TRS: A Decentralized Protocol for Trustless Financial Derivatives

PRE-REVIEW DRAFT

February 25, 2018

Abstract

We present a decentralized protocol to enable the creation, purchase, and settlement of financial derivatives for any underlying asset. We propose novel systems for maintaining collateral to enable market participants to trade without counterparty or settlement risk. We show that trustless derivatives remove all barriers to access for every financial market, creating a single global marketplace where any individual, smart contract, or decentralized autonomous organization can buy or sell any form of financial risk.

Contents

1 Introduction **2**

1.1 Brief History of Financial Derivatives 3

1.2 Total Return Swap Overview 4

2 Motivation **4**

2.1 Barriers Removed by Trustless Derivatives 5

2.1.1 Unrestricted Access to All Financial Markets 5

2.1.2 DAO and Smart Contract Access to Financial Markets . . 6

2.1.3 Unrestricted Access to Short Selling 7

2.1.4 Unrestricted Access to Leverage 8

2.1.5 Unrestricted Access to Customized, Bespoke Risk 8

2.2 Other Benefits of Trustless Derivatives 9

2.2.1 Stablecoin-like Price Stability Features 9

2.2.2 Simplification of Institutional Custody Requirements for Cryptocurrencies 9
2.2.3 Elimination of Institutional Counterparty Risk and Reduction of Systemic Financial Risk 10

3 TRS Protocol Specification **10**
3.1 Contract Architecture 10
3.1.1 Counterparty Public Addresses 11
3.1.2 Margin and Escrow Accounts 11
3.1.3 Economic Terms 12
3.1.4 Termination Terms 12
3.2 Contract Execution 14
3.3 TRS Order Messaging Protocol 15
3.4 Example of TRS Contract Lifecycle 15

4 Potential Issues and Risks **18**
4.1 Margin and Escrow Balance Security 18
4.2 Price Feeds and Market Data Oracles 18
4.3 Jump Risk and Margin Stop Outs 19
4.4 Third Party Exploitation of Transparency 20

5 Future Work **20**
5.1 Implementing the TRS Network 20
5.2 Margin Netting 20
5.3 Future Proofing: Currency and Blockchain Agnostic Approach . 21

6 Conclusion **21**

1 Introduction

The concept of programmable money has always been a core focus of blockchain research: Vitalik Buterin spoke of “providing users with more powerful ways of managing and entering into contracts using their money” in the original Ethereum white paper [1]. The TRS Protocol seeks to extend these efforts with a specification for trustless, decentralized financial derivatives. We show that trustless derivatives remove *all* barriers to access for *every* financial market, enabling a single global marketplace where any individual can buy or sell any form of financial risk. We further show that the TRS Protocol gives smart contracts and decentralized autonomous organizations (DAOs) the same access to *all* forms of financial risk, opening up dramatic new applications for decentralized financial products.

1.1 Brief History of Financial Derivatives

A financial derivative is an arrangement between two parties based on an underlying asset. Instead of exchanging the actual asset, agreements are made to exchange cash or other assets instead of the underlying asset itself. As the value of the underlying asset changes, so does the net present value (NPV) of the derivative agreement.

Financial derivatives enable market participants to hedge risks that are otherwise impossible to buy or sell, making them one of the most useful concepts of modern finance. Before derivatives, commodity producers had no means of hedging against price decreases of their future production; commodity consumers had no means of hedging against price increases of their future consumption.

The first modern derivatives emerged in 1930s to allow commodity producers and consumers to hedge against price volatility by agreeing to exchange a commodity at specific price in the future. These early derivatives used a centralized clearing house to set specifications of the quality and quantity of a given commodity and manage the margin requirements that ensured both parties would be paid out according to the terms of the contract. These centralized clearing houses created a standard protocol for buyers and sellers to exchange risk, laying the foundation for modern *exchanged traded* derivatives.

As financial markets expanded in the twentieth century, so did the demand for derivatives to hedge against new types of financial risk. The limitations of a standardized contract required by exchange traded derivatives led to the creation of the *over-the-counter* (OTC) derivative market. OTC derivatives are bespoke legal agreements between two counterparties specifying what each counterparty will pay the other as the value of the underlying asset changes.

