DESSERT
FINANCE

Meta Phoenix Protocol (PHO)

BEP-20 Audit

Performed at block **16498459**

PERFORMED BY DESSERT FINANCE
FOR CONTRACT ADDRESS: 0x0065dD397b67B199b60A2fb9F5fE4Ca71A09ea5b

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.

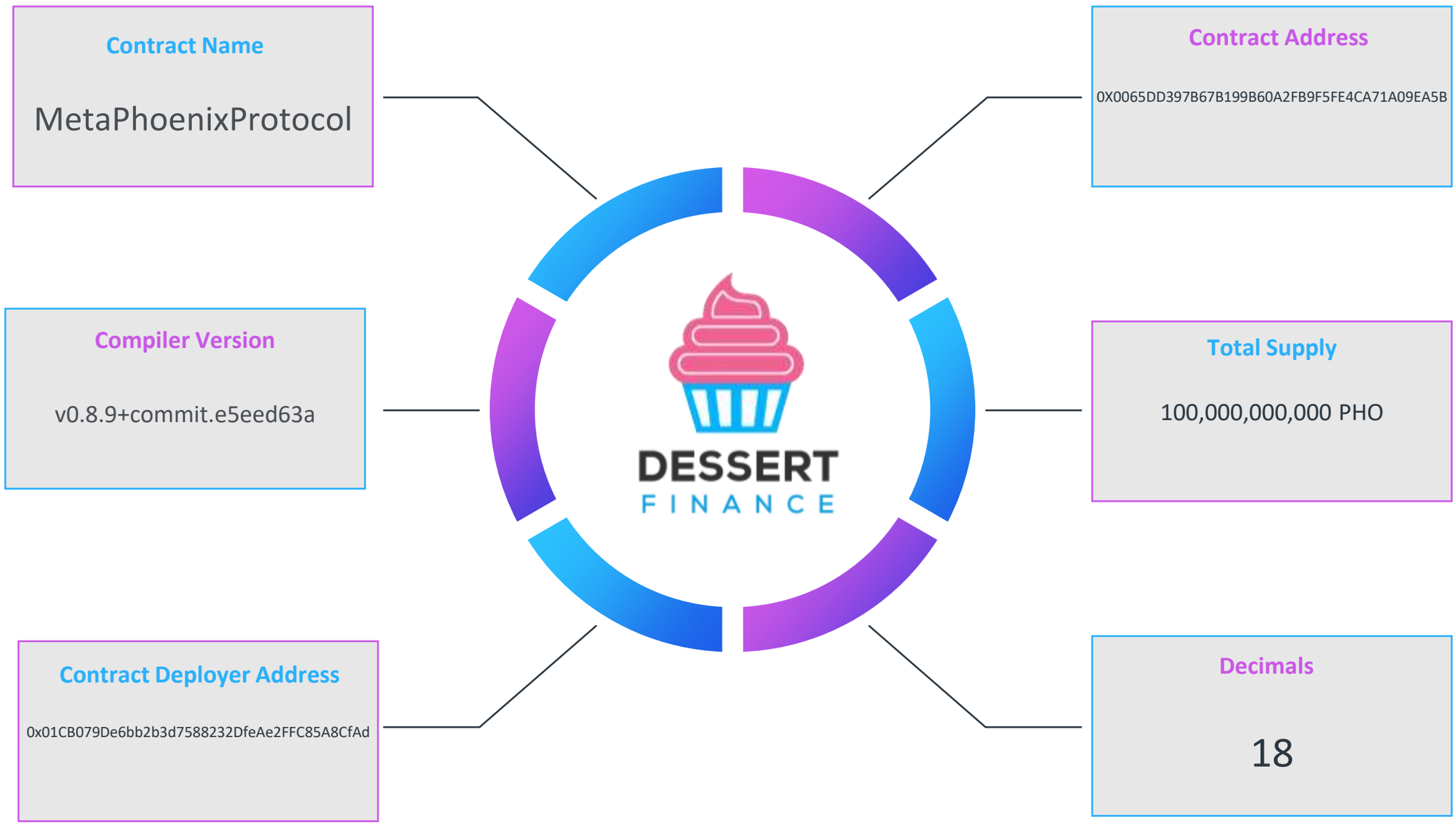
Dessert Finance in no way takes responsibility for any losses, nor does Dessert Finance encourage any speculative investments. The information provided in this audit is for information purposes only and should not be considered investment advice. Dessert Finance does not endorse, recommend, support,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

DessertDoxxed

DessertDoxxed is a KYC service offered by Dessert Finance that allows projects to do a private face reveal matched with an I.D to allow founders / team members to privately Doxx themselves to Dessert Finance.

