CDx: Credit Default Swaps on the Ethereum Public Blockchain

Andrew Young and Julian Wilson

NextGen Blockchain Technologies
andrew@cdxproject.com
julian@cdxproject.com
September 4th, 2018
Version 0.5

Abstract. CDx is a protocol that enables the issuance, trading, and resolution of tokenized credit default swaps on the Ethereum blockchain. The protocol serves as an open standard for participants to both price and trade different types of credit risk in a fully trustless, peer-to-peer setting. It follows the lead of 0x in utilizing off-chain relaying of orders and on-chain settlement of contracts. The protocol handles swap resolution through a decentralized determinations committee that is incentivized to act honestly through the staking of protocol tokens. Liquidity is incentivized at the protocol level through a novel proof-of-liquidity mechanism that rewards active participants. Rehypothecation enables swap sellers to partially collateralize swaps while still ensuring that buyers are fully protected from counterparty risk.

Table of Contents

1 Introduction

The emergence of public blockchains enables the creation of a decentralized financial system that does not rely on trusted intermediaries. These innovative properties have led to an explosion in the number and market value of crypto assets that leverage this functionality. While these assets offer unique opportunities to investors, the capital markets infrastructure surrounding the asset class remains underdeveloped. In particular, the lack of sophisticated structured derivatives limits investor flexibility in managing risk; it prevents investors from offloading unwanted tail risks, thereby increasing the risk premium they demand from the market, and dissuading more risk-averse investors from participating.

Credit risk is the uncertainty that arises from the potential failure of a counterparty to fulfill its obligations. It is an asymmetric risk with payoff distributions that are highly negatively skewed. While tokenized debt assets do not have mainstream usage yet, credit risk is still very present in the crypto ecosystem. Currently, the vast majority of crypto asset trading is done through centralized exchanges, and any investor that holds assets at such an exchange is, in effect, owed an obligation by the exchange to repay these assets upon request. In the all too common scenario that hackers compromise the wallets of the exchange, the likelihood of being repaid this obligation can fall substantially.

CDx addresses this problem by enabling the introduction of an entirely new asset class: tokenized credit default swaps. Credit default swaps are a form of insurance against the default risk of a counterparty, allowing investors to transfer the risk of default to another counterparty. This also allows for the pricing of previously unpriced risk, enhancing the efficiency of crypto asset capital markets. Most importantly, by leveraging the Ethereum public blockchain, CDx substantially reduces the opaqueness inherent in traditional credit default swaps, while still allowing individual market participants to remain pseudo-anonymous.

Over the last twelve months, a number of open protocols, such as 0x and Dharma, have been released. Each one is a crucial part of the core infrastructure of an open decentralized financial system built on top of the Ethereum blockchain. CDx looks to build on their work by offering a set of Ethereum smart contract standards that allow for the creation, issuance, and exchange of tokenized credit default swaps and other event-triggered financial assets.

2 Use cases

In a credit default swap, the seller of the contract will compensate the buyer in the event of a pre-defined credit event. Credit events generally fall under three categories: failure to repay, restructuring, and bankruptcy. In effect, the seller of the swap insures the buyer from some reference entity being unable to repay its obligations.

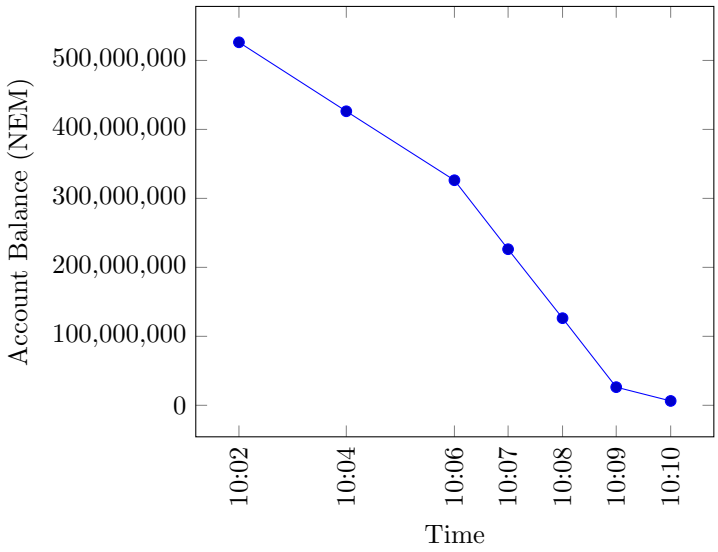
Credit default swaps have utility in any situation in which someone wants to transfer credit risk. The largest source of credit risk that currently exists in the crypto ecosystem is centralized cryptocurrency exchanges. The protocol is

flexible enough, however, to allow investors to trade the credit risk associated with tokenized debt as well.

2.1 Centralized exchanges

The vast majority of digital asset trading is currently done through centralized exchanges. Although these exchanges enable fast trading, efficient settlement, and customer service, they are trusted intermediaries that are highly prone to hacks and default. These hacks are so regular and large in size that hackers have stolen upwards of 14% of all Bitcoin and Ether in existence today [1]. A recent hack involved Japanese exchange Coincheck, which had its NEM wallet drained by a hacker in the span of minutes, resulting in the loss of $500 million of customer funds:

Coincheck Wallet NEM Balance on January 25th, 2018



It is currently impossible for an investor to hedge the risk of these exchanges being hacked and defaulting on their liabilities to customers. This is a major issue for active traders and market makers that must keep their holdings on an exchange in order to trade and for investors that cannot self-custody for legal reasons. This is a perfect situation for a credit default swap to enable the transfer of credit risk from one investor unwilling to bear the risk, to another that is.

For example, an active hedge fund with significant holdings on the Binance exchange wishes to insure against Binance being unable to repay its obligations. To do so, they could buy a credit swap in which they would pay a small premium to a seller in return for a much larger payout, if such an event were to occur. The seller would escrow the large payout into the swap contract and immediately receive the premium. After the expiry of the swap, the seller could call the

contract to return their full collateral. In the event of the agreed-upon credit event, however, the hedge fund could call the contract to collect the escrowed collateral.

2.2 Tokenized debt

While still nascent today, many expect there to be an explosion in the number of tokenized debt assets on public blockchains over the next 5-10 years, especially with the emergence of new tokenized debt protocols, such as Dharma [2]. As the space develops, the CDx protocol will be made compatible with all tokenized debt protocols to enable the issuance of credit default swaps on debt. Credit default swaps average around $1 trillion of notional volume per month, making them one of the largest asset classes in the world [3].

In this scenario, the swap would resemble a traditional credit default swap (CDS), with the default trigger matched up to some type of credit event, such as missed interest payments. A large holder of tokenized debt assets could buy a swap against a particular creditor to insure against the debt assets defaulting. The swap could also be structured to reference a basket of debt assets with different sub-conditional payouts, resembling a traditional collaterized debt obligation (CDO).

2.3 Other use cases

The modular nature of the protocol enables it to be utilized for a wide variety of use cases. Some of which, such as stablecoin insurance, can be easily solved for in the current design, while others, such as generalized insurance markets, would necessitate moderate extensions of the protocol design.

3 Related works

The decentralized finance space is growing rapidly, and there are several key projects in the ecosystem creating various "financial primitives," including options, short-selling, debt, prediction markets, and decentralized trading. Below is a quick summary of the related projects, as well as a brief comparison to CDx.

3.1 Dharma

Dharma is a protocol that enables decentralized origination, underwriting, issuance, and administration of tokenized debt assets in a highly generic and unopinionated construction [2]. The protocol takes inspiration from 0x and administers these debt instruments in an off-chain relaying and on-chain settlement style. Moreover, Dharma places no minimums on the collateralization of these instruments; the project suggests that relayers determine their KYC requirements, if any, and adjust the required collateralization accordingly. In comparison to CDx, Dharma is complimentary and compatible; the CDx core team envisions

a world where one can write credit default swaps on a basket of Dharma debt agreements from one or more creditors and create arbitrary payout rules based on the default of the underlying agreements.

3.2 Augur

Augur is a decentralized oracle and prediction market protocol built on the Ethereum blockchain. It allows one to forecast events and be rewarded for predicting them correctly [9]. In Augur, anyone can create a prediction market for any well-defined event and trade outcomes based on the probability of the event occurring. In a sense, CDx can be considered a prediction market protocol that is optimized for the unique characteristics of credit default swaps.

