FORT

Protocol Whitepaper

A DeFi Development and Application System with infinite Liquidity

V 2.0

fortdao@fortprotocol.com

March 12, 2022

1 Abstract

Following the trend of the development of on-chain applications, lack of liquidity has became the most vital problem for DeFi applications. The previous DeFi projects have tried to apply the traditional order book method and Automated Market Maker (AMM) models to solve this problem. However, these methods still are not ideal solutions since they cannot integrate all financial services into a single protocol and share the same liquidity, which yields to resource wasting and low performance. This paper proposes a new protocol: FORT, which introduces the concept of discount computers and on-chain decentralized currency units, can systematically solve the liquidity problem and integrate all financial contracts and derivatives into one protocol. It can be used for transferring all financial instruments, including tools for hedging and other economic relationship lock-in, to on-chain applications.

Contents

2 Introduction

In the traditional financial market (Centralized Finance), financial intermediaries are responsible for maintaining the fairness of the trading; they are the deal makers who help the traders to find the matched counterparties. In other situations, they are also responsible for pricing the financial products. The financial intermediaries include diverse exchanges and dealmakers of the loan and asset auction market. Together, they boost the efficiency of financial trading to the extreme. For instance, driven by high-frequency trading, the exchanges allow the frequencies of their order flow to arrive in nanoseconds. Although the traditional financial market has improved its efficiency, the credit risk within also increases. The inevitable hosting problem, as well as the power intervention of the financial intermediary, will raise the implicit costs of financial transactions. These problems account for the main reason for developing DeFi. If Alice, a trader, wants to buy/sell some asset, or open

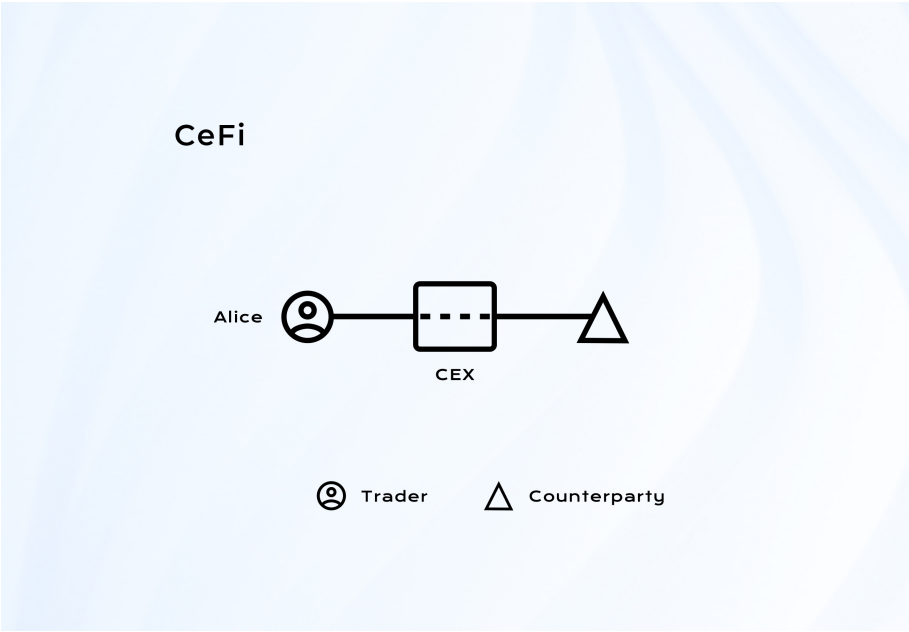

Figure 1: Diagram of CeFi (Centralized Finance)

a position in centralized financial market, she needs to go to a financial intermediary, centralized exchange in this case. After Alice filling her required trading in an

order table, the exchange will publish her order and find counterparties for her. The diagram of this process is exhibited in Figure 1.

DeFi (or "decentralized finance") is an umbrella term for financial services on public blockchains. With DeFi, participants can do most of the things that banks support, earn interest, borrow, lend, buy insurance, trade derivatives, trade assets, and more, but it's faster and doesn't require paperwork or a third party. As with crypto generally, DeFi is global, peer-to-peer (meaning directly between two people, not routed through a centralized system), pseudonymous, and open to all.

