

DESSERT
FINANCE



**Beast Masters Token
(MASTER)
BEP-20 Audit**

Performed at block **12923786**

PERFORMED BY DESSERT FINANCE
FOR CONTRACT ADDRESS: **0x23e2efA091112B85aF95Cf9195E505FD58a3DC8F**

INITIAL DISCLAIMER

Dessert Finance provides due-diligence project audits for various BSC 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 a project audit can be seen as a sign of confidence and is generally the first sign of trust for a project, but in no way guarantees that a team will not remove *all* liquidity (“Rug Pull”), sell off tokens, 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.

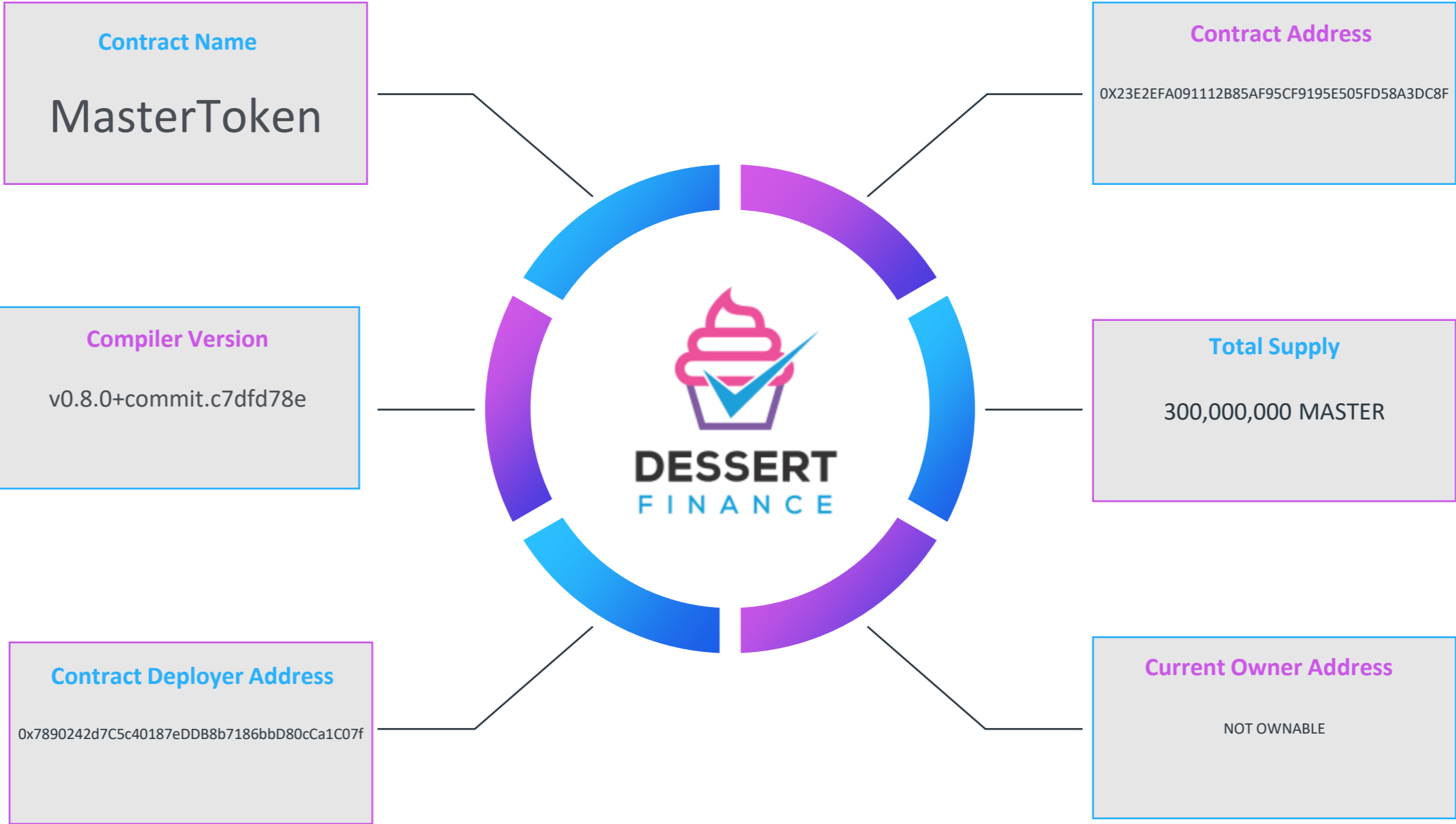
Dessert Finance in no way takes responsibility for any losses, nor does Dessert Finance encourage any speculative investments. The information provided in this audit is for information purposes only and should not be considered investment advice.

