iDOL White Paper

Lien Protocol*

Version 1.0: April 1, 2020

Abstract

In this paper, we propose a decentralized finance (DeFi) platform that provides a new type of stable coin. The platform, called *Lien*, first slices Ether into two tranches of derivatives, *Solid Bond Token* (SBT) and *Liquid Bond Token* (LBT). Almost all the exchange-rate risk in terms of the fiat value of Ether is absorbed by LBT, which will make the price of SBT stable. Our stable coin, the *iDOL token*, is created as a representative money backed by SBT. Unlike other extant crypto-collateralized stable coins, our system requires no over-collateralization, and no manual adjustment of parameters is necessary to maintain the pegged exchange rate, with the price of the iDOL token automatically adjusted to its target level through a market mechanism.

*Contact Author: lien-protocol@protonmail.com

Part I

Introduction

1 Price Volatility: Take it or Hedge it

Some people argue that cryptocurrency is more of a speculative phenomenon than a real currency. It is true that few people actually use cryptocurrency to buy a cup of coffee or pay their rents. A majority of people participating in the cryptocurrency market are simply holding cryptocurrencies while expecting the prices to rise and hoping to make a profit. These speculators are happy to take a risk for a greater profit, sometimes even circumventing regulations that limit leveraged trading in cryptocurrency.

Other people think that cryptocurrency is not suitable as a payment method because of its price volatility. For example, according to a survey conducted by Meter,[1] almost 90 percent of the respondents worry about price volatility and, of those who own cryptocurrencies, 60 percent said that volatility is the most inconvenient aspect of cryptocurrency. Indeed, if a user wants to conduct daily transactions or write a smart contract for businesses, price volatility would be a serious concern.

This situation is Pareto inefficient. On one hand, we have risk-takers willing to take on the exchange-rate risk (i.e. price volatility) in exchange for return. On the other hand, there are risk-hedgers who would like to use cryptocurrency only if the price was stable. Here, one way to improve the situation would be to have the risk-takers insure exchange-rate risk through a derivative contract, which the risk-hedgers can buy to mitigate the risk associated with price volatility.

[1] https://www.prnewswire.com/news-releases/new-study-shows-crypto-volatility-biggest-barrier-to-mainstream-adoption-300756698.html accessed on 03/25/2020.

2 Project Overview

In this project, we develop *Lien*, a new algorithmic system for issuing a stable coin that is backed by cryptocurrency and pegged to a fiat currency. The system minimizes the volatility of the token price by having speculators insure the risk. The supply of our stable coin, the *iDOL token*, is managed by a smart contract deployed on Ethereum; once it is launched, the system will automatically keep issuing stable coins to the market without human intervention, just as the blockchain network behind it does not require middlemen to keep operating.

We construct the stable coin system in two steps. First, we introduce a class of derivatives, the *Solid Bond Token* (SBT) and *Liquid Bond Token* (LBT). These two tokens are generated by splitting, or "tranching," Ether. On the maturity date, the derivative contract (a smart contract deployed on Ethereum) returns a fixed dollar amount to the SBT holder (whenever possible) and returns the remaining amount (if any) to the LBT holder. As you can see, (almost) all the exchange-rate risk will be assumed by the LBT holder, minimizing the volatility of SBT.

Second, we construct a basket of SBTs with various maturity dates and develop the iDOL token as a stable coin backed by the basket. Anyone can "sell" (deposit) SBTs to an iDOL contract and receive the corresponding iDOL tokens. The token holders can also spend their iDOL tokens to "buy" (redeem) SBTs. The amount of the iDOL tokens issued by the iDOL contract is adjusted to the value of the SBTs held by the iDOL contract.

iDOL is a representative money in the sense that its value is backed by a basket of SBTs held by the iDOL contract. The value of each SBT remains stable as it is part of "senior tranches" within the system and (almost) all the price volatility of ETH is absorbed by the LBT which is paired to the SBT. In this way, a fiat exchange rate of the iDOL token is kept stable.

3 Comparison with Extant Stable-Coin Projects

The iDOL token is much more stable than any other stable coin provided by various DeFi (decentralized finance) projects currently in existence.

For example, let us look at *DAI*, which is a stable-coin token issued by one of the earliest DeFi projects, MakerDAO. The Collateralized Debt Position (CDP) smart contract, which is a smart contract issuing DAI, mints (borrows) new DAIs when a user deposits Ether as collateral. Due to the highly volatile nature of Ether, the CDP smart contract requires "over-collateralization," where the value of DAIs issued has to be much lower than the total value of Ether the user deposited as collateral. When the collateral value in Ether goes down below a threshold specified by the protocol, the CDP smart contract auctions off the deposited Ether to burn DAIs to keep the tokens over-collateralized. Although this ensures that the collateral value remains high enough compared to the value of issued DAIs, in practice this mechanism alone does not achieve the price stability around the target level.

The mechanism for the price stability of DAI is merely a monetary policy. The CDP contract collects the Stability Fees from DAI holders in the form of interests. By adjusting the level of Stability Fees, MakerDAO controls the supply of and demand for DAI. The Stability Fee is determined by the MKR (Maker) token holders, who obtain dividends from the MakerDAO ecosystem by participating in the protocol governance. In other words, the Stability Fee is adjusted through human intervention rather than automatically, and there is no good algorithm for automatic price stabilization implemented in the protocol.

In contrast, the iDOL token is backed by SBT, the value of which is much more stable than Ether. As such, Lien does not require over-collateralization. Since the value of SBT is stable, we can simply provide as collateral the exact amount of SBTs that correspond to the value of iDOL to be issued. This "exact-collateralization" makes the iDOL token a representative money — anyone can redeem SBT for iDOL. Lien needs no manual adjustment of parameters through an equivalent of the Stability Fee. Once the system is launched, the price of iDOL will be automatically stabilized around the target level through market forces.