A typical OTC derivative trade involves one counterparty, the *taker*, requesting a quote for a specific set of terms from one or more *market makers*. The market makers agree to take the other side of the taker's trade as *principal*, meaning that the maker may not have a preexisting position in the risk the taker is looking to buy or sell. By acting as principal, the maker assumes the responsibility to hedge or warehouse the economic risk of the trade; this differs from exchange traded derivatives where natural buyers and sellers of the same risk meet.

OTC derivatives benefit from immense flexibility—they can be written for literally anything—and their usefulness led to the rapid growth of the OTC derivative market in the 1990s and 2000s. It is estimated that over \$540 trillion [2] in OTC financial derivatives are currently outstanding, making the OTC derivatives market the largest financial market in the world by an order of magnitude.

This flexibility comes at a cost: counterparty risk. Without a central clearing house, each market participant must trust that the other party will honor their contracts—even during violent swings in the underlying asset. When a counterparty fails (as *Lehman Brothers* and *Bear Stearns* did during the financial crisis of 2008), trust may fail and significant systemic risk is introduced into the market.

1.2 Total Return Swap Overview

One particularly useful form of OTC derivative is known as a *total return swap* (TRS). Although the TRS Protocol itself is extremely flexible and can be extended to almost any form of financial derivative, we focus on total return swap derivatives as a core example.

A total return swap is a bilateral financial contract where one counterparty pays the total return of a specified underlying asset, including any interest payments or dividends and any capital appreciation or depreciation, and the opposing counterparty pays a regular fixed cash flow [3]. The fixed cash flow can be thought of as the interest rate cost to borrow the capital needed to purchase the underlying asset. The underlying asset is commonly called the *reference asset* and can consist of any combination of corporate bonds, loans, equities or other financial assets. The swap is settled and terminated at a specified date in the future.

This structure allows a counterparty to receive all the economic benefits of owning (or selling short) an asset without the need to buy and custody (or borrow and sell short) the actual asset. This is a powerful tool to buy or sell risk on any type of asset.

2 Motivation

Open financial markets require fair access for all market participants. The TRS Protocol promotes fair and open markets by removing all barriers to access for every financial market, allowing any individual, entity or DAO to access any type of financial risk, without any centralization or single point of failure.

The ability to buy and own (or borrow and sell short) a specific asset has traditionally been the biggest hurdle to accessing a desired financial risk. Derivatives solve this by replacing the need to custody assets with a contract that references the price of those underlying assets. This introduces a second hurdle: the ability to trust that the counterparty of your derivative agreement will

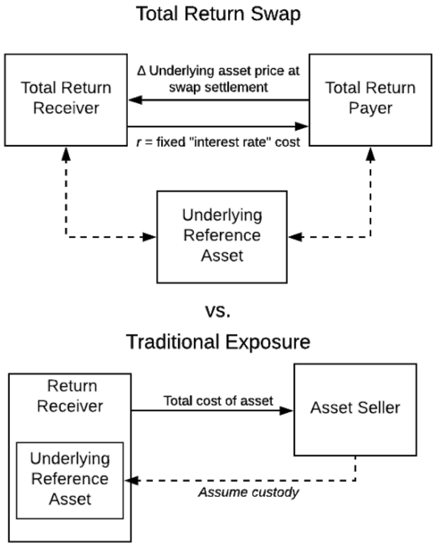

Figure 1: Total Return Swap (TRS) flow

honor the terms of the contract.

Trust, otherwise known as *counterparty risk*, is the achilles heel of financial derivatives today. Because of the risks involved, financial derivatives have only been made accessible to a small number of sophisticated institutional investors who rely on traditional due diligence and costly legal process to “trust” each other. It has never been economical or practical to offer the benefits of derivatives to anyone besides the largest institutional participants.

We show that trustless derivatives eliminate counterparty risk. In doing so, trustless derivatives fully eliminate all barriers to accessing financial markets, creating a single global marketplace. The benefits of this universally accessible, open marketplace are hard to understate.

2.1 Barriers Removed by Trustless Derivatives

2.1.1 Unrestricted Access to All Financial Markets

Traditionally, individuals and businesses can only buy and sell financial risks that are supported by their local government and infrastructure. Regulations and custody requirements can make it extremely difficult (or impossible) for an individual or entity to buy anything not explicitly supported by their local financial system. Sophisticated institutional investors have been able to sidestep these access challenges using tools like OTC derivatives that remove the need to physically own or custody assets; these tools, of course, have not been available to the rest of the market.

