SundaeSwap Fundamentals

Pi Lanningham
with input from the SundaeSwap team

June 1, 2021

Abstract

SundaeSwap is a decentralized exchange built for the Cardano blockchain. It allows participants of the blockchain to provide liquidity and create a market for others to exchange their native tokens. In return, swappers pay a small fee and liquidity providers earn a return on their deposit. This and future whitepapers will outline an initial product modeled after the protocol popularized by Uniswap, with several innovative adaptations for the Cardano blockchain.

1 Background

The term “Decentralized Exchange” (DEX) refers to an application accessible through a series of smart contracts running on a suitable blockchain, which enables financial services traditionally facilitated by a central entity. Instead, trustless parties can participate in a financial market, relying on the behavior of the smart contracts to secure the transactions.

In addition to decentralizing *access* to financial services, a DEX also typically decentralizes *profits* from those services. Participants who provide the liquidity to create the market collect a small fee, creating a vehicle for passive income at returns usually reserved for large institutions and unheard of for the individual.

Several successful DEXs have been built on existing blockchains: Uniswap and Curve on Ethereum, and PancakeSwap on the Binance Smart Chain, to list a few notable examples. SundaeSwap is a DEX being built for a new blockchain, the Cardano blockchain.

The Cardano blockchain is a new third generation blockchain focused on, among other things, proof of stake for throughput and energy efficiency. Some reports suggest that the entire Cardano network is 1.6 million times more efficient than Bitcoin, for example [3].

As this new ecosystem opens up, the users and businesses that choose to operate on the Cardano blockchain will have a great and pressing need for the financial services described above.

There's one difference between Cardano and other blockchains that is of particular note for this paper. The accounting model and virtual machine are dramatically different from those on other smart-contract enabled blockchains. Tokens are tracked as bundles of unspent outputs from previous transactions, and can be locked with a validation script that determines under what conditions they can be spent. We'll go into this in more detail in Section 3.

This first paper provides the necessary background information that future whitepapers will build upon for the SundaeSwap protocol. There are a number of models we are evaluating, and seek to remain flexible, so details may change as we approach launch date.

Outline The remainder of this paper is organized as follows:

- Section 2 gives an introduction to the principals of an "Automated Market Maker."
- We describe a naive implementation of this scheme on the Cardano blockchain in Section 3.
- We detail a plan for enabling long term protocol upgrades in Section 4.
- We discuss some promising opportunities for additional improvements in Section 5.
- Finally, Section 6 summarizes the work.

2 AMMs and Constant Product Liquidity Pools

In classical finance, an exchange acts as a central authority, maintaining an order book and matching buyers with sellers to facilitate the exchange. Because of the incredible potential this entity has for manipulating the market for further profit, there is an immense amount of existing regulation in place.

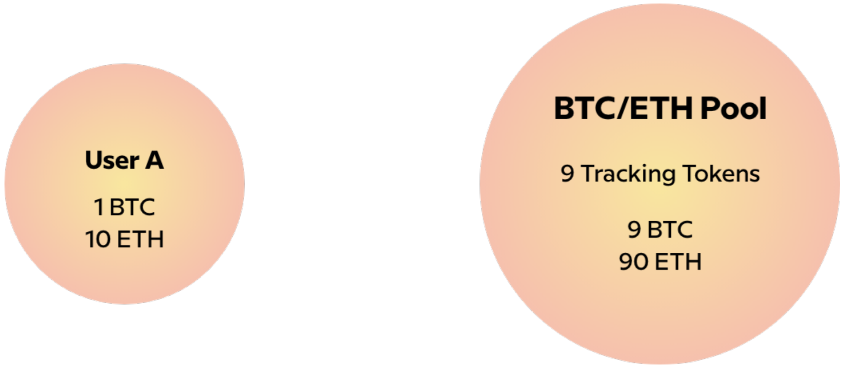
By contrast, one of the innovations brought by the decentralized finance space is the notion of an "automated market maker" (AMM). In such a model, the pricing and distribution of assets is satisfied by a mathematical formula or algorithm.

Initially, SundaeSwap will provide an automated market maker that is an adaptation of the model popularized by Uniswap. This section, then, will take some time to describe the Uniswap model for background. [1]

One useful analogy from classical finance to wrap your head around the notion of a liquidity pool is to think of it as an automatic ETF: A collection of securities traded and balanced against the market, of which you have a small ownership of.