Part II

Grand Design

4 Stable Coin as a Derivative Token

A stable coin is a financial asset whose value is pegged to a fiat currency (or other financial assets). For illustration purposes, we explain the construction of a stable coin pegged to the United States dollar (USD), although our scheme can provide a pegging mechanism for many other assets as well.

The stable coin we propose is a *derivative token* built on Ethereum. Ideally, the stable coin should be designed to be completely insensitive to the fluctuation of the USD/ETH exchange rate, as shown in Figure 1.

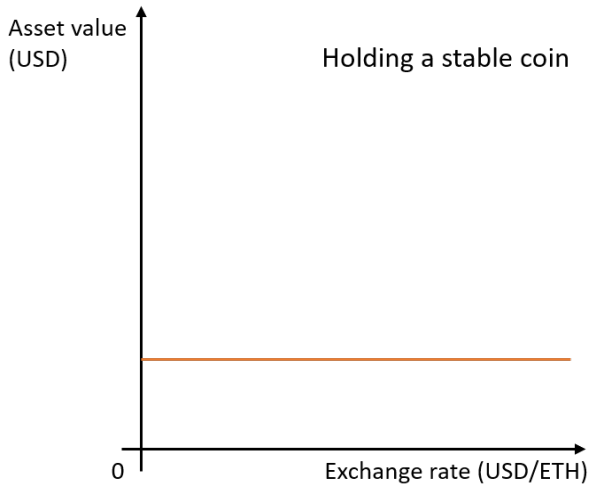
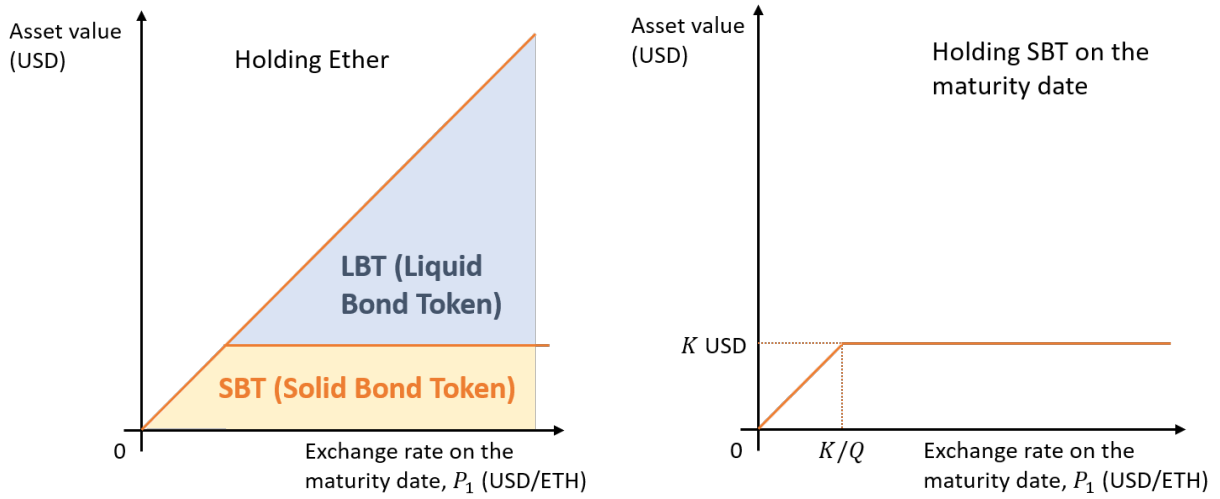


Figure 1: The value of an ideal stable coin.

Our goal is to construct an Ethereum-based derivative token that replicates the return of the ideal stable coin depicted above. The whole derivative contract is written as a smart contract implemented on the Ethereum network and we do not need legal entities for the enforcement of the transactions.



(a) Decomposition of 1 Ether into a Solid Bond Token (SBT) and a Liquid Bond Token (LBT)

(b) Profit from holding an SBT until the maturity date

5 Solid Bond Token and Liquid Bond Token

As a first step, we construct a derivative that simulates the behavior of the ideal stable coin. The 45-degree line in the following figure represents the profit (in USD) gained from holding Ether. Here, we split the return into two parts (Figure 2a).

The derivative contract will be implemented in the following fashion. First, Q ETH is deposited to a smart contract for the issuance of the derivative tokens. The smart contract then returns two tokens, the *Solid Bond Token* (SBT) and *Liquid Bond Token* (LBT). We will then peg the value of SBT to K USD. Now, let P_0 be the USD/ETH exchange rate on the issuance date and P_1 be the exchange rate on the maturity date. The USD/ETH exchange rate, a proxy for the peg will be provided by a reliable oracle (e.g. ChainLink). Upon maturity, the smart contract returns $\min\{K/P_1, Q\}$ ETH to the SBT holder and returns the rest of the deposit, or $Q - \min\{K/P_1, Q\} = \max\{Q - K/P_1, 0\}$ ETH, to the LBT holder.

As long as $Q \geq K/P_1$, or $P_1 \geq K/Q$, the return of the SBT will be K/P_1 ETH $= K$ USD. Hence, if we choose K and Q such that K/Q is sufficiently lower than the USD/ETH exchange rate on the issuance date (P_0), the SBT will end up being worth K USD on the maturity date with high probability. Since the value of SBT in USD is fairly stable, we can

use SBT to back up the value of our stable coin.