Table of Contents

Contract Code Audit – Token Overview



BEP-20 Contract Code Audit – Overview

Dessert Finance was commissioned to perform an audit on Meta Phoenix Protocol (PHO)

```
import "../abstracts/Auth.sol";
import "../interfaces/IBEP20.sol";
import "../interfaces/IDEXFactory.sol";
import "../interfaces/IDEXRouter.sol";
import "../interfaces/IDividendDistributor.sol";
import "./DividendDistributor.sol";

contract MetaPhoenixProtocol is IBEP20, Auth {
    using SafeMath for uint256;

    address WBNB;
    address DEAD = 0x000000000000000000000000000000000000dEaD;
    address ZERO = 0x0000000000000000000000000000000000000000;

    string constant _name = "Meta Phoenix Protocol";
    string constant _symbol = "PHO";
    uint8 constant _decimals = 18;

    uint256 _totalSupply = 1 * 10**11 * 10**_decimals;

    uint256 public _maxTxAmount = _totalSupply.div(100).mul(1);
    uint256 public _maxWalletToken = _totalSupply.div(1000).mul(25);

    mapping (address => uint256) _balances;
    mapping (address => mapping (address => uint256)) _allowances;

    mapping (address => bool) _isBlacklisted;
    mapping (address => bool) _isLockedTeamWallet;
    mapping (address => bool) isFeeExempt;
    mapping (address => bool) isTxLimitExempt;
    mapping (address => bool) isMaxWalletExempt;
    mapping (address => bool) isTimelockExempt;
    mapping (address => bool) isDividendExempt;

    uint256 public liquidityFee    = 2;
    uint256 public reflectionFee   = 2;
    uint256 public devFee          = 5;
    uint256 public marketingFee    = 5;
    uint256 public charityFee      = 1;
    uint256 public totalFee        = charityFee + marketingFee + devFee + reflectionFee + liquidityFee;
    uint256 public feeDenominator  = 100;

    uint256 public sellMultiplier  = 200;

    address public autoLiquidityReceiver;
    address public devFeeReceiver;
    address public marketingFeeReceiver;
    address public charityFeeReceiver;

    uint256 targetLiquidity = 99;
    uint256 targetLiquidityDenominator = 100;

    IDEXRouter public router;
    address public pair;

    bool public tradingOpen = true;

    DividendDistributor public distributor;
    uint256 distributorGas = 500000;
```

Contract Address
0x0065dD397b67B199b60A2fb9F5fE4Ca71A09ea5b

TokenTracker
Meta Phoenix Protocol (PHO)

Contract Creator
0x01cb079de6bb2b3d7588232dfeae2ffc85a8cfad

Source Code
Contract Source Code Verified

Contract Name
MetaPhoenixProtocol

Other Settings
default evmVersion

Compiler Version
v0.8.9+commit.e5eed63a

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.

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:

0xa6fcc346f5156169d78b63cbbb40cc757b7f97a7

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 – Authorized Accessible Functions