In this model, liquidity providers deposit equal values of two assets in a smart contract, and receive tracking tokens representing their portion of the pool of assets. For example, suppose a BTC/ETH liquidity pool has 9 BTC and 90 ETH and has issued 9 total tracking tokens to previous liquidity providers.

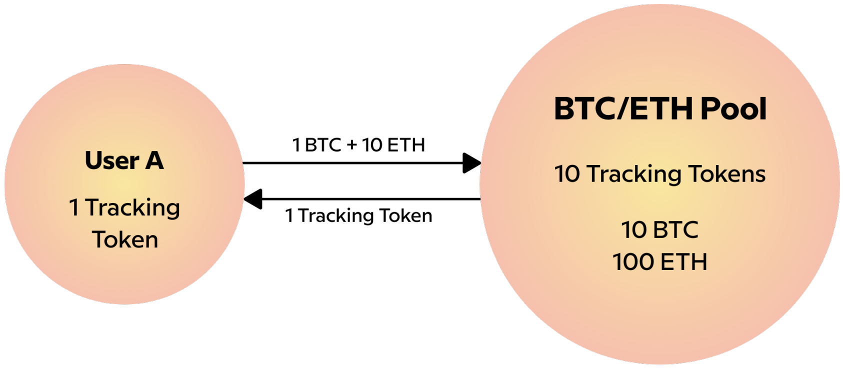
Figure 1: An example user and liquidity pool



If a user deposits 1 BTC and 10 ETH, since those represent 10% of the total value in the pool, the smart contract issues 1 new tracking token, for a total of 10. That 1 token out of 10 entitles the liquidity provider to 10% of the pool's total assets, which equals 1 BTC and 10 ETH, as expected.

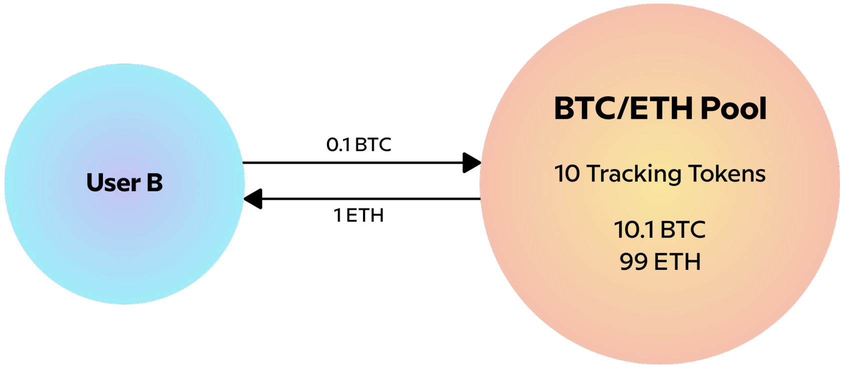
The pool also allows "swaps" to happen: someone deposits one asset, and

Figure 2: Depositing liquidity and receiving tracking tokens



receives the other, according to the exchange rate of the pool. In the above example, if I deposit 0.1 BTC, I might expect to withdraw 1 ETH.

Figure 3: An example swap of 0.1 BTC for 1 ETH

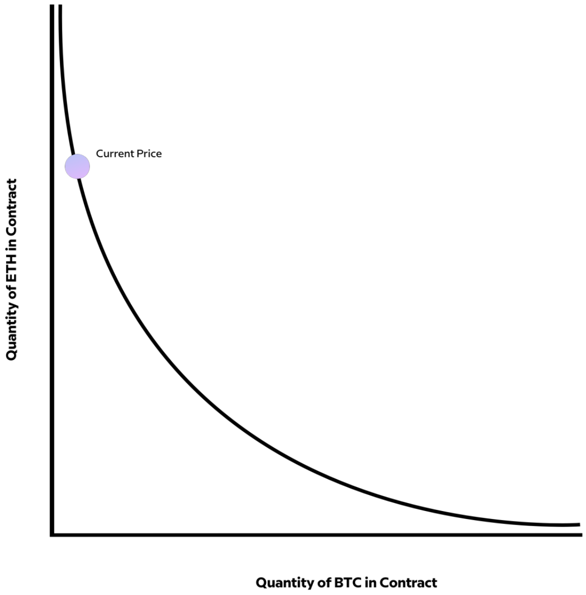


However, treating the pricing function as totally linear in this way opens the market up for abuse. As more and more of one asset is deposited, the ratio between them changes. Allowing a trade to execute at the *current* price puts larger trades at an advantage over smaller ones. Indeed, by depositing 10 BTC and withdrawing 100 ETH, a moderately sized trader could drive the price to infinity.