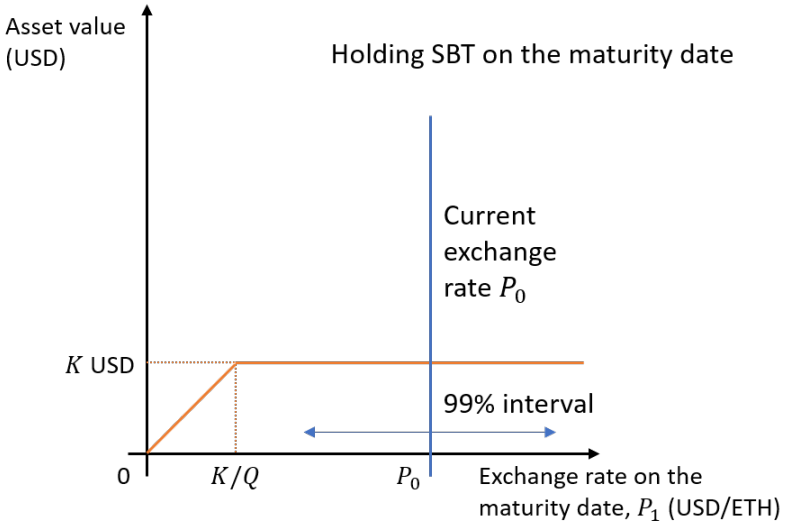


Figure 3: An SBT that is likely to pay off K USD upon maturity.

If you are concerned about the USD/ETH exchange-rate risk, you can hold SBT to hedge the risk. Being stable against USD, an SBT will, with high probability, pay out a fixed value to its holder upon maturity. For example, parties who want to write a business agreement as a smart contract will benefit from holding SBT.

In contrast, if you want to speculate on the changes in the exchange rate, you can use LBT and capitalize on its volatility. Note that, as long as the rate is higher than K/Q, the LBT holder takes all the risks. We expect a demand for LBT from investors who wish to speculate on the price development of Ether. LBT is riskier than holding Ether because its value becomes zero when the exchange rate goes below the threshold (K/Q). An investor can engage in leveraged trading by holding LBTs and, unlike leveraging through debt-finance, he does not have to provide a collateral to conduct the leveraged transaction. Hence, there is no margin call even if the exchange rate falls.

Given a large number of speculators in the cryptocurrency market, we could expect a reasonably high demand for LBT. At the same time, the demand for SBT will also be large because people can use the token as a hedge against the exchange-rate risk. We can therefore expect people with diverse risk profiles to generate many SBTs and LBTs by putting on Ether

as collateral. LBT will be traded as a class of liquid speculative assets on various platforms such as cryptocurrency exchange platforms while SBT will be utilized as a component for our stable coin, the mechanism of which will be described later.

Even if we choose K/Q to be much lower than P_0, there is a probability of the USD/ETH exchange rate dropping dramatically, causing the exchange rate on the issuance date to become lower than K/Q. In such a scenario, the value of SBT is no longer pegged to K USD. This is an inherent risk in the system which cannot be eliminated. In the unlikely event where the exchange rate does end up below K/Q, we would have to require the LBT holder to provide an additional deposit if we wanted to keep the value of SBT pegged to the target level. However, because we do not have any external enforcement scheme (such as a legal enforcement or reputation mechanism) that can "force" the LBT holder to provide additional deposit to keep the value of STB stable, the peg would be lost in this scenario.

6 Aggregation

If the parameters, the volume Q and the target K, are chosen appropriately, holding SBT can help reduce the exchange-rate risk. However, SBT is different from fiat currencies or other typical stable coins in that it has a maturity. The value of the "ideal stable coin" we discussed earlier is always equal to the value of the pegged asset and parties can hold the coin for as long as they want. In contrast, the peg of SBT to USD will be maintained only until the maturity date, at which point the SBT holder is forced to resolve the current position. This is because the LBT holder has agreed to insure the risks associated with the exchange rate only until the maturity date. If the SBT holder wants to continue to hedge the risk, she has to (i) renew the derivative contract, or (ii) find another counterparty that is willing to undertake the risk.

Now, if there are plenty of SBTs with different maturity dates, people can continuously hedge the exchange-rate risk by buying a new SBT when an old one expires. Our stable

coin, *iDOL*, imitates the ideal stable coin by automatically executing this procedure.