Table of Contents



1. Contract Code Audit – Token Overview
2. BEP-20 Contract Code Audit – Overview
3. BEP-20 Contract Code Audit – Vulnerabilities Checked
4. Contract Code Audit – Contract Ownership
5. Liquidity Ownership – Locked / Unlocked
6. Contract Code Audit – Mint Functions
7. Contract Transaction Fees
8. Website Overview
9. Social Media
10. Top Token Holders/Wallets
11. Location Audit
12. Review of Team
13. Roadmap
14. Disclaimers

Contract Code Audit – Token Overview



BEP-20 Contract Code Audit – Overview

Dessert Finance was commissioned to perform an audit on Beast Masters Token (MASTER)

```
// SPDX-License-Identifier: MIT
pragma solidity ^0.8.0;
import "../helpers/beastMastersToken/ERC20.sol";

contract MasterToken is ERC20 {
    uint256 private _totalTokens;

    constructor(address _playToEarnAddress, address _stakingAddress, address _teamMarketingFreelancers, address _presaleLiquidity) {
        _totalTokens = 300000000 * 10 ** uint256(decimals());

        _mint(_playToEarnAddress, uint256(_totalTokens * 75 / 100)); // 75% Play to Earn (225,000,000)
        _mint(_stakingAddress, uint256(_totalTokens * 10 / 100)); // 10% Staking (30,000,000)
        _mint(_teamMarketingFreelancers, uint256(_totalTokens * 8 / 100)); // 8% Team & Marketing & Freelancers (24,000,000)
        _mint(_presaleLiquidityAddress, uint256(_totalTokens * 7 / 100)); // 7% Presale and Liquidity (21,000,000)
    }

    function burn(uint256 _amount) public {
        _burn(msg.sender(), _amount);
    }

    function getBurnedTokensAmount() public view returns (uint256 amount) {
        return _totalTokens - totalSupply();
    }
}

// SPDX-License-Identifier: MIT
pragma solidity ^0.8.0;

import "../interfaces/IERC20.sol";
import "../interfaces/IERC20Metadata.sol";
import "../Context.sol";

/**
 * @dev Implementation of the (IERC20) interface.
 *
 * This implementation is agnostic to the way tokens are created. This means
 * that a supply mechanism has to be added in a derived contract using (_mint).
 * For a generic mechanism see (ERC20PresetMinterPauser).
 *
 * TIP: For a detailed writeup see our guide
 * https://forum.zepplin.solutions/t/how-to-implement-erc20-supply-mechanisms/226[How
 * to implement supply mechanisms].
 *
 * We have followed general OpenZeppelin Contracts guidelines: functions revert
 * instead returning (false) on failure. This behavior is nonetheless
 * conventional and does not conflict with the expectations of ERC20
 * applications.
 *
 * Additionally, an (Approval) event is emitted on calls to (transferFrom).
 * This allows applications to reconstruct the allowance for all accounts just
 * by listening to said events. Other implementations of the EIP may not emit
 * these events, as it isn't required by the specification.
 *
 * Finally, the non-standard (decreaseAllowance) and (increaseAllowance)
 * functions have been added to mitigate the well-known issues around setting
 * allowances. See (IERC20-approve).
 */
contract ERC20 is Context, IERC20, IERC20Metadata {
    mapping(address => uint256) private _balances;
}
```

Contract Address

0x23e2efA091112B85aF95Cf9195E505FD58a3DC8F

Token Tracker

Beast Masters Token (MASTER)

Contract Creator

0x7890242d7c5c40187eddb8b7186bbd80cca1c07f

Source Code

Contract Source Code Verified

Contract Name

MasterToken

Other Settings

default evmVersion

Compiler Version

v0.8.0+commit.c7dfd78e

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 BSCScan.

