Sigmaswap Protocol

The Whitepaper

October 2022

Sigmaswap Team
hello@sigmaswap.org

KEYWORDS

Cryptocurrency
DAO
Decentralized Exchange
Decentralized Finance
Hedera Hashgraph
HTS
SIGMA

ABSTRACT

Sigmaswap protocol is an open-source automated market maker and decentralized trading platform operated on the Hedera hashgraph. The protocol offers non-custodial all-in-one cryptocurrency services to all HTS tokens. The protocol is implemented by uncontrollable and immutable smart contracts from Hedera Smart Contract Service (HSCS). The logic and algorithm design are derived from the Uniswap V2 on the Ethereum blockchain, plus modifications made to optimize the hashgraph technology.

BACKGROUND

Decentralized Finance (DeFi) was recently in exponential growth in both terms of system development and adoption among crypto users. The Distributed Ledger Technology (DLT) that the vast majority of DeFi projects rely on is Blockchain. With the decentralization, security, and immutability it offers, developers are attracted to begin building decentralized products on the blockchain they see suitable. However, technology does not stop evolving. Hedera, a DLT organization, came with a new technology that has the potential to disrupt existing blockchains called Hashgraph. It allows smart contract developers to build decentralized applications at the same level of convenience as existing blockchains but with higher security and network efficiency [6]. Hence, Sigmaswap is built on the Hedera hashgraph.

I. INTRODUCTION

The DeFi industry utilizes Automated Market Makers[1] (AMMs) rather than the order book system used in many cryptocurrency exchanges. AMMs provide users with the ability to trade tokens, add liquidity, and earn from trading fees. In short, an AMM requires more participants to maintain its operational efficiency, and everyone in the ecosystem is incentivized by different reward types to contribute to the protocol.

Sigmaswap is a smart contract implementation based on the formula $x \cdot y = k$; this is called a "constant product formula" [1]. As mentioned, the logic is retrieved from the Uniswap V2. The reasons behind this decision are as follows:

I. The Uniswap V2 has proven itself over time in its security, and the model offers the feature for users to swap token A for token B in one or multiple hops. It allows traders to make trades without concerns about trading orders and matching.
II. All *Pairs*[2] can serve as an oracle that accumulates the relative price of the two assets within a pair, and this data can be used by external parties for price determination of HTS cryptocurrencies [2].
III. It enables "flash swaps" where advanced users can make an arbitrage to maintain the price of any asset in different exchanges to an appropriate level.
IV. The system can be used efficiently without a need to access web application interfaces for advanced users and without a need to connect with web application interfaces through direct smart contract calling.
V. The model of liquidity pairing is widely used by many protocols and accepted by global developers. Liquidity can be utilized by other protocols, which lead to integrations and mass adoption.

In this technical whitepaper, the Sigmaswap team provides explanations of the algorithm design in the Sigmaswap protocol. It covers all the contracts' features and details– including mathematics, logic, and decisions. Additionally, it clarifies all modifications made to optimize the performance and fit the network structure of the Hedera hashgraph.

II. HEDERA INTEGRATIONS

This section presents the modifications of smart contract implementation to suit the Hedera network.

A. Hedera Token Service (HTS)

In the original Uniswap V2 code, the fungible token type compatible with its operation is ERC20. The token itself is a smart contract created by any individual. Once a user interacts

[1] AMM is an algorithm that automates the market maker system; an agent can trade with the market maker by selecting a quantity of an asset to trade at a price specified by the AMM.
[2] *Pairs* are liquidity tokens that represent the liquidity of two assets, generated by smart contracts.

with a token, its contract changes the state variable. This leads to a slow process of transaction completion and high gas fees. Hedera, a hashgraph DLT, solves the issues by offering developers a set of pre-compiled token contracts named Hedera Token Service (HTS). It includes all types of tokens, features, and configurations for developers to make token creation easily manageable. Also, the security of the token is at the highest level possible, provided and audited by the network [4].

