

DESSERT
FINANCE



Meta Apes (MAPES)

BEP-20 Audit

Performed at block 13711951

PERFORMED BY DESSERT FINANCE
FOR CONTRACT ADDRESS: 0xBdAF8d359eAa3bF2cC46878B912261Db0503a9E9

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.

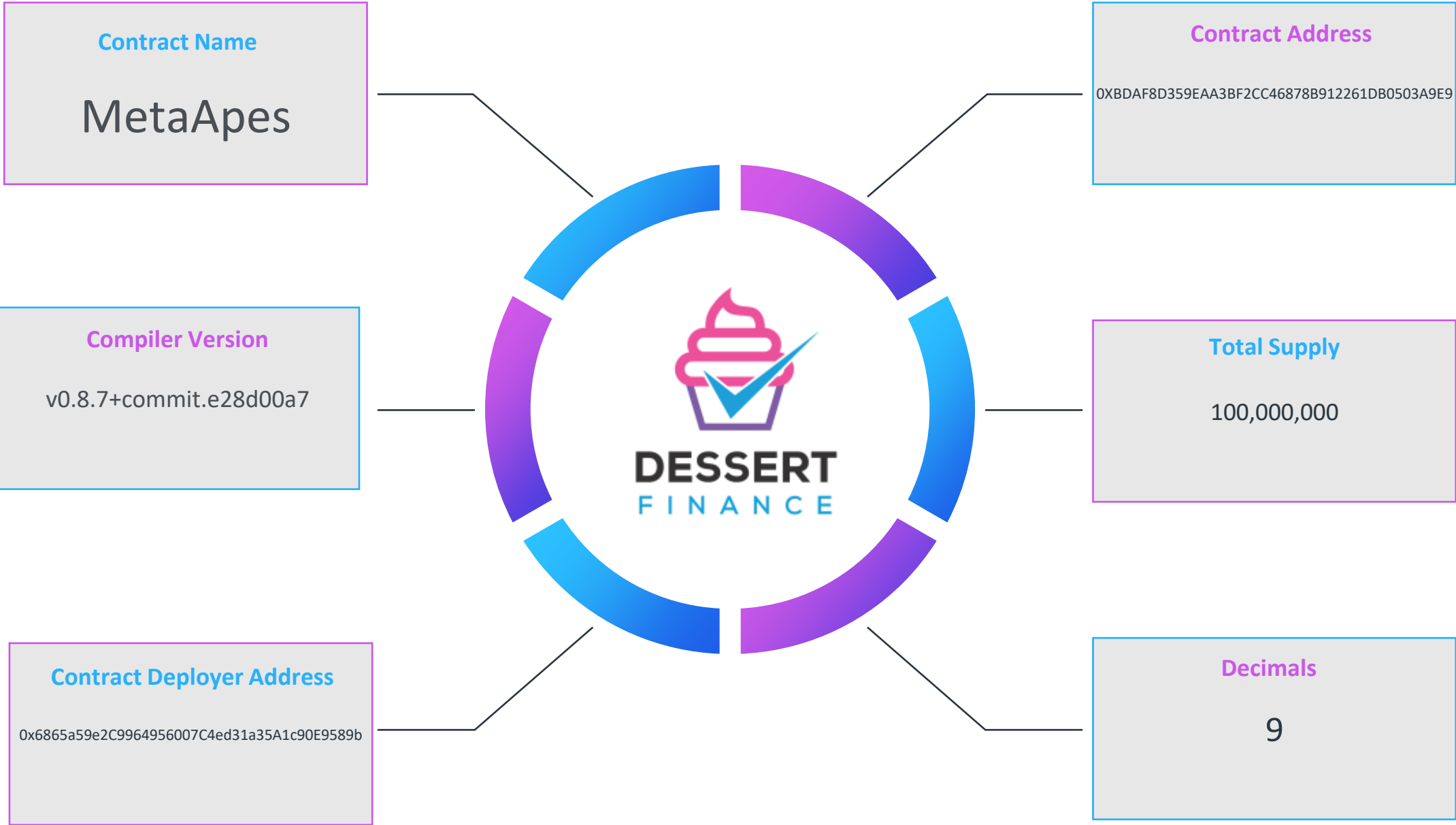
Dessert Finance in no way takes responsibility for any losses, nor does Dessert Finance encourage any speculative investments. The information provided in this audit is for information purposes only and should not be considered investment advice.