Instead, liquidity pools rely on a market maker function. In a “constant product pool," for example, the maximum amount of the other asset withdrawn is chosen such that the product of the two assets remains constant.

$$10\,\mathrm{BTC} \;\times\; 100\,\mathrm{ETH} = 1000\,\mathrm{BTC} \times \mathrm{ETH}$$

Figure 4: A constant product price curve and current price



So depositing 0.1 BTC would bring the total balance of BTC to 10.1, and we need to withdraw some ETH to bring the product back to the curve.

The amount of ETH left in the pool to maintain this constant must be:

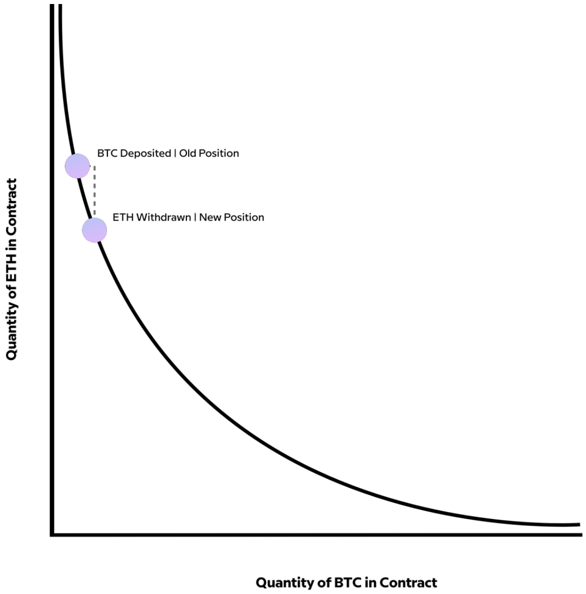
$$\frac{1000\,\mathrm{BTC} \times \mathrm{ETH}}{10.1\,\mathrm{BTC}} = 99.\overline{0099}\,\mathrm{ETH}$$

Meaning the user can withdraw up to

$$100\,\mathrm{ETH} - 99.\overline{0099}\,\mathrm{ETH} = 0.\overline{9900}\,\mathrm{ETH}$$

This introduces a measure of efficiency, known as "price slippage": $0.\overline{9900}\,\mathrm{ETH}$ is 99% of the value you would expect to get at the current price. If, instead, someone swapped 5 BTC, they would receive:

Figure 5: Constant product price curve after a swap



$$100\,\mathrm{ETH} - \frac{1000\,\mathrm{BTC} \times \mathrm{ETH}}{15\,\mathrm{BTC}} = 33.\overline{33}\,\mathrm{ETH}$$

$$\frac{33.\overline{33}\,\mathrm{ETH}}{15\,\mathrm{BTC} \times 10\,\mathrm{ETH}\,\mathrm{BTC}^{-1}} = 22.\overline{22}\%$$

This represents a staggering 22% efficiency.

For a given trade size, a smaller pool will have higher slippage, while a larger pool will have a lower slippage. Therefore it is beneficial to aggregate as much liquidity as possible to give participants the most efficient trades possible.

In order to incentivize liquidity into the smart contract, a further fee is taken off and left in the pool.

Because the tracking tokens held by the liquidity providers represent a percentage of the pool, when those tokens are burned, the liquidity provider withdraws an equivalent percentage of assets at the new price, plus a percentage of the fees that have been collected along the way.

There are further innovations on the Constant Product Pool model that we will discuss in Section 5.

3 Implementation on the Cardano Blockchain

The above model translates fairly naturally into Solidity and the Ethereum Virtual Machine: a small collection of functions and storage space that can be executed to deposit, withdraw, and swap liquidity, executed many times in sequence within a given block.

