Autradix – Self-optimizing, Decentralized, Non-custodial Trading DeFi Protocol

Steffen Kux
autradix.io, Chemnitzer Straße 49e, 09648 Mittweida

*Cryptocurrencies are characterized by high volatility, both in the short and long term. Experienced traders exploit this to make profits from price fluctuations by swing trading. However, this requires closely observing and analyzing the prices and trading positions at the right time. Only a few specialists, who spend time focusing on this, or optimized trading bots are able to actually make continuously profits. The **autradix protocol** is a self-optimizing and self-learning parametric trading algorithm that analyzes price actions in real-time and adaptively optimizes the algorithm's parameters to realize the user's investment objective. Embedded in an adaptive genetic algorithm, possible parameterizations are simulated and the optimal for the investigated trading pairs are calculated. The generic trading protocol API enables coupling with various crypto exchanges and decentralized protocols. A smart contract based decentralized, trustless, and tokenized fund, controlled by a DAO, enables users to invest, operate trading agents, and to participate in the profits generated according to their share.*

1. Introduction

The aim of the **autradix protocol and ecosystem** is to establish a decentralized and trustless DeFi protocol Instrumentalizing the chances of a very volatile market by implementing an automated and self-optimizing trading strategy, powered by adaptive parameter optimization and supported by artificial intelligence methods analyzing social media and other sources.

The autradix protocol opens these possibilities also for investors who have neither the necessary knowledge nor time to apply a good trading strategy. The target groups are smaller investors who also want to profit from the movements of the crypto market or want to stabilize the value of their asset collection as well as investors looking for a stable and automized trading bot.

1.1 Trading Objectives

There are two fundamentally different objectives for automated trading:

1. Trading to realize maximum gains (increase in value).
2. Trading to maintain the value of the investment despite price fluctuations (value preservation).

The autradix protocol can be configured and optimized for both approaches, allowing users to define their trading strategy and thus their chances of winning, but also the risk.

1.1.1 Increase in Value

If the goal is to achieve maximum profits, the attempt is made to take the best possible advantage of every price movement in order to increase the overall value of the portfolio. In particular, short-term price changes are of interest. Gains against the long-term trend can be realized as well. However, this focus on short-term gains also entails a significantly higher risk, since the relatively small short-term price fluctuations also quickly entail the risk of realizing negative trades. This is additionally limited by the fees since a trade can only be successful if it realizes more than the required fees.

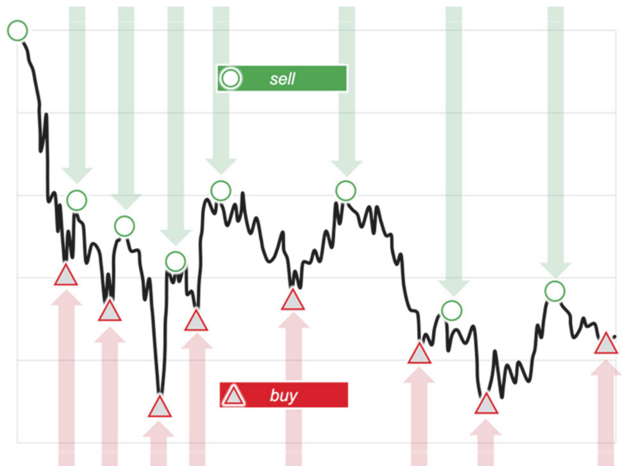

Figure 1: Short-term trading (principle)

1.1.2 Value Preservation

In contrast, the focus of value preservation is not on short-term fluctuations, but on compensating for long-term negative price changes and the associated loss in value of the portfolio by rebalancing the weighting of the assets in the portfolio in such a way that the value of the portfolio has a higher overall value compared with the original composition (simple "hodl"). The aim of this strategy is to achieve the most stable value within the specified portfolio limits. The risk of the strategy is lower than with short-term trading.

However, there is still the risk that the adjusted portfolio is less favorable and thus loses more value than the original one. Likewise, price fluctuations with very strong gradients may not be recognized in time, which can lead to the fact that such price jumps or price falls cannot be reacted to in time.

Especially in this approach, the support of appropriate artificial intelligence methods can be very useful to combine the investment strategy and the need for fast action.

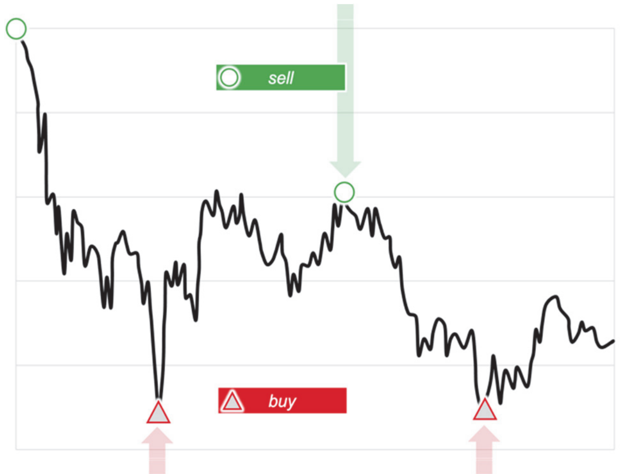

Figure 2: Long-term trading (principle)

2 Trading Model

2.1 Overview

Price developments of shares and cryptocurrencies in particular are characterized by a stochastic pattern. Therefore, future developments cannot be predicted from past data. Market influences, the illogical behavior of market participants (fear of missing out behavior or panic selling), the link to local and global events (catastrophes, wars, crises, economic and social events), the impact of influencers (e.g., when an influencer sends a positive or negative tweet about a cryptocurrency), and last but not least the very different expectations and forecasts can have quite unpredictable effects.

