

# Audit Report Eldorado Coin

08.08.2025

## Overview

The auditing division of TokenMinds consists of a highly experienced team of auditors and developers specializing in securing and optimizing smart contracts on all EVM-based, Solana, and Sui.

Our team is dedicated to delivering thorough and precise audits, putting in 100% effort to ensure that each contract is as secure and efficient as possible. We follow a structured four-phase testing process, starting with a manual review to analyze the code for potential issues, followed by functional testing to ensure correct operation. We then conduct security testing to identify vulnerabilities, and finally, perform fuzz testing with randomized inputs to uncover hidden bugs. This comprehensive approach enables TokenMinds to provide a strong, secure foundation for blockchain projects.

## Table of Contents

[Overview](#)

[Table of Contents](#)

[1. Disclaimer](#)

[2. Project Overview](#)

[3. Risk Classification](#)

[3.1. Impact](#)

[3.2. Likelihood](#)

[4. Auditing Process](#)

[5. Methodology](#)

[5.1. Source Code Examination](#)

[5.2. Technical Assessment and Analysis](#)

[5.3. Industry Standard Compliance](#)

[5.4. Detailed Remediation Guidelines](#)

[6. Graphical Representation of Eldorado Coin](#)

[7. Audit Executive Summary](#)

[Audit Version](#)

[Commit hash, Code base deployment and disclaimer](#)

[Public functions found in Smart contract](#)

[1. TransferChecked](#)

[2. BurnChecked](#)

[3. ApprovedChecked](#)

[4. Revoke](#)

[5. SetAuthority](#)

[6. CloseAccount](#)

[7. InitializeAccount3](#)

[8. Reallocate](#)

[Admin authority functions found in Smart contract](#)

[1. UpdateMetadata](#)

[2. WithdrawWithheldTokenFromAccounts](#)

[3. WithdrawWithheldTokensFromMint](#)

[4. HarvestWithheldTokensToMint](#)

[Finding Count](#)

[8. Overall Security - ZERO Bugs](#) 

[Invariant Testing and Functional Testing with Mocha Test](#)

[Other Checks](#)

[9. Conclusion](#)

## 1. Disclaimer

At TokenMinds, our audit process is designed to enhance the quality and security of smart contracts, helping to reduce the risks associated with cryptographic tokens and blockchain technology. However, no audit can guarantee the complete absence of bugs or vulnerabilities. Blockchain technology inherently presents a high level of ongoing risk, and while we aim to identify and mitigate as many issues as possible, absolute security cannot be assured.

This report should not be relied upon for investment decisions or considered as investment advice. Each company or individual is responsible for conducting their own due diligence and maintaining continuous security measures. TokenMinds does not provide any warranties or guarantees regarding the security or functionality of the technology we review, and we encourage follow-up reviews, bug bounty programs, and ongoing monitoring for the best security practices

## 2. Project Overview

<b>Project Name</b>	Digi World   Eldorado Coin
<b>Website</b>	<a href="https://eldoradodigiworld.com">https://eldoradodigiworld.com</a>
<b>Protocol Type</b>	Solana Token-2022 Program
<b>Chain</b>	Solana
<b>Language</b>	Typescript
<b>Deployment</b>	<a href="https://explorer.solana.com/address/6BnMbiZgVW1LuMtUXdvNigU6Sykf4MsU53XRgtWEDAVx">https://explorer.solana.com/address/6BnMbiZgVW1LuMtUXdvNigU6Sykf4MsU53XRgtWEDAVx</a>

### 3. Risk Classification

#### 3.1. Impact

- 3.1.1. **High:** Leads to a major loss in assets of protocol or affect large group of users
- 3.1.2. **Medium:** Little amount of funds can be lost or a core functionality of protocol is only affected
- 3.1.3. **Low:** Unexpected behavior in protocol without any loss in finances or some non critical protocol functions.

#### 3.2. Likelihood

- 3.2.1. **High:** Attack path is very likely to be possible with lost of asset outweighing the cost of attack
- 3.2.2. **Medium:** Conditionally Incentivised attack with high likelihood
- 3.2.3. **Low:** The cost of attack may outweigh the lost assets with low likelihood of happening

Severity	Impact: High	Impact: Medium	Impact: Low
Likelihood: High	Critical	High	Medium
Likelihood: Medium	High	Medium	Low
Likelihood: Low	Medium	Low	Low

## 4. Auditing Process

TokenMinds conducts a comprehensive four-level testing process for smart contracts, structured in a pyramid approach. The process begins with a meticulous manual review of the smart contract, ensuring that the code follows best practices and is free from apparent errors.

