TeddyDoge

Introduction

WHITE PAPER v1.0

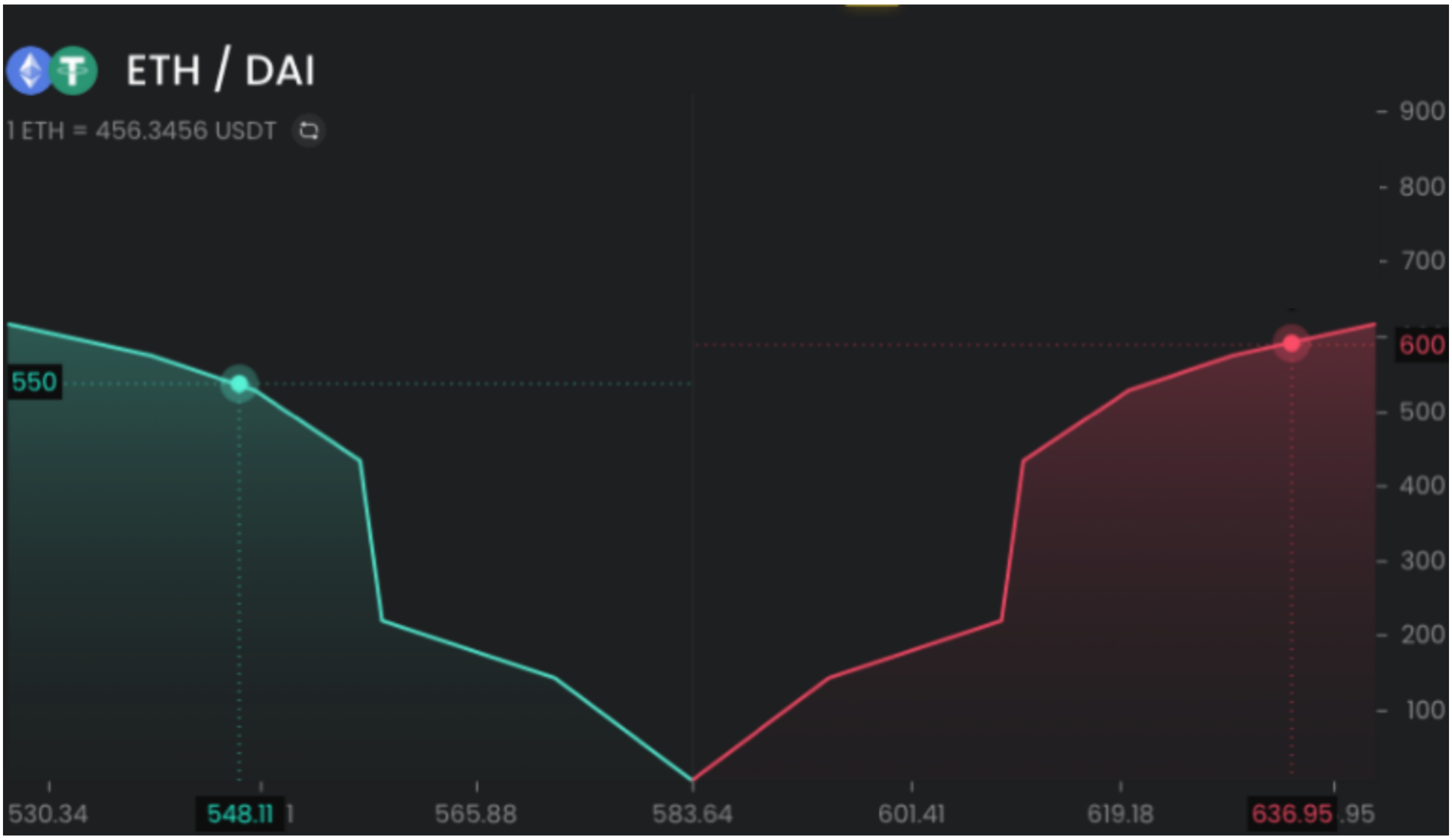
What is TeddySwap?

Teddyswap is a community-driven organization built to solve what might be called the "One-stop full-featured decentralized exchange." Teddyswap progress is intended to create a broader range of network effects. Rather than limiting itself to a single solution, Teddyswap intertwines many decentralized markets and instruments. Thus far, the core products, which will be described in more detail here, include swap, aggregate transactions,liquidity,Farm,15 mainstream chain transactions, 15 mainstream cross-chain,15 mainstream chain chart system,Token and liquidity multi-scheme locking,TeddyWallet and derivatives. Teddyswap's products are configured in a way that allows the entire platform to maintain decentralized governance of TEDDY token holders, while continuing to innovate on the collective foundations by design. Whereas major structural changes are voted on by the community, the day-to-day operations, rebalancing of pools and ratios, business strategy, and overall development is ultimately decided on by our Teddyswap team.

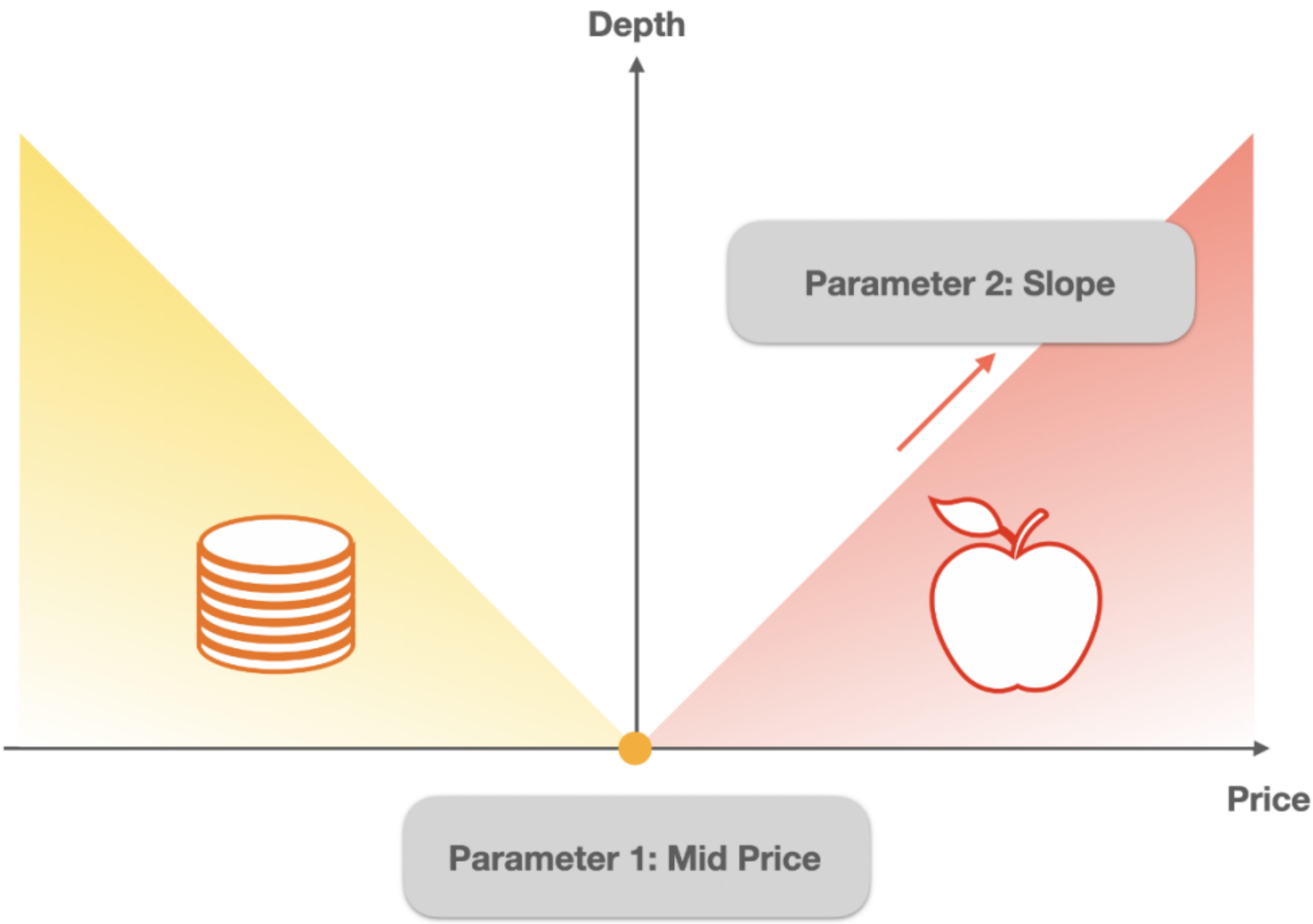
TeddySwap is an automated market-making (PMM) decentralized exchange (DEX) currently on the blockchain.
In addition to DEX, TeddySwap involves a collection of governance, operations and reward contracts that help grow the TeddySwap ecosystem and utilization.

