Dblue Protocol

Abstract

Dblue is a completely new AMM model that gives significantly better capital usage efficiency for liquidity providers and lower price slippage for traders than other AMM protocols. Instead of providing liquidity in token pairs, Dblue allows investing in single tokens, reuse token liquidity across all the possible exchange pairs and even to share liquidity pools with other native or partner protocols. Together with a dynamic reserve allocation model that protects from liquidity leakage if the price of any of the tokens in the pool changes, advanced farming features and yield pools Dblue is expected to become the most efficient and profitable liquidity protocol.

1 Introduction

Liquidity pool decentralized digital asset exchanges (DEX) powered by automatic market maker algorithms (AMM) are widely used in Ethereum, BNB Chain and other blockchain networks. Most of them are built on constant function market makers (CFMM). The constant product market maker (k = xy) used initially in Uniswap is still the most popular option.

One of the problems that all CFMM face is price slippage. Every swap operation shifts the price making it less profitable. This problem is mostly significant for small liquidity pools or large deals. Several solutions have been proposed for this problem. Liquifi [2] was our first contribution to this - we introduced time-locked swaps that gradually moved liquidity in and out of a pool, reducing price bouncing. The recent step in this direction is Uniswap V3 [1] that introduced a concentrated liquidity feature to reduce price slippage and increase the capital efficiency. Time will show whether this approach is suitable for liquidity providers, but one drawback is obvious - liquidity providers in Uniswap V3 cannot just invest and wait for profits, they now need to actively control their liquidity positions when market price moves.

Here is where Dbluecomes onto the scene with a completely different approach to liquidity management. We propose to change the common principle that liquidity providers must invest into token pairs. Instead, we will allow LPs to simply invest tokens and let a special liquidity management mechanism use these tokens for swaps with controllable slippage.

2 Protocol operation

2.1 Overview

Dblueuses portfolios instead of pairs to hold liquidity. Lists of accepted tokens in portfolios are managed by the community of protocol users. Liquidity providers can invest any of these tokens and acquire shares of the whole token portfolio. With this, LPs agree that their tokens can be freely exchanged within the portfolio so that the actual owned assets will vary over time. By limiting the list of acceptable tokens, we prevent risks of inflating the portfolio with low-quality assets.

There could be several portfolios, e.g. main portfolio, stablecoins portfolio, etc. LPs can also create new portfolios in a decentralized way. These custom portfolios are managed by their creators. The community of protocol users decides whether to accept a specific custom portfolio to be presented in the DblueUI.

Besides token pools, a portfolio defines virtual token pairs. A pair is an algorithmic entity that ties two tokens and defines their exchange price (ratio). A portfolio also defines a base currency - a token within a portfolio that is used to specify exchange prices of all other tokens.

Every pair works as a CFMM pool with virtual liquidity reserves. It means that when a swap operation is performed on a pool, the exchange price is calculated using the constant product AMM ($k = xy$). The amount of virtual reserves depends on existing token reserves in a portfolio and slippage factors of both tokens (see below). This mechanism protects the reserves from leakage if any token in a portfolio rapidly changes its price.

2.2 Providing liquidity to portfolios

LPs can add their liquidity to a portfolio at any time with the following rules:

- A LP can add amounts in one or several tokens in one transaction.
- The deposited token prices (P_{di}) are determined from the following equation: $P_{di} = min\{P_i, MA(P_i)\}$, where P_i is an internal oracle price, $MA(P_i)$ is a long-term exponential moving average (EMA) of the price. If $P_i = 0$ or $MA(P_i) = 0$, the transaction will fail. In this case the initial price can be only set by the portfolio management smart contract. For the base currency of the portfolio $P_{di} = 1$.

If a portfolio is not perfectly balanced, LPs can deposit tokens with a portfolio share less than a target weight specified by the portfolio management smart contract without fees. For a perfectly balanced portfolio LPs can add liquidity in all portfolio tokens at their target weights without fees.
If an LP deposits tokens to a porfolio over the target weights, the following procedure is applied.