Trustless derivatives bring this benefit to all market participants. They enable anyone to buy or sell exposure on any financial asset—market participants are limited only by what market makers are willing to price. Individuals in countries with weak financial infrastructures are no longer restricted to the limited investment options accessible in that jurisdiction. No walls exist—any person or entity with capital can access any risk, creating a single unified and truly global financial market.

Example 1: Investing in US stocks from the developing world

Many people around the world cannot access even the most liquid stock markets. Due to limitations or inefficencies in local financial infrastructures, it is extremely difficult for most individuals in the developing world to invest in foreign assets like the US stock market. With a TRS derivative, the foreign investor can enter into a total return swap that pays the same economics as directly investing in the US stock market, bypassing the limitations of that investor's local financial system.

2.1.2 DAO and Smart Contract Access to Financial Markets

It seems inevitable that smart contracts and programmable money will create many new financial innovations; these contracts will need to invest, hedge, and trade in financial markets.

For a smart contract to access a financial market, it needs to be able to access that market *on-chain*. This presents a problem: how do you credibly reference a traditional, real-world asset on the blockchain? Some approaches like *TrustToken* [4] use on-chain tokens to represent assets held in an *off-chain* legal structure; this approach is subject to a single point of failure and potential off-chain legal challenges.

Trustless derivatives represent another, potentially much more flexible solution: the smart contract simply becomes one of the counterparties of the derivative agreement. Since the derivative references the underlying asset without needing to own or custody the asset, this conveniently sidesteps the problem of credibly representing the asset on-chain.

This allows any smart contract to access any type of financial risk, with potentially dramatic implications: ideas like decentralized private pension plans, decentralized insurance and annuity products, and DAO governed hedge funds or endowments all become possible.

Example 2: Decentralized life insurance

The first "mutual" companies were built to help groups of laborers and artisans pool their collective mortality risk, providing economic aid for families of the deceased. A collective insurance product can be built using the TRS Protocol, enabling any group of individuals to pay into a pool that pays a fixed payment based on their tenure upon their death (either measured by off-chain oracle or the participant's failure to sign periodic challenges with their private credentials). This pool itself can invest into a diversified pool of assets (of both crypto and other traditional assets) using TRS contracts to access this risk.

2.1.3 Unrestricted Access to Short Selling

An obvious advantage of derivatives is the bilateral nature of the agreement: for every counterparty that goes *long* a certain type of risk, there is another counterparty that has the equal and opposite *short* risk. This provides a means to sell short or bet against assets that would otherwise be extremely difficult to borrow and short.

Sophisticated investors commonly use OTC derivatives to access short risk—some notable hedge funds used this technique to profit from the housing market collapse of 2008. But since traditional OTC derivatives are not accessible to most market participants, individuals and smaller entities have no way to access short risk in many markets.

Trustless derivatives remove this restriction. This creates better markets: basic economic theory posits that more market participants freely expressing their market expectations will create deeper, more efficient, markets. This is particularly true for traditionally *one-way* markets (like cryptocurrencies): frictionless access to short selling should reduce price volatility and promote price stability.

Example 3: Shorting a basket of altcoins

There has been explosive growth in both the number and value of crypto assets. But this market is almost entirely one-way: it is nearly impossible to bet that prices will decrease. The few options that do allow you to short assets are centralized solutions that require trust in an exchange or traditional legal agreement. New decentralized solutions like dYdX [5] are promising but still require a short seller to find an existing asset owner who will agree to lend their position. A trustless derivative would provide an easy, simple way to bet against a basket of altcoins: a taker simply enters into a derivative contract with any market maker willing to take the other side. Market makers can hedge their long exposure by entering into other contracts with market participants who want to buy that risk.

2.1.4 Unrestricted Access to Leverage

By not requiring either counterparty to buy the underlying asset, financial derivatives like total return swaps are very capital efficient. The structure gives both counterparties leverage—they can invest in an asset while only locking up a small portion of that capital required to buy that asset (in the required margin).

Traditionally viewed as a type of loan, centralized exchanges and OTC market makers have only offered leverage to trusted and reputable counterparties. Trustless derivatives extend leverage to any counterparty, regardless of their credit rating or pre-existing reputation. This again removes barriers to access, potentially opening up new markets for investing and lending that were historically inaccessible to individuals and smaller entities.

