



DESSERT
FINANCE

Sekuya (SKUY)

Audit Report

PERFORMED BY DESSERT FINANCE

FOR CONTRACT ADDRESS: 0xE327Ce757CD206721e100812E744fc56e4E0A969

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.

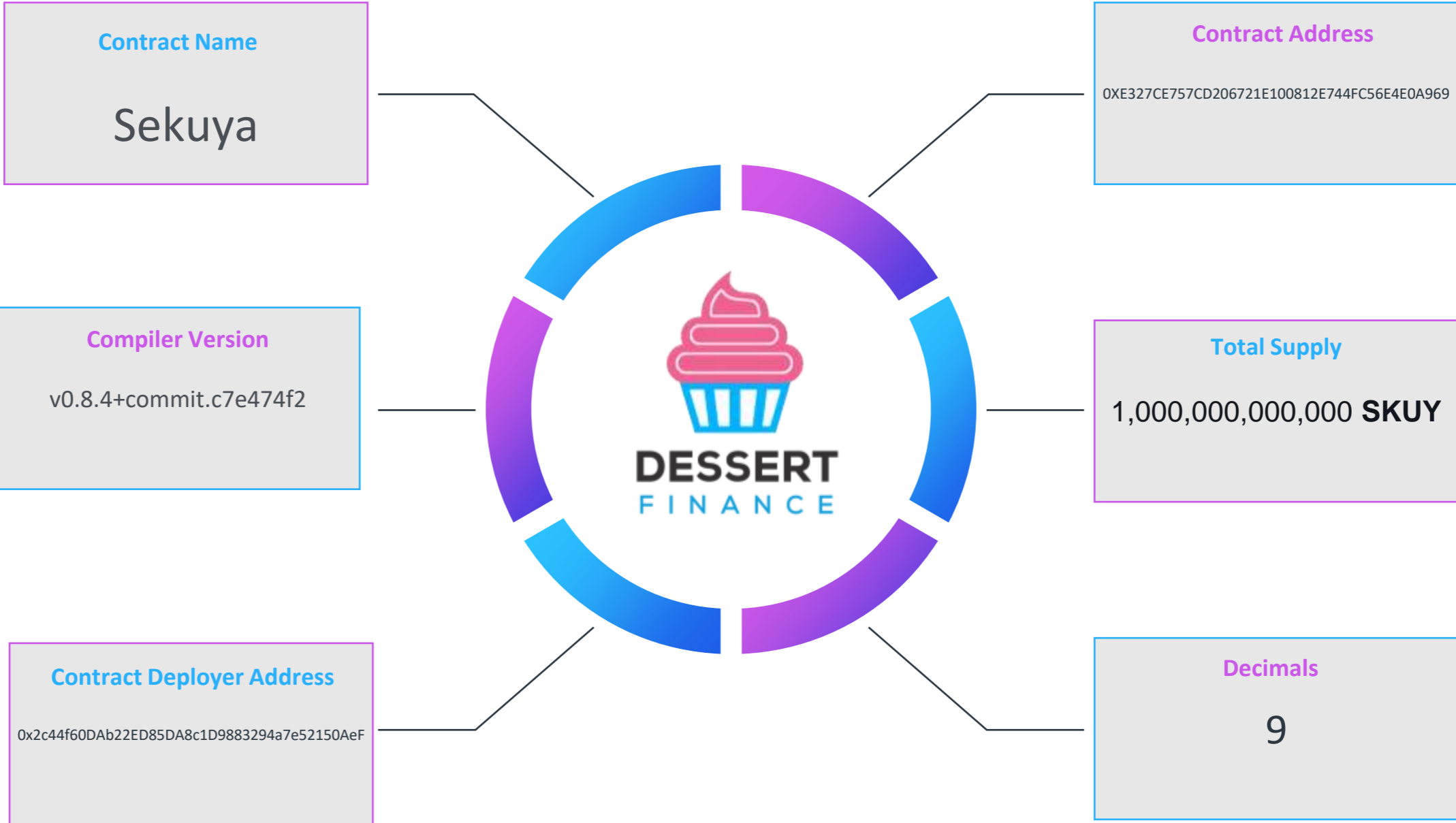
Dessert Finance in no way takes responsibility for any losses, nor does Dessert Finance encourage any speculative investments. The information provided in this audit is for information purposes only and should not be considered investment advice. Dessert Finance does not endorse, recommend, support,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



