)

- ✓ should mint batch by owner
- ✓ should revert mint batch to zero address
- ✓ should revert mint batch if length of ids and amounts are not equal
- ✓ Should return correct mint couter for each token (146ms)
- ✓ should revert if amount more than tokens left
- ✓ should revert if trying to mint 0 tokens
- ✓ Should revert when mint is not active
- ✓ Should revert when mint is ended
- ✓ Should revert if address is not allowlisted
- ✓ should revert if eth sent is not enough
- ✓ should revert if eth sent ot much
- ✓ Owner should be able to mint to address for free
- ✓ should revert if owner trying to mint ot 0x0 address
- ✓ should revert if trying to mint non-existing token
- ✓ should revert if trying to mint non-existing token by owner to an address
- ✓ Should revert if amount could make the stack overflow

Redeem

- ✓ Should return owner as redeemer by default
- ✓ Should be able to change redeemer
- ✓ Should be able to redeem a token
- ✓ Should revert if redeem is initiated by non-owner of the title deed
- ✓ should revert if trying to redeem from 0x0 address
- ✓ should revert if trying to redeem non-existing token

Owner

- ✓ Should be correct owner
- ✓ Should be able to withdraw ethers

Royalty

- ✓ Should return correct royalty info

TitleDeedsRedeemer

- ✓ invalid TitleDeeds address
- ✓ supplyOf
- ✓ exchangeBatch different length arrays
- ✓ exchangeBatch

Contract: ParcelNFT

Deployment

✓ Should set the right owner

Metadata

✓ should have correct name
✓ should have correct symbol
✓ should have correct contractURI
✓ should return correct tokenURI
✓ should return correct tokenURI string for tokenURI if token does not exist

Contract: CountersPlus

✓ starts at zero

incBy

✓ increase the counter by 1
✓ increase the counter by 5
✓ increase the counter by amount multiple times
✓ reverts on overflow

decBy

✓ decrease the counter by 1
✓ decrease the counter by 5
✓ decrease the counter by amount multiple times
✓ reverts on underflow

Contract: ERC1155Redeemable

✓ should have the owner as redeemer by default
✓ should be able to set the redeemer
✓ should be able to set the redeemer to 0x0
✓ should revert if redeeming by non-redeemer
✓ should revert if trying redeem more than balance

Test ERC2981Royalty

✓ Get signers
✓ Get signers and create contract (57ms)
✓ Check that only owner and royaltySetter can access functions (88m
✓ Check default royalty parameters
✓ Set specific token parameters
✓ Check specific royalty
✓ Delete default royalty
✓ Check default and specific royalties

- ✓ Delete specific royalty
- ✓ Check specific royalty
- ✓ Check ERC165

Test_ERC4906

- ✓ Check MetadataUpdate event
- ✓ Check supportsInterface function

Test ERC721BatchRetriver

- ✓ Create contract
- ✓ Mint tokens and check batch retriever (368ms)

Test ERC721Default

- ✓ Deploy default contract and get signers (64ms)
- ✓ Set URIs
- ✓ Try to set minter
- ✓ Try to mess up with royalties
- ✓ Check supportsInterface
- ✓ Check token uri

Test ERC165 linearization

- ✓ Check interface IERC20
- ✓ Check interface IERC165
- ✓ Check interface IERC721
- ✓ Check interface IERC721Mintable
- ✓ Check interface IERC1155
- ✓ Check interface IERC2981
- ✓ Check interface IERC4906

Test ERC721Metadata

- ✓ Create contract (52ms)
- ✓ Check default uris
- ✓ Check token URI
- ✓ Check token URI for non existent token (must be present)
- ✓ Update base URI
- ✓ Update contract URI
- ✓ Check that only URI updater can update URI (44ms)
- ✓ Check interface support
- ✓ Check notify metadata update
- ✓ Check batch metadata update
- ✓ Check that only METADATA_UPDATER can notify metadata up date (42ms)

- ✓ Check that only METADATA_UPDATER can notify metadata update for multiple tokens using notifyMetadataUpdateForMultipleTokens
- ✓ Check that only owner can add URI updater
- ✓ Check that only owner can revoke URI updater
- ✓ Check that only owner can add metadata updater
- ✓ Check that only owner can revoke metadata updater

Test ERC721Mintable

- ✓ Check that only owner or minter can mint
- ✓ Mint single token to user
- ✓ Mint multiple tokens to user
- ✓ Trying to re-mint already existing tokens
- ✓ Check interfaces (ERC165)

Test ERC721Roles

- ✓ Check that owner is owner
- ✓ Check that owner can add owners
- ✓ Check that owner cannot revoke himself
- ✓ Check that owner can revoke owners
- ✓ Check that owner cannot grant role
- ✓ Check that owner cannot revoke role
- ✓ Check that owner cannot renounce role