- Let N tokens are to be deposited with amounts T_i respectively.
- Total deposit value in the base currency is $D = \sum T_i \cdot P_i$. Let the current price of the portfolio LP token be PLP.
- Acceptable deposit amounts without fees for each non-base token are calculated as $U_i = max\{\frac{(V+D)W_i}{P_i} - R_i, 0\}$, where V is total portfolio value in the base currency, W_i are target token weights, R_i are current reserves of tokens.
- Excessive amounts of non-base tokens are then $S_i = max\{T_i - U_i, 0\}$.
- Perform swaps with fee of S_i amounts of each non-base token for B_i amounts of the base token using P_{di} as an initial price, $B = \sum B_i$. The resulting prices of non-base tokens are PP_i.

- The new deposit price of the rest non-base tokens $(T_i - S_i)$ is $P'_{di} = min\{PP_i, P_{di}\}$.
- Calculate an acceptable deposit amount without fees for the base token as $U_b = max\{(V + D)W_b - R_b, 0\}$, where W_b is the target weight for the base token, R_b is the current reserve of the base token.
- The excessive amount of the base token is $S_b = max\{T_b - U_b, 0\}$.
- Perform a swap with fee of S_b amount of the base token for T_{prt} amount of nominal PRT token that is built from all non-base portfolio tokens proportionally to their adjusted weights $W'_i = \frac{W_i}{\sum W_j}$, where j iterates through all non-base tokens. PRT token price is then $P_{prt} = \sum P_i W'_i$.
- As a result of the above swap, the new PRT token price is

  $$P'_{prt} = \frac{R_{prt}P_{prt} + S_b}{R_{prt} - T_{prt}}$$

  , where $R_{prt} = \frac{V1 - R_b}{P_{prt}}$ is the total reserve of all non-base tokens, V1 is the current total portfolio value after the first part of swaps, R_b is the reserve of the base token.
- Deposit the rest of the base token adjusted to the PRT token price change $(T_b - S_b)\frac{P_{prt}}{P'_{prt}}$ and T_{prt} amount of PRT token at price P_{prt}.
- The final amount of minted LP tokens is

  $$LP_{mint} = \frac{\sum_{non-base}(T_i - S_i)P'_{di} + B + (T_b - S_b)\frac{P_{prt}}{P'_{prt}} + T_{prt}P_{prt}}{PLP}$$

LPs can withdraw their liquidity at any time with the following rules:

- A LP can withdraw amounts in one or several tokens in one transaction.
- The withdrawn token prices (P_{wi}) are determined from the following equation: $P_{wi} = max\{P_i, MA(P_i)\}$, where P_i is an internal oracle price, $MA(P_i)$ is a long-term exponential moving average (EMA) of the price.
- If the last LP wants to withdraw all his assets, he must withdraw them in all tokens that are currently in the portfolio. In this case, the prices and the target reserve amounts are not considered.

If a portfolio is not perfectly balanced, LPs can withdraw tokens with a portfolio share more than a target weight specified by the portfolio management smart contract without fees. For a perfectly balanced portfolio LPs can withdraw liquidity in all portfolio tokens at their target weights without fees.

If an LP withdraws tokens from a porfolio below the target weights, the following procedure is applied.

