Mettalex Overview

Date: 2020-10-19
Author: Felix Nicolae Bucsa

Summary

Mettalex is the first decentralised derivatives exchange (i.e. **DEX**) powered by **Fetch.ai** technology. The main objective is to provide access to risk management tools to a large audience of physical commodity traders. Nowaday, users can access risk management tools via financial market intermediaries who charge high spreads and have limited liquidity. Mettalex enables market participants (e.g. Crypto Mining companies or steel mills) to gain economic exposure to the spot price of commodities and other assets thus allowing them to open hedging or leveraged positions on the market.

This revolutionary network creates the base for efficient and intelligent market making decisions thanks to an Autonomous Market Maker (i.e. AMM). Each smart contract is associated with a pair of specially designed tokens (i.e. **Position tokens**). The smart contract acts as a decentralized clearing house so it facilitates the matching between buyers and sellers on the market. In fact, traders can exit or open a position whenever they desire without the need to find a counterparty interested in the trade.

System Architecture

It is possible to distinguish different elements that represent the specific components of the Mettalex DEX architecture. Table 1 depicts the following **components**:

- *On chain smart contracts* for collateral, vaults, autonomous market makers, and Position tokens;
- *Asset reference price feed* from an index or datafeed provider;
- *Cloud components* controlled by Mettalex;
- *Web3 connection* (e.g. Infura);
- *Web3 wallets* (e.g. Metamask, Fireblocks, Hardware wallets);
- *JavaScript single page apps* to interact with smart contracts.

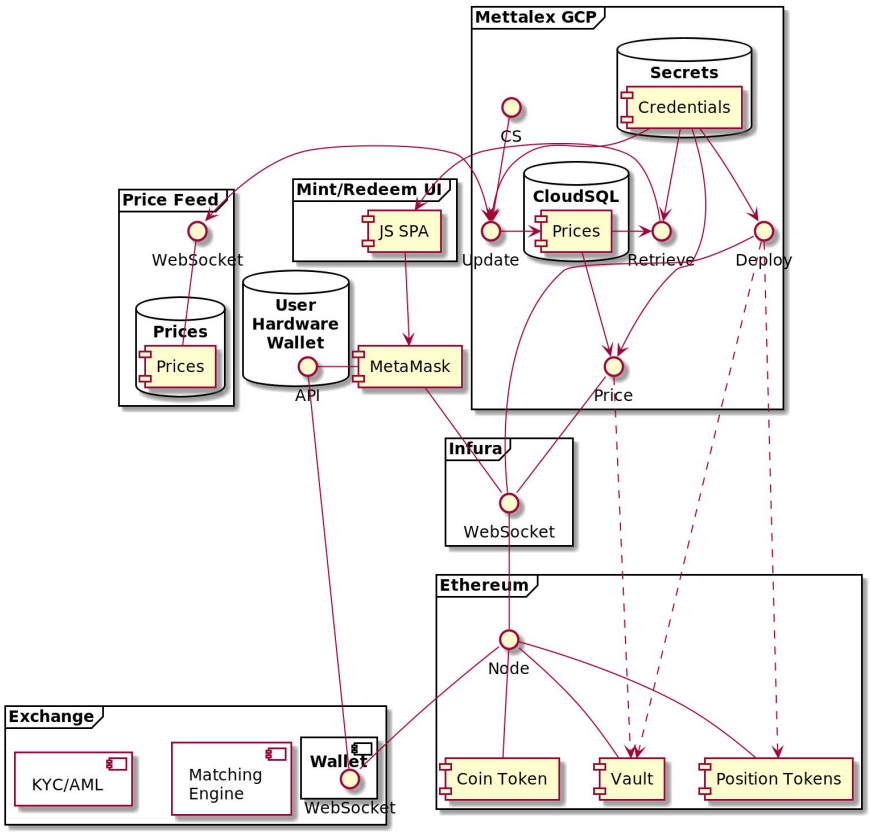

Table 1: MTLX system architecture.