CDx opts to take inspiration from the 0x protocol to enable a more efficient mechanism of trading the swap instruments instead of using on-chain order books, as in Augur. Second, instead of having a designated reporter for an event, and possibly a random reporter, all CDx events have a binary outcome. The default is that no event has occurred, and there is a designated committee to settle ambiguous outcomes if a challenge is issued. Moreover, CDx supports the use of on-chain credit event triggers, which do not require any dispute resolution. For these two reasons, the team was able to greatly reduce the complexity around resolving disputes and reporting outcomes that exists in Augur due to the generalized nature of the protocol.

Finally, the CDx protocol intentionally restricts the universe of markets to concentrate liquidity, and it incentivizes providers of liquidity with rebates proportional to their contributions (described in more detail in Section 5.2). It also introduces a novel mechanism called *rehypothecation* to reduce collateral requirements on sellers to further drive liquidity and lower premiums, a practice that is not possible in Augur's current design; see Section 10.1 for more details.

3.3 0x

0x is an open, permissionless protocol allowing for ERC20 tokens to be traded on the Ethereum blockchain. The protocol is intended to serve as an open standard and common building block, driving interoperability among decentralized applications (dApps) that incorporate exchange functionality [4]. 0x is the pioneer of the "off-chain relaying and on-chain settlement" idea of trading tokens in Ethereum, and this idea has been interpreted by several projects, including CDx, to additionally trade agreements, contracts, and more complex stores-of-value.

CDx extends the "off-chain relaying and on-chain settlement" idea to create and trade virtual collateralized contracts in addition to tokens. As a result, agreements are not symmetric the way they are in 0x in the sense that the order settlement behavior is different depending on the role of the maker and taker as a buyer or seller. Moreover, there are additional transfers of tokens and funds upon each successful order fill, including escrowing funds, and this additional complexity is reflected in the augmented 0x-style order payload and trading dynamics in the smart contracts.

4 Design

CDx creates one self-contained market for each type of swap. Each swap type has a set of parameters, including the base token, the start date, the expiration date, and the reference entity, which is defined differently depending on if the credit event is on-chain or off-chain.

Sellers of the swap list offer for the specified amounts they are willing to insure against as well as the demanded premium on a relayer. Relayers then broadcast these offers to the buyers who can then buy a portion of the total protection amount by sending a signed transaction to the protocol. For additional flexibility, the sellers do not always have to be the initiators, or "makers," in the market. Instead, it can be two-ways: the protocol supports the buyer as a maker and the seller as a taker as well. After receiving such a transaction, the smart contract transfers the premium in the base token from the buyer to the seller and the "taken" protection amount of base token to the contract for escrow. Furthermore, the buyer is minted a token unique to this swap with an amount proportional to the "taken" amount, representing their eligibility for protection in the case of a successful credit event challenge.

The buyers of the swap are able to exercise it and receive the escrowed collateral any time a credit event challenge is successful, provided they still own the token they were issued. If they have since traded the minted token, then it is the third party who is eligible for the escrowed collateral. Conversely, if there are no valid credit event challenges at expiry time, the sellers can simply collect their escrowed collateral back. Valid credit event challenges lead to a small time period during which eligible token holders collectively vote on the existence of a credit event.

4.1 Participants

There are five participants in the network: *sellers*, *buyers*, *relayers*, *determinations committee members*, and *swap keepers*.

- Sellers offer protection on credit events and escrow collateral for premiums.
- Buyers pay premiums for rights to the collateral if a credit event occurs.
- Relayers are web applications that facilitate order matching.[1]
- Determinations committee members decide if credit events have happened.
- Swap keepers aid in transitioning swap contracts between states.

4.2 Contract overview

There are five core smart contracts that dictate the issuance and trading functionality of CDx swaps: the `SwapFactory` contract,the `Proxy` contract, the `Swap` contracts, the `Exchange` contract, and the `DeterminationsCommittee` contract.

[1] Strictly speaking, relayers are not required in the protocol, but it becomes easier to understand the entire flow if one considers them to be a core participant.

4.2.1 SwapFactory The `SwapFactory` contract is responsible for creating all `Swap` contract types. Creators can create a new type of `Swap` by asking the factory to create one for them. The creator must always supply the required parameters to the `SwapFactory` and one of two groups of parameters, depending on if they are insuring against on-chain or off-chain credit events. In the case of off-chain credit events, once the `SwapFactory` receives a request to create a new `Swap`, it asks the determinations committee members if the parameters are well-defined. If approved, the swap is white listed, and all protocol participants can buy and sell the contract.

4.2.2 Proxy The `Proxy` contract is a contract that proxies for the seller and buyer in the swap transaction. As in the 0x architecture, participants can use the ERC20 allowance functionality (the `allowance()` function) to authorize the `Proxy` contract to be able to move their tokens. Once the user authorizes the contract, the protocol uses it to move tokens on the user's behalf when an order is filled.

4.2.3 Swap The `Swap` contract represents a unique credit default swap. One can think of each `Swap` as a template rather than an agreement, and each template has multiple agreements between pairs of sellers and buyers. CDx allows for a wide variety of `Swap` contracts to be created, with optionality for different triggers (off-chain or on-chain), base tokens, and expiry dates. Each `Swap` registers with the `Exchange` contract, which provides the `Swap` with decentralized trading functionality, as described below. Once the `Swap` is created using the `SwapFactory`, participants can call methods to enter into binding agreements with others. The protocol then records the counterparties involved in the transaction and the amounts. When the swap expires, participants can call a function on the contract to settle the swap and return the correct amounts to the sellers and the buyers depending on the outcome.

4.2.4 Exchange The `Exchange` contract keeps track of all orders ever filled or partially filled for every swap. It also allows buyers and issuers to trustlessly cancel any of their commitments. The fills are represented by a map from the hashes of Table 1 to a number representing how filled the order is. The parameters in the table are referenced when a taker wishes to buy or sell a swap from a maker. This is to ensure that the order is not totally filled yet or canceled. If it is not yet filled, a new mapping is created with the amount that it was filled by.

4.2.5 DeterminationsCommittee The `DeterminationsCommittee` contract can be thought of as the oracle for off-chain credit events. For off-chain events, the determinations committee is the ultimate source of truth, much like in the real world, and through a voting process it ultimately determines the outcome of the credit event. The determinations committee is not used for on-chain credit events because they are explicitly defined.

By default, credit events are deemed to have not occurred. If participants of the protocol believe otherwise, at any time between the start date and expiry date for a particular `Swap`, they can collectively stake one of the protocol's tokens past a dynamic threshold to trigger a new credit event challenge, at which time the committee members will vote on the outcome of the credit event. This is discussed in depth in Section 6.4.

5 Token system

The CDx protocol can rapidly gain network effects if it can be used to deliver trustworthy credit default swaps with minimal transactional costs. CDx credit default swaps will be trustworthy so long as credit events conform to participant expectations. This makes both the definition of on-chain triggers, and the rulings of the determinations committee, vital to the success of the network. Determination of credit events must be done in a transparent process that cannot be unduly influenced by the self-interest of any one party. The transactional costs of CDx credit default swaps will be determined by the level of liquidity in the network. Liquidity leads to lower bid-ask spreads, lower price impact upon execution, and enables a wider variety of participants. To achieve these two outcomes, CDx implements a unique dual-token system that directly rewards participants that contribute value, either in the form of determinations or liquidity, to the network.

Well-designed token systems should fulfill three core functions, as outlined in an essay by ConsenSys software developer Mike Goldin [10]. A token is a necessary element of a system if the use of any other token in its place would damage the system's normal functioning. A system is self-sustaining if it can continue to function normally in the indefinite absence of its creators. A system is a public utility if it is permissionless, rent-free, and does something useful.

The CDx dual-token system fits these three core requirements. First, the CDx protocol is a public utility. It provides usefulness via the ability to buy and sell insurance, it is rent-free, and it is permissionless. The CDx network is self-sustaining in the absence of its creators, as anyone is free to create, issue, and exchange credit default swaps. Lastly, as illustrated in the following sections, replacing the CDx token system with a stablecoin or ETH would make the network far more susceptible to attacks, damage the incentive to provide liquidity, and prevent the network from eventually implementing a decentralized governance model.

5.1 Token design

The CDx token system is comprised of two tokens: *native* and *cred*. While the two tokens have vastly different characteristics, functions, and purposes, their values are inherently tied to each other through the proof-of-liquidity mechanism outlined in Section 5.2.

5.1.1 Native token The *native* token is the medium-of-exchange for the network. In particular, it has three core features:

1. It can be used to pay for transaction fees on the network.
2. It can be staked to participate as a member of the determinations committee.
3. It is transferable, enabling it to be liquid and traded on secondary exchanges.