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. Contract Code Audit – Owner Accessible Functions
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 Sekuya (SKUY)

```
*Submitted for verification at BscScan.com on 2022-02-18
//
//
//
//
// Sekuya The New Earth Multiverse
// website:sekuya.io
// The First Multiverse Project In Crypto Space
//
// SPDX-License-Identifier: Unlicensed
pragma solidity ^0.8.4;

abstract contract Context {
    function _msgSender() internal view virtual returns (address payable) {
        return payable(msg.sender);
    }

    function _msgData() internal view virtual returns (bytes memory) {
        this; // silence state mutability warning without generating bytecode - see https://github.com/ethereum/solidity/issues/2691
        return msg.data;
    }
}

interface IERC20 {
    function totalSupply() external view returns (uint256);
    function balanceOf(address account) external view returns (uint256);
    function transfer(address recipient, uint256 amount) external returns (bool);
    function allowance(address owner, address spender) external view returns (uint256);
    function approve(address spender, uint256 amount) external returns (bool);
    function transferFrom(address sender, address recipient, uint256 amount) external returns (bool);
    event Transfer(address indexed from, address indexed to, uint256 value);
    event Approval(address indexed owner, address indexed spender, uint256 value);
}

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 sub(uint256 a, uint256 b, string memory errorMessage) internal pure returns (uint256) {
        require(b <= a, errorMessage);
        uint256 c = a - b;

        return c;
    }

    function mul(uint256 a, uint256 b) internal pure returns (uint256) {
        if (a == 0) {
            return 0;
        }
    }
}
```

Contract Address

0xE327Ce757CD206721e100812E744fc56e4E0A969

TokenTracker

Sekuya (SKUY)

Contract Creator

0x2c44f60dab22ed85da8c1d9883294a7e52150aef

Source Code

Contract Source Code Verified

Contract Name

Sekuya

Other Settings

default evmVersion, None

Compiler Version

v0.8.4+commit.c7e474f2

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 (x1)	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:

[0x2c44f60dab22ed85da8c1d9883294a7e52150aef](https://www.etherbase.net/etherbase/address/0x2c44f60dab22ed85da8c1d9883294a7e52150aef)

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
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.
excludeFromReward	address account	public	onlyOwner modifier is detected. Owner can call this function if the contract is not renounced.
includeInReward	address account	external	onlyOwner modifier is detected. Owner can call this function if the contract is not renounced.
excludeFromFee	address account	public	onlyOwner modifier is detected. Owner can call this function if the contract is not renounced.
includeInFee	address account	public	onlyOwner modifier is detected. Owner can call this function if the contract is not renounced.
excludeOriginFromFee	address account	public	onlyOwner modifier is detected. Owner can call this function if the contract is not renounced.
includeOriginInFee	address account	public	onlyOwner modifier is detected. Owner can call this function if the contract is not renounced.
excludeDestinationFromFee	address account	public	onlyOwner modifier is detected. Owner can call this function if the contract is not renounced.
includeDestinationInFee	address account	public	onlyOwner modifier is detected. Owner can call this function if the contract is not renounced.
setTaxFeePercent	uint256 taxFee	external	onlyOwner modifier is detected. Owner can call this function if the contract is not renounced.
setLiquidityFeePercent	uint256 liquidityFee	external	onlyOwner modifier is detected. Owner can call this function if the contract is not renounced.
setMaxTxAmount	uint256 maxTxAmount	external	onlyOwner modifier is detected. Owner can call this function if the contract is not renounced.
setDevAndMarketingDivisor	uint256 divisor	external	onlyOwner modifier is detected. Owner can call this function if the contract is not renounced.
setNumTokensSellToAddToLiquidity	uint256 _minimumTokensBeforeSwap	external	onlyOwner modifier is detected. Owner can call this function if the contract is not renounced.
setBuybackUpperLimit	uint256 buyBackLimit	external	onlyOwner modifier is detected. Owner can call this function if the contract is not renounced.
setMarketingAddress	address _marketingAddress	external	onlyOwner modifier is detected. Owner can call this function if the contract is not renounced.
setDevAddress	address _devAddress	external	onlyOwner modifier is detected. Owner can call this function if the contract is not renounced.
setMinimumBeforeBuyBack	uint256 _minimumBeforeBuyBack	external	onlyOwner modifier is detected. Owner can call this function if the contract is not renounced.
setPercentToBuyBack	uint256 _percentToBuyBack	external	onlyOwner modifier is detected. Owner can call this function if the contract is not renounced.
setSwapAndLiquifyEnabled	bool _enabled	public	onlyOwner modifier is detected. Owner can call this function if the contract is not renounced.
setBuyBackEnabled	bool _enabled	public	onlyOwner modifier is detected. Owner can call this function if the contract is not renounced.
recoverERC20	address tokenAddress, uint256 tokenAmount	public	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.