BEP-20 Contract Code Audit – Vulnerabilities Checked

Vulnerability Tested	AI Scan	Human Review	Result
Compiler Errors	Complete	Complete	✓ Low / No Risk
Outdated Compiler Version	Complete	Complete	✓ Low / No Risk
Integer Overflow	Complete	Complete	✓ Low / No Risk
Integer Underflow	Complete	Complete	✓ Low / No Risk
Correct Token Standards Implementation	Complete	Complete	✓ Low / No Risk
Timestamp Dependency for Crucial Functions	Complete	Complete	✓ Low / No Risk
Exposed _Transfer Function	Complete	Complete	✓ Low / No Risk
Transaction-Ordering Dependency	Complete	Complete	✓ Low / No Risk
Unchecked Call Return Variable	Complete	Complete	✓ Low / No Risk
Use of Deprecated Functions	Complete	Complete	✓ Low / No Risk
Unprotected SELFDESTRUCT Instruction	Complete	Complete	✓ Low / No Risk
State Variable Default Visibility	Complete	Complete	✓ Low / No Risk
Deployer Can Access User Funds	Complete	Complete	✓ Low / No Risk

The contract code is verified on BSCScan.

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

Liquidity Ownership – Locked / Unlocked

No locked liquidity information has been found.



This page will contain links to locked liquidity for the project if we are able to locate that information. Locked liquidity information was not found on the project's website.

Contract Code Audit – Mint Functions

This Contract Cannot Mint New MASTER 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.

Website Part 1 – Overview

www.beastmasters.io



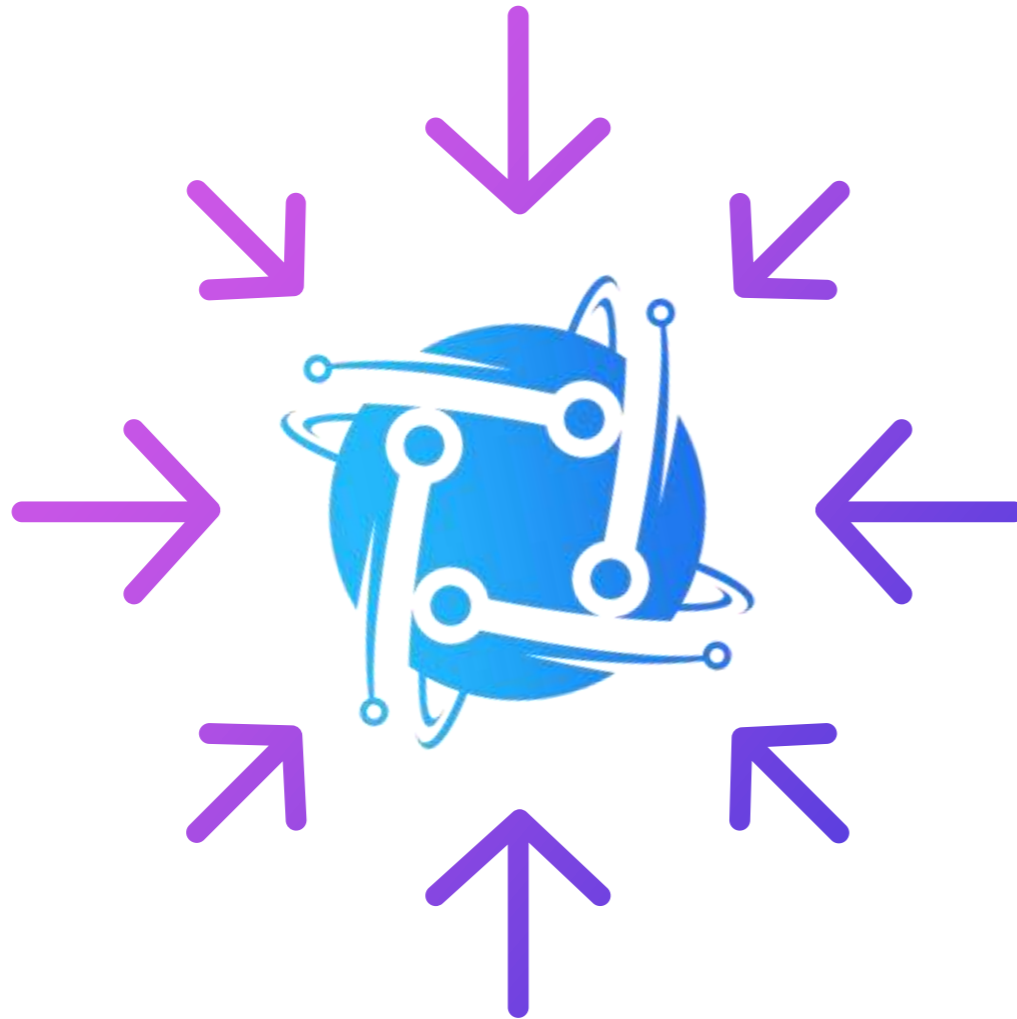
Above images are actual snapshots of the current live website of the project.

