Limit Order Token Reserves - Tokenized Exchange Orders as Hedging Vehicles

Sean Hart

July 25, 2021

Abstract

LOTR tokens are fungible, modularized exchange orders with embedded collateral that exist independent of an execution entity allowing for the composability of trade strategies. These order modules allow separation of order collateral from order redemption, creating a secondary use case as speculative or hedging vehicles. This paper examines this secondary use case more closely.

Introduction

The advent of programmable decentralized ledgers have introduced the ability to create highly complex fungible structured products that can exist independently from their issuing organization. This contract independence allows for the creation of products with direct ownership of embedded collateral that do not require the continued existence of a solvent external entity in an enforceable jurisdiction. This unique feature combination in addition to the rise of decentralized exchanges allows for the creation of previously impossible product types, such as encapsulated and fungible pending exchange orders that automatically execute when specific criteria are reached.

Specifically, Limit Order Token Reserve (LOTR) tokens are Tokenized (encapsulated and fungible) exchange orders with embedded collateral used for exchange payment when the targeted asset is confirmed to have reached the limit price on a decentralized exchange. Minters of LOTRs receive two tokens, one Limit Token representing the receiver of the target token when the limit price is reached, and one Redeem Token required to unlock the embedded collateral in the Limit Token any time before the target token limit price is reached. Both of these tokens are fungible, meaning that they can be paired with any opposing LOTR token, and therefore tradable on secondary markets. Although the original intent of LOTR tokens was to create modular trade strategies, this secondary market composability allows this product to be used as a hedging or speculative vehicle, with owners selling one half of the LOTR Token Pair with the expectation of their half diverging in market price from their purchase price when the underlying asset price changes relative to the LOTR limit order price. The LOTR Token Pair halves always sum to the limit order price (the value of redeemable embedded collateral) minus transfer fees.

Traders will find that the value of LOTR tokens is easier to calculate than perpetual options because they automatically execute when the asset reaches the limit order price, removing the unlimited best price problem. This automatic execution means LOTR tokens are therefore not "options". LOTRs are also more fungible than expiring options because LOTR products have one fewer variable (expiration date). Closer examination will also reveal behavioral similarities with futures contracts as either LOTR token half will converge with the limit price depending on the price of the target token and without the requirement for a product premium payment as with options contracts.

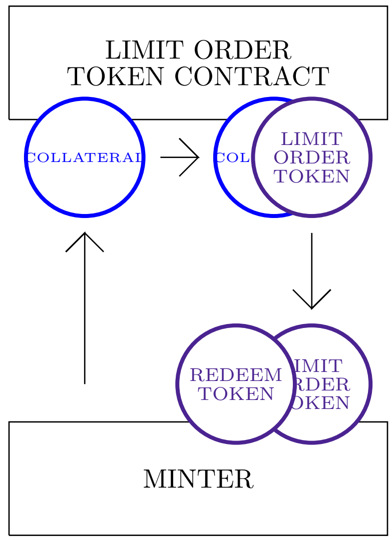
Component Terms

- Target Token Market Price ($Token Price)
- Target Token Limit Price (Limit Price)
- LOTR Limit Order Exchange Token (Limit Token)
- LOTR Collateral Redemption Token (Redeem Token)
- Limit Token & Redeem Token (LOTR Token Pair)

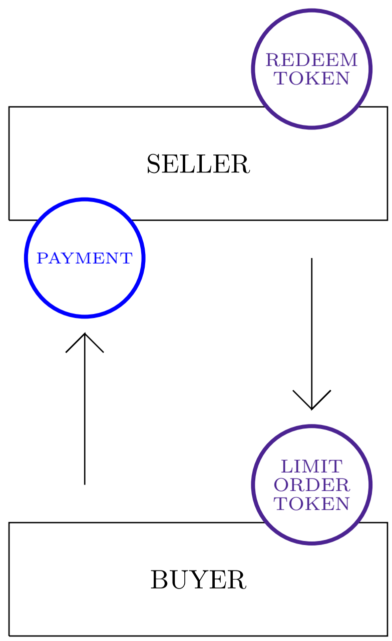
1 Usage

1.1 Minting & Trading

To be used as hedging or speculative vehicles, LOTR tokens must have a minting process that does not affect their price on the open market. Fortunately, LOTR Token Pairs always sum to the value of embedded collateral in the token, which is also the limit price for the limit order. Therefore, any entity can set up a token smart contract that issues the LOTR Token Pairs and stores the ownership balances on embedded ledgers.



All tokens minted by the same contract are fungible since they share a ledger and handling functions.

