

DESSERT
FINANCE



MetaSPAY (MetaSPAY)

BEP-20 Audit

Performed at block 13011249

PERFORMED BY DESSERT FINANCE
FOR CONTRACT ADDRESS: 0x8f82EB1171799D5Ac971f6eE88191506aC4944b6

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.

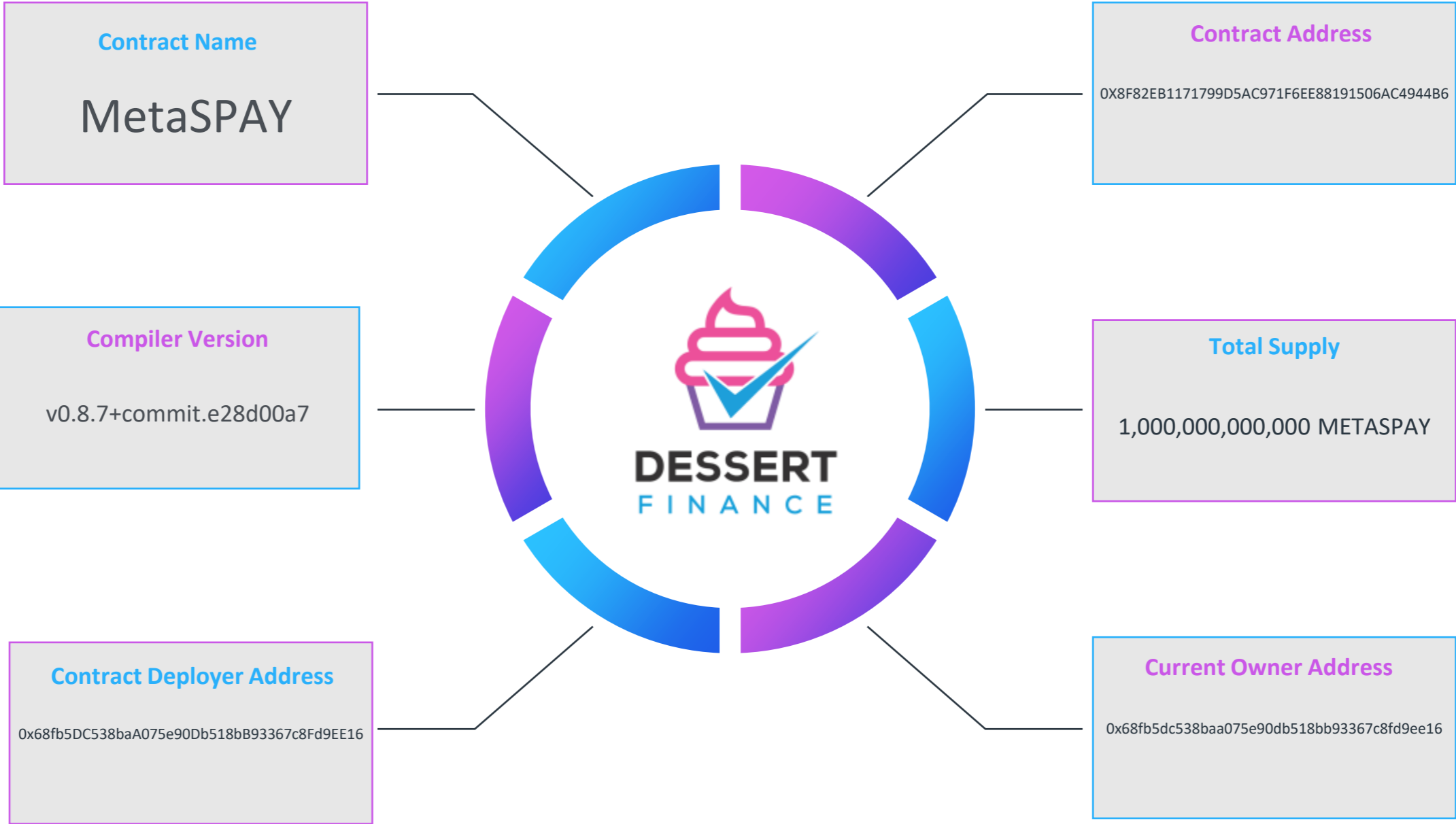
Dessert Finance in no way takes responsibility for any losses, nor does Dessert Finance encourage any speculative investments. The information provided in this audit is for information purposes only and should not be considered investment advice.