The first generation of DeFi (DeFi 1.0) projects are based on the traditional approach of order book. For example EtherDelta, the EtherDelta exchange is a cryptocurrency trading platform based on the Ethereum blockchain, as well as ERC20 standard tokens. The exchange was established in Chicago in 2017. The platform is based on smart contracts, with the help of which all contracts are carried out inside the website (purchase, sale, withdrawal of funds from the wallet, etc.). Since every order has been written into a contract, waiting for counterparties, this approach is very inefficient and naturally creates a barrier for improving the liquidity of the market. The problem of DeFi 1.0 is: 1) the liquidity on the chain cannot be effectively ensured; 2) finding the final matched counterparties still needs for centralized institutions; 3) once a price deviates from the real value, the gas fee cost of order adjustment is high. This traditional financial approach is not appropriate for on-chain applications. We extend our story of Alice: now she tries to trade on a DeFi 1.0 platform. She can participant in a smart contract which specifies her request and wait for counterparties to notice her order contract. The diagram of this process is exhibited in Figure 2.

The second generation of DeFi (DeFi 2.0) introduces Automated Market Maker (AMM) model, which makes decentralized exchange (DEX, e.g., Uniswap, Sushiswap, etc.) and lending projects possible. DEXs have improved the liquidity of the DeFi protocols and allow users to deposit tokens to earn trading fees. Lending projects (e.g., Aave, Compound, etc.) provide the users a platform to earn from providing

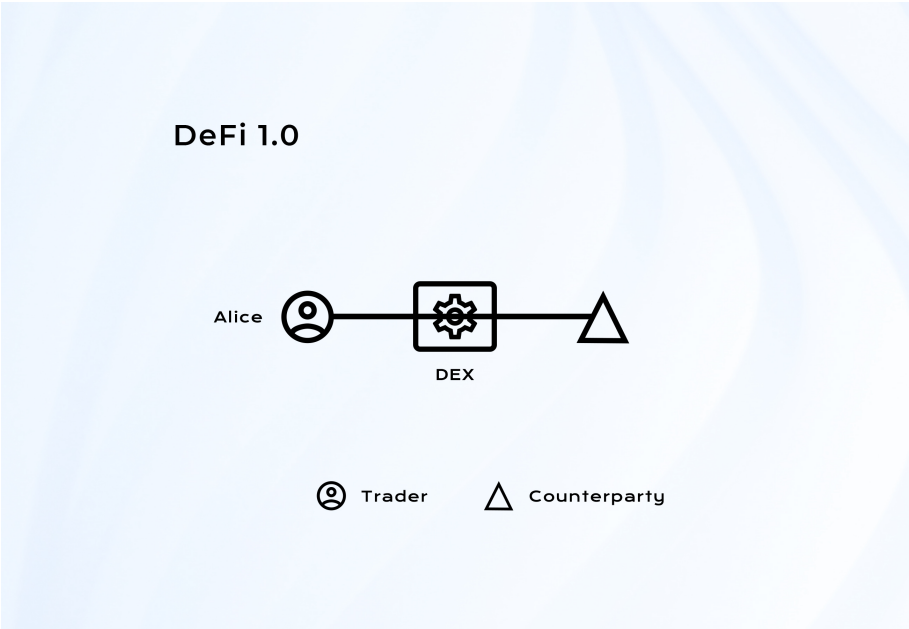

Figure 2: Diagram of DeFi 1.0

liquidity and to borrow. Projects like MakerDao vitalize DeFi market by allowing users to generate a kind of decentralized stablecoin, this kind of stablecoin is available to anyone, anywhere, like Dai (generated by MakerDAO) is backed by a surplus of collateral that has been individually escrowed into audited and publicly viewable Ethereum smart contracts. DeFi 2.0 led to a raising wave in crypto market in 2020 since DeFi 2.0 projects provided a better solution to the fundamental problem of DeFi industry: lack of liquidity. With a DeFi 2.0 platform. Alice can directly trade assets with the asset pools deposited by liquidity providers through participating in a smart contract. The diagram of this process is exhibited in Figure 3.

