Working Draft



A Non-Rent- Seeking Open Protocol for Decentralized Margin Trading with ERC20 Tokens on the Ethereum Blockchain

Vignesh Sundaresan,
Vignesh Meenakshisundaram,
Anand Venkateswaran

Lendroid.com | October 04, 2017

LENDROID
DECENTRALIZED MARGIN TRADING

Archived copy · cryp2nova.com

ABSTRACT

In this whitepaper, we elucidate a protocol on the Ethereum blockchain that makes an attempt to bring together lending, leveraged margin trading and short selling onto a single protocol – decentralized, trust-independent, non-rent-seeking – which seeks to overcome 'on-chain' limitations of inadequate computational power, latency and impractical gas cost by creating a symbiotic off-chain infrastructure supported by incentivized participants. The native token keeps the network operational, fuels the utility layer of the protocol, and offers security to community governance. An important goal of the protocol is to nurture a decentralized, global, shared lending pool with far-reaching applications beyond just the use case of margin trading. The modular and extensible nature of the protocol could make it the basis for all lending-related use cases for an increasing number of assets that are being tokenized across industries. The protocol's inherent non-partisanship, where no single group of stakeholders draws arbitrary benefit, is inspired from 0x.

CRYP2NOVA

Archived copy · cryp2nova.com

TABLE OF CONTENTS

1. INTRODUCTION

1.1. LENDROID'S LENDING ORIGINS

1.2. The 0x shift

1.3. An ethical parenthesis

1.4. Lendroid – An overview

1.5. How to use your Oracle

2. LENDROID FROM FOUR POINTS OF VIEW

2.1. THE LENDER

2.1.1. Defining the offer object

2.1.2. Cancelling a loan offer

2.2. THE RELAYER

2.2.1. Interface

2.2.2. Bookkeeping

2.2.3.` A shared, global liquidity pool

2.3. THE MARGIN TRADER

2.3.1. Locking in collateral

2.3.2. Availing loan

2.3.3. Managing trade positions

2.3.4. Closing a loan

2.4. THE WRANGLER

2.4.1. Margin level monitoring

2.4.2. 'Wrangling' underperforming margin account

2.4.3. Triggering an auction

3. THE LENDROID AUCTION

3.1. AN EVEN PLATFORM

3.2. AUCTION – AS IT HAPPENS

4. LENDROID SMART CONTRACT SYSTEM

4.1. UPGRADABLE, MODULAR AND EXTENSIBLE

5. THE LENDROID SUPPORT TOKEN (LST)

5.1. UTILITY TOLL

5.2. INCENTIVIZED PARTICIPATION

5.3. ENABLING GOVERNANCE

6. THE LENDROID VISION

7. SYNOPSIS

8. APPENDIX

9. ACKNOWLEDGMENTS

CRYP2NOVA

Archived copy · cryp2nova.com

1. INTRODUCTION

1.1. LENDROID'S LENDING ORIGINS

The lender is the primary participant on the Lendroid platform. The protocol's soundness is validated only by his sustained, profitable participation. The lender's trust in the trust-independent protocol will power the creation of a global lending pool. To this end, innovative rules and fail safes have been put in place to not let down his basic expectations – of protecting his capital and earning risk-free interest in a low-friction manner [1][2].

In point of fact, Lendroid was initially intended as a decentralized lending platform. Broadly, the process was designed thus: A borrower pledges digital assets into an escrow account, and the lender defines specific terms for the loan – amount, interest rate, loan period, loan to value (LTV) – set in a smart contract. If the lender's and borrower's terms match, the smart contract is executed, locking in the collateral and releasing the funds to the borrower. If the borrower satisfies the loan obligations, his collateral is unlocked automatically. If the borrower defaults or the collateral drops below the agreed LTV, the collateral is liquidated [3].

From here on, margin trading seemed a logical first-use case for Lendroid for two main reasons

a. A Rapidly Growing Market – Margin funding has soared, as recorded by some widely used exchanges like Poloniex and Bitfinex. On the latter exchange alone, USD funding has increased from 14.8 Million to 168.5 million (Oct 2016 to Oct 2017); BTC loans have gone from USD 6.12 Million to USD 183 Million (Oct 2016 to Sep 2017); and ETH loans have risen from USD 834,192 to USD 88.29 Million (Oct 2016 to Sep 2017) [4].

b. Need for Trust Independence is Immediate – Centralized exchanges that enable margin trading are left with the onerous, dual responsibility of funds and collateral management as well as keeping the traders from defaulting [5][8]. In order to provide this service, it might be argued, centralized exchanges have bypassed the inherent strengths of the blockchain ecosystem **[See Figure 1]** – premised on decentralization, on absolute protection and anonymity by global consensus.

Archived copy · cryp2nova.com

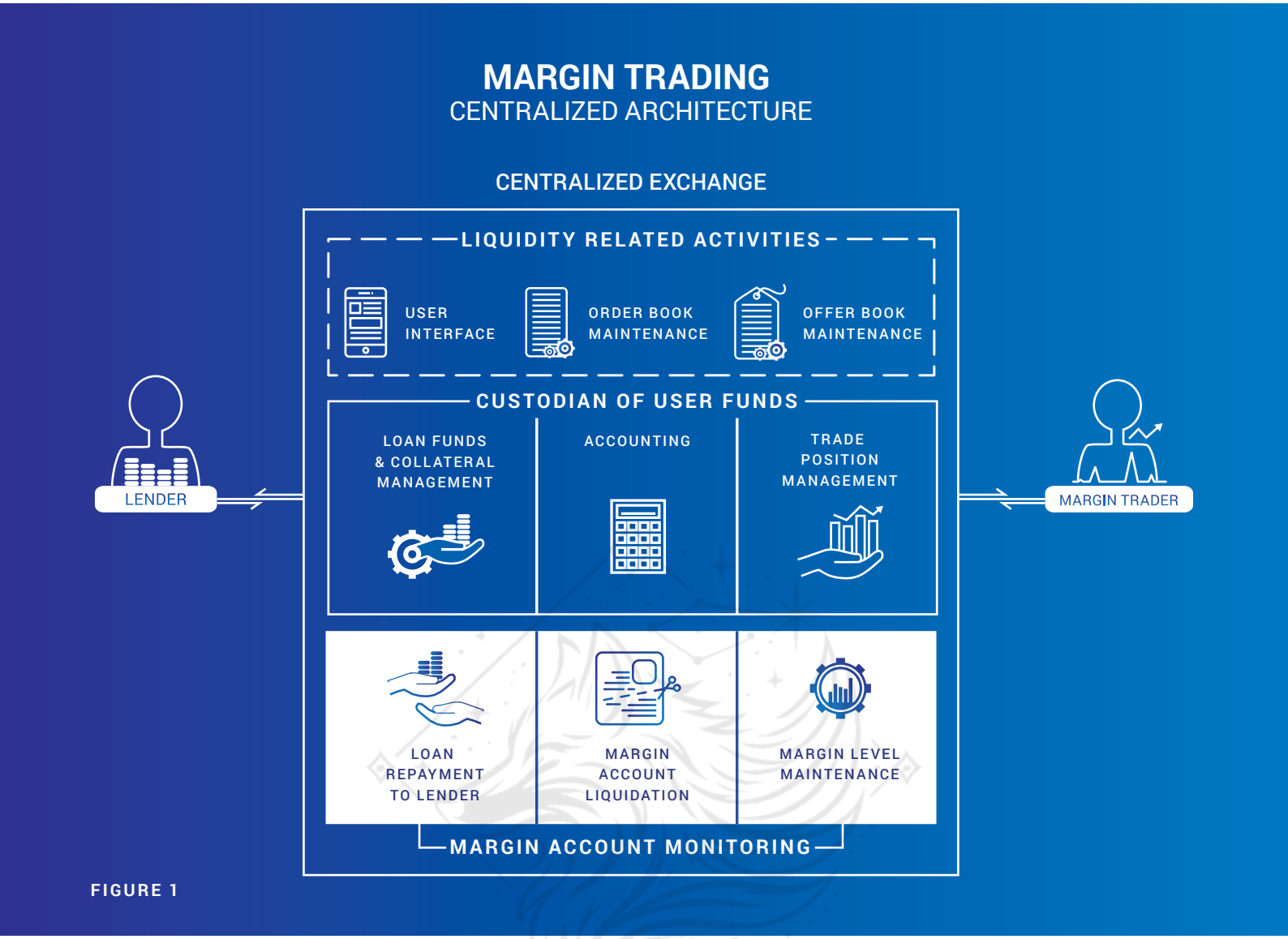


FIGURE 1

A decentralized margin trading protocol - a platform that is censorship resistant and can be accessed globally would help move away from central custodians holding user funds. Further, a global network of lenders will increase liquidity and reduce overall margin trading costs

1.2. THE 0X SHIFT

Consider the three main functions provided by current public exchanges – Exchange (regular trading), Margin Trading, and Lending. Ever since 0x – a protocol for peer-to-peer exchange of ERC20 tokens [6] – rolled out, a wholly decentralized ecosystem comprising all three functions has begun to seem highly plausible. The catalysts for this development are the 'Relayers', incentivized entities that facilitate the hosting of order books on 0x. The orders themselves are settled on the Ethereum blockchain using smart contracts.