Smart contracts, in the case of Sigmaswap, are only used to control token supply according to the features offered by the protocol, and none of Sigmaswap's contracts can be mutated after deployment. In advance, the smart contracts hold the Admin keys and Supply keys. Hence, they cannot be modified, fixed, or upgraded by any human. The HTS standard offers more configuration options for token creation compared to ERC-20, including Admin Key, Supply Key, Pause/Freeze Key, etc. As these additional properties[3] could lead to token manipulation in some cases, users need to understand all the properties of the token they are interacting with.

Unlike the Smart Contract Service[4] (HCSC), HTS is built in the native layer of the network. Thus, there are advantages in high transaction speed and low gas fee consumption. In addition, all HTS tokens are logically coded to have a requirement before the first interaction with any account or contract called "association" [5, 8]. The operation[5] behaves similarly to a shield protecting users from interacting with undesirable tokens. For all the mentioned reasons, Sigmaswap supports integrations to all HTS tokens.

B. Smart Contract Rent

Hedera has the policy to collect fees on data storage, including smart contracts. Fees are collected in HBAR and denominated in USD [13]. Correspondingly, all contracts of Sigmaswap contain a `receive()` function intentionally to obtain HBAR from external accounts or/and contracts. Plus, all contracts charge a fraction of HBAR on top of the network gas fees for all transactions. These adjustments guarantee that the protocol has sufficient HBAR to pay for contract rents. Also, it protects the protocol from contract creation spamming.

C. Burn Address

On other blockchains, token burn is typically done by transferring to the burn address/ null address/ address zero/ or 0x000…000. A justification is that this method costs fewer gas fees. However, at the time of writing, there is no burn address on the Hedera. An HTS token has to be burnt by a contract or an account that holds its supply key, as mentioned above. Sigmaswap uses immutable, decentralized smart contracts as key holders to burn all HTS tokens within the Sigmaswap Protocol and its ecosystem. For example, the factory holds the keys to control all liquidity pairs' supply. Once a user unpairs the token, the factory will burn the liquidity token, and he/she will receive the two deposited HTS tokens.

D. Wrapped Native Hedera Token (WHBAR)

The concept is the same as in all other blockchains. The native token of the network, HBAR, cannot fully interoperate with a smart contract by nature. A common solution is to wrap the token into another more compatible form. In the case of Hedera, HBAR is wrapped into HTS versioned HBAR or HBAR[HTS] or WHBAR.[6] The wrapping process authenticates that the amount of minted WHBAR is correlated to the amount of native HBAR.

III. FEATURES

This section includes Sigmaswap's services at its first launch; other than that can be found in separate papers.

A. HTS Decentralized Exchange

Sigmaswap uses an AMM model that utilizes the constant product formula as such:

$$x_\alpha y_\beta = k$$

where x and y represent the number of tokens in the pool reserve (token α and token β in the Pair(α, β), and k is the constant product invariant. During a trade, the number of x and y will change, and it will result in a new, greater number of k due to trading fees. Sigmaswap charges 0.25% trading fees for all trading transactions. These fees are used to incentivize liquidity providers and others to maintain the protocol. After the fee application, the new k will be satisfied as:

$$(x_\alpha - \Delta_\alpha)\left(y_\beta(1-0.25)\Delta_\beta\right) = k$$

The above Δ_α indicates the number of the trade-out token and Δ_β indicates the number of the trade-in token. For the best trading experience, the system allows users to create pairs and provide liquidity for any two HTS tokens without permission [15].

B. Time-Weighted Average Price Oracle

All liquidity pairs in the Sigmaswap protocol can serve as a price oracle to all parties on the Hedera network. The marginal average price at t time is computed by the sum of the token prices over the interval t divided by the number of intervals.

$$a_{(t)} = \sum_{i=1}^{t} P_i$$

Since the price data in the Sigmaswap is calculated by only trades in the protocol. Any asset price can be different from other exchanges. In reality, this situation is managed by arbitrageurs. The price will recover to an appropriate range in almost immediate time because it is an opportunity to create profits in a very short time [2]. Sigmaswap computes and records the price change before the first trade of each timestamp.

[3] In the case that these properties are not defined, there is not an account or a smart contract that can control the asset after deployment-including supply, token restriction, and any further modification.
[4] HCSC is a service of Ethereum Virtual Machine (EVM) that allow developers to write smart contracts on Hedera [8].
[5] An association is a transaction that costs a gas fee and requires a signature(s) to approve. Users may disassociate any token once completed the interactions for security purposes.
[6] WHBAR contract that is used by Sigmaswap is at address 0.0.1062663.