Example 4: Getting a margin loan on a portfolio of cryptocurrencies

Given the dramatic increases in cryptocurrency prices, some early investors are sitting on substantial paper gains. An early investor may want to get some cash (fiat) liquidity without reducing her long crypto exposure. This investor could sell half her crypto holdings and then enter a trustless derivative contract to re-establish their long position with no additional outlay of capital.

2.1.5 Unrestricted Access to Customized, Bespoke Risk

Complex risks can easily be expressed in the economic calculations of a derivative contract by referencing multiple underlying assets. This flexibility enables lower transaction costs by combining multiple trades into a single agreement—for example, a basket of assets that rebalances monthly can be represented by a single derivative transaction. It also enables agreements to be tailored for the specific risk a given counterparty wants to buy, sell or hedge.

Customized, tailored-for-you financial derivatives have never been available to individuals as the costs for doing so have been prohibitive. Trustless derivatives change this and make it possible to offer anyone a derivative that exactly fits their personal financial circumstances, in any market, globally.

Example 5: Investing in a “next-gen” robo-advisor

An investor wants to invest in a diversified portfolio similar to what’s offered by robo-advisors but with an exposure to crypto assets. This investor enters into a TRS agreement to receive the total return of a portfolio invested 50% in the US stock market, 30% in the US bond market, 10% in bitcoin and 10% in ethereum. The agreement rebalances back to the original weightings every time an asset allocation shifts more than 2% from the target exposure.

2.2 Other Benefits of Trustless Derivatives

2.2.1 Stablecoin-like Price Stability Features

The price volatility of Bitcoin and other cryptocurrencies is commonly cited as the biggest barrier to cryptocurrency adoption. Stablecoins, like *Basecoin* or *Fragments*, aim to solve this problem but have yet to obtain widespread adoption.

Derivative are not useful for everyday, commercial transactions, but they provide the same benefits of stablecoins in longer term investment or store-of-value use cases. The economics of a TRS Protocol derivative can reference any underlying asset, including USD based investments, and can therefore achieve whatever price stability or risk profile the counterparties seek.

2.2.2 Simplification of Institutional Custody Requirements for Cryptocurrencies

Investing in cryptocurrencies and other cryptoassets can be difficult for institutional investors. This is largely due to custody and accounting reasons: each new asset requires new systems and processes to be built, tested, and approved, creating significant barriers to entry for every new token or cryptographic system. Institutions can simplify this process by investing via derivatives and standardizing their risk, custody, and accounting systems around a single standard—the TRS Protocol.

Example 6: Accounting and compliance friendly investing in “untraceable” cryptocurrencies like Zcash or Monero

Zero-knowledge cryptographic systems that aspire to make payments untraceable present a serious challenge for an institutional investor. Since the flow of funds is untraceable, it is impossible to audit an investment, preventing many institutional players from investing in things like Monero or

Zcash. A trustless derivative solves this problem by letting a hedge fund profit from changes in the value of the underlying reference asset without having to invest and custody that untraceable asset directly; risk and accounting systems can reference the accounting and compliance friendly TRS contract.

2.2.3 Elimination of Institutional Counterparty Risk and Reduction of Systemic Financial Risk

As painfully witnessed during the 2008 financial crisis, global financial markets were severely affected by the counterparty risk surrounding the failure of major OTC swaps market makers like *Lehman Brothers* and *Bear Stearns*. Counterparties that faced these entities were unsure of their exposure to these potential defaults [6]; this fear affected market perception of other (solvent) counterparties and deteriorating trust nearly broke the entire financial system.

By permanently recording the economics, settlement, and margin terms of all derivative trades on a publicly visible blockchain, decentralized derivatives remove the fear of counterparty insolvency. Eliminating this counterparty risk would significantly decreases the systemic risk present in the existing financial system, helping prevent future financial crises.

3 TRS Protocol Specification

TRS defines a decentralized protocol to enable the creation, purchase, and settlement of financial derivatives for any underlying asset, and introduces novel systems for maintaining margin collateral to enable market participants to trade without counterparty or settlement risk. TRS does this by defining a generalized framework upon which financial contracts can be defined with mutually agreed economic terms, termination terms, and margin requirements. The TRS protocol uses smart contracts on the decentralized Ethereum VM to implement self-policing margin and escrow accounts; this allows the TRS protocol to be fully trustless and decentralized.