This is followed by functional testing, where the contract's behavior is thoroughly checked to confirm that it performs its intended functions accurately. Next, the smart contract undergoes rigorous security testing, identifying and addressing potential vulnerabilities that could be exploited. Finally, TokenMinds concludes the process with fuzz testing, which stress-tests the core functionalities by inputting random or unexpected data to ensure stability and robustness under diverse conditions.



This multi-tiered approach ensures a well-rounded assessment of the smart contract's security and functionality.

## 5. Methodology

Our comprehensive audit methodology encompasses several key phases:

### 5.1. Source Code Examination

- 5.1.1. Evaluating provided documentation, technical specifications, and implementation guidelines to fully grasp the program's architecture, complexity, and intended functionality
- 5.1.2. Conducting thorough line-by-line code analysis to identify potential security weaknesses and vulnerabilities
- 5.1.3. Cross-referencing implementation against provided specifications to verify accurate functionality alignment

### 5.2. Technical Assessment and Analysis

- 5.2.1. Implementing test coverage evaluation to measure code execution effectiveness and verify comprehensive testing parameters
- 5.2.2. Performing dynamic analysis to evaluate contract behavior under various input conditions and execution paths

### 5.3. Industry Standard Compliance

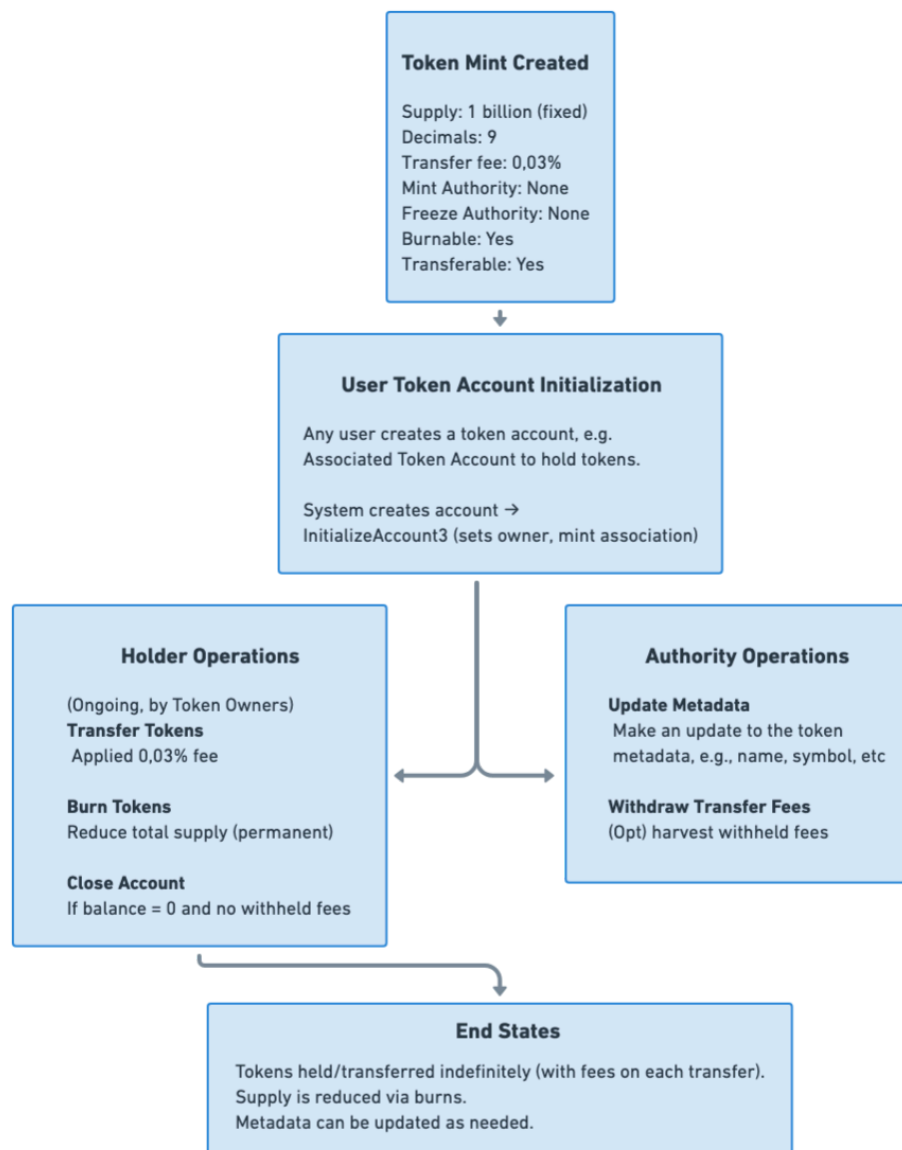
- 5.3.1. Assessing contract implementation against established blockchain security patterns
- 5.3.2. Evaluating code structure for optimization opportunities regarding gas efficiency, readability, and long-term maintenance
- 5.3.3. Incorporating latest security insights from both industry practitioners and academic research