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 MetaSPAY (MetaSPAY)

```
Submitted for verification at BscScan.com on 2021-11-30
// SPDX-License-Identifier: MIT
pragma solidity ^0.8.0;
...
// Standard SafeMath, stripped down to just add/sub/mul/div
library SafeMath {
    function add(uint256 a, uint256 b) internal pure returns (uint256) {
        uint256 c = a + b;
        require(c >= a, "SafeMath: addition overflow");
        return c;
    }
    function sub(uint256 a, uint256 b) internal pure returns (uint256) {
        return sub(a, b, "SafeMath: subtraction overflow");
    }
    function mul(uint256 a, uint256 b, string memory errorMessage) internal pure returns (uint256) {
        require(b >= 0, errorMessage);
        uint256 c = a * b;
        return c;
    }
    function div(uint256 a, uint256 b) internal pure returns (uint256) {
        if (b == 0) {
            return 0;
        }
        uint256 c = a / b;
        require(c * b <= a, "SafeMath: multiplication overflow");
        return c;
    }
    function div(uint256 a, uint256 b) internal pure returns (uint256) {
        return div(a, b, "SafeMath: division by zero");
    }
    function div(uint256 a, uint256 b, string memory errorMessage) internal pure returns (uint256) {
        // Solidity only automatically asserts when dividing by 0
        require(b > 0, errorMessage);
        uint256 c = a / b;
        // assert(a == b * c + a % b); // There is no case in which this doesn't hold
        return c;
    }
}
abstract contract Context {
    function _msgSender() internal view virtual returns (address) {
        return msg.sender;
    }
    function _msgData() internal view virtual returns (bytes calldata) {
        // !!! // solidity static analyzer warning without generating bytecode - see https://github.com/ethers-io/contracts/issues/2691
        return msg.data;
    }
}
contract Ownable is Context {
    address private _owner;
    event OwnershipTransferred(address indexed previousOwner, address indexed newOwner);
    // ...
    // @dev Initialize the contract setting the msg.sender as the initial owner.
    // ...
}
```

Contract Address

0x8f82EB1171799D5Ac971f6eE88191506aC4944b6

TokenTracker

MetaSPAY (MetaSPAY)

Contract Creator

0x68fb5dc538baa075e90db518bb93367c8fd9ee16

Source Code

Contract Source Code Verified

Contract Name

MetaSPAY

Other Settings

default evmVersion, MIT

Compiler Version

v0.8.7+commit.e28d00a7

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.](#)

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 (x28) | Complete | Complete | ✓ Low 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.

Contract Code Audit – Contract Ownership

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:

[0x68fb5dc538baa075e90db518bb93367c8fd9ee16](https://etherscan.io/address/0x68fb5dc538baa075e90db518bb93367c8fd9ee16)

Liquidity Ownership – Locked / Unlocked

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.

DxSale:

<https://dxsale.app/app/v3/dxlockview?id=0&add=0x68fb5DC538baA075e90Db518bB93367c8Fd9EE16&type=lplock&chain=BSC>

Contract Code Audit – Mint Functions

This Contract Cannot Mint New MetaSPAY 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.

18% Buy and Sell Tax

To provide every holder with the best value

10% SPAY Auto Reward

10%

8% Marketing and Dev team expenses

10%



3% Max Wallet

This is to protect you and all other investors from falling victim to whales dumping and manipulating the price.