The positive reception that has greeted 0x is the most recent illustration of the shift towards a wholly trust-independent, secure and decentralized exchange. The implications are tremendous, and new 0x relayers such as such as Ethfinex, Radar and The Ocean have already begun to enter the fray.

However, where 0x has enabled the Exchange to function seamlessly, the Lending and Margin Trading functions remain outside of the ecosystem. These are the missing pieces that Lendroid brings together to create a comprehensive solution.

Archived copy · cryp2nova.com

1.3. AN ETHICAL PARENTHESIS

Ethical considerations seldom find a place in discussions about the structure of new cryptoeconomic models. In the case of Lendroid, however, they assume unique context.

This protocol aims to take forward a culture of fairness inherent in the blockchain ecosystem. Lendroid sets out to achieve total decentralization and true trust-independence at the most basic, reliable levels. Lendroid is not agnostic or indifferent to trust, which is after all, a positive phenomenon. It simply recognizes and leverages a paradox – trust abounds only in an environment that does not depend on it. To paraphrase a celebrated Google adage, on Lendroid the hope is that one 'Can't Be Evil'.

Further on in this whitepaper, you will see that this end state is sought with a two-pronged approach –

a. Inviolability: Vital, sensitive, conventionally trust-dependent functions like fund management are executed on-chain and via publicly accessible, inviolable Ethereum smart contracts.

b. Symbiosis: Wherein the off-chain players are incentivized to carry out their functions and whose propensity for profit and growth is proportionate to that of the other players in the ecosystem.

Archived copy · cryp2nova.com

MARGIN TRADING

DECENTRALIZED ARCHITECTURE

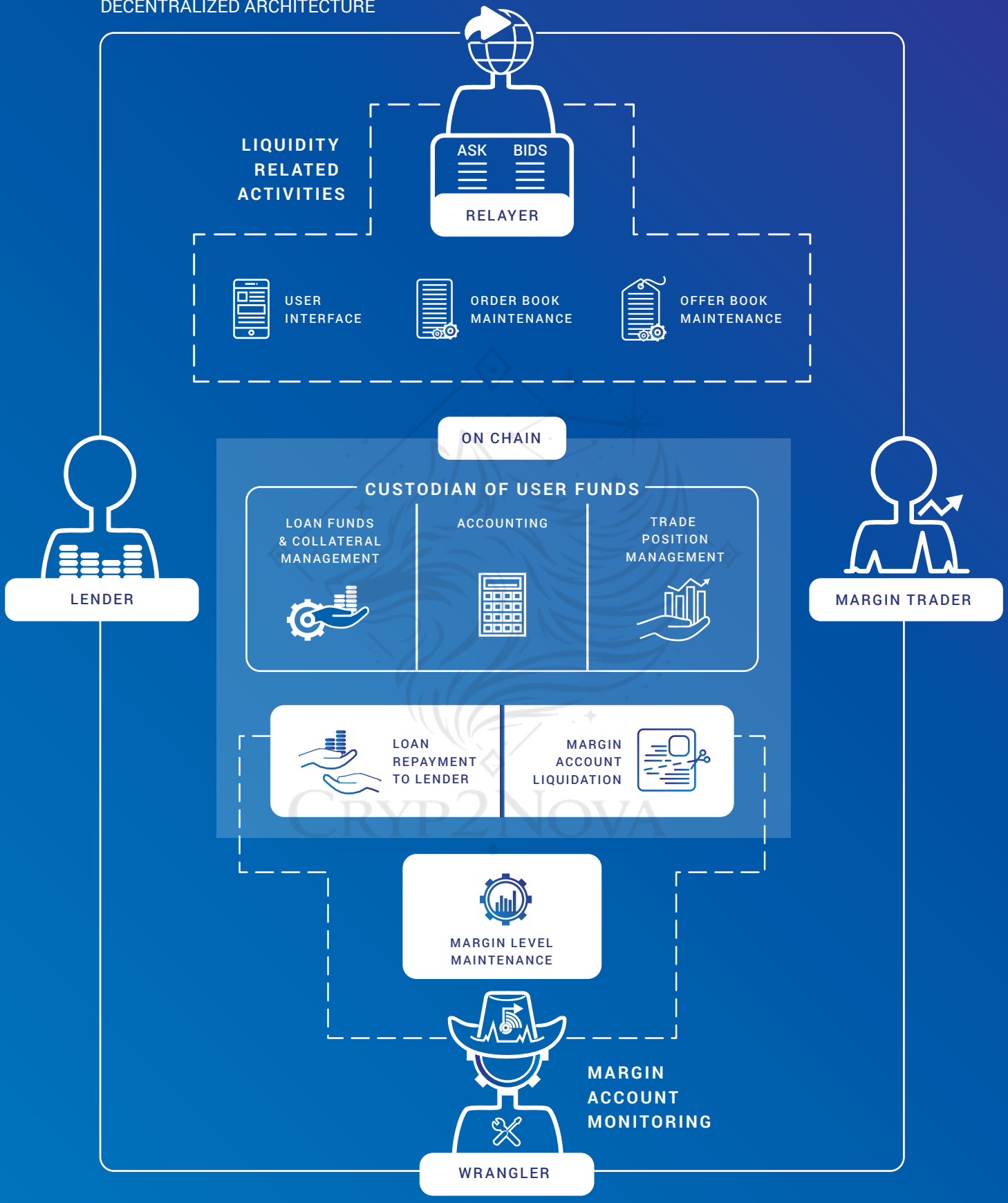


FIGURE 2

Archived copy - cryp2nova.com

1.4. LENDROID – AN OVERVIEW

A margin trader starts by depositing collateral. The trader is allowed to borrow up to a certain leverage to the point where the margin level is greater that the initial level (set through community governance).The borrowed funds come from the shared lending pool to which the lenders can submit offers. With the borrowed funds, the trader interacts with order-books to calibrate positions to reflect expectations from future price movements. The positions, while active, are held in the smart contract. Neither the collateral nor the tokens purchased using the borrowed funds can leave the margin account until the loans obligations are cleared by the margin trader. Once the traders feel it's time to unwind their position, the trader once again interacts with the order book. If the trader made a profit, he can repay the lenders and withdraw the profit along with his collateral. If a loss is incurred, the trader is expected to deposit the difference to compensate for the losses, repay the loans in full and only then withdraw collateral.

However, the above mentioned is only a best-case scenario; other less favorable scenarios can also occur. What happens if the asset price in the open positions or the value of the collateral starts to drop?

In a central exchange, such a situation would be manually solved by a margin call that the exchange would make to the trader, asking them to rebalance his margin account. In the decentralized context, this is handled by off-chaining such operations to a utility layer comprised of self-interested actors working to maintain stability within the network. One of the actors – Relayers – are based on a similar concept propagated by the 0x protocol [6]. The other – Wranglers – are an entirely original entity propounded by Lendroid. While a comprehensive description is provided in subsequent sections, a Wrangler's role might be defined as one related to margin account monitoring. The Wrangler aids in margin-level maintenance, calls out margin accounts at liquidation levels (for which he is incentivised with a 'bounty') and, on liquidation, repays the Margin Trader's loan to the Lender.

We propose a network consisting of two types of actions: off-chain actions that propagate an intent and on-chain actions that confirm the intent, thereby committing the state of the network on the Ethereum blockchain.

Archived copy · cryp2nova.com

1.5. HOW TO USE YOUR ORACLE

Use cases in the past have demonstrated the unviability of an Oracle-based approach on the Ethereum blockchain [7]. The drawbacks in terms of tag, efficacy of information and the fact that an Oracle is a centralizing component make such an approach cumbersome and less than completely reliable. Despite this, Oracles currently remain the simplest means of bringing external data on-chain.

On Lendroid, Oracles serve as a point of reference, and as a preventive against unwarranted triggers from Wranglers.

An Oracle function is commissioned by the smart contract to independently verify what a Wrangler identifies as a Margin Account at liquidation level. The gas expense from this verification is borne by the Wrangler. However, the data provided by Oracle, even when it corroborates the Wrangler's claim, does not result in liquidation of the account. It triggers a process unique to Lendroid.

No information provided by the Oracle forms the basis for sensitive decisions that involve fund management or account liquidation. The fate of the margin account is determined by a process that involves multiple actors, and with opposing interests. Such decisions are the result of a unique auction process, which will be elaborated in a separate chapter in this whitepaper. The crux is that this process is open to and involves wranglers other than just the one who identified the terminal account. It also involves the margin trader. These two sets of actors, with opposite intents, change positions, bid and quote towards something that is most in their economical interests, based on the external information they bring independently.

Collusion is thus ruled out, the situation is no longer centrally driven and, most importantly, whatever external information the actors subscribe to is translated into an actual transaction when they bring it on-chain.

2. LENDROID FROM FOUR USER JOURNEYS

A more comprehensive and effective understanding of the protocol can be achieved when approached from the user journeys of four key actors.