Cardano, however, uses a novel accounting and execution model, known as "Extended Unspent Transaction Outputs" (eUTXOs) [2], which makes translation of the above model not as straightforward as you might expect. The eUTXO model implements smart contracts more "passively" than an explicit function call, and heavily discourages the use of global state.

eUTXOs build on a simpler UTXO model. In the UTXO model, money is tracked as a chain of custody, by pointing at a specific (unspent) output from a previous transaction, which then gets consumed as the input to a new transaction. Very simple scripts can be attached and evaluated to determine if the output is allowed to be spent.

The model used by Cardano extends this in the following ways:

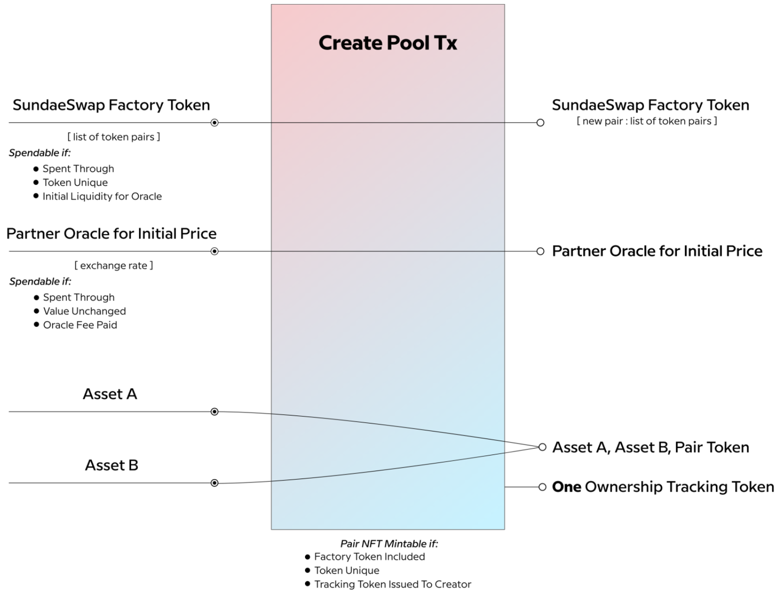
- The UTXO is equipped with an arbitrary datum
- The script locking the funds has access to input data, known as the "redeemer," as well as the entire transaction

We first present one natural way to translate the system described above to the eUTXO model, as well as a discussion of the trade-offs made along the way.

In this implementation, a global "SundaeSwap Pool Factory" unique token exists, locked via a script that allows the creation of specific "Asset Pair Liquidity Pool" unique tokens. The global token is used in conjunction with a minting policy to ensure that asset pairs stay unique, rather than diluting the available liquidity and suffering from poor slippage, as discussed above.

