

# Gimmer Technical Paper

Persio Flexa

January 15, 2018

## Contents

1	Introduction	4
2	The Problem	4
3	The Solution	4
4	General Description	4
5	Key Features	4
6	The DAPP	5
6.1	User accounts . . . . .	6
6.2	Dashboard . . . . .	6
7	Strategy Details	6
7.1	Informations . . . . .	6
7.2	Strategy . . . . .	6
7.2.1	Configuration . . . . .	6
7.2.2	Base . . . . .	7
7.2.3	Assets . . . . .	7
7.2.4	Indicator mode . . . . .	7
7.2.5	Period . . . . .	7
7.3	Bot Structure . . . . .	7
7.3.1	Indicator Interval . . . . .	7
7.3.2	Safety Interval . . . . .	7
7.3.3	Indicators . . . . .	7
7.3.4	Safeties . . . . .	7
7.3.5	Simulator . . . . .	8
7.4	Balances . . . . .	8
7.5	Activities . . . . .	8
8	Bot Store	8

9	Exchange Trade	8
10	Margin Trading	8
11	Lending	9
12	Smart Portfolio	9
13	Profile Matchup	9
13.1	Example portfolio allocations . . . . .	9
14	Copy Trader	10
15	Arbitrage	10
16	Exchange API Integrations	10
17	Technologies	10
17.1	Local Storage . . . . .	10
17.2	Public Storage . . . . .	11
17.3	Frameworks . . . . .	11
18	Technical Details	11
18.1	TradeManager Class . . . . .	11
18.2	Backtest Class . . . . .	11
18.3	Trade Class . . . . .	11
18.4	Context Object . . . . .	11
18.5	IIndicator Interface . . . . .	12
18.6	IndicatorManager Class . . . . .	12
18.7	ISafety Interface . . . . .	12
18.8	SafetyManager Class . . . . .	12
18.9	IBot Interface . . . . .	12
18.10	PrimaryBot Class . . . . .	12
18.11	MarginBot Class . . . . .	12
18.12	LendingBot Class . . . . .	12
18.13	ArbitrageBot Class . . . . .	13
18.14	PriceManager Class . . . . .	13
18.15	CacheManager Class . . . . .	13
18.16	IExchange Interface . . . . .	13
18.17	Period Enum . . . . .	13
18.18	CurrencyPair Class . . . . .	13
18.19	ResultManager Class . . . . .	13
18.20	BotStoreManager Class . . . . .	13
18.21	PositionManager Class . . . . .	14
18.22	Portfolio Class . . . . .	14

18.23	ConfigManager Class . . . . .	14
18.24	LogManager Class . . . . .	14
19	Securities	14
19.1	User Account . . . . .	14
19.2	API KEYS . . . . .	15

## 1 Introduction

The challenge of creating bots to trade in your behalf is worth taking. Based on what we have produced and achieved to date and the support give by our community we are on the right path to become a global reference.

Earlier in 2017 we launched our Alpha to a group of people to validate it. During that period we discovered many things that could be improved to enhance our product and the eagerness of our users made us take the next steps.

## 2 The Problem

Because everything is on the same server, we started having problems with the limit of REST calls to the major exchanges, which resulted driving the price of the service up.

Another problem we faced was the computing, which is necessary to run the back-tests. We even had to limit the display dates as to distribute better the service across all users.

All of this combined made the growth of the platform put us in a complicated situation to overcome.

## 3 The Solution

It was then that we found in the midst of the growth of the ICOs, with that, beyond capitalizing our project, we had the potential to decentralize our platform. Everything that we compute could be brought to the user's machine and therefore bring the full potential of our Application.

With our own coin, we are going to create a healthy ecosystem inside the platform, and with that make so that the user is non dependable on our servers. Everything that should be of public knowledge will be stored in the Ethereum network and private information will stay in the user's machine. The renting and everything that will be charged for inside the platform will be due in GMR.

## 4 General Description

The Gimmer Decentralized Application (DAPP) is planned for release in March 2018.

We love open source apps and we want Gimmer to expand to this territory, the speed at which we achieve this goal will ultimately depend of the success of Gimmer Token Sale.

## 5 Key Features

- A standard automated crypto-trading bot is offered for free. This includes the use of one indicator, one safety and one pair, without leverage and money invested

limit.