2.1. THE LENDER

Analogous to the Maker from the 0x protocol [6], the lender broadcasts the loan-offer to all decentralized exchanges (off-chain action).

Of the different types of accounts within the Lendroid Smart Contract system, the Funding Account is available to the lenders to deposit funds into and offer loans from.

Each loan offer is a data packet called the offer object containing the terms of the loan, offer parameters, and an associated ECDSA signature. The loan terms and parameters are concatenated and hashed to produce a 32 byte-long Keccak SHA3 signature called the offer-hash. The lender signs this offer-hash with their private key to produce the ECDSA signature.

As discussed in section 1, it is imperative that the interests of the Lender - who contributes to the shared liquidity pool - are protected. This is enabled by empowering the Lender to define key parameters within the offer object.

CRYP2NOVA

Archived copy · cryp2nova.com

2.1.1. Defining the Offer Object

Off chain, the Lender defines the offer object with a range of parameters [See Table 1], key among them

a. Loan Amount and Interest Rate: The Lender defines the offer amount (token offered and number of units of the token) and sets the interest to be calculated on a daily basis.

b. Loan Expiration Period: The validity of the loan from the time it is availed. On expiry, the loan is called in and, should the borrower so choose, rolled over with another loan.

c. Offer Expiration: The validity of the offer itself. Based on the demand for a token, a Lender might want to change his offer to reflect a different interest rate or number of token units. Setting a prudent expiry time for the offer allows for this flexibility, without having go on-chain.

Name	Datatype	Description
loanContract	address	Address of the Smart Contract that handles availing and canceling of loan offers
lender	address	Address offering the loan
borrower	address	(Optional - For point-to-point offers) Address availing the loan
loanToken	address	The token offered as loan
maxLoanExpiration	uint256	Maximum time up to which the loan is active since the offer was accepted (Time in seconds since unix epoch)
offerAmount	uint256	Total units of loan token offered by lender
interestRate	uint256	Rate of interest calculated on a daily basis
offerExpiration	uint256	Time at which the offer expires (Time in seconds since unix epoch)
salt	uint256	Arbitrary number that provides uniqueness for the offer's Keccak SHA3 hash
relayer	address	Exchange address via which the offer is broadcast
relayerFeeLender	uint256	Highest amount of LST units paid to a relayer by the lender for closing a loan. If the loan is closed prematurely, the relayer is paid pro-rata.
relayerFeeBorrower	uint256	Highest amount of LST units paid to a vigilante by the borrower for closing a loan. If the loan is closed prematurely, the relayer is paid pro-rata.
wranglerFeeLender	uint256	Highest amount of LST units paid to a wrangler by the lender for closing a loan. If the loan is closed prematurely, the wrangler is paid pro-rata.
wranglerFeeBorrower	uint256	Highest amount of LST units paid to a wrangler by the borrower for closing a loan. If the loan is closed prematurely, the wrangler is paid pro-rata.
v	unit8	ECDSA signature of the above arguments
r	bytes 32	ECDSA signature of the above arguments
s	bytes32	ECDSA signature of the above arguments

TABLE 1

Though not part of the offer object as currently envisioned, automatic renewals of loans too can be explored.

Archived copy · cryp2nova.com

It is important to note that every loan also requires some LST for initialization. To lubricate the inherent processes, the lender deposits some LST meant for the Relayer at the end of the loan period. Some he deposits for the Wrangler in the event that it was he who identified the Margin Account at liquidation level and helped close the loan. The LSTs required for these processes are locked in at the time of initiation. If there is not enough LST for these processes, the loan is not initialized.

2.1.2. Cancelling the Loan Offer

A loan offer is deemed invalid, and therefore is cancelled, in the following cases:

a. If a loan offer is availed after its expiration period: In this case, the Loan Smart Contract recognizes that the offer cannot be availed anymore, and triggers an event indicating to all Relayers that the loan with the corresponding hash is invalid. It is in the Relayers' best interests to not display expired loans in the first place and, even if they do, they have the option to listen in on the LoanOfferCancelled event from the Loan Smart Contract.

b. If a Lender decides to cancel an offer that has been left unavailed and unfilled: In this case, the lender can call the Loan Smart contract's CancelLoan function which further triggers the LoanOfferCancelled event from the same Smart Contract, thus notifying all listening Relayers. Cancelling a loan offer is a fallback mechanism and costs gas. Therefore, it is in the best interests of the lender to set a suitable ExpirationPeriod on the offer to avoid on-chain transactions.

2.2 THE RELAYER

The Relayer is the first point of contact for both lender and trader on the Lendroid protocol. Off chain, this entity effectively takes up the liquidity related functions of an exchange in a wholly decentralized manner. The one significant addition to the Lendroid Relayer over its 0x counterpart is the management of loan offers. Relayers provide an interface, and manage order books and offer books. They are key to creating the shared global lending pool.

2.2.1. Interface

As actors that operate almost entirely off-chain, the Relayer's potential for creating a smooth, fast, user experience, is limitless. Available interfaces today offer an increasingly sophisticated dashboard, with periodic upgrades in granularity, of data native to the protocol, as well as relevant external information from the market, to help lenders and margin traders make more informed decisions.

Loan offers are captured and offered as part of this interface. The richness of the information on display depends on the depth that a particular Relayer chooses to showcase. Without exception, every Relayer has complete access to the shared lending pool – the collective of loan offers recorded with not just one particular Relayer, but with the entire community of Relayers on the protocol.

Relayers can also offer APIs to facilitate programmatic access to offer books.

With access to the entire lending pool, and with off-chain speed and flexibility at their disposal, Relayers can maintain liquidity and create an interface that encourages patronage.

Archived copy · cryp2nova.com

2.2.2. Bookkeeping

The same off-chain advantage in interface extends to the bookkeeping function of the Relayer as well. With computational power unbridled by on-chain lags and costs in gas, a Relayer can maintain Offer Books in a real-time, seamless manner.

The Relayer is incentivized with a fee in the Lendroid Support Token (LST) for recording and relaying the loan offer. A fee is paid by the Lender and, when the loan is availed, by the borrower/margin trader as well. Attracting liquidity into the platform, the Relayer is paid only when a loan is availed, and closed properly. The Relayer is able to match and offer with an ask even if it originated with another Relayer. Despite the incentive, and the fact that the Relayer has access to the entire shared lending pool, the protocol remains trust independent because of two key 0x-inspired characteristics [6], written into the protocol itself.

a. Relayers cannot execute transactions on behalf of the Lender or the Margin Trader. They can only recommend an ideal offer.

b. A clear fee structure when a loan is availed. The Relayer with whom the offer was recorded is paid by the Lender, and the same Relayer is paid by the Margin Trader once the loan is paid back.

2.2.3. A Global, Shared Liquidity Pool

Due to the inherent trust-independence of the protocol, access to loan offers is not limited to any one Relayer or a particular set of Relayers on Lendroid. As an extension, all the lenders and the Margin Traders on the platform contribute to liquidity in the ecosystem.

As Figure 3 illustrates, a lender is free to engage with any Relayer on the protocol. Irrespective of who they record the loan offer with, it goes into the shared lending pool. When catering to the Margin Trader, a Relayer is free to recommend loan offers received from any Relayer. In the interest of providing a 'sticky' experience that could make him the Primary Relayer, it is in his best interest to connect the Lender or Margin Trader to the most ideal offer.

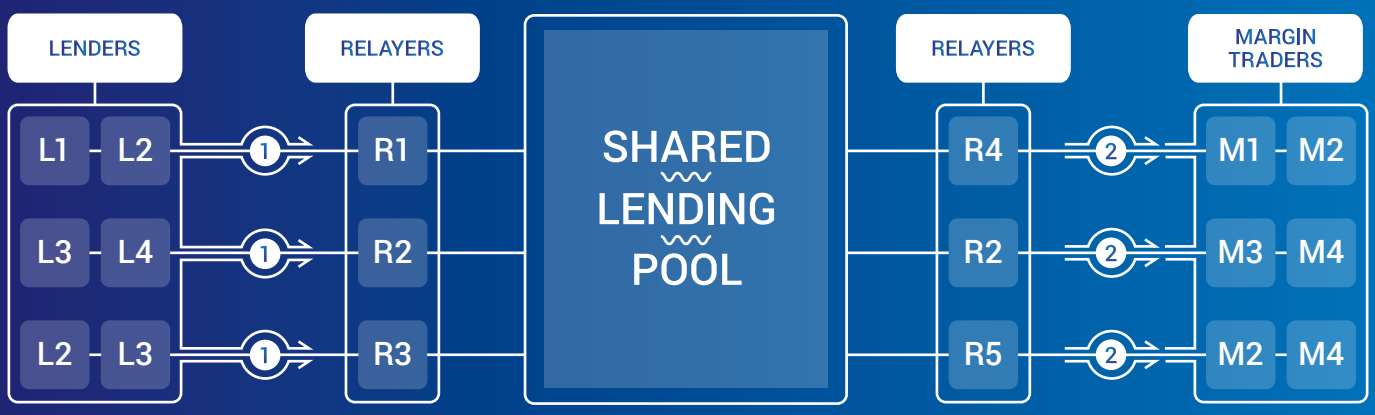


FIGURE 3

Archived copy · cryp2nova.com

2.3 THE MARGIN TRADER

On Lendroid, the Margin Trader enjoys leveraged lending in a uniquely decentralized environment. He deposits collateral, avails the loan offered by lenders and engages in margin trading/short selling. He opens a margin account, within which he is free to change positions or add collateral. He can withdraw his collateral or liquidate his Margin Account when there are no loans owed. Here's how those actions play out, broadly.