On the other hand, the impact of regulatory efforts (promotion or hindrance of market activities, regulation of mining or other crypto activities, or restrictions of DeFi activities, ...) but also the occurrence of common trading strategies (such as triggering stop-losses or other trading triggers) can also influence at least the probability of a certain market behavior within certain limits.

2.2 Mathematical Model

Many automated trading strategies are based on applying mostly heuristic methods of price analysis or AI approaches and giving the trading bot an advantage due to its 24/7/365 availability and quick response capabilities. The basic algorithm of autradix is a purely mathematical approach that derives trigger events for trading currency pairs from the analysis of real-time data.

2.2.1 Numerical Curve Analysis

For this purpose, the data of the price are suitably smoothed. The price data $[t_i, y_i]$ are obtained as pairs of discrete values of the time t_i and the price at this time y_i with an average sampling rate of 1 second from a price oracle. This can be the price information of a custodial exchange on which trading is to take place or a decentralized price oracle.

The data is mathematically smoothed so that the resulting curve can then be further processed. The smoothed value $\hat{y}_i$ at index i with the length l for the smoothing interval is calculated by a weighted averaging using the weighting function $\omega_i(k)$

$$\hat{y}_i = \sum_{k=i-l}^{i} \omega_i^o(k) y_i \quad with \quad \sum_k \omega_i^o(k) = 1.$$

For an assumed simple weighting function

$$\omega_i^o(k) = \begin{cases} \frac{1}{l} & i-l \le k \le i \\ 0 & else \end{cases}$$

the smoothing function is

$$\hat{y}_i = \frac{1}{l} \sum_{k=i-l}^{i} y_i \ .$$

The length l of the smoothing interval determines whether the long-term or short-term behavior of the price trend is more interesting. A more risky but maximum performance-oriented strategy will tend to work with a short smoothing interval (e.g., of a few seconds or minutes) while a strategy to hedge against loss of value will work with much longer intervals, since here the longer-term trend and less the short-term fluctuations are of interest.

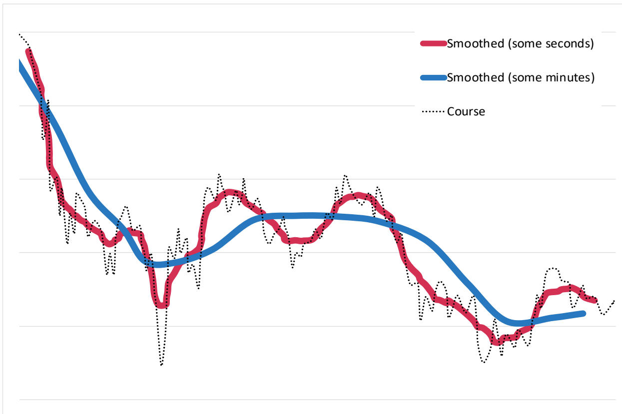

Figure 3: Price curve and different smoothing intervals

Short smoothing intervals lead to smoothed functions following fast the course of the stochastic course, while long intervals representing the long-term trend of the course, react however accordingly sluggishly to short term changes. With the weighting function the characteristic of the smoothing can be tuned. With suitable weighting functions (other than the simple approach above), it can thus be achieved that movements at current time are better recognized even with longer smoothing intervals.

An indicator that a potential local maximum or minimum of the price has passed is a minimum or maximum on the smoothed price curve. For this purpose, the smoothed curve is derived numerically. The difference quotient is calculated for all positions of the smoothed curve. Except at the edges of the curve, the central difference quotient is used

$$\dot{y}_i = \frac{\Delta \hat{y}}{\Delta t} = \frac{\hat{y}_{i+1} + \hat{y}_{i-1}}{t_{i+1} - t_{i-1}}$$

This is possible because the smoothed curve is sufficiently smooth and at least once differentiable. For the decision whether it is a minimum or a maximum, the second derivation is needed also. To make this numerical stable, the curve of the first derivative is smoothed again appropriately

$$\hat{\dot{y}}_i = \sum_{k=i-l}^{i} \omega_i^1(k) \dot{y}_i \quad with \quad \sum_k \omega_i^1(k) = 1 \ .$$

Then the second derivation is calculated by once again calculating the difference quotient, now from the smoothed derivation curve

$$\ddot{y}_i = \frac{\Delta \hat{\dot{y}}}{\Delta t} = \frac{\hat{\dot{y}}_{i+1} + \hat{\dot{y}}_{i-1}}{t_{i+1} - t_{i-1}} \ .$$

A potential high point at i_{sell} which signifies a selling signal, is present when applies

$$\dot{y}_{i_{sell}} = 0$$
$$\ddot{y}_{i_{sell}} < 0$$

and a potential low point at i_{buy} signalling a buy chance is expected at

$$\dot{y}_{i_{buy}} = 0$$
$$\ddot{y}_{i_{buy}} > 0$$

The curve is always analyzed backwards from its most current position, since only the last local extreme points are relevant for a potential decision.

2.2.2 Thresholds

The identification of an extreme point is only a trigger for a potential trading chance. An event must not be triggered immediately, as this would lead to many wrong decisions due to the partly strong stochastic price fluctuations, since a valid trading signal requires a trend that allows a value gain to be achieved through a suitable trade.

For this reason, thresholds θ_{buy} and θ_{sell} are defined that must then be exceeded or undershot in the predicted direction to trigger an actual event. These threshold values must not be too large, as this would lead to a trading event being triggered very late (and perhaps too late) in a trend progression. If the threshold value is too small, it may happen that short-term stochastic fluctuations already trigger an event, which may lead to a trading action that moves against the trend and thus realizes losses.