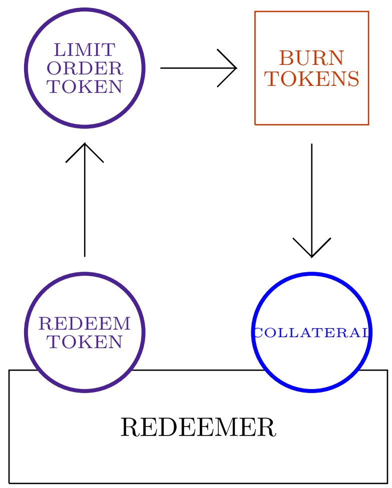


1.2 Outcomes

LOTR tokens have two possible outcomes: redemption of the embedded collateral or execution of the limit order. Both outcomes burn the expended tokens.

1.2.1 Redemption

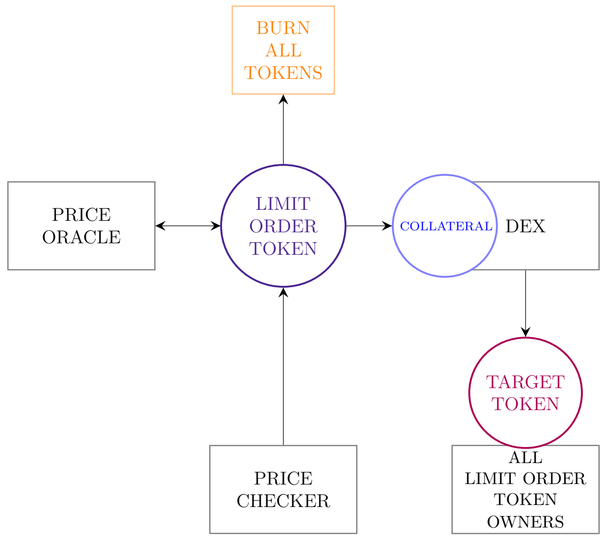
Any holder of both LOTR token halves can choose to redeem the embedded collateral, minus transfer fees, at any time. This process is initiated by passing the Redeem Token address to a function in the Limit Token. This process burns the two token halves involved in the redemption function and transfers the embedded collateral ownership to the final owner of the LOTR Token Pair used (the ownership address of both token halves must be the same).



1.2.2 Execution

The execution outcome is the most complicated function of the LOTR token contract. The end goal of this function is to transfer all embedded collateral to all owners of Limit Tokens, and burn all Limit Order and Redeem Tokens for all current owners. When this function completes, this LOTR token contract's balances are zero and its balances on external token contracts (embedded collateral contracts) are also zero (it should not own any embedded collateral). This process resets the LOTR token contract so it can be used again. This cycle repeats perpetually.

The execution function can be engineered to be triggered in a variety of ways. Either a centralized entity can monitor the market and trigger the function when the limit price is reached, or a decentralized incentive process can be used to encourage any on-chain actor to call the execution function and be rewarded with a cut of fees if the limit price is confirmed to have been reached.



Certain advanced on-chain external contract calls are required in the execution function. First, the market price should be checked to ensure the limit price has been reached. This check should occur whether the execution function is triggered by a centralized entity or a decentralized on-chain actor. Ideally the oracle(s) used to check the market price use a variety of reporting sources to ensure accuracy. It is also recommended to use an oracle related to the exchange used in the swap process to ensure the swap occurs as close to the limit price as possible. Second, an exchange must be called to process the swap orders. Ideally the exchange is decentralized with enough liquidity to process the orders without significant price slippage.

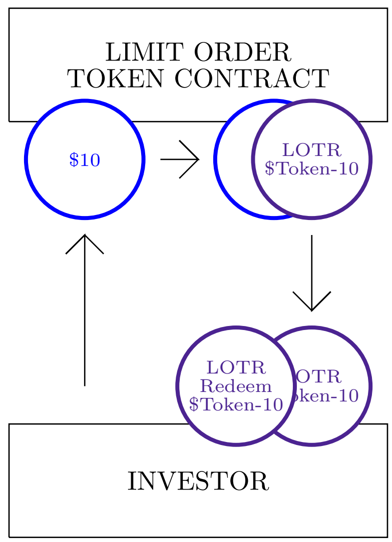
The use of external oracles and exchanges preset opportunities for transaction failure. The execution function should include appropriate fallback procedures. Some fallback options are centralized, such as the LOTR issuer manually retrying execution, but other options are more simple such as releasing the embedded funds to Limit Token owners to allow them to manually attempt the token swap on their own terms. Depending on the complexity of other various setups of the LOTR token contract, these necessary fallback procedures can be more or less complex.