2.3.1. Locking in Collateral

Lendroid is compatible with any ERC20-based token. This gives the Margin Trader a high level of flexibility with respect to the nature of collateral he can deposit against a loan. A Margin Trader is free to deposit not just one kind of collateral in a token of his choice, but multiple kinds of ERC20-based tokens at the same time, in the same margin account. However, the support for a token has other considerations besides compatibility alone.

In what is envisioned as a community-driven governance model, the protocol will add or remove support for a token based on the current volatility of the token.

If the Margin Trader finds that the health of the margin account has dipped, he can add collateral at any time before the liquidation process is initiated on his account.

2.3.2. Availing a Loan

As illustrated in section 2.1, a Margin Trader can avail a loan through his Relayer. It is in the interfacer provider's interest to connect him to the ideal offer.

Offers are signed by the respective Lenders who made them are available with the Relayer. When a Margin Trader finds one that is suitable for his needs, he calls the availLoan function on the Loan Smart Contract, by including the offer object in the function call.

Using the ecrecover function, the smart contract will then verify that the signature is valid and, based on the timestamp, that the offer has not expired. The Smart Contract also checks the Lender's funding account to make sure there are sufficient funds to carry out the offer. Once ratified, the funds are credited to the Trader's Margin Account and debited from the Lender's Funding Account.

Archived copy - cryp2nova.com

23.3. The Margin Account

The Margin Trader stakes LST when opening a Margin Account. This stake is refundable, when the Trader closes trade positions and repays the loan. However, in case the account drops below the liquidation level, the deposit becomes a bounty for the Wranglers.

In the Margin Account, neither the collateral nor the positions can be withdrawn without honouring the terms of the loan. However, a Margin Trader is free to manipulate both elements at any time to maximize returns. He can top up collateral in a flagging account, and calibrate his positions with the help of decentralized order book protocols.

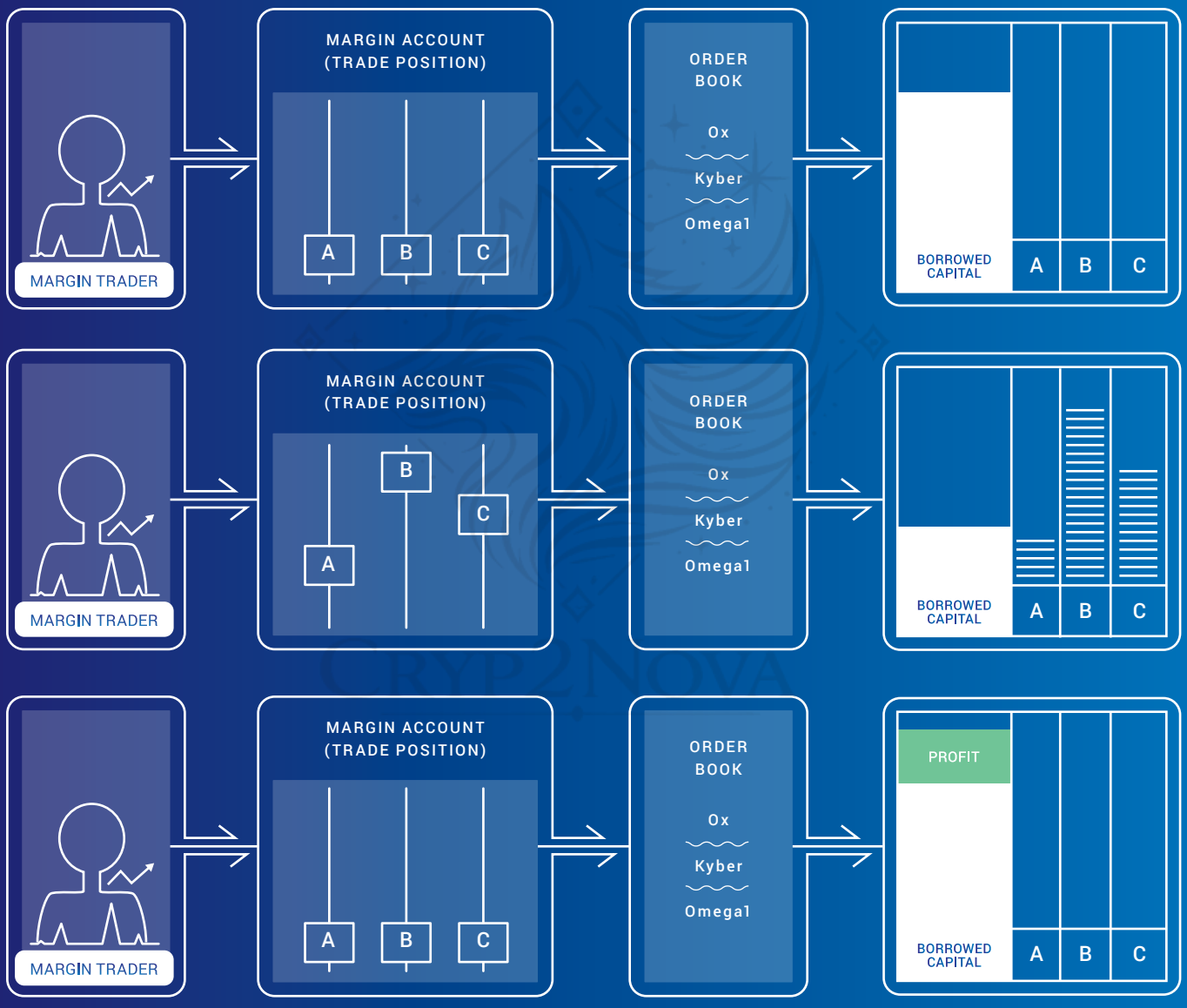


FIGURE 4

Figure 4 illustrates part of the user journey of a Margin Trader. His relationship with the Margin Account, dynamic and protected at the same time, can be understood in three states.

Archived copy · cryp2nova.com

In **Figure 4a,** there's the Margin Account on the extreme right, which contains the capital he borrowed for the trade, and the possibility of creating trade positions out of available funds.

In **Figure 4b,** we see that the Margin Trader has calibrated his trade positions. This intent is recorded on-chain, and executed using a decentralized order book protocol. Lendroid is capable of integrating with multiple order book protocols, including 0x, Kyber, Omega, and others. When positions need to be updated or rearranged, the Margin Trader can delegate the function to the most liquid order book protocol in the ecosystem.

In **Figure 4c,** we see that at the end of a trading session, the trade positions have done well for the Margin Trader and a profit has been accrued over and above the borrowed capital.

2.3.4. Closing a Loan

The loan terms are specified by the Lender right at the beginning, when the offer is made. There are three scenarios in which the loan is closed.

The first is the scenario described in **Figure 4c.** The Margin Trader has made a successful trade and is able to close the loan, repaying it without incident. Alternatively, the term of one or many loans might be close to expiry even when the Margin Trader is not ready to unwind his trade positions. Monitoring this can be delegated to incentivized actors called Wranglers, who will help replace or 'roll-over' an expiring loan with a fresh one. Though not built?into the protocol, this specific function is a logical extension of a Wrangler's primary role – that of monitoring the margin account.

In the third scenario, a Margin Account underperforms and warrants immediate action. A Wrangler has identified the account. The Wrangler then repays the loan to the Lender himself, and also triggers an auction of the Margin Trader's collateral and positions.

A fourth and less likely scenario is a Black Swan event, where a Margin Account has been left unattended by Wrangler and Margin Trader and, for enough time to cause damage, by the Lender himself. In such a case, the Lender can call in his loans and as a result will take over and choose to withdraw the collateral and trade positions as a compensation.

CRYP2NOVA

2.4 THE WRANGLER

The Wrangler is an entity conceptualized for the Lendroid ecosystem. One of two incentivized actors within the protocol, the Wrangler is intended to perform a computationally intensive role: that of monitoring the Margin Accounts. By monitoring and 'Wrangling' terminal accounts (at liquidation level), the Wrangler maintains the general health of the ecosystem, while also protecting the interests of the Lenders. However, the applications of such processing power spill beyond the requirements of the protocol, and can begin to enhance the experience of the other participants.

2.4.1. Margin Level Monitoring

Every on-chain event is broadcast on the protocol. It is this broadcast that the Wrangler tunes into. The Wrangler keeps tabs on each development: from the creation of a Margin Account, to every state change in position, adding of collateral, etc until the account is liquidated.

Archived copy · cryp2nova.com

Whenever there is a change in the state of a Margin Account as a result of calling a function from the Lendroid Smart Contract system, the Account Smart Contract triggers an event notifying and specifying the function name and the corresponding Margin Account.

