Ethervista Whitepaper

The Ethervista Standard: Rethinking Decentralized Exchange Dynamics for Sustainable Blockchain Growth

Vitalik Nakamoto

August 26, 2024

Abstract

Automated Market Makers (AMMs) currently encounter a notable challenge – they struggle to effectively encourage the long-term success of blockchain projects. The problem arises because token creators are encouraged to prioritize profits by rapidly withdrawing liquidity and discreetly selling tokens. Liquidity providers tend to prefer short-term commitments, withdrawing and selling their liquidity as token value rises. This misalignment in structure hampers the growth of projects designed for long-term success. Addressing this challenge is vital, as the current AMM model lacks the necessary incentives to foster continuous growth and resilience in the blockchain ecosystem.

1 Introduction

The current AMM standard incurs a 0.3% fee on every swap paid in tokens. The Ethervista standard is the first to set a custom fee paid only in native ETH. This fee is cleverly distributed among all liquidity providers and token creators within a specific pool, every swap using a novel mechanism which allows us to distribute rewards to millions of users with minimal gas cost. The creator fee is a protocol fee which can be assigned to a smart contract and treasury. Various use cases include auto-buys, staking rewards and many other DeFi applications. A key feature to this model is that market makers and creators benefit from transaction volume rather than token price, incentivizing longevity over short-sighted price action. Investors benefit from a delayed liquidity removal mechanism, preventing developers from quickly rug-pulling. This approach not only mitigates the risk of sudden market upheaval but also enhances the overall success of their investments.

Eventually, Ethervista is set to expand beyond traditional pools, moving towards building ETH-BTC-USDC pools to offer lending, futures, and fee-less flashloans, aiming to become an all-in-one decentralized application

2 A Technical Overview: The mathematics behind the EtherVista model

As previously mentioned, every swap collects a native ETH fee which is distributed among the liquidity providers and the protocol. There are four uint8 fee variables which must be initialized for every pool corresponding to the fee distribution for both the buy and sell transactions. These variables correspond to USDC amounts for which the corresponding ETH fee is calculated on every swap using an on-chain oracle. For example, the pool could be initialized with a 10$ fee for buys and 15$ for sells. A user decides to sell his tokens, he must now pay 15$ worth of ETH to the protocol and liquidity providers. The protocol-assigned smart contract could use this fee to add permanently locked-liquidity, establishing a rising floor price for this token and to provide sustainable revenue for creators. A liquidity provider can immediately claim his entitled share of reward the swap collected. In this section we describe how the liquidity provider fee is distributed and in the next section we elaborate over how the protocol fee is handled.

The Ethervista pair smart contract maintains a sequence of ascending numbers known as Euler amounts. These values are updated each time native ETH is transferred to the pair contract. Each Euler amount is determined by adding the previous Euler amount to the ratio of the fee to the current total supply of liquidity provider tokens (LP). The initial Euler amount is set to zero.

Mathematically, this update can be represented as:

$$\text{Euler}_n = \text{Euler}_{\text{n-1}} + \frac{\text{fee}}{\text{LP supply}} \tag{1}$$

With the corresponding sequence of rising numbers:

$$\{\ \text{Euler}_1,\ \text{Euler}_2,\ \text{Euler}_3,\ \text{Euler}_4 \ldots\ \text{Euler}_n\}$$

This sequence is particularly of interest to liquidity providers. Each provider is represented by a struct which stores the LP holdings of each user and a variable called euler0 which is suggestively named after the Euler amounts in our sequence.

```
struct Provider {
    uint256 lp;
    uint256 euler0;
};
```

This uint256 number represents the latest Euler amount in our sequence at the time the user adds liquidity, in which case we would have $\text{euler0} = \text{Euler}_n$. Suppose the user decides to claim rewards a thousand swaps later. In that moment, the latest Euler amount is $\text{Euler}_{\text{n+1000}}$. The exact amount of rewards that this provider accumulated during those thousand swaps is

$$\text{Reward} = \text{lp} * (\text{Euler}_{\text{n+1000}} - \text{euler0}) \tag{2}$$

This approach operates under the assumption that the LP balance remains constant throughout the period. Therefore, whenever a provider takes any action, such as adding/removing liquidity, the variable euler0 will be refreshed to reflect the latest Euler amount in our sequence. This measure prevents a liquidity provider from manipulating their own share of rewards. As such, it is advisable for a liquidity provider to always claim rewards before adjusting their LP balance. LP tokens are non-transferable except when being burned or added/removed from the liquidity.

The essence of this mathematical approach lies in its ability to accurately determine the share each user receives per individual swap, regardless of the continuous changes in the total supply of LP tokens due to liquidity providers adding or removing liquidity.

3 Pool Configuration and Protocol Fee

The individual who initiates liquidity provision becomes the Creator, endowed with write access to configure pool settings. This includes determining the pool fees, the protocol address and the metadata. The pivotal parameter is the protocol-assigned smart contract address. While this parameter is optional, it defaults to the Creator's address. Subsequently, this address receives ETH from the protocol fee, managed through the smart contract's custom logic, enabling an endless range of DeFi applications that were not previously possible on current AMM standards. This new form of revenue generation shifts the focus from *largely* prioritizing short-term gains and price action to *largely* prioritizing activity, longevity and utility.

Creators can define on-chain **metadata** for their token, including details such as website URL, logo, project description, social media handles, and chat URLs. This information will be accessible to users via the Explorer window of the Ethervista DEX, along with other relevant details. This allows creators to present their projects effectively while ensuring that users have access to verified, secure information, reducing the risk of phishing attacks. Developers can seamlessly launch their projects on Ethervista using the integrated Launcher window.

Ethervista also features **SuperChat**, a global live chat integrated directly into the DEX platform, enabling users to exchange information quickly. Access to SuperChat is tier-based, depending on the number of VISTA tokens a user holds.

Creators have the option to renounce their write access rights, effectively locking all settings permanently. Creators wishing to limit their token trading exclusively to Ethervista can restrict the ERC20 transferFrom function to the Ethervista router address, a variable stored within the factory contract address.

In addition to the pool and protocol fees, a fixed $1 fee is allocated to the ongoing development of the Ethervista DEX and $VISTA. This fee will fund the implementation of feeless flash loans, futures, and lending features, as well as support potential CEX listings and marketing initiatives.

4 VISTA

$VISTA is the native currency of the DEX, with a capped supply of 1 million tokens. Ethervista is a **value-compounding deflationary token**. The Ethervista protocol's smart contract implements an on-chain process whereby each burn event not only decreases the circulating supply but also incrementally raises the token's price floor. This effect is sustained through the continuous acquisition and destruction of tokens, financed by fees generated within the protocol every transaction. Hence, VISTA's mechanics act as a hedge against inflation by tying activity to supply reduction and price floor growth, strengthening VISTA's value with every transaction, driving sustained growth and scarcity.

5 Conclusion

It's moontime baby!