5.1.2 Cred token The *cred* token is a non-transferable staking token that is rebated to participants who provide liquidity to the network. In particular, it has three core features:

1. It represents stake-weighted votes on protocol governance decisions.
2. It can be staked to participate as a member of the determinations committee.
3. It can be converted back to native tokens at a discount.

The dual-token structure enables a wide variety of participants to hold tokens and contribute value to the network. Shorter-term focused participants can utilize the native token effectively as a discount on transaction fees, while longer-term focused participants can utilize the cred token to have voting rights on the direction of the protocol.

5.2 Proof-of-liquidity

Cred tokens are more important than native tokens for the network's long-term success, as they provide voting rights over protocol parameters. However, because cred tokens are non-transferable, a mechanism for distributing them to market participants is required. Ideally, this system would distribute tokens to stakeholders that have contributed value to the network.

The CDx protocol implements a distribution mechanism that achieves this, which is called *proof-of-liquidity*. In proof-of-liquidity, participants are credited with cred tokens for creating liquidity for the network by posting limit orders that get filled. The amount credited is equal to the amount of native tokens used to pay fees, which are then redirected to a burn address. Cred tokens can be converted back to native tokens at a significant discount.

5.2.1 Wash trading Proof-of-liquidity rewards market participants who provide liquidity to the network with cred tokens. This allows for the possibility that a participant could try to gain undue influence on the determinations committee and protocol governance by simply wash trading swaps with themselves, particularly in the early stages of the network.

There are three measures that make this a very costly proposition. First, because only the maker gets rewarded cred tokens, the participant would be effectively paying $2 for every $1 of cred tokens. Second, because transaction fees scale with the amount of swap notional traded, the participant would need a large reserve of base tokens available in order to lock-up the necessary collateral into the swaps. Third, because cred tokens are naturally rewarded to participants as they provide liquidity, an attacker would have to continually wash trade on the network to maintain any initial head start.

5.3 Token value

It has become increasingly understood that utility tokens, whose sole purpose is to pay for goods or services, are likely to have very high money velocity, and therefore, capture little value. For a utility token to have significant fundamental value, there must be explicit reasons for participants to hold onto the token. In general, there are three core utility token model designs that can be expected to capture significant fundamental value:

1. Work: Tokens staked to perform service in exchange for rewards (e.g. Augur).
2. Governance: Users stake tokens to vote on changes to the protocol (e.g. 0x).
3. Discount: Tokens can be used to lower transaction fees (e.g. Binance Token).

Finally, even if tokens have no utility aside from being used for payments, they can still be valuable if there is a supply sink. Supply sinks, as popularized by Vitalik Buterin, are a mechanism in which some amount of tokens are permanently destroyed on each transaction [11]. Sinks decrease supply over time, thereby increasing the value of each individual remaining token. As a network scales and transaction volume increases, the rate at which tokens are destroyed can accelerate.

5.3.1 Utility Both native and cred tokens incorporate a number of utility elements in their design. The core utility of both tokens is the ability to be staked to perform work for the determinations committee in return for a share of transaction fees. Native tokens can also be considered discount tokens, as rebates of cred tokens effectively lower participants' transaction costs. Cred tokens have the additional utility of being a governance token, due to their voting rights over protocol parameters.

5.3.2 Supply sinks There are also a number of sinks in the network that steadily decrease the supply of both tokens over time. First, native tokens that are used to pay transaction fees are redirected to a burn address. Second, cred tokens can be converted back to native tokens, but at a significant discount, initially set to one half. This means that every conversion between the token types leads to a permanent decrease in the maximum supply of both. Finally, the cred tokens of the determinations committee members can be partially burned for misbehavior, which also reduces supply.

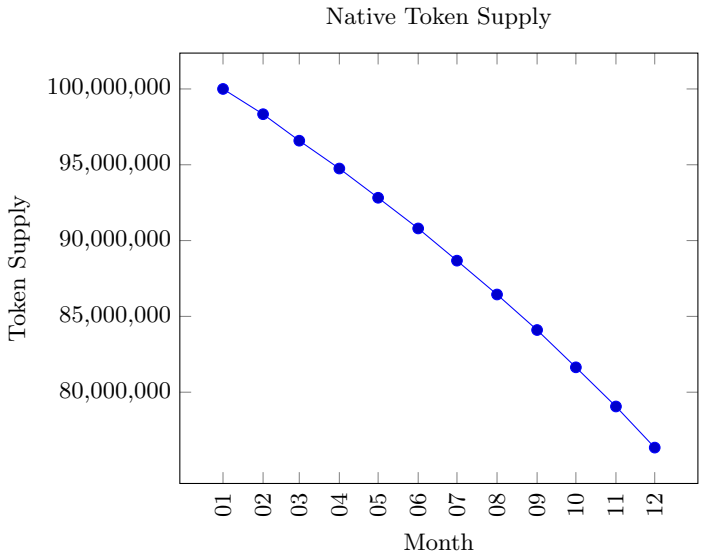
The effect of these sinks can be demonstrated by a sample transaction. Assuming a starting price of $1 per native token and a token supply of 100,000,000:

1. Seller, acting as a maker, posts limit order to sell $200,000 of swaps.
2. Buyer, acting as a taker, fills the entire limit order.
3. Buyer pays the seller's requested premium, seller posts collateral into swap.
4. Buyer pays $1,000 of transaction fees in base tokens (i.e. wrapped ETH).
5. Seller pays $1,000 of transaction fees in native tokens (i.e., 1,000 tokens).
6. Seller is credited back 1000 cred tokens.

7. Seller does not want cred, so they convert their cred back to native tokens.
8. Seller loses 1,000 cred tokens, receives 500 native tokens back.
9. The native token supply is now equal to 999,999,500.
10. Protocol collateral pool increases by $1,000 of wrapped ETH.

The amount of supply burned is dependent on the current market price of native. This produces a self-correcting mechanism in which the total amount of native tokens burned increases if the market price declines. Since the supply is permanently burned, the intrinsic value of each remaining native token permanently increases over time as transactions occur. This holds true even if participants use base tokens, instead of native tokens, to pay for transaction fees. This is because all fees paid with base tokens are sent to the protocol collateral pool, thereby increasing the value of being on the determinations committee.

This process can be modeled out over longer periods of time. For example, assuming a steady native token price of $1, a starting supply of 100,000,000 tokens, a starting daily notional volume of $10,000,000 traded, and a 5% monthly increase in volume, the native token supply would decline over a 12-month period as follows:



In reality, the amount of supply burned would likely be less than illustrated because the market price of native would presumably increase as supply is burned. Cred tokens can only be accrued by utilizing native, meaning that as native tokens are burned, the potential supply of cred tokens decreases as well. This also creates a large incentive to be an early adopter of the network, as cred tokens become progressively harder to accrue as transaction volumes increase and the market price of native rises. However, if the increasing scarcity of cred

tokens drives market prices of native to extremely high levels, cred token holders will have an incentive to convert to native tokens to sell.

6 Swap phases

A swap goes through four phases before full resolution, at which point one can consider the swap "dead," and no further actions can be taken. This section will go through each phase in detail and in chronological order, starting with the *initialization* phase, followed by the *agreement generation* phase, the *dispute* phase (if necessary), and finally, the *settlement* phase. It also goes through the exact definition of a credit event depending on the use case as well as the specifics of the collateral pool. [2]

6.1 Initialization

Before a swap can be traded, it must be initialized and approved by the determinations committee in the case of off-chain credit events. Creating a swap is as simple as calling the `SwapFactory` and passing in some parameters depending on the credit event used. For the required parameters, the creator must supply the expiry date, the start date, and the base token:

- The `baseToken` is the address of the ERC20 token that the swap is priced in. It is also the address of the token that the premium is paid in.
- The `expiryDate` is the expiry time of the swap in Unix ticks, after which no further agreements can be made on this particular market.
- The `startDate` is the start date of the swap in Unix ticks, before which no agreements can be made on this particular market.

For on-chain credit events, the creator must additionally supply the agreement IDs, the threshold numerator, and the threshold denominator. We refer to the latter two as simply the "threshold fraction".

- The `agreementIds` is an array of strings (hashes) that correspond to unique identifiers of debt agreements in the Dharma protocol.
- The `thresholdDenominator` is the denominator of the threshold fraction and must be the same length as `agreementIds`.
- The `thresholdNumerator` is the numerator of the threshold fraction, and it must be at least one and less than or equal to the `thresholdDenominator`.