- Automated crypto-trading bots that are quick and easy to set up with our own trading strategies
- Earn GMR tokens by renting your strategies to other customers through the bot store
- Rent bots that use tried and tested strategies of other traders
- Easy to set up with no programming skills required
- Multi-coin trading allowing you to select multiple coins to trade against
- Free backtesting allowing you to test your strategy over a previous trading period to see how it would have performed based on historical data
- Simulator mode allowing you to trial a strategy in real time, without real currency so the user can get confidence
- Add from over 80 indicators to your strategy to maximise your trading
- Add from over 10 safeties to your strategy to limit or prevent any losses
- Choose between different trading modes
- Arbitrage trading, make profits by trading the different coin prices between exchanges.
- Social trading network so you can learn from, discuss and connect with other traders
- Automated crypto-lending - we are creating a special bot to make loans, and invest in the currency that has the most deficit, thus, earning a greater interest. The loan bot is almost zero risk, as it lends cryptocurrencies to traders who use the Exchange and does not allow withdrawal, the loans are backed by the lenders own cryptocurrency.
- Configurator - Onboarding process to identify the user's risk profile
- Lightweight decentralized application DAPP - secure, fast and cost efficient for customers

## 6 The DAPP

The DAPP will provide the possibility to create many local accounts. Each account will have state of the art encrypted password protection.

## 6.1 User accounts

- Dashboard is the main panel which displays overall balances and the activity of bot strategies.
- There is a chart to show the performance of currency pairs.
- Gimmer will provide a recommendation of profitable strategies to be rented based on the user's risk profile.
- Gimmer bots show what they are "thinking" by indicating their actions via the trading band, such as when the bot is about to open a position or about to close a trade.
- Users can easily create a new bot strategy via the Dashboard.

## 6.2 Dashboard

- The bots are user friendly and simple to configure without any knowledge of programming required.
- Users can create a trading strategy from scratch or use a predefined profitable template as a starting point then customise freely.
- When you create a bot, the available strategy types will automatically be displayed for you to choose from.
- You can backtest your trading strategy using historical data to see how it would have performed.
- Users will be able to choose from various types of bot strategies such as Trade, Margin Trading, Multicoon, Lending and Investment Portfolio. We are also working on more variations and new markets.

# 7 Strategy Details

A strategy has 4 tabs of information:

## 7.1 Informations

You can edit the bot name, description and select the Exchange or even delete your bot strategy.

## 7.2 Strategy

### 7.2.1 Configuration

Trade Mode Trade Margin Trading

## 7.2.2 Base

You can select the asset base, the cumulative asset that you want as a base. This will vary between exchanges.

## 7.2.3 Assets

Having the Multicoins feature, users can select as many assets as they wish, so the bot will analyse and make a decision based on the most profitable assets to be traded at that specific moment.

## 7.2.4 Indicator mode

First Trade: when an indicator indicates a favourable position the bot will invest all money on that trade. Diversified Trade: when any indicator indicates a position the bot gets a proportion of the available funds to trade. All Trade: when all indicators indicate a position the bot gets all the funds and executes the trade.

## 7.2.5 Period

The period used for trade, can be daily, hourly or per minutes, the indicators use this property to calculate the position seen on the trade band.

## 7.3 Bot Structure

### 7.3.1 Indicator Interval

Interval that the bot checks the exchange server. This can be daily, hourly and by the minute.

### 7.3.2 Safety Interval

Interval that the bot checks if the trade is profitable or not and makes a decision. The safety interval can be triggered daily, hourly or every minute.

### 7.3.3 Indicators

Gimmer will have over 100 indicators to choose from, all of which can be configured. This offers the user greater control and a vast number of possibilities and performances.

### 7.3.4 Safeties

Protect the capital and prevent dumps and save the profit when the bot is profiting.

### 7.3.5 Simulator

All bots can be tested in real time to validate the strategy. This uses live market data without using real cryptocurrency.

### 7.4 Balances

This describes the amount that was started and the current amount, it also shows the indicator target, sentiment and the current status.

### 7.5 Activities

Show all activities for the bot, the list and chart of Long, Short and Close positions. Gimmer has a centralized bot store, users can share their bot strategies for free, or they can set their own price and rent their strategies to other users.

## 8 Bot Store

- Users that publish their strategies on the bot store will have their trading profiles made accessible to other users. Users can then review and rate bot strategies based on their performance.
- All strategies will be approved by the Gimmer team before being displayed on the Bot Store to maintain quality and ensure the store has profitable strategies.
- All variations of bot strategies can be published on the store.
- The Store has filters to enable users to find the right strategy for them.
- Filters will enable strategies to be displayed based on the user's risk profile.
- The users can backtest any strategy found in the store to validate strategies before making a decision.