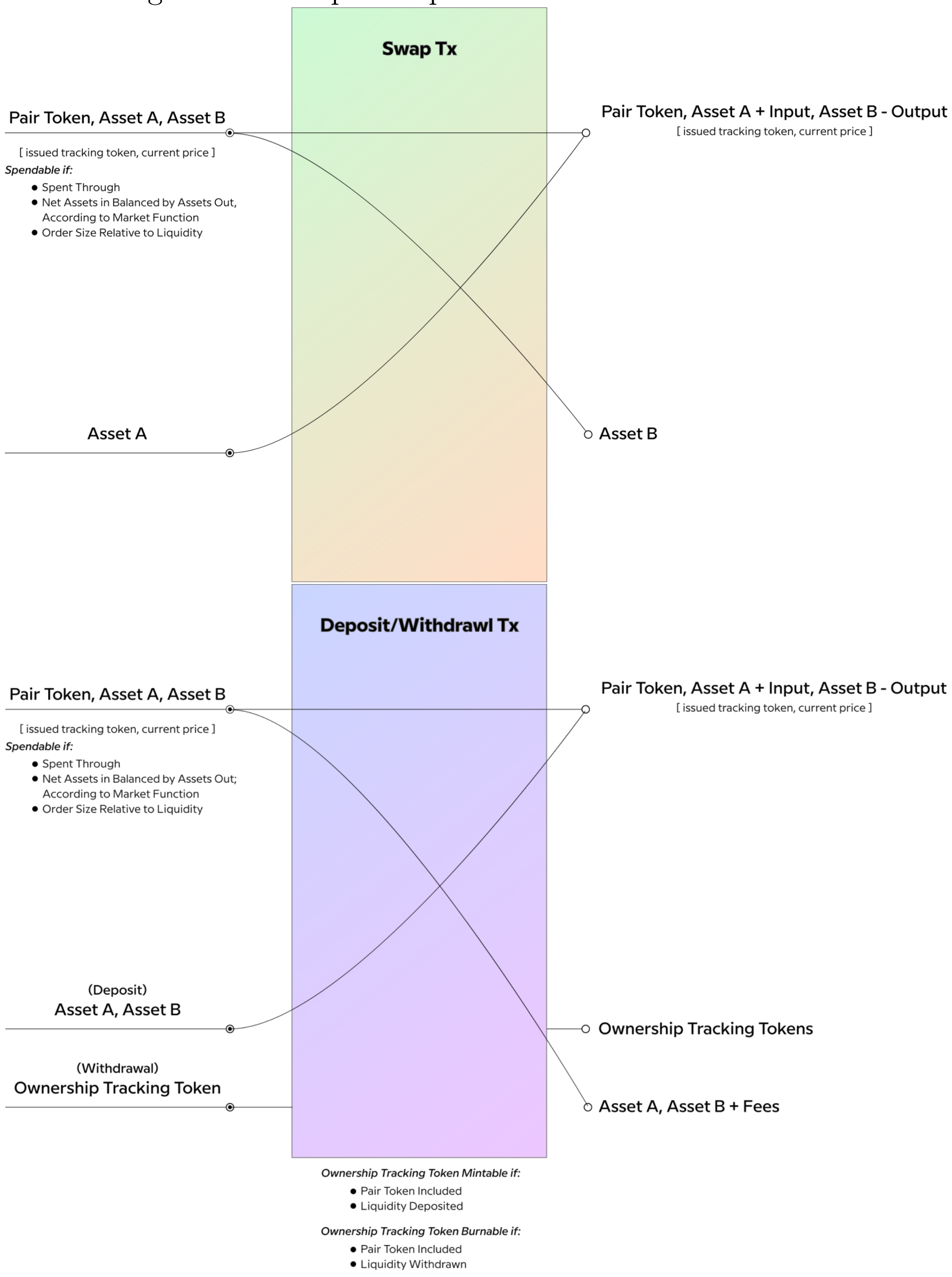
Then, these unique tokens are always locked in a eUTXO alongside the liquidity stored in the pool, using a validator script that enforces the constraints of the pool:

Figure 6: A transaction to create a new liquidity pool



- A minting policy allows tracking tokens to be minted so long as the appropriate liquidity is deposited
- The same minting policy allows tracking tokens to be burned so long as the appropriate liquidity is withdrawn
- The validator script allows swaps to occur, so long as they respect the pricing function and fee structure

Figure 7: Example swap and withdrawal transactions


Swap Tx
Pair Token, Asset A, Asset B
[issued tracking token, current price]
Spendable if:
● Spent Through
● Net Assets in Balanced by Assets Out, According to Market Function
● Order Size Relative to Liquidity
Asset A
Pair Token, Asset A + Input, Asset B - Output
[issued tracking token, current price]
Asset B
Deposit/Withdrawl Tx
Pair Token, Asset A, Asset B
[issued tracking token, current price]
Spendable if:
● Spent Through
● Net Assets in Balanced by Assets Out; According to Market Function
● Order Size Relative to Liquidity
(Deposit)
Asset A, Asset B
(Withdrawal)
Ownership Tracking Token
Pair Token, Asset A + Input, Asset B - Output
[issued tracking token, current price]
Ownership Tracking Tokens
Asset A, Asset B + Fees
Ownership Tracking Token Mintable if:
● Pair Token Included
● Liquidity Deposited
Ownership Tracking Token Burnable if:
● Pair Token Included
● Liquidity Withdrawn

This model, however, has a fatal flaw. Because any given eUTXO can only be spent once, as part of one transaction, it appears as if only one swap can happen per block. On the Cardano blockchain, there is roughly one block every 20 seconds. This would be abysmal throughput for a decentralized exchange.

