EASH stable coin white paper

1. Back Ground

As digital money continues to soar in popularity indices and investor interest, it has become increasingly widely used. But the fluctuation of the price of digital money hinders their use as an exchange medium and accounting unit. Popular digital assets such as bitcoin and Ethernet are too unstable to be used for everyday money. The value of bitcoin often fluctuates dramatically, rising or falling by up to 25% in one day.

One proposed solution is the creation of a stable value coin (often called a stablecoin), whereby an issuer distributes a cryptographic token to customers in exchange for a specified fiat currency, like the US dollar, at a fixed 1:1 exchange rate. Because the U.S. dollar is a highly desirable medium of exchange, as well as a globally accepted unit of account, it is an ideal anchor for a stablecoin.

Several stable coin solutions, such as USDT, GUSD, Base Coin, Maker Dao, have been proposed, centralized and secured guarantee solutions included.

Most of these stable coins are based on the ERC20 token issued on the Ethernet platform. To expand the application of the EtherZero platform, we have implemented the EASH stablecoin, which is a collateral-supported encrypted currency. Anchoring the price of the US dollar to achieve 1:1 convertibility, its value is stable relative to the US dollar.

2. Create, Redeem, Trade

EASH is an ERC20 token issued on the EtherZero platform, generated by ETZ mortgage and can be transferred to the EtherZero system. The functions of creation, redemption and transaction are realized through intelligent contracts.

Anyone can use their EtherZero assets to generate EASH on the platform. Once generated, EASH can be used in the same way as any other encrypted coin, and it can be freely sent to others for payment of goods and services, or held as long-term savings. The emergence of EASH has created a powerful platform for distributed margin trading, providing a platform for the application of stablecoin.

2.1. Mortgage ETZ Get EASH

Users first send transactions to smart contracts to create mortgage positions, where the package is used to generate the collateral ETZ for the EASH and obtain the corresponding amount of the EASH. At the same time, the collateral contract accumulates an equivalent amount of debt, locking the ETZ until they pay the outstanding debt.

The mortgage ratio is set to 1:1, and the mortgage price refers to the real-time price of ETZ, the price refers to the weighted average prices of the mainstream trading platform.

Examples:

Allice has 100000 ETZ in hand, the current price of ETZ is 0.3USD/ETZ. Allice can then mortgage the 100000 ETZ in the intelligent contract to get 100000*0.3=30000 EASH.

2.2. Redeem ETZ

Users can retrieve collateral into their wallets by sending transactions to the contract, unaffected by the rise or fall of ETZ prices. Addresses that have been mortgaged ETZ are recorded by smart contracts, and the same address can be unloaded at the mortgage price initially agreed upon in the contract to obtain the original number of mortgages ETZ. Mortgage receipts are held for three months before they can be redeemed.

Users retrieve their own mortgage positions on the redemption page, and can either choose to redeem all of the mortgage or choose to redeem a part.

Examples:

Alice has 100000 ETZs in her hand and gets 30000 EASHs at 0.3 USD / ETZ.

Then She can redeem the ETZs at any time by the price 0.3 USD / ETZ.

Assuming Alice redeems all 30000 EASHs, she will get 30000 /0.3 = 100000ETZ.

Assuming Alice redeems 15000EASHs, she will get15000/0.3= 50000 ETZs.

The remaining part of the mortgage continues to exist, and Alice has the right to continue to redeem the remaining part of the EASH.

2.3. market maker

Only market makers can mortgage and redeem EASH. The threshold for market makers is to mortgage 100,000 ETZ. When mortgage exceeds 100,000 ETZ, they automatically upgrade to market makers. Market makers mortgage ETZ to get EASH, every 10 days unlock the total amount of 1/10, to the end of 100 days, the mortgaged EASH all unlock can be traded.

2.4. EASH Exchange

When the user holds the EASH but does not have the corresponding mortgage contract, they can trade with the trader to get back ETZ and maintain the liquidity of the EASH transaction. The transaction price is calculated according to the real time price of ETZ.

Examples:

Alice has 10 EASH in her hand, the current ETZ price is 0.3USD/ETZ, and Alice does not have any mortgage. Then she can convert EASH into ETZ through transaction.