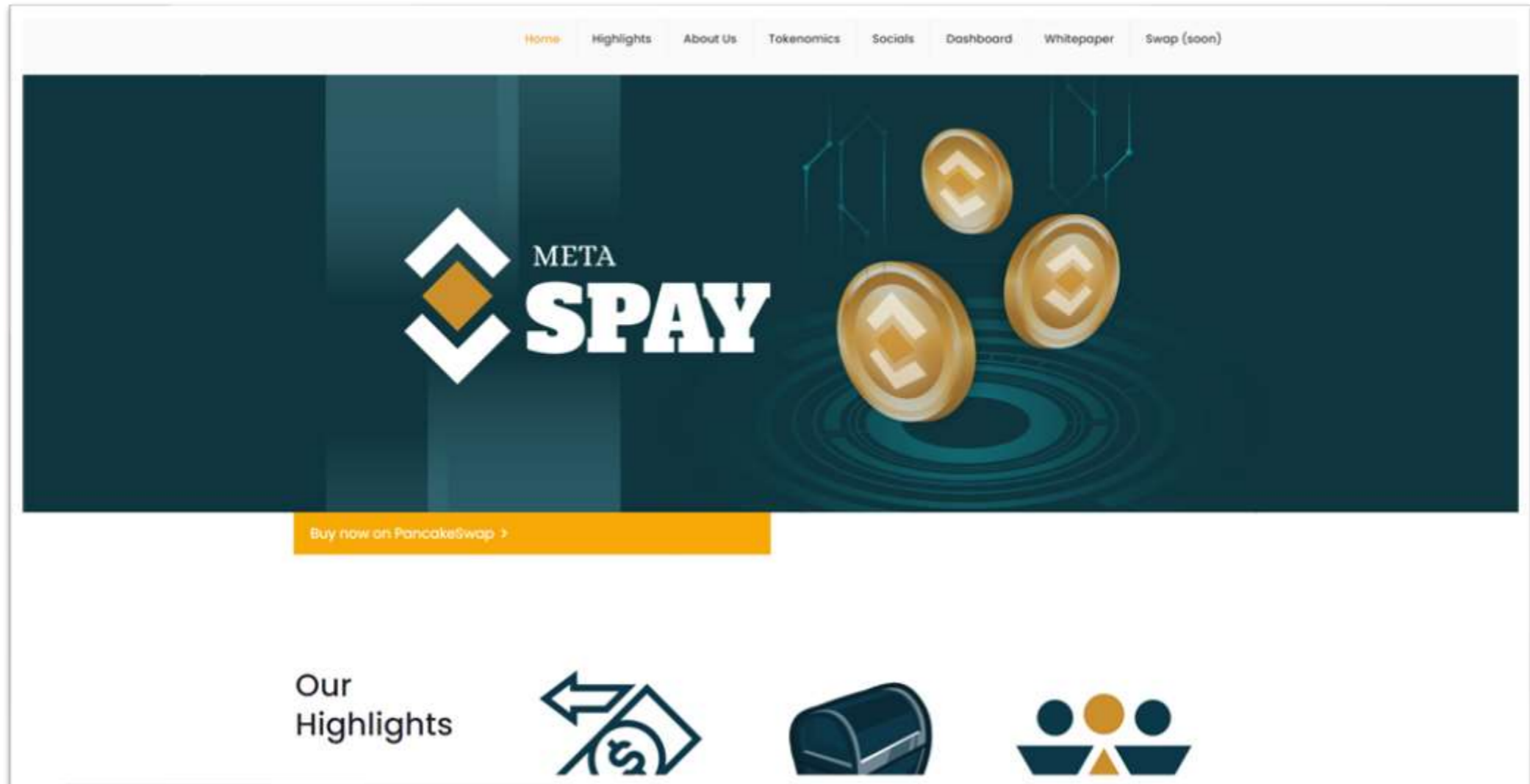


Specifics

Total Supply: 1 Trillion
Initial Liquidity: 3BNB

Website Part 1 – Overview

www.metaspay.finance



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

Website was registered on 11/10/2021, registration expires 11/10/2022.

X This does not meet 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 some JavaScript errors. No typos, or grammatical errors were present, and we found a valid SSL certificate allowing for access via https.

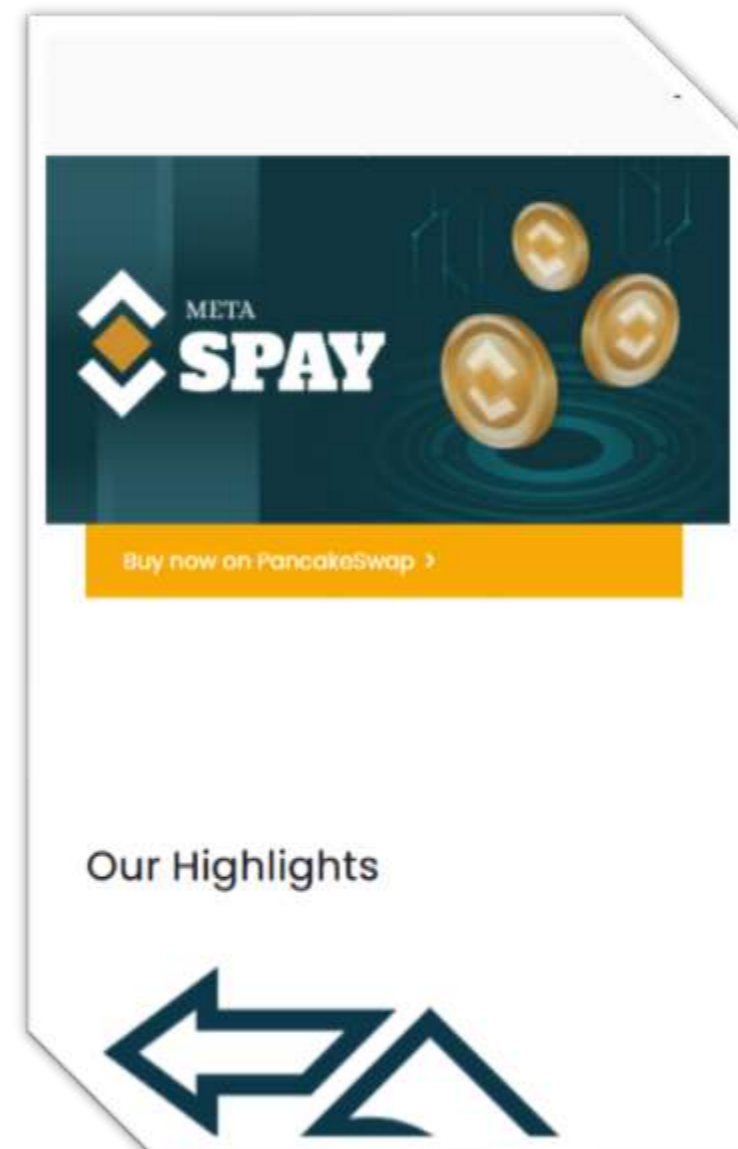
No additional issues were found on the website.