Proactive Market Making Algorithm
PMM: A Universal Liquidity Framework#

Markets contain huge amounts of information that represent buyers and sellers' sentiments and valuation of assets. In essence, a market reacts to changes in available information and requires sophisticated mechanisms to do so efficiently when the amount of data is very large. In a blockchain context, building a mechanism that incorporates all the important information needed for market making and is also able to operate quickly and dynamically within current technological constraints is not an easy task, and requires a prioritization of different kinds of market information.
To keep our market-making algorithm running smoothly and efficiently, we need to boil the vast sea of market information down to its most crucial core metric. So, what is a market's "most important metric"? The answer is liquidity. Liquidity can be graphically represented by a market depth chart, as shown below.



A depth chart consists of two roughly triangular (though not necessarily symmetrical) shapes, representing bids (buy orders) on the left and asks (sell orders) on the right, along the price x-axis and the depth y-axis. The two triangles can be mathematically described by two parameters, mid price and slope, or how “steep” the triangle is.



Let us closely examine the depth triangle on the right hand side first. This is the ask side, where ask (sell) prices are quoted. We can see that the more base tokens are sold, the higher the price. This linear relationship can be captured by the following formula:

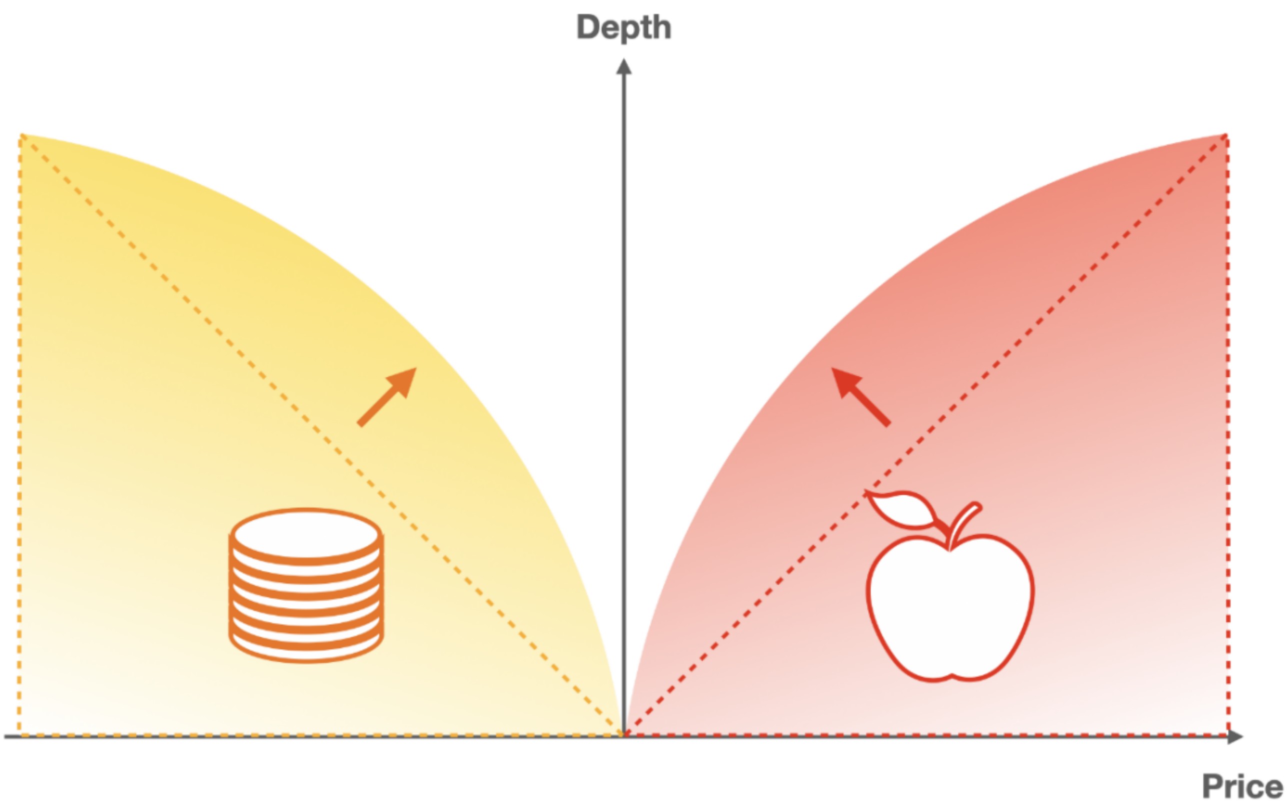
P = i + ik(\frac{B_0-B}{B_0})P=i+ik(B0B0−B)

where i is Parameter 1, the mid price, and k is Parameter 2, the slope. B is the number of base tokens currently in the inventory and B_0 is the initial number of base tokens in the inventory. (B_0-B)/B_0 is the portion of base tokens that have been removed from the ask side due to transactions, relative to the initial base token balance. This formula stipulates that as the number of base tokens that have been traded increases, the base token price rises linearly.

