DEMtToken – Smart Contract Audit Report

Executive Summary

Der DEMtToken Smart Contract wurde vollständig auditiert, basierend auf Solidity 0.8.25 und OpenZeppelin Contracts v5.5.0. Während des Audits wurden keine kritischen oder hochpriorisierten
Schwachstellen identifiziert. Die Implementierung erfüllt die Anforderungen eines sicheren ERC20/TRC20-Tokens
und folgt den Best Practices der OpenZeppelin-Bibliotheken.

Audit Scope

• Analyse der Datei DEMtToken.sol

• Untersuchung des Flatten-Files

• Prüfung der Rollenlogik (AccessControl)

• Untersuchung der Pausable-Integration (_update Hook)

• Überprüfung der Mint-Mechanik und Berechtigungen

• Prüfung auf klassische Smart Contract Risiken

Compiler & Build-Einstellungen

• Solidity Version: 0.8.25

• Optimizer: enabled, runs = 200

• OpenZeppelin Version: v5.5.0

• Netzwerk: TRON (Nile & Mainnet kompatibel)

Contract Architektur

• ERC20 / TRC20 Standard

• Rollen: DEFAULT_ADMIN_ROLE, PAUSER_ROLE, MINTER_ROLE

• Pausable integriert über neuen Hook _update (OZ 5.x)

• Minting nur durch MINTER_ROLE

• Keine externen unsicheren Calls

• Keine Upgradable-Mechanik (immutable Contract)

Audit Findings

Critical: Keine

High: Keine

Medium: Keine

Low: Empfehlung: Einsatz von Hardware-Wallet für Admin und Minter.

Info: Gute Code-Struktur, sauber dokumentiert.

Positive Observations

• Nutzung von OpenZeppelin 5.5 – industrieweit anerkannter Sicherheitsstandard.

• Klares Rollenmodell verhindert unbefugtes Minting oder Pausieren.

• Keine Reentrancy-Probleme durch saubere Architektur.

• Pausable korrekt via _update integriert.

Empfehlungen

• Admin-Rolle ausschließlich über Hardware-Wallet sichern.

• Minter-Rolle vorzugsweise MultiSig oder Agent-Wallet.

• Source-Code & Flatten öffentlich dokumentieren (GitHub).

• Deployment-Hash & Build-Konfiguration archivieren.

Abschließendes Urteil

Der DEMtToken Contract gilt als sicher, stabil und entspricht den Best Practices für ERC20/TRC20 Tokens.
Der Contract kann ohne Bedenken auf dem Mainnet eingesetzt werden.