### 5.4. Detailed Remediation Guidelines

- 5.4.1. Providing specific, implementable security enhancement recommendations
- 5.4.2. Outlining clear steps for vulnerability mitigation and overall contract strengthening



## 6. Graphical Representation of Eldorado Coin



## 7. Audit Executive Summary

### Audit Version

Audit Version	Delivery Date
v.1.0	6th August, 2025
v.1.1 (latest)	8th August, 2025

### Commit hash, Code base deployment and disclaimer

Filename	Commit Hash
spl-token-deployer/spl/mint.ts	36798edd17985307944b748ae46f04d6cdc72dbe

### SPL Metadata and authority

Key	Value
Address	<a href="#">6BnMbiZgVW1LuMtUXdvNjgU6Sykf4MsU53XRgtWEDAVx</a>
Supply	1 Billion (fixed supply)
Decimals	9
Burnable	Yes (will reduce the supply)
Transferable	Yes (with
Mintable	No (fixed supply)
Mint authority	None
Freeze authority	None

Metadata update authority	<a href="#">B5pTfBPemSZFp8UJvVj3v5NwUPUhiicTiv4zGzHcqAoC</a>
Transfer tax config authority	Renounced (current 0,03%)
Transfer tax withdraw authority	<a href="#">B5pTfBPemSZFp8UJvVj3v5NwUPUhiicTiv4zGzHcqAoC</a>

**Disclaimer:** Snapshots taken on 7th August 2025, files with a different hash value or contract address than those listed above have been modified after the audit by TokenMinds. TokenMinds is not responsible for auditing any files other than those specified above.

## Public functions found in Smart contract

### 1. TransferChecked

Transfers tokens to another account, checking decimals and applying the **0.03%** transfer fee (fee withheld in the recipient's account as **withheld\_amount**).

**Constraints:** Source must have sufficient balance; fee is automatic and cannot be bypassed

**Effect:** Tokens move, fee collected for later withdrawal/harvest

### 2. BurnChecked

Burns (destroys) tokens from the owner's account, reducing total supply.

**Constraints:** Enabled (burnable: yes); owner must have sufficient balance.

**Effect:** Permanent supply reduction.

### 3. ApprovedChecked

Approves a delegate to transfer a specified amount of tokens on behalf of the owner.

**Constraints:** None specific; delegates can then use TransferChecked.

**Effect:** Sets delegation authority.

### 4. Revoke

Revokes any existing delegate authority.

**Constraints:** Can be called by the owner or current delegate.

**Effect:** Removes delegation.

## 5. **SetAuthority**

Changes the close authority or delegate authority on a token account (not mint-level).

**Constraints:** Only for user-owned accounts; cannot set mint or freeze authorities (already none).

**Effect:** Updates control over the account.

## 6. **CloseAccount**

Closes a token account and recovers rent (SOL).

**Constraints:** Balance must be 0; no withheld fees (or withdraw them first).

**Effect:** Account deleted; rent refunded.

## 7. **InitializeAccount3**

Initializes a new token account for holding this token.

**Constraints:** Must be called when creating a new account.

**Effect:** Prepares account for tokens.

## 8. **Reallocate**

Resizes a token account to add/remove extensions (if needed).

**Constraints:** Rarely used unless adding custom extensions post-creation.

**Effect:** Adjusts account size.

## Admin authority functions found in Smart contract

### 1. UpdateMetadata

Updates the token's metadata (e.g., name, symbol, URI).

**Constraints:** Only callable by metadata update authority (B5pTfBPemSZFp8UJvVj3v5NwUPUhiicTiv4zGzHcqAoC).

**Effect:** Changes token metadata without affecting supply or balances.

### 2. WithdrawWithheldTokenFromAccounts

Withdraws accumulated transfer fees from specified token accounts to a recipient account.