Function Name	Parameters	Visibility	Audit Notes
setMaxWalletSize	uint256 amount		authroized modifier is detected. Authorized wallets can call this function.
setMaxWalletSizePercBase1000	uint256 percBase1000		authroized modifier is detected. Authorized wallets can call this function.
setTxLimit	uint256 amount		authroized modifier is detected. Authorized wallets can call this function.
setTxLimitPercBase1000	uint256 percBase1000		authroized modifier is detected. Authorized wallets can call this function.
setIsLockedTeamWallet	address holder, bool locked		authroized modifier is detected. Authorized wallets can call this function.
clearStuckBalance	uint256 amountPercentage		authroized modifier is detected. Authorized wallets can call this function.
clearStuckBalance_sender	uint256 amountPercentage		authroized modifier is detected. Authorized wallets can call this function.
setSellMultiplier	uint256 _sellMultiplier		authroized modifier is detected. Authorized wallets can call this function.
tradingStatus	bool _status		authroized modifier is detected. Authorized wallets can call this function.
setBuyCooldown	bool _status, uint8 _interval		authroized modifier is detected. Authorized wallets can call this function.
setSellCooldown	bool _status, uint8 _interval		authroized modifier is detected. Authorized wallets can call this function.
setIsDividendExempt	address holder, bool exempt		authroized modifier is detected. Authorized wallets can call this function.
setIsBlacklisted	address holder, bool blacklisted		authroized modifier is detected. Authorized wallets can call this function.
setIsFeeExempt	address holder, bool exempt		authroized modifier is detected. Authorized wallets can call this function.
setIsTxLimitExempt	address holder, bool exempt		authroized modifier is detected. Authorized wallets can call this function.
setIsMaxWalletLimitExempt	address holder, bool exempt		authroized modifier is detected. Authorized wallets can call this function.
setIsTimelockExempt	address holder, bool exempt		authroized modifier is detected. Authorized wallets can call this function.
setIsExempt	address holder, bool exempt		authroized modifier is detected. Authorized wallets can call this function.
setFees	uint256 _liquidityFee, uint256 _reflectionFee, uint256 _devFee, uint256 _marketingFee, uint256 _charityFee, uint256 _feeDenominator		authroized modifier is detected. Authorized wallets can call this function.
setFeeReceivers	address _autoLiquidityReceiver, address _devFeeReceiver, address _marketingFeeReceiver, address _charityFeeReceiver		authroized modifier is detected. Authorized wallets can call this function.
setSwapBackSettings	bool _enabled, uint256 _amount		authroized modifier is detected. Authorized wallets can call this function.
setTargetLiquidity	uint256 _target, uint256 _denominator		authroized modifier is detected. Authorized wallets can call this function.
setDistributionCriteria	uint256 _minPeriod, uint256 _minDistribution		authroized modifier is detected. Authorized wallets can call this function.
setDistributorSettings	uint256 gas		authroized modifier is detected. Authorized wallets can call this function.
multiTransfer	address[] calldata addresses, uint256[] calldata tokens		authroized modifier is detected. Authorized wallets can call this function.

The functions listed above can be called by an authorized user.

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

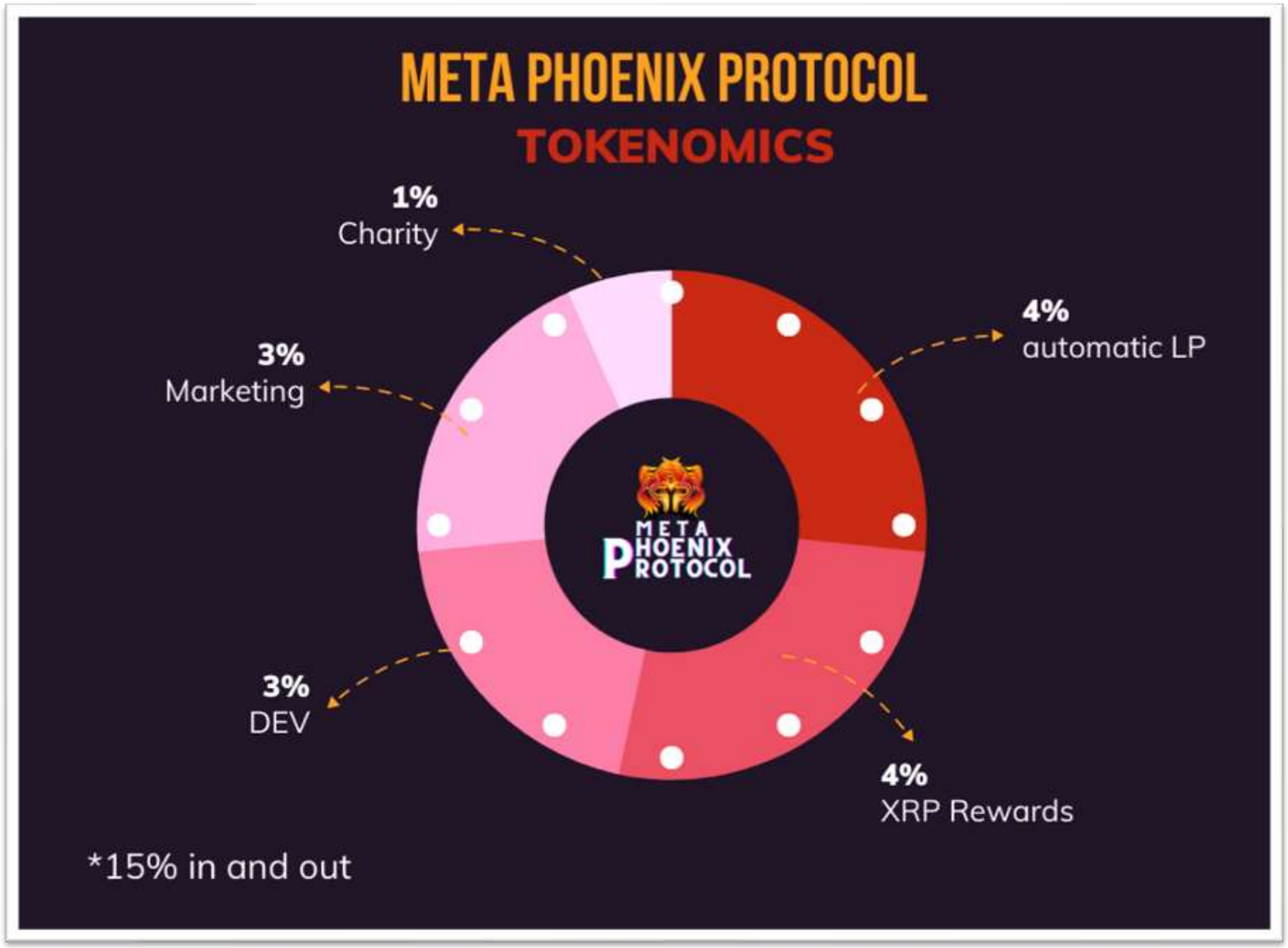
This Contract Cannot Mint New PHO Tokens.