DxSale:

<https://dxsale.app/app/v3/dxlockview?id=0&add=0x2c44f60DAb22ED85DA8c1D9883294a7e52150AeF&type=lplock&chain=BSC>

Contract Code Audit – Mint Functions

This Contract Cannot Mint New SKUY Tokens.

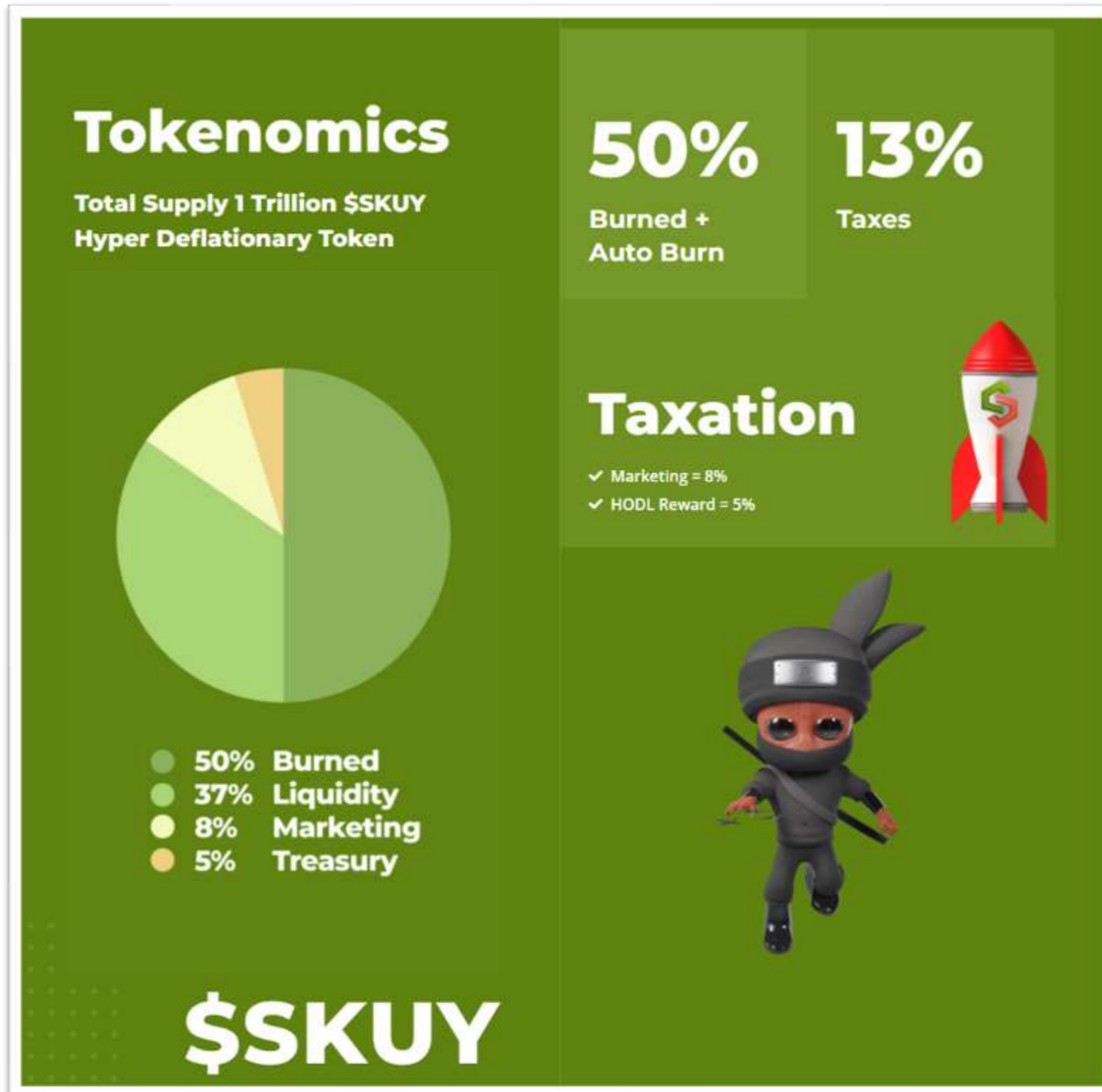


We do understand that sometimes mint functions are essential to the functionality of the project.

A mint function was not found in the contract code.

