DRAFT

NOT VALID UNTIL FINALIZED & PUBLISHED ON DESSERTSWAP.FINANCE

DESSERT FINANCE

DTGC

Pulsechain Audit

Performed at block **SAMPLE**

PERFORMED BY DESSERT FINANCE

FOR CONTRACT ADDRESSES: 0X22F0DE89EF26AE5C03CB43543DF5BBD8CB8D0231
0XEBC6802E6A2054FBF2CB450AEC5E2916965B1718

VERIFY THIS REPORT IN THE @DESSERTSWAP TELEGRAM, CLICK HERE

INITIAL DISCLAIMER

Dessert Finance provides due-diligence project audits for various projects. Dessert Finance in no way guarantees that a project will not remove liquidity, sell off team supply, or otherwise exit scam.

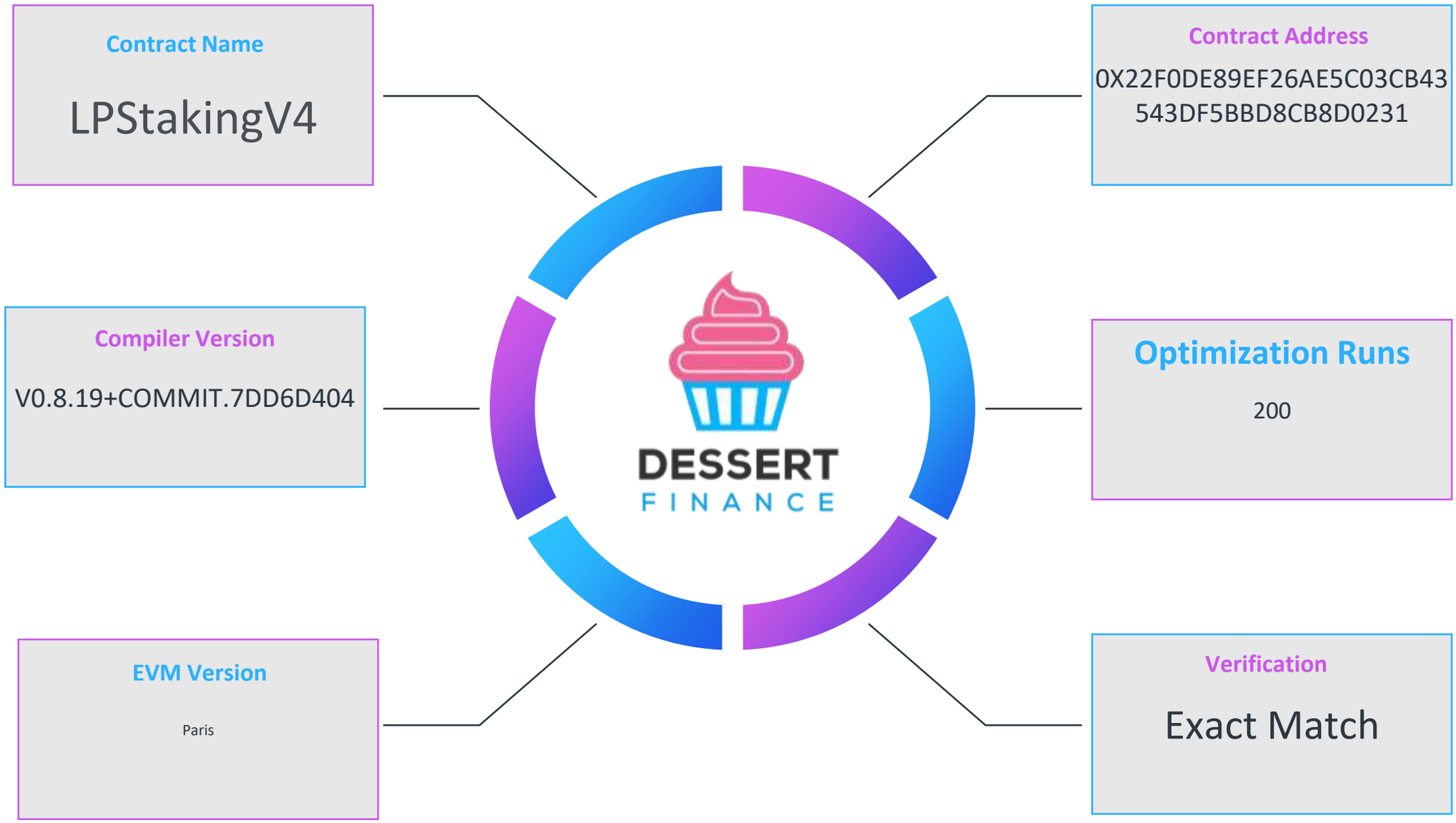
Dessert Finance does the legwork and provides public information about the project in an easy-to-understand format for the common person.