On the Ethereum blockchain[7], price and transaction manipulation are serious concerns since transaction ordering is controlled by nodes (miners) [11]. Nonetheless, such issues are prevented by the fair ordering of Hedera. "*Hashgraph elects no leader and therefore removes the possibility of tampering with fair ordering. The place of a transaction within the consensus order is calculated by the nodes as a network, not assigned by a leader. It reflects the order in which the transaction was received by all the nodes.*" [12]. Thus, the time-weighted average price oracle plays a perfect fit to the Hedera hashgraph.

To estimate the time-weighted average price in the range between time *t1* and *t2*, it is necessary to compute the average price in *t1* and *t2*, and then make the endpoint subtract the starting point as follows:

$$P_{(t_1,t_2)} = \left[\frac{a_{(t_2)} - a_{(t_1)}}{t_2 - t_1}\right] = \left[\frac{\sum_{i=1}^{t2} P_i - \sum_{i=1}^{t1} P_i}{t_2 - t_1}\right]$$

In order to verify an updated price, the Sigmaswap contract caches its reserves after each contract interaction and uses them to update the price oracle.

C. Flash Swaps

The feature is a fork of Uniswap v2 that helps users to make trades without having to put down any capital. To clarify, there is a `callback()` function incorporated with the `swap` to let users use any amount of token reserves to make trades. As long as they repay the exact amount of the token plus trading fees, the transaction can be completed. This allows arbitrageurs to profit from price differences on various exchanges without the need for base capital.

D. Liquidity Mining

In addition to earning trading fees, liquidity providers can participate in liquidity mining to earn additional rewards offered by the Sigmaswap protocol. The rewards are classified into two sources: decentralized rewards and SIGMA rewards.

In the decentralized reward system[8], anyone can add any type of reward token to a designated pool. The configuration for time and amount of release can be set by the reward providers. There are no restrictions on rewarding any HTS fungible token and the amount.

For SIGMA rewards, the SIGMA token, which is the governance and utility token of the Sigmaswap protocol, is used to incentivize liquidity providers in the pool that pass the governance criteria of the Sigmaswap. A total of 58% of the total SIGMA supply will be distributed to liquidity mining.

TABLE 1: SIGMA emission schedule in liquidity mining.

MONTH	EMISSION/SECOND	EMITTED SIGMA
1	0.7	1,814,400
2	0.7	3,628,800
3	0.7	5,443,200
4	0.7	7,257,600
5	0.7	9,072,000
…	…	…
…	…	…
…	…	…
31	0.7	56,246,400
32	0.7	58,000,000
TOTAL		58,000,000

IV. GOVERNANCE

A. SIGMA

Fig. 1: SIGMA token logo

The Sigmaswap token or SIGMA is the governance and utility HTS token powering the Sigmaswap protocol and its ecosystem. Each SIGMA holder is the owner of the Sigmaswap protocol according to their shares. The token is used for all services and events related to the protocol. Its total supply is 100,000,000 tokens.

The token's use cases at the time of writing are swapping, providing liquidity, liquidity mining, staking, and governance voting. In future developments, the number will be extended as new features are launched. SIGMA allocation is shown in Fig.2.

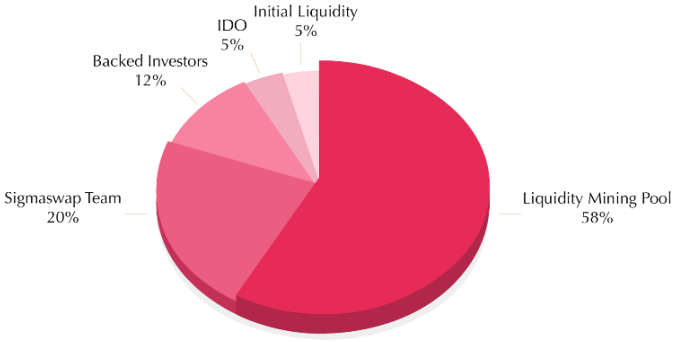

Fig. 2: SIGMA allocation

SIGMA is created as a deflationary HTS token. This means the more SIGMA is used, the more SIGMA will be eliminated from the circulation and total supply. Sigmaswap's deflationary mechanisms include:

I. **Burn**: tokens are burnt to be reminted as rewards in liquidity mining and the SIGMA staking pool.

[7] Ethereum is the first blockchain that offers full compatibility with Solidity smart contracts.
[8] Since all individuals are free to reward the pools, it is crucial for users to authenticate the tokens before making claims.