Is this an accurate representation of market reality? Not exactly, as this linear model has two limitations:

1. In practice, most liquidity is concentrated near (immediately above or below) the mid price, because that is the most capital-efficient strategy for market makers. The linear model does not reflect this uneven distribution and is thus an oversimplification
2. The linear model returns a liquidity of zero after the price exceeds or goes below a certain threshold. However, in reality, no matter how favourable the quoted price is (e.g. for ETH/USDC, a bid order at $100 and an ask order at $1,500), there is liquidity present at that price. This model fails to take such scenarios into account

Therefore, we need to make this pricing curve/depth chart nonlinear to align it with market patterns, but we also don’t want to introduce additional parameters. How should we go about doing that?



We want to make the depth chart nonlinear to depict the fact that depth is more concentrated in the vicinity of the mid price.
Mathematically, the most obvious and straightforward solution is to change the addition in the aforementioned linear formula to multiplication, like this:

P = i(\frac{B_0}{B})P=i(BB0)

In this formula, P increases as B decreases, and it also doesn't have an upper or lower bound (technically it has a lower limit of 0, but a subzero price doesn't make sense anyway). But what about the slope? The solution is to refactor the B_0/B term and add a new parameter k that we can use to control the magnitude of the change in price due to B.

P = i(1-k + k\frac{B_0}{B})P=i(1−k+kBB0)

When B_0/B >= 1, P is directly proportional to B_0/B in the previous formula, but in this new formula, k dictates the extent of which P is affected by B_0/B. More specifically, k is in the range [0, 1] and governs the slope of the pricing curve.
When k = 0, the formula becomes P = i, so the price does not change regardless of other parameters.
When k = 1, the formula reverts back to (2).
When k is in (0, 1), as k increases, so does the price elasticity, meaning that the price becomes more sensitive to changes in base token quantity (i.e. B). Conversely, as k decreases, the price elasticity also decreases.
This model seems sufficiently complete to cover all scenarios, but there is another issue. In a transaction, the total amount of tokens that needs to be paid is the area under the pricing curve, so we will have to take the integral of the curve, but the curve formula above makes this calculation cumbersome as B_0/B introduces a logarithmic term during derivation. To make computation easier, we square the B_0/B term to eliminate all instances of log:

Incredibly, when k = 1, this curve is identical to the AMM bonding curve. This reaffirms our belief that this algorithm has captured the essence of market activities and patterns.

Similarly, without loss of generality, we apply the same derivation procedure for the bid side depth chart, substituting base tokens with quote tokens (denoted by Q) and using division instead of multiplication. We get:

P=i/(1-k+(\frac{Q_0}{Q})^2k)P=i/(1−k+(QQ0)2k)

Combining both formulae, we get the proactive market maker (PMM) pricing formula, described in mathematical terms below.

P_{margin}=iRPmargin=iR

Where RR determined by the following formula:

if B<B_0, then
R=1-k+(\frac{B_0}{B})^2kifB<B0,thenR=1−k+(BB0)2k

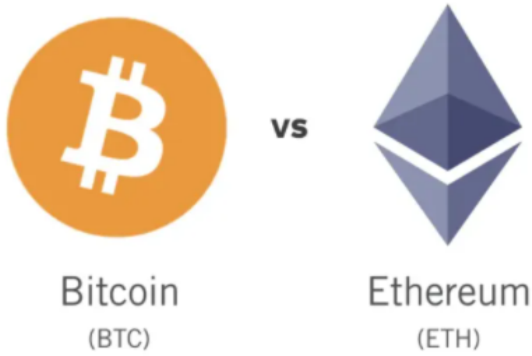
if Q<Q_0, then
R=1/(1-k+(\frac{Q_0}{Q})^2k)ifQ<Q0,
thenR=1/(1−k+(QQ0)2k)

else R=1elseR=1

The PMM algorithm is a “high-fidelity” abstraction of the orderbook-based market, defined and regulated by a handful of simple parameters, but it is also highly flexible and optimized for on-chain operations.
We will now enumerate several promising use cases for PMM that can be achieved by fine-tuning parameters and instituting different withdrawal/deposit rules.

Use case1#
Proactive market making with external price guidance

For mainstream assets, such as BTC and ETH, external markets have much higher volumes and are thus a price source for other platforms from which to retrieve market prices. PMM is capable of proactively adjusting these fetched mid prices to minimize impermanent loss (IL) and achieve higher capital efficiency than AMM platforms. This mechanic also means unlocking single-token liquidity provision — market makers are not forced to deposit tokens Uniswap-style.



The configurations required for this use case are:

- Mid price i is set to the price retrieved from external sources.
- Parameter k is set to below 1.
- Everyone is given the single-token liquidity provision option.

Use case2#
Low barrier-to-entry automated market making

This use case mainly applies to long-tail asset markets (i.e. predominantly newly issued assets with little sell-side liquidity on AMM platforms). PMM can help these assets with the initial liquidity they desperately require for their long-term growth and sustainability. With PMM, asset issuers do not need large amounts of capital on standby to pair up with their assets when initializing liquidity pools. For instance, if a team wants to issue their token X on PMM, they have the option to initialize liquidity with 100% X and 0% stables or ETH. This drastically reduces the barrier-to-entry for smaller projects.

In this use case, PMM gives the pricing power to takers entirely — makers have no control over the price discovery mechanic whatsoever.

The configurations required for this use case are:

- Mid price i is set to the initial offering price designated by the asset issuers.
- Parameter k can be set to any arbitrary number in [0, 1].
- The first liquidity deposit can be made in arbitrary proportions, and it does not change the price.
- All subsequent liquidity deposits and withdrawals must be made in proportion to the current pool ratio (i.e. similar to Uniswap liquidity pools).

Use case3#
Fully customizable and free market making

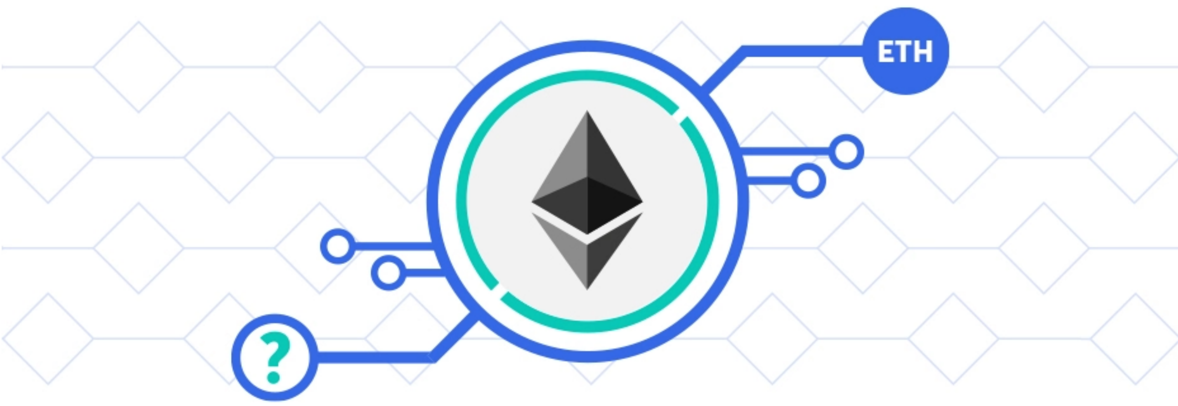
This use case is intended for experienced and ambitious market makers (both institutions and individuals), who want the highest degree of freedom and customizability possible to execute their own market making strategies. In this use case, all liquidity in the liquidity pools belongs to the market makers themselves and they also have full control over all the pool parameters. Market makers can dynamically adjust the asset price by changing these parameters based on their assessment of market sentiment, valuation, and other factors. Moreover, market makers can deposit to and withdraw from these liquidity pools in arbitrary ratios, without affecting the asset price.