Name	Datatype	Description
marginAccount	address	Address of themargin account whose state has changed
functionName	string	The function call that resulted in the state change.

TABLE 2

The following pseudocode highlights the set of off-chain and on-chain computations performed by the Wrangler while monitoring Margin Accounts. Figure 5 illustrates the process.:

• defaultingAccount = None;
• while defaultingAccount is None:
- (on-chain) pick a random position pr from the list of open positions;
- (on-chain) obtain the units from pr[collateralUnits], pr[totalBorrowedUnits], and pr[positionUnits] by calling the getPositionDetails(pr) function from the MarginTrade Smart Contract;
- (off-chain) query the market values for these units;
- (off-chain) calculate collateral value;
- (off-chain) calculate total borrowed value;
- (off-chain) calculate position value;
- (off-chain) calculate the resulting profit / loss;
- (on-chain) obtain the loan terms by calling getLoanOfferDetails(pr[offerHash]) function from the Loan Smart Contract;
- (off-chain) calculate interest accrued based on the loan terms;
- (off-chain) calculate the margin level from formula f1;
- If the margin level falls below 20%:
- defaultingAccount = pr[account];
• if defaultingAccount is not None:
- (on-chain) call triggerAuction(defaultingAccount) from the Auction Smart Contract

For accurate and contextual computation, the Wrangler needs both on-chain and off-chain data.

It is important to note that margin monitoring, recognized as an aspect of risk management, was hitherto performed by a centralized exchange to protect the Lender's interests and the market integrity. Conventional exchanges would make what is known as a 'margin call' – an actual phone call to the trader, asking him to deposit additional funds – when the account dipped below a certain percentage of leverage [8]. Since the concept of a margin call is improbable on the blockchain ecosystem, the Wrangler's role becomes all the more vital for the health of the market. This is why the Wrangler's role, powered by computational capability, is incentivized in innovative ways. For instance, as seen in 2.3.4., a Wrangler can be delegated – for a fee by the Lender – to not just monitor the Account, but also help with rolling over an expiring loan with a fresh one.

The Wrangler picks up a bounty when he 'wrangles' an Account at liquidation level. He also stands to earn the right to the Margin Trader's collateral or his trade positions, when he triggers an Auction.

Archived copy · cryp2nova.com

MARGIN ACCOUNT MONITORING

1. Actively listening to state changes in any Margin Account
2. Recursive querying of market data
3. Reporting a defaulting Margin Account & initiating liquidation
4. Verifying the report
5. Liquidating the Margin Account & Rewarding the Vigilante

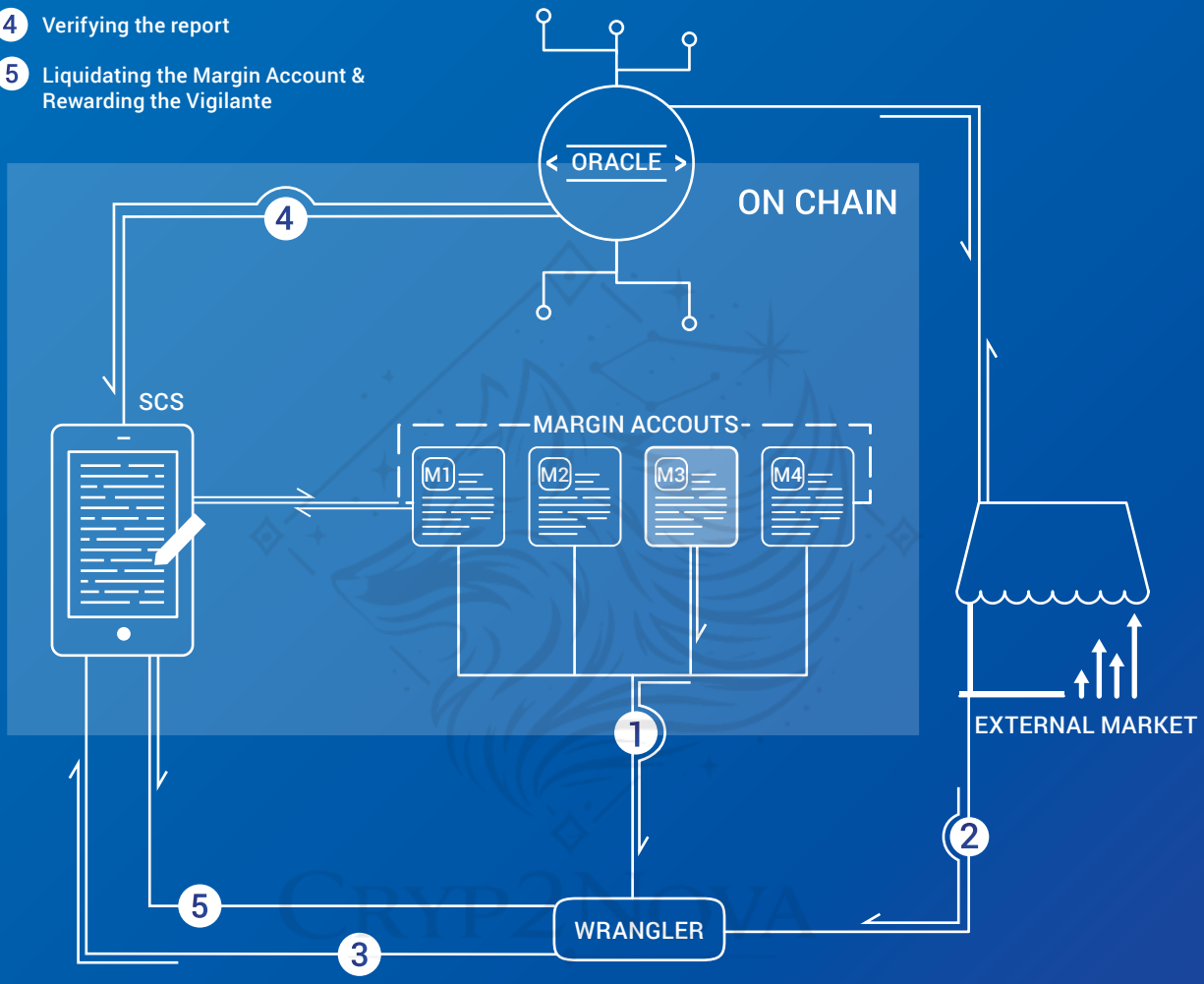


FIGURE 5

2.4.2. Triggering an Auction

Having identified what he deems a terminal Margin Account, the Wrangler already stands to win the bounty attached to it. However, there is more at stake. The Wrangler cannot withdraw the collateral or liquidate the positions any more than the Margin Trader can. He triggers an auction, a competitive process, which demands that the claim is verified by the Lendroid Smart Contract system and also by other competing Wranglers via independent sources of market information. The actual auction process is described in a subsequent section.

Archived copy - cryp2nova.com

3. THE LENDROID AUCTION

The concept of a market order or limit order is inefficient in conventional exchanges and improbable on the blockchain. It requires intractable trust in the centralized exchange. When the collateral and positions of a Margin Account need to be repurposed, they are taken over by one or more Wranglers following a competitive bid. To be clear, this is not a winner-take-all auction. Every Wrangler chooses to repay a small portion of the total outstanding loan and his bid will be proportionately considered. In other words, the Wrangler takes over the account, but only the corresponding percentage of the loan that he repays.

3.1. WHO IT HELPS

The auction's primary aim is to repay the Lender's loan with interest. The auction is complete only if this basic criterion is met. To exercise their right on the collateral and the positions, the Wranglers repay the lender funds proportionate to the percentage they won in the auction. Because the competition prompts Wranglers to bid progressively lower for the collateral, the Margin Trader stands a chance to salvage some of his collateral at the end of the exercise, instead of having to lose it all. The Wrangler who identified the account is paid the bounty, and every Wrangler who bid successfully gets to win a portion of the assets.

In short, the Wranglers get to further their assets at a potentially discounted price, the Margin Trader is not impacted beyond repair, and the Lender's interests are protected.

3.2. HOW IT HAPPENS

Multiple Wranglers can monitor the margin account at the same time. The Wrangler who first identifies the terminal account triggers the startAuction function in the Auction Smart Contract. The Smart Contract triggers an Oracle call and subsequently checks the account's margin level. The oracle, in this case, serves as an anti-spam feature. If the margin level is below the liquidation level, then the auction starts.

Once the auction commences, it remains open for a total of 3 minutes allowing various Wranglers to submit their bids. Say the total amount owed by the trader is 100 ETH. If Wrangler1 pays 10 ETH and Wrangler2 pays 30 ETH, they receive 10% and 30% of the Margin Account respectively. In addition to receiving trade positions, they can also request a portion of the collateral. The collateral requested is transferred to compensate for losses in the trade positions taken over by them. Both the LSTs and the collateral are distributed based on % of the position closed by the Wranglers. The only exception being

Archived copy · cryp2nova.com

the first Wrangler to detect the unhealthy Margin Account who is guaranteed 25% of the LSTs staked as a reward for hisvigilance. The remaining 75% is then distributed among the rest.

To summarize, the % of the position closed will determine how much of the LST and collateral each Wrangler receives, while the % of collateral requested will determine the order in which the Wranglers repay the loan.