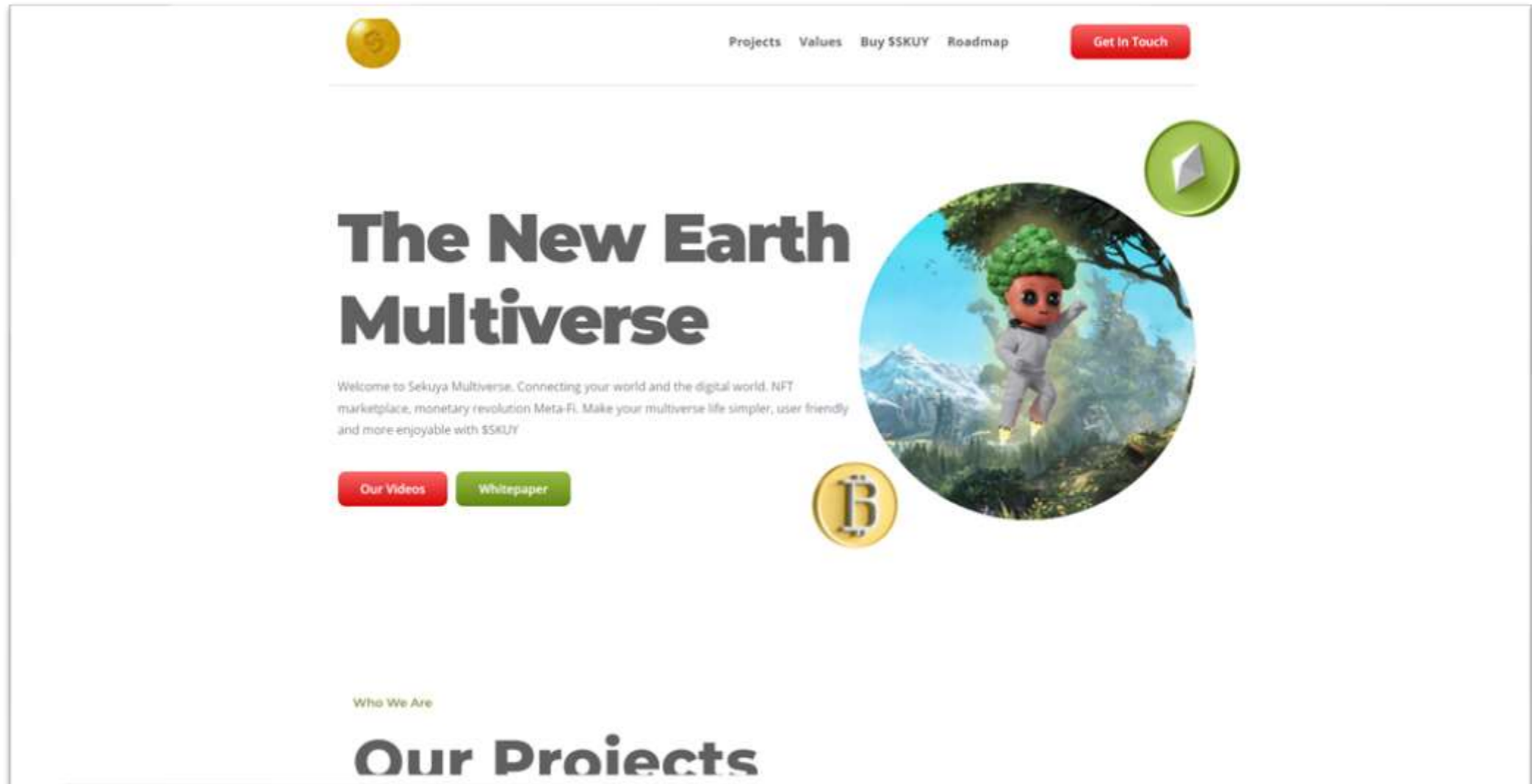
Contract Transaction Fees

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.sekuya.io