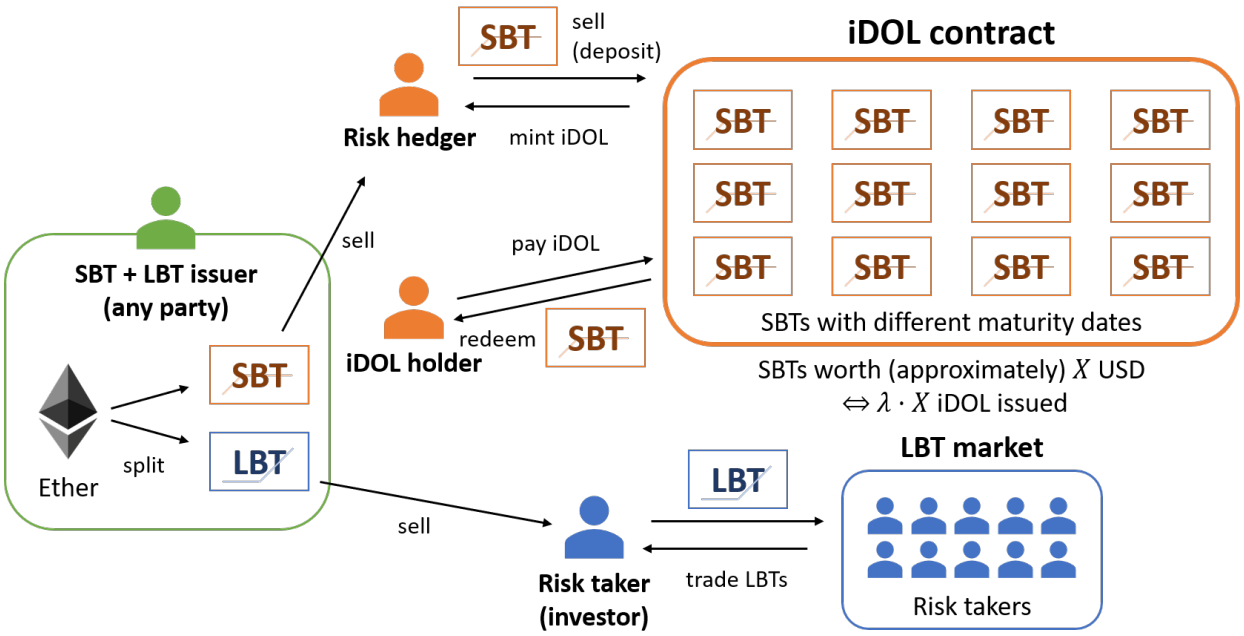


Figure 4: Flows of Ether, SBT, and LBT.

The exact mechanism works in the following manner: First, we launch a smart contract named the *iDOL contract*. The contract holds many SBTs with various maturity dates, volumes, and targets. The contract can also issue the iDOL token. Anyone can have the contract issue new iDOL tokens by "selling" (depositing) SBT to the contract, while iDOL holders can also purchase SBT with the iDOL token.

The value of the iDOL token is backed by the value of SBTs deposited to the iDOL contract. Since the value of SBTs denominated in USD is stable as long as the current price of Ether in USD is sufficiently larger than K/Q, the value of the iDOL tokens created from those SBTs will also be stabilized in the USD term. Furthermore, unlike the token (SBT) that backs up its value, the iDOL token has no maturity date. Whenever a speculator establishes a long position in LBT, SBT is also produced, thus creating a constant supply of SBT. As long as SBTs are supplied to the iDOL contract, the iDOL token holders can rest assured knowing that its value will be kept stable.

Part III

iDOL Contract

7 Minting: SBT to iDOL

Anyone can mint new iDOL tokens by depositing SBT into the iDOL contract. Consider a situation where the current USD/ETH exchange rate is P and you have an SBT which is created from sending Q ETH to the contract, with the maturity and the target price set to M and K USD, respectively. When K is sufficiently lower than PQ and M is moderately small (the maturity date is assumed to be relatively close to the current time), the SBT almost certainly returns K USD (more precisely, the ETH tokens worth K USD) on the maturity date. The iDOL contract only accepts such least risky SBTs (i.e. SBTs whose prices are expected to remain close to the target price upon maturity). The maximum value of the deposited SBT is capped to K USD and, unless the price of ETH falls dramatically, the actual market value of the SBT should remain close to K USD.

Let λ iDOL/USD be the current target exchange rate between iDOL and USD. We start from a fixed rate of $\lambda = 1$ but it will be adjusted in an emergency situation (described later). By depositing a least risky SBT with a target parameter set to K USD, you can mint λK iDOL. You can receive $\lambda(1-\beta)K$ iDOL immediately when you deposit SBT, where $\beta \in (0, 1)$ is a *buffer parameter* preset by the iDOL contract. The rest of minted iDOL tokens, $\lambda\beta K$, is reserved by the iDOL contract and all or part of it will be returned to you when the deposited SBT is sold out.

The buffer parameter β is set in such a way that the total value of the deposited SBT never becomes less than the value of $\lambda(1-\beta)K$ iDOL. Since the risk of an SBT depends on the maturity date M and the the target/volume ratio K/Q, the buffer parameter should be chosen as a function of M and K/Q. After agreeing on the risk tolerance, we can estimate of

the volatility of the USD/ETH exchange rate from the market data and use it for calculating the actual value of the buffer parameter.

How the iDOL contract returns the reserve depends on the revenue from the redemption procedure. If the price of the SBT, R, is larger than λK (i.e. $R > \lambda K$), a full amount of the reserve, $\lambda\beta K$ will be returned to you (Note that, even in such a case, the depositor's additional return does not exceed his initial reserve). If the price R falls somewhere between λK and $\lambda(1-\beta)K$ (i.e. $\lambda(1-\beta)K \leq R \leq \lambda K$), only $R - \lambda(1-\beta)K$ is returned to you and the rest of the reserve, $\lambda K - R$, will be burned. Therefore, $(\lambda K - R) + R = \lambda K$ iDOL is burned in total, which equals the amount issued when the SBT was deposited. If the price R is smaller than $\lambda(1-\beta)K$ (i.e. $R < \lambda(1-\beta)K$), you will not get anything returned and the whole reserve $\lambda\beta K$ will be burned. However, you will not be required to make an additional deposit either. Note that, although the burned reserve is minted, it is never circulated in the market.

The idea behind this minting procedure is as follows. Let us suppose you bring an SBT that is worth R iDOL to the basket. In this case, you have the right to receive R iDOL ($= R/\lambda$ USD) in exchange for the SBT you have deposited. Here, the value of R is determined when the SBT is redeemed, not when it is deposited. Given this, the iDOL contract mints and pays $\lambda(1-\beta)K$ iDOL in advance when the SBT is deposited so depositors do not have to wait until the value is realized in order to receive the iDOL tokens. The buffer parameter β is chosen in such a way that R becomes larger than $\lambda(1-\beta)K$ with high probability. The iDOL contract pays the remaining amount once the precise value of the SBT is determined (i.e. once the SBT is sold out).