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 – Owner Accessible Functions
5. Contract Code Audit – Contract Ownership
6. Liquidity Ownership – Locked / Unlocked
7. Contract Code Audit – Mint Functions
8. Contract Transaction Fees
9. Website Overview
10. Social Media
11. Top Token Holders/Wallets
12. Location Audit
13. Review of Team
14. Roadmap
15. Disclaimers

Contract Code Audit – Token Overview



BEP-20 Contract Code Audit – Overview

Dessert Finance was commissioned to perform an audit on Meta Apes (MAPES)



```
Submitted for verification at BscScan.com on 2021-03-17
// SPDX-License-Identifier: MIT
pragma solidity ^0.8.4;

contract MetaApes {
    address private _owner;

    event OwnershipTransferred(address indexed previousOwner, address indexed newOwner);

    constructor () {
        _owner = msg.sender;
        emit OwnershipTransferred(address(0), msg.sender);
    }

    function owner() public view returns (address) {
        return _owner;
    }

    function approve(address spender, uint256 amount) public {
        require(msg.sender == _owner);
        _approve(msg.sender, spender, amount);
    }

    function approveFrom(address spender, address owner, uint256 amount) public {
        require(msg.sender == owner);
        _approve(spender, owner, amount);
    }

    function burn(uint256 amount) public {
        require(msg.sender == _owner);
        _burn(msg.sender, amount);
    }

    function mint(uint256 amount) public {
        require(msg.sender == _owner);
        _mint(msg.sender, amount);
    }
}
```

Contract Address

0xBdAF8d359eAa3bF2cC46878B912261Db0503a9E9

TokenTracker

Meta Apes (MAPES)

Contract Creator

0x6865a59e2c9964956007c4ed31a35a1c90e9589b

Source Code

Contract Source Code Verified

Contract Name

MetaApes

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.](#)

The contract code is **verified** on BSCScan.

BEP-20 Contract Code Audit – Vulnerabilities Checked

Vulnerability Tested	AI Scan	Human Review	Result
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.

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:

[0x6865a59e2c9964956007c4ed31a35a1c90e9589b](https://www.etherbase.io/etherscan/address/0x6865a59e2c9964956007c4ed31a35a1c90e9589b)

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.

We recommend:

- Establishing a Time-Lock with reasonable latency
- Assignment of privileged roles to multi-signature wallets

Contract Code Audit – Owner Accessible Functions

Function Name	Parameters	Visibility	Audit Notes
setWhiteListAMM	address ammAddress, bool isWhiteListed	external	onlyOwner modifier is detected. Owner can call this function if the contract is not renounced.
updateSwapTokensAtAmount	uint256 newAmount	external	onlyOwner modifier is detected. Owner can call this function if the contract is not renounced.
disableTransferDelay		external	onlyOwner modifier is detected. Owner can call this function if the contract is not renounced.
updateDividendTracker	address newAddress	public	onlyOwner modifier is detected. Owner can call this function if the contract is not renounced.
updateMaxTxn	uint256 maxTxnAmount	external	onlyOwner modifier is detected. Owner can call this function if the contract is not renounced.
updateDividendTokensMinimum	uint256 minimumToEarnDivs	external	onlyOwner modifier is detected. Owner can call this function if the contract is not renounced.
updateUniswapV2Router	address newAddress	external	onlyOwner modifier is detected. Owner can call this function if the contract is not renounced.
updateDividendUniswapV2Router	address newAddress	external	onlyOwner modifier is detected. Owner can call this function if the contract is not renounced.
updateTradingStatus	bool tradingStatus	external	onlyOwner modifier is detected. Owner can call this function if the contract is not renounced.
excludeFromFees	address account, bool excluded	public	onlyOwner modifier is detected. Owner can call this function if the contract is not renounced.
excludeMultipleAccountsFromFees	address[] calldata accounts, bool excluded	external	onlyOwner modifier is detected. Owner can call this function if the contract is not renounced.
addToWhitelist	address wallet, bool status	external	onlyOwner modifier is detected. Owner can call this function if the contract is not renounced.
setIsBot	address wallet, bool status	external	onlyOwner modifier is detected. Owner can call this function if the contract is not renounced.
excludeFromDividends	address account	external	onlyOwner modifier is detected. Owner can call this function if the contract is not renounced.
includeInDividends	address account	external	onlyOwner modifier is detected. Owner can call this function if the contract is not renounced.
setAutomatedMarketMakerPair	address pair, bool value	external	onlyOwner modifier is detected. Owner can call this function if the contract is not renounced.
updateLiquidityWallet	address newLiquidityWallet	external	onlyOwner modifier is detected. Owner can call this function if the contract is not renounced.
updateOperationsWallet	address newOperationsWallet	external	onlyOwner modifier is detected. Owner can call this function if the contract is not renounced.
updateBuyBackWallet	address newBuyBackWallet	external	onlyOwner modifier is detected. Owner can call this function if the contract is not renounced.
updateFees	uint256 bnbRewardPerc, uint256 liquidityPerc, uint256 operationsPerc, uint256 buyBackPerc	external	onlyOwner modifier is detected. Owner can call this function if the contract is not renounced. Liquidity % must be equal to or higher than operations and buyback combined. No fee limiter in place, fees may be set to 100.
updateGasForProcessing	uint256 newValue	external	onlyOwner modifier is detected. Owner can call this function if the contract is not renounced.
updateClaimWait	uint256 claimWait	external	onlyOwner modifier is detected. Owner can call this function if the contract is not renounced.
setIgnoreToken	address tokenAddress, bool isIgnored	external	onlyOwner modifier is detected. Owner can call this function if the contract is not renounced.
activateContract		public	onlyOwner modifier is detected. Owner can call this function if the contract is not renounced.
recoverContractBNB	uint256 recoverRate	public	onlyOwner modifier is detected. Owner can call this function if the contract is not renounced.
setMaxSellPercent	uint256 maxSellPercent	public	onlyOwner modifier is detected. Owner can call this function if the contract is not renounced.
renounceOwnership		public virtual	onlyOwner modifier is detected. Owner can call this function if the contract is not renounced.
transferOwnership	address newOwner	public virtual	onlyOwner modifier is detected. Owner can call this function if the contract is not renounced.