Agreeing to an audit in no way guarantees that a team will not remove ***all*** liquidity (“Rug Pull”), remove liquidity slowly, sell off tokens, quit the project, or completely exit scam. There is also no way to prevent private sale holders from selling off their tokens. It is ultimately your responsibility to read through all documentation, social media posts, and contract code of each individual project to draw your own conclusions and set your own risk tolerance.

Dessert Finance in no way takes responsibility for any losses, nor does Dessert Finance encourage any speculative investments. The information provided in this audit is for information purposes only and should not be considered investment advice. Dessert Finance does not endorse, recommend, support, or suggest any projects that have been audited. An audit is an informational report based on our findings, We recommend you do your own research, we will never endorse any project to invest in.

Table of Contents

Contract Code Audit –Overview (LPStakingV4)



Contract Code Audit – Overview

Dessert Finance was commissioned to perform an audit on LPStakingV4

```
/**
 * @title LPStakingV4
 * @notice LP Staking with UNLIMITED stakes per user - Remix Compatible
 */
contract LPStakingV4 is ReentrancyGuard, Ownable, Pausable {
    using SafeERC20 for IERC20;

    struct Stake {
        uint256 amount;
        uint256 startTime;
        uint256 unlockTime;
        uint256 lockPeriod;
        uint256 aprBps;
        uint256 boostBps;
        uint8 lpType;
        bool isActive;
        uint256 lastClaimTime;
    }

    struct LPConfig {
        address lpToken;
        uint256 lockDays;
        uint256 baseAprBps;
        uint256 boostBps;
        uint256 minAmount;
        bool isActive;
    }

    IERC20 public immutable rewardToken;
    mapping(address => Stake[]) public userStakes;
    mapping(uint8 => LPConfig) public lpConfigs;
    mapping(uint8 => uint256) public totalStakedByType;

    uint256 public totalStakers;
    uint256 public totalRewardsPaid;
    mapping(address => bool) public hasStaked;
    uint256 public earlyWithdrawFeeBps = 2000;
    address public feeRecipient;
```

Contract Address
0x22f0DE89Ef26AE5c03CB43543dF5Bbd8cb8d0231

Associated Token
Dump Tires Gold Coin (DTGC)

Source Code
Verified (Exact Match)

Contract Name
LPStakingV4

EVM Version
Paris

Compiler Version
v0.8.19+commit.7dd6d404

Optimization Enabled
Yes with 200 Runs

Code is truncated to fit the constraints of this document.
The code in its entirety can be viewed here.

The contract code is **verified** on Pulsechain Scan.

Contract Code Audit – Vulnerabilities Checked (LPStakingV4)

Vulnerability Tested	AI Scan	Human Review	Result
Compiler Errors	Complete	Complete	✓ Low Risk
Outdated Compiler Version	Complete	Complete	✓ Low Risk
Integer Overflow	Complete	Complete	✓ Low Risk
Integer Underflow	Complete	Complete	✓ Low Risk
Correct Token Standards Implementation	Complete	Complete	✓ Low Risk
Timestamp Dependency for Crucial Functions	Complete	Complete	✓ Timestamp used / Low Risk
Exposed _Transfer Function	Complete	Complete	✓ Low Risk
Transaction-Ordering Dependency	Complete	Complete	✓ Low Risk
Unchecked Call Return Variable	Complete	Complete	✓ Low Risk
Use of Deprecated Functions	Complete	Complete	✓ Low Risk
Unprotected SELFDESTRUCT Instruction	Complete	Complete	✓ Low Risk
State Variable Default Visibility	Complete	Complete	✓ Low Risk
Deployer Can Access User Funds	Complete	Complete	✓ Low Risk

The contract code is **verified** on pulsechain scan.

The vulnerabilities listed above were not found in the token's Smart Contract.

Contract Code Audit – Contract Ownership (LPStakingV4)

Contract Ownership has not been renounced at the time of Audit

The contract ownership is not currently renounced.

We have placed the contract owner address below for your viewing:

0xC1CD5a70815E2874D2db038F398f2D8939d8E87C

The address above has authority over the ownable functions within the contract.

This allows the owner to call certain functions within the contract. Any compromise to the owner wallet may allow these privileges to be exploited.