- Let N tokens are to be withdrawn in exchange to LP_i of the portfolio LP tokens tokens for each token respectively.
- Total withdrawal value in the base currency is $D = \sum LP_i \cdot PLP$, where PLP is the current price of the portfolio LP token.
- Acceptable withdrawal amounts without fees for each non-base token (nominated in the base currency) are calculated as $U_i = max\{R_i P_i - (V - D)W_i, 0\}$, where V is total portfolio value in the base currency, W_i are target token weights, R_i are current reserves of tokens.
- Excessive amounts of non-base tokens in the base currency are then $B_i = max\{LP_i \cdot PLP - U_i, 0\}$.
- Perform swaps with fee of B_i amounts of the base token for S_i amounts of each non-base token using P_{wi} as an initial price. The resulting prices of non-base tokens are PP_i.
- The new withdrawal prices of the rest non-base tokens are $P'_{wi} = max\{PP_i, P_{wi}\}$.
- Calculate an acceptable withdrawal amount without fees for the base token as $U_b = max\{R_b - (V - D)W_b, 0\}$, where W_b is the target weight for the base token, R_b is the current reserve of the base token.
- The excessive amount of the base token is $B_b = max\{LP_b \cdot PLP - U_b, 0\}$, where LP_b is the withdrawal amount of LP tokens for the base token.
- Withdraw $T_{prt} = \frac{B_b}{P_{prt}}$ amount of nominal PRT token at the price P_{prt}, where $P_{prt} = \sum PP_i \cdot W'_i$, $W'_i = \frac{W_i}{\sum W_j}$.
- Perform a swap with fee of T_{prt} amount of PRT token for S_b amount of the base token.

- As a result of the above swap, the new PRT token price is

  $$P'_{prt} = \frac{R_{prt}P_{prt} - S_b}{R_{prt} + T_{prt}}$$

  , where $R_{prt} = \frac{V1-R_b}{P_{prt}}$ is the total reserve of all non-base tokens, V1 is the current total portfolio value after the first part of swaps, R_b is the reserve of the base token.

- The final returned amounts of non-base tokens are

  $$T_i = \frac{LP_i \cdot PLP - B_i}{P'_{wi}} + S_i$$

  .

- The final returned amount of the base token is

  $$T_b = (LP_b \cdot PLP - B_b)\frac{P'_{prt}}{P_{prt}} + S_b$$

  .

The above algorithms help reaching two important goals:

1. Force LPs to make deposits / withdrawals in a way that leads to proper balances of portfolios.

2. Protect from unlimited arbitrage using deposit / withdrawals of single tokens without a price impact.

2.3 Exchange fees

The Dblueprotocol charges an exchange fee (α) from every swap operation (except arbitrage swaps with reduced reserves, see below). The fee consists of the two components:

- liquidity providers' fee - is kept in portfolios and added to liquidity shares of LPs;

- protocol fee - is accumulated and then exchanged for BLUES tokens on surplus auctions.

2.4 Swap operation

Let R_i be a total reserve of token i in the pool. Let P_{ij} is a current exchange price for a pair ij: $P_{ij} = \frac{P_j}{P_i}$, where P_i, P_j are the prices of tokens i and j respectively in the base currency. Then, a swap operation of input amount A_i of token i for token j is performed as follows.
F_i is a token (i) slippage factor that is used to prevent liquidity losses if for some token in the portfolio its market price rapidly changes:

$$F_i = min\{\frac{P_i}{MA(P_i)}, \frac{MA(P_i)}{P_i}\},$$

where $MA(P_i)$ is a long-term exponential moving average (EMA) of the price. Then, the following algorithm is applied:

1. Calculate instant pair reserves r_i and r_j:

   (a) $r_i = min\{R_j \cdot F_j \cdot P_{ij}, R_i \cdot F_i\}$

   (b) $r_j = min\{R_j \cdot F_j, \frac{R_i \cdot F_i}{P_{ij}}\}$

2. E_i and E_j are token portfolio disbalance factors that are used to equalize price impacts for tokens with different portfolio target weights:

   $$E_i = max\{\frac{r_i}{R_i \cdot F_i}, \frac{1}{max\{\frac{W_i}{W_j}, 1\}}\} \; if \; W_j \neq 0, E_i = 1 \; if \; W_j = 0$$

   $$E_j = max\{\frac{r_j}{R_j \cdot F_j}, \frac{1}{max\{\frac{W_j}{W_i}, 1\}}\} \; if \; W_i \neq 0, E_j = 1 \; if \; W_i = 0$$

   where W_i and W_j are target portfolio weights of the tokens.