II. **Unopenable Lock**: a contract that locks all received tokens forever. It can be used as a burn address for any party.
III. **Treasury**: the treasury stores tokens for Sigmaswap DAO. The fund will be used for further development decided by the Sigmaswap governance.

B. Consensus Voting

As of being a SIGMA token holder, he/she has the power to direct the protocol. All the changes in Sigmaswap are committed through consensus votes. This governing strategy makes the Sigmaswap a Decentralized Autonomous Organization (DAO) [3]. Users have to take multiple steps to complete the process. Since the resources and participation guides can be extended at any time in the future, voting processes and channels will be updated periodically [14].

V. OTHERS

A. Fees & Fee Setters

Fees in Sigmaswap are collected in all services, and they are used to fulfill the ecosystem. For the swaps, trades are charged 0.25% denominated in the input token. The funds are allocated as follows;

- 0.17% to the liquidity providers.
- 0.045% to the SIGMA staking pool.
- 0.01% for SIGMA deflationary methods.
- 0.025% as a protocol fee.

Fees in Liquidity Mining and SIGMA Staking;

- 1% of the liquidity mining rewards distributed is collected at claims.
- 2% of the SIGMA rewards are collected at claims.

The purpose of accumulation is to deflate the SIGMA token. For more information, please visit Fees at https://docs.sigmaswap.org/protocol/advanced/fees.

All fee collections are sent to the `feeTo` contract which is managed by the account `feeToSetter`.[9] The same account also has the privilege to adjust the rent fees in all smart contracts to the maximum of 1 HBAR. Additionally, deployment addresses and account details can be found on the official Sigmaswap Documentation [14].

B. Liquidity Pair Initialization

Pair initialization requires more computational steps than supplying liquidity to an existing pair. The first depositor sets the paired token price, which will be changed once others join the pool. It means that the value of each liquidity pool is dependent on the ratio at which liquidity is deposited in the first place. This pricing mechanism allows traders and arbitrageurs to participate in liquidity contributions since there is no guarantee that the ratio reflects the true price. The formula for calculating any initially minted pair is the geometric mean;

$$Share_{minted} = \sqrt{x_{deposited} \cdot y_{deposited}}$$

The formula ensures that a liquidity share in the pool is always equal to or slightly greater than the geometric mean in the reserve in that pool.

The number of bits in the quantity of shares is the mean of the number of bits in the quantity of assets x and y in the reserves, which could be checked by:

$$\log_2 \sqrt{x \cdot y} = \frac{\log_2 x + \log_2 y}{2}$$

C. Maximum Token Balance

Sigmaswap only supports reserve balances of up to $2^{112} - 1$. The maximum value is over 1 quadrillion; therefore, it is enough for all assets in the network to be traded on the protocol. According to the Uniswap V2, the justification of this number is to maximize the price oracle's efficiency [1]. Even if the transaction may fail when a reserve balance goes above the limit, users can call the `skim( )` function to remove excess assets from the pool reserve.

D. Error Handling

For the best trading experience, it is essential to cover all the errors that possibly occur when breaking the rules in smart contracts. The following error handling list indicates different issues during a trade.

- EXPIRED: a user exceeds the transaction deadline.
- INSUFFICIENT_LIQUIDITY: there is not enough liquidity in the pool.
- INSUFFICIENT_INPUT_AMOUNT: there is not enough input token amount.
- INSUFFICIENT_OUTPUT_AMOUNT: there is not enough output token amount.
- INVALID_TO: the receiver address is not an account.
- K: the new *k* is less than the old *k*.