**Constraints:** Only callable by transfer tax withdrawal authority (B5pTfBPemSZFp8UJvVj3v5NwUPUhiicTiv4zGzHcqAoC).

**Effect:** Moves fees to authority's control.

### 3. WithdrawWithheldTokensFromMint

Withdraws withheld fees that have been harvested to the mint.

**Constraints:** Only callable by withdrawal authority (B5pTfBPemSZFp8UJvVj3v5NwUPUhiicTiv4zGzHcqAoC).

**Effect:** Clears withheld from mint.

### 4. HarvestWithheldTokensToMint

Moves withheld fees from token accounts to the mint for later withdrawal (optional step before WithdrawWithheldTokensFromMint).

**Constraints:** Only callable by withdrawal authority (B5pTfBPemSZFp8UJvVj3v5NwUPUhiicTiv4zGzHcqAoC).

**Effect:** Aggregates fees at mint level.

### Finding Count

Severity	Amount
Critical	0
Medium	0
Low	0
<b>Total Findings</b>	<b>0</b>

*After an extensive and intensive round of testing, TokenMinds reported **ZERO BUGS** in the overall contract, ensuring it is ready for secure mainnet deployment. ✓*










## 8. Overall Security - ZERO Bugs ✓

After undergoing four rounds of rigorous testing, the Eldorado Coin smart contract has been confirmed to be free of any bugs or vulnerabilities, demonstrating a high level of security and reliability. The comprehensive tests involved stress testing all key functions, including but not limited to:



- **ApproveChecked:** Allows a user to delegate token spending rights to another account.
- **TransferChecked:** Facilitates the transfer of tokens between two parties, applying the 0.03% transfer fee.
- **BurnChecked:** Permits token holders to destroy tokens, reducing the total supply.
- **Revoke:** Removes any existing delegate authority.
- **Update (Metadata):** Allows the metadata authority to update token metadata.
- **WithdrawWithheldTokensFromAccounts:** Enables the transfer tax withdrawal authority to withdraw accumulated fees.

Each of these functions has been thoroughly validated, and no security flaws or issues were detected during the testing phases. This confirms that the SPL Token mint is secure, reliable, and ready for use without concerns of exploitation or malfunction.

### Invariant Testing and Functional Testing with Mocha Test

Function Name	Pass/Fail  	Optimization Remark
TransferChecked()		Optimized
ApproveChecked()		Optimized
Revoke()		Optimized
BurnChecked()		Optimized
Update (Metadata)		Optimized
burn()		Optimized
WithdrawWithheldTokensFromAccounts()		Optimized

### Other Checks


Checks	Description	Presence	Remarks
Upgradeability	A contract that can be upgraded to modify their code, while preserving their state, address and balance		This token mint cannot be upgraded (uses fixed Token-2022 program)
Mints	Ability to mint new tokens		The mint issues 1,000,000,000 tokens at creation and cannot mint any further (mint authority: none)



Burn	Token holders can burn tokens, reducing supply	✓	N/A
Blacklist	No ability to blacklist or freeze addresses	✗	No freeze authority present
Pausable	The token operations cannot be paused	✗	No pause function in SPL Token-2022
Centralized Privileges	Authorities for metadata updates and fee withdrawals	✓	Metadata update authority and transfer tax withdraw authority exist (both: B5pTfBPemSZFp8UJvVj3v5NwUPUhiicTiv4zGzHcqAoC); transfer tax config renounced
Withdraw	Withdraw function to transfer accumulated fees from transfers	✓	WithdrawWithheldTokens* instructions available to authority (for 0.03% transfer fees)

## 9. Conclusion

TokenMinds has concluded the audit of the Eldorado Coin. Following a comprehensive review, no bugs or vulnerabilities were found in the smart contract. The contract exhibits a highly optimized structure, adhering to best practices in both security and efficiency. The audit included a detailed manual code analysis, functional and security testing, as well as fuzz testing to ensure the contract's robustness.

Company name	<b>PURPLE MINDZ MEDIA PTE. LTD</b>
Company address	<b>139 CECIL STREET, #03-10, YSY BUILDING, Singapore 069539</b>
Signature of the company representative	<b>Rob Eijgenraam</b> 
Company stamp	<b>TOKEN</b> <b>MINDS</b>