SaucerSwap v1 Core

Matthew DeLorenzo
Solidity Lead

Joseph Bergvinson
Tokenomics Lead

July 2022

Abstract

This technical whitepaper places SaucerSwap in the context of Uniswap v2 core contracts. Hedera smart contracts were upgraded to allow Hedera native tokens to be traded using an automated market maker protocol. Several modifications to Uniswap v2 are covered, including a novel structure of rent payment, use of the factory contract as the burn address, and ERC20 inheritance for HTS precompiles. Wrapped hbar – an analogue to weth – is introduced. These modifications demonstrate the re-architecture of Ethereum smart contracts such that they take full advantage of the high performance and predictable fee structures of the Hedera network.

1 Introduction

SaucerSwap is a fork of Uniswap V2, an on-chain system of smart contracts on the Ethereum blockchain – that leverages the Hedera Smart Contract Service (HSCS) to include Solidity smart contract integration with the Hedera Token Service (HTS). These smart contracts implement an automated market maker (AMM) protocol based on the constant product formula,

$$xy = k$$

For more information on the Uniswap v2 algorithm, please refer to the Uniswap v2 core whitepaper. [1]

On the Ethereum network, the dominant ERC20 and ERC721 standards are used for token operations like mint, burn, and transfer. In these standards, tokens are contracts and token operations change contract states. The Hedera ecosystem differentiates itself from Ethereum in that token operations are performed on HTS. HSCS was recently upgraded to allow smart contracts to use HTS through Ethereum Virtual Machine (EVM) precompiles. [2] After these upgrades, it became possible to create a decentralized exchange (DEX) using tokens on HTS.

[1] Hayden Adams, Noah Zinsmeister, Dan Robinson. *Uniswap v2 Core*. March 2020. `https://uniswap.org/whitepaper.pdf`
[2] Hedera Team. *Hedera Hashgraph Announces Mainnet Launch of EVM-Compatible Smart Contracts 2.0*. February 2022. `https://hedera.com/blog/hedera-hashgraph-announces-mainnet-launch-of-evm-compatible-smart-contracts-2-0`

While it was possible to create a DEX using the ERC20 standard, the Hedera ecosystem at the time of writing is more amenable to tokens created by and controlled through HTS, and represents the novelty of SaucerSwap.

An important feature of Hedera is the network's consistent and predictable fee structures for token operations. For example, a token transfer is pegged to approximately $0.0001, payable in Hedera's native token, hbar, on HTS.

For more information on the Hedera ecosystem, including HSCS and HTS, please visit their docs. [3]

2 Modifications to UniswapV2

2.1 Establish rent payer for all contracts

At the time of writing, Hedera plans to charge rent to smart contracts as a function of the number of key/value pairs in a contract. Various models have been presented, such as pay-per-use, donation-based, and dynamic fees. [4] SaucerSwap is designed to charge fees in hbar for expanding contract state. Fees are ultimately transferred from the contract to a designated rent paying account (`rentPayer`).

The fees on SaucerSwap are denominated in U.S. dollars, but are payable in hbar. SaucerSwap smart contracts fetch the U.S. dollar to hbar conversion factor using an exchange rate precompiled contract (at address `0x168`). This exchange rate is relevant to fee calculations on the Hedera network, or in other words, how many hbar are needed to achieve a successful token operation without reversion. The precompile should not be used as a reliable financial instrument for determining the hbar/USD exchange rate because it is only designed to calculate fees charged by Hedera.

SaucerSwap's `UniswapV2Factory` contract uses the exchange rate precompile, as this requires a fixed fee (which can be adjusted by `feeToSetter`) to create a new liquidity pool and pair contract. At launch, $1.00 in hbar is sent to the pair contract from msg.value to create the contract's LP token. The pair contract's address is used as the treasury key of the LP token, meaning that tokens are minted to the pair contract. It is also used as the supply key of the LP token, which gives the contract sole minting and burning rights over its LP token. This is immutable and achieves decentralization of the liquidity pool.

The fixed fee (minus the hbar required to create the LP token) is sent to the `rentPayer` account, from which Hedera may draw the funds to pay smart contract rent. Every pair contract inherits the rent paying account of the factory contract, which is set to `rentPayer`. This structure of rent payment ensures the pair contracts will not be delinquent.