Scenario:

Total margin account loans: 100 ETH

Total interest accrued: 1.5 ETH

Total owed to lenders: 101.5 ETH

Wranglers start bidding for a portion or the whole Margin Account, and how much collateral they require in addition to portion of the open positions they have rights over.

	Loan Repaid	Collateral % Requested
Wrangler 1	10	100
Wrangler 2	30	45
Wrangler 3	50	65
Wrangler 4	30	50

Once 3 minutes have passed, the smart contract proceeds to process the bids and allocates the positions, collateral and LSTs accordingly.

Archived copy · cryp2nova.com

The bids are ordered by collateral % requested:

	Loan Repaid (ETH)	Collateral % Requested	Accepted	LST %	Actual Collateral % Distributed
Wrangler 2	30	45	30	10.2	= 45 * 30/101.5 = 13.3%
Wrangler 4	30	50	30	10.2	=50* 30/101.5 = 14.82
Wrangler 3	50	65	41.5 accepted and 8.5 refunded	54.6	= 50 *41/101.5 26.58%
Wrangler 1	10	100	Rejected and Refunded	25	0
Margin Trader					Refunded = 45.3%

TABLE 4

Lendroid will subsequently introduce two types of scripts. One for monitoring Margin Accounts and defaulting loans, and the other to bid auctions that Wranglers can put to use. These scripts are designed to minimize effort in monitoring and bidding.

4. LENDROID SMART CONTRACT SYSTEM

The Smart Contract is a pivotal aspect of the Lendroid protocol. Permissionless access and censorship resistance have been two key focuses of the Ethereum Smart Contract. Considering the evolving nature of the Lendroid protocol, and the immutability of the Smart Contract, we have worked towards a design that enables upgradability and modularity.

CRYP2NOVA

It is important to note that Lendroid is a Smart Contract System. It runs not on one central Smart Contract, but a system of coded rules of how to handle values inside the Ethereum blockchain.

4.1. UPGRADABLE AND MODULAR

In the Lendroid Smart Contract System, there are the Fund Smart Contracts, Business Logic Contracts and the Proxy Contracts.

Funds are held and handled in a Fund Smart Contract. The inviolable nature of these contracts is effectively put to use when defining loan contracts, or executing a range of cryptoeconomic functions.

The Business Rules Smart Contracts, on the other hand, are community-driven contracts. Market parameters, token support and other considerations will be determined by the community. When such modifications are writ, the Proxy Smart Contract redirects to the relevant Business Rules Smart Contract.

Archived copy · cryp2nova.com

5. LENDROID SUPPORT TOKEN

The Lendroid Support Token, or LST, is the native token of the Lendroid protocol. Participation requires a user to possess LSTs. Those that do not are incentivized to earn LSTs by engaging in activities that contribute to the health of the ecosystem.

As with all ERC20 tokens, LSTs will be tradeable following the crowd sale (details in annexure).

The LST has been envisioned for a three-fold objective: to lubricate process and drive utility on the protocol; to incentivize participation; to empower governance. As a non-rent-seeking protocol, self-sustenance can be achieved only if the LST is tied to value creation.

5.1. UTILITY TOLL

To quote Ethereum Foundation advisor William Mougayar, ""...there needs to be a specific linkage between user actions and the resulting effects of those actions on the overall value to the organization."

The LST enables utility and lubricates process on the protocol. There are four players on the protocol, two of which provide utility - the Relayers and the Wranglers - and two others who avail of these utilities and participate to maximize profits - the Lenders and the Borrowers/Margin Traders. On Lendroid, the toll for service or utility providers on the protocol is baked into the system.

Consider the following table. When floating an offer, for instance, the Lender deposits LST fees meant for the Relayer as well as for the Wrangler. When the Margin Trader avails the loan, he too deposits his toll for the Wrangler and the Relayer. The toll is paid only when the loan is closed, and only pro-rata depending on how long the loan was open. However, the deposit is always made in full - the maximum possible fee for the full loan term.

CRYP2NOVA

Transaction	Lender	Relayer	Wrangler	Margin Trader
Avail Loan	-	-	-	-
Close Loan	Pays	Receives	Receives	Pays
Margin Position changed	-	Receives	Receives	Pays
Margin Call	Pays	-	Receives	Pays

TABLE 5

Archived copy · cryp2nova.com

5.2. INCENTIVIZED PARTICIPATION

Even without possessing a single token of LST, a user might participate on Lendroid. As a Wrangler, he can monitor Margin Accounts and choose to earn LSTs by identifying those accounts at liquidation level. In other words, they can be part of the network even without LSTs, by pitching in to provide vital services like monitoring loan lifecycles and margin accounts. In incentivizing the community, Lendroid hopes to sustain its operations and maintain the health of its Margin Accounts.

5.3. ENABLING GOVERNANCE

Even when considering the blockchain ecosystem as a whole, governance is an evolving aspect. Not all the models that work in real life have been transmuted for the ecosystem and nor do they necessarily have the same results [9].

The LST has been envisioned as a vehicle to enable Code upgrades, Network parameters, and Supported assets. The last aspect, for instance, cannot be arbitrary or centrally controlled. While Lendroid is designed to be compatible with all ERC20 tokens, there might be, on a particular day, a particular token that proves too volatile to lend or trade in. In such instances, the community can choose to suspend support to that particular token.

During the initial stages of the system, the code, terms and parameters will be set by the Lendroid team. These will subsequently become community-based processes, allowing for such models as M of N and the Decentralized Autonomous Organization (DAO) or other models that the community adopts [10]. The vehicle for such governance will, however, be the LST.

Archived copy · cryp2nova.com

6. THE LENDROID VISION

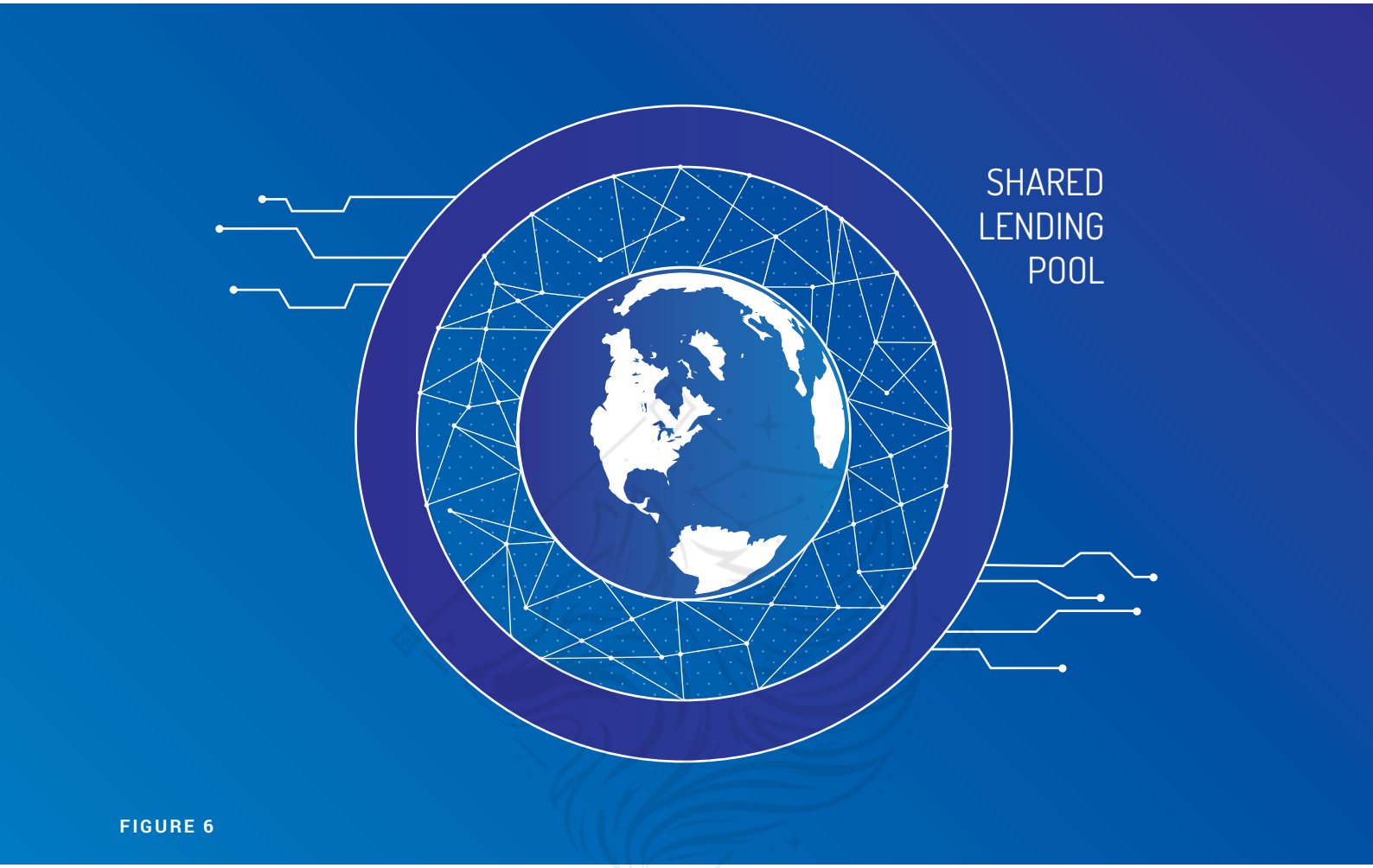


FIGURE 6

LEND AGAINST ANYTHING THAT CAN BE TOKENIZED

We believe that assets in the future will be tokenized and tradeable globally in real time. We believe Lendroid holds limitless applications in marketplaces across industries.

