ETHEREALUSD V1 LITEPAPER

ZYGOMEB
zygomeb@gmail.com
ethereal.systems

April 19, 2023

1 ABSTRACT

With precursors of MakerDAO, experience has found the system unable to defend against upwards depeg, amended by adding harder stablecoins as collateral, effectively voiding the premise of decentralization to a large extent.

Liquity, a novel approach to fully decentralized stablecoins suffers from capital inefficiency with a steep requirement of supply to be placed within its stability module, thinning the market and going against the core value proposition. It is also hard limited on depegging upwards by the lowest possible collateralization ratio for instant arbitrage. Under normal operation this, on efficient ledgers, could reach very tight margins, but has not been proven to work even under reasonably small volatility.

We present the EtherealUSD model, that adds a capacity for instant redemption that takes inspiration from both aforementioned systems, with focus on capital efficiency and ability to ensure a tight peg. While the document focuses on a USD-pegged synthetic asset, the necessary prerequisite to mint any asset requires only a reliable exchange rate feed between synthetic and the collateral asset.

2 HIGH LEVEL

It's crucial to preface this paper by saying that anything presented herein is still subject to change, and is only meant to sketch the broad vision of the protocol.

Here the description focuses on the most important functionality, omitting anything regarding fees, treasuries, and technical considerations. It is merely a high-level design draft, outlining the specific design choices and meant to be understood by most of users. The parameters mentioned in this document are variables that can be changed via governance and are expected to be depending on ecosystem developments.

The system outlined in this paper is the base layer of the EtherealDAO ecosystem, and omits mention of the governance and oracle systems, which are to be explored in further papers. It can be thought of, accurately, as a mix of Liquity and Maker with new and innovative features added in.

Before diving into the system modules, a high level description of the system's core innovation is allowing users to tap into available liquidity within the system by allowing any user to mint the stablecoin at a rate of 1:1+fee. This allows a reasonably tight peg on the open market and high capital efficiency. The rest of the system is built around this feature to make it as safe and secure under stress as possible.

The main concern of the protocol is its Total Collateral Ratio (TCR), based on which certain features and variables adjust in order to ensure a good balance of efficiency and stability.

2.1 ECDP

Within the system exists ECDPs, short for **Ethereal CDP**, a user-owned position and a balance sheet of assets to liabilities. We use the term interchangably with CDP in this document but for user-facing interfaces the ECDP term should be used as they are enhanced with market making capabilities via **Mandatory Pegging**. Focus is placed on a single-collateral, single-liability system for V1, appropriately dubbed EtherealUSD.

From the CDP we derive its collateral ratio **CR**, tracking the value of assets over liabilities expressed as a %. This variable determines whether or not a CDP can be **Liquidated**, and, whether it is chosen for **Mandatory Pegging**.

The system tracks a system-wide **TCR**, total collateral ratio, total assets in all of CDPs over total liabilities, that allows it to change between **System Modes**.

2.2 LIQUIDATION

The system has a dynamic variable, **MCR**, minimal collateralization ratio, that is a function of TCR. When the CDP CR goes under MCR, the position is liquidated.

To liquidate a CDP, its assets are sold using a Dutch auction, starting low and going up higher to total assets, aiming to buy all of the debt back. This may be done in multiple transactions, partially filling the order. When the auction is finished, the difference between sold assets and total assets is transferred to **Treasury**, external to the system.

Should the auction end without a sale, or by at any point during it becoming bad debt, as a fallback, the system distributes the position's assets and liabilities over available CDPs.

2.3 MANDATORY PEGGING

This module consists of Burning and Minting components. They allow anyone to either mint or burn the stablecoin directly against the system itself without creating a CDP.

This functionality, for a small fee, allows for a well maintained peg on the open market. When the system is disabled, it is expected for the quality of the peg to be impacted. For a user that wants to burn the stablecoin, the liquidiy is taken first from the *least* collateralized CDP. For a user that wants to mint the stablecoin, the liquidity is exchanged for liability with the *most* collateralized CDP.

While burning of the stablecoin will always raise a CDPs's CR, the action of minting will always lower it. As a liquidation preven-

tion measure, using the Minting component will not force a CDP to go under the **BackstopPoint**, which, at the point of the action's execution, will prevent liquidation and may only be liquidated by downward price action, with a margin of safety depending on **TCR** and **BackstopDelta**.