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.

✘ ▶Mixed Content: The page at '<https://metaspay.finance/>' was loaded (index):21 over HTTPS, but requested an insecure script '<http://metaspay.finance/wp-includes/js/wp-emoji-release.min.js?ver=5.8.2>'. This request has been blocked; the content must be served over HTTPS.



Website Part 4 (GWS) – General Web Security



SSL CERTIFICATE

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

Offered to: metaspay.finance

Issued by: Cloudflare Inc

Valid Until: 11/09/2022



CONTACT EMAIL

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

Contact

N/A



SPAM / MALWARE / POPUPS

No malware found

No injected spam found

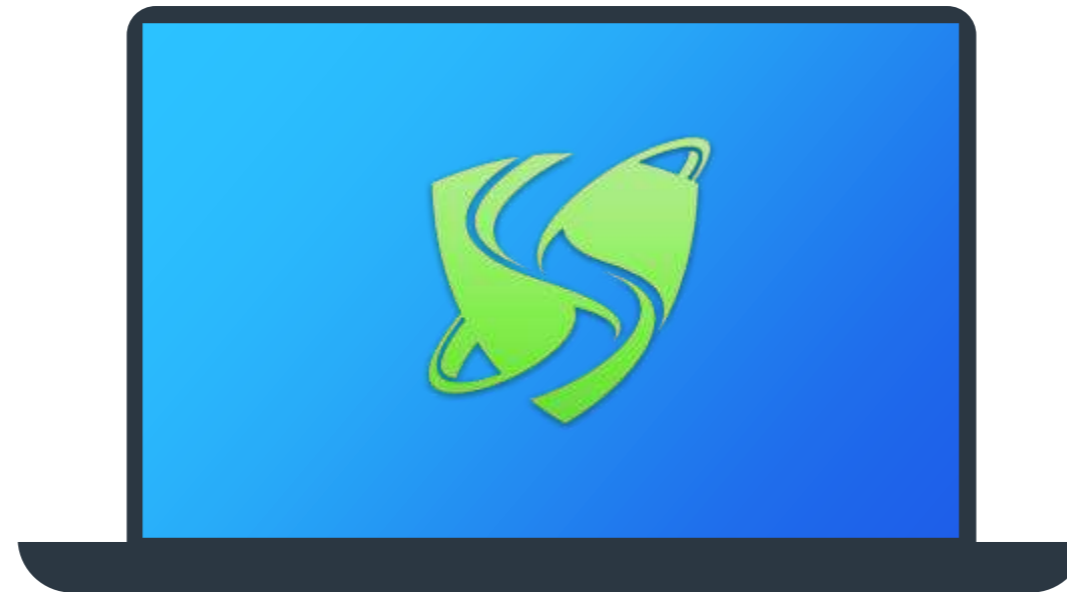
No internal server errors

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](#)