3.1 Contract Architecture

The TRS contract contains 5 core components:

- *Public addresses* of both counterparties (the maker and taker)
- *Margin* and *escrow* subaccounts: an escrow account accessible only to the contract, and an margin account for each counterparty, accessible only to that counterparty and the contract

- Logic to calculate the *economic terms* of the agreement (known at the net present value or NPV)
- Logic to govern the *termination terms* and *settlement procedures* of the agreement
- *Contract functions* to add/withdraw margin balances and remargin or terminate the contract

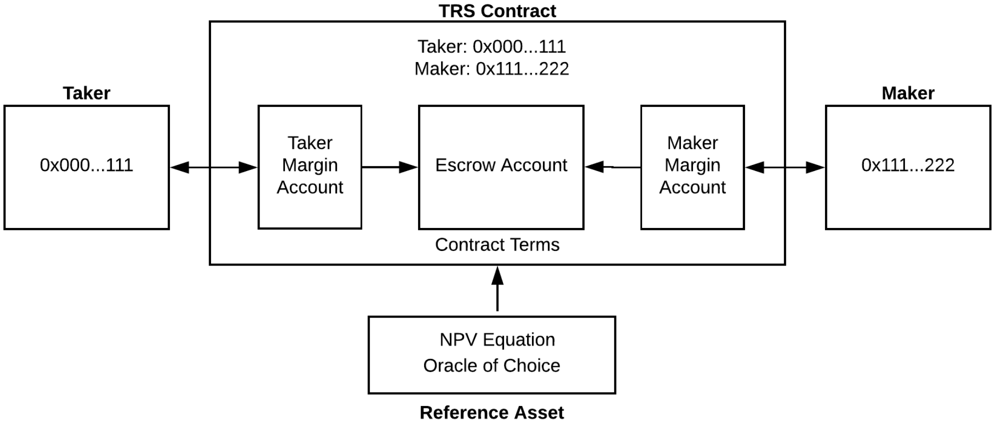

Figure 2: TRS smart contract structure

3.1.1 Counterparty Public Addresses

Since the TRS protocol is fundamentally an agreement between two counterparties, the public addresses of both counterparties are immutably recorded in the smart contract at the creation of the agreement.

3.1.2 Margin and Escrow Accounts

Each TRS contract has three subaccounts: one *escrow* account and two margin accounts, $margin_{\text{taker}}$, and $margin_{\text{maker}}$.

The maker and taker margin accounts exist as subaccounts inside the contract and can be only be accessed by the maker and taker respectively, as well as by the contract's escrow account. The maker and taker are free to add or withdraw funds from their margin accounts at any time; they are only required to keep an appropriate balance in those accounts to meet the agreed upon margin requirements to avoid any potential early termination or default penalties.

The contract escrow account can only be accessed by the logic on the contract itself. Every time the contract is run and the economics terms (NPV) are recalculated, the contract moves the appropriate balance between the margin accounts and the escrow account so that the escrow account balance equals the current NPV of the contract (which will be owed to either the maker or taker).

3.1.3 Economic Terms

The economic terms of the agreement are immutably recorded in the code of the smart contract. This code references one or more price feed *oracles* that return the current price of the underlying reference asset; the challenge of securing the oracle price feed is discussed in more detail later in this paper.

As an example, assume the taker wants to receive the total return of $10mm worth of ETH for one year at the current price of 1ETH = $1000. Assume a maker agrees to this in exchange for being paid 5% on $10mm for the length of contract. (This would approximate the maker's cost of borrowing the funds needed to hedge their sale of ETH with a small profit margin.) The economic terms of this agreement would be:

$$\begin{aligned}\text{Taker Receives} &= (price_{\text{current}} - price_{\text{original}}) * notional \\ &= (ETH_{\text{price}} - \$1000) * \$10\text{mm}\end{aligned} \tag{1}$$

$$\begin{aligned}\text{Maker Receives} &= (date_{\text{today}} - date_{\text{start}})/365 * rate * notional \\ &= (date_{\text{today}} - date_{\text{start}})/365 * 5\% * \$10\text{mm}\end{aligned} \tag{2}$$

$$\text{NPV} = [(ETH_{\text{price}} - \$1000) - (date_{\text{today}} - date_{\text{start}})/365 * 5\%] * \$10\text{mm} \tag{3}$$

If positive, the NPV is owed to the taker; if negative, it goes to the maker.