For a more concrete example, a ETH/USDT market maker in this use case can choose to market-make near ETH=700USDT with a very small k in order to provide highly competitive liquidity and earn considerable transaction/swap fees from trading activity. When the market maker foresees or predicts an increase in ETH price, they can then react accordingly by removing some ETH from their pool to reduce their market risk exposure. This maneuver does not affect the liquidity on the USDT side, however, so trading activity can continue as usual.

This use case also applies to issuers of new assets, who can choose to only deposit the tokens they are issuing, without any capital (e.g. ETH, USDT, or other stablecoins). They can set the initial offering price and a small k to ensure low price elasticity, so that the token price does not fluctuate too dramatically due to the influx of trading activity. This design also means that when token issuers need capital for development and operations, they can simply withdraw capital from the liquidity pool without affecting the sell-side liquidity.

The only configuration required for this use case is that:

- Deposits/Withdrawals are set so that only market makers (owners/creators of the pools) are allowed to perform such operations.
- Single-token liquidity provision/removal is allowed.



PMM Core Concepts
Base & Quote Tokens#

Base and quote are two concepts that will be mentioned frequently. Two easy ways to distinguish between them are:

- In a trading pair, the base is always the token before the hyphen, and quote is after it
- In transactions, teh price refers to how many quote tokens are needed in exchange for one base token

For example, in the ETH-USDC trading pair, ETH is the base token and USDC is the quote token.

PMM Parameters#

The funding pool of PMM is described by four parameters:

B_0B0: base token regression target - total number of base tokens deposited by liquidity providers

Q_0Q0: quote token regression target - total number of quote tokens deposited by liquidity providers

BB: base token balance - number of base tokens currently in the pool

QQ: quote token balance - number of quote tokens currently in the pool

PMM Pricing Formula#

The PMM price curve is plotted by the following pricing formula:

P_{margin}=iRPmargin=iR

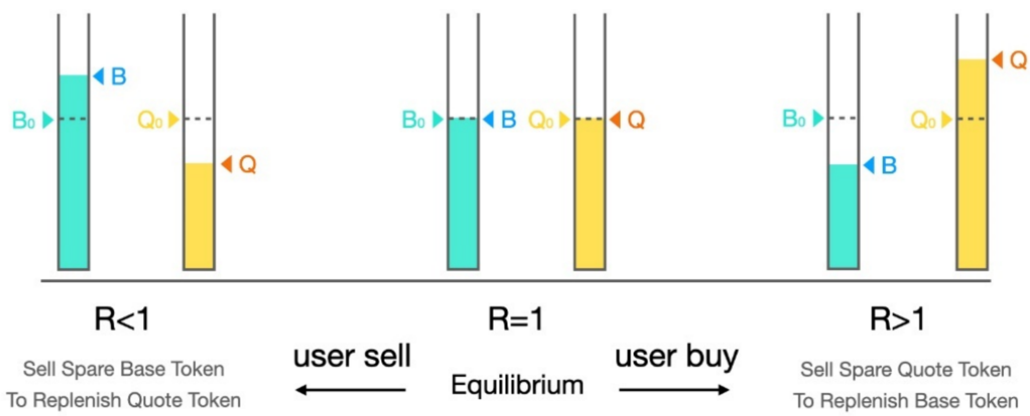
Where RR is defined to be the piecewise function below:

if \ B<B_0, \ R=1-k+(\frac{B_0}{B})^2kif B<B0, R=1−k+(BB0)2k
if \ Q<Q_0, \ R=1/(1-k+(\frac{Q_0}{Q})^2k)if Q<Q0, R=1/(1−k+(QQ0)2k)

else \ R=1else R=1,
ii is the market price provided by an oracle, and kk is a parameter in the range [0, 1].

The Three Possible States in PMM#

At any given time, PMM is in one of three possible states: equilibrium, base token shortage, or quote token shortage.



Initially, i.e. prior to any transaction, the capital pool is in equilibrium, and both base tokens and quote token are at their regression targets. That is, B=B_0B=B0 and Q=Q_0Q=Q0.

When a trader sells base tokens, the base token balance of the capital pool is higher than the base token regression target; conversely, the quote token balance is now lower than the quote token regression target. In this state, PMM will try to sell the excess base tokens, lowering the base token balance and increasing the quote token balance, in order to move this state back to the state of equilibrium.

When a trader buys base tokens, the quote token balance of the capital pool is higher than the quote token regression target; conversely, the base token balance is now lower than the base token regression target. In this state, PMM will try to sell the excess quote tokens, lowering the quote token balance and increasing the base token balance, in order to move this state back to the state of equilibrium.

The parameter RR in the pricing formula above assumes a critical role in facilitating this regression process. The more the capital pool deviates from the equilibrium state, the more RR deviates from 1. When the price given by the PMM algorithm deviates from the market price, arbitrageurs step in to help bring the capital pool back to the equilibrium state.

Liquidity Provider Fee#

A small transaction fee is charged for every trade. This fee is called the liquidity provider fee and is distributed to every liquidity provider proportionate to their stake in the capital pool.
More specifically, liquidity provider fees are collected from what buyers receive and distributed to liquidity providers who supply this kind of asset to the capital pool. In other words, liquidity providers are rewarded in the same asset denomination.
For example, when traders buy ETH tokens with USDC tokens, liquidity provider fees will be charged in the form of ETH tokens, and these tokens will be distributed to the liquidity providers who deposited ETH tokens into the capital pool.
When traders sell ETH tokens for USDC tokens, liquidity provider fees will be charged in the form of USDC tokens, and these tokens will be distributed to the liquidity providers who deposited USDC tokens into the capital pool.

Maintainer fee#

A maintainer fee is also collected from what buyers receive, and is directly transferred to the maintainer. The maintainer may be a development team, a foundation, or a staking decentralized autonomous organization (DAO).
Currently, the maintenance fee on Teddy is 0.