3. The swap is performed according to the following procedure:

   (a) Calculate an $k = xy$ constant product swap (i → j) with instant reserves r_i, r_j, normal fee and an exact input A_i. Get a token j output $B_j : B_j = r_j - \frac{r_i r_j}{r_i + \gamma A_i}$, where $\gamma = 1 - \alpha$

   (b) If both tokens i and j are not base currencies:

      i. Update a token i price in Oracle contract $P_i' = \frac{R_i \cdot F_i \cdot E_i \cdot P_i}{R_i \cdot F_i \cdot E_i + A_i}$

      ii. Update a token j price in Oracle contract $P_j' = \frac{R_j \cdot F_j \cdot E_j \cdot P_j}{R_j \cdot F_j \cdot E_j - B_j}$

(c) If token i is the base currency:

i. Update a token j price in Oracle contract $P'_j = \frac{R_j \cdot F_j \cdot E_j \cdot P_j + A_i}{R_j \cdot F_j \cdot E_j - B_j}$

(d) If token j is the base currency:

i. Update a token i price in Oracle contract $P'_i = \frac{R_i \cdot F_i \cdot E_i \cdot P_i - B_j}{R_i \cdot F_i \cdot E_i + A_i}$

2.5 External oracles

Dblueprotocol relies on external price oracles for arbitrage operations with reduced reserves. This ensures that reduced reserves will not be possible to use in flash loan attacks on the protocol.

For the sake of transaction fee economy and to prevent certain types of attacks, Dbluedoes not use on-chain price oracles. Instead, the protocol operation requires trusted off-chain agents that monitor token prices on external exchanges (AMM DEX, centralized exchanges, etc.). These agents sign their price feeds using EIP-712 compatible signatures. These signatures must be passed to the Dblueportfolio smart contracts together with pricing data. The smart contracts can then check the validity of the signatures and use the pricing data as external oracle prices.

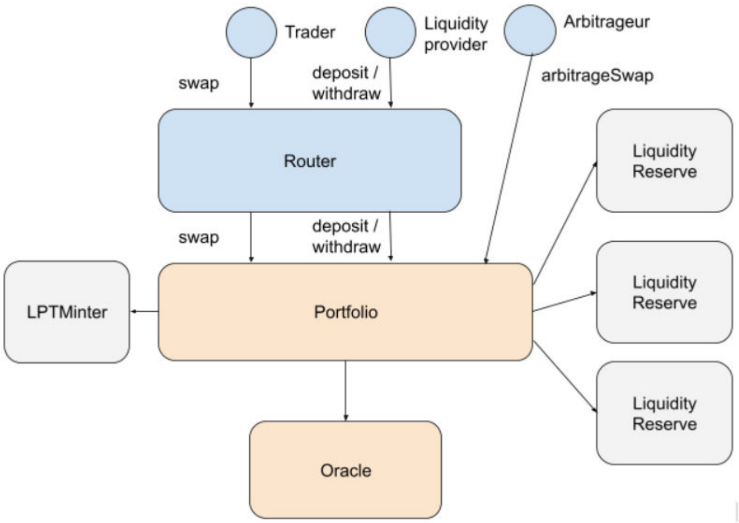

Figure 1: Dblueprotocol architecture

3 Protocol architecture

The main Dblueprotocol smart contracts are: Portfolio, Oracle, LiquidityReserve and LPTMinter. Portfolio smart contract keeps track of tokens in a portfolio that are stored on LiquidityReserve smart contracts. Portfolio is also responsible for trader's, liquidity provider's and arbitrageur's operations. It relies on Oracle smart contract to get and record token prices after each operation. When a LP invests liquidity in a portfolio, he gets ERC20 / BEP20 LP tokens minted by LPTMinter smart contract.
DblueRouter smart contract allows traders to get the best exchange rates performing swap operations across several portfolios and old-style token pairs.