In this whitepaper, we elucidate a protocol on the Ethereum blockchain that makes a founding attempt at bringing together lending, leveraged margin trading and short selling, onto a single protocol – decentralized, trust-independent, non-rent-seeking. The protocol seeks to overcome 'on-chain' limitations of inadequate computational power, latency and impractical gas cost by creating a symbiotic off-chain infrastructure supported by incentivized participants. The native token keeps the network operational, fuels the utility layer of the protocol and offers security to community governance. An important goal of the protocol is to nurture a decentralized, global, shared lending pool, with far-reaching applications beyond just the use case of margin trading. The modular and extensible nature of the protocol could make it the basis for all lending-related use cases for an increasing number of assets that are being tokenized across industries. The protocol's inherent non-partisanship, where no single group of stakeholders draws arbitrary benefit, is inspired by 0x.

Archived copy · cryp2nova.com

7. SYNOPSIS

• Enabling Lending, Leveraged Margin Trading and Short Selling on a Single Protocol

- Non-rent seeking, publicly accessible, decentralized and trust-independent
- Compatible with all ERC20 tokens
- Publicly accessible smart contracts for enhanced interoperability

• Decentralized Lending

- Borrower pledges collateral, eligible for 2.5 times leverage
- Lender defines terms of loan - amount, interest, expiration period - in offer object, with ECDSA signature
- ecrecover function verifies signature, Smart Contract checks validity of loan, Lender's funding account. Funds then credited to Trader's Margin Account

• Shared Lending Pool

- All loan offers are recorded and available to every Relayer on the protocol
- Irrespective of which Relayer records the loan offer, it is accessible by all Relayers
- A Shared Lending pool has far-reaching applications, beyond just margin trading

• Margin Trading

- A Margin Trader can deposit not just one kind of collateral in a token of his choice, but multiple kinds of ERC20-based tokens at the same time, in the same margin account.
- Stakes LST when opening a Margin Account. This is refunded if he closes his positions and repays the loan, and is paid as bounty to Wranglers if the account reaches liquidation level.
- On successful trade and meeting loan terms, Margin Trader may withdraw collateral and profits.

• Interface and Bookkeeping

- Relayers maintain offer books and order books off-chain
- Relayers can offer APIs to facilitate programmatic access to offer books
- With access to the entire lending pool, Relayers can maintain liquidity and create an interface that encourages patronage.
- Relayers are incentivized with the native token LST. They are paid when loans are closed.

• Margin-Level Monitoring and Loan Repayment

- Wranglers are Lendroid-exclusive players, incentivized to monitor Margin Accounts and close loans.

CRYP2NOVA

Archived copy · cryp2nova.com

- Wranglers monitor margin accounts off-chain, 'wrangle' accounts at liquidation level, repay Lender's loan, take over trader's options and collateral.
- Loan repayment is an incentivized aspect of the protocol; encourages symbiosis among the players.
- Loan is closed either by the Trader after successful trade, by Wranglers after taking over Margin Account at liquidation level. In the unlikely case neither trader or Wrangler repay, the lender calls in the loan, withdraws collateral and trade positions.

• Lendroid Auction

- When a Wrangler identifies a terminal margin account, the claim is verified by the smart contract. If ratified, the Wrangler is awarded a bounty (part of the loan offer object, deposited by the Lender) and the triggerAuction function is called.
- The Lendroid auction is not winner take all. The trader's collateral and positions are taken over, but not by any single Wrangler. Every Wrangler chooses to repay a small portion of total outstanding loan and his bid would be proportionately considered.

• Lendroid Support Token

- Powers the utility layer of the protocol.
- LSTs are baked into the protocol for the service providers – Relayers and Wranglers.
- Incentivizes participation on the protocol. For example, without a single LST, a user is authorized to monitor margin accounts and if she identifies an account at liquidation level, stands to earn LSTs
- Envisioned as a vehicle to enable code upgrades, Network parameters, and Supported assets.

8. APPENDIX

From here on, margin trading seemed a logical first use case for Lendroid.

8.1 MARGIN TRADING 1.0

If a trader wishes to monetize his confidence in the price of an asset increasing/decreasing, he would take a position by purchasing/selling the asset. If he is extremely certain, he would build a leveraged position by borrowing additional funds from a lender to magnify his profits [11].See Figure 7.

While the positions are subject to market risks and the trader's decisions, the loan terms are typically defined by a contract, replete with interest rates, a specific time frame for repayment of the loan amount, and legal/punitive action in case of defaulting. Despite its volatility (or perhaps because of it), margin trading is a widely adopted process globally. It is typically carried out through a centralized exchange, which manages the Margin Account.

Archived copy · cryp2nova.com

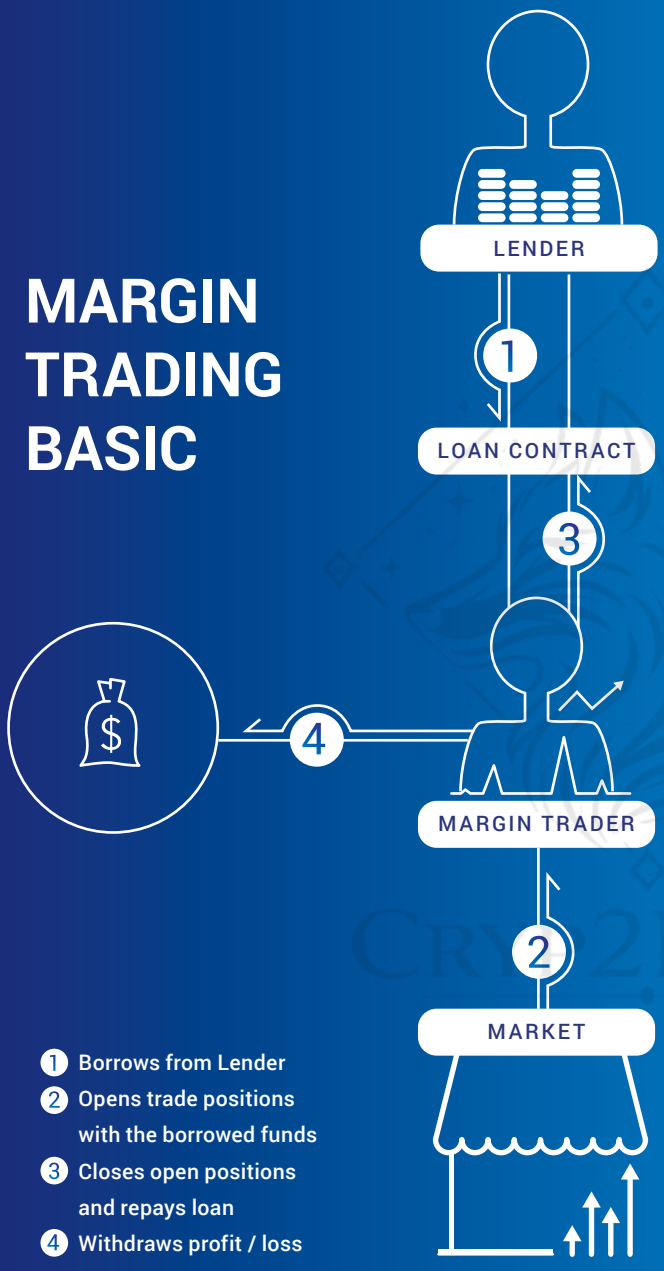


FIGURE 7

8.1.1. Margin Account and Levels

A Trader is mandated to maintain some predetermined levels in his Margin Account at all times. A dip in these levels can trigger warnings or lead to automatic liquidation of the Margin Account. To guard against risk, while exchanges have defined a range of levels exclusive to particular economic systems, some of these levels are used consistently in exchanges around the world.

a. Initial Level: The trader deposits an initial margin, which is a certain percentage of the total traded value. The deposit can be in funds and/or in securities. This initial margin too is determined by the exchange. Centralized exchanges typically set the initial level at 40%. Trading can commence once the initial level is deposited.The trader's adherence to the initial level is used to determine whether or not a loan can be granted.

b. Liquidation Level: Volatility and price are dynamic during trading, and the exchange lends itself to a certain level of uncertainty. However, there is a point at which confidence in the performance of the margin account disappears. The absolute level at which an account is liquidated - the collateral and positions sold off - is the liquidation level. Typically, if the initial level is at 30% to 40%, the liquidation level is set at 15% to 20%.