Withdrawal Fee#

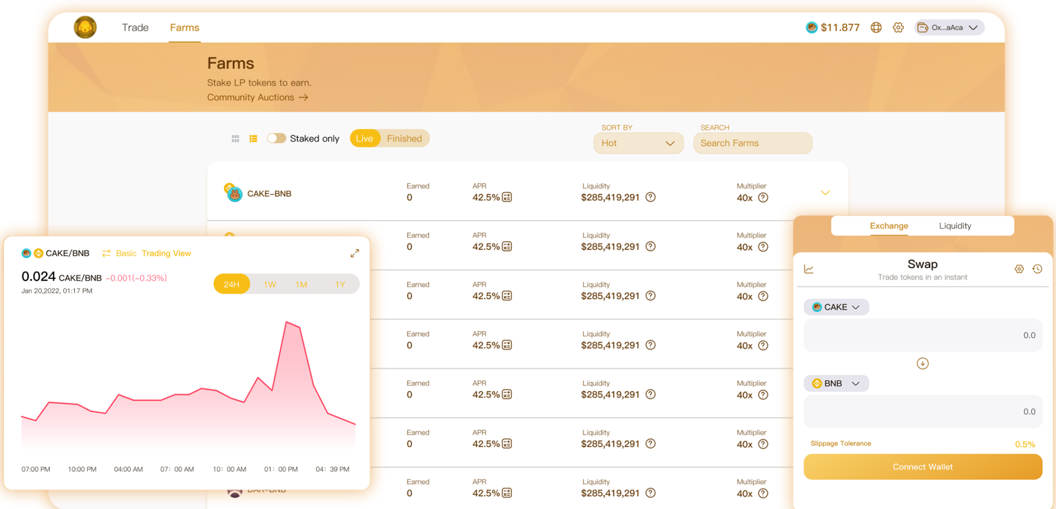
A withdrawal can change the PMM price curve and may harm the interests of other liquidity providers. Teddy charges a withdrawal fee from liquidity providers who withdraw their assets and distribute it to all remaining liquidity providers.

IMPORTANT

Normally, the withdrawal fee is 0 or an extremely small percentage (<0.01%) of what you withdraw. The withdrawal fee will increase significantly only if the funding pool suffers from a serious shortage of either base or quote tokens and liquidity providers intend to withdraw the type of token which is in short supply.
The withdrawal fee serves as a protection mechanism for liquidity providers who maintain their supplies of liquidity and contribute to the sustainability and overall health of the Teddy platform.

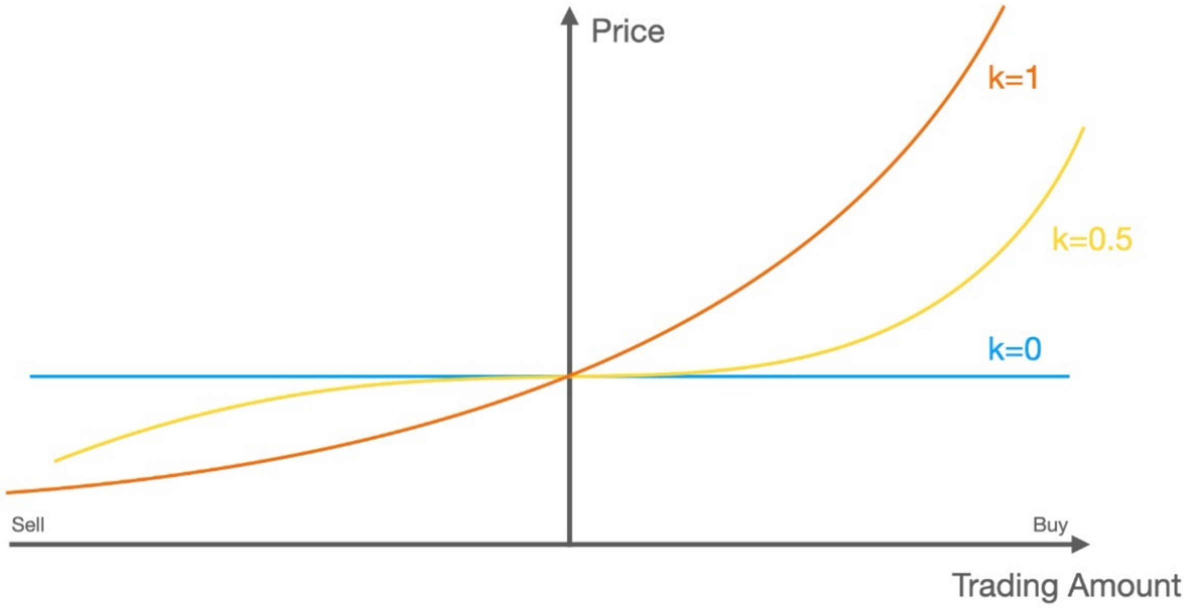
Deposit Rewards#

Rewards will be distributed to those who make a deposit of base or quote tokens when the capital pool faces a shortage of that type of token.
In the next section, we will explain the math behind these core concepts.



Flexibility and kk the "Liquidity Parameter"#

Last but not least, we will introduce the Teddy's "liquidity parameter", kk. The parameter kk gives Teddy the flexibility to handle different market situations.



When kk is 00, Teddy naively sells or buys at the market price, as shown by the flat, blue line. As kk increases, Teddy’s price curve becomes more “curved”, but, consequently, liquidity becomes increasingly jeopardized, because more funds are placed far away from the market price and are thus underutilized or not utilized at all. When kk increases to 11, the flat section near the market price is completely eliminated and the curve essentially becomes the standard AMM curve used by Uniswap. Normally, kk is recommended to be a relatively small value, such as 0.10.1, which could provide liquidity 10 times better than the standard AMM algorithm.

The Math Behind PMM

Core PMM#

The core of PMM is essentially calculating one integral and solving two quadratic equations.

The Price Curve Integral#

For traders, the most important thing is the average transaction price. The average transaction price is the integral of the marginal price P_{margin}. Let's take the base token shortage scenario as an example.

$$\Delta Q = \int^{B_2}_{B_1} P_{margin} dB$$
$$= \int^{B_2}_{B_1} (1-k)i + i(B_0/B)^2 k dB$$
$$= i(B_2-B_1)*(1-k+k\frac{B_0^2}{B_1B_2})$$

This tells the trader how much they should pay if they buy **B_2-B_1** base tokens.
Rearranging the equation above, the average transaction price is thus:

$$P=\frac{\Delta Q}{B_2-B_1}=i*(1-k+k\frac{B_0^2}{B_1B_2})$$

We found that the average transaction price is only dependent on the state of the system before and after the transaction, so the price calculation methods for both buying and selling are the same: integrating P_{margin}.