Note that the `expiryDate` in this case must be less than or equal to the actual expiry date of any of the Dharma debt agreements. Collectively, these parameters reference a credit event and bring the behavior of the swap closer

[2] In a sense, you can think of each contract as a finite state machine with terminal state *resolution*. Certain protocol participants cycle through states by calling appropriate functions on the protocol's contracts.

to a traditional credit default swap. The idea is that by supplying the threshold fraction, users of the protocol can create specialized payoffs and risk functions for their swaps. For instance, one could package several loans from a single creditor into a swap that pays out if over three of the five reference debt agreements default.

Conversely, for off-chain credit events, the creator must additionally supply the reference entity and the reference coins:

- The `referenceEntity` is simply the specific real-world entity that this swap is referencing to determine if a credit event has occurred. It could be a cryptocurrency exchange, for instance, in which case the user would enter "Binance."
- The `referenceCoins` is an array that refers to the canonical symbols of the coins being watched for a credit event. In most cases, this will be coins an exchange has an abundance of, such as "BTC" or "ETH".

At the release of the protocol, we will only allow swaps to be created by the protocol maintainers. This restriction will certainly be removed later, but it is initially put in place to restrict the market to standardized contracts in order to improve liquidity:

- For all swaps, the base token must be either WETH or TUSD.
- For swaps with on-chain credit events, there can be no duplicate contracts where all of the agreement IDs in the `agreementIds` array are the same.
- For swaps with off-chain credit events, the reference coins array must be either length one or two, the reference entities must be well-known cryptocurrency exchanges, and the expiry date must be three months from the contract's creation.[3]

Once the swap has been created, cred token holders hold a vote that lasts three days. If a majority votes "yes," the contract is white listed, and an ERC20 token is minted to represent buyer eligibility in the case of a successful credit event challenge, henceforth referred to as the *protection token*. The protection token can then be traded like a normal token via any medium the buyer sees fit. The `startDate` parameter is included to allow for this voting period; the idea is to set the start date for at least three days after asking the `SwapFactory` to register the swap.

6.2 Agreement generation

The agreement generation phase starts as soon as the contract is instantiated and approved, and is valid between the `startDate` and the `expiryDate`. CDx

[3] Three months was chosen because it is important to balance both the buyers' and sellers' interests. Sellers are happier with shorter time horizons as it allows them to collect premiums more frequently, while buyers prefer longer time horizons to avoid having to continuously enter new agreements. The latter can be solved off-chain with a bot.

leverages ideas from the 0x protocol to facilitate new agreements between sellers and buyers. It is important to note that this should not be confused with creating new swaps with the `SwapFactory` contract. Since the markets on each swap contract are two-sided, two cases need to be considered. The first is when the seller is the market maker, which will likely be the most common, and the second is when the buyer is the market maker. To differentiate between the two, the parameter `isMakerBuyerOfProtection` is used, which is `false` for the former and `true` for the latter.

- If the seller is the market maker, one can imagine the seller essentially broadcasting a message saying, "I am looking to sell up to x of protection on [credit event] for a premium of y expiring in n days in base token z."
- If the buyer is the market maker, one can imagine the buyer broadcasting a message saying, "I am looking to buy up to x of insurance on [credit event] for a premium of y expiring in n days in base token z."

The maker broadcasts a signed message off-chain similar to the 0x message format specifying this intent, as in Table 1. This is called a *maker order*.

This is a cryptographically binding agreement that might eventually be fully executed by a taker, or several takers, and hence, the maker must be prepared to fulfill the full `protectionAmount` or `premium`, unless the maker cancels a portion of its order through a separate transaction. After the maker has created the above message, the protocol allows the maker order in Table 1 to be broadcasted through any desired channel (email, social media, etc.) [4]. The relayer can then aggregate the maker's order into a global order book for everyone to see.

Once there is a taker that wishes to enter the other side of the deal (not necessarily for the full order size), they first authorize the `Proxy` contract to withdraw sufficient tokens from their wallet, and then they call the fill order function on the `Swap` contract with the amount of the swap they would like to purchase/sell (the `takenAmount`) along with the signed message, as in Table 1.

When paying the maker and taker fees, we give the users the option to either pay it to the determinations committee in base tokens, or in native tokens. The protocol effectively offers a rebate if the maker fee is paid in native tokens, but it does induce friction on the end-user, and so it is simply an option and not a requirement. The maker and taker fees will be set by the determinations committee and can be changed quarterly.

For the relayer fee, it is simple: just pay the fee in base token, which will initially be WETH. It is important to note that the relayer ultimately decides what the relayer fee is through its public fee schedule, and it is up to the maker to comply by creating maker orders that satisfy the fee schedule.

The fill order function, when called by a taker after the start date and before the expiry date, does the following in an atomic fashion:

1. Validates that the maker order is still valid.
2. Validates that the swap itself is not expired.
3. Validates that the message payload was indeed signed by the maker and that they have sufficient base tokens.

Table 1. A maker order sent to a relayer indicating intent to make a market on a swap

Parameter	**Description**
`maker`	The address of the maker.
`protectionAmount`	The total amount of base token the maker is offering to insure / willing to buy.
`premium`	The percentage of the eventual taken amount to be paid as a premium to the seller.
`offerExpiry`	The expiration time of the *offer* of this swap. This is not the *expiry* of this swap.
`swapContract`	The ID of the `Swap` contract that they wish to make a market on.
`isMakerBuyerOfProtection`	Indicating if the maker is a buyer or seller of protection.
`takerFee?`	The percentage of the eventual taken amount to pay the determinations committee in base tokens or native tokens. This is not required for on-chain events.
`makerFee?`	The percentage of the eventual taken amount to pay the determinations committee in base tokens or native tokens. This is not required for on-chain events.
`relayerFee`	The percentage of the eventual taken amount to pay the relayer in base tokens.
`isMakerFeeInCdx?`	`true` or `false` depending on if the taker wishes to pay the fee in native tokens. This is not required for on-chain events.
`relayer`	The address of the relayer to receive the `relayerFee`.
`v,r,s`	ECDSA signature of the message which should match the `maker`.

4. Validates that each party, the buyer and seller, has sufficient balances and ERC20 allowances on their accounts to execute the requested trade. This differs depending on if the taker is a buyer or a seller.
5. Validates that the taker has sufficient base tokens to pay the relayer fee.
6. For off-chain events:
   (a) Validates that the taker has sufficient base tokens to pay the taker fee.
   (b) Validates that the maker has sufficient native or base tokens to pay the maker fee.
7. Uses the `Proxy` contract to transfer base tokens from the buyer's account to the seller's account with the amount of `takenAmount * premium` (recall that the premium paid is a fraction of `takenAmount`). The seller now has the premium in base tokens in his/her account.
8. Uses the `Proxy` contract to withdraw `takenAmount` base tokens from the seller's account to the `Swap` contract in escrow.
9. Uses the `Proxy` contract to transfer the relayer fee, `relayerFee*takenAmount`, from the taker's account to the relayer's account.
10. Uses the `Proxy` contract to transfer the taker fee, `takerFee*takenAmount`, from the taker's account to the determinations committee collateral pool, discussed later.
11. Mints the buyer a `takenAmount` of the protection token for this swap.
12. Records that this particular maker order was filled up to the amount requested by the taker.

Additionally, if either the maker or taker fee is paid in native tokens, then that amount of native tokens is burned. Furthermore, if the maker fee is paid in native tokens, then the maker is minted an equal amount of non-transferable cred tokens. Otherwise, the maker fee simply goes to the protocol collateral pool.[4]

To visualize this flow, Figure 1 shows the movement of tokens and funds where the maker is the seller and pays the maker fee in native tokens. In the diagram, the "committee" node is not strictly an entity that holds tokens; one can think of it like a filter that sorts maker and taker fees, routes them to the appropriate destinations, and mints cred tokens.

If there are no takers for a particular swap, the signed message in Table 1 will never be executed, and the funds will never leave the maker's account. As in the 0x architecture, it is additionally possible for the maker to submit another special order, a *cancel order*, that can effectively cancel an unfilled or partially filled maker order.

The astute reader will note that there is a slight issue with the price discovery of native tokens since the swap is denominated in base tokens. In a sense, the protocol requires a way to convert `takenAmount` worth of base tokens into either ETH or native tokens. A solution to this is discussed in Section 8.1.