4 Reserve pools reusability

Token pools are designed in a way that allows reserves to be used by several protocols:

- Blueshift DEX protocol
- Other Blueshift native protocols (e.g. Lending)
- External protocols

Let total token reserves be denoted by T_i. By default, token reserves belong to the DEX protocol and $T_i = R_i$. Additional liquidity consumers can request amounts from the reserves with a maximum limit set up per each consumer. When a consumer requests an amount C_i of a token i, it is subtracted from the total reserve and, therefore, $T_i = R_i + C_i$.
Consumers pay an interest rate on the funds borrowed from Dbluereserves. This interest rate is defined individually for each consumer. The earned interests are added to pools in the same way as swap fees and increase profits of liquidity providers.

5 Low impermanent loss and revenues from slippage

Impermanent loss is a negative factor that decreases liquidity provider revenues in most AMM-based decentralized exchanges. The cause of the impermanent loss is in arbitrage operations. Although arbitrage is a necessary component of AMM - as it makes AMM exchange prices stay close to market prices, on the other hand, arbitrageurs earn profits making operations at a better-than-market price, effectively taking these profits from liquidity providers. The outcome is that the greater the price changes, the greater is the impermanent loss for liquidity providers. Of course, if the price ever returns to the initial level, the impermanent loss disappears. However, if a liquidity provider does not wait for this and takes his assets from a pool, the loss becomes permanent.

In Dbluewe propose an ultimate solution to decrease the impermanentloss . The idea is that arbitrage operations should use only a small share of reserves (e.g. 10%). It is possible to achieve this using Dbluevirtual reserves model. Thus, arbitrageurs will get higher price slippage on their operations and therefore cause 10 times less impermanent loss for liquidity providers! But the question is still: why would arbitrageurs agree with this? To ensure that most part of arbitrage operations use reduced reserves, in Dbluesuch arbitrage operations will have zero fee. So, when the market price differs from the AMM price for less than the fee value, this "micro arbitrage" will become profitable while full-reserve arbitrage still has negative output. We expect that in normal market conditions micro arbitrage opportunities should usually precede full-reserve arbitrage opportunities.

Reduced impermanent loss is not the only benefit of the scheme described above. Another important and very profitable result for liquidity providers is that most part of profits from price slippage is not going to arbitrageurs anymore. 90% of profits from price slippage will stay in a portfolio and increase liquidity providers' revenues. Taking into account also the high capital efficiency, we expect that APY for liquidity providers in the Dblue protocol will be able to outperform all the known AMM protocols so far.

Arbitrage operations with reduced reserves may also become a source of attacks when somebody changes the price with high slippage and then returns it back with full reserves. To prevent the possibility of such attacks, we check an external oracle price (for arbitrage swaps only). The arbitrage with reduced reserves is only allowed if it pushes an internal oracle price closer to the external price. Otherwise, the transaction will fail.

6 Portfolio management

As token prices are not tied to liquidity reserves, Dblueprotocol gives an opportunity to manage assets in a portfolio. We can involve a professional portfolio manager controllable by the community of protocol users who will advise to protect a portfolio from low quality assets and to add new assets with high growth potential. The manager can advise changes in a portfolio, then all the Dblueusers can vote for them.
There is a special PortfolioManager smart contract that has the following methods callable after the Dblueusers voting:

- Set target token weights for each portfolio.
- Add a new token to a portfolio.
- Sell all the available amount for some token and remove the token from a portfolio.
- Swap some amount of any token to other token in a portfolio on an external market (e.g. another AMM DEX).

Using these methods, the community of Dblueusers will be able to control portfolio assets according to market conditions and expectations. A professional portfolio manager will get revenues in the Dbluetokens and bonuses depending on portfolio performance for his advices. Thus, Dblue will be the first protocol to offer crypto portfolio investments with additional revenues from DEX fees.
It is also one of the first examples of a decentralized portfolio management solution. Only the community of Dblueusers will be able to control the target weights of tokens in portfolios.

7 Farming features

In the core of the Dbluefarming and staking architecture is the Dblue utility token - Dblue.