Economic model behind Mettalex DEX

Mettalex overall structure is composed of four different **layers**:

- *Tokenisation* Layer
- *Exchange* Layer
- *Liquidity Provision* Layer
- *Governance* Layer

Inside the Mettalex platform an important role is played by Position tokens. As mentioned, these tokens are minted in pairs by the Mettalex smart contract by locking up collateral into it (i.e. ERC-20 stablecoin such as USDT, BUSD, USDC, DAI). This process takes place on the **Tokenisation Layer**. In the latter, the **Mettalex Vault** is fundamental as it contains the collateral that backs the pairs of L and S Position tokens.

The **Mettalex Exchange Layer** provides a reference price feed from an index or data-feed provider. The fairness of the information is guaranteed by the multiple sources providing different data (e.g. QUANDL, Chainlink, REFINITIV) and more data feeds will be added in the near future (e.g. Platts and Fetch.ai agents) thus providing a wide variety of possible oracles from which gather data. These multiple sources allow the communication between Mettalex and the real world. The price data gathered is thus processed by the *Mettalex CloudSQL* component in order to allow the implementation of new markets, the recovery of price feeds and to ensure the continuous updating of prices displayed to on-chain and off-chain components. Traders are presented with reliable indices providing valuation for the commodities listed which are denominated in either fiat currency or stablecoin. On this layer, Position tokens are used to track the change in the price of an underlying asset and users are free to open positions on the market. Position tokens can be divided into **Long tokens** (i.e. **L tokens**) and **Short tokens** (i.e. **S tokens**). These respectively track the positive and negative change in a given price. Traders can buy or sell using stablecoin collateral by trading with other participants or an Mettalex AMM. In this case, the latter refers to the **AMM pool** which contains a mix of USDT, L and S tokens that can be swapped among traders, while it adjusts the price of the tokens depending on liquidity demand and data from outside oracles.

Position tokens are a disruptive technology equipped with various properties, such as:

- No need for further funding as these are always fully collateralized (i.e. there are not any margin requirements);
- They are ERC-20 tokens, so it is easy to store them in wallets (e.g. MetaMask);

- No fixed expiry date so holders do not need to rollover their positions as token expiry reaches;

- These allow holders to open a leveraged position (i.e. in this case, the ratio between the price of L tokens and S tokens will differ from the spot price available on the market).

These tokens do not require the entire value of the collateral to be locked up in a smart contract: it means that contracts of much larger sizes can be traded with fewer collateral requirements on Mettalex. Let's consider the following example:

> **E.g.**: *A trader wants to open a position on the market via buying an asset like Bitcoin (i.e. BTC). If the BTC price on the market is BTCUSD = $10'000,00, with Cap = $11'000,00 and Floor = $9'000,00 the collateral required on Mettalex DEX to mint a pair of L/S tokens would be equal to $2'000,00 rather than the market spot price equal to $10'000,00.*

As outlined in the example above, distinctly from conventional market, Mettalex leverage requirements refer to price movements: the trade happens in a range of price, called **Δ** (i.e. **delta**), until this one hits a pre-set price band of reference (i.e. **floor** or **cap**). The value of a L/S pair of tokens is equal to the Δ, so the pair can be minted just by depositing a collateral as worth as Δ**.** The underlying spot price moves inside this range of values and as it trades away from the centre price (i.e. the central value of the band), it will be cheaper to buy tokens representing the trend opposite to the price movements. If the spot price breaches the band, the L/S pair will be settled automatically and all of the backed collateral will be distributed back to the token holders. The new spot price will initialise the smart contract around the new price value with the same Δ. At this point new position tokens will be minted and trading starts again. The following example helps to visualise the idea:

> ***E.g.**: Table 2 depicts the historical price graph for the London Metal Exchange (i.e. LME) Steel Scrap:*

Table 2: LME Steel Scrap historical price graph.