D. SIGMA Programs

Sigmaswap has several programs in place to drive value for SIGMA tokens. As mentioned previously, SIGMA is used in all aspects of the Sigmaswap protocol. On top of this, SIGMA will be used for integrations and collaborations with all external organizations that desire to join the ecosystem. Updates on these programs will be regularly announced on Sigmaswap's media channels.

VI. RESULTS

Users have numerous options to make their HTS tokens in use on the Sigmaswap protocol. Due to the decentralization offered, trust and middlemen are eliminated from all financial statements, meaning that users can be sure that their transactions will be executed as they expect. It also appears that this smart contract application helps create a dependent environment where each person's action has an impact on others. For example, when there is a large trade in a pool, and the price of the trade-out token increases, another trader may see that there is a price difference for the exact token in another exchange. The person could then arbitrage, making the price

[9] `feeToSetter` has the privilege to adjust the destination of collected fees. It is used when there is a feature update or contract migration.

return to normal. From this situation, the first trader would receive the desired currency, the arbitrageur would make profits, the liquidity providers would have capital gains from trading fees, and the protocol would sustain the token price. In summary, it is a win-win situation.

Liquidity providers can have extra earnings by adding their liquidity tokens into a liquidity mining pool. On top of the SIGMA rewards, any person or protocol that wishes to bootstrap liquidity of their tokens on Sigmaswap is free to fill the pool with any HTS fungible tokens as incentives for liquidity providers.

For SIGMA token holders, certain privileges are preserved. In most Dapps, a utility token is only used to govern the platform, and its value is out of focus. Nonetheless, SIGMA tokens are designed to be an asset that represents the protocol. Its maximum supply is set and cannot be altered by any account or contract. All the features in Sigmaswap systematically help decrease the token circulation and supply. For clarity, the more use of any service in Sigmaswap, the higher the deflation rate in SIGMA.

For other protocols, the algorithms and resources of Sigmaswap are accessible and can be used by all.[10] This availability could lead to uncountable applications and utilizations. For example, a lending protocol can utilize flash swaps to offer users a flash loan service. A DEX aggregator can connect the router to be able to serve their clients the best exchange rate. A newly built DeFi protocol can kickstart their liquidity volume by participating in Sigmaswap's liquidity provision instead of building their liquidity pool and also be able to incentivize users with Sigmaswap decentralized reward. Finally, the Sigmaswap protocol can be a connection hub where all DeFi projects meet, share, and ultimately develop the ecosystem together as a network.

VII. Limitations and Risks

Although every service in Sigmaswap is efficient and fair to all individuals, there are some limitations by its nature.

A. Exchange

There are two risks that a trader on Sigmaswap may face.

I. **Slippage**: Slippage indicates the difference between the expected transaction price and the actual transaction price. It correlates with the liquidity density in each pool. The more liquid the liquidity reserve, the better the trading depth and the lower the slippage. Once a transaction occurs, resulting in a change in the reserve of an asset in the pool, the actual transaction execution price of the asset will change which causes a slippage [10]. Therefore, the larger the transaction volume and the deeper the destruction of the liquidity reserve of the capital pool, the higher the slippage. Slippage calculation is done by subtracting the expected token price from the limit or actual token price in a trade.[11]

$$P_{limit} - P_{expected}$$

In practice, users who perform a large trade in a low-liquidity pool could receive fewer output tokens than expected since their transactions could significantly bring up the price; in other words, they buy the asset at a higher price. To compute the slippage percentage,

$$S_{(\%)} = \left[\frac{\left(P_{limit} - P_{expected}\right)}{S_{(USD)}}\right] \cdot 100$$

II. **Price Impact**: In simple terms, price impact is the difference in a token's price before and after each trade. For example, if a trader makes a trade that causes the price of a token to increase by 5%, the trader has created a 5% price impact to the pool. In a more complex situation, there may be cases of transaction ordering, where two people swap the same token with the same currency input and amount. If the person who submits the transaction a fraction of a second faster and gets it finalized first receives more tokens, and if his or her trade created a price impact, the second person who gets the transaction approved and finalized shortly after would be affected by the price impact, resulting in having less output token compared to the first trader.