In a regular situation (i.e. $\lambda K \geq R \geq \lambda(1-\beta)K$), R iDOL will be minted when the SBT, which is worth R iDOL, is deposited. When the SBT is redeemed, R iDOL will be burned by the iDOL contract. This ensures that the values of the deposited SBT and the minted iDOL tokens are always identical.

If $R > \lambda K$, λK iDOL will be minted from this SBT and R iDOL will be burned when

it is auctioned off. However, this will cause the value of iDOL to appreciate since $R > \lambda K$. In such an irregular situation, we revalue the iDOL token by decreasing the target exchange rate λ and the gains resulting from the revaluation would be equally shared across all iDOL holders.

If $R < \lambda(1-\beta)K$, the depositor will not be able to obtain additional payments upon maturity. In such a case, $\lambda(1-\beta)K - R$ iDOL is minted but remains unburned. resulting in iDOL being diluted. In such an emergency situation, we devalue the iDOL token by increasing the target rate λ and the losses caused by the devaluation would be equally shared across all iDOL holders. (See Section 9)

8 Redemption: SBT to iDOL

8.1 Opportunities

When either (i) the maturity date nears or (ii) the USD/ETH exchange rate drops and the risk of holding SBT increases, the deposited SBT will be auctioned off. During the auction, payments are made in iDOL, with such payments subsequently burned by the iDOL contract. In addition, before the auction happens, anyone can redeem their deposited SBT by paying its full prices.

8.2 Regular Auctions

As the maturity date of an SBT approaches, it will be sold in a *regular auction*. The auction will be conducted following the *multi-unit Vickrey auction* (Vickrey 1961) format, which is believed by many researchers in the field to be one of the best auction formats when it comes to designing a multi-unit auction.

Strategy-Proofness The Vickrey auction satisfies the property called *strategy-proofness*, which makes truthful bidding one of the most profitable strategies regardless of the

other agents' behaviors. For example, if you believe that the value of the auctioned SBT should be evaluated with the Black-Scholes formula, you should bid a price based on the formula. This property eliminates all the strategic complications from the "game" and facilitates the participants' decision making processes.

Ex Post Individual Rationality Each participant is never charged more than the willingness to pay she declared as her bid, which reduces the risk of participating in the auction.

Efficiency In this truthful equilibrium (i.e. an equilibrium in which all agents truthfully reveal their valuations as their bids), the resultant allocation of SBT maximizes the benefits enjoyed by the participants.

Maximum Revenue Among all possible auction formats that satisfy strategy-proofness, ex post individual rationality, and efficiency, the Vickrey auction yields the largest possible revenue. This incentivizes depositors to create new SBTs and deposit them to the iDOL contract.

False-Name Proofness In auctions held on a blockchain, an agent easily can submit multiple bids using multiple accounts. In the Vickrey multi-unit auction, no agent can increase her profit by using multiple accounts.

It is also worth pointing out that, with some additional technical assumptions, the Vickrey auction can be proven to be the only mechanism that satisfies strategy-proofness, ex post individual rationality, efficiency, and a maximum revenue (Green and Laffont 1977, 1979; Holmström 1979). You can learn more about the theoretical advantages of the Vickrey auction by seeing, for example, Chapter 12 and 13 of Krishna (2009).

Now, let us assume that the target iDOL/USD exchange rate when the SBT is deposited is λ_0 (this is different from the current iDOL/USD exchange rate), the current USD/ETH exchange rate is P, the auctioned SBT's volume (i.e. the amount of ETH used to create

the SBT) is Q ETH, the maturity parameter is M, and the target price is K USD. The reservation price is set to $r = \lambda_0(1-\beta)K/Q$. In this multi-unit sealed-bid VCG auction, the auctioned SBT is regarded as a divisible good, and bidders can claim any fraction of the SBT. Specifically, each participant i submits her *bidding list*, $B_i = (b_i^1, b_i^2, \ldots)$, with each *bid* b_i^h composed of the price p_i^h and quantity q_i^h. By placing the bid b_i^h, the agent i essentially declares, "I want to buy the auctioned SBT up to q_i^h units if the unit price for the SBT is lower than p_i^h." The reservation price is $r = \lambda_0(1-\beta)K/Q$ and all bids with a lower price than the reservation price are ignored. Since the auctioned SBT is created from Q ETH in total, if an agent wins q units of it, he obtains an SBT whose volume is q ETH and whose target is $K \times q/Q$ USD. Each bidder i can make such a bid, b_i^h, as many times as she wants. The maximum amount of units the bidder i may buy with this bidding list is $\sum_h q_i^h$ whose price is $\sum_h p_i^h q_i^h$.

When the agent i submits a bidding list, she must deposit at least $\sum_h p_i^h q_i^h$, which corresponds to the maximum amount to be paid in the auction. Using a standard commitment scheme, each agent commits to a bidding list without disclosing the details. Once the bid deadline comes, all agents reveal their bidding lists and the iDOL contract automatically calculates the price and quantity assigned to each agent. Finally, the contract transfers SBTs to each agent and returns the deposit after subtracting the amount paid for the auction.

After collecting all the bidding lists, all bids are sorted by price in descending order. The tied bids will each have an equal chance of being selected. The iDOL contract greedily picks the bids with the highest price into the set of winning bids and stops once the capacity constraint is reached (the volume of the auctioned SBT is Q ETH). Typically, the last winning bidder is not allowed to buy the bid's full amount, q_i^h because of this capacity constraint. In such a case, the last winning bidder buys a fractional amount.