Contract Code Audit – Owner Accessible Functions (LPStakingV4)

Function Name	Parameters	Visibility	Audit Notes
setLPConfig	uint8 lpType, address lpToken, uint256 lockDays, uint256 baseAprBps, uint256 boostBps, uint256 minAmount, bool isActive	external	onlyOwner modifier is detected. Owner can call this function if the contract is not renounced.
setEarlyWithdrawFee	uint256 feeBps	external	onlyOwner modifier is detected. Owner can call this function if the contract is not renounced.
setFeeRecipient	address _feeRecipient	external	onlyOwner modifier is detected. Owner can call this function if the contract is not renounced.
pause		external	onlyOwner modifier is detected. Owner can call this function if the contract is not renounced.
unpause		external	onlyOwner modifier is detected. Owner can call this function if the contract is not renounced.
recoverTokens	address token, uint256 amount	external	onlyOwner modifier is detected. Owner can call this function if the contract is not renounced.

Owner is able to modify LP configuration, pause, and unpause settings. Please ensure owner wallet is secure with proper access authority. Bad actors may cause issues.

The functions listed above can be called by the contract owner.

If contract ownership has been renounced there is no way for the above listed functions to be called.

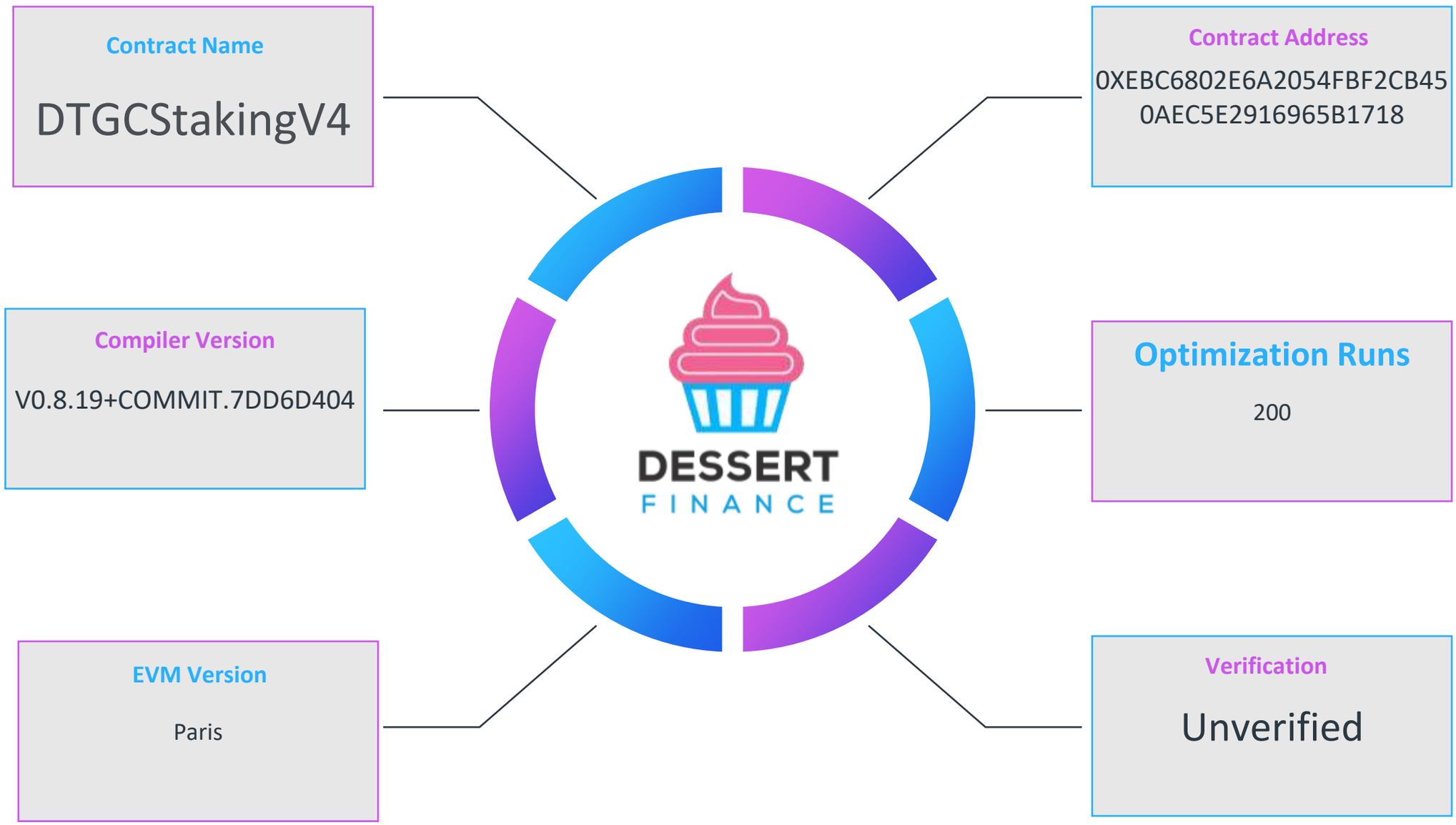
Contract Code Audit – Mint Functions (LPStakingV4)

This Contract Cannot Mint New Tokens.