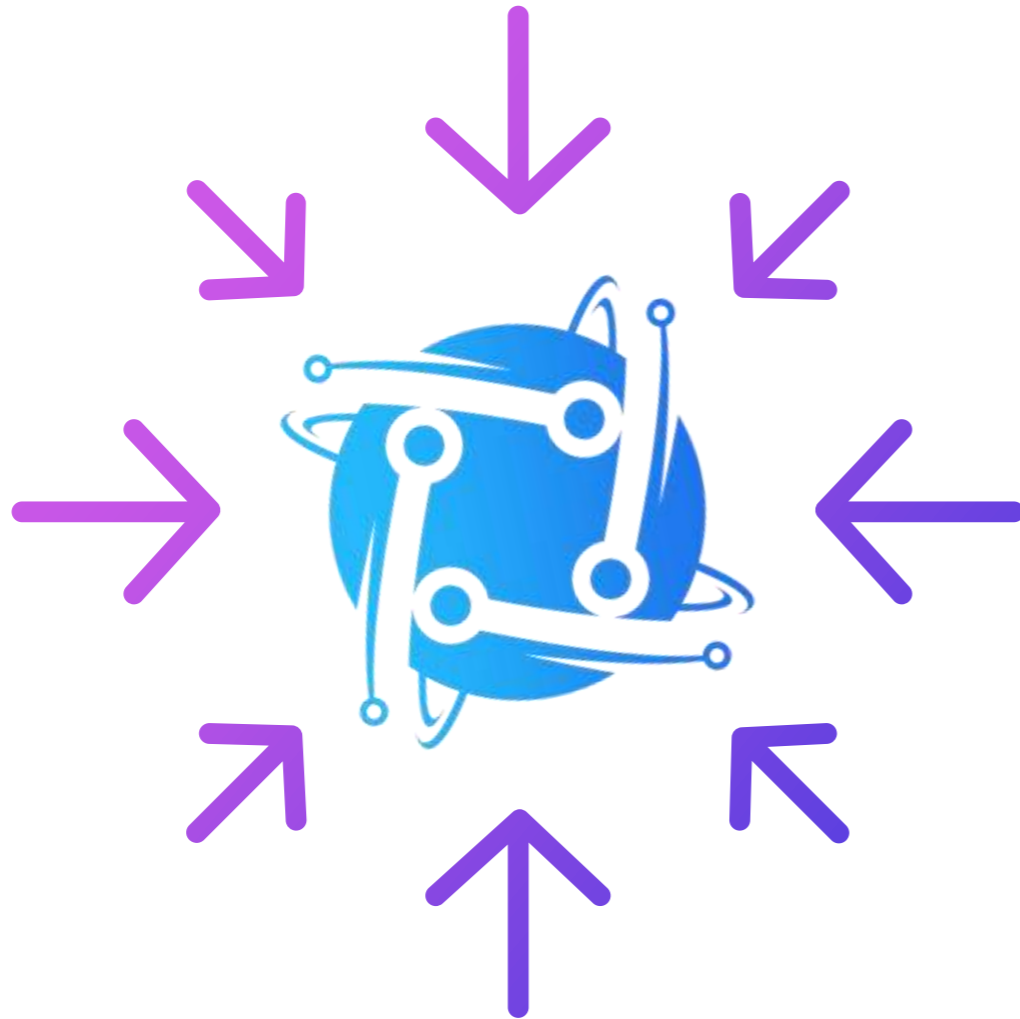
Above images are actual snapshots of the current live website of the project.