However, whether it is AMM or liquidity pools, the solution to the liquidity problems is at the expense of the liquidity provider's flexibility: a liquidity provider needs to fix his own trading strategy and bear the fluctuations of the external market. Once the price is favorable to the liquidity provider, the counter party trader can opts to back out of the deal. Once an arbitrage opportunity appears, the arbitragers flock in and cause impermanent losses to the liquidity providers. The liquidity providers

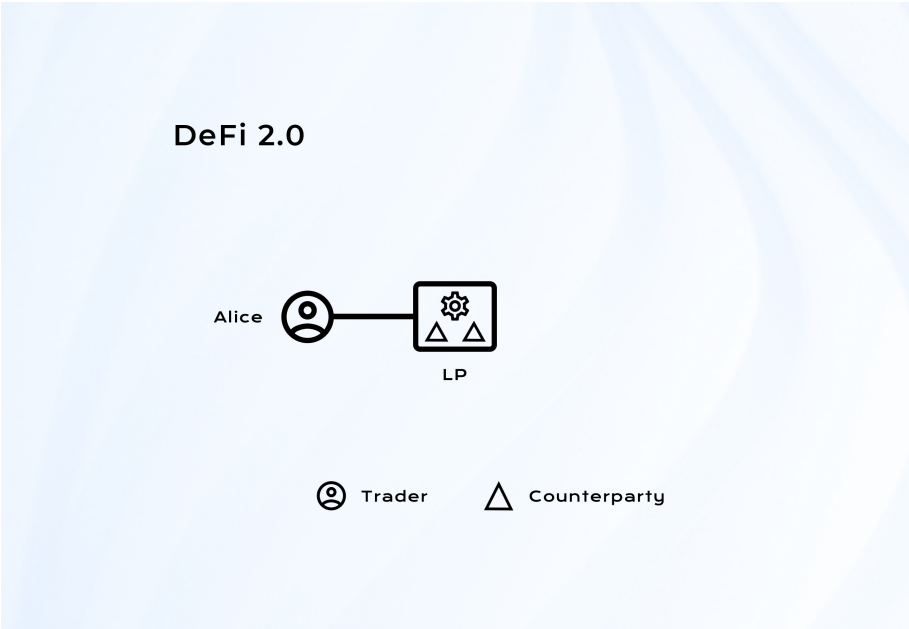

Figure 3: Diagram of DeFi 2.0

have no choice but hopping that for subsidies from mining and commission or interest earned can compensate their losses. Although this asymmetric design temporarily alleviates the liquidity shortage, but it generates the following consequent problems in the long run: firstly, a large amount of assets are occupied and deposited in DeFi protocols while only a small volume of transactions are supported, which results in resources wasting. Most of the total value locked (TVL) is still rushing to liquidity mining for profits. In the long term, the lower the capital utilization rate, the greater the loss of intrinsic value of the DeFi products. Secondly, to the AMM, the core variables, such as asset price and interest rate, are related to the size of the pool. When the size is small, it is easy to create arbitrage opportunities by high volume trading on one hand, and it is difficult to attract many traders and lenders when the pool is not large enough on the other hand. Moreover, the composability of DeFi products and liquidity sharing are only at the technical level since the TVL of various DeFi platforms cannot be shared. This kind of liquidity created at the expense of sacrificing the liquidity providers' flexibility is not a perfect idea under

the decentralized architecture.

It is better to completely erase this asymmetry between liquidity providers and other traders, creating a brand new form of Defi 3.0: the system plays the role of the counterparty to all traders. This idea complies the decentralized spirit. Everyone is in the same position to "gamble" with the smart contract, whether the underling asset is Bitcoin, Ethereum or other crypto-assets. The system burns the tokens immediately after any participant pays the system tokens for some financial products, and mints new tokens to settle the financial products if the participant deserves. We call this model ILM (Infinite Liquidity Model), and comparing with AMM, we call the system OMM (Omnipotent Market Maker). It frees us from the traditional financial trading models and forms a new financial paradigm which not only ensures composability but also uniforms programmability for all DeFi products within the same framework.

