BlueStone Protocol

Yiran Tao, Schowen Peng

A decentralized banking protocol that captures the time value of crypto assets.

Part I

Introduction

We start our journey to design the BlueStone Protocol with one future in mind: the demand for on-chain banking services will grow exponentially as the crypto economy continues to expand. The diversity and complexity of use cases will require more advanced and scalable financial infrastructure which can be on par with real world banking services.

There are three building blocks of the BlueStone protocol:

Fixed Borrowing Rate

At the bootstrap phase, the borrowing rate will be based on external market prices and set through oracles; an autonomous interest rate model will be developed in parallel as more deposit and loan data is collected. The lending rate will then be calculated on-chain based on the borrowing rate and the utilization of liquidity.

Customized Deposit and Loan Term

Unlike traditional loans which have due dates, most decentralized lending protocols allow their borrowers to repay at any time. The lack of commitment comes with two major downsides: 1) borrowers are not incentivized to borrow for exact terms they need, thus hurt the overall efficiency of liquidity matching; and 2) lenders can't withdraw their funds at will, as there is no guarantee that there will be enough liquidity in the pool at the time.

By Introducing terms, BlueStone is able to differentiate borrowing demand, offer favorable rates and guarantee liquidity at the same time.

Personalized Financial Records

The lack of identities and financial records has greatly impeded the potential of on-chain lending. By engaging with the BlueStone protocol, both lenders and borrowers will be producing meaningful data directly on

the blockchain. The protocol will provide an interface for third parties to leverage this data.

For example, Dapp developers will be able to query the loan history of an Ethereum address and decide whether this address is eligible to enjoy certain benefits. In this way, prices can be targeted for different borrowers and funds can be more efficiently allocated. In addition, other parties are incentivized to develop on top of the BlueStone protocol and lay the groundwork for an identity system.

Basics Features and Implementations

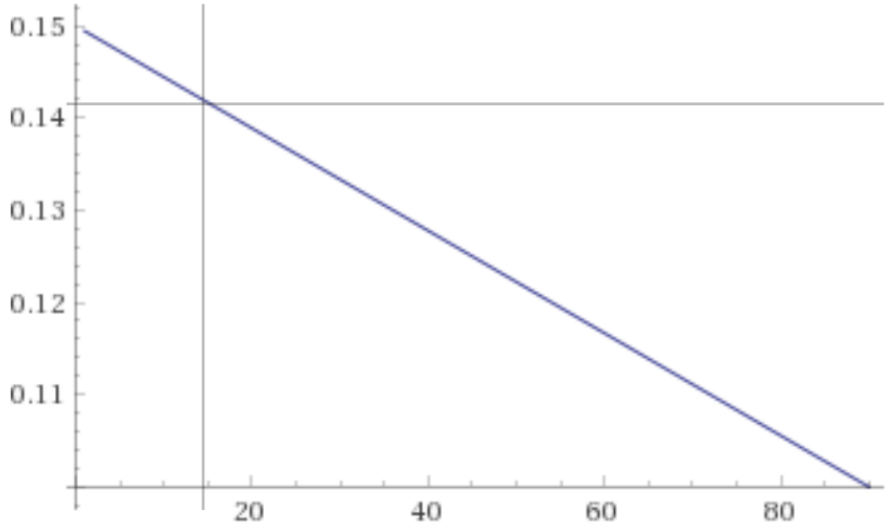
In Phase 0, users can interact with BlueStone portal, a Dapp that provides basic lending and borrowing services.

- After the lenders deposit assets into the protocol, the deposits will be stored inside a PoolGroup. The length (M) of a PoolGroup represents the maximum term available to the lenders and borrowers and consists of liquidity pools with available terms up to [T].

- Borrowers are allowed to borrow for any term (T) between range [1,M], by providing collateral accepted in the protocol; for each term he or she chooses, the rate is determined by the following on-chain function :

$$BorrowRate = H - \frac{H - L}{M} * T \qquad (1)$$

(L,H) is the borrowing rate range.