3.1.4 Termination Terms

The termination terms of the agreement are also recorded in the code of the smart contract. These terms are purposefully designed to be flexible and up to the mutual agreement of the counterparties. In most circumstances, the termination terms would include:

- The *expiry date* of the contract, after which the contract would terminate
- The *settlement procedure* for expired contracts
- *Required margin* balances for both the maker and taker

- The *default procedure* if the required margin balances are not met
- Any *default penalties* to be paid in the event of a default
- Any provisions for *early termination*

By defining this logic in deterministic, immutable code, the TRS Protocol simplifies many of the operational aspects found in traditional OTC swaps. The settlement procedure for expired contracts would empty the contract of all funds by sending the final contract value (the terminal NPV of the contract's economic terms) to the in-the-money counterparty and returning any remaining margin balances back to the addresses of the maker and taker respectively.

TRS Contract Functions	
calcNPV():	Recalculate the economic terms of the function
terminate():	Do the following: (i) Check if either party defaulted on the margin terms; if true, execute the default procedure defined by the contract (ii) Check if the contract has expired; if true, execute the settlement procedure (iii) Do nothing (the contract is still valid)
withdraw():	Allow either the maker or taker to withdraw funds from their respective margin accounts into their public wallet addresses
deposit():	Transfer incoming funds into the margin account of either the maker or taker
remargin():	Do the following: (i) Call `calcNPV()` to determine the current contract value (ii) Move funds between the margin and escrow accounts such that the escrow balance equals the current NPV (iii) Call `terminate()` to determine if the contract should be terminated

Table 1: TRS Contract Functions

At the initial creation of the contract, both the maker and taker are required to contribute, at a minimum, the required margin balance to their respective margin accounts. Counterparties have an incentive to contribute more than this minimum required balance since the contract will terminate under the default procedure if the NPV of the contract causes either party's margin balance to drop below this minimum. As a further incentive to maintain sufficient margin, counterparties can agree to a default penalty to be paid on top of the NPV to any party that defaults. Although the contract cannot force a counterparty to pay a default penalty in excess of the balance in their margin account, if the required margin balance is set sufficiently above the default penalty amount counterparties can be reasonably sure they will get paid this penalty out of whatever funds remain in that margin account.

The margining terms of the contract are extremely flexible by design. Counterparties would change these terms to match the projected volatility of underlying reference asset—more volatile contracts would require more margin. It is also be possible to dynamically adjust the margin requirements by querying an oracle for the historic or implied volatility of the underlying asset.

Early termination is an optional feature of the contract. If both parties agree at the creation of the contract, the termination logic could include a mechanism for either counterparty to request an early termination which may optionally include an early termination fee.

3.2 Contract Execution

The key constraint of any smart contract system is the cost of executing computations on a public blockchain. Our system is no different. It would be optimal if the `remargin()` function of the TRS contract could be continuously executed—the result would be a continuously, perfectly margined contract, with no need for any excess margin at any point. The realities of executing code on platforms like the Ethereum VM make this unrealistic.

Although the cost of on-chain computation is very expensive, modern cloud computing platforms make the off-chain calculation of the contract NPV and termination logic extremely cheap. For this reason, the TRS platform pushes the responsibility for monitoring and remargining the contracts back to the counterparties themselves. The TRS specification allows anyone, at any time, to run the `remargin()` function of any contract on-chain (so long as the requisite gas or equivalent computation cost is paid). Counterparties are therefore incentivized to continuously monitor the economic and termination terms of their contracts off-chain, and then pay the gas to `remargin()` on-chain only when it is economically beneficial for them to do so. Since the economic and termination logic of the contract is embedded into the public blockchain, there is no risk that off-chain observers run the wrong code or calculate an incorrect NPV.

One potential downside of this structure is that more sophisticated counterparties will develop better technology and systems to monitor and remargin their contracts, creating an advantage over less sophisticated counterparties who may “forget” to remargin. Since anyone can call the `remargin()` function, this can be solved by third party *margin custodians* that agree to monitor a less sophisticated counterparty’s contracts for a small fee. Redundancy can further be introduced into this system by using multiple margin custodians.

3.3 TRS Order Messaging Protocol

The TRS Order Messaging Protocol defines a system for swap counterparties to successfully match. The objective is to create a deep and liquid marketplace that matches a counterparty looking to express a certain risk (the taker) with multiple counterparties willing to take the other side of that risk (makers). The taker can then select the maker with the best possible price; competitive forces between makers will naturally push bid/offer spreads to the minimum cost required to hedge the desired risk. Our messaging protocol is modeled after the OTC swap market, arguably the deepest and most liquid financial market in the world.