Figure 4: Diagram of DeFi 3.0

With Defi 3.0, Alice now can directly trade with the system by participating in a smart contract. The system will be her counterparty in this situation. The diagram

of this process is exhibited in Figure 4.

3 Principles of FORT

The essence of any financial trading or financial product is a swap of different returns with different risks, no matter the financial activity is an instant transfer, lending, an option, or any complicated derivatives.

Consider stochastic processes $\{S_t\}_{t\geq 0}$ and $\{r_t\}_{t\geq 0}$ defined on $(\Omega, \mathbb{P}, \mathcal{F})$. We use S_t to describe the asset price and r_t to describe the interest rate at time t. Let $\{\mathcal{F}_t\}_{t\geq 0}$ be a filtration generated by S_t and r_t. It is the information stream of the market at time t. Consider a financial product which needs to be evaluated at time $t = 0$. Denote R_t the cash flow in at time t, and C_t the cash flow out at time t. Let τ_0 and τ_1 be the time sets of cash flow out and in respectively, then the diagram of financial products or derivatives can be illustrated as:

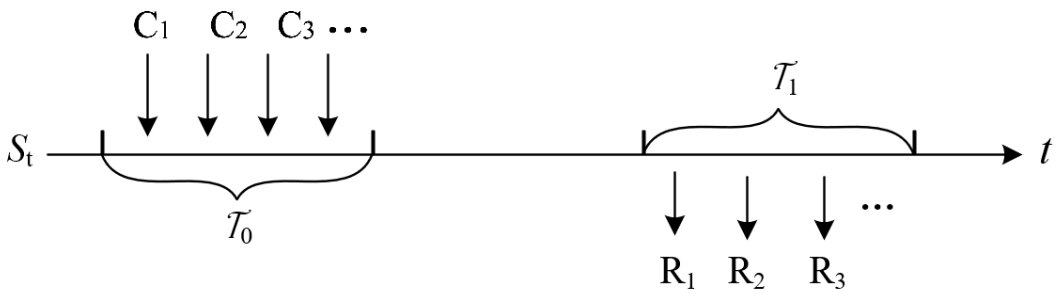

Figure 5: Diagram of financial products or derivatives

The principle of FORT is that, according to a given discounting algorithm, any financial products or derivatives can be decomposed into a stream of cash flow in and a stream of cash flow out. The cash flow in is the return of the financial product, and the cash flow out is the cost of the product. Moreover, the sum of the expectation of discounted future cash flow in streams is equal to or slightly smaller (in the case for deflation) than the sum of the expectation of discounted current cash flow out streams are equivalent. Both cash flow in and out are uniformly paid

by the decentralized currency unit (DCU). With respect to the sigma algebra $\mathcal{F}_0$ at time $t = 0$, the time of the evaluation, this process can be described as following:

$$\sum_{t_i \in \tau_1} \mathbb{E}(e^{-\int_0^{t_i} r_s ds} R_{t_i} | \mathcal{F}_0) \leq \sum_{t_j \in \tau_0} \mathbb{E}(e^{-\int_0^{t_j} r_s ds} C_{t_j} | \mathcal{F}_0) \tag{1}$$

Since a financial product can consist of a linear combination of basic revenue functions (BRF), each of which corresponds to a basic discount function (BDF), the cost of the product is a linear combination of these basic discount functions. Thus, we can design the basic revenue function and its discount function as a developable model: a discounted computer - any financial product can be developed by this computer, and the basic revenue function is like the instructions of the computer. The basic discount function is like the cost of these instructions or EVM-like gas, except that the gas is paid in DCUs, and the instructions produce DCUs. The calculation of the financial product P and its cost C(P) can be described as:

$$P = x_1 \cdot BRF_1 + x_2 \cdot BRF_2 + ... = \mathbf{BRF} \cdot \mathbf{X}^T \tag{2}$$

$$C(P) = x_1 \cdot BDF_1 + x_2 \cdot BDF_2 + ... = \mathbf{BDF} \cdot \mathbf{X}^T \tag{3}$$

$\mathbf{X}$ is the linear combination coefficient of the **BRF** representation and **BDF** is the discount function of the **BRF**, where

$$\mathbb{E}(BRF_i | \mathcal{F}_0) \leq \mathbb{E}(BDF_i | \mathcal{F}_0) \tag{4}$$

The discounted computer can develop a variety of financial products, including options, perpetuals, leveraged trading, swaps, and lending products, to name just a few, and almost any financial product can be produced.