For example, the length of the liquidity pool(M) is set to 90, and (L, H) is set to (10%, 15%). Alice wants to borrow 2000 DAI for 15 days, the

calculated APR_15 = 15% - $\frac{15\% - 10\%}{90}$* 15 = 14.16% and the required collateralization ratio is 150%, she needs to collateralize at least 2000 * 150% = $3000 worth of ETH to secure her debt position; and pay back 2000 + 2000 * 14.16% *15/365 = 2011.6384 DAI in 15 days after this loan was originated.

- Lenders can deposit their assets (amount A) with selected term T and withdraw the principal (A) and accrued interest after the deposit matures. Lenders are given the flexibility to choose [T] between [1, M] as well, so that they can adjust their supply time according to the market. After a lender deposits into the protocol with term [T], his or her deposit will be stored inside PoolGroup[T]; since this pool rotates from { PoolGroup[M], PoolGroup[M-1], …, PoolGroup[0] } and accepts deposits at different T, the protocol assigns a weight (W) to each deposit, ensuring the fairness of interest allocation for depositors.

$$W = A*T \qquad (2)$$

  For example, if Bob deposits 100 DAI for 15 days, his deposit will be stored at PoolGroup[15] and his deposit weight W = 100* 15 = 1500. If after 5 days, Charlie deposits another 100 DAI for 10 days, his deposit will be stored in the same pool as Bob's deposit and his W = 100 *10 =1000. If this pool accrued 20 DAI in interest after it matures, Charlie would receive 20 * 1500 / 2500 = 12 DAI as interest, and Charlie will receive 20 * 1000 / 2500 = 8 DAI as interest.

The decoupling of liquidity pools, terms, borrowing rates and deposit weights allows us to optimize the efficiency of the protocol continually. We will discuss the optimization process further in part II.

Tradable CD (Certificate of Deposit)

In Phase 1, the discounting system will be available so that lenders can get instant liquidity by simply selling their Certificates of Deposits (a claim to the underlying deposit and interest at maturity) to an open market with discounts.

For example, if Alice deposits $100 worth of DAI with a 30-day term; on Day 20, she is entitled to the principle and $4 in interest. Alice decides that she needs to withdraw her DAI; she can sell her Certificate of Deposit to an open auction contract at an initial price of $102. If Bob and Carol start to bid this CD and Bob wins the auction at the price of $103; when the loan matures in 10 days, Bob can recoup the principal and interest of this deposit.

PART II

Deposit Weight Model

As we discussed above, the purpose of deposit weight model is to ensure the fairness of deposit interest allocation; we will employ the basics function
W = A T at first and continue to train this model as we gather more real deposit data. The mathematics explanations behind this training are as follows:*

A precise weight distribution function can be defined as $F(A_1,T_1;\ A_2,T_2;\ A_3,T_3\ldots)$, *the value of the function F are in the range [0,1].* $F_i\ (A_1,T_1;\ A_2,T_2;\ A_3,T_3\ldots)$ *represents the deposit weight of user i and* F_i *meets identity equation :* $\sum_{i=1}^{n} F_i = 1$, *n is the total number of users in a single pool. Thus the basics model can be written in the following format:*

$$F_i = \frac{A_i * T_i}{\sum_{i=1}^{n} (A_i * T_i)}$$

Another potential solution of $F_i\ (A_1,T_1;\ A_2,T_2;\ A_3,T_3\ldots)$ *could be* ***softmax function,*** *where* $F_i = \frac{\exp(A_i * T_i)}{\sum_{i=1}^{n} \exp(A_i * T_i)}$

in addition to the two options above, we can iterate on the weight distribution function through real data training. First, we need to define a loss function which indicates the "badness" of model performance; the most common loss function is ***the mean square error*** *which is defined in the following equation :*

$$E = \frac{1}{2n} \sum_{i=1}^{n} \left(\tilde{F}_i - F_i\right)^2$$

F_i *is the output of the weight distribution function,* $\tilde{F}_i$ *is the real interest data we collects, and n is the total number of users in a single liquidity pool. The model performs better if the value of loss function is closer to zero and the purpose of our training is to find a function F, which can minimize the value of E.*

Borrow Rate Model

The on-chain borrow rate model allows us to simulate the real world's rate autonomously and we prefer to use the linear regression to model the relationship between the on-chain rate and the real world rate.