We do understand that sometimes mint functions are essential to the functionality of the project.

A mint function was not found in the contract code.

Contract Code Audit –Overview (DTGCStakingV4)



Contract Code Audit – Overview

Dessert Finance was commissioned to perform an audit on DTGCStakingV4

```
/**
 * @title DTGCStakingV4
 * @notice DTGC Staking with UNLIMITED stakes per user - Remix Compatible
 */
contract DTGCStakingV4 is ReentrancyGuard, Ownable, Pausable {
    using SafeERC20 for IERC20;

    struct Stake {
        uint256 amount;
        uint256 startTime;
        uint256 unlockTime;
        uint256 lockPeriod;
        uint256 aprBps;
        uint256 bonusBps;
        uint8 tier;
        bool isActive;
        uint256 lastClaimTime;
    }

    struct TierConfig {
        uint256 lockDays;
        uint256 aprBps;
        uint256 minAmount;
        bool isActive;
    }

    IERC20 public immutable dtgcToken;
    mapping(address => Stake[]) public userStakes;
    mapping(uint8 => TierConfig) public tierConfigs;

    uint256 public totalStaked;
    uint256 public totalStakers;
    uint256 public totalRewardsPaid;
    mapping(address => bool) public hasStaked;
    uint256 public earlyWithdrawFeeBps = 2000;
    address public feeRecipient;
    uint256 public rewardPool;

    event Staked(address indexed user, uint256 indexed stakeIndex, uint256 amount, u
    event Withdrawn(address indexed user, uint256 indexed stakeIndex, uint256 amount
    event RewardsClaimed(address indexed user, uint256 indexed stakeIndex, uint256 r
    event EmergencyWithdraw(address indexed user, uint256 indexed stakeIndex, uint25
    event RewardPoolFunded(address indexed funder, uint256 amount);

    constructor(address _dtgcToken, address _feeRecipient) {
        require(_dtgcToken != address(0), "Invalid token");
        require(_feeRecipient != address(0), "Invalid fee recipient");

        dtgcToken = IERC20(_dtgcToken);
```

Contract Address
0xEbC6802e6a2054FbF2Cb450aEc5E2916965b1718

Associated Token
Dump Tires Gold Coin (DTGC)

Source Code
Unverified – Repository Audit

Contract Name
DTGCStakingV4

EVM Version
Paris

Compiler Version
v0.8.19+commit.7dd6d404

Optimization Enabled
Yes with 200 Runs

Code is truncated to fit the constraints of this document.
The code in its entirety can be viewed here.

The contract code is **not verified** on Pulsechain Scan.

Contract Code Audit – Vulnerabilities Checked (DTGCStakingV4)

Vulnerability Tested	AI Scan	Human Review	Result
Compiler Errors	Complete	Complete	✓ Low Risk
Outdated Compiler Version	Complete	Complete	✓ Low Risk
Integer Overflow	Complete	Complete	✓ Low Risk
Integer Underflow	Complete	Complete	✓ Low Risk
Correct Token Standards Implementation	Complete	Complete	✓ Low Risk
Timestamp Dependency for Crucial Functions	Complete	Complete	✓ Timestamp Used / Low Risk
Exposed _Transfer Function	Complete	Complete	✓ Low Risk
Transaction-Ordering Dependency	Complete	Complete	✓ Low Risk
Unchecked Call Return Variable	Complete	Complete	✓ Low Risk
Use of Deprecated Functions	Complete	Complete	✓ Low Risk
Unprotected SELFDESTRUCT Instruction	Complete	Complete	✓ Low Risk
State Variable Default Visibility	Complete	Complete	✓ Low Risk
Deployer Can Access User Funds	Complete	Complete	✓ Low Risk

The contract code isn not **verified** on pulsechain scan.

The vulnerabilities listed above were not found in the token's Smart Contract.

Contract Code Audit – Contract Ownership (DTGCStakingV4)

Contract Ownership has not been renounced at the time of Audit

The contract ownership is not currently renounced.

We have placed the contract owner address below for your viewing:

UNVERIFIED

The address above has authority over the ownable functions within the contract.

This allows the owner to call certain functions within the contract. Any compromise to the owner wallet may allow these privileges to be exploited.

Contract Code Audit – Owner Accessible Functions (DTGCStakingV4)

Function Name	Parameters	Visibility	Audit Notes
setTierConfig	uint8 tier, uint256 lockDays, uint256 aprBps, uint256 minAmount, bool isActive	external	onlyOwner modifier is detected. Owner can call this function if the contract is not renounced.
setEarlyWithdrawFee	uint256 feeBps	external	onlyOwner modifier is detected. Owner can call this function if the contract is not renounced.
setFeeRecipient	address _feeRecipient	external	onlyOwner modifier is detected. Owner can call this function if the contract is not renounced.
pause		external	onlyOwner modifier is detected. Owner can call this function if the contract is not renounced.
unpause		external	onlyOwner modifier is detected. Owner can call this function if the contract is not renounced.
recoverTokens	address token, uint256 amount	external	onlyOwner modifier is detected. Owner can call this function if the contract is not renounced.