Website was registered on 10/12/2021, registration expires 10/12/2022.

X 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

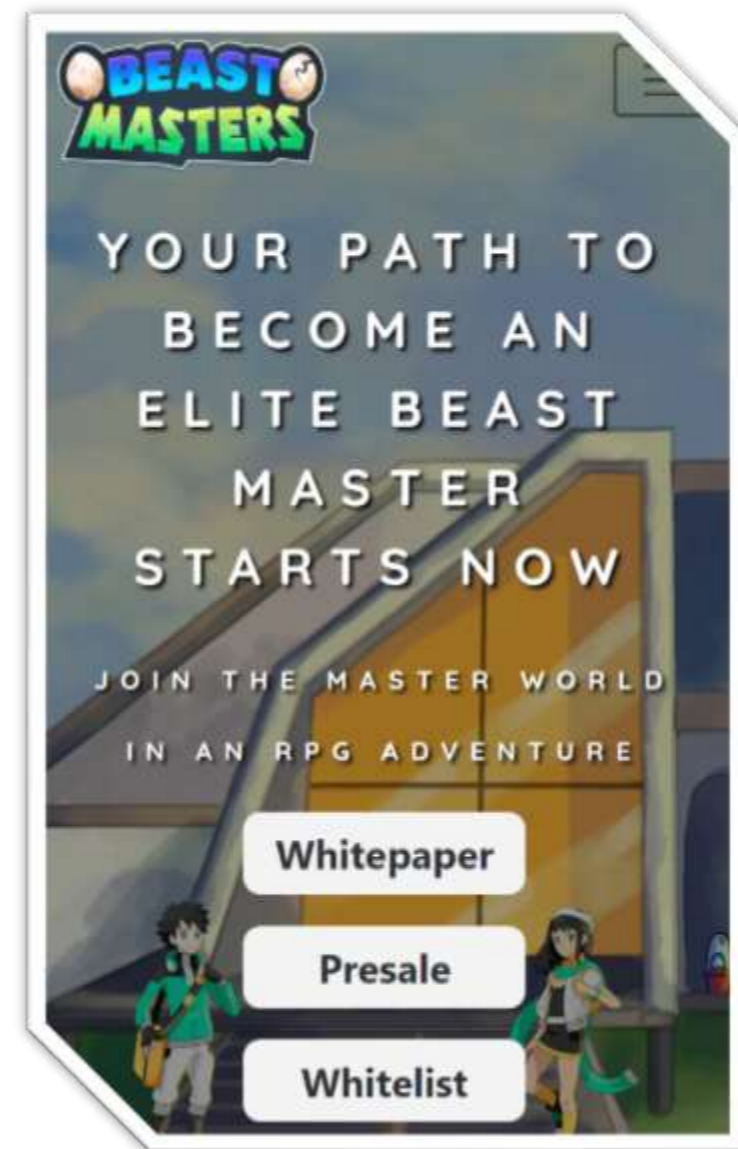
The website contained no JavaScript errors. No typos, or grammatical errors were present, and we found a valid SSL certificate allowing for access via https.