As a final remark, CDx is only compatible with ERC20 tokens as base tokens due to its dependence on the allowance function in the `Proxy` contract. Due to

[4] Note that unlike the maker fee, there is *no* reward to paying the taker fee in native tokens

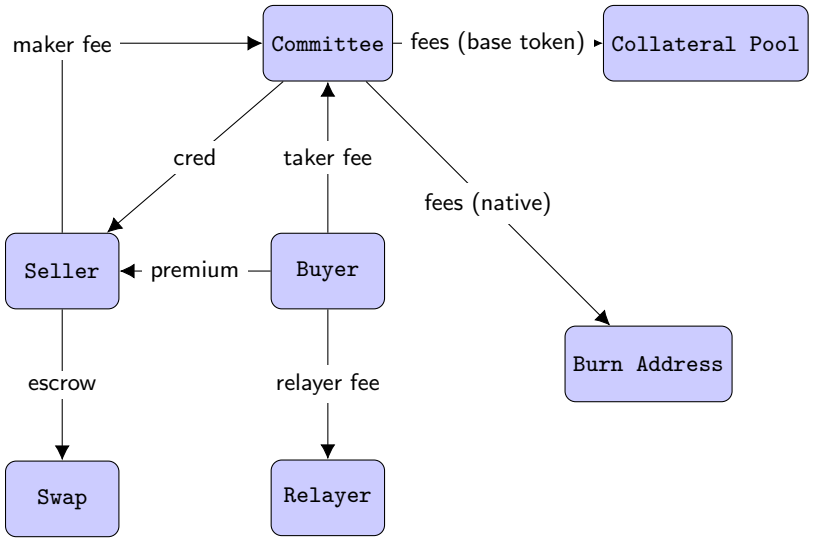


Fig. 1. Token flow resulting from filling a maker order where the seller is the maker and pays in native tokens.

this, CDx will follow the lead of 0x and encourage the base token to either be TUSD or WETH, the True USD stablecoin and 'wrapped ether' respectively. WETH implements the ERC20 contract and can be exchanged easily for ETH at any time by sending ETH to the WETH contract and vice-versa, which leads to minimal friction for users. Alternatively, TUSD can be useful if the parties are interested in trading risk denominated in a nominal dollar amount, as opposed to an amount in ether.

6.3 Credit events

Credit events ultimately determine if the reference entity defaulted or not. For all contracts, it is important to note that a credit event trigger is only valid during the agreement generation period. For swaps watching the repayment status of Dharma debt contracts, a credit event has occurred if at any time during the agreement generation period `getExpectedRepaymentValue() > getValueRepaid()` is true for at least `thresholdFraction * agreementIds.length` [2]. For instance, suppose a creditor is in debt agreements with five different counterparties and the `thresholdFraction` is three of five. It is then said that a credit event occurs if at least three of five of the counterparties did not receive an interest payment before the expiry date of the swap.

For swaps watching off-chain credit events of exchanges, it is said that a credit event has occurred if at any time during the agreement generation period there was a continuous period of n hours where exchange withdrawals were disabled for either of the reference coins. The committee will ultimately determine n, but

based on an analysis of past exchange defaults, a reasonable default time at the start is 120, i.e., 5 days. Note that the days must be continuous; this definition allows for maintenance and other benign short-term downtime-inducing events commonly experienced by exchanges.

6.4 Challenges

There are two possible outcomes of a swap contract: default or expiry. As mentioned above, there is the possibility that the credit event triggers do not accurately represent reality. The determinations committee is used to resolve these cases, with the exception of use cases utilizing on-chain credit event triggers.

6.4.1 Inspiration The design regarding disputes is inspired by the real world Determinations Committee structure utilized by the International Swaps and Derivatives Association (ISDA), which governs the traditional credit default swaps market. In the traditional credit default swaps market, the determinations committee contains 15 members. Ten are sell-side market participants that are selected based on their total aggregate trading volume. The other five are buy-side market participants that are elected. Membership is reassessed annually for both types of members, all votes are counted equally, and to achieve consensus on a particular credit event, an 80% supermajority is required [6]. It is important to note here that "sell-side" and "buy-side" do not necessarily mean seller or buyer of a swap respectively; rather, "sell-side" refers to banks and "buy-side" refers to hedge funds, pension funds, and brokers.

Given the trillions of dollars in notional volume traded in CDS markets today, the traditional system is a reasonable starting point for designing a public blockchain-based credit default swap protocol. Hence, a lot of the design choices that follow result from closely studying both the history and evolution of the current ISDA structure, while adjusting for the inevitable differences of being hosted on a public blockchain.

6.4.2 Committee structure & membership The determinations committee in CDx is composed of two types of members: members who are automatically eligible to be placed in the committee based on their native token balance and members who are automatically eligible to be placed in the committee based on their cred token balance. Initially, the total number of committee members will be 20, and the makeup will be composed of the ten largest native token accounts to opt-in and the ten largest cred token accounts to opt-in.

Every committee term, which will initially be three months, the participants of the committee will have the opportunity to be changed. Two weeks before the beginning of each new committee, high balance cred and native token holders will opt-in to the committee by temporarily escrowing all of their tokens. At the end of this two-week period, which is referred to as the *opt-in period*, the top 10 cred and native token holders who opted in become committee members, and the remaining token holders receive their escrowed tokens back. The protocol

encourages those close to the top ten holders of each token to opt-in anyways, as it is possible several token holders do not wish to be a member of the committee during this term. For a visualization of this timeline, see Figure 2.

The reasoning behind splitting the committee into two groups is due to the cred token dynamics discussed in Section 5.2. The CDx core team expects the market to be very much seller-driven, i.e., the makers will *usually* be the sellers of protection. As a result, since we only reward makers cred tokens to encourage liquidity, over time the sellers of protection will become the increasingly dominant cred token holders. If the committee was not split into two groups, sellers would eventually gain outsized power in the network.

6.4.3 Committee voting To issue a challenge, any native token holder can stake a set amount of native tokens on a particular `Swap` contract. If, at the expiry date, the amount of native staked on this contract is greater than some dynamic threshold based on the "popularity" of the swap, denoted by s, then the contract enters the dispute phase.[5]

Once the swap has entered the *dispute* phase, the determinations committee is called to make a decision on the actual outcome of the credit event in question. This voting lasts for 5 days, and members of the committee can call the vote function once and pass their vote, *true* representing that the credit event happened, and *false* if it did not.[6] For accuracy, the committee members eligible to vote for a particular swap contract are those that were committee members at the swap's creation even if the swap was created during an opt-in period.

At the end of the 5 days, two things are possible: either there is a 51% majority or there is not. If there is a majority, then anyone can call the resolution function to transition the contract to the *settlement* phase. If there is not a majority, then the existing decision will be used. Effectively, this means that unless at least one half of the committee believe that a credit event occurred, a credit event will be deemed to have not occurred.

Lastly, provided the final decision is the same result as what was challenged initially, all of the challengers will receive all of their staked native tokens back. If the final decision is *not* the same result as what was challenged initially, then the challengers' staked native tokens will be burned; this is necessary to prevent the spamming of challenges. To encourage a majority, if a committee member votes on the wrong side of the final decision, then there is a penalty of cred or native tokens depending on the type of member proportional to s. Finally, the contract gets resolved, and the contract can move on to the settlement phase.

6.4.4 Voter apathy and non-votes DAO-type structures are notorious for voter apathy, i.e., there is not a large enough incentive for the DAO token hold-

[5] Technically, there needs to be a participant calling this function. This is discussed further in Section 8.3.

[6] In the implementation, this is a commit-reveal scheme to prevent committee members from influencing others prematurely.

ers to vote on particular protocol functions. To encourage active and correct participation in the committee, the protocol has two direct mechanisms:

1. Committee members who do not vote have their staked tokens burned proportional to s.
2. Committee members who do not vote forfeit their rights to their equal share of the collateral pool for this term.

6.5 Settlement

Regardless of using on-chain or off-chain credit events, swaps have only two possible outcomes that can arise on settlement: *expiration* or *default*.

6.5.1 Expiration Most of the time the swaps will expire worthless, i.e., the credit event did not end up occurring. If the swap expires, then the seller can redeem their escrowed base tokens by calling the swap contract after the grace period *after* the expiry date. The swap contract would do the following:

1. Check that the swap has expired.
2. Transfer the seller's base tokens back to his/her account.
3. Set the swap's internal balance of the seller to 0.