The determination of appropriate thresholds depends strongly on the characteristics of the price trend (e.g., how strong the usual stochastic fluctuations are) and also on the trading strategy. Optimal values differ for different currency pairs and do not remain the same for the same pair over time.

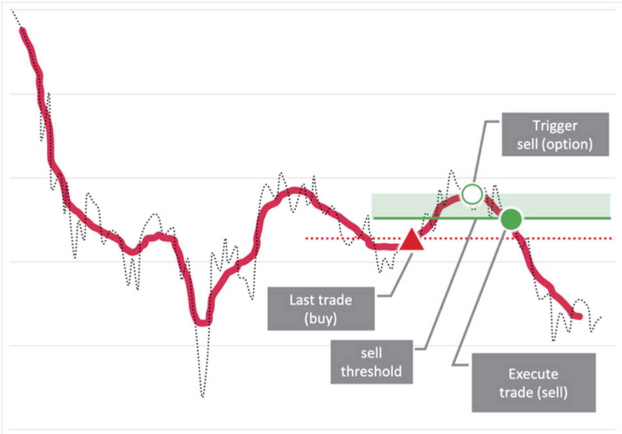

Figure 4: Trading mechanism (trigger, threshold, trade)

2.2.3 Impact Analysis

Furthermore, a portfolio impact analysis is performed before a trading event is triggered. This means that it is analyzed whether the trade actually has a positive impact on the total value of the portfolio or would cause a negative performance due to the local development. Also trading to another token may be more promising.

The impact analysis must be weakened over time, because otherwise the trading could be completely prevented if the price trend runs in the wrong direction and no positive trade is possible anymore (e.g., a purchase was made and before a sale could be triggered, the price turned and has fallen since then, so that the short-term fluctuations could be used, but no sale is possible due to the negative impact). Then a negative trade may be necessary to get back into business.

The impact analysis has also parameters I_{buy} and I_{sell} to control its behavior.

2.3 Parametric Model

The model described has a number of parameters that can be used to adapt the model to the requirements. All relevant parameters that determine the characteristics of the model can be controlled externally and are available as optimization variables.

In section 2.2, parameters for the mathematical model are already introduced:

- l ... length of the smoothing interval
- l_{der} ... length of the smoothing interval of the first derivation

- ϕ ... form parameter of the weight function for the smoothing algorithm (not described in this paper)
- θ_{buy} ... threshold to trigger a potential trade
- θ_{sell} ... threshold to trigger a potential trade
- r_{trade} ... maximum relative amount of an asset that may be used for a trade
- I_{buy} ... Impact analysis form factor (not further described in this paper)
- I_{sell} ... Impact analysis form factor

Other parameters (like additional chart analysis criteria) are not described here and may be introduced later.

2.4 Investment and Portfolio Strategy

When putting together a balanced portfolio, care should be taken to ensure that the assets that belong to it have appropriate weightings. When trading takes place, this weighting will be shifted. If everything is allowed, each token can be swapped partly or completely into other tokens without restrictions.

Since the selection of trading opportunities is done by the algorithm based on mathematical indicators, it does not take into account the risk attached to these tokens and so on. If the composition of the portfolio is to be actively influenced, it is possible to specify how large the minimum and maximum share of a token may be.

Based on these specifications, the trading algorithm can optimize its strategy to meet the portfolio composition and take these constraints into account when triggering a trading event.

2.5 Adaptive Parameter Optimization

With the parametric model, it is now possible to formulate a mathematical optimization task[1] to determine the optimal parameter configurations. The aim is to tune the model so that the best possible decisions can be made with this model for a currency pair or some combined currency pairs.

2.5.1 Objective function

Assume that there are potentially j assets α_j in the portfolio and each asset α_j has a value $\psi(\alpha_j) \geq 0$. The total value of the portfolio is

$$\Psi = \sum_j \psi(\alpha_j)$$

For the calculation of the potential asset value, starting from an initial value, trades are simulated on real data from the past to the present. The simulation interval Δt must be large enough that sufficient potential trades can be executed, but should also not be too large in order to keep the computational effort as low as possible, and also to be able to react to changes in the characteristics of the price trend.

The objective is to maximize the total value Ψ of the assets under the given constraints, simulated in the interval $[t_{now-\Delta t}, t_{now}]$:

$$\Psi_{[t_{now-\Delta t}, t_{now}]} \to max$$

Although the behavior of each currency pair can be viewed as independent by itself with respect to the behavior of other pairs with the value function $\psi(\alpha_j)$, these are not intependend if the portfolio strategy is considered as well as it is done in the portfolio value function Ψ. This means that the parameters for all currency pairs need to be optimized together to find the optimal parameter configuration for the whole portfolio.

2.5.2 Optimization variables

The already described parameters of the model and, if necessary, additional variables from chart analysis methods are used as optimization variables. Each variable is constrained by a suitable interval.

These parameters must be set for each currency pair.

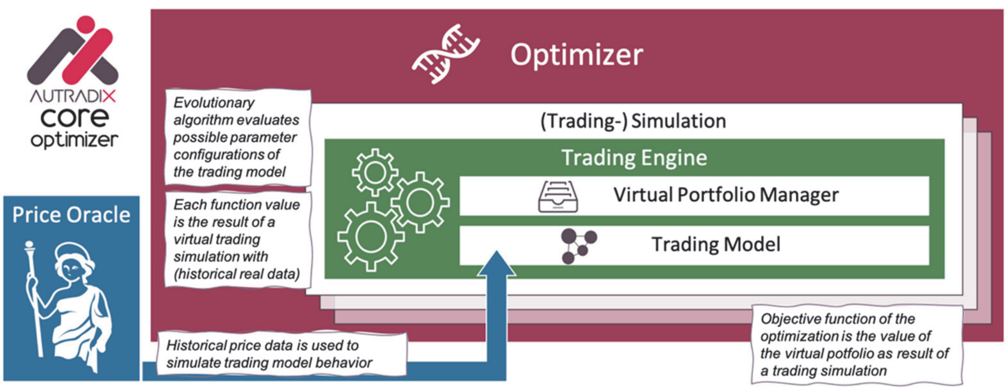

Figure 5: Optimization principle to find optimal parameter configuration for trading model

[1] see figure 5

2.5.3 Constraints

The specifications of the portfolio composition are formulated as constraints. This means that the ratio of the individual cryptocurrencies can already be taken into account when optimizing the parameters.