Owner is able to modify LP configuration, pause, and unpause settings. Please ensure owner wallet is secure with proper access authority. Bad actors may cause issues.

The functions listed above can be called by the contract owner.

If contract ownership has been renounced there is no way for the above listed functions to be called.

Contract Code Audit – Mint Functions

This Contract Cannot Mint New Tokens.

We do understand that sometimes mint functions are essential to the functionality of the project.

A mint function was not found in the contract code.

Contract Transaction Fees

At the time of Audit the transaction fees (“tax”) listed below are the fees associated with trading. These fees are taken from every buy and sell transaction unless otherwise stated.

Transaction fees are not associated with these contracts.

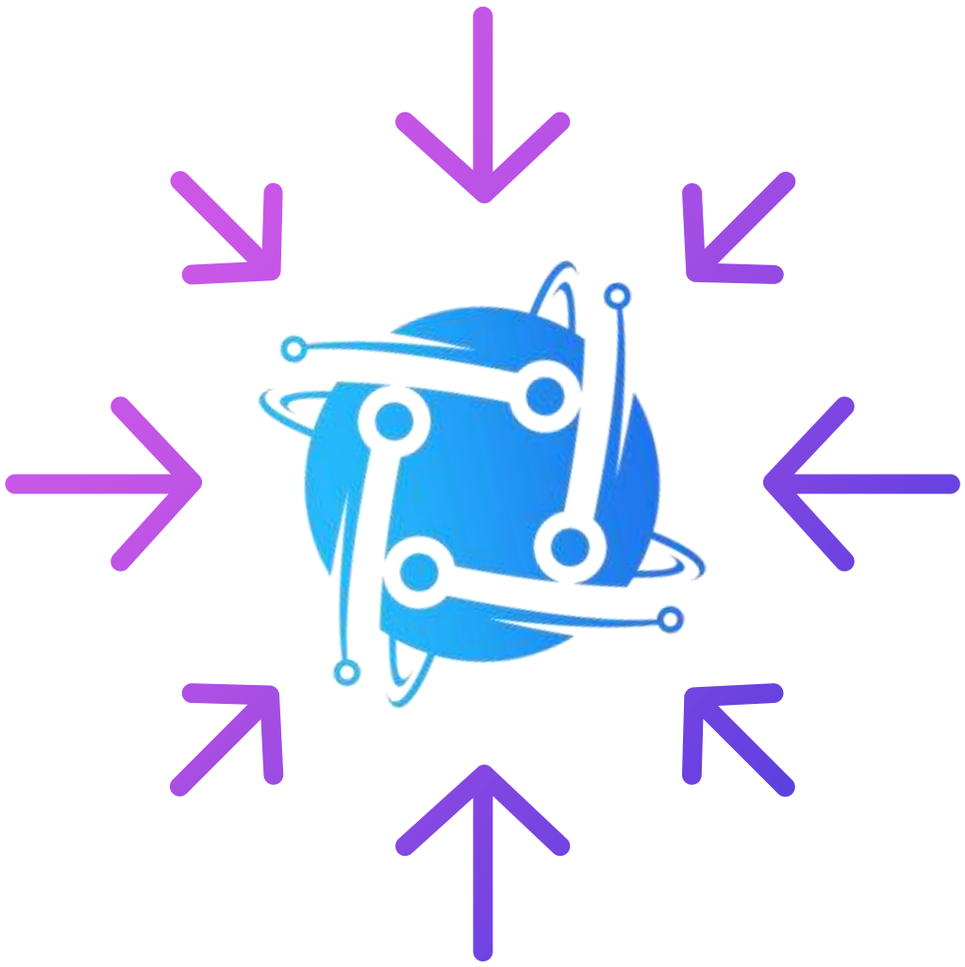
Website Part 1 – Overview
https://dtgc.io

Above images are actual snapshots of the current live website of the project.

Website was registered on 12/31/2025, registration expires 12/31/2026.

✗ This meets the 3 year minimum we like to see on new projects.

Website Part 2 – Checklist



- ✓ Mobile Friendly
- ✗ No JavaScript Errors
- ✓ Spell Check
- ✓ SSL Certificate

The website contained no severe JavaScript errors. No typos, or grammatical errors were present, and we found a valid SSL certificate allowing for access via https.

No additional issues were found on the website.