1) Linear Model

Currently, we consider the following four factors for $BorrowRate$ ***(BR)*** : maximum term M ,loan term T, borrow rate upper bound H, and lower bound L.

$$BorrowRate = W_1*H + W_2*L + W_3*T + W_4*M + B$$

$$BorrowRate \in [L, H]$$

W_1 、W_2、W_3、W_4 *are Weights,* B *is Basis; they are scalar parameters.*
$\widetilde{BorrowRate}$ $(\widetilde{BR})$ *is the model output which simulates real borrow rate.*
$\widetilde{BorrowRate}$ $(\widetilde{BR})$ *is the real borrow rate we collects; we usually allow some errors between* BR *and* $\widetilde{BR}$.

2) Data Selecting & Loss Function

We collect borrow data {H, L, T, M, $\widetilde{BR}$} from real word to help us minimize the errors. *Suppose hat the number of sample data we collect is n; for the ith sample,* $BR_i = W_1*H_i + W_2*L_i + W_3*T_i + W_4*M_i + B$ and its matrix form is:

$$(BR_1,\ BR_2 \cdots BR_n) = (W_1,\ W_2,\ W_3,\ W_4)\begin{pmatrix} H_1 & \cdots & H_n \\ L_1 & & L_n \\ T_1 & \ddots & T_n \\ M_1 & \cdots & M_n \end{pmatrix} + (B, B \cdots B)$$

Again, we *choose the most common loss function,* ***mean square error,*** *to express the errors between* BR *and* $\widetilde{BR}$. *It can be written as follow:*

$$E = \frac{1}{2n}\sum_{i=1}^{n}\left(\widetilde{BR}_i - BR_i\right)^2 =$$
$$\frac{1}{2n}[\widetilde{BR}_i - (W_1*H_i + W_2*L_i + W_3*T_i + W_4*M_i + B)]^2$$

we want to find a set of model parameters, referred to as W_1, W_2, W_3, W_4, B *, to make the loss value* E *minimum.*

3) Analytical Solution

Since our model and loss function are relatively simple, the minimum value of the loss function can be derived directly. This type of solution is called analytical solution.

Now we express the loss function in the matrix multiplication form and rewrites H_i, L_i, T_i, M_i *as* X_{1i}, X_{2i}, X_{3i}, X_{4i} *respectively. For the convenience of calculation, we rewrite W as* $\overline{W}$= (W_1 、W_2、W_3、W, B)*, and adds a row of 1s at the end of the X.*

$$\overline{X} = \begin{pmatrix} H_1 & \cdots & H_n \\ L_1 & & L_n \\ \vdots & \ddots & \vdots \\ 1 & \cdots & 1 \end{pmatrix},$$

$$\boldsymbol{E} = \frac{1}{2\boldsymbol{n}} \sum_{i=1}^{\boldsymbol{n}} \left(\tilde{\boldsymbol{BR}}_{\boldsymbol{i}} - \boldsymbol{BR}_{\boldsymbol{i}} \right)^2$$

