Securd Protocol v1

DeFyLabs by PyratzLabs

Anthony HAYOT
anthony.hayot@pyratzlabs.com

Mohamed FRIHAT
moh@pyratzlabs.com

February 2023

Abstract

This White Paper introduces the decentralized protocol **Securd** and described every mechanism behind it.

Securd provides every crypto holder, from individuals to large company treasury, with a Saving Account where its digital assets generates passive income in a purely decentralized way. No intermediary, no third-party.

To generate revenues for Depositors, **Securd** automatically lend deposits to the most reliable borrowers in the market: Liquidity Providers. Their solvency is guaranteed by a digital proof of their repayment capacity : LP Tokens. By locking this proof into the **Securd** Protocol, they provide a much stronger protection for Depositors than any other lending protocol.

Contents

1 Introduction **3**
1.1 Objectives 3
1.2 Indirect Liquidity Providing 4
1.3 Leveraged Liquidity Providing 5

2 The Securd Protocol **6**
2.1 Collateral Model 6
2.2 Diversification Model 9
2.3 Interest Rate Model 10
2.4 Compounding Model 11
2.5 Liquidation Model 12
2.6 Fee Model 12

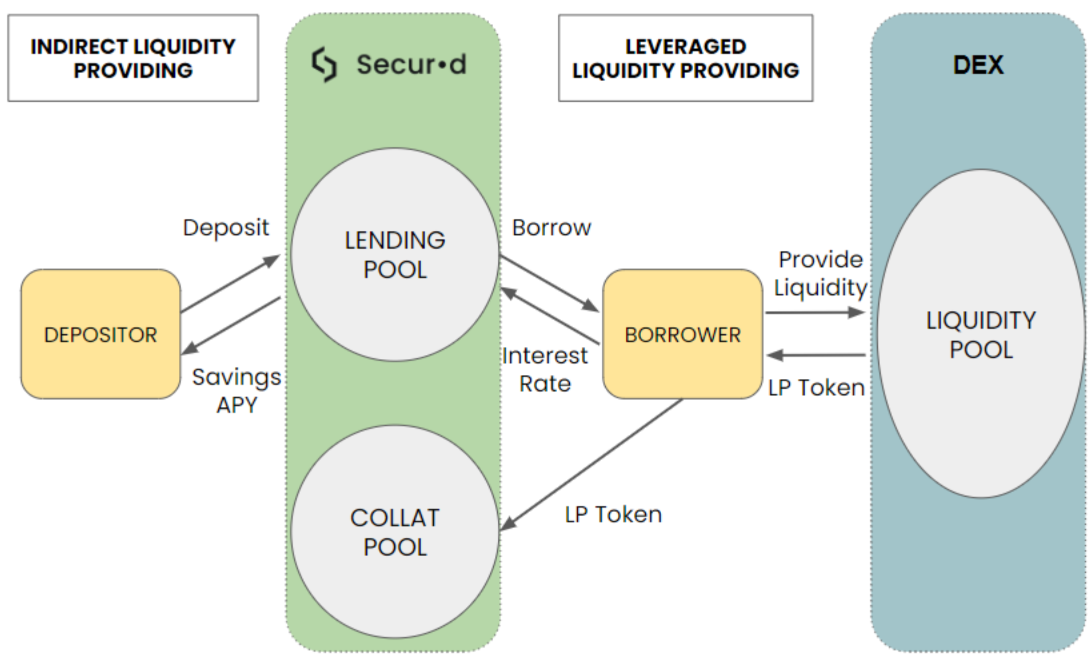
1 Introduction

1.1 Objectives

Providing liquidity to pool-based decentralized exchanges (DEX) is one of the main sources of yield in the DeFi space. However, this comes with a certain level of risk and complexity that may not suit every crypto holders such as managing impermanent loss, claiming rewards or participating in protocol governance.