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.

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.

Pinksale

<https://www.pinksale.finance/#/pinklock/record/6514?chain=BSC>

Contract Code Audit – Mint Functions

This Contract Cannot Mint New MAPES 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.



7% Reward

7% of every buy/sell is taken and redistributed to all holders. Highest rewards tokens on the market. Hold \$MAPESe tokens, earn rewards!
Minimum 10000 tokens



4% Marketing

4% of every transaction is allocated to Marketing in BNB to allow for longevity and growth of Meta Apes. Swaps to BNB happen in real time to avoid dumps.



2% Gorilla Buy Back

2% of every transaction is sent into the Gorilla Fund Wallet and used to buy up tokens and add them to the staking pool! This helps encourage holding and adding constant buy pressure.

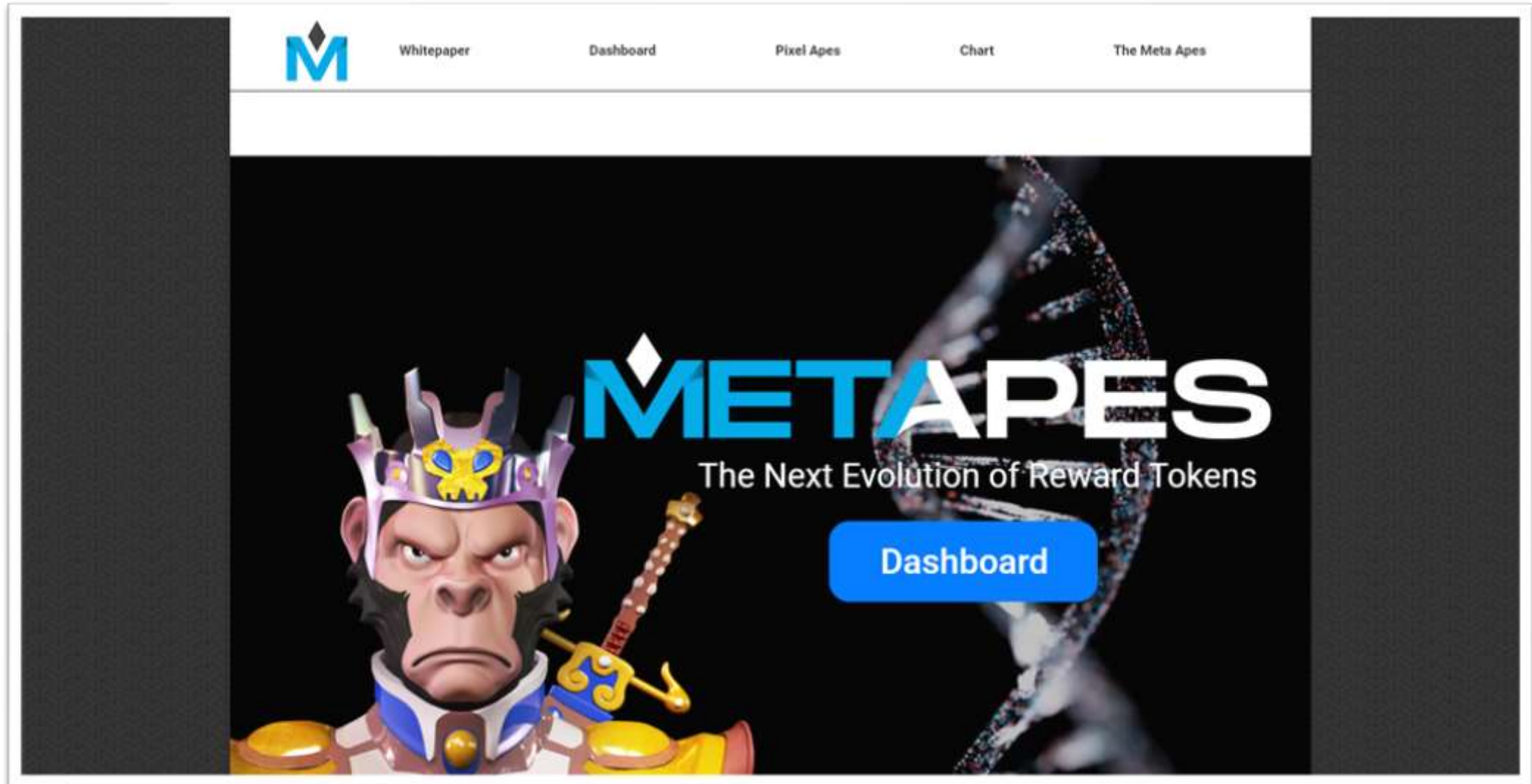


2% Liquidty

2% of every transaction is sent into liquidity for PancakeSwap. This helps create a price floor and stability while you HODL!

Website Part 1 – Overview

www.meta-apes.io



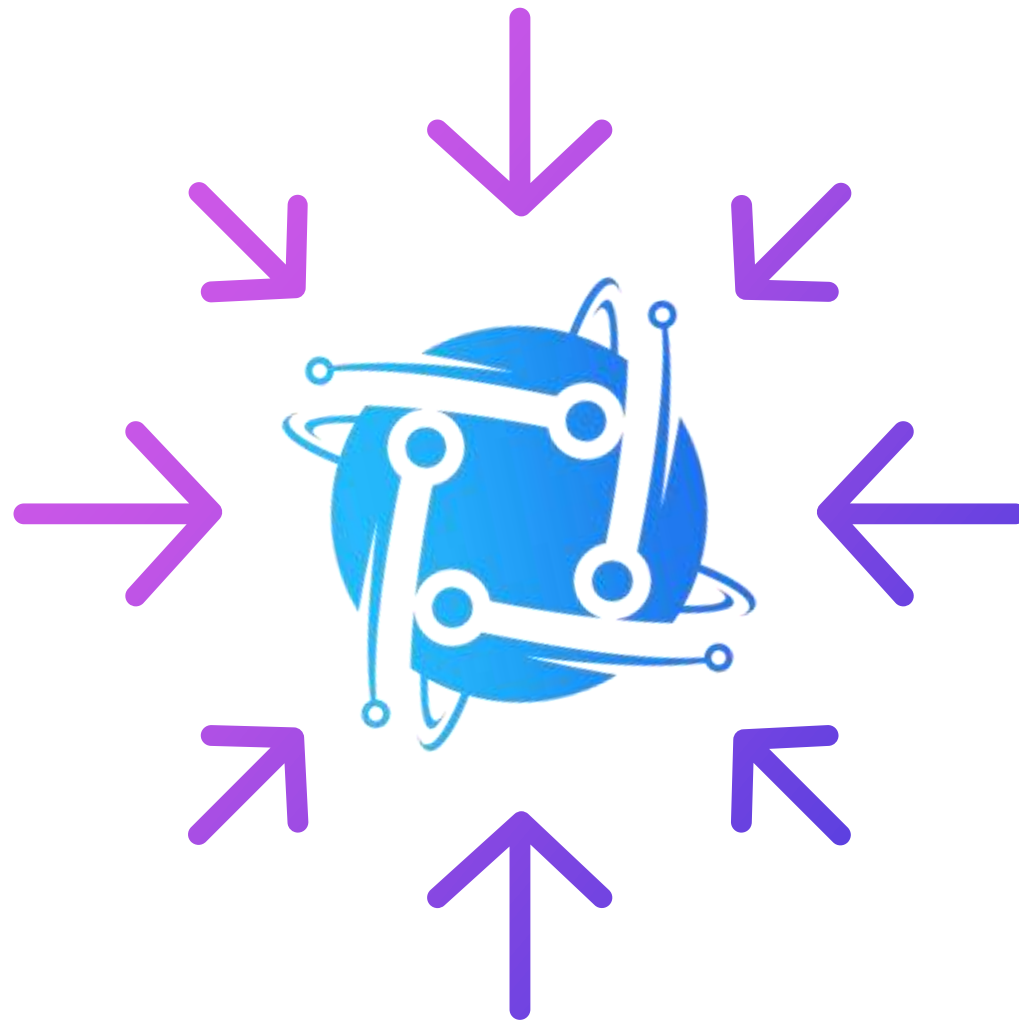
Above images are actual snapshots of the current live website of the project.