Website Part 3 – JavaScript Errors

The JavaScript errors present on the website are shown above. These are not critical errors however should be addressed.

Website Part 4 – Responsive HTML5 & CSS3

No issues were found on the Mobile Friendly check for the website. All elements loaded properly and browser resize was not an issue. The team has put a considerable amount of thought and effort into making sure their website looks great on all screens.

No severe JavaScript errors were found. No issues with loading elements, code, or stylesheets.

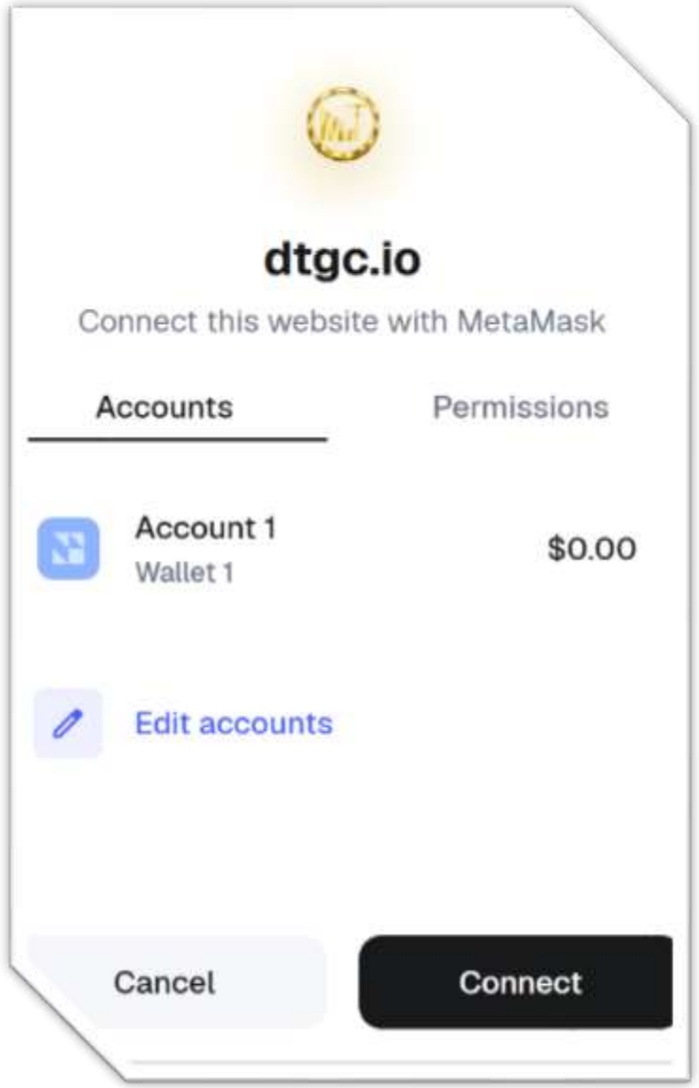
Website Part 5 – Wallet Connection

Dessert Finance was able to successfully verify wallet connection through a web3 interface.

Users are able to safely connect to the Dapp, transactions will always ask for approval before signing.

At the time of audit, we were not able to detect any malware.

✓Wallet Connection Verified



Website Part 4 (GWS) – General Web Security

SSL CERTIFICATE

A valid SSL certificate was found. Details are as follows:

Offered to: dtgc.io

Issued by: R12

Valid Until: March 2026

✓

CONTACT EMAIL

A valid contact email was found on the official website. Contact email is listed as shown below:

Contact

N/A

NOT CLEARLY LISTED

✗

SPAM / MALWARE / POPUPS

No malware found

No injected spam found

No internal server errors

No popups found

Domain is marked clean by Google, McAfee, Sucuri Labs, & ESET

✓

Social Media

We were able to locate a variety of Social Media networks for the project.

All links have been conveniently placed below.

Twitter

Telegram

✗ **At least 3 social media networks were found.**

Top Token Holders (Associated Token)