Archived copy · cryp2nova.com

CONSIDER THE FOLLOWING SCENARIO:

Assumption: 1 XYT(a fictional token) = 0.10 ETH; 1 XXT (a fictional token) = 0.20 ETH

A Margin Trader deposits 1000 XYT as collateral. For an initial margin of 40%, the Margin Trader gets 2.5 times the collateral value, that is, upto 250 ETH. This is the total borrowed amount which she then uses to open a position on XXT for an initial value of 1250 XXT. For a liquidation level set to 20%, Table 1 shows how a few possible scenarios can play out over a timeline. At each time ti, the Margin Level is calculated by the formula:

Margin Level = ((Net Value of Margin Account - Loss in Position) / total borrowed amount) * 100 %

T0:

Collateral value = 1000 XYT * 0.1 ETH = 100 ETH

Total borrowed value = 250 ETH

Position value = 1250 XXT * 0.2 ETH = 250 ETH

Interest accrued = 0 ETH

P/L = Position value - Total borrowed value = 0 ETH

Margin level = Collateral value/Total borrowed value =100/250 = 40%

Margin level = (Collateral value-Interest accrued + P/L) / Total borrowed value = (100-0+0) / 250 = 40%

Time	Collateral Units	Collateral Value	Borrowed Units	Borrowed Value	Position Units	Position Value	Profit / (Loss)	Interest Accrued	Margin Level (%)
t_0: 1 XYT = 0.10 ETH; 1 XXT = 0.20 ETH	1000 XYT	100 ETH	250 ETH	250 ETH	1250 XXT	250 ETH	0 ETH	0 ETH	(100-0+0) / 250 = 40%
t_1: 1 XYT = 0.10 ETH; 1 XXT = 0.25 ETH	1000 XYT	100 ETH	250 ETH	250 ETH	1250 XXT	312.5 ETH	62.5 ETH	1.2 ETH	(100-1.2 + 62.5) / 250 =64.52 %
t_2: 1 XYT = 0.10 ETH; 1 XXT = 0.16 ETH	1000 XYT	100 ETH	250 ETH	250 ETH	1250 XXT	200 ETH	(50 ETH)	2.3 ETH	(100-2.3-50) / 250 = 19.08 % (Margin Call!)

TABLE 6

The Margin Account is liquidated once the margin level reaches a below 20%

Archived copy · cryp2nova.com

8.2. CENTRALIZED EXCHANGE

A centralized exchange is a complex structure, built upon three layers. See Figure 1.

- **Liquidity related activities**
  - User interface for lender and trader
  - Order book maintenance
  - Offer book maintenance
- **Custodianship of user funds**
  - Loan funds and collateral management
  - Accounting
  - Trade position management
- **Margin account monitoring**
  - Margin-level maintenance
  - Margin account liquidation
  - Loan repayment to lender

Centralized exchanges are heavily regulated by the governments of the countries they operate in. They are monolithic, inaccessible except to pre-approved participants, and subject to censorship.

Even under ordinary circumstances, the level of risk can seem prohibitive. For instance, when a margin account reaches the liquidation level, there is no taking over of positions; just straightforward liquidation at the market price. It is not unusual for such 'unrestricted' liquidation to trigger what is known as a flash crash, a domino effect of dropping levels among margin accounts.

But most of all, centralized exchanges have proven to be porous to human error or malicious attacks. Even periodic auditing might not be an airtight solution, since the vulnerability is inherent, even if the threat has not materialized yet. Such exchanges are subject to the same costs, risks and vulnerabilities that any centralized repository of funds or private data would be. This makes such a system the antithesis of the blockchain ecosystem, which is premised on decentralization, on absolute protection and anonymity by global consensus.

However, even within the blockchain ecosystem, margin trading happens, for the most part, via centralized exchanges.

8.3. WHY BYPASS THE BLOCKCHAIN ADVANTAGE?

There are a few digital asset exchanges (poloniex, bitfinex, kraken) that offer margin trading to their customers today. They do this by inviting lenders to their platform and creating a peer-to-peer (P2P) lending pool. The lenders have little to no legal recourse in terms of repayment, and thus fully rely on the exchanges to make sure they get repaid properly.

Archived copy · cryp2nova.com

Thus, exchanges on the blockchain ecosystem, like their counterparts in conventional financial systems, are left with the onerous, dual responsibility of funds and collateral management as well as keeping the traders from defaulting [8].

The exchanges hedge part of the risk by demanding that margin traders hold sufficient margins at any given time, and intervene if there is a risk of default. A relatively high initial margin (20%-40% compared to 2%-10% requested by traditional derivatives exchanges) is demanded and the margin accounts are continuously monitored, and margin traders are requested to rebalance their margin from time to time. If the margin trader fails to maintain sufficient margin levels, the exchange could step in and partially or full liquidate positions. This operation of managing margin accounts is particularly difficult with crypto due to the high price volatility, but exchanges have managed to handle margin calls quite well (barring a few incidents [12][13]) and continue to maintain market integrity.

However, as custodians of customer funds, centralized exchanges expose them to attacks such as DDoS and hacks [14]. This shatters market confidence, and becomes an extra burden on the decision the lender has to make. This dissuasion results in higher overall costs of margin trading.

The exchanges also come under the tight scanner of regulatory agencies in the countries they operate in. These agencies might instruct censoring of certain tokens, ceasing users funds and in some cases denying access to certain users by segregation (geographically) [15].

A decentralized margin trading protocol would help mitigate these issues by moving away from central custodians holding user funds, and by creating a platform that is censorship resistant and can be accessed globally. Further, a global network of lenders will increase liquidity and reduce overall margin trading costs.

However, there are a few shortcomings that are preventing this from happening today.

- Firstly, participants in the decentralized network are not bound by a legal system, but by autonomously binding smart contracts. To decrease the likelihood of a margin trader defaulting, the system has to ensure that at any given time, the margin trader has put up enough collateral to cover any losses he might incur due to sharp changes in market conditions.
- The system should be vigilant in liquidating those margin accounts that are at risk of defaulting.
- The design and computing capabilities of Ethereum are not suited to support on-chain margin trading as it demands recurring computations, access to external data, and a small cost to every operation performed on chain.
- Without a platform for the lenders and borrowers to interact, they are left to discover an appropriate match on their own, making this an expensive and tedious process.

0x paved the way to viable alternatives.

Archived copy · cryp2nova.com

9. ACKNOWLEDGEMENTS

We gratefully acknowledge the principles, research and pure innovation of others on the blockchain ecosystem. Lendroid is but an attempt to take forward the good work others have done. We also acknowledge the support of our backers and advisors. We are grateful for the Lendroid team - talented, passionate people who don't take offense at being called dreamers.

9.1. REFERENCES

1. Koudijs P., Voth H.J., Leverage and Beliefs: Personal Experience and Risk-Taking in Margin Lending, American Economic Review, 2016, 106, 11, 3367-3400.
2. Fortune, P. 2001. "Margin Lending and Stock Market Volatility", New England Economic Review 4, Federal Reserve Bank of Boston.
3. Margrabe W., "The Value of an Option to Exchange One Asset for Another", Journal of Finance, XXXIII (1978), 177-86.
4. https://www.bfxdata.com, Accessed October 1, 2017.
5. Rademacher R. (Lincolnshire, IL, US)., Adkisson D. (Glenview, IL, US), & Maloy D. (Sleepy Hollow, NY, US), 2009, Leverage margin monitoring and management, United States, UBS AG (Zurich, CH), 7577601, http://www.freepatentsonline.com/7577601.html.
6. Warren W., Bandeali A., 2017: "0x: An open protocol for decentralized exchange on the Ethereum blockchain", https://www.0xproject.com/pdfs/0x_white_paper.pdf, Accessed October 1, 2017.
7. Eskandari S., Clark J., Sundaresan V., & Adham M., "On the feasibility of decentralized derivatives markets", https://users.encs.concordia.ca/~clark/papers/2017_wtsc.pdf, Accessed October 1, 2017.
8. Santa-Clara P., & Saretto A., 2007, Option strategies: Good deals and margin calls, UCLA.
9. O'Mahony S., Ferraro F., 2007. The emergence of governance in an open source community, Acad. Management J. 50(5) 1079–1106.
10. Davidson, S., De Filippi, P. & Potts, J. 2016a. Disrupting governance: The new institutional economics of distributed ledger technology. SSRN: https://papers.ssrn.com/sol3/papers.cfm?abstract_id=2811995.
11. Hartzmark, M. L., 1986, The effects of changing margin levels on futures market activity, the composition of traders in the market, and price performance, Journal of Business 59, 147-180.
12. https://blog.gdax.com/eth-usd-trading-update-5d8142b5bdc1, Accessed October 1, 2017.
13. https://cointelegraph.com/news/bitcoin-exchanges-kraken-poloniex-to-be-scrutinized-for-possible-insider-trading-manipulation, Accessed October 1, 2017.
14. http://blog.bitfinex.com/announcements/security-breach, Accessed October 1, 2017.
15. https://www.bitfinex.com/posts/216, Accessed October 1, 2017.

Archived copy · cryp2nova.com



CRYP2NOVA

LENDROID

DECENTRALIZED MARGIN TRADING

Archived copy · cryp2nova.com