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.

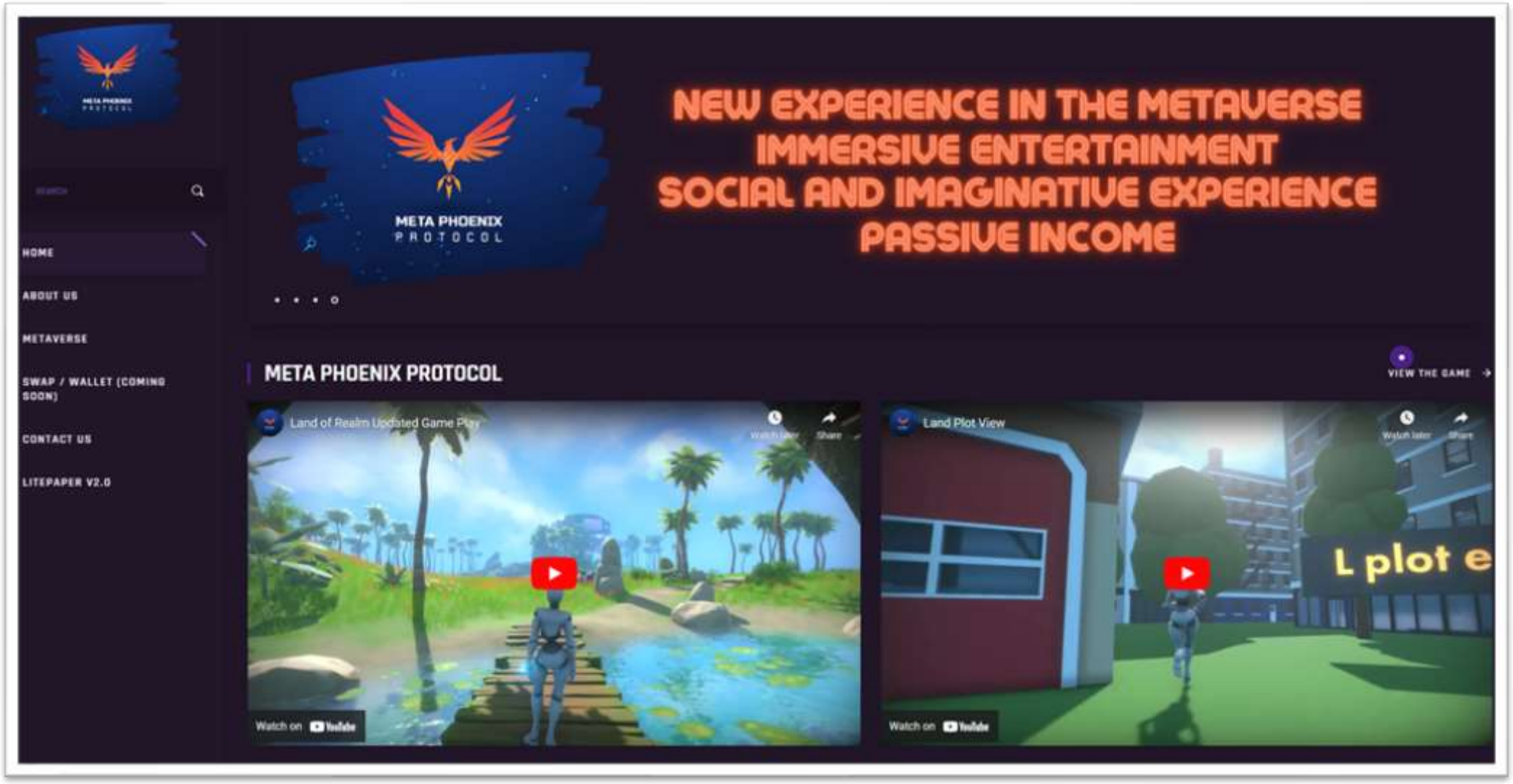
Contract Transaction Fees

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.