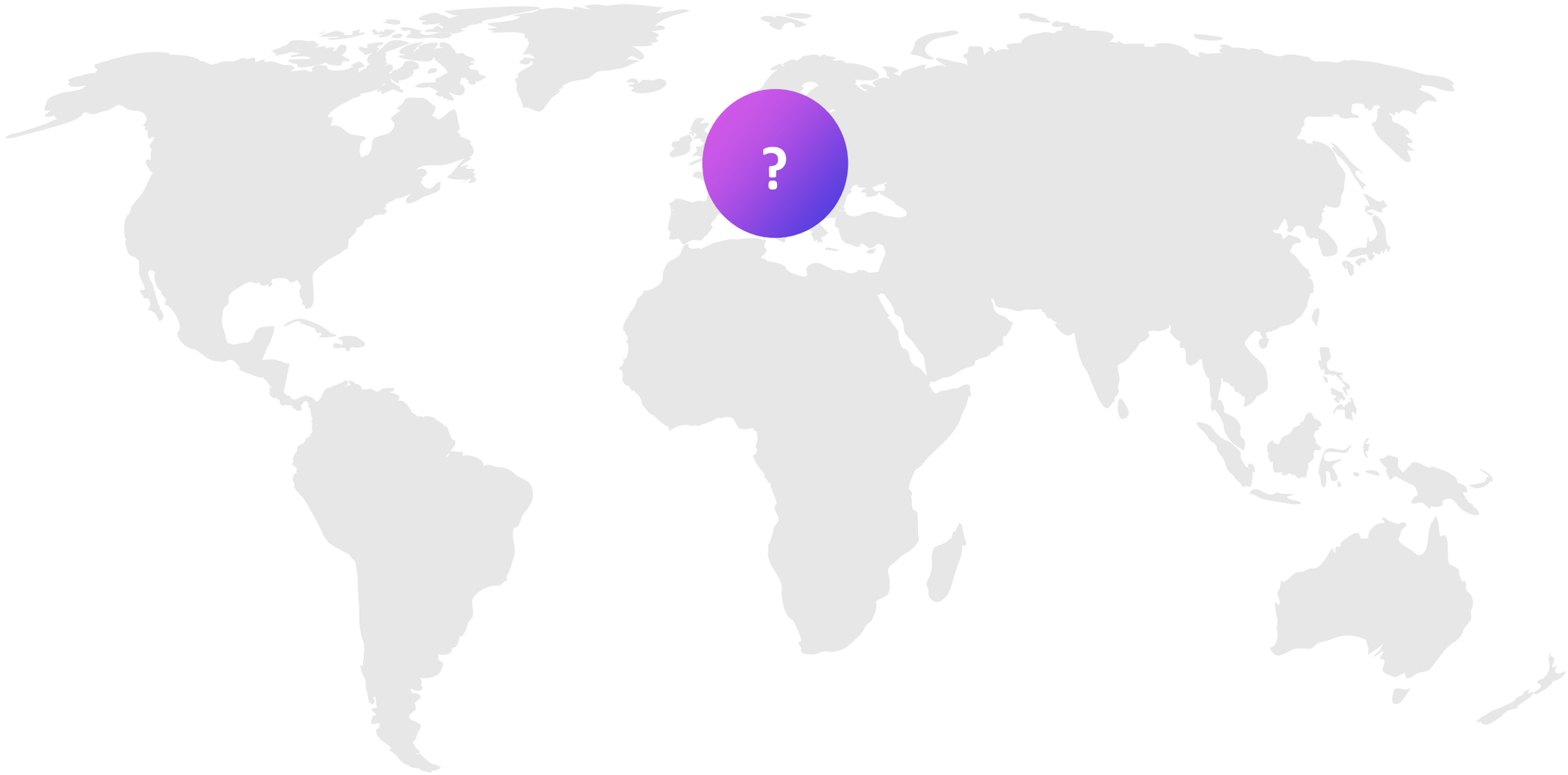
The top token holders at the time of the audit are shown below.

[Click here to view the most up-to-date list of holders]