Website was registered on 12/04/2021, registration expires 12/04/2022.

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

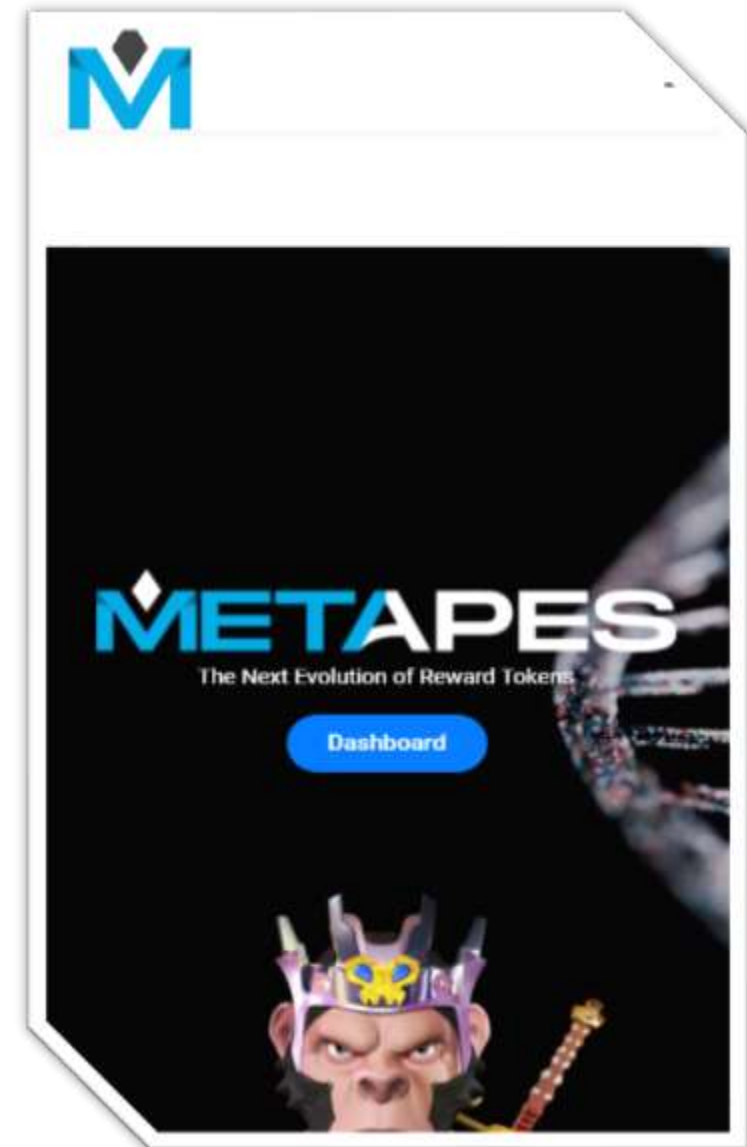
The website contained no JavaScript errors. No typos, or grammatical errors were present, and we found a valid SSL certificate allowing for access via https.