No additional issues were found on the website.

Website Part 3 – 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 4 (GWS) – General Web Security



SSL CERTIFICATE

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

Offered to: beastmasters.io

Issued by: R3

Valid Until: 01/22/2022



CONTACT EMAIL

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

[Contact](mailto:contact@beastmasters.io)

contact@beastmasters.io



SPAM / MALWARE / POPUPS

No malware found

No injected spam found

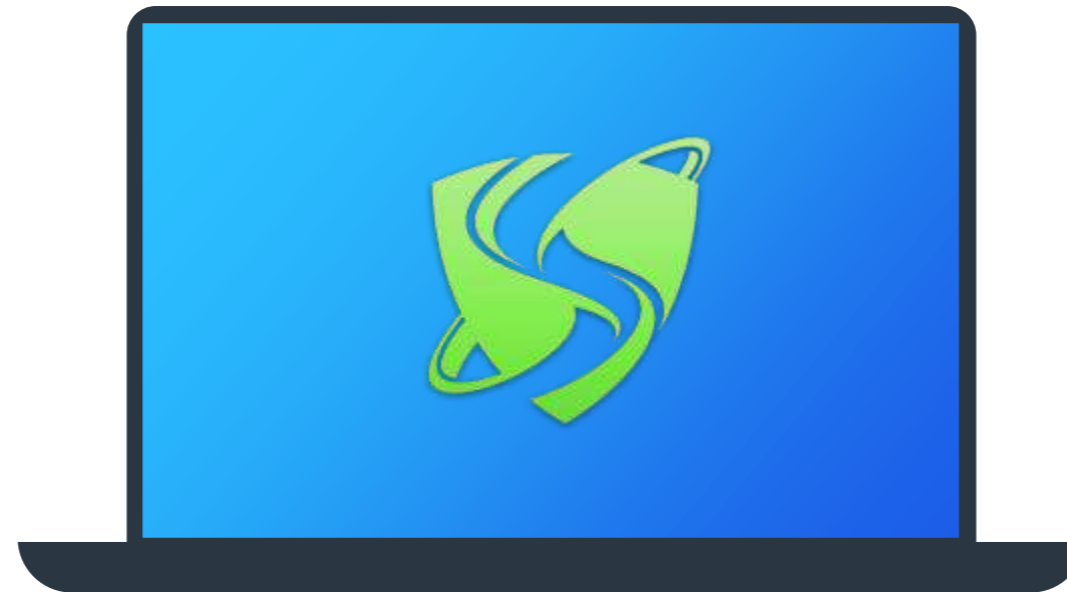
No internal server errors

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](#)



[Discord](#)