As most of the existing DeFi protocols, Dblueoffers the token farming opportunities:

- Liquidity farming - all LPs get shares of BLUES tokens minted every block proportionally to their invested liquidity amounts.
- Yield pools - all BLUES token holders can earn more BLUES tokens or other tokens by staking Dbluetokens.

Minter smart contract is responsible for Dbluetoken mint / burn control according to the logic described above. Farming smart contract allows making liquidity pool token deposits and claiming Dbluetokens. Staking smart contract has two responsibilities: first, it allows staking Dbluetokens and claiming more of Dbluetokens; second, it allows voting for portfolio management proposals. Thus, Dbluetoken staking not only gives an earning possibility, but is also required to participate in portfolio management.

8 Dblue token tokenomics

8.1 Token generation and bridging

Dbluetokens are initially generated as Cardano native tokens. Bridging of Dbluetokens will be supported to:

- Milkomeda blockhain;
- Ethereum blockchain and L2 chains;
- BNB Chain.

8.2 Tokenomics type and limits

The Dbluetokenomics has a hard limit on the max supply. **The maximum supply of Dblue tokens is 100,000,000**.
Part of the max supply (45.67%)
is used to incentivize liquidity providers and Dblueholders for
staking their LP and Dbluetokens.

8.3 Initial Dblue distribution

At the token generation event, there will be the following distribution of Dbluetokens:

- 19.5% - for investors of seed, private rounds and IDO.
- 5% - for strategic partnerships.
- 15% - for the team.
- 2% - for advisors and marketing.
- 7.5% - token liquidity.
- 5.33% - for the community.
- 45.67% - for incentivizing LPs and holders.

8.4 Dblue token emission destinations

The Dbluetokenomics has a flexible token emission scheme with a possibility to distribute the minted tokens every block to several destinations (see fig. 2). The three emission directions are:

- to farms - tokens distributed among all the LP token holders in each farm proportionally to the number of LP tokens considering a farming multiplier. If the farming multiplier of one farm is 1 and another farm is 2, then the second farm gets 2 times more tokens every block.
- to yield pools - tokens distributed among all BLUES token holders proportionally to the number of tokens staked.
- minting decrease - decrease of the emission speed.

8.5 Mint / burn rates

The base minting rate (20 Dblueper block) is fixed and determines the maximum possible token emission speed. The other aspects will be controllable:

- minting decrease rate (Z) that decreases the emission speed;

- additional mint rate to the Protocol development and marketing fund;
- burn rate from services.

Possibility to control the mint / burn rates gives BLUES token a strong mechanism that will allow to find an optimal balance and maintain a stable token exchange price. Moreover, the minting decrease rate Z will grow on a weekly basis so that effective BLUES token emission will go lower over time.

8.6 Dblue token distribution control

An important feature of the Dbluetokenomics is an algorithm that controls token distribution between farms and yield pools:

- If X tokens are sent to farms, Y tokens are sent to yield pools, Z is the minting decrease, then we calculate
  - $X = \beta(20 - Z)$
  - $Y = (1 - \beta)(20 - Z)$
- β is a balance factor that indicates token distribution between farms and yield pools:
  - $\beta = 0.5$ - farms and yield pools get equal token amounts every block;
  - $0 < \beta < 0.5$ - yield pools get more tokens every block than farms;
  - $0.5 < \beta < 1$ - farms get more tokens every block than yield pools.

Using this approach, we incentivize Dbluestaking (and therefore buying)w hen the price is going low and farming when the price is going high. As ar esult, the token price becomes more stable.