Alice will get 10/0.3=33.33 ETZ if she choose to exchange.

3. Contract Design

Contract Architecture

We implement the EASH mortgage system by building a Cooperative Intelligent contract system, which is described in more details below. The core component of the system is the EASH contract, which is responsible for achieving stablecoin issuance in accordance with ERC20 standards on the EtherZero platform. Mortgage and Price Feeder contracts are responsible for mortgage and price feeding, respectively. The contract provides an interface with which token holders can interact and perform operations such as transferring tokens and viewing token balances.
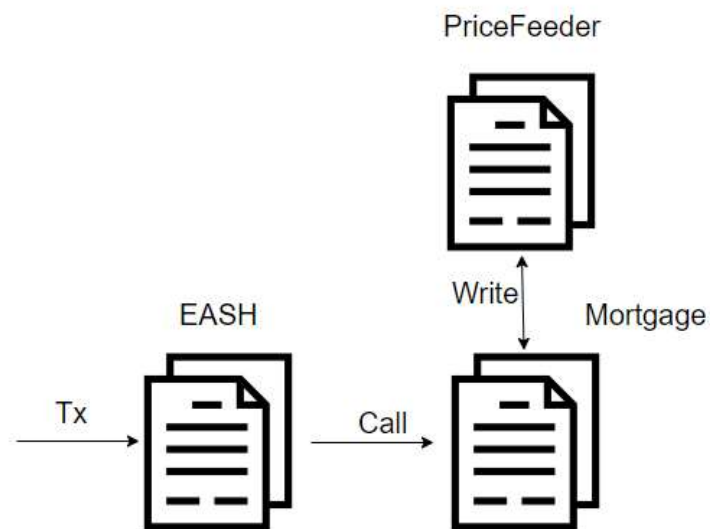
Price determination

The price determination scheme is determined from multiple price sources and then voted to ensure the stability and security of the price on the chain.

As follows, there are five price sources offering prices. These five price providers are distributed on different servers. The contract uses the average of the closest three prices as the measured price. In this way, even if two of the price source problems occur, the system can still function normally.

Price Feeder

Mortgage and redeem

The process of user mortgage generating EASH is to invoke Mortgage contract and send ETZ to the contract. Mortgage, as a trusted contract for authorization, calls the EASH token contract, issues the corresponding number of EASHs, and then sends the created EASH to the user's account address. The quantity of the issuance is calculated according to the price provided by the PriceFeeder contract, which is based on the following formula:

$$\text{EASH}_{amount} = \text{ETZ}_{amount} * Price$$

Redemption is the reverse process, destroying the EASH in the user account address and returning the locked ETZ to the user account.
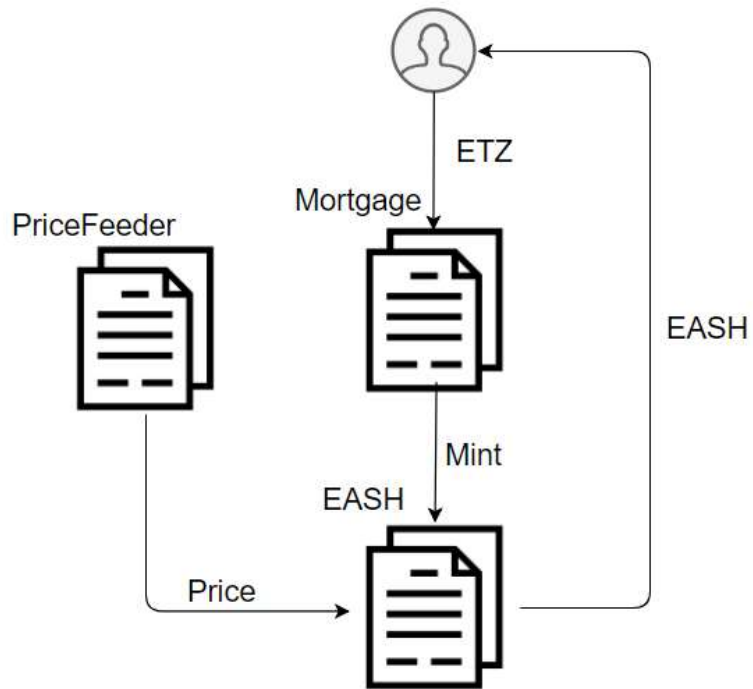
Exchange