Solving the quadratic equation for trading#

Without the loss of generality, the integral becomes the following when there is a shortage of quote tokens:

\Delta B = \frac{1}{i}(Q_2-Q_1)*(1-k+k\frac{Q_0^2}{Q_1Q_2})ΔB=i1(Q2−Q1) (1−k+kQ1Q2Q02)

Let's derive how to calculate the price when there is a shortage of quote tokens and only the number of base tokens you want to buy or sell (i.e. \Delta BΔB) is given. Now that \Delta B, Q_0, Q_1ΔB,Q0,Q1 are given, we need to calculate Q_2Q2, which is found by solving a quadratic equation. Transforming the equation into standard form:

(1-k)Q_2^2+(\frac{kQ_0^2}{Q_1}-Q_1+kQ_1-i\Delta B)Q_2-kQ_0^2=0(1−k)Q22+(Q1kQ02−Q1+kQ1−iΔB)Q2−kQ02=0
let \ a=1-k, \ b=\frac{kQ_0^2}{Q_1}-Q_1+kQ_1-i\Delta B, \ c=-kQ_0^2let a=1−k, b=Q1kQ02−Q1+kQ1−iΔB, c=−kQ02

Because Q_2>=0Q2>=0, we discard the negative root, and so

Q_2=\frac{-b+\sqrt{b^2-4ac}}{2a}Q2=2a−b+b2−4ac

It can be proven that:
When \Delta B>0ΔB>0, Q_2>Q_1Q2>Q1; trader buy base tokens, and should pay Q_2-Q_1Q2−Q1
When \Delta B<0ΔB<0, Q_2<Q_1Q2<Q1; trader sell base tokens, and will receive Q_1-Q_2Q1−Q2
When \Delta B=0ΔB=0, Q_2=Q_1Q2=Q1.
Teddy focuses on verifying the special case of k=0, and k=1 to support a constant selling price and the bonding curve of the standard AMM.

Solving the quadratic equation for regression targets#

When the system is not in the equilibrium state, changes to the oracle price can bring profit or loss. For example, assume that shortage of base tokens is the current state, and then the oracle price goes up. It is clear that the excess quote tokens cannot buy enough base tokens to return the base token balance to the base token regression target. Thus, LPs who deposited base tokens will suffer a loss. Conversely, if the oracle price drops, the excess quote tokens can buy more base tokens, causing the base token balance to exceed the base token regression target, and LPs who deposited base tokens will make a profit.
In summary, the regression target is influenced by the oracle price. To calculate the regression target at a certain oracle price, we make the following derivation:

Given \Delta Q = i(B_2-B_1)*(1-k+k\frac{B_0^2}{B_1B_2})ΔQ=i(B2−B1) (1−k+kB1 B2B02)

Since we are doing regression, B_2=B_0B2=B0. Rearraging the equation with respect to B_0B0 gives

\frac{k}{B_1}B_0^2+(1-2k)B_0-[(1-k)B_1+\frac{\Delta Q}{i}] = 0B1kB02+(1−2k)B0 −[(1−k)B1+iΔQ]=0

The negative root does not make sense and is discarded, so B_0B0 is:

B_0=B_1+B_1\frac{\sqrt{1+\frac{4k\Delta Q}{B_1 i}}-1}{2k}B0=B1+B12k1+B1 i4kΔQ−1
In this case, \Delta Q=Q-Q_0ΔQ=Q−Q0. It can be proven that, when \Delta Q \ge 0ΔQ≥0, B_0\ge B_1B0≥B1.

This fact is extremely important, because it ensures that the base token balance and the quote token balance will never be greater than the regression target simultaneously, or less than the regression target simultaneously. This means that PMM will only switch between the three states discussed in the Core Concepts section.
Similarly, the formula for quote token regression target Q_0Q0 is
Q_0=Q_1+Q_1*\frac{\sqrt{1+\frac{4k\Delta B i}{Q_1}}-1}{2k}Q0=Q1+Q1 2k1+Q1 4kΔBi−1

Peripheral#

This section will deal with the math pertaining to the peripheral functioning of PMM.

Trades#

As mentioned above, the regression target depends on the oracle price, and the price curve in turn depends on the regression target. We should therefore calculate the regression target for each trade well in advance to fix the price curve.
In addition, since the price curve given by PMM is segmented, if a transaction involves different states (for example, when a trader sells an astronomical amount of base tokens during a base token shortage and forces the state into a quote token shortage), the price needs to be calculated in segments as well.
Please be advised that this calculation requires a high degree of accuracy. The smart contract provides six trading functions for the three possible states.

Deposit#

Depositing and withdrawing base tokens when there is a shortage of base tokens, or quote tokens when there is a shortage of quote tokens, will change the price curve. This requires us to process the deposit and withdrawal with caution and care in order to keep the capital pool sustainable and fair.
Let's analyze what happens when an LP makes a deposit when there is a shortage of base tokens.
According to the calculation formula of B_0B0 derived above,

B_0=B_1+B_1*\frac{\sqrt{1+\frac{4k\Delta Q}{B_1 i}}-1}{2k}B0=B1+B1 2k1+B1 i4kΔQ−1

After an LP deposit bb base tokens, B_1B1 increases by bb, and B_0B0 increases more than than bb's magnitude. This means that this deposit helps make a profit for all LPs who provided base tokens. The reason is that the deposit makes the price curve smoother, and the same amount of \Delta QΔQ can now buy more base tokens.
In this case, as soon as the LP makes a deposit, the LP makes a profit. This is referred to as the deposit reward. The essential source of this reward is the slippage paid by the trader who made the system deviate from the equilibrium state.

Withdrawal#

Similarly, after an LP withdraws bb base tokens, B_1B1 decreases by bb, and B_0B0 decreases by more than bb's magnitude. This withdrawal causes all LPs who owe base tokens to suffer losses. This is because this withdrawal makes the price curve more steep, and the excess quote tokens have less purchasing power in terms of base tokens.

The PMM algorithm stipulates that a withdrawal fee is required to withdraw tokens in this case. The magnitude of the fee is equal to the aggregate loss of all LPs caused by the withdrawal. This fee will be directly distributed to all LPs that have not yet withdrawn.

Factoring in the deposit reward from the previous section, if an LP makes a withdrawal immediately after depositing, the withdrawal fee will be greater than the deposit reward, thus eliminating any possibility of risk-free arbitrage trading.

It is worth noting that both the deposit reward and withdrawal fee are only significant when the system deviates very far from the equilibrium state and the deposit/withdrawal amount is large. Traders thus often overlook the existence of this gain or loss. Of course, traders are also welcome to extract value from the system by exploiting this if they so wish. In order to do that, they can first deposit to earn deposit rewards when the system deviates from the equilibrium, and then withdraw once the system returns to the equilibrium to avoid the withdrawal fee.