2.5.4 Optimization Algorithms

The optimization task is a highly dimensional, nonlinear blackbox problem.

The function value $\Psi_{[t_{now-\Delta t}, t_{now}]}$ is the result of a number of numerical simulations based on real price trends from the past.

No statement can be made about continuity and differentiability. Gradient information is not available. These could be calculated by numerical differentiation, but this would be very computationally expensive due to the many optimization variables. Furthermore, no information regarding the smoothness of the function can be predicted. In particular, discontinuities and jump points cannot be excluded.

The optimization method must therefore be an algorithm that does not require gradient information, works globally, and can also cope with non-continuous objective functions and constraints.

Genetic/Evolutionary Algorithms

Genetic or evolutionary algorithms are nature-inspired optimization methods[2] that apply the laws of heredity (e.g., Mendel's laws) to mathematical problems. Through repeated operations such as

- crossover (exchange of properties) and
- mutation (random changes),
- selection (sorting out good and bad solutions)

these methods tend to convert towards the optimum.

In addition, these algorithms do not impose any requirements on the smoothness, continuity, and differentiability of the objective functions and constraints, and no gradient information is needed. Thus, they are well suited for the described optimization problem. The fact that these algorithms require relatively many function evaluations for the optimization is not a serious problem, since the optimization does not have to be performed in real time.

2.6 Artificial Intelligence Support

Secondary information can give very good indications of whether a trading event is imminent or in which direction a price trend is highly likely to move. This information can come, for example, from social media, relevant forums, and communication platforms. Very different information could be considered.

[2] for more information, see [1]

2.6.1 Influencer

Influencers are people who have a significant impression on other people due to their strong presence in social media or their high visibility due to their celebrity. Beyond this flow of information, influencers can have a very strong impact on the decisions of their followers. Statements made by such people quickly go viral and are spread very quickly through their network of followers.

For example, Elon Musk has significantly moved the price of cryptocurrencies like Doge and Bitcoin up or down with a few tweets. Analogous actions are also not unknown from stock market trading. So, a statement on the interest rate level of the European Central Bank by the head of the Central Bank can have an influence on buying or selling decisions of bonds.

The AI model is trained to filter and recognize those messages from the flood of social media postings that come from key influencers and have the potential to go viral and drive others to more or less predictable trading behavior.

However, if a potential influencing event is detected by the AI, this is not immediately acted upon (i.e., immediately translated into a trade), but this information is used to tune the parametric trading model so that it will react faster and more actively to a resulting price change.

This is done, for example, by changing the parameters of the smoothed curve l and ϕ and the limit values θ_{buy} or θ_{sell} in order to be able to react more quickly to an imminent price rise or fall. The trading ratios r_{trade} can be adjusted, too, to optimize the volume in order to achieve a higher profit.

Especially the impact analysis needs to be tuned if predictable behavior is expected to prepare the algorithm to be ready to act.

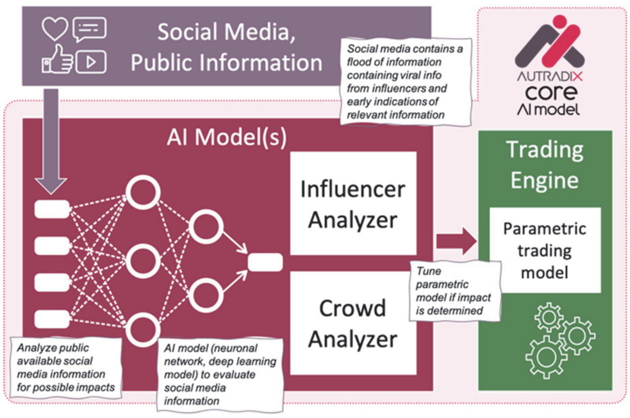

Figure 6: AI models to evaluate social media information

2.6.2 Crowd behavior

Social media and relevant discussion forums can provide even more helpful information. For example, if many people discuss that a cryptocurrency (or the project behind it) has run into difficulties (e.g., due to a software hack), the expectation can also be derived from this that a price drop is to be expected with a certain probability. Conversely, positive news (such as the achievement of a financing round or the release of a new version) are also indicators of a possible price increase.

Of course, this information is no guarantee for such a price movement, because often the market does not act logically, but can be an important indication that such a movement is being expected.

The AI model used for this purpose is trained to recognize that positive or negative information related to a cryptocurrency is accumulating. Similar to influencer analysis, these trigger signals are used to adapt the parametric model so that it can react more effectively to such behavior.

3 Technology

3.1 Generic Trading Interface (GTI)

The trading strategy of the autradix algorithm is by itself completely independent of the underlying trading platform. This means that centralized exchanges such as Crypto.com, Coinbase, Kraken, Binance, etc. can be used, as well as decentralized exchanges such as Uniswap and others.

Many of the centralized exchanges provide APIs for managing accounts and executing trades. The autradix protocol embeds these APIs in a generic interface so that users can choose self-determined which exchanges they want to use. The generic interface also allows decentralized exchanges to be integrated in the same way. In this way, it is possible to integrate and use the whole variety of crypto ecosystems according to the requirements and wishes of the users.