## 9 Exchange Trade

Will work without leverage, the max level available will depend on the balance and the max percent for works defined by the user. All exchanges have this functionality.

## 10 Margin Trading

Gimmer will work with leverage margin if set by the user, the max level available will depend on the Exchange being traded on. Margin trading is only available on Exchanges that support this functionality.

## 11 Lending

- Gimmer Lending bots will select the asset with the most profitable daily rate for investing. The bot will then automatically place funds on the exchange for lending.
- Lending is available only on Exchanges where this functionality already exists.

## 12 Smart Portfolio

- Portfolio's are made up from a collection of bot strategies which combined can potentially maximise gains and minimise losses.
- The portfolio can be smart or static, the smart portfolio protects the account and sells when the market goes down.
- Renting a portfolio is suitable for anyone, but in particular those with limited knowledge or background in trading.

## 13 Profile Matchup

When a user creates their account, Gimmer will assess how risk adverse a user is, this will be done via an onboarding questionnaire. The risk assessment will later be used to match users with appropriate trading, investing and lending strategies.

### 13.1 Example portfolio allocations

- Conservative Investor

The portfolio will be composed of bots that have low risk.

70% Lending bot.

30% Trade bots.

- Moderate Investor

The portfolio will be composed of bots that have medium risk.

50% Lending bot.

40% Trade bots.

10% Margin trading bots.

- Aggressive/Risky Investor

The portfolio will be composed of bots that have high risk.

30% Multicoins Trading bots.

50% Margin Trading bots.

10% Lending bot.

10% Arbitrage bot.

## 14 Copy Trader

- Traders can make any of their portfolios or strategies public, if they do, other users can copy their strategies and rent the same bots.
- Details of the strategy are not made public, neither is the profit generated.
- High level information such as start date, running period, currency pairs and percent gained is visible to all users.
- User can look inside a public portfolio and choose to copy one or more of the individual strategies from within it.

## 15 Arbitrage

The arbitrage bot is only available for the Exchanges that are already integrated with Gimmer.

- The bots will instantly transfer cryptos between exchanges, choosing the exchange with the best rates for the given coin, then selling at a profit.
- Arbitrage is the only service that will require Gimmer to access and withdraw fund from a users exchange wallet.
- For users wishing to automatically trade using arbitrage, a strong anti-virus and security package will be required on the their computer.
- User wishing to benefit from the arbitrage bot will need to accept special terms stating that the user is aware of the potential risks involved.

## 16 Exchange API Integrations

Gimmer will be continually expanding our partnerships and connections with the major exchanges, such as: BITFINEX, POLONIEX, KRAKEN, OK COIN, BITTREX and many more. Our aim is to introduce at least one new exchange every month.

## 17 Technologies

### 17.1 Local Storage

Everything that won't be shared will be stored only locally on the user's machine.

## 17.2 Public Storage

Every piece of datum that is needed to be public will be made so by utilizing the Ethereum Network, with Smart Contracts.

This way, we will give the power to the user to choose what he agrees should become public and in what moment, to publish his profile and rent his strategies at his own time.

## 17.3 Frameworks

We use Electron and Angular for the development of the DAPP as they are consolidated frameworks, also making it possible to utilize the same node modules in which the application was built on top of, not being necessary great work with conversion.

# 18 Technical Details

Technical details of the development

## 18.1 TradeManager Class

It is the class that manages the calls of all the strategies based on their respective periods, running at separate times indicators and safeties. This is where the GMR will be charged if the bot takes action, if the bot does nothing, the user does not spend anything. All bot types go through this manager.

## 18.2 Backtest Class

This class when recognized by TradeManager is automatically populated with test values to check the past market, here it multiplies this bot multiple times each with a period, the context is passed from one test period to the other to keep the results to be used in the test, so it is possible to simulate how that strategy behaved in the past in a solid way.

## 18.3 Trade Class

When the bot is run derived from this class, calls with API Keys are necessary because the bot must make real calls, the context is filled with its balance, OHLCV and many other information that is described below, all necessary informations is provided by the context.

## 18.4 Context Object

The context is a global variable (Facade Pattern) that contains everything related to a strategy to use, OHLCV, bot information, balance and actions that have already been performed, history, among others.

## 18.5 IIndicator Interface

All indicators are derived from the class IIndicator, which contains the fundamental interface for running the indicators, for all the indicators is passed the context bringing the OHLCV for the execution of the same. The strategy must take action depending on the return of your call, which may be buy, sell or close. (buy indicator) means for the strategy to keep buying, (sell indicator) to sell and (close) to close if it has any open position and keep in standby.