Rode Map

2021

August

- Start building

November

- Complete the exchange code

2022

January

- Complete your farm code
- Create a public social channel on the 15th
- Private placement completed on the 20th

February

- Coins issued on February 3
- Completion of cmc ICO on the 5th
- Community ICO completed on February 6
- Liquidity added to PC and teddyswap for the first time on February 7th
- The farm officially opened on February 15th

March

- On March 15th, the cross-chain Mainnet will be added to 500 million to teddyswap

April

- On April 15, cross-chain Polygon, added to 500 million to teddyswap
- Completed TeddyLock lock system on April 30

May

- On May 15, cross-chain Fantom, added to 500 million to teddyswap

June

- On June 15, cross-chain Avalanche, added to 500 million to teddyswap
- Completed TeddyTools chart integration website on June 20

July

- On July 15, cross-chain Arbitrum, added to 500 million to teddyswap

TeddyDoge

2023

September

- On September 15th, cross-chain xDai, added to 500 million to teddyswap

August

- On August 15th, cross-chain Optimism, added to 500 million to teddyswap
- Build TeddyWallet wallet system

October

- On October 15th, cross-chain Moonriver, added to 500 million to teddyswap
- Complete the TeddyWallet wallet system

November

- On November 15, cross-chain Harmony, added to 500 million to teddyswap

December

- On December 15th, cross-chain Celo, added to 500 million to teddyswap

January

- On January 15th, cross-chain Optimism, added to 500 million to teddyswap

February

- On February 15th, cross-chain cronos, added to 500 million to teddyswap

March

- On March 15th, the cross-chain OEC was added to 500 million to teddyswap

April

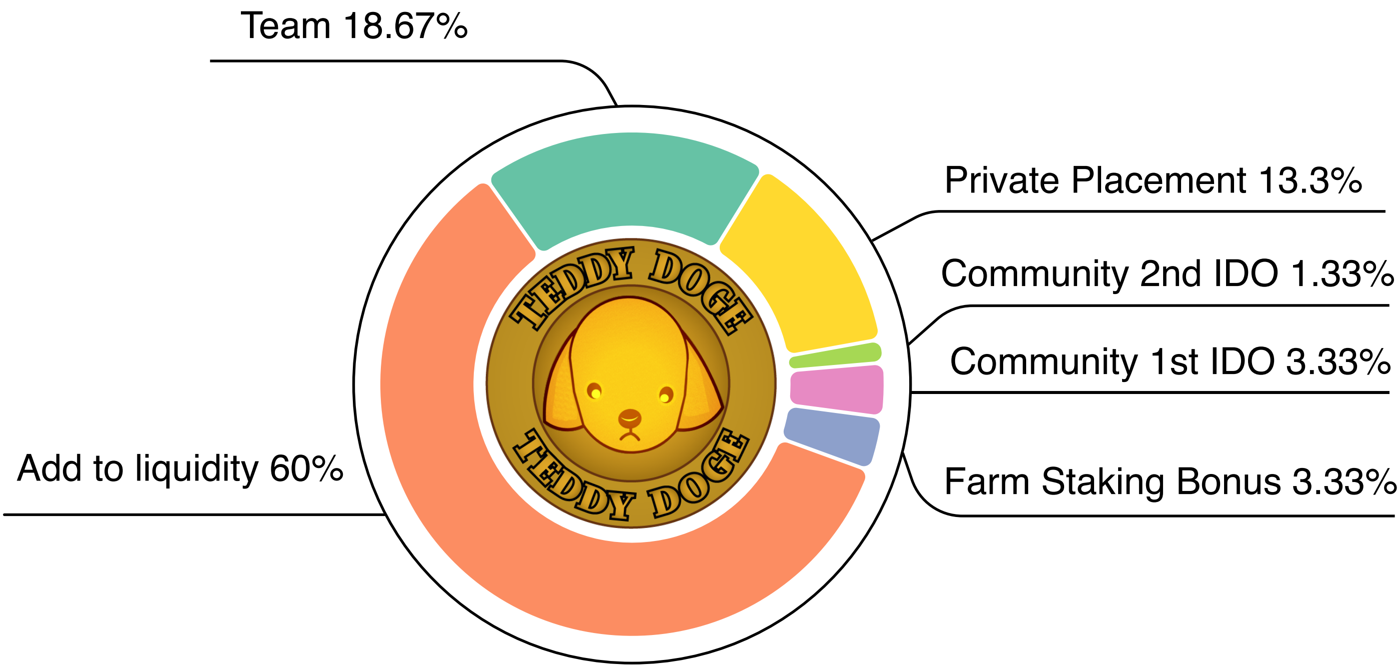
- On April 15, cross-chain heco, added to 500 million to teddyswap

Construction started in August 21
Completed the exchange code in November 21
Complete your farm code by the end of 1/22
15.1.22 Create a public social channel
Completed private placement on January 20, 2022
2 billion private placement, 20 private placement units, lock-up for 30-90 days, lock-up and address announcement. The total private equity fund is 300,000 USDT.
Issued on February 3, 22
Completed community 1st IDO on February 1, 2022
The IDO fund of 500 million community 1nd is 100,000 USDT, and the issue price is 0.0002$. The maximum subscription for each unit is 500 USDT.
Completed community ICO on February 6, 2022
The IDO fund of 200 million community 2nd is 100,000 USDT, and the issue price is 0.0005$. The maximum subscription for each unit is 500 USDT
2/7/22 First added liquidity to PC and teddyswap
1 billion pancakeswap BSC + 200,000 USDT locked for 3 months. The addition date is February 7th. Issue price 0.0002$.

The farm officially opened on February 15, 2022
3/15/22 Cross-chain Mainnet, added 500 million to teddyswap
4.15.22 Cross-chain Polygon, added to 500 million to teddyswap
Completed TeddyLock lock system on April 30, 2022
5/15/22 Cross-chain Fantom, added to 500 million to teddyswap
6/15/22 Cross-chain Avalanche, added to 500 million to teddyswap
6/20/22 Complete TeddyTools Chart Integration Website
7/15/22 Cross-chain Arbitrum, added to 500 million to teddyswap
8/15/22 Cross-chain Optimism, added to 500 million to teddyswap
In August 22, the TeddyWallet wallet system was established
9/15/22 Cross-chain xDai, added to 500 million to teddyswap
10/15/22 Cross-chain Moonriver, added to 500 million to teddyswap
10/30/22 Complete TeddyWallet wallet system
15.11.22 Cross-chain Harmony, added to 500 million to teddyswap
12/15/22 Cross-chain Celo, added to 500 million to teddyswap
15.01.23 Cross-chain Optimism, added 500 million to teddyswap
2/15/23 Cross-chain cronos, added to 500 million to teddyswap
3/15/23 Cross-chain OEC, added to 500 million to teddyswap
4.15.23 Cross-chain heco, added to 500 million to teddyswap
Reserve 1 billion for other cross-chain exchange liquidity additions

Token Economics

15 billion total issuance TeddySwap
Token distribution plan: 9 billion liquidity, 2 billion private placement, 2 billion team, 0.2billion Community 2nd IDO, 500 million community ICO, 500 million farm pledge reward.