6.5.2 Default If a credit event occurred, either through an automated trigger or via the determinations committee, then the protection token holder (which might no longer be the original buyer) is eligible for a portion of the escrowed base tokens proportional to how many protection tokens they own. For on-chain events using tokenized debt, the following steps occur:

1. Check that indeed a credit event occurred. For Dharma, this is as simple as asking the terms contract and passing in the debt `agreementIds` [2].
2. Check that the caller owns protection tokens for this swap.
3. Send escrowed base tokens to the protection token holder proportional to how many protection tokens they own.
4. Burn all of the caller's protection tokens.

For off-chain credit events, the behavior is similar:

1. Check that the determinations committee has voted on this particular contract and determined that a credit event occurred.
2. Check that the caller owns protection tokens for this swap.
3. Send escrowed base tokens to the protection token holder proportional to how many protection tokens they own.
4. Burn all of the caller's protection tokens.

6.6 Collateral pool

As mentioned in Section 5.2 and the flow in Figure 1, all base token fees are sent to the collateral pool and paid out to the determination committee members. To be precise, at the end of every committee term, committee members that voted on every dispute will have the opportunity to call the committee contract and receive an equal share of the collateral pool, regardless of their native/cred token balance. If any committee member forgets to call the contract, there is no recourse, and the collateral pool will accumulate for the next committee

Figure 2 shows a summary timeline of a committee's term including valid opt-in periods, lock-ups, fee payout distributions, as well as its relation to a swap challenge and subsequent resolution.

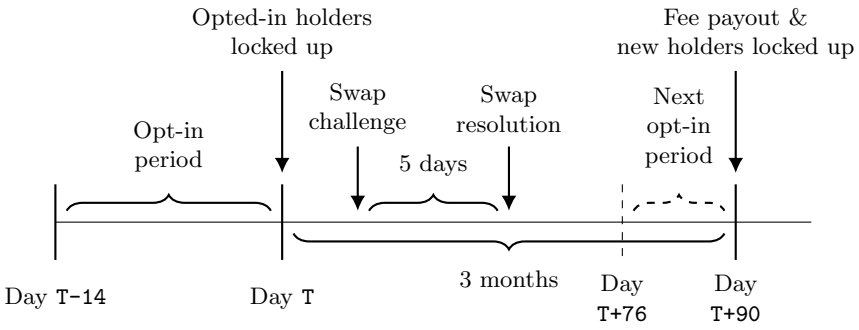


Fig. 2. The timeline of a committee's term and related events.

7 Workflow examples

Below are two sample workflows for Insure Co. acting as a seller of swaps to clarify the ideas presented in Section 6. First, the team presents a scenario using Dharma debt, and secondly, a scenario using off-chain credit events for an exchange.

7.1 Defaulting tokenized debt

Suppose New Co. has recently issued multiple debt instruments through the Dharma protocol to several counterparties. They've raised a large sum of ETH and are paying a premium of 2% monthly (24% per annum). Without going into the specifics of the Dharma protocol, they end up with a number of `agreementIds` that uniquely reference these deals.

The following is a sample workflow for Insure Co. wishing to sell swaps on this particular deal. For simplicity, assume that no such swap contract exists yet,

and so Insure Co. will create a new market through the `SwapFactory` contract expiring in three months from now, denominated in WETH. Insure Co. calls the required function on the `SwapFactory` with the parameters in Table 2.

Table 2. Insure Co.'s default swap

Parameter	Value
baseToken	0x2956356cd2a2bf3202f771f50d3d14a367b48070
agreementIds	[0x1ef, 0x1f0, 0x1f1, 0x1f2, 0x1f3]
thresholdDenominator	5
thresholdNumerator	3
startDate	1539533064
expiryDate	1539792264

In this swap, Insure Co. is insuring against the event that at least three of five agreement IDs default before three months from now. The swap is created, along with a protection token to represent it, and now Insure Co. can signal an intent to issue 100 WETH worth of protection for an annual 5% premium by signing a message similar to Table 3.

Table 3. Message sent by Insure Co. to a relayer indicating intent to sell a swap for 100 WETH protection on New Co.

Parameter	Description
maker	0x2A4F41Fba8928188DE255583B41Eb076906Ae19D
protectionAmount	100
premium	5
offerExpiry	1539792264
swapContract	0xD137eD189a8a3e9bF5541431C9b9Ff1D4EB5c196
isMakerBuyerOfProtection	false
relayerFee	0.1
relayer	0x7bDE6E8b61F30778517b4016E910f92e91770e40
v,r,s	3046022100bb30ee4ce...23170c9

Here, the `makerAddress` is Insure Co.'s Ethereum address, and the `swapContract` is referencing the swap created in the previous step.

A month after Insure Co. sends out their maker order in Table 3, four buyers have entered the other side of the deal and bought this swap, as they believe that New Co. will default on their debt obligations based on new information. To buy this swap, they each took the order in Table 3 and passed it to a relayer with a `takenAmount` of 25, i.e., each of them bought 25 WETH worth of protection. Each of the buyers were then issued 25 of this swap's protection token.

A week after the four buyers entered into the swap, New Co. failed to pay interest payments on four of their Dharma debt agreements, which exceeds the

three of five threshold on the swap. One of the swap keepers was running a bot executing state transitions for the network and successfully reported this event. As a result, the swap was immediately terminated, and each buyer collected their 25 WETH that was escrowed by Insure Co. by presenting their protection tokens.

7.2 Off-chain credit event

Suppose the Binance Exchange is looking to offer 3 months of insurance for their customers' deposits denominated in ether. Binance does not see an existing market for their exchange, and so they ask the swap factory to create a new swap for them. Binance calls the required function on the `SwapFactory` and waits for the determinations committee to approve the contract. Here is what they pass into the function:

Table 4. Binance's default swap

Parameter	**Value**
`baseToken`	`0x2956356cd2a2bf3202f771f50d3d14a367b48070`
`startDate`	`1534271804`
`expiryDate`	`1534531004`
`referenceEntity`	`"Binance"`
`referenceCoins`	`["ETH"]`

The determinations committee votes on this new contract and agrees that the `expiryDate`, the `referenceEntity`, and `referenceCoins` are appropriate and valid, and a new protection token is created to represent this swap. Binance (`0x3e4f41Fba8928198DE255583B41Eb076906Ae19d`) is happy to be the main seller of swaps on themselves, effectively betting against the event that they are unable to keep withdrawals open for an extended period of time. To submit the first sell order of 1000 WETH, Binance sends the following maker order to one of the relayers (refer to Table 5).

Once the order is posted on a relayer's orderbook, several buyers who are presumably holders of crypto on the Binance exchange each decide to take a small amount of the order, and after a month it is fully filled. Binance receives the equivalent premium for the full `protectionAmount`, which is 50 WETH, along with `makerFee*protectionAmount` worth of cred tokens.

During the length of the swap, Binance suffers a severe hack and halts all withdrawals for longer than five days. Before the expiry date of the swap, the buyers of the swap initiate a valid challenge. The contract enters into the dispute phase, and the determinations committee votes on the true outcome of the credit event.

The determinations committee votes unanimously that a credit event occurred, as the downtime lasted for longer than five days. A swap keeper recognizes that the vote is finished and transitions the swap into the settlement

Table 5. Message sent by Binance to a relayer indicating intent to sell a swap for 1000 WETH protection on themselves

Parameter	Description
maker	0x3e4f41Fba8928198DE255583B41Eb076906Ae19d
protectionAmount	1000
premium	5
offerExpiry	1534531024
swapContract	0xe137fd189a83659ae9941431c9b9ee1a4Ec5c196
isMakerBuyerOfProtection	false
takerFee	0.1
makerFee	0.1
relayerFee	0.1
isMakerFeeInCdx	true
relayer	0x7bDE6E8b61F30778517b4016E910f92e91770e40
v,r,s	3abcd22a00bb30ef4cc...23ab0c9

phase. The swap buyers call the swap contract and receive their eligible protection amount, and the swap is now terminated.

8 Protocol maintenance

8.1 Price discovery

As referenced in Section 6.2, the protocol requires a method to "discover" the price of native tokens in terms of base tokens because the protocol allows and encourages the maker and taker fees to be paid in native tokens.[7] The core team evaluated a few techniques to solve this issue and settled on a trusted, centralized price feed that feeds data into a `PriceDiscovery` contract at a frequent interval, which is then referenced by each `Swap`. This approach is inspired by MakerDAO who uses this technique to determine the collateralization levels of their CDPs [8]. Here are two other alternatives the core team considered:

1. Using a tool such as Oraclize to call a collection of exchange APIs and retrieve the prices of the required pairs. The protocol could then volume weight the results and use the final price to determine the exchange ratio. The advantage to this approach is that the protocol maintains strong decentralization; however, it would degrade the user experience, as filling orders now requires two transactions, instead of one.
2. Creating a token curated registry for prices on the 0x protocol. The advantages to this approach include strong decentralization as well as no degradation of the user experience. Liquidity on the 0x protocol is low, however, making the TCR prone to manipulation and/or stale prices.