Order matching works as follows:

1. The taker initializes a TRS smart contract with the terms of the agreement they are looking to enter (specifying the expiry date, notional, economic formula, termination terms, and other required terms). The taker does not specify which direction they want to go (*i.e.* long or short the underlying asset).
2. The taker transfers the minimum margin required into the empty contract.
3. The taker selects one or more market makers and sends them the incomplete contract.
4. Makers respond with two-way quotes on the contract (they do not know the direction of the trade). Makers also authorize the smart contract to withdraw that minimum margin required if the trade is confirmed. The quotes and margin authorizations expire after a short amount of time, known as the *wire time*.
5. The taker selects the quote that is most favorable for the direction they want to go and confirms the trade with that maker before the wire time expires. After confirming the trade the maker owns the risk and decides if, how, and when to hedge.
6. The smart contract withdraws the authorized margin from the winning maker and deauthorizes any margin authorizations from losing quotes. All details are immutably recorded on the blockchain.

3.4 Example of TRS Contract Lifecycle

Assume Alice, a taker, wants to receive on the total return of $10mm of BTC for 1 year vs paying a fixed interest rate. Based on the current volatility of BTC, Alice decides to set the minimum margin requirements at $1mm, and sets a default penalty of $500k. Note that although though margin requirements in this example are expressed in USD, they are paid/stored in ETH inside the contract. Assume initial prices of 1BTC = $\$11,000$ and 1ETH = $1000.

Alice initializes an open contract and deposits 1.5k ETH (currently worth $1.5mm).

Alice's Open TRS Contract	
Alice's Address:	`0x0000... 1111`
Maker's Address:	`null`
Escrow Balance:	`0`
Alice's Margin:	`1500 ETH ($1.5mm)`
Maker's Margin:	`0`
Required Margin:	`$1mm`
Economics:	Pay (or receive) total return of $10mm USD of ETH purchased at $1000 ETH/USD vs receiving (or paying) X% interest on $10mm USD
Termination:	One year or in event of default
Default Penalty:	`$500k`

Table 2: Alice's Open TRS Contract

Alice sends this open contract to two known market makers, Bob and Charlie. Both Bob and Charlie decide to "make a market" on this contract, and both authorize the smart contract to withdraw 2000 ETH (worth $2mm) into the contract's maker margin account if they win the trade (this more than satisfies the required margin of $1mm).

Bob responds saying he will (i) pay the total return to Alice vs receiving 5%, or (ii) he will receive the total return from Alice vs paying 4.75%. Charlie quotes a market of (i) paying the total return vs receiving 5.2% or (ii) receiving the total return vs paying 4.9%. Both Bob and Charlie tell the smart contract that they will hold their markets for 5 seconds (the wire time).

Since Alice wants to receive the total return of the BTC, she accepts Bob's offer to pay her the total return vs receiving 5%. (Alice would rather pay a fixed rate of 5% to Bob than 5.2% to Charlie). The smart contract informs Bob he is "done" on the trade and transfers the 2000 ETH Bob already authorized into his margin account within the smart contract. The contract also cancels the margin withdraw authorization Charlie gave. The trade is confirmed and the complete contract details are recorded on the blockchain.

The next day BTC rallies to 1BTC = $\$12,000$ while the price of ETH remains unchanged. Off-chain both Alice and Bob monitor and recalculate the

Final Contract Between Alice and Bob	
Alice's Address:	`0x0000... 1111`
Bob's Address:	`0x1111... 2222`
Escrow Balance:	`0`
Alice's Margin:	`1500 ETH ($1.5mm)`
Bob's Margin:	`2000 ETH ($2mm)`
Required Margin:	`$1mm`
Economics:	*Alice receives* the total return of $10mm USD of ETH purchased at $1000 ETH/USD vs *paying Bob* 5% interest on $10mm USD
Termination:	One year or in event of default
Default Penalty:	`$500k`

Table 3: Finalized contract between Alice and Bob

NPV of the contract and margin requirements. Since BTC/USD rallied 9.09% (from $11k to $12k) and since one day of interest has passed (worth $1370), both Alice and Bob calculate an NPV of $909,090 - $1370 = $907,270 in Alice's favor. Bob knows that if the contract's `remargin()` function is called on-chain, his margin balance of $2mm will be depleted by $907k and he will be dangerously close to dropping below the minimum margin requirement (and risk paying the default penalty of $500k). He therefore deposits an additional 1000 ETH (worth $1mm) into his margin account inside the smart contract.