9 billion liquidity added
1 billion pancakeswap BSC + 200,000 USDT locked for 3 months. The addition date is February 5th. Issue price 0.0002$.
After 1 month, the cross-chain Mainnet will be added to 500 million to teddyswap
1 month later, cross-chain Polygon, added to 500 million to teddyswap
1 month later, cross-chain Fantom, added to 500 million to teddyswap
1 month later, cross-chain Avalanche, added to 500 million to teddyswap
After 1 month, cross-chain Arbitrum, added to 500 million to teddyswap
1 month later, cross-chain Optimism, added to 500 million to teddyswap
1 month later cross-chain xDai, added to 500 million to teddyswap
After 1 month, cross-chain Moonriver will be added to 500 million to teddyswap
1 month later, cross-chain Harmony, added to 500 million to teddyswap
1 month later, cross-chain Celo, added to 500 million to teddyswap
1 month later, cross-chain Optimism, added to 500 million to teddyswap
Cross-chain cronos after 1 month, added to 500 million to teddyswap
After 1 month, cross-chain OEC will be added to 500 million to teddyswap
1 month later cross-chain heco, added to 500 million to teddyswap

2 billion private placement, 20 private placement units, lock-up for 30-90 days, lock-up and address announcement. The total private equity fund is 300,000 USDT.

2.8billion held by the team, and the address will be announced.

The 500 million community 1st IDO funds are 100,000 USDT, and the issue price is 0.0002$.

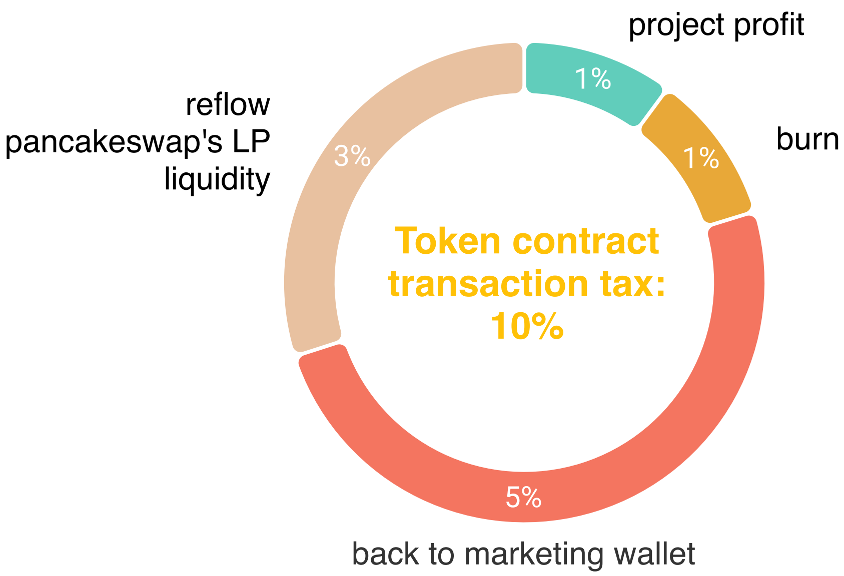
The 200 million community 2nd IDO funds are 100,000 USDT, and the issue price is 0.0005$.

500,000,000 is used as farm staking bonus, and 500,000 is automatically injected into the farm reward every day. 1000 days of injection is completed.

Reserve 1 billion for other cross-chain exchange liquidity additions.

Token contract transaction tax: 10%

5% return to the marketing wallet for the promotion of traffic websites such as cmc dextools.
3% reflow pancakeswap's LP liquidity
1% project profit
1% burn



Token Lock

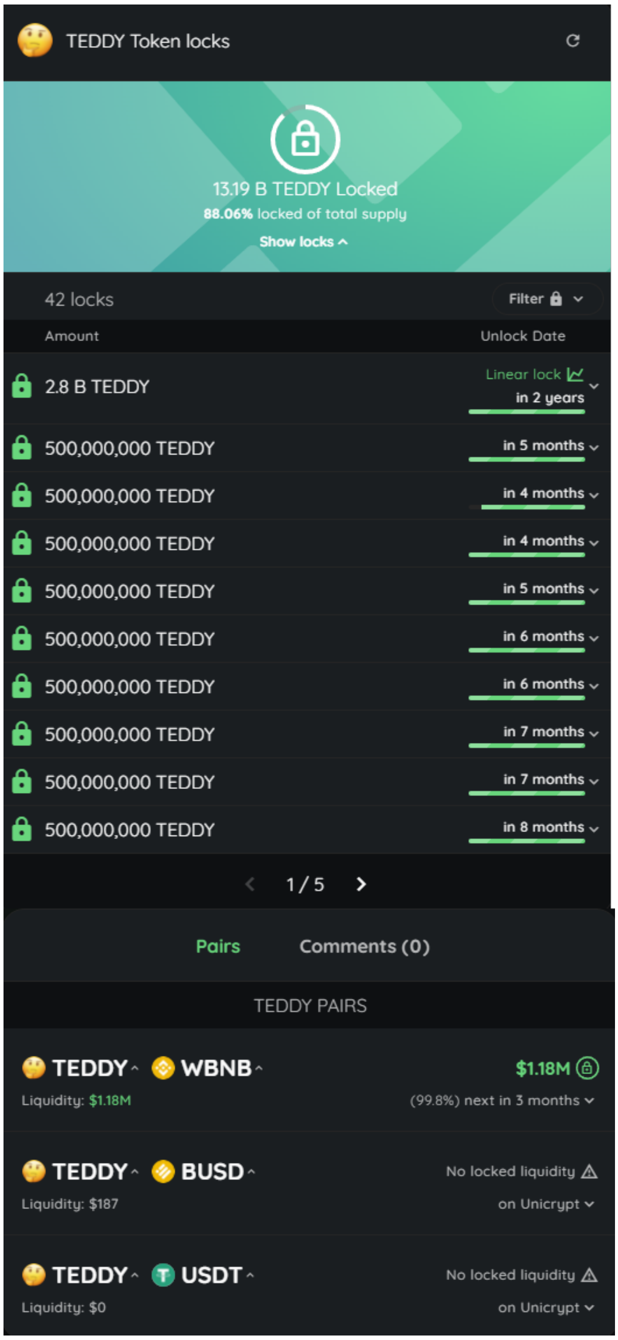
Executive Summary ：The total amount of locked tokens was 13.19 billion, accounting for 88.06% of the total tokens, including ICO locks, farm locks, non newly added liquidity token locks and team held locks.

ICO ：Two billion private placement, 20 addresses of private placement units, lock up for 30-90 days.

•Amount 25 million Unlock date:
31 May 2022 31 May 2022
31 May 2022 31 May 2022

•Amount 50 million Unlock date:
23 April 2022 01 June 2022

•Amount 100 million Unlock date:
30 April 2022 31 May 2022
09 June 2022 11 June 2022
01 July 2022 01 July 2022
08 August 2022 01 September 2022
01 September 2022 15 September 2022
31 December 2022 30 January 2023
01 April 2023 08 May 2023
30 June 2023 30 June 2023
30 July 2023 09 August 2023



Liquidity Lock : The liquidity of 1 billion Teddy and US $200000 joined pancakeswap for the first time, and all positions have been locked, with a locking ratio of 99.8%.

Liquidity Unlock data： 22,April,2022

Liquidity/Token Lock Address ： https://app.unicrypt.network/amm/pancake-v2/token/0x10f6f2b97F3aB29583D9D38BaBF2994dF7220C21