The generic trading interface (GTI) provides the following functions:

- Query of price information on the selected currency pairs.
- Manage the wallet/account.
- Set and stop trades.

3.1.1 Centralized Exchange APIs

Centralized marketplaces allow their users to trade coins and tokens. In addition to spot trading, leveraged products are often offered (margin trading). Even though the autradix protocol is very generic, leveraged trading is currently not supported.

In addition to direct buying and selling at the current market price, exchanges offer more complex trading rules such as limits, stop loss, stop limit, ... These are also not supported at present.

The fee model for bid/ask trades of central exchanges is mostly volume dependent. For each trade a fee is charged, which depends on the value of the transaction. This means that a small transaction will cost less than a large one. For the autradix protocol results that a transaction is only profitable if the relative profit is higher than the relative fees.

For trading, relative profit must always be maximized on centralized exchanges, not the absolute profit. On the other hand, it is possible to trade small volumes without incurring excessive fees, as long as the relative profit is sufficient.

Since the trade is only one transaction in the exchange's system database and does not require any transaction

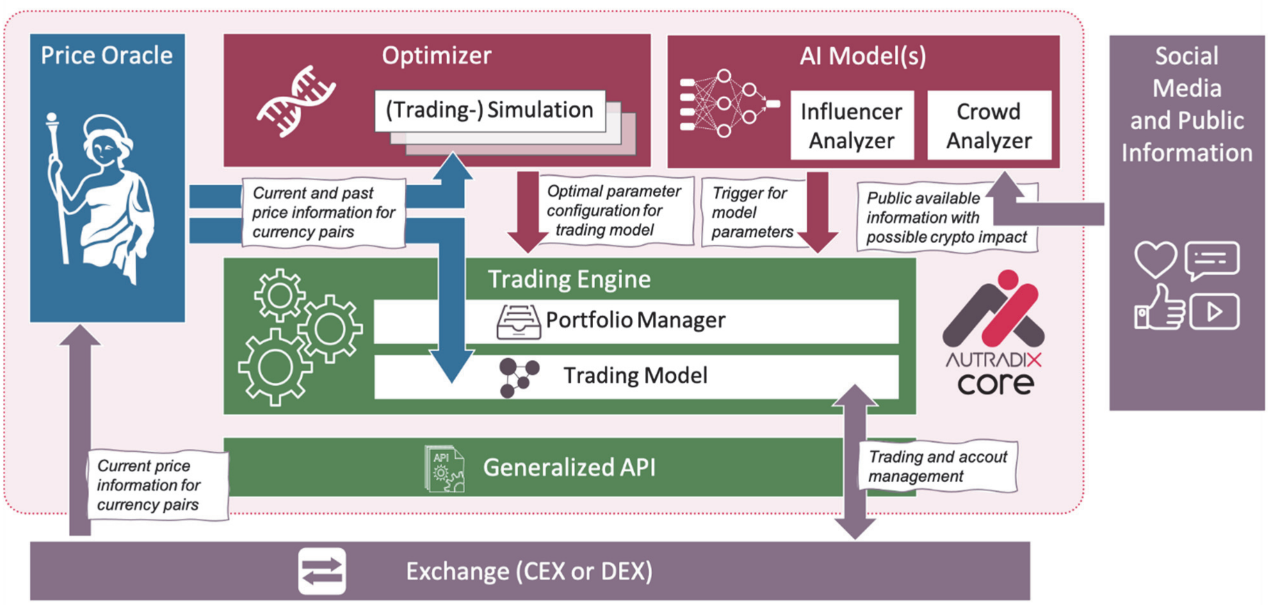

Figure 7: autradix core components

on the blockchain, the execution time is usually very short. That is, the transaction is final almost in real time.

The tradable volume and the resulting price depend heavily on the liquidity and the available supply or demand of the exchange.

3.1.2 Decentralized Exchange APIs

In decentralized exchanges, a trade means a transaction on the blockchain. The fee which is the blockchain's transaction fee is independent of the volume of the transaction. Instead, it depends on the current gas price and how much gas is required for the interaction with the smart contract of the exchange.

If the usable gas for the transaction is set too low, it is possible that the transaction of the trade will take a very long time to be executed or even will not be executed at all. The offered gas has to be high enough to ensure that the transaction is executed quickly. This is necessary to reduce the risk that the price has changed too much between triggering the trade event and the actual trade which could lead to a negative trade if the price moves into the wrong direction in between.

Compared to trades at a central exchange, DEX transactions take significantly longer to become final, as the same rules apply as for any other transaction on the blockchain.

The fees for such a transaction are independent of the volume which means that even large transactions do not incur higher fees. Conversely, however, this also means that high transaction costs can be incurred for small volumes, which makes this transaction less effective or even negative.

When trading on DEX, it is therefore not the relative profit but the absolute profit of the transaction that is relevant for executing a successful trade.

As many DeFi products are currently working on processing transactions in Layer 2 (e.g., rollups), this will also bring essential advantages for this application, as it will drastically reduce transaction costs.

However, this also cannot be scaled arbitrarily. With increasing volume and depending on the liquidity of the pool of the currency pair, a trade has increasing influence on the price. This means that a potential sale of a large volume can cause the price to drop. The same is true in the other direction as well.

For that reason, the autradix protocol is considering this as well when a trade is going to be created.

3.2 Price Oracle Service

For the simulation described above which is among others the basis for the calculation of the objective function value of the optimization task, historical price trends of the currency pairs under consideration are required.