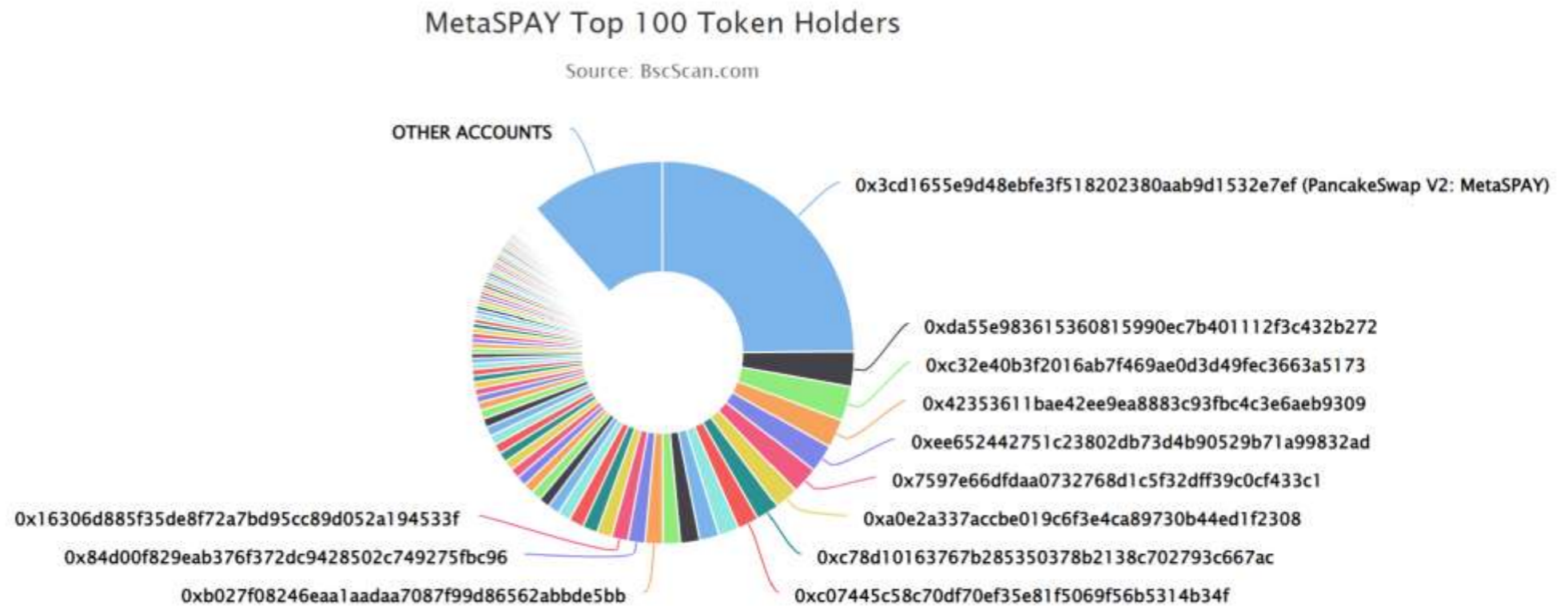
[Telegram](#)

X At least 3 social media networks were found.