Securd solve this issue by splitting the liquidity providing position into two strategies:

- **Indirect Liquidity Providing**, where Depositors receive passive income while transferring risk and complexity to experienced liquidity providers (Borrowers).
- **Leveraged Liquidity Providing**, where experienced liquidity providers can optimize their capital usage and multiply their returns by accessing extra funding at a reduced cost.



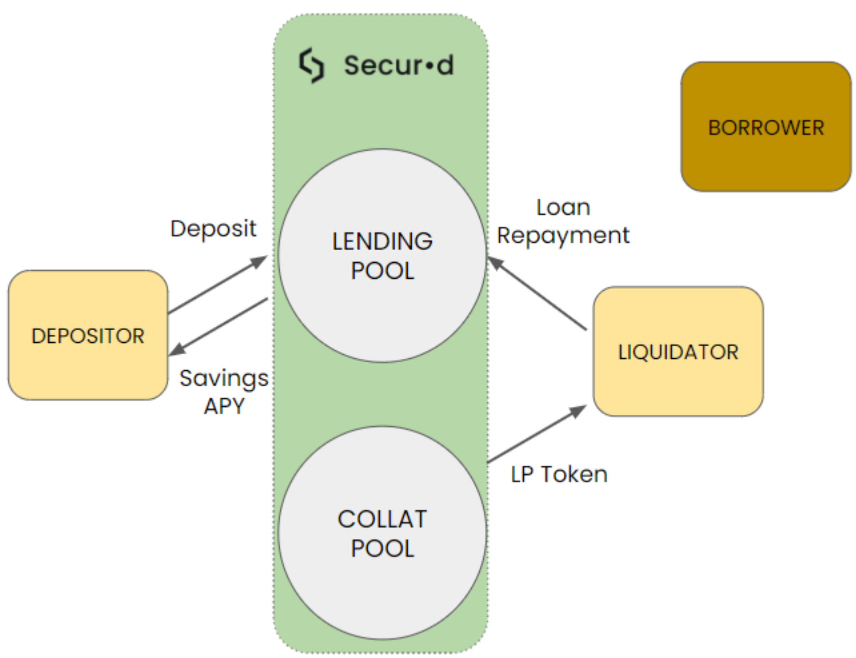
1.2 Indirect Liquidity Providing

Crypto holders looking to generate simple passive income can deposit crypto assets in their **Securd Savings Account** and become Indirect Liquidity Providers. These deposits, grouped by asset in **Lending Pools**, are used to grant loans to Liquidity Providers. The loan interest paid by Liquidity Providers (Borrowers) generates revenues in the Depositors' Savings Balance.

Like most lending protocols, **Securd** protects deposits from repayment defaults by requiring Borrowers to lock assets in the Collateral Pool. The key difference is that **Securd** uses a very stable collateral: LP Tokens. These LP Tokens represent the Borrower share of a DEX liquidity pool and can be redeem for a certain quantity of the original tokens.

In order to cover for repayment at any point in time, the ratio between the collateral value and the related loans value, called Collateral Factor or CF, should be maintained above a certain value, called Liquidation Threshold or LT (defined by the Collateral Model and always above 100%).

If the Collateral Factor reaches the Liquidation Threshold, **Securd** will seize the Borrower's collateral and sell it to a Liquidator. The sale proceeds will be used to pay back the related loans. This liquidation mechanism offers a strong capital protection to Depositors.



1.3 Leveraged Liquidity Providing

Liquidity Providers who are familiar with the risks and complexity of LP positions may be interested in extracting larger returns through leverage. By accessing borrowable funds, they are able to boost their return with the same amount of initial capital.