8.7 Multi-account minting system

High minting leads to high token price inflation. In other words, high rewards lead to significant token price decrease. As a result if a protocol mints high token amounts as rewards to its users, the users will get almost no value.
Our solution is to connect staking rewards with TVL goals: major part of rewards will be accumulated but not allowed for harvesting unless the Total Value Locked (TVL) of the protocol reaches the goal, remaining part is always allowed for harvesting.
(20 - Z) Dbluewill be minted each block. $W_{tvl0}(20-Z)$ Dbluetokens will be distributed immediately among stakers. $W_{tvlN}(20-Z), N > 0$ Dblueto-kens will be accumulated each block and unlocked for harvesting once specific TVL goals are achieved. $\sum_{N \geq 0} WtvlN = 1$.
The example TVL targets could be the following:

- tvl1: $200 million Total Value Locked
- tvl2: $500 million Total Value Locked

All rewards accrued by the users are distributed immediately across their personal three, yet separated accounts and accumulated. One account is immediately available for harvesting. The other accounts are locked for harvesting until the defined TVL goals are reached.

8.8 Dblue token utility

To summarize, the utility of Dbluetoken includes the following:

1. **Rewards for liquidity providers.** Liquidity providers can stake their LP tokens, received when investing liquidity to Dblueportfolios, in the Farms and earn Dbluetokens every block proportionally to their staked amount.

2. **Staking rewards for token holders.** Dbluetoken holders can stake their tokens in the Yield pools and earn additional tokens.

3. **Portfolio management.** Dbluetokens can be used by the community of Dblueprotocol users to vote on portfolio management decisions:

    (a) add / remove liquidity portfolios;

    (b) add / remove tokens to/from a portfolio;
    (c) change target token weights in portfolios.

4. **Distribution of the protocol fee.** The accumulated protocol fee is exchanged for Dbluetokens on surplus auctions where token holders can bid their tokens to receive the fee. After the auction has ended, the tokens received from the winning bidder will be burnt.

5. **Payments for Dblue services.** Dbluewill offer services to help new projects to launch markets for their tokens: listing services, marketing support, etc. These services will get paid in Dbluetokens. Part of these acquired tokens will go to the team that will actually provide the services. The other part of the tokens will be burnt.

No corporate actions on the Dblue legal entity may be controlled by Dblue token holders.

9 Protocol advantages summary

- For traders:
  - Low price slippage.
  - All possible swap pairs are available between tokens in a portfolio.
- For liquidity providers:
  - Possibility to invest single tokens (not pairs).
  - Get fees from all transactions on the platform, in any token pair.
  - Low impermanent loss.
  - Revenues from slippage.
  - Professional portfolio management.
  - Additional revenues from external protocols.
  - Farming and yield pools.
- For token holders:
  - Controllable token mint / burn schedule will support Dbluetoken price and ensure capital growth.
  - Voting on portfolio management decisions.

10 Deployment plan

10.1 Step 1: Milkomeda blockchain

Milkomeda from DcSpark is an EVM-compatible L2 chain for Cardano. With the launch on Milkomeda, and powered by a unique capital efficient AMM and portfolio manager, Dblueis set to cement its status as a pillar DeFi primitive on Cardano and outclass industry standard AMM DEX primitives.

10.2 Step 2: Ethereum L2 chains

To continue the Dblueexpansion, we will launch on L2 chains / rollups for Ethereum (e.g. Arbitrum). This will set a path to future release on Ethereum / Ethereum 2.0 mainnets.

10.3 Step 3: Support other EVM-based blockchains

Dbluecan operate on any Ethereum Virtual Machine compatible chain. We consider BNB Chain and other EVM-based chains in our deployment plans as well.

References

[1] Hayden Adams, Noah Zinsmeister, Moody Salem, River Keefer, and Dan Robinson. Uniswap v3 core. `https://uniswap.org/whitepaper-v3.pdf`.

[2] Igor Struchkov, Alexey Zubritskiy, Igor Mikhalev, and Bihao Song. Liquifi protocol. `https://liquifi.org/pdf/whitepaper-liquifi.pdf`.

Disclaimer

This paper is for general information purposes only. It does not constitute investment advice or a recommendation or solicitation to buy or sell any investment and should not be used in the evaluation of the merits of making any investment decision. It should not be relied upon for accounting, legal or tax advice or investment recommendations.