Website Part 1 – Overview

www.phoenix-protocol.com

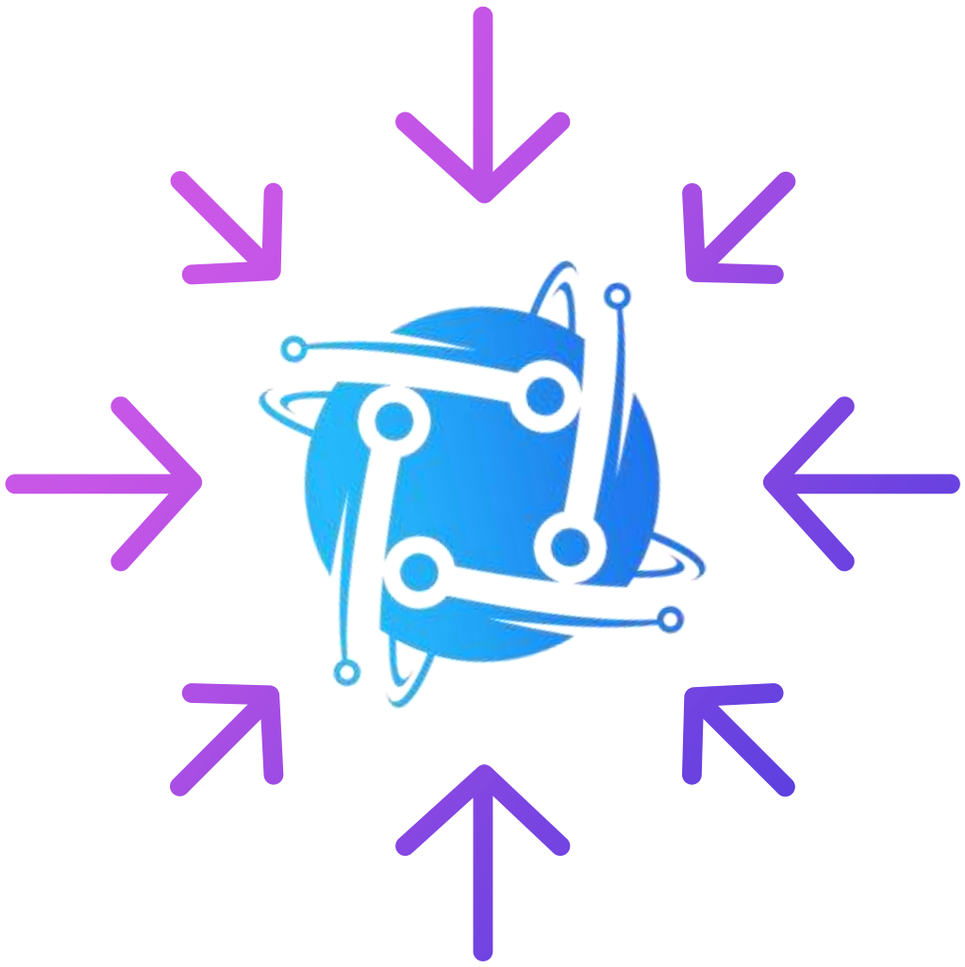


Above images are actual snapshots of the current live website of the project.

Website was registered on 01/17/2022, registration expires 01/17/2023.