The function declaration of `createPair` is,

```
function createPair(address tokenA, address tokenB)
        external payable costsTinycents(pairCreateFee) override returns (ad-
dress pair) {
```

`createPair` was made payable to collect the payment required for LP token creation and the

[3] Hedera Documentation. `https://docs.hedera.com/guides/`

[4] Gehrig Kunz. *Smart contract rent is coming to Hedera*. April 2022. `https://hedera.com/blog/smart-contract-rent-is-coming-to-hedera`

future rent for the new pair contract. The modifier `costsTinycents(pairCreateFee)` enforces that `msg.value` is greater than `pairCreateFee` (in tinycents, where 1 cent = 108 tinycents).

2.2 Burn address for `MINIMUM_LIQUIDITY`

On the Hedera network, one cannot send HTS tokens to a burn address to which no one owns the private keys. Accounts must associate with the tokens they wish to receive. Association serves as a protection to limit out of control state expansion using spam tokens, which poses a threat due to low fees on Hedera.

Because no burn address exists, SaucerSwap uses the factory contract as the burn address to hold `MINIMUM_LIQUIDITY`. Upon creation of the pair contract and its LP token, the factory associates itself with the newly created LP token and holds `MINIMUM_LIQUIDITY`.

2.3 ERC20 inheritance for HTS precompiles

The SaucerSwap pair contracts inherit an abstract contract which calls HTS precompiles to mint, burn, transfer, and associate HTS tokens, as described in HIP-206. [5] Each precompile call requires a successful response from HTS, and reverts otherwise.
The pair contract gets its balances of `token0` and `token1` using the IERC20 implementation described in HIP-218. [6] The function `balanceOf` is used because it is important for the pair contract to know its own balances of `token0` and `token1` to correctly calculate its own reserves. The pair contract also uses the HIP-218 implementation of the function `totalSupply` in the pair contract's mint and burn external functions.

3 Wrapped hbar

Uniswap v2 makes extensive use of wrapped ether (weth) to convert the native token ether to one that conforms to the ERC20 standard. Likewise, SaucerSwap uses wrapped hbar (whbar) to convert hbar to one conforming to HTS token standards. The whbar token is created using a token create precompile in the constructor of the wrapped hbar contract. Much like the LP tokens of pair contracts, the wrapped hbar contract controls the minting and burning of whbar tokens.

The only functions present in the wrapped hbar contract are deposit and withdraw. Users may transfer and associate whbar tokens outside of HSCS, or by calling HIP-206 precompiles in smart contracts.

---

[5] Danno Ferrin. *HIP-206: Hedera Token Service Precompiled Contract for Hedera Smart Contract Service*. November 2021. `https://hips.hedera.com/hip/hip-206`

[6] Danno Ferrin. *HIP-218: Smart Contract interactions with Hedera Token Accounts*. December 2021. `https://hips.hedera.com/hip/hip-218`

References

[1] Hayden Adams, Noah Zinsmeister, Dan Robinson. *Uniswap v2 Core*. March 2020. https://uniswap.org/whitepaper.pdf

[2] Hedera Team. *Hedera Hashgraph Announces Mainnet Launch of EVM-Compatible Smart Contracts 2.0*. February 2022. https://hedera.com/blog/hedera-hashgraph-announces-mainnet-launch-of-evm-compatible-smart-contracts-2-0

[3] Hedera Documentation. https://docs.hedera.com/guides/

[4] Gehrig Kunz. *Smart contract rent is coming to Hedera*. April 2022. https://hedera.com/blog/smart-contract-rent-is-coming-to-hedera

[5] Danno Ferrin. *HIP-206: Hedera Token Service Precompiled Contract for Hedera Smart Contract Service*. November 2021. https://hips.hedera.com/hip/hip-206

[6] Danno Ferrin. *HIP-218: Smart Contract interactions with Hedera Token Accounts*. December 2021. https://hips.hedera.com/hip/hip-218

4 Disclaimer

This technical whitepaper is for general information purposes only. It does not constitute investment advice or a recommendation or solicitation to buy or sell any investment and should not be used in the evaluation of the merits of making any investment decision. It should not be relied upon for accounting, legal, tax advice, or investment recommendations.