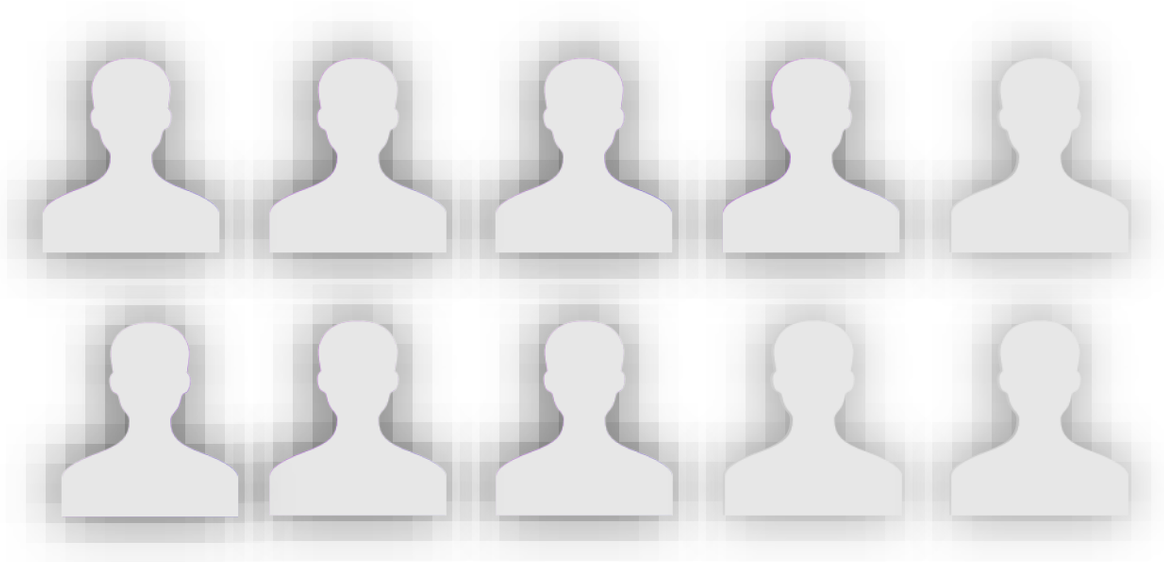
Holder	Quantity	Percentage
0x777d7f3aD24832975AEC259AB7D7b57Be4225AbF	309,314,925.433070098172458606	30.93%
0x22289ce7d7B962e804E9C8C6C57D2eD4Ffe0AbFC	274,791,216.184182818185851812	27.48%
0xC1CD5a70815E2874D2db038F398f2D8939d8E87C	73,277,305.778911901162363296	7.33%
0xEbC6802e6a2054FbF2Cb450aEc5E2916965b1718	71,077,685.066818528	7.11%
0x22f0DE89Ef26AE5c03CB43543dF5Bbd8cb8d0231	52,000,000	5.2%
0x0ba3d882f21b935412608d181501d59e99a8D0f9	50,760,933.49079392335561874	5.08%
0x0b0A8A0b7546FF180328aa155D2405882c7AC8C7	33,360,527.044094597577856578	3.34%
0x0000000000000000000000000000000000000369	22,241,890.561002991174427711	2.22%
0xD850Aa92968F6167E0240990984022E5bEf8FeBb	13,761,468	1.38%
0x97f0B5b011E8de04EbE30F70455E0cC2516f6E4C	13,761,468	1.38%
0x587e7D01C060A1037d0bEbc2eF655873B8C06C15	12,173,204.03033903444614089	1.22%
0x995A2D00C2b233F174AeA5b353492ebe83962b9b	10,011,339	1%
0xb1044960e004B581DF99cC54DF2F68B24A3ADC76	7,864,054	0.79%

Location Audit

We were unable to identify a primary location for the project at this time or a location has not been declared.



Team Overview



We are unable to find any information about the team on the website at this time. Projects may choose to stay anonymous for a myriad of reasons.

Roadmap

A roadmap was found on the official website, we have conveniently placed it on this page for your viewing.

Disclaimer

The opinions expressed in this document are for general informational purposes only and are **not intended to provide specific advice or recommendations for any individual or on any specific investment**. It is only intended to provide education and public knowledge regarding projects. This audit is only applied to the type of auditing specified in this report and the scope of given in the results. Other unknown security vulnerabilities are beyond responsibility. Dessert Finance only issues this report based on the attacks or vulnerabilities that already existed or occurred before the issuance of this report. For the emergence of new attacks or vulnerabilities that exist or occur in the future, Dessert Finance lacks the capability to judge its possible impact on the security status of smart contracts, thus taking no responsibility for them. The smart contract analysis and other contents of this report are based solely on the documents and materials that the contract provider has provided to Dessert Finance or was publicly available before the issuance of this report (issuance of report recorded via block number on cover page), if the documents and materials provided by the contract provider are missing, tampered, deleted, concealed or reflected in a situation that is inconsistent with the actual situation, or if the documents and materials provided are changed after the issuance of this report, Dessert Finance assumes no responsibility for the resulting loss or adverse effects. Due to the technical limitations of any organization, this report conducted by Dessert Finance still has the possibility that the entire risk cannot be completely detected. Dessert Finance disclaims any liability for the resulting losses.

Dessert Finance provides no guarantees against the sale of team tokens or the removal of liquidity by the project audited in this document. Even projects with a low risk score have been known to pull liquidity, sell all team tokens, or exit-scam. Please exercise caution when dealing with any cryptocurrency related platforms.

The final interpretation of this statement belongs to Dessert Finance.

Dessert Finance highly advises against using cryptocurrencies as speculative investments and they should be used solely for the utility they aim to provide.

Thank You

DESSERT FINANCE PROJECT AUDIT HAS BEEN COMPLETED FOR XXX 1 DSRT HAS BEEN SENT TO AUDITED PROJECT'S CONTRACT ADDRESS FOR VERIFICATION OF THIS AUDIT AT BLOCK NUMBER: **DRAFT**

THIS AUDIT IS ONLY VALID IF VIEWED ON HTTPS://WWW.DESSERTSWAP.FINANCE

www.dessertswap.finance
https://t.me/dessertswap