[7] Note that the conversion between native tokens and cred tokens is implicit due to the haircut set by the committee.

Hence, a centralized oracle chosen by cred token holders is a reasonable solution to this issue, as it provides the best user experience with the most accurate prices while giving stakeholders the ability to react to abuse.

8.2 Fee enforcement

Since the CDx protocol functions just fine without a relayer, the protocol additionally requires a way to enforce maker and taker fees on-chain. Without it, the maker can simply set zero maker and taker fees and hence undermine the token dynamics discussed in Section 5.2. The core team believes that the simplest and most practical solution is to create a contract that maps base tokens to required fees, and have it be committee-controlled with the option to be changed every 3 months. Any order that passes through the `Exchange` contract that does not conform to the fee schedule will not be accepted.

8.3 Swap keepers

As both the swaps and the determinations committee travel through various states, it is necessary to have protocol members watch for valid opportunities for state transitions and execute them for the good of the network. Efficient state transition is necessary for a healthy network, and the most practical way to achieve this is to place the responsibility in the hands of the relayers. The CDx core team will publish and maintain open source "bots" that seamlessly execute these state transitions, minimizing the technical and gas costs on relayers.

An alternative approach is the creation of a decentralized bounty mechanism that incentivizes protocol keepers to execute state transitions for the network. For instance, as proposed by Nadav Hollander, one could allow this bounty rate to be set by market forces, with users effectively bidding for the service through a decentralized market. The idea is that the protocol user with the most vested interest in the state transition would place a sufficiently large bounty to incentivize keepers to execute the transition on their behalf. While it is certainly a more elegant and trustless solution, it introduces additional complexity and attack vectors. The core team will consider this approach in the future after gathering more empirical evidence and conducting further research.

9 Governance

CDx is intended to serve as an open standard for market participants and dApps to create, issue, and exchange credit default swaps amongst one another. The protocol will initially be governed by individuals with a vested interest in the success of the network, but it will eventually be controlled by cred token holders. At the beginning of each new committee term, cred token holders will have the opportunity to change any of the following parameters, effective at the start of the next committee term:

1. The whitelist of swap markets and eligible expiration times.

2. The size of the determinations committee.
3. The penalty incurred during the conversion from cred to native tokens.
4. Maker and taker fees charged on transactions.
5. The challenge fee threshold and the length of credit event challenge voting periods.
6. The oracle used to provide price discovery on native tokens to base tokens.

10 Partial collateralization

Traditionally, debt repayment is enforced through the legal system. Debt is typically structured as legal contracts that give creditors the right to pursue debtor assets in case of a default. However, this structure is not well suited for public blockchain-based debt because public blockchains have pseudo-anonymous accounts, a globally distributed user base, and permissionless access. Pseudo-anonymity makes traditional legal enforcement impossible as lenders are unable to identify debtors. Furthermore, even if identities are known, the globally distributed nature of a public blockchain makes traditional legal enforcement costly and inefficient. While these problems may be solved through rigorous user identification and restrictions on the geographies of users, they eliminate the core benefits of utilizing a public blockchain in the first place.

Like other decentralized finance protocols, CDx deals with this limitation by requiring positions to be fully collateralized. This means that a participant selling 100 WETH in notional of swaps has to post 100 WETH of collateral in order to receive the premium. This locked up collateral guarantees full payment for the buyer in the case of a credit event, thereby eliminating any counterparty risk. However, this reduces swap seller profitability due to the large capital reserves needed to sell the swaps. In effect, it prices out many potential swap sellers because they may have insufficient collateral, which thereby reduces market liquidity and increases premiums for buyers.

An ideal system would guarantee swap repayment to buyers but enable sellers to only partially collateralize their positions. *Rehypothecation* may be the solution. Rehypothecation is the practice by which a party that receives a pledge of collateral then re-pledges that same collateral to *another* party in a separate deal. It is widely used by prime brokers to lower collateral requirements for hedge funds trading derivatives as it allows participants to "re-use" collateral. The practice comes with its own set of risks though, particularly if there is little transparency and if the rules are unclear and/or unenforced. However, since CDx is on a public blockchain, the risks are transparent and the rules are automatically enforced, making the application of the practice much safer.

10.1 Rehypothecation

The following is a preliminary proposal to extend the CDx protocol to enable swap sellers to pledge claims on previously locked-up collateral in order to lower the effective collateral requirements on subsequent swaps. This extension involves

a number of additions to the protocol design, and will likely need to be gradually implemented across different stages.

Swap sellers will now receive "seller swap tokens", which represent rights to any collateral they lock-up when they sell a swap. These tokens are similar to the protection tokens awarded to buyers upon purchase, but will be non-fungible with them given that they represent different claims to the collateral depending on the swaps resolution (seller swap tokens pay out if the credit event challenge is unsuccessful or if the swap expires, and protection tokens pay out if credit even challenge is successful). The amount of seller swap tokens rewarded is equal to the amount of new collateral that has been posted. For example, a swap seller that sells a 100 WETH swap but only posts 50 WETH of collateral in addition to 50 WETH *worth* of other seller tokens would only receive seller swap tokens equal to 50 WETH worth of claims. This prevents collateral from being "re-used" more than once.

As alluded to above, swap sellers can pledge seller swap tokens as collateral on subsequent swaps they sell, however there are strict rules regarding the practice. First, the base token denomination of the pledged seller swap tokens must be identical to the new swaps base token. This is necessary to prevent any mismatches between collateral types. Secondly, the start dates and end dates of the different markets must be identical. Thirdly, there is a minimum base token collateralization level, which is equal to a half of the swaps notional value (i.e. 50 WETH of new collateral for every 100 WETH of notional). Finally, when collateralizing new swaps with seller swap tokens, sellers must over-collateralize the positions. For instance, if a seller is selling 100 WETH of protection and half the collateral is 50 WETH, the other half composed of seller tokens must add up to more than 50 WETH in total claims.

Rather than a set over-collateralization ratio, swaps must be collateralized with a diversified pool of seller swap tokens. In particular, the swap must remain fully collateralized even if two other credit events occur. For example, a swap seller selling 100 WETH of notional that wanted to partially collateralize with swap tokens would have to stake 50 WETH and then four unique types of swap tokens, each with a minimum of 25 WETH worth of claims. In effect, a total of 150 WETH worth of collateral would have to be posted, comprised of the 50 WETH and the 100 WETH worth of claims. This protects the swap buyer in the event that multiple markets default at the exact same time. In this example, the swap would remain collateralized unless four separate credit events were to happen in the same quarter: the market that the original swap references in addition to credit events on three of the four markets referenced by the seller swap tokens.

10.2 Risks & Mitigation

Diversification is considered the one "free lunch" in finance. By combining two investments with non-perfect correlations, diversification enables investors to improve their risk-to-return ratio. This is what this protocol extension attempts to take advantage of, as it enables swap sellers to effectively pool non-correlated

collateral claims as collateral itself. This diversification benefit enables sellers to sell partially collateralized swaps while still guaranteeing full swap payment for buyers in nearly all cases.

However, the downside with this extension is that it raises the risk that swap buyers may not be paid out in full in the unlikely event that there are multiple simultaneous credit events. This was a huge problem in the traditional CDS market during the 2007-08 mortgage crisis, as many large CDS sellers (see: AIG) became insolvent when multiple credit events were triggered simultaneously. While the crisis had many contributing factors, one of the biggest mistakes made by these swap sellers was the underestimation of default correlation risk, i.e. the risk that credit events of seemingly unrelated entities could become correlated during a crisis. This is a risk that CDx is susceptible to as well. For example, while exchange hacks have historically been uncorrelated with each other, a global regulatory crackdown on exchanges is a shared macro risk that could make credit events correlated.

To address this and re-assure swap buyers, there will be a protocol collateral pool that will be used to help backstop collateral shortfalls in the event of multiple, simultaneous credit events. This protects swap buyers in the unlikely event that a multitude of distinct credit events occur at the same time. Cred token holders will govern this collateral pool and ensure that the protocols collateralization parameters are adequate to fully protect swap buyers in even the most extreme cases of credit event correlation. This protocol collateral pool will be seeded upon network launch with adequate reserves of ETH, TUSD, and native tokens. Furthermore, as mentioned in Section 6.6, this pool receives all protocol trading fees, including native tokens that would have otherwise been burned in the current design.