## 18.6 IndicatorManager Class

It is he who manages all the indicators and how they will interact, there is a property that talks about whether they will act together, separate, or each with a quantity of assets.

## 18.7 ISafety Interface

This interface is used for all classes of safeties, it has only one standard method that is to say if the operation should be terminated. stop is a boolean property, when this property is true the position that is open is terminated.

## 18.8 SafetyManager Class

It should take care of all safeties selected by the user, checking in an orderly manner and leaving free for future implementation of new methods.

## 18.9 IBot Interface

Interface for generic types of bots, such as ExchangeTrade, MarginTrading, Lending, Arbitrage, Portfolio and several others that should derive from this interface that is used by the TradeManager class.

## 18.10 PrimaryBot Class

Executes Exchange Trade, using only buy and sell information provide by indicators, close indication is not used on this type of bot.

## 18.11 MarginBot Class

Run Margin Trading using the buy info for LONG and sell for SHORT and close to sell and keep in standby.

## 18.12 LendingBot Class

If Exchange has API enabled for lending, the bot searches for which currencies offer the best rate to invest and invests in the one with the best interest return.

### 18.13 ArbitrageBot Class

The arbitrage bot must assume 2 types of situations, the situation in which can triangulate the price difference within the exchange itself, the more coins the better. Another type of situation is checking the exchange between the exchange that the user enter their keys. So the bot looks for advantages by taking the rates to have the gain in that price difference.

### 18.14 PriceManager Class

Whatever is selected by the user within the strategy will be displayed in real time for the consumption of GMRs for that setting, it will show how many GMRs it will spend daily or monthly. This helps the user to get an idea dynamically of everything they will spend.

### 18.15 CacheManager Class

It is a necessary class to store information that can be shared between the bots to avoid excessive calls in the exchanges, this is very useful to avoid punishment in the exchanges and to do the IP never to be blocked.

### 18.16 IExchange Interface

This interface is responsible for standardizing the methods of the exchanges within the platform, which unifies the API returns for the platform to always converse with the same result type.

### 18.17 Period Enum

In it you will find all the periods that the bot must work, but it is limited by what the exchanges return because it is necessary to display the same periods as that which is exported in the OHLCV.

### 18.18 CurrencyPair Class

A class useful for doing conversions from pair object to string and vice versa.

### 18.19 ResultManager Class

This class takes care of all the results that are executed inside the bot, saving all the cumulative data from the beginning of the execution of the bot.

### 18.20 BotStoreManager Class

The store will take care of displaying all the bots published by the users, the approval of the bots must be done by the store administrators, for a bot to be published the user

must have ethers in his wallet, the ethers are necessary for payment of the GAS at the moment publish a bot in the store. To rent a bot you need to have ETH and GMR in your wallet, ETH to pay the GAS and GMR to send the payment to the owner of the bot (70%) and 30% to the Store. The bots that the user places in the store and are rented from their owner will automatically receive the GMRs in their wallet. The user can do whatever they want with this GMRs that have been earned, use inside the Gimmer or withdraw and sell.

### 18.21 PositionManager Class

Take of showing the current status of the position that was opened, passing information on how much you are earning or losing in real time, so the safeties have everything they need to know about the current open position.

### 18.22 Portfolio Class

It is the class that informs the balance of the user's portfolio related to the current bot, in it is informed the starting money, the current money and the percentage compared to the beginning of the execution of the bot.

### 18.23 ConfigManager Class

All user-set configurations for the bot are saved in this class, this class passes which indicators have been enabled, its settings, periods, intervals, safeties intervals and everything else the bot must check to perform its actions.

### 18.24 LogManager Class

Everything that happens with the bot, all its actions, warning and errors will be stored in a local file inside the root folder of the system, this file can be easily found by the name of log.txt, it is with this file that we can verify the errors reported by the users and verify the actions of everything that happens inside the system.

## 19 Securities

### 19.1 User Account

The user account is based on ethereum account, all currencies are tied to this wallet and everything is linked to this account. All sensitive data will be encrypted (aes-256). The password will never be stored under any circumstances, requiring the user to save it to itself and backend the keystore that will be provided by the DAPP. Thus it is necessary for the user to enter his password every time it closes and reopens the application.

## 19.2 API KEYS

The KEYS as all sensitive data will be encrypted (aes-256) using the user's password, and will not be saved locally decrypted in any way.