The EASH obtained by the user through other channels can be exchanged. The transaction is completed by a legal dealer. The user sends the EASH to the dealer and gets the ETZ of the corresponding value. The number of ETZ obtained is calculated by the following formula:

$$\text{ETZ}_{amount} = \text{EASH}_{amount}/Price$$
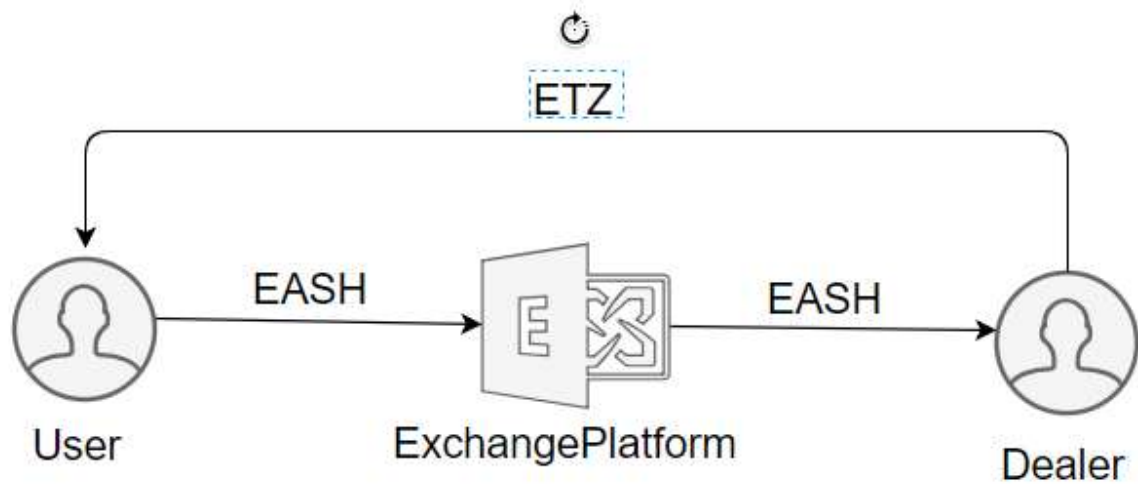
4. Risk Control

Customer mortgage obtains EASH, when the ETZ appreciation, the user can redeem the same amount of ETZ and enjoy the dividend brought by the ETZ appreciation. When the price of ETZ falls, it also needs to bear some of the losses brought down by price. Assuming that in extreme cases the ETZ suddenly collapses and the ultimate debt in the contract exceeds the value of its collateral, more ETZ will be injected into the platform to ensure the stability of the EASH.

This makes the stable currency on the chain more transparent and secure, and does not have to worry about having no equivalent amount of assets behind it to support the price of EASH.

5. Contract Guarantee

A hierarchical approval mechanism is designed for some high-risk behaviors in the contract system. Each intelligent contract in the system requires the approval of the trustee. The trustee can be another intelligent contract. A trustee can view another trustee B, which may continue to view the next trustee C, and so on, forming a chain of supervision or "trusteeship."

The system implements the following security functions:

1) offline key: keys that approve high-risk actions are stored offline in Cold Storage System.

2) key generation: Keys are generated, stored and managed on the hardware security module (HSM). We only use HSM, and the technical specifications for each "signer" have reached FIPS PUB 140-2 Level 3 or higher.

3) Double control (multiple signatures): High-risk behavior requires the approval of at least two signers (i.e. multiple signatures). We chose M = 2 using the design scheme to be signed by M members in N, which provides security and fault tolerance.

6. Application Examples

Anyone can use the EASH system without any restrictions of registration. Here are examples of EASH application scenarios.

Example : Mortgage Consumption

Tinna needs a loan, so she decides to generate 100000EASH. She locks the ETZ worth 30000 EASH to a mortgage contract and uses it to generate 100000EASH. 30000EASH is sent directly to her account. Tinna can consume part of EASH to buy goods on e-commerce platform. If he decides to take back his mortgaged ETZ in one year, contract will send her the ETZ.