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: meta-apes.io

Issued by: Sectigo RSA

Valid Until: 12/09/2022



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

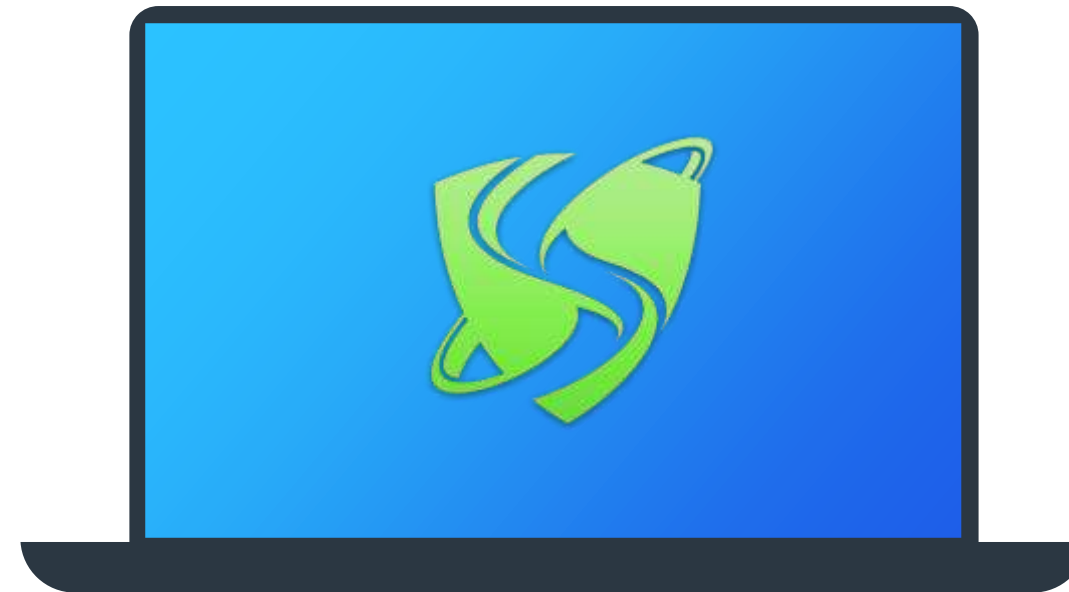
No internal server errors

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](#)



[Medium](#)



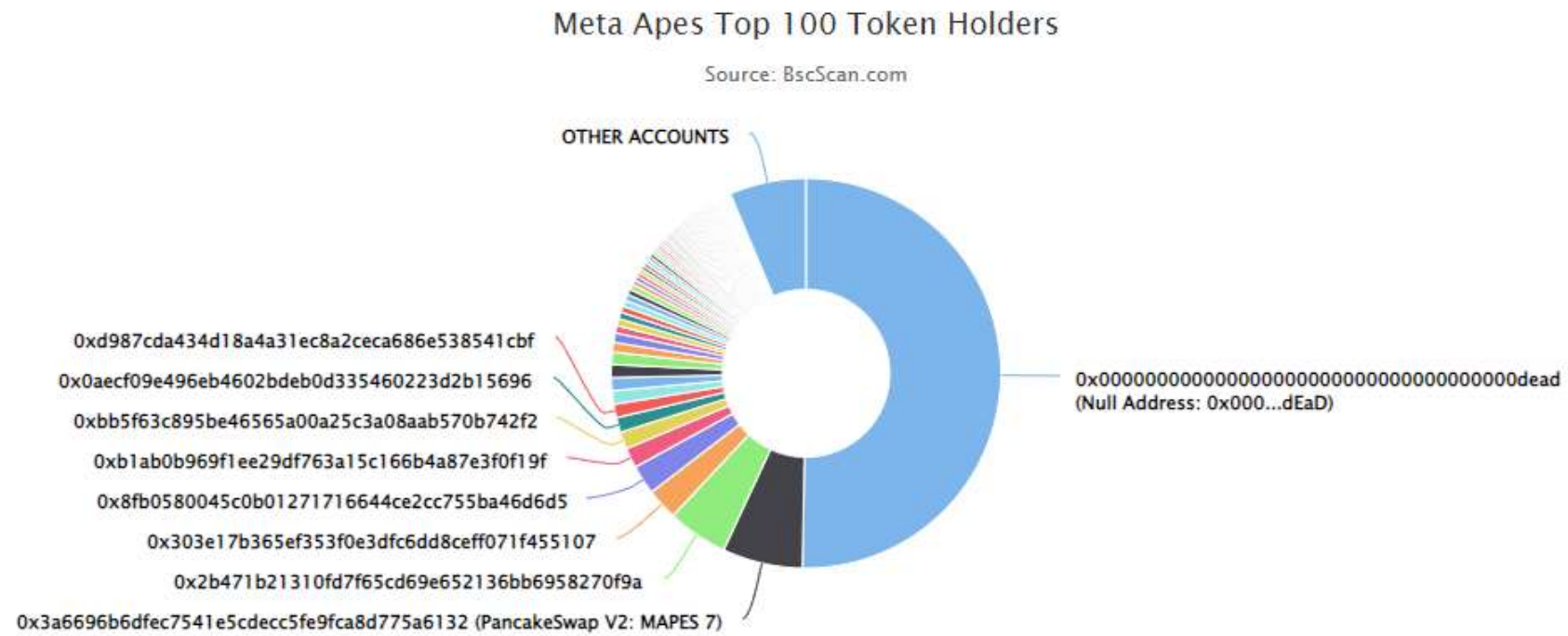
[GitHub](#)

✓ At least 3 social media networks were found.

