Swaptor Whitepaper

Marko Ivanković[1]

[1]Berry Block, marko@berryblock.io

Abstract

Peer-to-peer (P2P) swaps on blockchain have the potential to greatly benefit users by eliminating the need for intermediaries in the exchange of assets. However, the use of P2P swaps on blockchain also presents several challenges, including trust issues that must be addressed in order to ensure their success. Since there is no intermediary to oversee the exchange of assets, users must rely on the trustworthiness of the other party to the transaction. In the absence of a trusted third party, it is difficult to verify the authenticity and quality of the assets being exchanged, which can lead to disputes and losses for users.

Furthermore, P2P swaps on blockchain are subject to potential security risks, such as hacking and fraud. Since the transactions are conducted directly between users, there is a greater risk of malicious actors attempting to exploit vulnerabilities in the system. This risk is exacerbated by the fact that blockchain transactions are irreversible, meaning that users have no recourse if their assets are stolen or lost.

In this paper we describe Swaptor, a decentralized P2P exchange dapp which aims to eliminate problems mentioned above.

Contents

1 Architectural Overview

As most modern dapps, Swaptor's architecture is a mix of on-chain and off-chain components:

- **On-chain**: Smart contracts
- **Off-chain**: Backend, Frontend and Database

1.1 On-chain Architecture

Smart contracts are written in Solidity, and their architecture is designed to be minimalistic. Specifically, the Swaptor contract is a singular entity whose sole purpose is to verify digital signatures and execute swaps upon successful verification. Digital signatures encode the specific terms under which a given swap is considered valid.

1.1.1 Signatures

In order to verify the signature, the smart contract must possess both the original swap elements and the signature itself. Prior to being passed as an argument to one of the *acceptSwap* functions, these elements are encoded using the *abi.encode* function. The signature is generated by creating a *keccak256* hash of the encoded arguments and signing it with the private key belonging to the seller. This method dramatically reduces the gas fees because the seller never had to transfer their assets to Swaptor contract.

1.1.2 Swap types

At present, Swaptor facilitates the swapping of ERC-721 and ERC-20 tokens, enabling the execution of any combination of swaps involving these types of tokens. In the future, Swaptor intends to extend its support to include native currency and ERC-1155 tokens.

1.1.3 Fees

To accept the swap, an address will need to pay a fixed fee of $5 in the native currency, which is calculated using Chainlink oracles[1]. It should be noted that the fee is not fixed and may be subject to change.

1.2 Off-chain Architecture

1.2.1 Backend

1.2.2 Frontend

1.2.3 Database & Security

2 Tokenomics

2.1 Token Mechanics

The price of Swaptor Token (SWPTR) is influenced by several factors, including activity on the Swaptor Protocol, the price of Ether, and liquidity pool activity. As mentioned earlier, every time a successful swap occurs, a fixed fee in the native currency

is exchanged for SWPTR token in the Ether/SWPTR liquidity pool. The amount of SWPTR token is then burned, leading to an increase in the price of SWPTR token.

Given that Ether is on the other side of the pool, the price of SWPTR is directly affected by the price of Ether. The decision to use Ether as the currency for the exchange fee is based on the fact that users already pay gas fees in Ether, and adding an additional swap would result in gas wastage. Moreover, Ether is the backbone of many other coins, and a decrease in its price would effectively eliminate a significant portion of the Ethereum DeFi ecosystem.

2.2 Token Allocation

Total of $1,000,000$ SWPTR tokens will be minted and distributed over time as described by Figure 1.

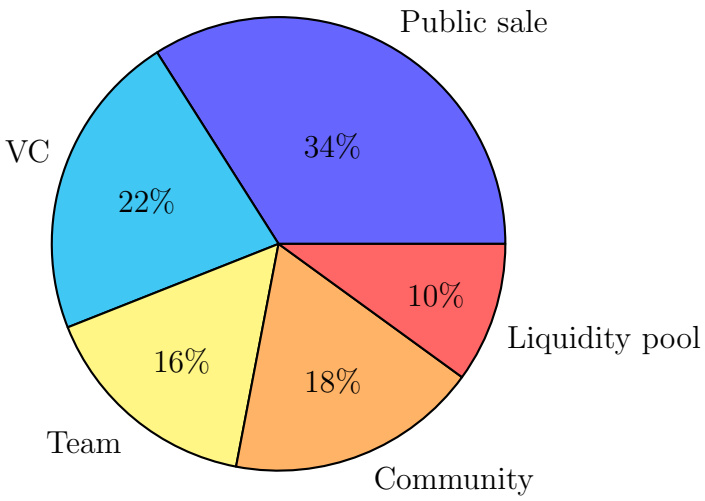

Figure 1: Pie chart showing the distribution of tokens

The tokens allocated to the community, as illustrated in Figure 1, will be distributed from the Swaptor contract. Upon swap execution, both the buyer and seller will be granted a single SWPTR token. Buyer and seller are motivated to hold the token as its price will grow proportional to Swaptor activity.

2.3 Utility

At present, the SWPTR token serves only to reflect the strength of the project and the level of traction it has gained. Rational investors who believe in the project's bright future may choose to hold the token, as its price increases every time a swap is executed.

During the community token distribution phase, it is expected that the price increase will be slightly lower, as two SWPTR tokens will be released from the Swaptor contract, giving actors the option to swap them for Ether. However, this scenario is highly unlikely, as the amount earned during this phase will be negligible, partly due to the gas fees required for swapping. Any impact on the price of SWPTR during community distribution phase due to actors who decide to swap tokens for Ether is partially dampened by the swap and burn mechanism executed by the Swaptor contract.

2.4 Vesting

Token vesting will occur at the end of every three-month period. This vesting will apply to tokens acquired by venture capitalists, public sales, and the team. This

measure has been introduced to increase the trust of the community by regulating token dumping and aligning the interests of the team with the project itself.

3 Roadmap

Q3 2023
Description of item 1 goes here.

Q4 2023
Description of item 2 goes here.

Q1 2024
Description of item 3 goes here.

References

[1] What is an oracle in blockchain? explained — chainlink, `https://chain.link/education/blockchain-oracles`