The Vickrey auction does not calculate the amount to be paid by a winner from the willingness to pay he declared, i.e. $(p_i^1, p_i^2, \ldots)$. Instead, the Vickrey auction calculates the winner i's payment amount from the strongest losing bids made by agents other than i. If

the winner i obtains q_i units of SBT in total, we find the highest losing bids $(b^1_{-i}, b^2_{-i}, \ldots, b^H_{-i})$ whose total quantity is $q_i = \sum_h q^h_{-i}$ and determines the amount the winner i is required to pay, which will be $\sum_h p^h_{-i} q^h_{-i}$. If the total quantity of losing bids posted by the other agents is smaller than q_i, the agent i buys $q_i - \sum_h q^h_{-i}$ at the reservation price, $r = (1-\beta)K/Q$, per unit.

Example 1 (Vickrey Auction)**.** Assume that $\lambda_0 = 1$, $P = 100$ (USD/ETH), $Q = 3$ (ETH), and $K = 150$ (USD). Since PQ is sufficiently larger than K, we can assume that the value of the auctioned SBT will likely resolve to K. Let us assume that $\beta = 0.1$ and the reservation price of $r = \lambda_0(1-\beta)K/Q = 45$ (iDOL/ETH).

Suppose there are three agents participating in the auction with the following bids:

$$\begin{aligned}
\text{agent 1}: &\quad b^A_1 = (49.0, 1.0), b^B_1 = (48.0, 0.7), b^C_1 = (47.0, 0.6), \\
\text{agent 2}: &\quad b^A_2 = (49.5, 0.2), b^B_2 = (48.5, 0.8), b^C_2 = (47.5, 0.3), \\
\text{agent 3}: &\quad b^A_3 = (48.5, 0.7), b^B_3 = (46.0, 0.5).
\end{aligned}$$

First, the iDOL contract sorts all bids in descending order based upon their prices. Table 1 shows the sorted bids. Note that b^B_2 and b^A_3 have the same price, with both of them having the equal chance of being selected. In this example, although b^B_2 happens to be listed earlier than b^A_3, the latter has the same probability of being chosen as the former.

bid	b^A_2	b^A_1	b^B_2	b^A_3	b^B_1	b^C_2	b^C_1	b^B_3	r
price	49.5	49.0	48.5	48.5	48.0	47.5	47.0	46.0	45.0
quantity	0.2	1.0	0.8	0.7	0.7	0.3	0.6	0.5	∞
cumulative q	0.2	1.2	2.0	2.7	3.4	3.7	4.3	4.8	∞

Table 1: The sorted bids in Example 1.

Then, the iDOL contract determines the winning bids by sequentially picking up the highest bids, until the total quantity of the winning bids reaches $Q = 3$. In this example, b^A_2, b^A_1, b^B_2 and b^A_3 will be selected as winning bids. Since the total quantity of these bids is

2.7, the iDOL contract can still fill in the remaining quantity, 0.3. Given that the quantity of the next highest bid, b_1^B, is 0.7, which is larger than 0.3, b_1^B is split and has only part of it (0.3 out of 0.7) be included in the winning bids. At this point, the quantity obtained by each agent is finalized: the agent 1 obtains 1.3 units (from b_1^A and part of b_1^B), the agent 2 obtains 1.0 units (from b_2^A and b_2^B), and the agent 3 obtains 0.7 units (from b_3^A).

The amount to be paid by each agent for the iDOL contract is determined as follows. First, the iDOL contract lists all losing bids. Recall that only part of b_1^B is a winning bid and the remaining fraction of b_1^B (which is $0.7 - 0.3 = 0.4$) is incorporated into the list of losing bids.

bid	b_1^B	b_2^C	b_1^C	b_3^B	r
price	48.0	47.5	47.0	46.0	45.0
quantity	**0.4**	0.3	0.6	0.5	∞

Table 2: The losing bids in Example 1.

First, we will calculate the payment to be made by the agent 1, who won 1.3 units. The losing bids made by the agent 1 (i.e. b_1^B and b_1^C) will be ignored when we calculate the payment amount. We sequentially pick b_2^C and b_3^B, but the total quantity for these two bids is 0.8, which is still smaller than 1.3. For the remaining 0.5 units, we use the reservation price, $r = 45.0$, to calculate the payment amount. Hence, to acquire 1.3 units, the agent 1 pays

$$\underbrace{47.5 \times 0.3}_{p_2^C \times q_2^C} + \underbrace{46.0 \times 0.5}_{p_3^B \times q_3^B} + \underbrace{45.0 \times 0.5}_{r \times \left[q_1 - \left(q_2^C + q_3^B\right)\right]} = 59.75 \text{ iDOL}.$$

Next, let us look at the agent 2's payment amount. The agent 2 acquired 1.0 units. Since b_2^C is posted by agent 2 herself, we should exclude it. Given that the total quantity of b_1^B (excluding the winning part) and b_1^C is exactly equal to 1.0, the agent 2 pays

$$\underbrace{48.0 \times 0.4}_{p_1^B \times q_1^B} + \underbrace{47.0 \times 0.6}_{p_1^C \times q_1^C} = 47.4 \text{ iDOL}.$$

Finally, we calculate how much the agent 3 pays for the 0.7 units she acquired. We should exclude b_3^B following the same procedure, although its price is rather small and it would not have a material impact on the calculation if we included it. Here, the total quantity of b_1^B (excluding the winning part) and b_2^C is exactly 0.7. Hence, the agent 3 pays

$$\underbrace{48.0 \times 0.4}_{p_1^B \times q_1^B} + \underbrace{47.5 \times 0.3}_{p_2^C \times q_2^C} = 33.45 \text{ iDOL}.$$

This terminates the Vickrey auction.