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

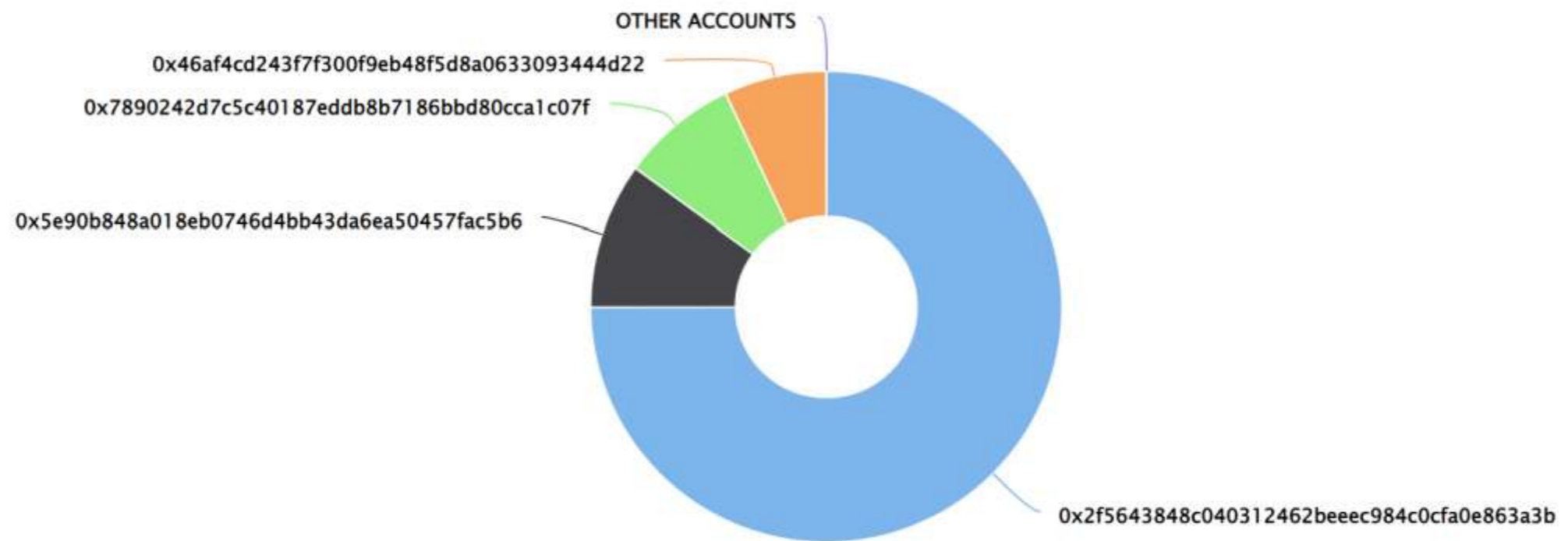
Top Token Holders

The entire supply was in a few wallets at the time of audit. We expect this to change as the project goes through initial distribution phases. Please use the link below to view the most up-to-date holder information.

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

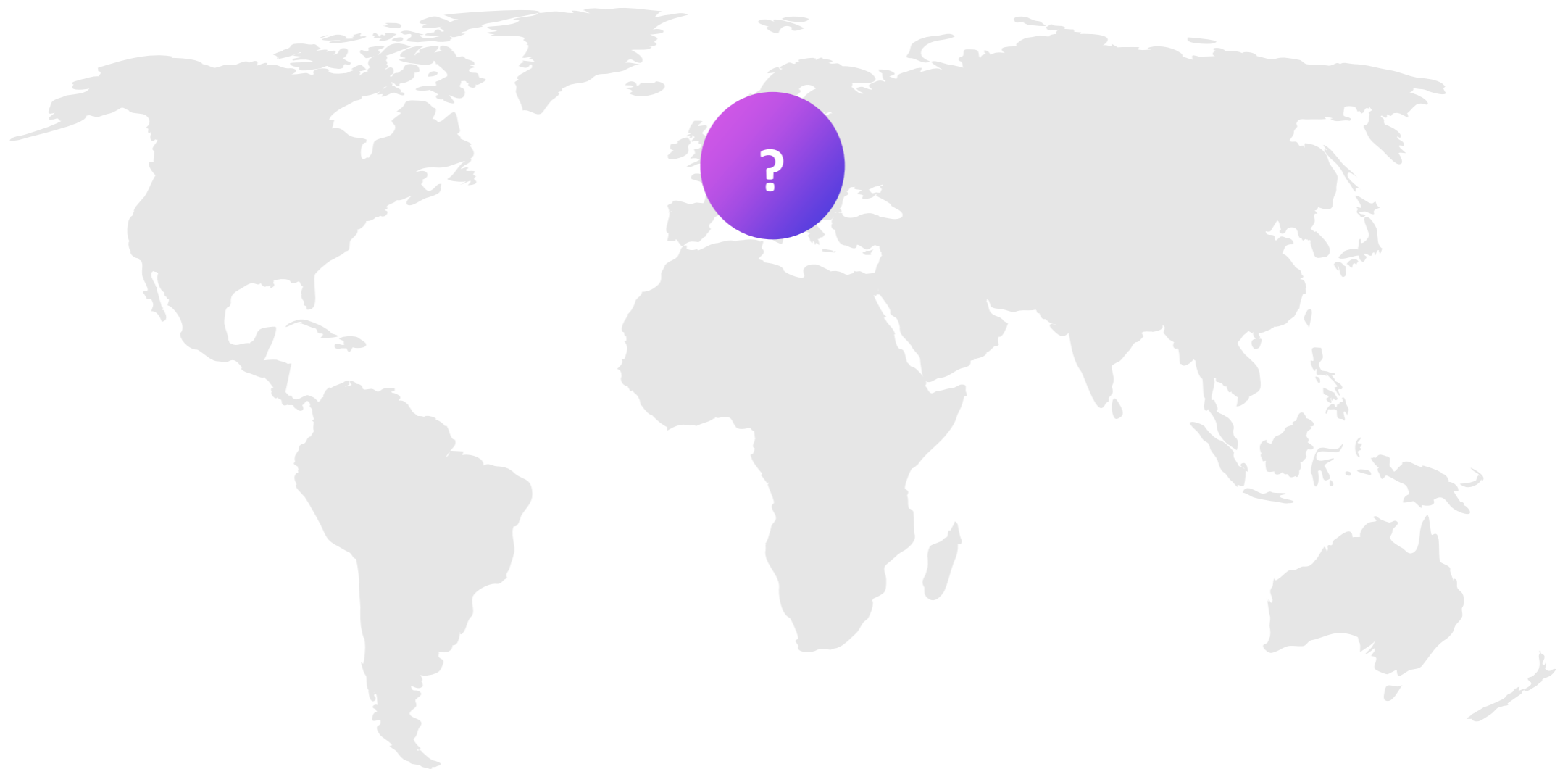
Beast Masters Token Top 100 Token Holders