Once the limit price has been passed, the LOTR token contract should also track the new price direction of the current market (below or above the limit price). During the price check or use of a fallback procedure, the directionality is often needed to confirm the market price passed the limit price away from the previous price direction.

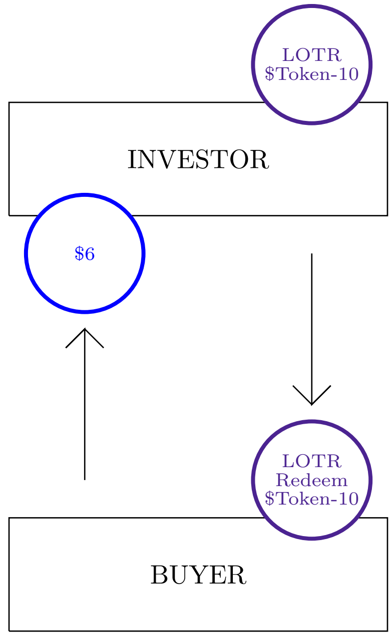
2 Examples

2.1 Hedging Example

Consider an investor seeking to reduce the risk of an investment in $Token. $Token is currently trading at $12 per token. The investor mints an LOTR Token Pair at the limit price of $10, paying $10 in collateral and receiving the Token Pair. Expense = $10.

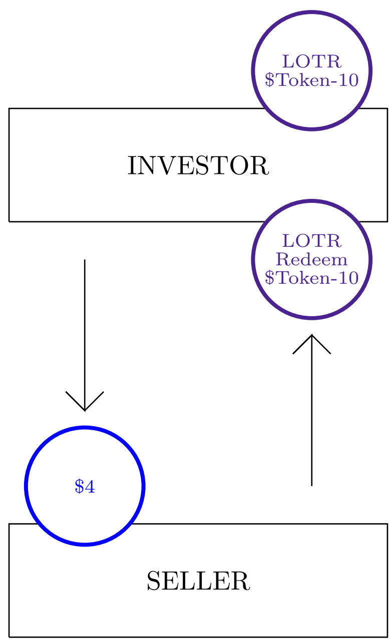


Due to the historical volatility of $Token, the Limit Token is valued at $4 and the Redeem Token is valued at $6. This total equals the value of collateral embedded in the Limit Token. The investor wants to protect their long position and profit if $Token decreases in value in order to offset the loss on their long position. Therefore, they keep the Limit Token and sell the Redeem Token. Revenue = $6.

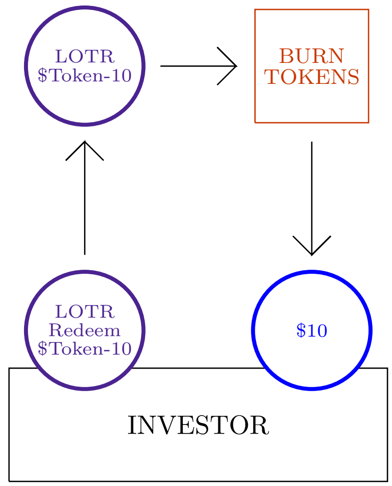


Over time, $Token loses $1 in value, now trading for $11 per token. Because the market price is now closer to the limit order target price, the Limit Token is now valued at $6 per token. Loss = $1, Gain = $2.

The investor does not think $Token will continue to decrease in value, so they buy back a Redeem Token, which is now valued at $4 per token. Expense = $4.



The investor redeems the LOTR collateral of $10. Revenue = $10.



The investor's account net change:

− $10	LOTR mint cost
+ $6	sale of Redeem Token
− $1	unrealized loss in $Token
+ $2	value increase of Limit Token
− $4	repurchase of Redeem Token
+ $10	LOTR collateral redemption
+ $3	

2.2 Speculative Limit Order Example

We will consider a simplified scenario with the limit order above the current market price (see section 3.2 for understanding this possibility). The current market price for $Token is $12 and our Limit Price is $15. Due to the historical volatility of this token, the market values the Limit Token component at $9. We can sell the Redeem Token for $6. $Token market price increases and reaches our limit price of $15. The LOTR token contract executes an exchange swap from collateral into $Token and transfers all resulting $Tokens to Limit Token owners and then burns all Limit Order and Redeem Tokens for all token owners, resetting the LOTR token contract.

− $15	LOTR mint cost
+ $6	sale of Redeem Token
+ $15	market value of final $Tokens transferred after swap execution
+ $6	

Please note that fees were not considered in these examples, and can create considerable outcome differences depending on the hosting blockchain native token price, high gas limits, and other trading fees.