Let's consider that a Liquidity Provider owns 100 LP Token A/B, representing his initial deposit of 50 Tokens A and 50 Tokens B in a DEX Liquidity Pool. By locking his LP Tokens in the **Collateral Pool**, he is now able to borrow 40 Tokens A and 40 Tokens B. His **Collateral Factor** is:

$$CF = \frac{100}{80} = 125\%$$

If he deposits the borrowed tokens in the DEX Liquidity Pool, he will receive additional LP Token A/B. He can then increase his collateral position and borrow additional Tokens A and Tokens B while maintaining is CF at 125%. By iterating this operation, he will be able to multiply his initial position by a leverage of 5. In general, Leverage can be computed as follows :

$$Leverage = \frac{CF}{CF-1}$$

CF	110%	115%	120%	125%	130%	135%	140%	145%	150%
$Leverage$	11.0x	7.7x	6.0x	5.0x	4.3x	3.9x	3.5x	3.2x	3.0x

However, these iterations require a lot of manual transactions and are not efficient in terms of gas fees. **Securd** circumvents this by implementing an Automatic Leverage function.

By selecting a leverage of 5, our Liquidity Provider will, in a single transaction, increase his LP token position to 500 while contracting 2 loans of 200 Tokens A and 200 Tokens B. If the Liquidity Providing APY is 20% and the borrow rate is 5%, the Leveraged Liquidity Providing position is:

	Inital Position	Leveraged Position
LP Token	100	500
Token A	0	-200
Token B	0	-200
CF	∞	125%
Leverage	1x	5x
Total APY	20%	80%

Deleveraging works the opposite. When the Borrower uses the Deleverage function, collateralized LP Tokens will be redeemed from DEX for Token A and Token B. These tokens will be used to reduce the loan position.

If one of the loans is entirely repaid, the residual tokens will be swapped to repay the other loan. If both loans are repaid, the remaining tokens are sent to the Borrower.

2 The Securd Protocol

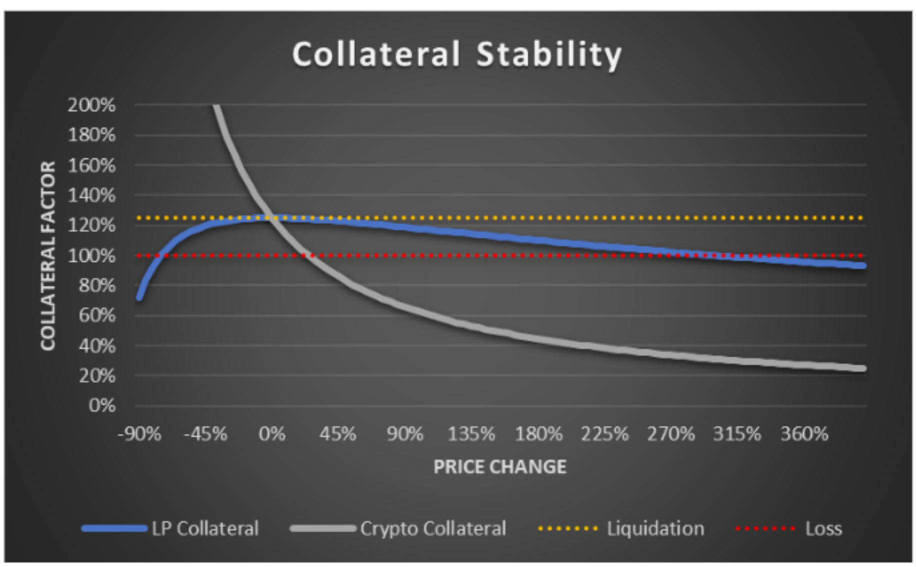
2.1 Collateral Model

Like most DeFi lending protocols, **Securd** protects Depositors from repayment default by requiring Borrowers to pledge assets against their loans.For this protection to be efficient :

- the Collateral Token and the Loan Token should be positively correlated
- the Liquidation Threshold should be defined such that when liquidation is triggered, the proceeds from collateral sale is enough to repay the corresponding loans.

For the first criteria, LP Tokens are natively correlated to their underlying tokens ensuring a stable Collateral Factor. Hence, the capital protection provided by Securd to Depositors is stronger than most lending protocols.The Graph below illustrates how the price fluctuation between Token A and Token B can impact the Collateral Factor in 2 cases:

- A loan 50% in Token A and 50% in Token B with LP Token A/B collateral
- A loan 100% in Token A with Token B collateral



We can observe that the "no loss" price range supported by the LP Token Collateral is much larger than the one supported by the Crypto Collateral.