✗ 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

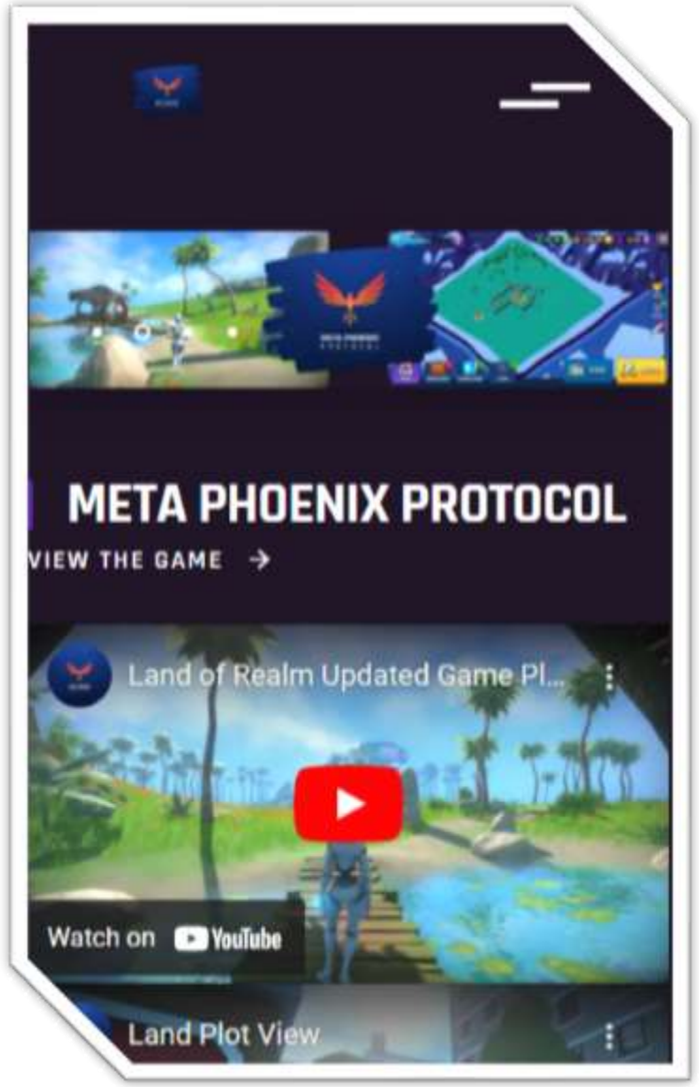
The website contained no JavaScript errors. No typos, or grammatical errors were present, and we found a valid SSL certificate allowing for access via https.