3.1 DCU Issuance, Settlement and Pricing

Decentralized Currency Unit (DCU) is issued by FORT protocol and has an unlimited supply, with the initial DCU offering of no more than 100 million. In FORT protocol, DCU is the only currency for cash settlement. For example, when a participant into a contract, which assures the participant to receive a certain amount of

DCU after a certain number of blocks if the underlining asset price meet some certain conditions within the blocks. In order to get into this contract, the participant needs to pay DCUs to the system, and the system will burn the tokens immediately after receiving them. If the condition have been met, the system will mint new DCUs and pay the participant with them.

As readers can see, all DCU holders share the risks and benefits of DCU issuance or destruction and participate in the equilibrium of supply and demand in the secondary market for DCUs: the demand for DCUs is from those who buy financial products on the chain and invest in DCUs, while the supply is determined by the initial offering and the DCUs settled by the FORT protocol. The advantage of sharing the same settlement unit is that we can build all financial services such as trading, lending, derivatives without issuing too many tokens through continuously improving the liquidity of DCUs.

According to

$$\mathbb{E}(BRF_i|\mathcal{F}_0) \leq \mathbb{E}(BDF_i|\mathcal{F}_0)$$

Total discounted supply G_y satisfies:

$$\mathbb{E}(G_{t_2}) \leq \mathbb{E}(G_{t_1}),\ t_2 \geq t_1 \tag{5}$$

Total demand (D_t) is determined by trading demand, and the price of DCUs (P_t) is determined by the equilibrium of (D_t, G_t). Considering the growth of demand and the long-term deflationary nature of supply, there is a logic for a sustained rise in P_t.

As the DCU is named, combined with FORT contracts, it is an on-chain universal currency with scenarios, which BTC and ETH cannot achieve: BTC has no on-chain scenarios with fixed issuance, ETH follows all scenarios as gas, but its issuance is according to a fixed algorithm, not incremental according to scenarios. DCU is guaranteed to clear in every scenario, which matches the completely decentralized currency envisioned by many economists, and is a further development after BTC-ETH.

3.2 The Oracle

The decentralized oracle adopted follows this idea: given an on-chain price stream, how to design a decentralized game such that the game equilibrium can output a price stream with the smallest possible deviation from the on-chain price stream. The oracle solves this problem with quotation mining, two-way options, validation cycles, price chains and β factors. The oracle provides a price sequence that does not change the distribution of asset prices, but approaches a discrete sampling model, which is determined by the structure of the decentralized game, where the quote deviation and quotation density depend on the depth of the arbitrage market and the price of the oracle token. Overall, the oracle provides an efficient decentralized way that maintains the fundamental traits of asset prices. In FORT, we tend to use highly efficient market prices, and hence choose the most liquid underlying assets such as BTC and ETH, etc. The basic price model follows the Geometric Brownian Motion (GBM) model. Considering the characteristics of prices deviation and discrete time, we correct the prices using the k-factor as follows,

$$k = \max(\frac{|p_2 - p_1|}{p_1}, 0.002) + \sqrt{t} \cdot \max(\sigma, \sigma_0) \tag{6}$$

where p_2 and p_1 represent the current and previous prices respectively, t, measured by second, represents the difference between the time transaction happens and the time p_2 becomes effective. Furthermore, σ the instantaneous volatility follows

$$\sigma = \frac{|p_2 - p_1|}{p_1\sqrt{T}}$$

where T represents the time lapse between p_1 and p_2 becoming effective. σ_0 denotes the regular volatility, set by the protocol (generally different value for different financial product).

The correct procedure follows

- when it comes to a call option, the long price is $(1 + k)p$ while the short price is $\frac{p}{1+k}$

- when it comes to a put option, the long price is $\frac{p}{1+k}$ while the short price is $(1+k)p$

where p represents the base price.