Website was registered on 01/13/2022, registration expires 01/13/2023.

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: *.sekuya.io

Issued by: E1

Valid Until: 04/29/2022



CONTACT EMAIL

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

Contact

marketing@sekuya.io



SPAM / MALWARE / POPUPS

No malware found

No injected spam found

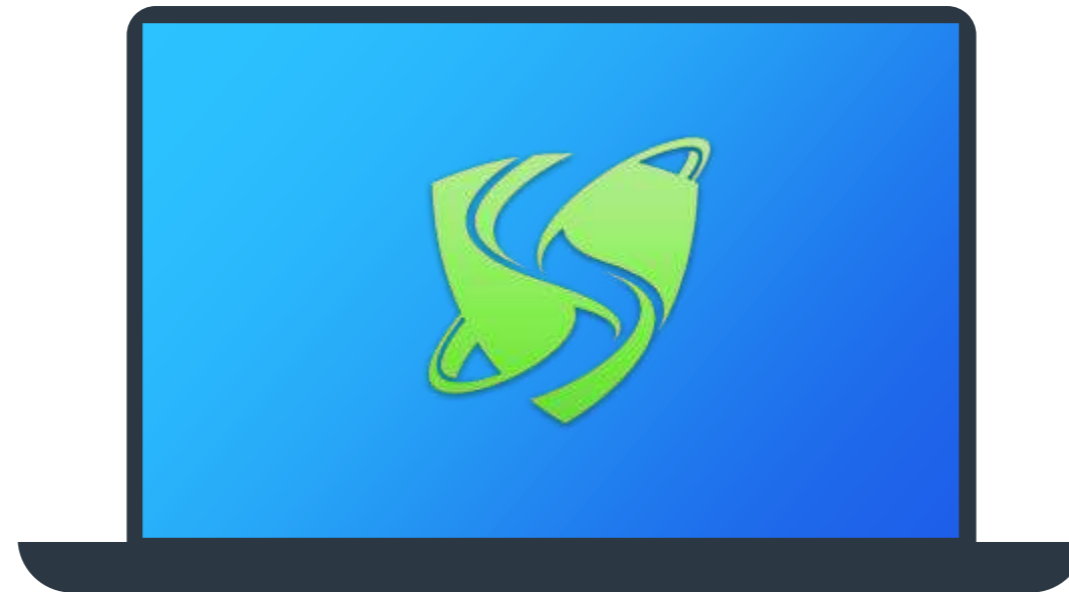
No internal server errors

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.

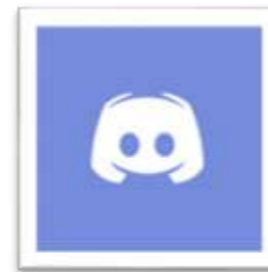
All links have been conveniently placed below.