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: phoenix-protocol.com

Issued by: Encryption Everywhere

Valid Until: 01/24/2023

CONTACT EMAIL

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

Contact

Contact Form

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

Discord

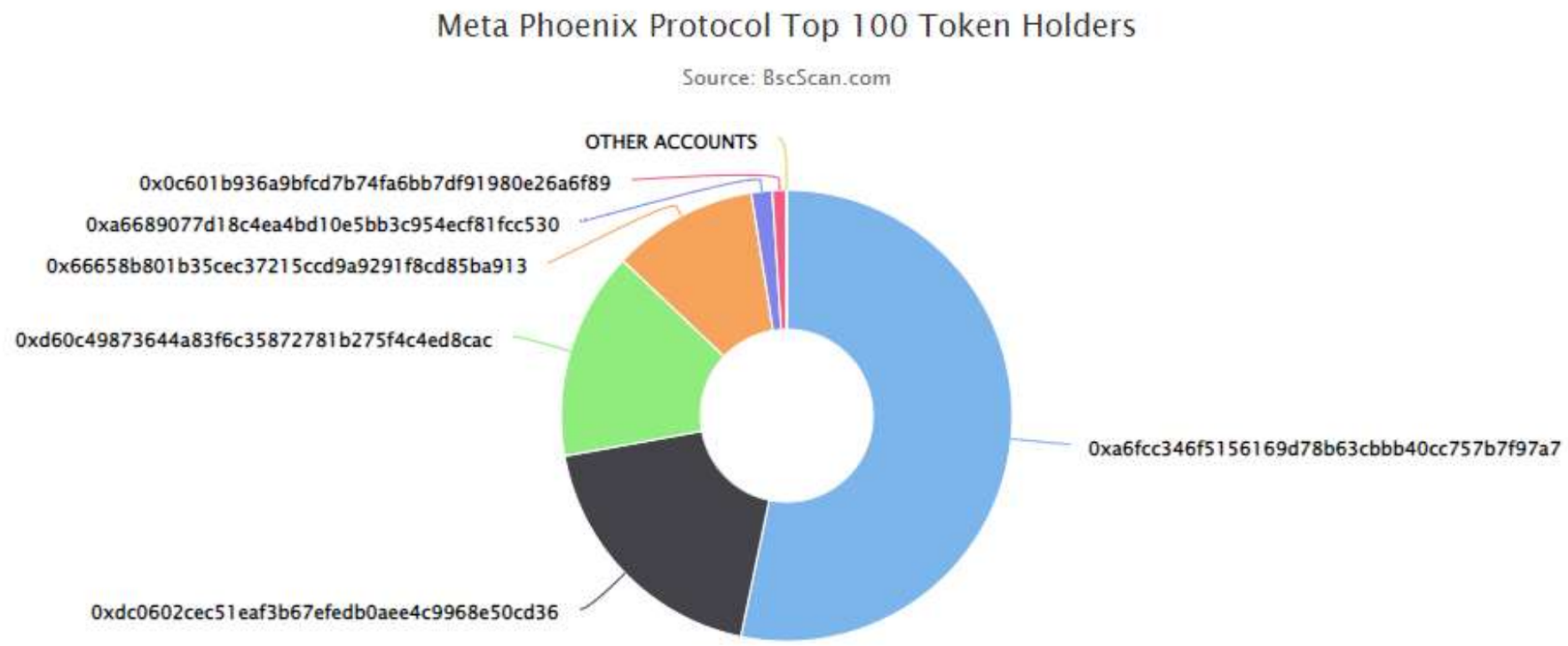
YouTube

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

Top Token Holders

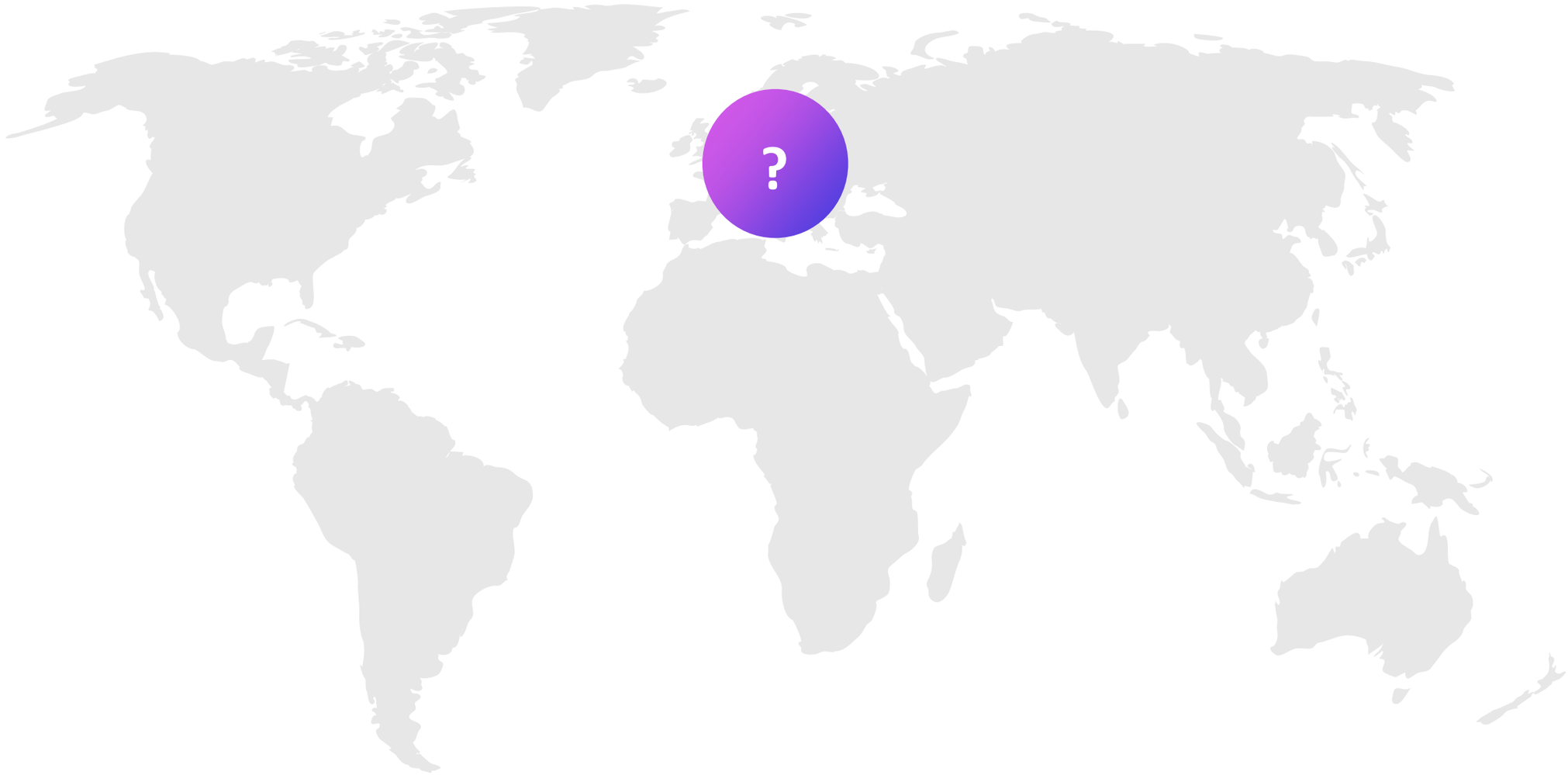
The entire supply was in a few wallets at the time of audit. We expect this to change as the project goes through initial distribution phases. Please use the link below to view the most up-to-date holder information.