Let us denote the revenue of the auction by R. Since we set the reservation price (per unit) to $r = \lambda_0(1-\beta)K/Q$, the revenue R should be larger than or equal to $\lambda_0(1-\beta)K$. The amount $\lambda_0(1-\beta)K$ is burned and the rest of the revenue, $R - \lambda_0(1-\beta)K$, is returned to the depositor of the auctioned SBT. The rest of the reserve, $\lambda_0 K - R$, is burned.

Example 1 (continued)**.** Since $\lambda_0 K = 150$ iDOL, this SBT generates 150 iDOL when the SBT is deposited. $\lambda_0(1-\beta)K = 135$ iDOL is immediately allocated to the depositor, with $\lambda_0 \beta K = 15$ iDOL reserved by the iDOL contract. The revenue R for the auction is the sum of the total amounts paid by all agents and is calculated as follows:

$$R = \underbrace{59.75}_{\substack{\text{Agent 1's} \\ \text{payment}}} + \underbrace{47.4}_{\substack{\text{Agent 2's} \\ \text{payment}}} + \underbrace{33.45}_{\substack{\text{Agent 3's} \\ \text{payment}}} = 140.6 \text{ iDOL}.$$

On the other hand, $\lambda_0(1-\beta)K = (1-0.1) \times 150 = 135$ iDOL. Hence, the depositor of the SBT additionally receives $140.6 - 135 = 5.6$ iDOL from the reserve, and the rest of the reserve, $15 - 5.6 = 9.4$ iDOL is burned.

Remark 1. The target parameter K (relative to the volume Q) and the buffer parameter β are set in such a manner that the demand for the SBT exceeds Q units given the reservation price of $r = \lambda_0(1-\beta)K/Q$. For this reason, the total quantity demanded by various agents will likely reach or exceed Q. The iDOL contract is thus expected to sell all SBTs that it

holds. If the demand for the SBT fails to reach Q units, however, the contract decreases the reservation price r and holds an auction again. This process continues until a buyer is found who is willing to buy the SBT. If the auction price becomes lower than $\lambda_0(1-\beta)K/Q$, then the iDOL contract fails to burn all the iDOL tokens associated with the auctioned SBT. In that case, the total supply of iDOL tokens exceeds the amount of SBTs that back them up and, therefore, we must execute a devaluation of iDOL. (See Section 9)

8.3 Emergency Auctions

Thus far, we have seen how an auction is held when the maturity date nears. However, an auction is also triggered if the value of the SBT decreases significantly due to a huge drop in the USD/ETH exchange rate. Here, we have to make sure that the value of 1 iDOL is pegged to $1/\lambda$ USD, which we can achieve by having the iDOL contract only accept least risky SBTs. When an SBT gets riskier (i.e. when the possibility of SBT's value crashing down increases), the contract will eject the SBT through an auction. This *emergency auction* will be held when it is expected that the revenue from the auction may not reach a certain threshold value, which is larger than $\lambda_0(1-\beta)K$. As long as the revenue is larger than $\lambda_0(1-\beta)K$, a devaluation of iDOL will not occur.

Emergency auctions are conducted exactly the same way as a regular auction: We have a Vickrey auction with the reservation price of $r = \lambda_0(1-\beta)K/Q$ and with the price gradually reduced until all SBTs are sold out.

8.4 Full-Price Redemption

You can also redeem the deposited SBT before it is auctioned off by paying its full price. While you would normally wait until the SBT is auctioned off to redeem the token, you can utilize the full-price redemption option and take advantage of an additional arbitrage opportunity.

Since an SBT with a target parameter K USD pays off at most K/P_1 ETH $= K$ USD

on the maturity date, the value of the SBT does not exceed K USD. Hence, if you pay λK iDOL (which is worth K USD), you can redeem the SBT immediately.

This is equivalent to buying a full volume (Q) of the SBT at the price of λK iDOL. After the full-price redemption, we return the reserve to the original depositor following the protocol procedure. If $\lambda K > \lambda_0 K$ (recall that λ denotes the current iDOL/USD exchange rate while λ_0 denotes the iDOL/USD exchange rate when the deposit was made), the iDOL contract returns the full amount of the reserve, $\lambda_0 \beta K$, to the depositor. If $\lambda_0(1-\beta)K \leq \lambda K \leq \lambda_0 K$, then the iDOL contract returns $\lambda K - \lambda_0(1-\beta)K$ to the depositor, with the remaining reserve burned. If $\lambda K < \lambda_0(1-\beta)K$, the whole reserve is burned and the depositor obtains nothing.

9 Revaluation/Devaluation

This section describes how the protocol determines the exchange rate of the iDOL token (λ). Because the iDOL token should function as a stable coin, the system must be designed to stabilize its exchange rate as much as possible. However, under some special circumstances, the fixed exchange rate cannot be maintained. In such emergency cases, the iDOL contract algorithmically executes revaluation or devaluation of the token in order to keep the Lien protocol operating.

The devaluation or revaluation could occur only when the deposited SBT is sold at an abnormal price. The protocol chooses the buffer parameter β such that the price of the SBT with the target parameter K USD always lies in the range $[\lambda(1-\beta)K, \lambda K]$. As long as this condition holds, the iDOL contract never changes the target exchange rate, λ. When the SBT worth R iDOL ($\approx \lambda K$ USD) is deposited, R iDOL is newly minted. When redeemed, R iDOL is burned. Throughout this minting and burning procedure, the value of the iDOL tokens is kept stable, with no depreciation or appreciation occurring.