For commodity Position tokens we expect the spot price to usually trade inside Δ. If we consider the steel scrap this might range from $225 to $375 per tonne (i.e. a delta of $150 centred around $300). We can use this trading range to set the floor and cap prices for the token pair. The value of a long and short pair of tokens is equal to the delta, here $150,00, with floor set at $225,00 and cap at $375,00. That is, depositing a collateral worth $150,00 will allow the trader to mint a L/S pair. Assuming the spot price when the tokens are minted is $300, each token will have a value of $75,00, representing a leverage of $300,00/$75,00 = 4x.

Choosing the floor and the cap basesing on historical delta allow us to create a non-expiring perpetual token that can be held indefinitely as long as the spot remains within the historical range. This has the consequence that as the spot trades away from the centre price it becomes cheaper to buy exposure for the position opposite the movement.

Following the example above, if spot increases to $350,00 the value of a long token becomes $125,00 (i.e. leverage of $350,00/$125,00 = 2.8x) while the short token is now worth $25,00 (i.e. leverage of $350,00/$25,00 = 14x). This is an incentive for market participants to provide liquidity for the short position at low.

Mettalex offers to **Liquidity Providers** (i.e. **LPs**) the chance to supply liquidity directly to Mettalex system through the **Liquidity Provision Layer**. The liquidity collected throughout this layer is stored into a decoupled liquidity pool. The AMM uses Mettalex smart contract to convert supplied collateral into position tokens for market making operations. In exchange for locking up their collateral, LPs are rewarded with a further aggregated yield on the capital invested via transaction fees and trading spreads between prices, according to the amount and duration of liquidity supplied into the system. Compared with other liquidity pools, like Uniswap or Balancer, Mettalex allows LPs to deposit just a token in order to provide liquidity, as only one asset (i.e. **USDT**) is contained in the **Liquidity Providers pool** (i.e. the **LPs pool**). Other liquidity pools liquidity providers would need to supply a pair of tokens (e.g. ETH/USDT) in order to provide liquidity into them. The Mettalex decoupled LPs pool works as a decoupled liquidity pool whose role is to guarantee the provisioning of liquidity to the AMM so minimizing the timing risk of any impermanent losses the AMM is exposed to. This liquidity will be then converted internally by the AMM pool (i.e. it is a different Balancer-based pool) into a mix of USDT, Long and Short tokens which will be supplied to the Exchange layer in order to be traded among traders on the market. This is, the latter pool contains a mix of tokens whereas LPs pool contains just stablecoin collateral (e.g. USDT). The mechanism is based on the fact that LPs take on the timing risk in exchange for a return from the AMM. The contract is very simple as it focuses on only one variable: total liquidity provided (i.e. deposits). The objective is having always enough liquidity in the pool. So, the liquidity provision layer represents the component that funds autonomous market making.

Because of this, the **Governance Layer** rewards LPs with governance tokens (i.e. **MTLX Tokens**) via a liquidity mining process that rewards in proportion to the amount and duration of supplied liquidity in the system. These tokens were initially distributed among Fetch.ai stakeholders (i.e. that is, FET token holders) in accordance to the quantity of FET owned.

This layer enables decentralized governance of the platform itself: MTLX tokens enable stakeholders to take part in the decision making process regarding the platform. These allow them to govern on the policies and fees applied inside the platform itself. Furthermore, stakeholders (i.e. liquidity providers) can express their vote on different aspects and policies, including:

- System parameters (e.g. the choice of the AMM to back with liquidity from the liquidity pool);
- Creation of new markets;
- Usage of the collected exchange fees;
- Percentage of spread going into the liquidity pool;
- The buy-back and borrowing rates from the liquidity pool.

The following table offers an overview of the Mettalex Decentralized Exchange Platform (Table 3).