FILE	% STMTS	% BRANCH	% FUNCS
TitleDeedExchanger.sol	100	84,09	100
TitleDeedRedeemer.sol	100	56,25	100
TitleDeeds.sol	100	100	100
BaseNFT.sol	60	100	66,67
BlueprintNFT.sol	0	100	0
ParcelNFT.sol	100	100	100
ERC2981Royalty.sol	100	100	100
ERC4906.sol	100	100	100
ERC721BatchRetriever.sol	100	100	100
ERC721Metadata.sol	90	83,33	90
ERC721Mintable.sol	100	100	100
ERC721Roles.sol	100	100	100
CountersPlus.sol	100	100	100
ERC1155Redeemable.sol	66,67	50	80
ERC721Default.sol	100	100	100
All files	**76,66**	**90,05**	**77,77**

CODE COVERAGE AND TEST RESULTS FOR ALL FILES, PREPARED BY BLAIZE SECURITY TEAM

TitleDeedsFindings

- ✓ should not emit Redeem event when calling exchangeBatch with 0 amount
- ✓ should not increment redeemIndexer when calling exchangeBatch with 0 amount
- ✓ should revert setting zero address as operator

Contract: TitleDeedsExchanger

- ✓ Should set new operator
- ✓ Should revert if TitleDeeds address is invalid (49ms)
- ✓ Should mint new NFT with bonus
- ✓ Should mint NFT for few users with bonus (63ms)
- ✓ Should revert if msg.sender is not owner
- ✓ Should revert batch issue if arrays length don't match
- ✓ Should revert issue new nft if nonce is already taken
- ✓ Should revert if tokens array is empty
- ✓ Should issue batch without bonus

Contract: TitleDeedsRedeemer

- ✓ Invalid TitleDeeds address
- ✓ supplyOf
- ✓ exchangeBatch different length arrays
- ✓ exchangeBatch

Contract: BlueprintNFT

- ✓ Should support ERC2981 interface
- ✓ Should set uri properly
- ✓ Should change registry
- ✓ Should change registry state
- ✓ Should transfer tokens

Contract: ParcelNFT

- ✓ Should support ERC2981 interface
- ✓ Should set uri properly
- ✓ Should change registry
- ✓ Should change registry state
- ✓ Should transfer tokens

Contract: ERC721Metadata

- ✓ should revert when not uriUpdater tries to set Base URI
- ✓ should revert when not uriUpdater tries to set Contract URI
- ✓ should revert when not owner tries to add URI updater
- ✓ should revert when not owner tries to remove URI updater
- ✓ should revert when not owner tries to add Metadata updater
- ✓ should revert when not owner tries to remove Metadata updater

Contract: ERC721Mintable

- ✓ # mint
- ✓ should revert when not minter tries to mint
- ✓ # mintBatchTo
- ✓ should revert when minter tries to mint batch of some of already minted Ids

Contract: ERC721Roles

- ✓ # addOwner
- ✓ should revert when not owner tries to add owner
- ✓ # revokeOwner
- ✓ should revert when not owner tries to add owner

Scenario: Users mint NFTs, Redeem NFTs, back-end calls issueBatchWithBonus -> Users receive Parcels and BluePrints.

- ✓ User1 purcases 30 TD_PARCEL_COMMON and 20 TD_PARCEL_RARE ...
- ✓ User2 purcases 10 TD_PARCEL_EPIC and 5 TD_PARCEL_LEGENDARY ...
- ✓ User1 and user2 redeem their tokens...
- ✓ Executing issueBatchWithBonus...
- ✓ Users should receive Parceks and BluePrints.

FILE	% STMTS	% BRANCH	% FUNCS
TitleDeedExchanger.sol	100	100	100
TitleDeedRedeemer.sol	100	100	100
BaseNFT.sol	100	100	100
BlueprintNFT.sol	100	100	100
ParcelNFT.sol	100	100	100
ERC2981Royalty.sol	100	100	100
ERC4906.sol	100	100	100
ERC721BatchRetriever.sol	100	100	100
ERC721Metadata.sol	90	83,33	90
ERC721Mintable.sol	100	100	100
ERC721Roles.sol	100	100	100
CountersPlus.sol	100	100	100
ERC721Default.sol	100	100	100
All files	**97,11**	**95,5**	**98**

DISCLAIMER

The information presented in this report is an intellectual property of the customer including all presented documentation, code databases, labels, titles, ways of usage as well as the information about potential vulnerabilities and methods of their exploitation. This audit report does not give any warranties on the absolute security of the code. Blaize.Security is not responsible for how you use this product and does not constitute any investment advice.

Blaize.Security does not provide any warranty that the working product will be compatible with any software, system, protocol or service and operate without interruption. We do not claim the investigated product is able to meet your or anyone else requirements and be fully secure, complete, accurate and free of any errors and code inconsistency.

We are not responsible for all subsequent changes, deletions and relocations of the code within the contracts that are the subjects of this report.

You should perceive Blaize.Security as a tool which helps to investigate and detect the weaknesses and vulnerable parts that may accelerate the technology improvements and faster error elimination.