3.3 Time Domain

The time domain denoted by τ_0 and τ_1 can be instants or intervals. A moment can be a definite moment or a random moment (e.g., a stoppage), and in finance, an interval is often used to determine some mean or stoppage. Although the time on the blockchain is discrete, these discrete differences can be ignored for a longer period and compensated for in a shorter period based on the k-factor, and thus can be interpreted approximately in terms of continuous time intervals.

3.4 Discounted Computers

We abstract all financial products (services) as an exchange of revenue streams and expenditure streams, and the revenue streams are represented by linear combinations of the basic revenue functions. Then any financial product development only needs to determine the linear combination of basic revenue functions to obtain its cost (present value) by the linear combination of basic discount functions. Such linear combinations are the same as we use computer programming, therefore we call this figuratively as the discounted computer. Any financial product corresponds to a piece of computer programming, so that the composable DeFi becomes program designing and calling in the same framework, reducing the difficulty of understanding and risk management.

3.5 Basic Revenue Function and Discount Function

The basic revenue function (BRF) can be a deterministic value (e.g., block 13678933 to get 1000 DCUs) or it can be a random variable after introducing the oracle price information. Here, we consider the basic types of deterministic values, random vari-

ables of pricing oracle, and probability random variables, each of which consists of polynomial functions, exponential functions, logarithmic functions, absolute value functions, maximum-minimum functions, and definite integral operators. The base discount function (BDF) contains normal distribution functions, polynomial functions, exponential functions, logarithmic functions, etc. Considering that the reality does not need so many revenue functions as well as the complexity of the calculation, we choose a relatively simple list of functions, which can be gradually improved later. As mentioned earlier, the basic revenue function is the basic instructions of the discounted computer while a financial product is a program which is a combination of these instructions.

3.6 Discount Rate and Interest Rate Oracle

In principle, the discount rate reflects the risk-free return of the on-chain world. We can choose a risk-free interest rate statistic on the chain such as the PoS yield of ETH or the decentralized interest rate oracle (a design as follows: given the number of DCU issues per year, anyone who locks DCUs can share these tokens proportionally) as the discount rate. However, this paradigm is more suitable for a traditional centralized world. In a decentralized world, we can take the discount rate to a relatively small value, even zero, in order to give the DCU deflationary properties and thus ensure a steady rise in the DCU price.

3.7 Pricing Unit Transformation

If a fiat or ETH is required as the numeraire in FORT, it is sufficient to introduce the DCU/USDT or DCU/ETH price, which can be obtained through the oracle. If the liquidity of DCU is large enough that the settlement of a single financial product has little impact on the price, the financial product with the introduced price is no different from the traditional financial product. The pricing based on the risk-neutral measure ($\mathbb{E}^Q$) can perfectly solve the calculation of the discount function,

which can be used for hedging or asset portfolio management.

$$\mathbb{E}^Q(BRF_i|\mathcal{F}_0) \leq \mathbb{E}^Q(BDF_i|\mathcal{F}_0) \tag{7}$$

3.8 Financial Product Development

The development of a financial product in FORT is the same as writing smart contracts, i.e., finding a vector with BRF as the base for the target return, and that vector represents this financial product. The product of that vector and the corresponding BDF base is the cost of the financial product, i.e., it is only necessary to pay this cost in the time domain τ_0 to obtain the financial product. That financial product will get the DCUs minted by the FORT contract in the time domain τ_1, and its quantity is the product of that vector and the BRF. This process is just like writing smart contracts by which all financial products can be implemented with the discounted computer programming of FORT. Developers do not have to operate the liquidity of tokens, and just need the DCUs to be liquid enough.

4 Applications

FORT has a wide range of applications, covering almost all financial services and different trading structures (including peer-to-peer, many-to-many, etc.). It is a revolutionary design in the history of blockchains, and having the ability of lock-in various on-chain economic relationships.

4.1 Options and Options Coins

Options products become pretty simple: giving an expiration date and an exercise price, users can obtain a call or put option whose cost is determined by discount functions. Although this formula is not a risk-neutral measure when DCU price is not quoted, so care needs to be taken to understand the implications. When the DCU price is quoted, it is the same as traditional options, except that the interaction becomes much simpler and there is not much brokering to consider.