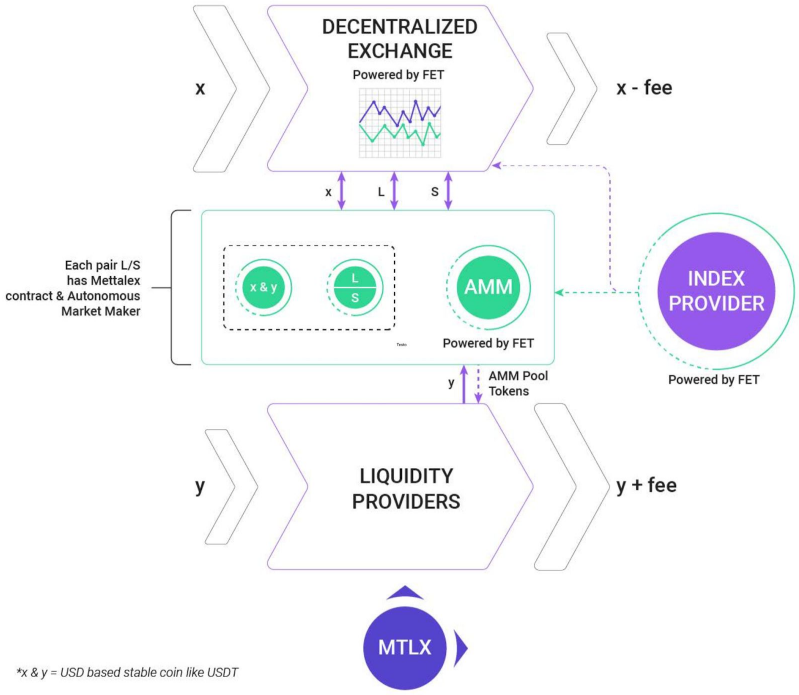

Table 3: MTLX system diagram. Source: Mettalex Litepaper, September, 2020.

As an additional part of the mechanism, Mettalex DEX uses a fraction of the exchange fees earned on the platform to algorithmically buy MTLX tokens back from the market. Governance tokens are minted at a linear rate to incentivize early liquidity providers in the system. However as the total liquidity in the pool increases, liquidity mining will become more difficult.

In conclusion, as we can see in Table 3, Mettalex DEX comprehends three different pools of liquidity. Even if all these liquidity pools are connected one to each other at the same time these work separately and each one of them accumulate a different type of collateral tokens. We can distinguish the following *3 pools* of tokens:

- **Mettalex vault**: contains the liquidity coming from the collateral used to back a pair of L and S Position tokens (i.e. Tokenisation Layer);

- **LPs pool**: LP supplies coin to earn a return return from trading fees (i.e. Liquidity Provision Layer);

- **AMM pool**: AMM that contains a mix of stablecoins in addition to L and S tokens. The liquidity contained in the AMM pool is for traders to swap between each other (i.e. Exchange Layer).

Exchange UI

Interactions with Mettalex DEX will require the ability to call some Mettalex contract events, (e.g. Mint L/S Position tokens from coin tokens (USDT), redeem L/S token pair for coin tokens or redeem single Position token for coin tokens if contract is settled).

This means that each user needs to call different methods accordingly to the liquidity pool they want to interact with (i.e. Mettalex Vault, LPs pool or AMM pool)

Possible Events: Mettalex Vault

The possible scenarios a user can face when operating with position tokens in the Mettalex Vault are the following:

Allowance

```
allowance(address owner, address spender)
```

This method returns the remaining number of tokens the `spender address` is allowed to spend on behalf of the `owner address` through the method `transferFrom`. This value is `zero` by default. This value changes when `approve` or `transferFrom` are called.

Approve

```
approve(address spender, uint256 amount)
```

This method sets an amount `uint256` of tokens as the allowance of the `spender address` over the caller's tokens. It returns a boolean value indicating whether the operation succeeded or not.

BalanceOf

```
balanceOf(address account)
```

Returns the amount of tokens owned by the account.