Both centralized and decentralized exchanges offer APIs or oracles to read price data. Depending on the API, access to historical data may also be possible to a certain extent. Depending on the simulation used, more extensive data sets are required than can't be retrieved directly from the APIs.

The autradix Price Oracle Service provides a service that uses the APIs to one or more exchanges to retrieve price data at the required sampling rate and stores it in its own database, and then in turn provides a REST API through which this data can be retrieved from the autradix protocol.

For a system to be tamper-proof, it must be ensured that this data is correct and available. This can be done by a trusted node or better by a decentralized trustless network of oracle providers.

3.3 Solutions

The autradix core software, consists of

- The price oracle which collects price data and provide historic data for the simulation,
- The trading engine with the parametric trading model and the portfolio manager,
- The optimizer for the trading model parameters,
- The simulation engine to simulate trading with historic data,
- The AI model to analyze and evaluate social media and other information to tune the mathematical model, and
- The interface to the exchanges.

The goal of the software architecture is to enable a system that can meet the different requirements of the users in terms of convenience, self-sovereignty and compliance.

In figure 7 is visualized, how these core components work together and interact with exchanges and information sources.

3.3.1 Software-as-a-Service

Many applications are offered to the user as software-as-a-service (SaaS). The actual software service is hosted on a server (see figure 8). Users can then create an account and use the service offered within this framework.

In terms of user-friendliness, this is very simple. The software, which is offered as a web application or mobile or desktop app, is accessed via the user's account which is defined and secured by user name and password and, if appropriate, 2FA for greater security.

The backend is operated by the service provider, without the end user coming into contact with it, except through configuration via the application interface, if that is required.

This software application is operated as a central cloud service. The end user must trust the operator – regarding the running software components, the IT infrastructure, and the account and key management to access the service. The service provider may also charge a fee (e.g., share of profits, setup fee, ...).

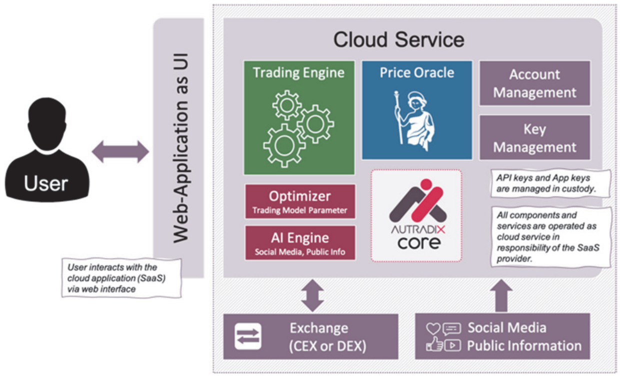

Figure 8: SaaS Architecture

The user has no influence on which software version is used or which exchanges are connected and does not have to worry about it and the operation of the IT infrastructure. If the assets are managed on a central exchange, the user remains in full control of them (or correctly, the exchange which acts as custodian for the user), as the autradix SaaS service can only trade via an API key and this key can be restricted in terms of permissions (only read and manage trades, but no withdrawals or transfers not approved for trading).

As far as self-sovereignty is concerned, since the assets are held in custody at the exchange, the user retains control and the autradix service cannot access the assets directly.

The situation is somewhat different for decentralized exchanges. The assets are held in a wallet in which the user has full control over the private keys. Transactions are signed with these key(s). In order for a central service in the form of a SaaS to trade on a DEX, the service needs access to these keys to sign the required transactions.

This means that the end user must make the private key(s) available to the autradix service. The service would then have full control over the assets and thus the service would have to be trusted to handle the key correctly and not misappropriate the assets.

This problem is solved by using a smart contract multisig wallet with special app keys which have restricted permissions only. The multisig wallet has keys that only the owner can control and with which all transactions can be executed.

Furthermore, the wallet has a special application key that can only execute transactions that are required to execute trades on the corresponding DEX, although the recipient of the returned tokens must be the wallet itself. This application key can then be made available to the service and it can be used for trading activities without gaining control over the assets beyond the granted permissions.

From a compliance[3] point of view, it will be necessary for the users who have an account with the service (centralized exchange) to go through an appropriate KYC process. As the autradix service has no control over the assets at any time, no additional compliance actions are needed.

If a DEX is used and the user gives the autradix service only an app key with limited rights, the control remains with the user and no additional KYC is required.

3.3.2 Local Node

Truly self-sovereign operation of the autradix system is possible by the users running their own autradix node. The backend software is largely the same as in the SaaS approach, but there is no central service operator of the platform. Instead, the user runs the service locally on a node. This node is configured to connect to the user's trading venues and trades there with the user's accounts (central exchanges) or the user's wallet (decentralized exchanges)[4].

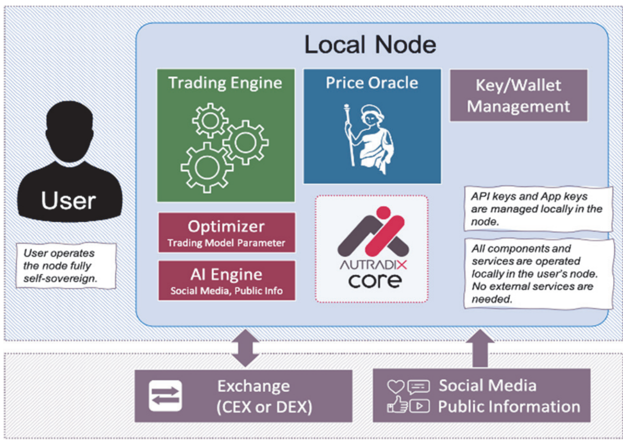

Figure 9: Local Node Architecture

Since the node is in full control of the user, he/she is always in full control of his/her assets[5]. For security reasons, API keys should nevertheless be limited in permissions to what is necessary to trade. Similarly, for use with DEX, it makes sense that the wallet is also implemented as a multisig wallet with app keys restricted to perform only the needed trading transaction to grant the trading algorithm only the minimum necessary rights.