III. **Multiple hopping**: Even though most of the trades can be accomplished by swapping in a pool, there is a case where users may need to make more than one trade in different pools to receive the output token. It occurs in exotic pairs or new tokens that have low liquidity depth. Multiple hopping helps traders achieve these tokens within a single transaction.

B. Liquidity Provision

All liquidity suppliers may experience the following issues:

I. **Impermanent Loss (IL)**: It is a temporary loss of funds occurring when providing liquidity. IL takes effect once there is a change in the asset ratio of a liquidity token. The IL begins into effect once a user provides liquidity almost immediately as all asset prices are always moving in reality. It is called impermanent since there are always possibilities that prices can return to the entry level. The impermanent loss is not in effect for a user if the current asset is priced the same as his/her entry price [9, 10]. The following formula shows how to calculate IL.

$$IL_{(k)} = \left(\frac{2\sqrt{k}}{k+1}\right) - 1$$

II. **Liquidity Fragmentation**: Liquidity fragmentation is one of the limitations from a liquidity provision design. According to VII. A. III. Multiple hopping, a user needs to pay fees twice or more depending on the liquidity availability of their selected token. To produce maximum trading efficiency in this model, liquidity density should be the main focus for all traders.

[10] View code repositories at https://github.com/SigmaSwap

[11] Users may need to specify slippage tolerance before making a trade. Otherwise, the default rate of 0.5% will be automatically applied.

However, this liquidity fragmentation helps limit the risk of impermanent loss to only a pair of tokens, compared to other services that use 3-token pairs or even more. Also, all the risks of price fluctuation in an asset are limited to its pair(s).

C. Liquidity Mining

There is a situation that could be a serious concern for all participants. It is a problem of reward filtration. As mentioned in section III. FEATURES, there is no restriction in reward provision. Any person or organization can incentivize liquidity providers with their deposited tokens, and it could lead to a lack of token verification. Sigmaswap solves this problem by adding a token address to all reward tokens. Users can see and verify the addresses before making claims. Second, if a user does not want to claim any specific token, he/she can claim one by one rather than use *claim all.* This is called claiming on-demand.

VIII. Protocol Priorities

The Sigmaswap protocol is built under 3 principles as follows:

I. **Security**: This is the most important focus of Sigmaswap. In the industry of finance, security is a critical concern and should be taken seriously. All smart contracts and codes in Sigmaswap are carefully reviewed by the team and community and audited by external reliable organizations before deployment.

II. **Transparency**: All resources and processes are publicly available to all. The entire protocol is developed under the MIT license which implies that it is fully open-source and can be used for any purpose by everyone.

III. **Verifiability**: All individuals have access to verify all parts of Sigmaswap. Users are encouraged to authenticate the product before use.

IX. Integration & Contribution Allowance

The protocol aims to be a community-based project joined and developed by global enthusiasts. Any individual can be a part of Sigmaswap through contribution. Likewise, any organization can connect and build projects on top of the Sigmaswap protocol. However, if the development requires or causes significant changes in the core system, the process and decision must be evaluated and approved by the Sigmaswap governance.

X. Conclusions and Future Developments

The development of Sigmaswap includes logical and mathematical experiments established and proven since the Uniswap V2. The results indicate that it can serve users with an automated, decentralized finance protocol. Liquidity is ranked as the heart of the system, freely used by all users as they see appropriate. The protocol rewards all ecosystem participants; in return, everyone involved in any Sigmaswap activity helps power the entire system directly and indirectly. In future work, more features will be created to offer users end-to-end cryptocurrency services on Hedera. Ultimately, Sigmaswap may implement new versions of code and migrate to the new ones when more efficient metrics are discovered.[12]

Disclaimers

This paper is for general information purposes only. It does not constitute any financial advice or/and a recommendation or solicitation to make any investment and should not be used in the evaluation of the metrics of making any investment decision. It should not be relied upon for accounting, legal, or tax advice or financial guides. This paper reflects the current opinions of the authors and is not made on behalf of the Sigmaswap Labs or/and their affiliates and does not necessarily reflect the opinions of the Sigmaswap Labs, their affiliates, and individuals associated with them. The information herein is subject to change without being updated.