IncreaseAllowance

```
increaseAllowance(address spender, uint256 addedValue)
```

This method is called when the user wants to increase the allowance related to his address. `Uint256` indicates the value added to the existing allowance.

DecreaseAllowance

```
decreaseAllowance(address spender, uint256 subtractedValue)
```

This method is called when the user wants to decrease the allowance related to his address. `Uint256` indicates the value subtracted to the existing allowance.

MintPositions

```
mintPositions(uint256 _quantityToMint)
```

This method is called to mint new position tokens where `uint256` indicates the amount of positions to be minted and transferred to the user's account.

MintFromCollateralAmount

```
mintFromCollateralAmount(uint256 _collateralAmount)
```

This method is called to mint new position tokens depending on the amount of collateral deposited. The amount of collateral to be deposited is given by `uint256`.

RedeemPositions

```
redeemPositions(address _to, uint256 _redeemQuantity)
```

This method allows the user to redeem the collateral to send to the sender address.
The method is called to redeem the given amount of positions held by the user. The address the user wants to use to transfer the collateral redeemed is indicated by `address`. The amount of positions to redeem is given by `uint256`.

SettlePosition

```
settlePositions()
```

This method allows the users to settle by returning collateral and burning position tokens.

BulkSettlePositions

```
bulkSettlePositions(address[] calldata _settlers)
```

This method allows the users to settle multiple accounts by returning collateral and burning position tokens. The parameter `_settlers address[]` indicates the array of user accounts to be settled.

TotalSupply

```
totalSupply()
```

Returns the amount of tokens in existence.

Transfer

```
transfer(address recipient, uint256 amount)
```

This method is called when a user wants to move an amount `uint256` of tokens from the caller's account to the `recipient address`. This method returns a boolean value indicating whether the operation succeeded or not.

TransferFrom

```
transferFrom(address sender, address recipient, uint256 amount)
```

When this method is called it generates a transfer event in which the user wants to move an amount `uint256` of tokens from `sender address` to `recipient address` using the `allowance` mechanism. The amount transferred will be deducted from the caller's allowance. This method returns a boolean value indicating if the operation succeeded.

Possible Events: LPs pool

The scenarios a liquidity provider can face when operating with the the LPs pool can be the following:

Deposit

`deposit(uint _amount)`

This method allows users who want to provide liquidity to deposit the quantity of liquidity desired into the pool. `uint _amount` indicates the amount of liquidity to be deposited.

Withdraw

`withdraw(uint _shares)`

With this method users who want to withdraw their liquidity from the LPs pool. `uint _shares` indicates the amount of liquidity the user wants to withdraw.

WithdrawAll

`withdrawAll()`

With this method the user can withdraw all of their liquidity from the LPs pool.

Earn

`earn()`

This method transfers money from the vault to the autonomous market maker contract in order to earn trading fees from the supplied funds.

GetPricePerFullSare

`getPricePerFullShare()`

This method is called when the user wants to get the price for the entire amount of liquidity he owns in the liquidity pool.

Possible Events: Autonomous Market Maker

ClaimFees

`claimFee(address _to)`

This method is called to claim the fee accumulated by the vault. The parameter `address _to` indicates the address to transfer the fees accrued.

GetExpectedOutAmount

`getExpectedOutAmount (address fromToken, address toToken, uint256 fromTokenAmount)`

`returns (uint256 tokensReturned, uint256 priceImpact)`

This method is called to get the expected amount of tokens (i.e. `uint256 fromTokemAmount`) given out in a swap operation from the `fromToken` address to the `toToken` address.

This method returns the amount of tokens returned to the user's account (i.e. `uint256 tokensReturned`) and the price impact due to the operation (i.e. `uint256 priceImpact`).