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



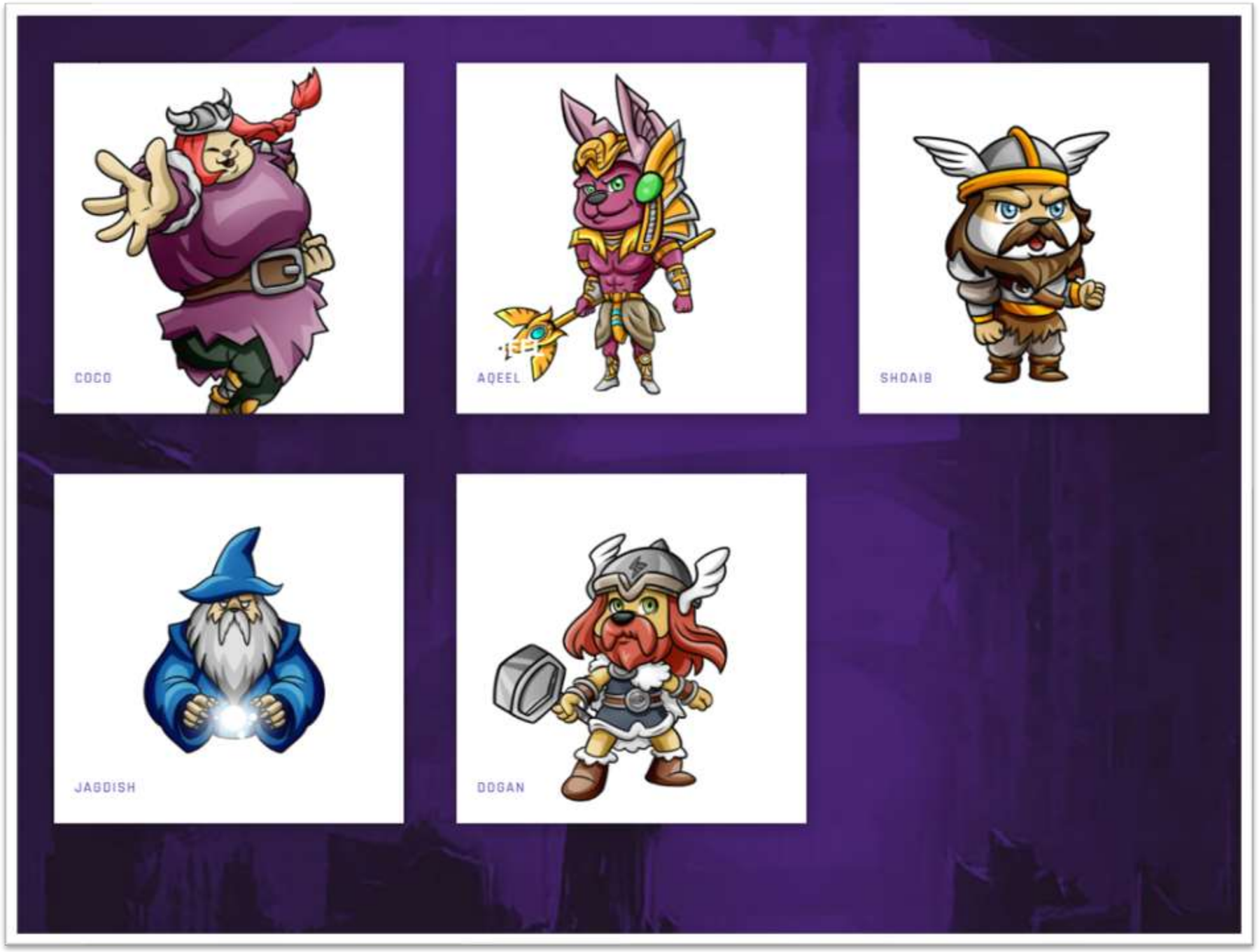
Location Audit

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



Team Overview

The following information was found on the projects website.



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 META PHOENIX PROTOCOL (PHO) AT BLOCK NUMBER: **16498459**

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

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