A better model is to issue the options as a token, i.e., the same token for a given expiration date and strike price, regardless of when the issue starts. The advantage of this model is that it allows traditional derivatives exchanges to dispense with the issue and settlement. In order to meet the issue demand, they either need to do a lot of matchmaking or find a market maker, and in order to settle, they often need margin management. Market makers also need to develop a hedging strategy, which is a huge financial auxiliary system. Although traditional finance is happy to do so, but its cost is much higher than the FORT model. Because the issuance and settlement of the FORT model, exchanges only need to solve the problem of secondary market trading of derivatives.

4.2 Perpetual Contracts, Leveraged Trading and Leveraged Coins

Perpetual contracts or leveraged trading can also be very simple, which is a dynamic settlement model based on basic revenue functions. We can develop perpetual or leveraged transactions into a model called a leveraged coin: dynamically changing the balance of its tokens based on prices, which has been practiced in current algorithmic stablecoins.

4.3 Trading, Price Coins and Stablecoins

A native asset is actually equal to a price coin (DCU), which is equivalent to splitting an asset into a dynamic price and a fixed settlement unit. This model can only be effectively implemented in a fully decentralized world: the traditional centralized world has the credit risk of cashing out. Thus, trading is equivalent to exchanging DCUs for various price coins, or settling out of DCUs with various price coins. Or using the native asset to exchange the corresponding price coin at a ratio of 1 to 1 (which is actually slightly off, due to the price deviation of oracles). By analogy, a stablecoin pegged to a fiat currency such as the US dollar is a USDT price coin.

4.4 Exponential and Logarithmic Coins

A new paradigm is the exponential coin, where the percentage of price fluctuations feeds into the growth of returns in an exponential manner. Compared to leveraged coins, exponential coins have many advantages: no need to close positions, faster growth, interchangeable transfers, free stacking at the same address, etc. For example, when the price doubles, the exponential coin with the e base can grow 7.4 times, and when the price triples, the exponential coin grows 20 times.

4.5 Revenue Swaps

Revenue swaps are cost swaps, as they are all equivalent to discounted of revenue streams.

4.6 Lending

Lending becomes much easier by pledging asset accepted by FORT to receive the corresponding DCUs, repaying it to receive the collateralized asset, and then being liquidated when the liquidation ratio is exceeded, where the core parameters are the liquidation ratio, collateral rate, interest rate, etc.

4.7 Insurance

Based on the characteristics at the end of the event, a price insurance can be made to swap the loss with the premium.

4.8 Interest Rate Derivatives

It is possible to design various interest rate derivatives through price information of the base interest rate oracle.

4.9 Probability Coins

Design probability coins that get DCUs at a given moment with a predetermined probability. For example, coins with 1/10 probability, each probability coin have a 1/10 probability of getting 10 DCUs.

4.10 NFT Applications

It is possible to lock in all economic relationships in on-chain games or NFTs based on DCU, i.e. all game assets can be designed to some probability coins or derivatives above. In this way, its game assets corresponding to NFT can be cashed in FORT, regardless of the existence of that game, thus building a consistent variable in the game world.

4.11 Multi-Party Trading

To design a transaction between two or more participants: A and B can create a contract based on FORT, each paying a certain amount of DCUs at the current moment and receiving the corresponding DCUs in the future, so that FORT can participate in the allocation between them, which realizes a multi-player competition and game structure.

5 Summary

FORT offers a new paradigm: financial products are considered as programming over basic discount functions. The cost is the expense of calling those functions, just like EVM. The difference is that the economic relations of the discounted computer are inherent. The new paradigm can cover almost all financial products (services) which can be bought at any time and settled with unlimited liquidity, where market makers, margin (call), and fear of being unable to settle are not required. As long as the DCU liquidity is sufficient, it would be extremely easy to recreate the traditional financial market. Moreover, as difficulties of issuance and settlement are

solved, traditional derivatives exchanges can focus on the secondary market, thus significantly reducing their costs. In addition, FORT can bring the fundamental consistent variables for the metaverse, with the ability to traverse different games to lock in economic relationships.