GetExpectedInAmount

`getExpectedInAmount (address fromToken, address toToken, uint256 toTokenAmount)`

`returns (uint256 tokensReturned, uint256 priceImpact)`

This method is called to get the expected amount of tokens (i.e. `uint256 toTokemAmount`) received in a swap operation from the `fromToken` address to the `toToken` address.

This method returns the amount of tokens returned to the account (i.e. `uint256 tokensReturned`) and the price impact due to the operation (i.e. `uint256 priceImpact`).

GetSwapFee

`getSwapFee()`

It shows the fee related to the swap operation.

GetBalance

```
getBalance(address token)
```

It is used in order to get the balance of tokens owned by the `address`

HandleBreach

```
handleBreach()
```

Settle all Long and Short tokens held by the contract in case of Commodity breach. It should be called only once.

SwapExactAmountIn

```
swapExactAmountIn (address tokenIn, uint256 tokenAmountIn, address tokenOut, uint256 minAmountOut, uint256 maxPrice)
```

```
returns (uint256 tokenAmountOut, uint256 spotPriceAfter
```

This method is called when the user wants to trade an amount `uint256 tokenAmountIn` of `tokenIn` taken by the pool, in exchange for an amount `uint256 minAmountOut` of `tokenOut` given to the user from the pool, with a maximum marginal price equal to `uint 256 maxPrice`.

This method returns the amount of tokens taken out from the user's account (i.e. `uint256 tokenAmountOut`) and the new spot price after the swap operation (i.e. `uint256 spotPriceAfter`).

UpdateOracle

```
updateOracle(address _newOracle)
```

This method changes the address of the Oracle contract. `address` is the new Oracle address.

UpdateSpot

```
updateSpot(uint256 _price)
```

This method updates the spot price of an asset. `uint256` is the new updated price. The update can take place only if the arbitration price is within the contract bounds otherwise the contract settles.

UpdateCommodityAfterBreach

```
updateCommodityAfterBreach(address _vault, address _ltk, address _stk)
```

This method is called to update the contract addresses after a band breach occurs. `address _vault` is the vault address whereas `address _ltk` and `address _stk` are respectively the Long and Short token addresses.

Overall risks and benefits

Nowadays most of the trades are conducted through centralized exchanges. These are relatively efficient as they require traders to deposit the assets first in order to proceed with the trade. Dishonest or unprofessional exchange operators may confiscate or lose the assets given into custody. Moreover, centralized exchanges are susceptible to be attacked and therefore are exposed to malicious third parties. On the other hand, decentralized exchange protocols mitigate these issues by removing the trust requirement. Users do not need to deposit their assets as these remain in their control until the trade is executed. Smart contracts represent the linking bridge between traders. Both sides of the trade are performed in one indivisible transaction, mitigating the *counterparty credit risk*. The contracts can assume the roles of custodians, escrow agents and central counterparty clearing houses (i.e. CCP).

For more information about the Decentralized Finance (i.e. DeFi) world and the overall risks to which traders are exposed to, visit the link below:

Leitfaden Decentralized Finance (DeFi) – A new Fintech Revolution?

Asset holders are equipped by the system with risk management tools without charging high spreads. Users are not exposed to counterparty risk as all transactions and trades happen through the Mettalex smart contract. This decentralization can additionally be guaranteed through Position tokens which are backed by a fully collateralized collateral so removing the need for margin requirements and settlement. That is, no further funding is needed. Mettalex platform users have the opportunity to generate their own liquidity by creating more position tokens by locking-up new collateral. These new minted token pairs can be used to mitigate exposure to risks or create an exposure to certain commodity assets thanks to leveraging

opportunities. The autonomous market maker uses a liquidity sensitive algorithm with bounded loss to manage market risk.