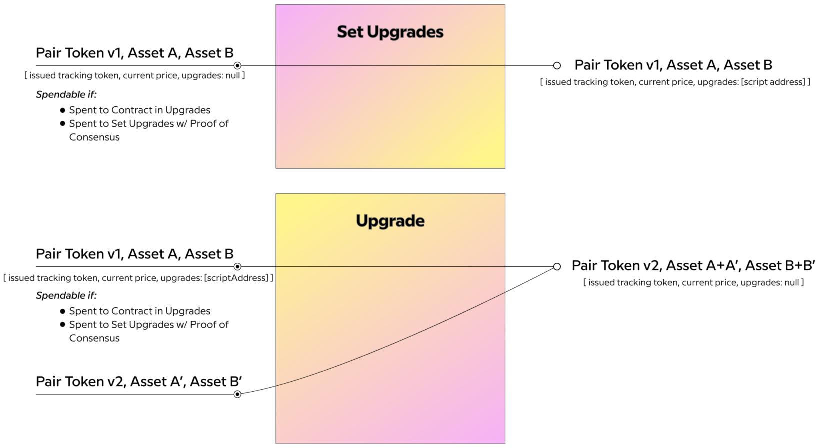
We will discuss the SundaeSwap scaling solution in a future whitepaper.

4 Planning for Upgrades

Given that the space is evolving quickly, it would be peak hubris to assume that the first model we implemented would be the best. Additionally, releasing a new protocol with improvements is at a disadvantage in that it has no initial liquidity, as liquidity is locked up in the previous version of the contract. It is important, therefore, to plan early for that upgrade path.

At SundaeSwap, we plan on enabling a seamless upgrade. To achieve this, the validator contracts holding liquidity allow it to be spent by a future version of the protocol. This list of future versions will initially be null, but through a vote of SUNDAE token holders, this value can be updated to point to a new contract. Once that value is updated, this locked liquidity can be spent directly into the new version of the protocol without user interaction.

Figure 8: An example upgrade transaction


Set Upgrades
Pair Token v1, Asset A, Asset B
[issued tracking token, current price, upgrades: null]
Spendable if:
Spent to Contract in Upgrades
Spent to Set Upgrades w/ Proof of Consensus
Pair Token v1, Asset A, Asset B
[issued tracking token, current price, upgrades: [script address]]
Upgrade
Pair Token v1, Asset A, Asset B
[issued tracking token, current price, upgrades: [scriptAddress]]
Spendable if:
Spent to Contract in Upgrades
Spent to Set Upgrades w/ Proof of Consensus
Pair Token v2, Asset A′, Asset B′
Pair Token v2, Asset A+A′, Asset B+B′
[issued tracking token, current price, upgrades: null]

This allows future versions of the protocol to be bootstrapped from day one with large amounts of liquidity, accelerating adoption of the improved protocol once community consensus has been achieved.

Obviously this is a sensitive attack surface, and a huge portion of our efforts will go towards securing this upgrade mechanism.

5 Future Work

There are a number of extensions to the protocol above that we are finalizing the details on, and which will be discussed in future whitepapers. For example:

- The role of the SUNDAE token
- A mechanism to increase throughput dramatically
- A mechanism to provide concentrated liquidity for more efficient market leverage
- A mechanism to provide secondary derivative markets
- A mechanism to further decentralize the role of the liquidity pool

6 Conclusion

The Cardano blockchain offers dramatic and exciting improvements in terms of thoughput, fees, energy efficiency. With the launch of Smart Contracts later this year, it will be poised for a huge surge in economic activity and utility. As more and more native tokens are created to track and satisfy real world value, there will be an incredible need for markets to trade and acquire these tokens.

The above model provides a simple, scalable solution to meet these early needs, well suited for the Cardano blockchain, and positions SundaeSwap to expand into more efficient and more sophisticated protocols and instruments in the future.

References

[1] Hayden Adams, Noah Zinsmeister, and Dan Ribinson. Uniswap v2 Core. Technical report, Uniswap, 03 2020.

[2] Manuel M.T. Chakravarty, James Chapman, Kenneth MacKenzie, Orestis Melkonian, Michael Peyton Jones, and Philip Wadler. The Extended UTXO Model. Technical report, IOHK and University of Edinburgh, 01 2020.

[3] Steven Ehrlich and Charles Hoskinson. Cardano and ethereum founder analyzes the newest evolutions in crypto and blockchain technology, Apr 2021.