Source: BscScan.com

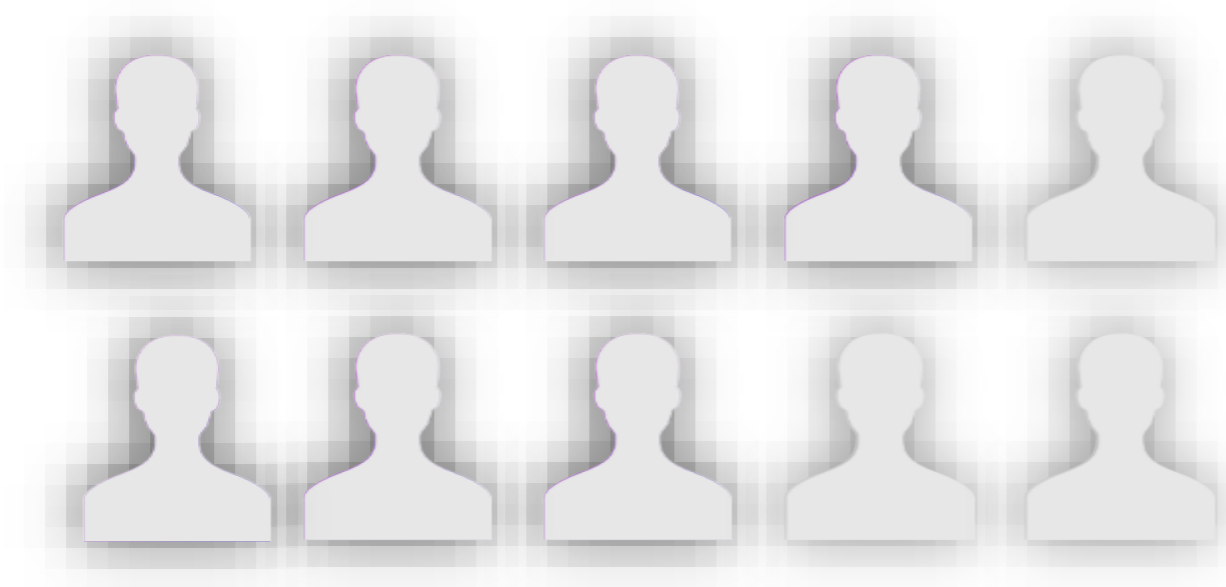


Location Audit

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 BSC 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 BEAST MASTERS TOKEN (MASTER) 1 DSRT HAS BEEN SENT TO AUDITED PROJECT'S CONTRACT ADDRESS FOR VERIFICATION OF THIS AUDIT AT BLOCK NUMBER: **12923786**

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