Top Token Holders

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

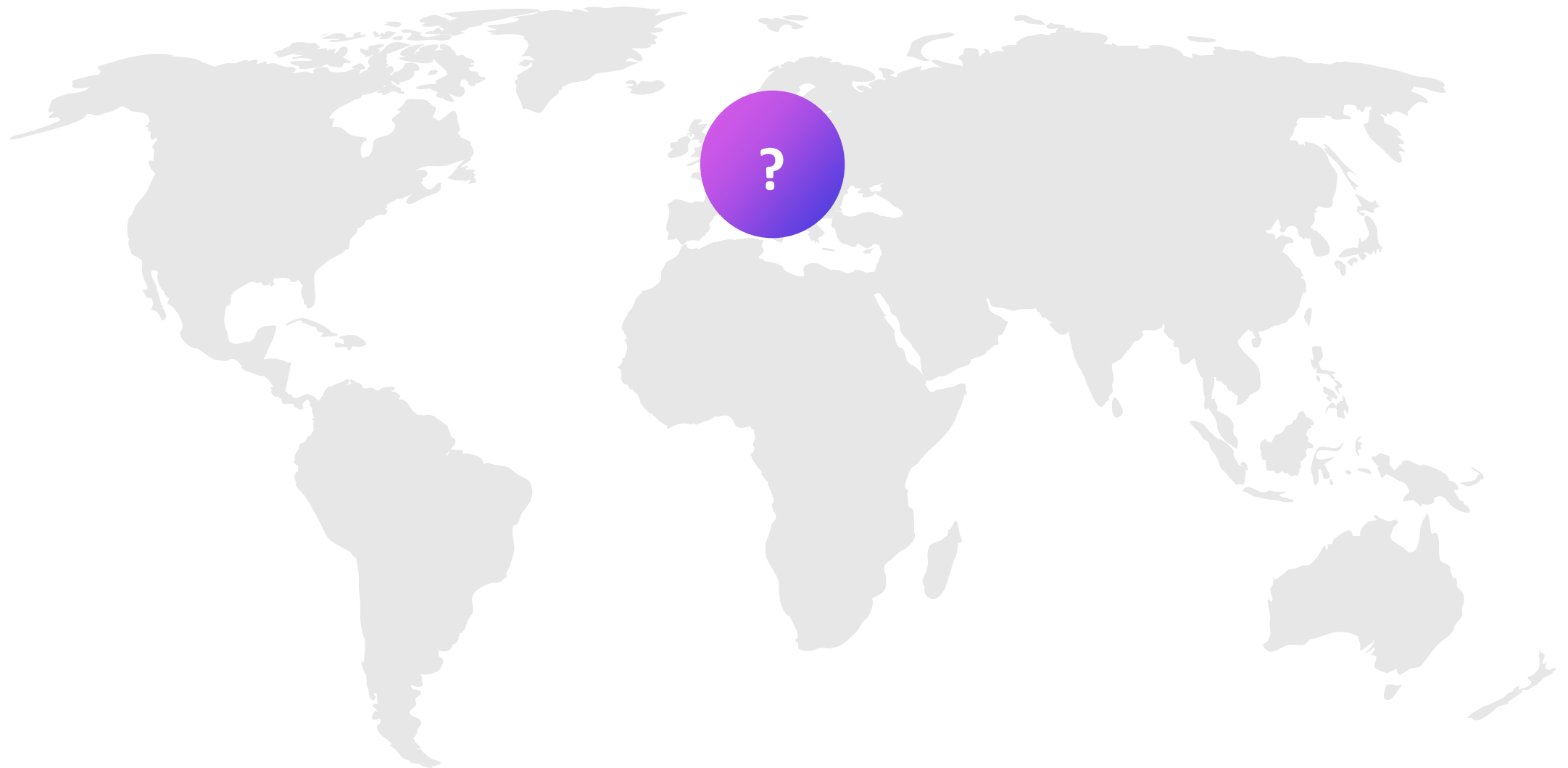
[Click here to view the most up-to-date list of holders](#)