Top Token Holders

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

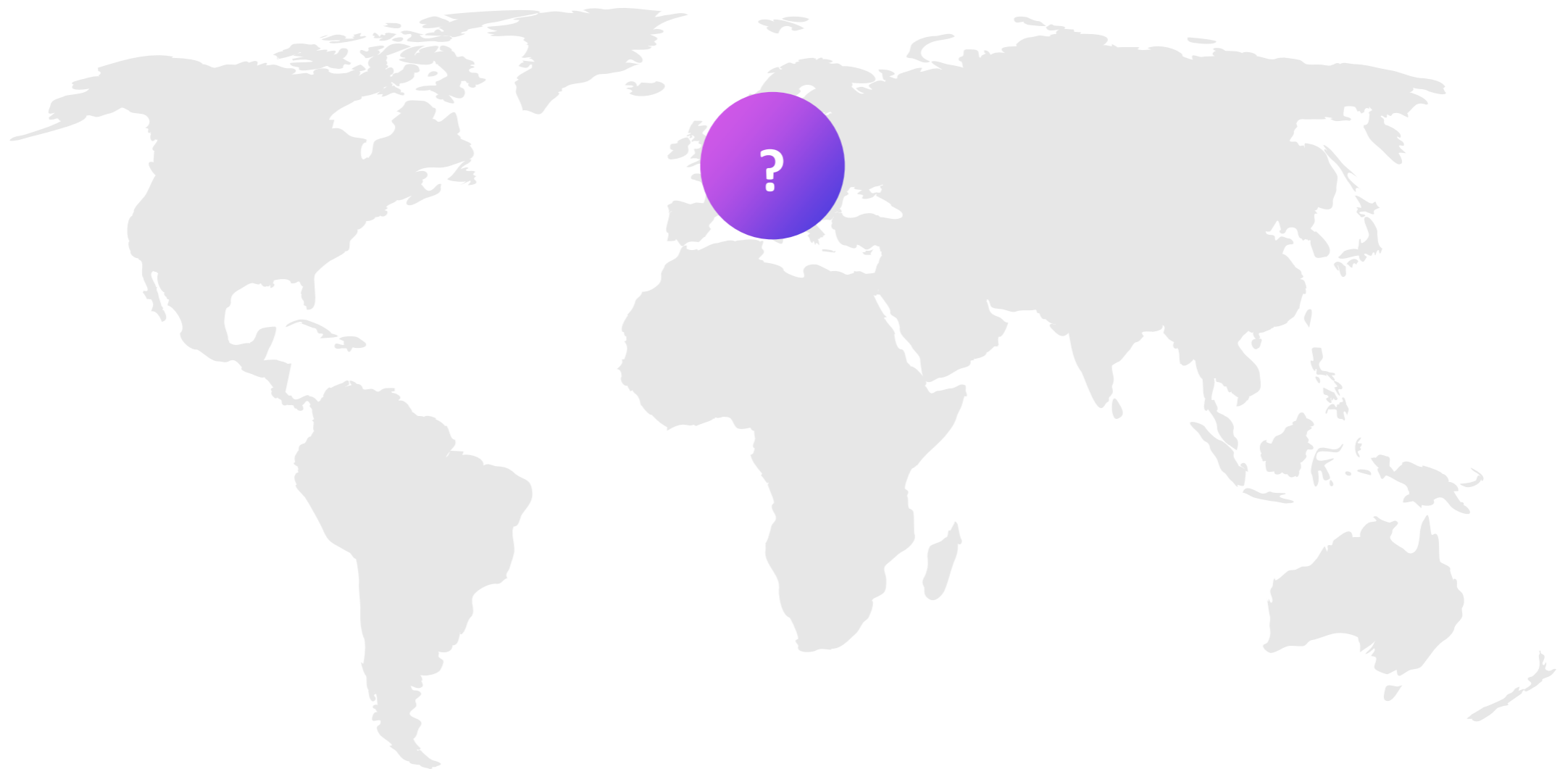
[Click here to view the most up-to-date list of holders](#)



| Rank | Address | Quantity (Token) | Percentage |
|------|--|---------------------------|------------|
| 1 | PancakeSwap V2: MetaSPAY | 248,902,094,282.632107251 | 24.8902% |
| 2 | 0xda55e983615360815990ec7b401112f3c432b272 | 29,642,536,287 | 2.9643% |
| 3 | 0xc32e40b3f2016ab7f469ae0d3d49fec3663a5173 | 29,097,628,010.451 | 2.9098% |
| 4 | 0x42353611bae42ee9ea8883c93fbc4c3e6aeb9309 | 24,114,635,181 | 2.4115% |
| 5 | 0xee652442751c23802db73d4b90529b71a99832ad | 23,438,506,710.079341234 | 2.3439% |

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.

Phase 1:

- Drawing board, connecting dots of our project Vision
- Having test and trial run on the token's contract
- Finalization of suggested tokenomics by tokenomics specialists
- Launching official website, Telegram group & Twitter account
- Pre-fairlaunch preparation
- Pre-fairlaunch marketing
- Smart Contract coding & debugging

Phase 2:

- Deploying & fair-launching on PancakeSwap
- Post-launch marketing implementations
- Official launch of the MetaSPAY Swap DEX
- Engaging Youtubers and Twitter influencers
- Collaboration with Telegram owners
- 1,000 holders milestone
- Market Capitalization of \$250,000 & above

Phase 3:

- CoinGecko listing
- CoinMarketCap listing
- 2,500 holders & above
- Market Capitalization of \$1 million & above
- Exploring the possibility of adding more utilities into MetaSPAY's ecosystem: NFT marketplace, NFT gaming, NFT drops, Metaverse

Phase 4:

- Market Capitalization \$10 million & above
- Official launch of other added utilities one by one
- Expanding the ecosystem of MetaSPAY

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 METASPAY (METASPAY) 1 DSRT HAS BEEN SENT TO AUDITED PROJECT'S CONTRACT ADDRESS FOR VERIFICATION OF THIS AUDIT AT BLOCK NUMBER: **13011249**

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