Acknowledgements

This paper is a part of the Sigmaswap protocol. This work has been created in the first version of the Sigmaswap protocol by the Sigmaswap Labs and the team. They do not provide, own, or control the protocol, which is run by the smart contracts deployed on the Hedera hashgraph. Upgrades and modifications are managed in a community-driven way by holders of the SIGMA token. All single components are developed and published under the MIT license. All individuals are free to use all assets and resources from the Sigmaswap protocol for their own benefit. This paper does not include all components and complete guides to build a Dapp on Hedera, readers may find this information in Hedera documentation [7].

References

[1] Adams, H., Zinsmeister, N., & Robinson, D. (2020, March). Uniswap v2 Core. Retrieved October 3, 2022, from https://uniswap.org/whitepaper.pdf

[2] Angeris, G., & Chitra, T. (2020, October 1). *Improved price oracles: Proceedings of the 2nd ACM Conference on advances in Financial Technologies*. ACM Conferences. Retrieved October 8, 2022, from https://dl.acm.org/doi/abs/10.1145/3419614.342321

[3] Chohan, Usman W. (2017, December 4). *The Decentralized Autonomous Organization and Governance Issues*. SSRN. Retrieved October 3, 2022, from https://papers.ssrn.com/sol3/papers.cfm?abstract_id=3082055

[4] Ferrin, D. (2021, November 4). *HIP-206: Hedera Token Service Precompiled Contract for Hedera SmartContract Service*. Hedera Improvement Proposals. Retrieved October 5, 2022, from https://hips.hedera.com/hip/hip-206

[12] For the latest updates, see all media channels at https://docs.sigmaswap.org/Ecosystem/official-links.

[5] Ferrin, D. (2021, November 12). *HIP-218 : Smart Contract interactions with Hedera Token Accounts*. Hedera Improvement Proposals. Retrieved October 14, 2022, from https://hips.hedera.com/hip/hip-218

[6] Hedera Improvement Proposals. Retrieved October 14, 2022, from https://hips.hedera.com/hip/hip-218

[7] Hedera Documentation. (n.d.). Hedera. Retrieved October 3, 2022, from https://docs.hedera.com/guides

[8] *Hedera Hashgraph Announces Mainnet Launch of EVM-Compatible Smart Contracts 2.0*. Hedera. (2022, February 3). Retrieved October 5, 2022, from https://hedera.com/blog/hedera-hashgraph-announces-mainnet-launch-of-evm-compatible-smart-contracts-2-0

[9] Karaivanov, D. (2021, May 3). *Impermanent loss explained with Examples & Math*. The Chain Bulletin Retrieved October 8, 2022, from https://chainbulletin.com/impermanent-loss-explained-with-examples-math

[10] Labadie, M. (2022, April 4). *Impermanent loss and slippage in Automated Market Makers (AMMs) with constant-product formula*. SSRN. Retrieved October 3, 2022, from https://papers.ssrn.com/sol3/papers.cfm?abstract_id=4053924

[11] Mackinga, T., Nadahalli, T., & Wattenhofer, R. (2022, April 12). *TWAP Oracle Attacks: Easier Done than Said?* Cryptology ePrint Archive. Retrieved October 8, 2022, from https://eprint.iacr.org/2022/445

[12] Madsen, P. (2019, September 13). *Fair Access v. Fair Ordering, & Transactions Per Second Examined & Explained*. Hedera. Retrieved October 14, 2022, from https://hedera.com/blog/fair-access-v-fair-ordering-and-transactions-per-second-examined-and-explained

[13] Marquez, E. (2022, July 29). *Smart Contract Rent on Hedera is Coming: What You Need to Know*. Hedera. Retrieved October 5, 2022, from https://hedera.com/blog/smart-contract-rent-on-hedera-is-comingwhat-you-need-to-know

[14] Sigmaswap Documentation (n.d.). *Technical Documentation For The Sigmaswap Protocol.* Retrieved Oct 3, 2022, from https://docs.sigmaswap.org/

[15] Zhang, Y., Chen, X., & Park, D. (2018, October 24). *Formal Specification of Constant Product (x * y = k) Market Maker Model and Implementation*. Github. Retrieved October 6, 2022, from https://github.com/runtimeverification/verified-smart-contracts/blob/uniswap/uniswap/x-y-k.pdf