Compliance requires no restrictions. The provisions for any central exchange and non-custodial wallets used apply. Beyond that, there are no further restrictions, as the

[3] Compliance requirements still need to be evaluated in detail

[4] see figure 9

[5] With the restrictions on central exchanges

entire system is always in full control and operation of the user without any custodian services.

3.3.2.1 System Requirements

The following components run on an autradix node:

1. Price oracle (or a remote oracle is used for this).
2. Trading engine
3. Parameter optimizer (including trading simulators for each traded currency pair)
4. AI engine
5. Wallet / Account manager

In terms of memory requirements, the database of the price oracle and the database of the trading engine require the most space. Also, the AI models require some space. Depending on the number of actively used currency pairs and how far into the past the data is kept (as needed for the simulation), memory is required. However, data for one day and one currency pair requires only a few MB. The trained AI models may require some hundred MB.

Fast SSDs are advantageous to speed up read performance and shorten the simulation time. Alternatively, in-memory databases are well feasible at this size and advantageous for performance reasons.

The parameter optimizers are by far the most computationally intensive components, since these adaptively calculate the optimal parameterization of the trading algorithm and must calculate some hundred or thousand simulations for each objective function value.

Since this optimization does not have to be performed in real time, it does not require very high computing power. A multi-core environment is beneficial, since the function evaluations can be parallelized well.

In order to operate a node effectively, it must have a permanent internet connection to retrieve information (price oracle, account information) and execute trades.

The aim is to optimize the software for use on a single-board computer (e.g., Raspberry Pi 4) or other small computers in order to be able to implement cost-effective nodes.

Also, the implementation as module for the DAppNode[6] is a possible approach. Then the user can operate his autradix node as subtask at this node.

Alternatively, an autradix node can be set up on a cloud server. In contrast to an own hardware node, it is very easy to secure a 24/7 operation and performance requirements are easily met even with small configuration packages.

3.3.3 DeFi Fund

Another way to make the autradix system accessible and thus usable to users is a decentralized fund into which users can invest and then participate in the gains of the fund according to their shares.

In contrast to the approaches described as centralized service or as decentralized independent node, in which in both cases only the user's assets are managed and traded, this approach manages a portfolio that contains and controls the assets of all users.

In figure 10 the architecture of the decentralized DeFi fund is visualized.

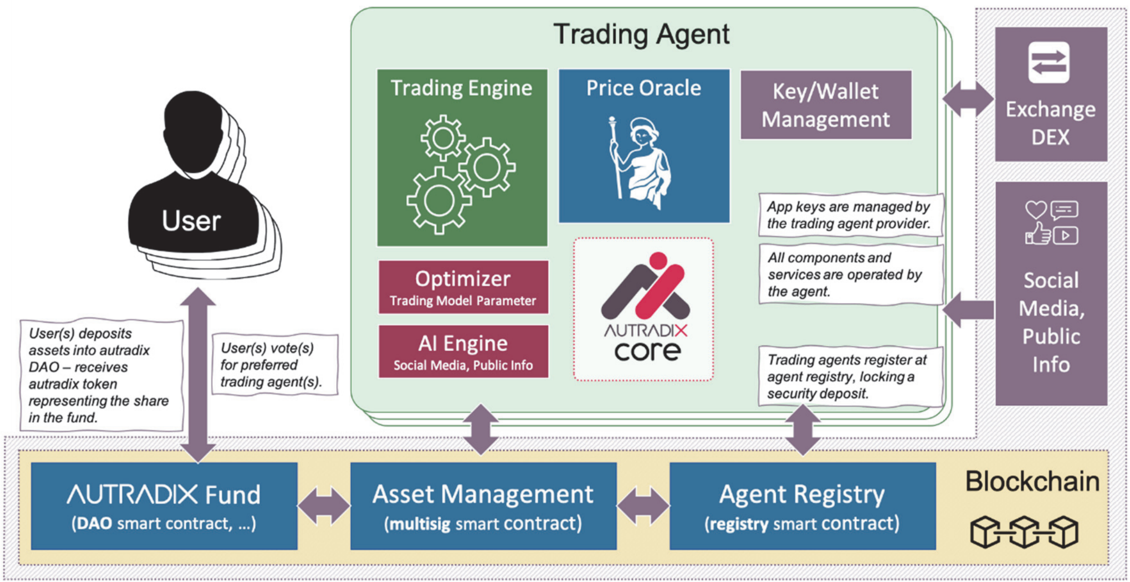

Figure 10: Decentralized Fund Architecture

[6] see https://dappnode.io/

3.3.3.1 Asset Management

The assets are managed in a smart contract. Users deposit their assets in the smart contract and release them for trading. Upon deposit, the user's share in the value of the overall portfolio is determined and allocated to the user in the form of tokenized shares (share tokens are mined upon deposit).

If a user wishes to withdraw his share of the portfolio, the share tokens are burned and the equivalent value of the share is transferred back to the user.

A suitable oracle must be used to determine the total value of the portfolio, as other ERC20 tokens are also part of the portfolio in addition to Ether and their value cannot otherwise be determined on-chain.

Suitable waiting periods until the deposited assets are traded or until withdrawn assets can be released must be defined and transparently disclosed so that oracle manipulations, attacks such as frontrunning, etc. can be prevented.

3.3.3.2 Trading

Only decentralized exchanges can be used for trading the assets from the fund, since otherwise an account would have to be used for trading on a centralized exchange, in which the assets are held in custody. However, since the entire protocol is to be designed as trustless as possible, this will not be an option.