In order to prevent a price manipulation attack on the system the amount of stablecoins minted via the Minting module per day/hour is limited and governed by a system parameters **MintingLimitDay** and **MintingLimitHour**. The normal minting via CDP creation is not subject to those limits.

Both actions, the moment when it is taken by a user, are profitable to the CDP owner.

2.4 SYSTEM MODES

To best understand the EtherealUSD modes of opeation we shall discuss points at which the system changes its behavior, listed in order of descending (mostly) % values. TCR crossing those points being the trigger for change in system behavior.

Above **RecoveryPoint**, a %-value parameter, every system is enabled. The MCR equals InitMCR and the Mandatory Pegging module is fully functional.

TCR going under the point, however, makes the MCR grow linearly from InitMCR at RecoveryPoint up to RecoveryPoint at InitMCR. Further, the stablecoin redemption/burn fee begins to slowly decay and go negative, becoming a rebate for burning stables back after sufficient time passes.

The **BackstopPoint** is the point at which the Mandatory Pegging's Minting feature is disabled, and the minimum CR a CDP may be taken to via Mandatory Pegging. It's a dynamic point, derived by adding BackstopDelta to the maximum of MCR and Convexity Point.

Aforementioned **ConvexityPoint** is the fixed point of the MCR function (for a linear one, also the midpoint between the InitMCR and RecoveryPoint). In and of itself nothing changes within the system crossing this point, but it is worth naming. At this point the MCR will only go up higher in situations of instant downward

price movement.

Lastly, for situations of great peril, **CriticalPoint**, that for all intents and purposes serves only as a bank run prevention measure. It disables Mandatory Pegging's Burning feature. Meant to be set fairly low, close to 100%.

3 MOTIVATION

A series of brief motivations for the design choices in the previous section. For the more defi-versed.

3.1 LIQUIDATION

The concept of a stability pool, while allowing for a quick, reliable, and gas-efficient way to liquidate positions, is not exactly one that should be called capital efficient. A symptom most often being a substantial portion of the market cap sitting idly in it instead of contributing to the core value proposition and being used for creation of a liquid market, and facilitating an ecosystem upon this stablecoin.

Importantly when bad debt is created during a moment of downward volatility the CDP owners will be made to cover it via the backstop liquidation method.

3.2 MANDATORY PEGGING

The system is an extension to Liquity's Redemptions, which are just the Burning part of this module. Adding Minting allows for a peg to be maintained from the upside with a much tighter band than without it, addressing a large problem with it, as well as removing reliance on fiat backed stablecoins as collateral to maintain peg.

Importantly to the discussion of the system's safety, the Minting module has been fairly restricted in this version of the design. The BackstopPoint is assumed to be fairly conservative in order to prevent a stable minting attack vector that at scale attempts to

liquidate CDPs, as well as the direct limit on the daily amount of stables minted using the module.

At sufficient scale the system must be of sufficiently high capital efficiency in order to prevent upwards depeg. There are times in which the price may still go above the target rate if the MintingLimit is hit or not sufficient amount of capital is being allocated towards support of it. Time should naturally restore the peg while maintaining a reasonable degree of security. Lower Minting fees will allow for a much tighter spread on the open market.

3.3 SYSTEM MODES

To those familiar with Liquity, their Recovery Mode instantly pushes MCR to 150% (from 110%, and the highest it can get). The mode presented here is smoother, and as such can last a longer in a down market.

Under normal operation, the system is a highly efficient money market for a decentralized stablecoin, wherein every ECDP is responsible for maintaining a healthy market.

When the Recovery Mode is engaged the MCR begins to smoothly increase, which is important to reducing volatility and allowing higher capital efficiency even under moments of system stress.

The key realization, and incentive for pushing TCR higher is the burn fee decay that becomes a factor at under Recovery Point. Without it, the system may get stuck in low TCR under Backstop-Point losing efficiency and interest.

Most important to this discussion is the Backstop Point, itself being a core efficiency throttle as it sets a hard limit on how many stablecoins can be minted at once. This is a necessary limitation to prevent a liquidation attack and give a concrete and quantifiable backstop to how much margin for error one has when maintaining a CDP.

4 FUTURE TRACKS

The version presented herein is not final, and subject to change before launch. Comments and contributions are welcome.

[REDACTED]