3 Valuation

Note: The discovery of the ideal formula describing the market valuation of LOTR tokens is ongoing. For now we will consider a generalized valuation model that represents the incomplete portion of the model with a single volatility constant (σ) that can be modified according to traders' preferences.

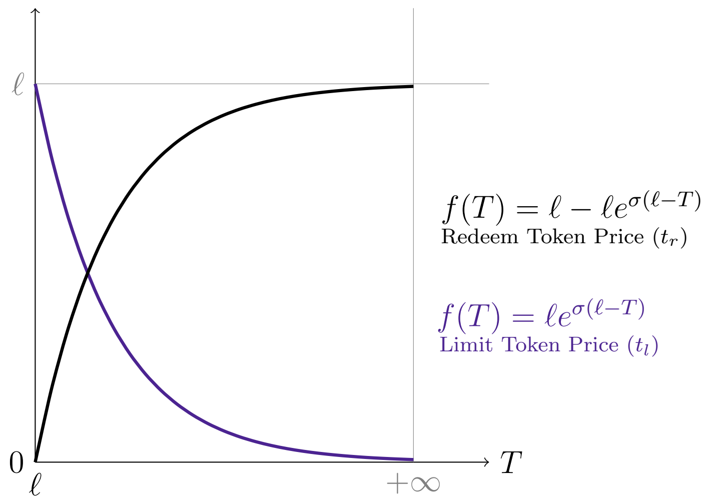
3.1 All Conditions

- $Token Price ($T$)
- $Token volatility constant (σ)
- Limit Price (ℓ)
- Limit Token Price (t_l)
- Redeem Token Price (t_r)

Assuming an efficient market, an LOTR valuation formula must include these conditions (ignoring fees):

- T and t_l converge at ℓ
- T and t_l converge at 0
- $t_l + t_r = \ell$

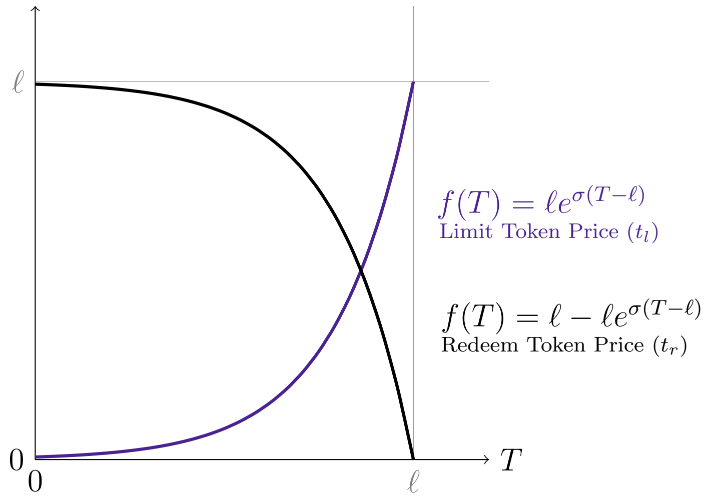
3.2 Limit Orders Below Market Price



When T is greater than ℓ, as T increases away from ℓ, t_l approaches 0 since the chances of T reaching ℓ in the near future diminishes. As T approaches ℓ, t_l also approaches ℓ since the chances of T reaching ℓ and executing the limit order (and receiving the value of ℓ minus fees) increases. t_r follows the inverse of t_l valuation curve, since it is always the difference between ℓ and t_l.

3.3 Limit Orders Above Market Price

Limit Orders above the current market price do not traditionally exist since market makers will automatically fill orders of this type. Typically a Stop Order would be placed first to prevent the broker from issuing the Limit Order to the market maker until the Stop price is reached, but the unique nature of smart contracts allows us to create seamless fungible products, and so we will call these orders simply "Limit Orders" so that the product can be perpetually referenced consistently regardless of its current internal state.



When T is between 0 and ℓ, as T approaches 0, t_l also approaches 0 since the chances of T reaching ℓ in the near future diminishes. As T approaches ℓ, t_l also approaches ℓ since the chances of T reaching ℓ and executing the limit order (and receiving the value of ℓ minus fees) increases. t_r follows the inverse of t_l valuation curve, since it is always the difference between ℓ and t_l.

3.4 Ongoing research

The $Token volatility constant (σ) is inadequate as a single constant. The best pricing formula will likely utilize the historical drift rate of the market price to determine the likelihood of reaching the limit price within certain time frames.

Additional research is needed to determine the proper allocation of LOTR assets to achieve a delta neutral portfolio. This allocation strategy will likely assume a liquid market of LOTR limit tokens priced at regular intervals with no fees and zero swap slippage.