[Twitter](#)



[Telegram](#)



[Discord](#)



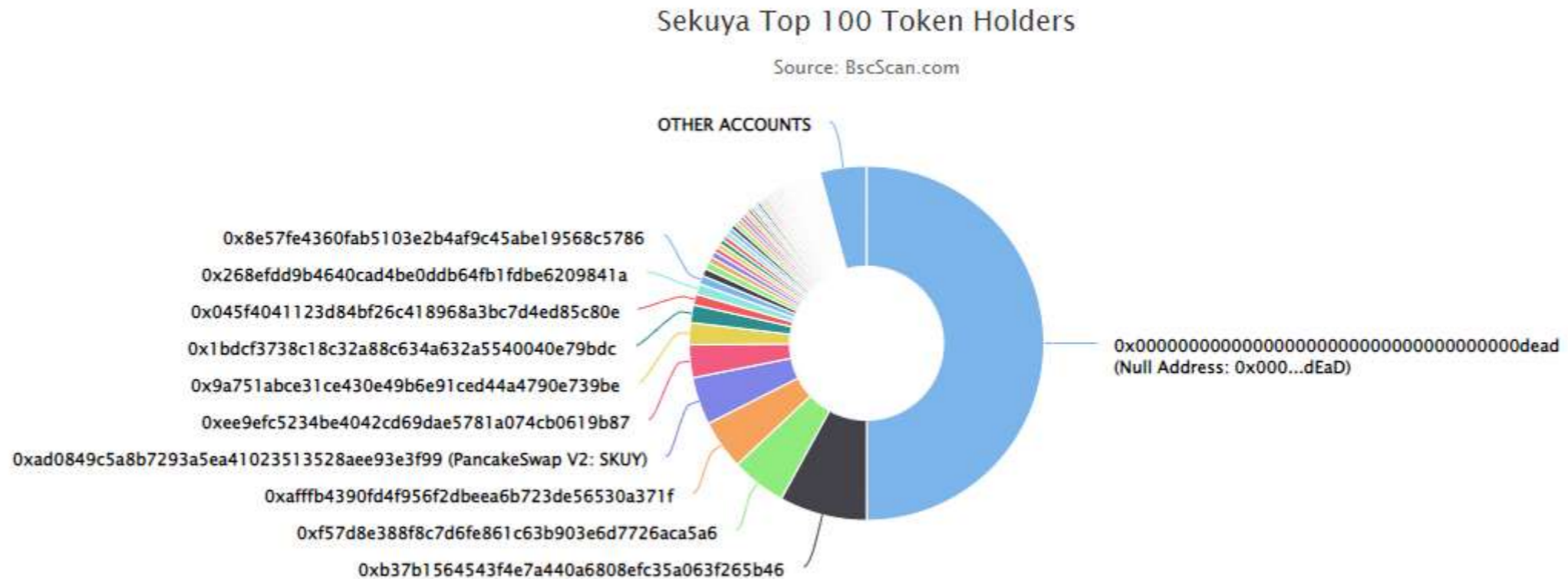
[YouTube](#)

✓ **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	500,000,000,000	50.0000%
2	0xb37b1564543f4e7a440a6808efc35a063f265b46	80,000,740,077.690518321	8.0001%
3	0xf57d8e388f8c7d6fe861c63b903e6d7726aca5a6	50,002,416,282.731509957	5.0002%
4	0xafffb4390fd4f956f2dbeea6b723de56530a371f	45,200,000,000	4.5200%
5	PancakeSwap V2: SKUY	43,490,241,373.572992931	4.3490%

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 about the team was located on the projects website.



Haibara | CEO

Haibara is a CEO with more than 6 years experience in startups and passion for the tech industry.



Noel Newton | CMO

Noel Newton has more than 8 years experience as a Marketing & Brand Consultant and has helped more than 100+ brands.



Hoshiko | CTO

Hoshiko is a developer with more than 10 years experience in the tech industry.