As mentioned before, a trade happens inside a band of price values (i.e. Δ). If traders open a long position and then the spot price goes up then they will experience an increase in the value of the L token and a decrease in the value of the S one. Demand and liquidity in the decoupled liquidity pool play an essential role in the automated market making process as the AMM adjusts the price of each token according to liquidity demand and the reference index fed from outside oracles. As demand arises, the price of L tokens will go up until it reaches the cap. At this point the value of L tokens would be equal to the value of the backing collateral and S tokens would be worthless. Viceversa, when the spot price falls and reaches the floor of the reference band, the price of S tokens would be equal to the price of the backed collateral whereas L tokens price would be equal to zero. In this case, traders who had opened a short position would have gained a profit. That is, traders are exposed to *volatility risk*.

On the other hand, liquidity providers expose themselves to *timing risk* when providing liquidity to the AMM. liquidity providers will see substantial returns on their investments due to trading fees. This way, it is possible to collect enough liquidity to provide to the AMM if needed to guarantee the execution of transactions. Therefore, the Governance Layer rewards LPs with MTLX tokens, which will be earned by them via a liquidity mining process in proportion to the liquidity provided and its duration in the pool. These governance tokens will increase LPs governance influence and can gain additional yield via yield farming strategies. Furthermore, LPs can trade using their position tokens on the market by opening a long or short position according to their market strategies and risk propensity as the trader will have to face volatility risk connected to the underlying asset price. However, each LP is allowed to exit a position at any point.

An additional way Mettalex DEX reduces risk is through the properties of position tokens. With no expiry date or delivery associated with the tokens, the need for market participants to engage in complex rolling of positions or carry trades is eliminated. Mettalex's cap and floor trading band gives confidence to traders as any potential losses will be limited to the initial acquisition cost of the tokens. Mettalex DEX therefore offers the potential for scalable liquidity to be brought to sectors of the commodities market that have either been thinly-traded or inaccessible to the vast majority of traders, providing a decentralized model that also helps to keep transaction fees to a minimum.

On the other hand, liquidity providers expose themselves to *timing risk* when providing liquidity to the AMM. liquidity providers will see substantial returns on their investments while also earning governance tokens that will increase their governance influence (i.e. LPs providing liquidity in the decoupled LPs pool can gain additional yield via **yield farming strategies**).

In Mettalex, liquidity is provided in a single token and not as a ratio of multiple tokens. Also, the liquidity is provided using stable coins (e.g. USDT/USDC) which reduces the risk of volatility. The stable coins are then used as a collateral to issue position tokens whose combined value is

always equal to the collateral value. Whereas there is always a risk of loss in supplying liquidity to pools, in Mettalex LPs due to lower volatility of the underlying assets and provision of liquidity in stablecoins, the risk of impermanent loss is very low. This way, it is possible to collect enough liquidity to provide to the AMM if needed to guarantee the execution of transactions. Therefore, the Governance Layer rewards LPs with MTLX tokens, which will be earned by LPs via a liquidity mining process in proportion to the liquidity provided and its duration in the pool. Furthermore, LPs can trade using their Position tokens on the market by opening a long or short position according to their market strategies and risk propensity as the trader will have to face volatility risk connected to the underlying asset price. However, in regular circumstances, liquidity suppliers have the opportunity to exit the system in times of high demand. This provides them with an opportunity to recoup capital before re-engaging with the system. By the way, in periods where need is high, they may be required to wait a short period in order for a baseline level of liquidity to be reached. This is not only to make sure the system remains afloat, but also to protect the investment they have made into the exchange.

References

"*Announcing the MTLX token*", website available at: Announcing the MTLX token. It's the news you have been waiting… | by Fetch.ai | Fetch.ai

"*Decentralized Finance (DeFi) – A new Fintech Revolution?*", website available at: Leitfaden Decentralized Finance (DeFi) – A new Fintech Revolution?

"*Decentralized Finance*": On Blockchain- and Smart Contract-based Financial Markets, website available at: (PDF) Decentralized Finance: On Blockchain- and Smart Contract-based Financial Markets

"*What Is DEx (Decentralized Exchange)?*", website available at: What Is DEx (Decentralized Exchange)?