Now let's focus on determining the appropriate Liquidation Threshold. In a Constant Product AMM, with P the price of 1 Token A in Token B unit, Q^A and Q^B the quantity of Token A and B in the liquidity pool, we have :

$$Q_0^A Q_0^B = K = Q_1^A Q_1^B \qquad \text{and} \qquad \begin{array}{l} P_0 = \frac{Q_0^B}{Q_0^A} \\ P_1 = \frac{Q_1^B}{Q_1^A} \end{array}$$

If the price P changes by a relative move of c, we can recompute the quantity of Token A and Token B in the liquidity pool:

$$P_1 = (1 + c)P_0$$

$$Q_1^A = \frac{Q_0^A}{\sqrt{1+c}} \qquad \text{and} \qquad Q_1^B = Q_0^B \sqrt{1 + c}$$

Case 1: Balanced Loan

Let's consider a loan, issued at t=0, of 1 Token A and P_0 Token B with a Collateral Factor CF_0. We have the following:

Loan	Number of Tokens	Value in Token B unit
Token A	1	P_0
Token B	P_0	P_0
Total		$2P_0$

Collateral	Number of Tokens	Value in Token B unit
Token A	CF_0	CF_0P_0
Token B	CF_0P_0	CF_0P_0
Total		$2CF_0P_0$

We can notice that this loan is perfectly balanced in value between Token A and Token B. With BR being the ratio between the Token A loan value and the total loan value, we have :

$$BR_0 = \frac{v_0^{loan\,A}}{v_0^{loan}} = 50\%$$

After a price change of c, we have:

Loan	Number of Tokens	Value in Token B unit
Token A	1	$(1+c)P_0$
Token B	P_0	P_0
Total		$(2 + c)P_0$

Collateral	Number of Tokens	Value in Token B unit
Token A	$\frac{CF_0}{\sqrt{1+c}}$	$CF_0P_0\sqrt{1 + c}$
Token B	$CF_0P_0\sqrt{1 + c}$	$CF_0P_0\sqrt{1 + c}$
Total		$2CF_0P_0\sqrt{1 + c}$

We can recompute the new Collateral Factor:

$$CF_1 = \frac{2CF_0\sqrt{1+c}}{2+c}$$

The Balanced Liquidation Threshold, or BLT, is the minimum Collateral Factor such that a balanced loan can resist a price change of c:

$$BLT = \frac{2+c}{2(1-LD)\sqrt{1+c}}$$

with LD the Liquidation Discount defined in section 2.5

Case 2: Unbalanced Loan
Let's now consider a loan of 2 Token A, 0 Token B and a Collateral Factor of CF_0:

Loan	Number of Tokens	Value in Token B unit
Token A	2	$2P_0$
Token B	0	0
Total		$2P_0$

Collateral	Number of Tokens	Value in Token B unit
Token A	CF_0	CF_0P_0
Token B	CF_0P_0	CF_0P_0
Total		$2CF_0P_0$

We have:

$$BR_0 = 100\%$$

After a price change of c:

Loan	Number of Tokens	Value in Token B unit
Token A	2	2(1+c)P_0
Token B	0	0
Total		$2(1+c)P_0$

Collateral	Number of Tokens	Value in Token B unit
Token A	$\frac{CF_0}{\sqrt{1+c}}$	$CF_0P_0\sqrt{1+c}$
Token B	$CF_0P_0\sqrt{1+c}$	$CF_0P_0\sqrt{1+c}$
Total		$2CF_0P_0\sqrt{1+c}$

We can recompute the new Collateral Factor and the corresponding Unbalanced Liquidation Threshold ULT:

$$CF_1 = \frac{CF_0}{\sqrt{1+c}}$$
$$ULT = \frac{\sqrt{1+c}}{1-LD}$$

The actual Liquidation Threshold is an average between BLT and ULT weighted by the portion of balanced and unbalanced loans:

$$LT_t = 2 * min(BR_t, 1 - BR_t)BLT + (1 - 2 * min(BR_t, 1 - BR_t))ULT$$

However, not all LP Tokens are equivalent. Depending on various criteria, such as the volatility of the token pair or the liquidation pool size, a LP Token value can be more or less volatile. To reflect this, we divide LP Tokens in 3 risk categories, from the safest AAA to the riskiest CCC and attribute a potential price jump c to each category. We can then compute BLT and ULT:

Category	c	BLT	ULT
AAA	200%	122%	182%
BBB	300%	132%	211%
CCC	400%	141%	235%

The figure below displays the Liquidation Threshold for each category as a function of the loan Balance Ratio:

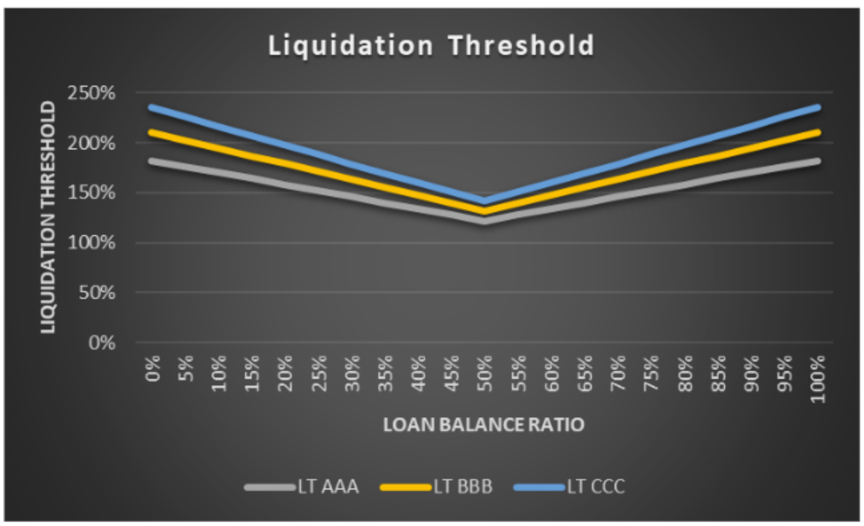


Fig1.Liquidation Threshold

Finally, in order to avoid immediate liquidation, every new loan must be collateralized with a minimum buffer over the Liquidation Threshold. We will set this minimum initial buffer, b:

$$b = 10\% LT$$

2.2 Diversification Model

In addition to overcollateralization, **Securd** offers an extra capital protection through collateral diversification.

If a Lending Pool is backed by a single LP Token and this LP Token value drops such that CF= 95%, the Lending Pool would suffer a loss of 5%.

With a diversified Collateral Pool (for example 5 different LP Tokens with 20% share each), a similar market move will only cause a 1% loss to the Lending Pool.

While overcollateralization reduces the probability of loss, diversification drastically reduces the amount of loss in the case it happens.

In this version, collateral diversification is only passive, by allowing multiple LP Tokens for each Lending Pool. In a following version, **Securd** will incentivize collateral diversification through interest rate mechanisms.

2.3 Interest Rate Model

Automating lending operations while maintaining a certain level of liquidity requires an efficient Interest Rate Model that balances supply and demand for each token around an optimal level of utilization.

Let's define the notion of Utilization Rate U. For a given Token A at time t,

$$U_t = \frac{B_t}{S_t}$$

where B_t is the total Borrow Balance of Token A
and S_t is the total Supply Balance of Token A.

- When U is low, it means that most of the deposits sits in reserve. It is easily accessible for immediate withdrawal but generates low interest for Depositors.
- When U is high, it means that most of the deposits has been lent. It generates high interest for Depositors but very little liquidity is available for immediate withdrawal.

We define U^* as the optimal level of utilization for Depositors. The objective is to find the optimal borrowing rate r^* for which $U = U^*$.

- If U settles below U^*, we can assume that the borrowing rate is not attractive enough for borrowers and needs to be decreased.
- If U settles above U^*, we can assume that the borrowing rate is too attractive for borrowers and needs to be increased.