The assets are managed in a multisig with special app keys, which only allows the trading agents the necessary permissions to swap the assets, but not to withdraw or perform other transfers.

3.3.3.3 Ecosystem and Incentives

The autradix fund ecosystem recognizes the following roles:

Investors – these are the users who pay assets into the fund in order to achieve a good return.

Shareholders – by investing in the fund, the users get voting rights to participate in the strategic decisions of the DAO, especially the selection of trading agents.

Trading agents – these are the operators of the trading bots who want to get a good return for providing the IT service to the fund.

The entire system operates in a fully decentralized manner and does not require any central instance, operator, or coordinator. The system is also trustless[7] which means that individual users (regardless of their role) do not have to trust other users.

Each role has an economic incentive to behave honest and according to the rules of the ecosystem. Fraud or misbehavior is already prevented at the protocol level or it is strongly motivated by crypto-economic incentives to behave correctly, since otherwise it would lose its security deposit or not receive revenue, for example.

3.3.3.4 Trading Agents

A trading agent is an off-chain software that can manage and trade assets similar to a SaaS bot or local node. A centralized service could be implemented for this, but would in turn make the whole system no longer trustless, not to mention that the operation of such a custodian trader is complicated from a compliance point of view (requires appropriate regulation for crypto trusteeship) and would be a single point of failure as well. The approach also works fully decentralized and trustless through algorithmic and crypto-economic protection.

The decentralized system allows trading agents to register on the platform at the agent registry. For security, a deposit must be made, which will be slashed if the agent can be proven to be acting maliciously. A newly registered agent cannot trade for the time being, but must first be elected by the token holders. This is done at regular intervals (epochs).

Elected traders will then receive an asset volume to work with. This volume must be large enough to realize meaningful trades, but still small enough for multiple trading agents to work with.

Each agent can specify how high his maximum trading volume should be and it will be allocated the actual volume depending on its reputation in the election process. As an incentive for operating an agent, the agent receives a percentage of the profits it generates as tokens.

3.3.3.5 Autradix DAO

The investors which have contributed assets to the fund will receive tokens according to their share of the fund balance, which at the same time gives them the corresponding voting weight in decisions, for instance

- Selection and volume approval of trading agents,
- Blocking or exclusion of malicious trading agents, etc.
- Fund strategy

3.3.3.6 Autradix Token

The autradix token is a central element of the DeFi fund. It has the following tasks and properties:

- Fungible token, following the ERC20 or ERC777 standard.
- It represents the countervalue of the assets managed in the DeFi fund. The number of issued tokens always correlates to the value of the fund.

[7] Trust is established by the software protocol only

- It can be redeemed at any time for the corresponding share of the deposit.
- It gives voting rights in decisions of the DAO.
- All payments (e.g., compensation of trading agents) are made in the token.
- A community or development fund can be realized as a percentage of the epoch profit, too. This fund can be used to finance future developments at the decision and permission of the DAO.
- The token is stable to the value of the portfolio. It can be traded on exchanges, but will always have the equivalent value of the share of the fund portfolio. This stability can be hedged algorithmically.
- The token does not represent a share of a company or organization, but it is a utility and governance token that is used exclusively to operate the autradix fund. A legal classification has yet to be made.

3.3.3.7 Compliance and Regulation

The autradix DeFi fund is decentralized and is not operated by a service provider. It is a smart contract running on the public Ethereum blockchain. The investors (users who deposit assets into the smart contract) receive tokens in exchange for their deposit, representing the value of their share in the total fund. The trading agents or other users do not have full control over the assets at any time. The portfolio is managed in a smart contract without a trustee. All relevant decisions are made and approved by the token holders in the form of a DAO.

Even though the token is designed as a utility and governance token, it still represents a store of value with profit expectation. Therefore, it is necessary to examine to what extent it might be subject to trading restrictions.

Necessary regulatory obligations are from a current point of view not different from other already existing DeFi solutions. However, a detailed legal examination is still necessary.

4 Summary and Outlook

With autradix, a trading protocol is presented that makes it possible to automate trading with volatile cryptocurrencies on the basis of an adaptive self-optimizing model and make it accessible to a larger group of users. An adaptive evolutionary optimization algorithm is used to set the parameters and thus the properties of the trading model to best match the user's goals.

With the targeted use of tuned AI models, it is possible to support the mathematical models and to react triggered to predictable events. Information from social media and other public information sources is evaluated and used to identify possible events with an impact on the crypto market in order to adjust the trading strategy accordingly.

With a generalized API, the protocol is not bound to a specific exchange or decentralized protocol, but can be ported and optimized to the trading platforms the user needs.

The protocol can be realized as a cloud service, as a fully self-sovereign node under the user's own responsibility, but also as a decentralized, trustless, and tokenized DeFi protocol. This makes trading as easy as simply buying a token. Investors and operators of trading agents thus come together and can share in the profits without the need for an intermediary, trustee, or coordinator and without having to trust the other participants in the ecosystem.

4.1 Prototypes and possible Roadmap

The parametric model is the basis of the autradix protocol. In PoC-1 the trading model, the price oracle service and the generic interface is implemented.

Based on this, the optimizer with the trading simulators is realized in PoC-2.

With the generic interface and the implementation for one or more exchanges PoC-3 can be achieved.

The continuation of this prototype allows the Local Node to be realized and used practically (release 1). Building on this, the cloud service can be developed (release 2).

In parallel or as the next development step, the decentralized DeFi fund can be implemented in release 3.

References

[1] S. Kux: Hybride Optimierungsstrategien für komplexe technische Aufgabenstellungen. University of Applied Sciences Mittweida, Master Thesis, Mittweida, 2011.