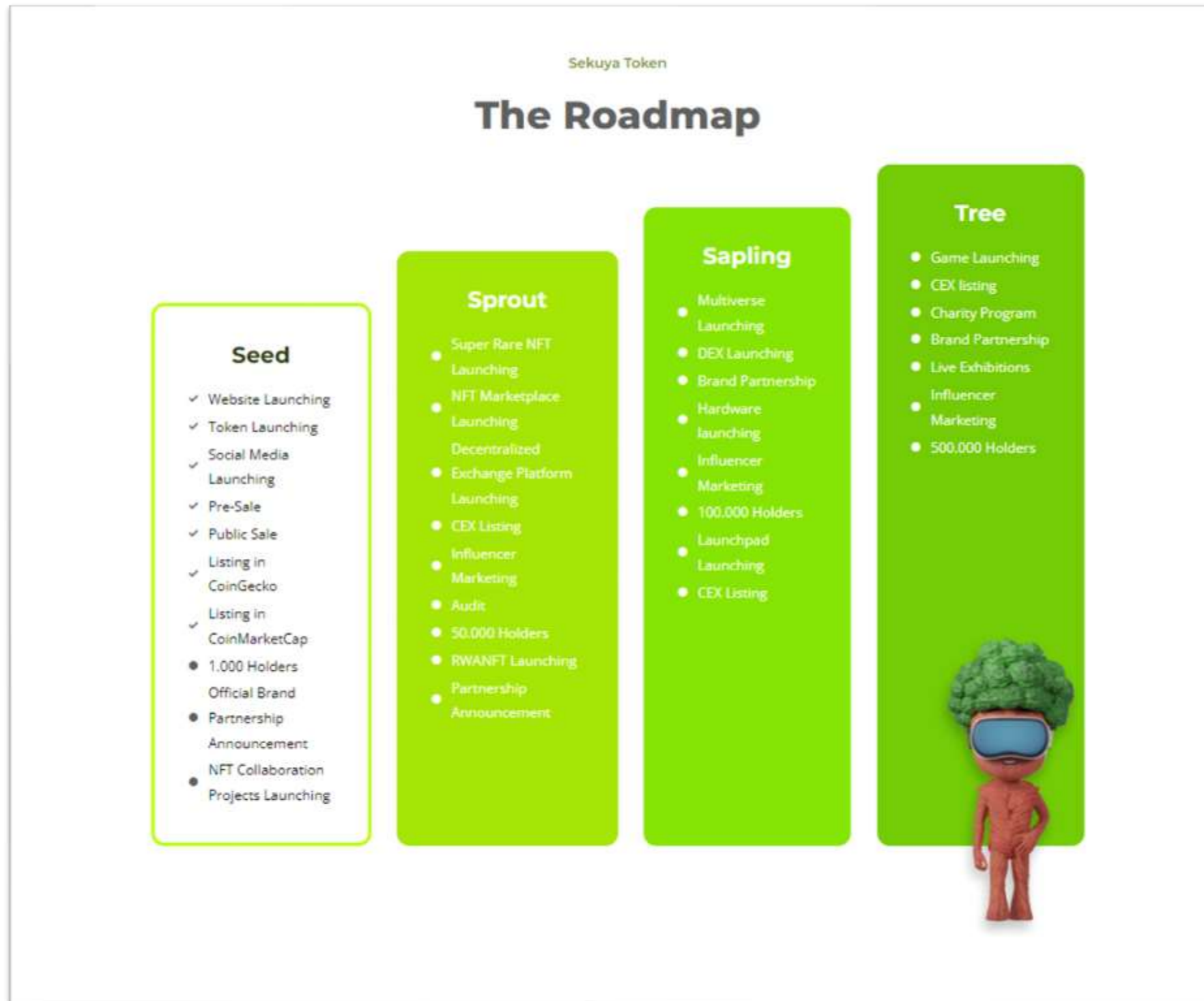


GodMother | CCO

Aiko has more than 13 years experience as an architect and multi-disciplined businesswoman.

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 SEKUYA (SKUY) 1 DSRT HAS BEEN SENT TO AUDITED PROJECT'S CONTRACT ADDRESS FOR VERIFICATION OF THIS AUDIT AT BLOCK NUMBER: **15862500**

THIS AUDIT IS ONLY VALID IF VIEWED ON [HTTPS://WWW.DSSERTSWAP.FINANCE](https://www.dessertswap.finance)

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