Unfortunately, the function that links Utilization and Interest Rate is unknown and evolves over time. But we can assume that we have 2 fixed points:

- $U = 100\%$ when $r = r_{min}$
- $U = 0\%$ when $r = r_{max}$

By using these two fixed points, we can approximate, at each block, r* with the following equation:

$$r_{t+1} = r_t + s_t(U^* - U_t) \quad with \quad \begin{cases} s_t = \frac{r_t - r_{min}}{U_t - 100\%} & if \quad U_t < U* \\ s_t = \frac{r_t - r_{max}}{U_t} & if \quad U_t > U* \end{cases}$$

This innovative interest rate model allows a better convergence towards optimal utilization levels as well as a better adaptation to evolving supply and demand. A comparative research paper will be published soon.

2.4 Compounding Model

In order to provide maximum flexibility to both Depositors and Borrowers, interest payments are automatically integrated in Deposit and Borrow Balances and are compounded at every block.

To compute each Borrow Balance, we will use a specific Denominated Currency called lTokens. When a new Lending Pool is created for Token A, we will create a lToken, whose price in Token A unit will evolve as follows:

$$L_0^A = 1 \qquad \text{and} \qquad L_{t+dt}^A = L_t^A(1 + r_t dt)$$

where dt is the time period expressed in year.

By updating this price , each Borrow Balance B_t^i can easily be incremented by the interest accrued during the period. With LB_t^i the balance of lToken, we have:

$$B_t^i = LB_t^i L_t^A$$

For Depositors, we will follow a similar approach by introducing a Deposit Token, or dToken, that represents their ownership in a specific Lending Pool. As opposed to lToken, these dToken will be minted and transferred to Depositors in exchange for their deposits. Upon withdrawal, they will be sent back to Securd and burnt.

In order to materialize the accruing interest over deposits, the price of dToken, D_t^A, will be incremented at the Deposit Rate R_t:

$$D_0^A = 1 \qquad \text{and} \qquad D_{t+dt}^A = D_t^A(1 + R_t dt)$$

$$R_t = (1 - F^{Interest})U_t r_t$$

with $F^{Interest}$ the Interest Fee defined in section 2.6.

Borrow Balance and Supply Balance will be updated at every block generating a compounding effect. With N the number of block per year, Borrow APY and Savings APY can be computed:

$$BorrowAPY = (1 + \frac{r_t}{N})^N \; SavingsAPY = (1 + \frac{R_t}{N})^N$$

2.5 Liquidation Model

In the event where, on a specific Collateral Account, the Collateral Factor reaches the Liquidation Threshold, a liquidation process takes place in order to protect Lending Pools from losses. With LD the Liquidation Discount on the collateral price, two cases can arise:

- If $v^{coll}(1 - LD) > v^{loan}$, **Securd** determines the fraction of collateral to deliver to the Liquidator against repaying both loans. This fraction LR can be computed:

$$LR = \frac{v^{loan}}{(1-LD)v^{col}}$$

- If $v^{coll}(1 - LD) < v^{loan}$, the resulting loss on the loan j will be reflected in the Deposit Rate of both Lending Pools:

$$R_t = \frac{(1-F^{Interest})\sum_{i \neq j} B_t^i r_t dt - BR_t(v^{loan} - (1-LD)v^{coll})}{S_t dt} \text{ for Token A}$$
$$R_t = \frac{(1-F^{Interest})\sum_{i \neq j} B_t^i r_t dt - (1-BR_t)(v^{loan} - (1-LD)v^{coll})}{S_t dt} \text{ for Token B}$$

2.6 Fee Model

A Protocol Reserve will be built over time by applying two types of fees:

- A Borrowing Fee, F^{Borrow}, of 0.1%, applied when a new loan is contracted as a fixed percentage of the Loan Amount
- An Interest Fee, $F^{Interest}$, of 10%, applied on collected interest from Borrowers

We will soon publish our Tokenomics Principles that will describe in detail how this Protocol Reserve will be used.