Cred token holders are in charge of protocol governance, which includes setting the minimum base token collateralization level, the amount of distinct seller swap tokens that must be pledged, and the relative value of seller swap tokens claims relative to the outstanding swap notional. If these parameters are set too aggressively and multiple simultaneous credit events occur, the risk that the protocol collateral pool will be insufficiently collateralized increases. In such an event, the protocol would trigger a reverse public dutch auction to sell the reserve of native tokens to cover the claims. This negatively impacts cred token holders because the release of native tokens effectively drives down the scarcity of cred tokens through the conversion process. Hence, cred token holders are incentivized to set conservative collateralization parameters and/or conduct periodic public auctions during low volatility periods to increase the collateral pool balances.

One criticism of this system would be that this extension seems to be overly benefiting swap sellers at the expense of swap buyers. This extension does create the risk that buyers may not be repaid in full if the collateral pool was drained and the subsequent reverse dutch auction was unsuccessful. As a result, this may lower the premium at which buyers will be willing to purchase swaps. However, this will be more than offset by the lower swap premiums that sellers demand

in return, resulting in much high swap transaction volumes than the previous design.

10.3 Workflow Example

Below is a sample workflow to clarify the ideas presented in the previous section. Insure Co. is acting as a seller of swaps that reference cryptocurrency exchanges. In this hypothetical scenario, the assumed likelihood of a credit event is 2.50% per quarter for each exchange. For providing this protection, we assume swap buyers are willing to pay an additional risk premium of 2.50%, thus making the actual premiums paid equal to 5.00%. Finally, the protocol collateral pool contains 1000 WETH and 200 native tokens. Protocol transaction fees are set at 0.50% of swap notional, paid by both buyers and sellers.

Insure Co. has just finished selling 3-month swaps on each of the Bittrex, Poloniex, Coinbase, and Huobi exchanges, each with a notional of 250 WETH. It fully collateralized each of these swaps, thereby receiving the full amount in seller swap tokens i.e. four unique tokens each representing rights to 250 WETH of collateral. For this service, Insure Co. received 50 WETH of premiums (5.00% premium multiplied by 1000 WETH of total swap notional). Furthermore, it posted 1000 WETH of collateral and paid 5 WETH in protocol fees.

Insure Co. is now looking to sell an additional 3-month swap on the Binance cryptocurrency exchange, equal to 1000 WETH of notional. It posts the minimum 500 WETH of collateral and then pledges the four seller swap tokens (totalling 1000 WETH worth of claims) it received from the prior swap transactions to collateralize the rest. Insure Co. receives an additional 50 WETH of premiums (5.00% premium multiplied by 1000 WETH of swap notional) and paid an additional 5 WETH in protocol fees. However, it only posted 500 WETH of additional collateral in this transaction, meaning that it only receives 500 WETH worth of new seller swap tokens.

In total, it has now sold 2000 WETH of notional with only 1500 WETH of collateral for 100 WETH of gross premiums. Accounting for the 10 WETH in fees, Insure Co. earned 90 WETH of net premiums equalling a total return of 6.00% on its capital. Additionally, it now has 500 WETH worth of seller swap tokens referencing the Binance market that it can use to lower collateralization on subsequent swaps. Furthermore, the protocol collateral pool grew by 20 WETH as both buyers and sellers paid 0.50% in fees on total notional of 2000 WETH.

During the length of the swap, Bittrex, Poloniex, and Binance each halt withdrawals for longer than five days. Before the expiry dates, the swap buyers each initiate credit event challenges that resolve successfully. The Bittrex and Poloniex swap buyers each exercise their protection tokens and receive the 250 WETH of posted collateral. The Binance swap buyers exercise their protection tokens and receive the 500 WETH of posted collateral in addition to the 500 WETH of collateral locked in the Huobi and Coinbase swaps. Under these assumptions, and assuming no default correlation, the protocol suffered an event with a probability of 0.0015625% (2.50% * 2.50% * 2.50%) and still maintained full repayment to buyers.

However, if Huobi and Coinbase were to then suffer credit events during this same quarter, there would now be a collateral shortfall of 500 WETH. Note that this would be a highly unlikely event given the priced-in risk (probability of 0.00000097%, or one in 102 million). In this scenario, the protocol collateral pool would be used to cover the buyers claims. Thus, when swap buyers exercise their protection tokens, they would automatically receive their eligible protection amounts but it would flow directly from the protocol collateral pool instead. The collateral pools WETH balance would therefore be reduced from 1020 to 520 and in effect, cred token holders - through their ownership of the protocol collateral pool - backstopped the swaps. Finally, this may push the protocol collateral pool below the emergency threshold set by cred token holders, thus triggering a public auction of the 200 native tokens held in reserve to replenish the pool.

11 Summary

- CDx is a protocol for issuing tokenized credit default swaps on the Ethereum blockchain in a fully trustless and peer-to-peer manner with support for both on-chain and off-chain credit events.
- Credit default swaps are a form of insurance investors can purchase to lower their default risk to certain counterparties.
- In the Ethereum ecosystem, centralized exchanges represent the largest source of credit risk.
- Credit events are handled by a decentralized determinations committee, which is comprised of liquidity providers to the ecosystem and native token holders.
- Cred tokens are earned through a proof-of-liquidity mechanism, in which participants that contribute liquidity to to the network are rewarded.
- Protocol governance will eventually be handled by cred token holders.
- Rehypothecation offers a sustainable way for sellers to reduce their effective collateral requirements, thereby increasing their profitability.

12 Acknowledgements

There are a number of people to acknowledge who have provided valuable feedback and support to the team:

- Max Stein (ConsenSys) for comments on cryptoeconomic design.
- Greg Markou (ChainSafe) for detailed feedback on smart contract design.
- Alex Tapscott (NextBlock) for feedback on strategy and use cases.
- Adam Rabie (BigTerminal) for comments regarding on-chain triggers.
- Sam Pajot-Phipps and team (AION) for comments on protocol design.
- Ryan Roebuck (XDL Group) for advice on business development.
- Brendan Forster and Nadav Hollander (Dharma) for detailed feedback.
- Tekin Salami (Polychain) for feedback around dispute resolution.
- Ryan Zurrer (Polychain) for the idea of a non-transferable reputation token.

Finally, special thanks to NextGen Blockchain Technologies co-founder Jake Hannah for his insights and valuable feedback on the paper, in addition to running the business development side of the project.

References

1. Kharif, O.: Hackers Have Walked Off With About 14% of Big Digital Currencies. https://www.bloomberg.com/news/articles/2018-01-18/hackers-have-walked-off-with-about-14-of-big-digital-currencies (2018)
2. Hollander, N.: Dharma: A Generic Protocol for Tokenized Debt Issuance. https://whitepaper.dharma.io/ (2017)
3. Khwaja, K.: Monthly swaps data review https://www.risk.net/comment/5323946/monthly-swaps-data-review-credit/volumes-peak-in-june (2017)
4. Warren, W., Bandaeli, A.: 0x: An open protocol for decentralized exchange on the Ethereum blockchain. https://0xproject.com/pdfs/0x_white_paper.pdf (2017)
5. Juliano, A.: dYdX: A Standard for Decentralized Derivatives. https://whitepaper.dydx.exchange/ (2017)
6. ISDA: The ISDA Credit Derivatives Determinations Committees https://www.isda.org/a/CHDDE/agm-2012-dc-anniversary-appendix-043012.pdf (2012)
7. Lesaege, C., Ast, F.: Kleros https://kleros.io/assets/whitepaper.pdf (2018)
8. The Maker Team: The Dai Stablecoin System https://makerdao.com/whitepaper/DaiDec17WP.pdf (2017)
9. Peterson, J., Krug, J., Zoltu, M., Williams A., Alexander, S.: Augur: a Decentralized Oracle and Prediction Market Platform https://www.augur.net/whitepaper.pdf (2018)
10. Goldin, M.: Mikes Cryptosystems Manifesto https://docs.google.com/document/d/1TcceAsBlAoFLWSQWYyhjmTsZCp0XqRhNdGMb6JbASxc/edit (2018)
11. Buterin, V.: On Medium-of-Exchange Token Valuations https://vitalik.ca/general/2017/10/17/moe.html (2017)
12. Oraclize Foundation: A Scalable Architecture for On-Demand, Untrusted Delivery of Entropy http://www.oraclize.it/papers/random_datasource-rev1.pdf (2017)