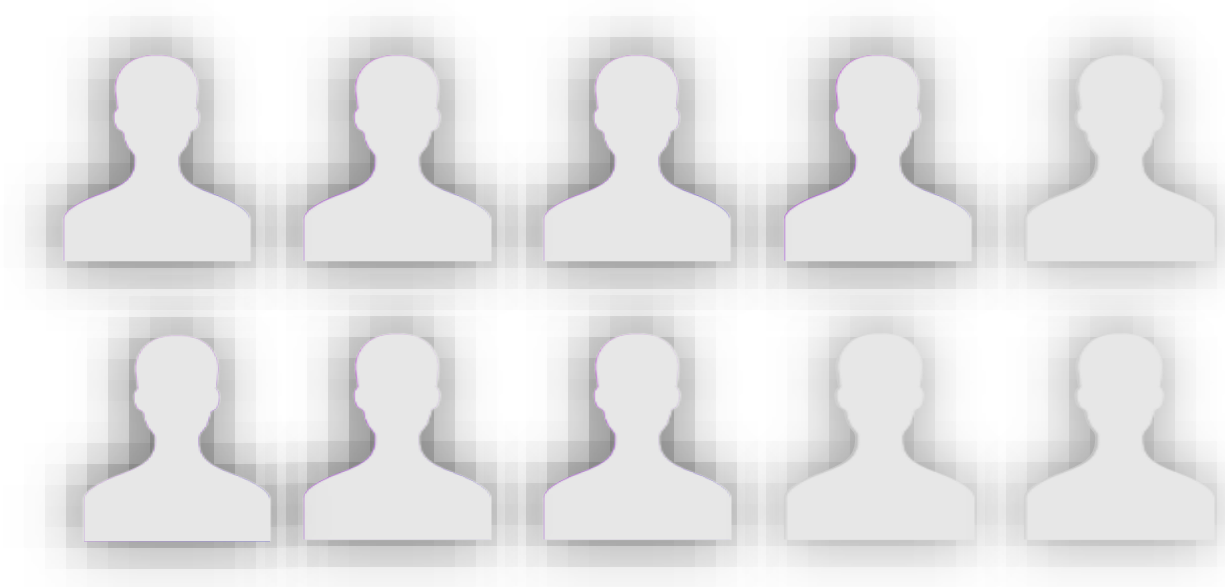
Rank	Address	Quantity (Token)	Percentage
1	Null Address: 0x000...dEaD	50,305,298.4579721	50.3053%
2	PancakeSwap V2: MAPES 7	6,659,694.859124713	6.6597%
3	0x2b471b21310fd7f65cd69e652136bb6958270f9a	5,000,000	5.0000%

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.

Roadmap THE PIXEL ERA



PHASE 1

The Pixel Era involves all the stages a meme token must go through. Meta Apes is looking to get full CMC listing/tracking, Coinbase & Cyprio.com price tracking, Trust Wallet & Poocoin logo complete. The Pixel Apes Club will continue to release upcoming seasons while a new NFT set by Meta Apes will be introduced

THE META ERA



PHASE 2

The Meta Era for Meta Apes involves its expansion into the NFT scene and the release of its first high quality NFTs (ERC721). Gamefi platform for staking/transforming Meta Apes NFTs will also be introduced . The Meta Apes Trading Card NFT game will be announced aswell.

THE MUTANT ERA



PHASE 3

The Mutant Era involves the launch of Meta-Muta Apes Mutant NFTs that you will be able to evolve from the gamefi staking platform. The alpha of Meta Apes Trading Card NFT Game will be released with the ability to buy random card packs. Introductions to Meta Apes A.I. generated NFTs will be released.

THE GOLDEN ERA



PHASE 4

The Golden Era is the final stage in Meta Apes road map. This will involve the release of a A.I. generated NFT set and will also include the release of Meta Apes TCG NFT game beta where players will be able to test early game features. Collab with BNB Shinobi Metaverse to introduce Meta Apes NFT's into the game. The release of Meta Apes new road map will also be launched.

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 META APES (MAPES) AT BLOCK NUMBER: **13711951**
THIS AUDIT IS ONLY VALID IF VIEWED ON WWW.DSSERTSWAP.FINANCE
1 DSRT HAS BEEN SENT TO AUDITED CONTRACT FOR VERIFICATION

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