$$= \frac{1}{2\boldsymbol{n}} \sum_{\boldsymbol{i}=1}^{\boldsymbol{n}} [\tilde{\boldsymbol{BR}}_{\boldsymbol{i}} - (\boldsymbol{W}_1 * \boldsymbol{H}_{\boldsymbol{i}} + \boldsymbol{W}_2 * \boldsymbol{L}_{\boldsymbol{i}} + \boldsymbol{W}_3 * \boldsymbol{T}_{\boldsymbol{i}} + \boldsymbol{W}_4 * \boldsymbol{M}_{\boldsymbol{i}} + \boldsymbol{B})]^2$$

$$= \frac{1}{2\boldsymbol{n}} \sum_{\boldsymbol{i}=1}^{\boldsymbol{n}} [\tilde{\boldsymbol{BR}}_{\boldsymbol{i}} - (\boldsymbol{W}_1 * \boldsymbol{X}_{1\boldsymbol{i}} + \boldsymbol{W}_2 * \boldsymbol{X}_{2\boldsymbol{i}} + \boldsymbol{W}_3 * \boldsymbol{X}_{3\boldsymbol{i}} + \boldsymbol{W}_4 * \boldsymbol{X}_{4\boldsymbol{i}} + \boldsymbol{B})]^2$$

$$= \frac{1}{2\boldsymbol{n}} \left(\tilde{\boldsymbol{BR}} - \overline{\boldsymbol{W}} * \overline{\boldsymbol{X}} \right) * \left(\tilde{\boldsymbol{BR}} - \overline{\boldsymbol{W}} * \overline{\boldsymbol{X}} \right)^{\boldsymbol{T}}$$

** is the matrix multiplication, T is the transpose of the matrix.*

Next, we calculate the derivative of loss function ***E.***

$$\frac{\partial \boldsymbol{E}}{\partial \overline{\boldsymbol{W}}} = \nabla_{\overline{\boldsymbol{W}}} \left(\frac{1}{2\boldsymbol{n}} \left(\tilde{\boldsymbol{BR}} - \overline{\boldsymbol{W}} * \overline{\boldsymbol{X}} \right) * \left(\tilde{\boldsymbol{BR}} - \overline{\boldsymbol{W}} * \overline{\boldsymbol{X}} \right)^{\boldsymbol{T}} \right)$$

$$= \frac{1}{\boldsymbol{n}} \left[\nabla_{\overline{\boldsymbol{W}}} \left(\tilde{\boldsymbol{BR}} - \overline{\boldsymbol{W}} * \overline{\boldsymbol{X}} \right) \right] * \left(\tilde{\boldsymbol{BR}} - \overline{\boldsymbol{W}} * \overline{\boldsymbol{X}} \right)^{\boldsymbol{T}}$$

$$= -\frac{1}{\boldsymbol{n}} \overline{\boldsymbol{X}} * (\tilde{\boldsymbol{BR}} - \overline{\boldsymbol{W}} * \overline{\boldsymbol{X}})^{\boldsymbol{T}} = \boldsymbol{0},$$

$$\overline{\boldsymbol{W}} = \tilde{\boldsymbol{BR}} * \overline{\boldsymbol{X}}^{\boldsymbol{T}} * \left(\overline{\boldsymbol{X}} * \overline{\boldsymbol{X}}^{\boldsymbol{T}} \right)^{-1}$$

As a result, we obtain the optimal values of W and B. These two values can make the loss function E reach the minimum value, and $\boldsymbol{BorrowRate}$ *can fit the real world rate better.*

$$\overline{\boldsymbol{W}} = \tilde{\boldsymbol{BR}} * \overline{\boldsymbol{X}}^{\boldsymbol{T}} * \left(\overline{\boldsymbol{X}} * \overline{\boldsymbol{X}}^{\boldsymbol{T}} \right)^{-1}$$

$$\overline{\boldsymbol{W}} = (\boldsymbol{W_1}、\boldsymbol{W_2}、\boldsymbol{W_3}、\boldsymbol{W_4}、\boldsymbol{B})$$

For both deposit weight and borrow rate model, we will consider more complex parameters in the future, such as, the amount of borrow, the utilization of liquidity pols and etc.