The system needs to devalue iDOL when the revenue is smaller than $\lambda(1-\beta)K$ iDOL

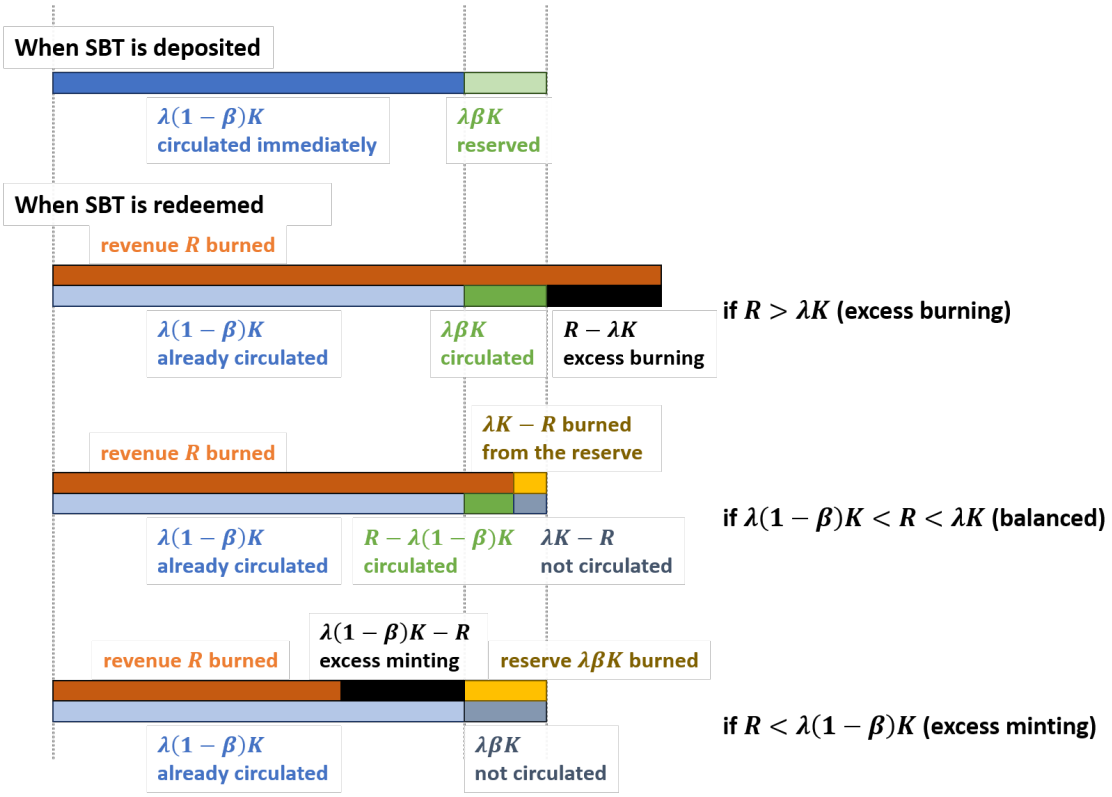


Figure 5: The balance of the iDOL minting and burning.

(more precisely, $\lambda_0(1-\beta)K$, if the exchange rate has changed after the deposit). When the deposit is made, $\lambda(1-\beta)K$ iDOL is already released to the depositor and the iDOL contract cannot force him to return the token. The only thing the iDOL contract can do is to burn the reserve, $\lambda\beta K$, but it is not enough to compensate the reduction in the revenue. Since $R < \lambda(1-\beta)K$, $\lambda(1-\beta)K - R$ is minted when the SBT is deposited but not burned even after the SBT is auctioned off. When such an incident happens, $\lambda(1-\beta)K - R$ iDOL is not backed by the value of the basket.

Currency devaluation of iDOL is executed if and only if such cases occur. Initially, the total amount of iDOL tokens issued by the iDOL contract was X iDOL and its value was backed by a basket of SBTs, whose sum of target values was Y USD. If $R + \lambda\beta K$ iDOL is burned and an SBT with the target value of K USD is redeemed, we will have $X - R$ iDOL backed by $Y - K$ USD. Note that the ratio is not kept the same if $R + \lambda\beta K < \lambda K$, or $R < \lambda(1-\beta)K$, in which case we should devalue iDOL by increasing λ (devalue iDOL) to

keep the ratio stable.

Similarly, when the auction revenue exceeds λK, the excess revenue will not be returned to the depositor. Instead, it will be burned by the iDOL contract. When that happens, the amount of burned iDOL tokens will exceed its minted amount, thereby increasing the value of the iDOL token. In such a case, we should lower the parameter λ (revalue iDOL) to maintain the ratio constant.

The stabilization of the iDOL token through devaluation and revaluation can be achieved in the following simple manner. If there are X iDOL tokens issued and the sum of the target parameters for SBTs in the basket is Y USD, we can set the exchange rate to $\lambda = X/Y$ iDOL/USD. As long as the revenue from redemption ends up in the range $[\lambda(1-\beta)K, \lambda K]$, λK iDOL will be burned when an SBT with the target price of K USD is redeemed, with the value of λ kept stable.

References

Green, J. and J.-J. Laffont (1977): "Characterization of satisfactory mechanisms for the revelation of preferences for public goods," *Econometrica*, 427–438.

——— (1979): *Incentives in public decision-making*, Elsevier North-Holland.

Holmström, B. (1979): "Groves' scheme on restricted domains," *Econometrica*, 1137–1144.

Krishna, V. (2009): *Auction theory*, Academic press.

Vickrey, W. (1961): "Counterspeculation, auctions, and competitive sealed tenders," *The Journal of Finance*, 16, 8–37.