Meanwhile Alice decides that is it worth paying the gas to remargin the contract on-chain. She calls the `remargin()` function and pays the necessary gas. The `remargin()` function calls `calcNPV()` which queries the BTC/USD oracle and determines that the current NPV of the contract is $907,720 in Alice's favor. The contract then moves 907.720 ETH (worth $907,720) from Bob's margin account into the contract's escrow account such the the escrow balance now equals the NPV of the contract. Bob's margin account has now changed from 2000 ETH (at the start of the contract) to 3000 ETH (after Bob deposited an additional 1000 ETH) to 2092.280 (after 907.720 ETH where moved into the contract escrow account when `remargin()` was called on-chain).

Alice observes this entire process since the margin and escrow balances of the contract are publicly available. Alice sees that the 907.720 ETH in the contract's escrow account that would be paid to her if the contract were to terminate. Since she also has 1200 ETH in her margin account, the contract currently has 2107.720 ETH (worth $2.107mm) of value attributed to her, leaving her over collateralized by 1107.720 ETH (or $1.107mm) based on the $1mm required margin amount. Alice could withdraw this 1107 ETH if she so chooses.

The process continues with Alice and Bob continuously monitoring the NPV of the contract as well as the margin and escrow balances. Since Alice is not a professional market maker, she may also decide to hire a third party margin custodian to monitor and remargin her contract according to certain parameters.

After one year, BTC has rallied to 1BTC = $14,000. On the day of termination, Alice calls `remargin()` a final time. The final NPV is calculated as a total return of $2,727mm (the appreciation of $10mm worth of BTC from $11k to $14) less a fixed interest cost of $500k (the 5% interest on $10mm), leaving a total of $2,227mm owed to Alice. The `remargin()` function moves this amount out of Bob's margin account and into the contract's escrow account. The `terminate()` function then returns any remaining margin in Alice and Bob's margin accounts to their respective public wallet addresses and deposits the final escrow account value of $2,227mm into Alice's wallet.

4 Potential Issues and Risks

4.1 Margin and Escrow Balance Security

The transferability of funds contained within the margin and escrow subaccounts of the smart contract is severely encumbered by design. At creation of a TRS agreement, the public addresses of the maker and taker are immutably recorded on the blockchain. The logic of the contract then permits funds in the margin account of each counterparty to be transferred only to one of two places: the escrow account of the contract, or the public address of that counterparty. Margin account funds cannot be transferred to any other address.

Funds in the contract's escrow account are even more encumbered: the logic of the contract forbids funds in the escrow account to be transferred anywhere other than the margin accounts of the counterparties.

This design means funds embedded in the contract will never be transferred to any address other than those of the counterparties as agreed at the start of the contract.

4.2 Price Feeds and Market Data Oracles

The TRS Protocol requires reliable, consistent, and accurate price feeds to calculate the net present value of any agreement. Since no blockchain or cryptographic system has the innate ability to know things like the EUR/USD exchange rate, TRS requires a trusted oracle to communicate price and market data. If the oracle is compromised, the contracts could be manipulated.

The oracle problem is relevant to many domains outside of just derivatives, and much work has been done to solve it. In 2014, Vitalik Buterin first proposed a game-theoretic approach of using Schelling points to find a truthful value from a crowd [7]. More recently, Zhang et al. proved the security properties of their *Town Crier* system for authenticated data feeds for smart contracts [8]. Commercial companies like *Oraclize* and *SmartContract* currently provide authenticated feeds using the TLSNotary [9] or Town Crier specifications, optionally using Intel SGX trusted hardware systems. Systems for combining multiple authenticated data feeds have also been proposed [10]: for example, price feeds like BTC/USD can be pooled from multiple exchanges with feeds weighted by trading volume or with outliers thrown out.

These technologies greatly reduce the risk of oracle manipulation. Since both counterparties agree to the economic terms of the contract at its creation, which includes oracle selection, both counterparties have an incentive to adopt the most trusted oracle.

Critics of oracle-based systems often point to the computational cost of pinging oracles on the blockchain, arguing that the cost and latency of these on-chain transactions falls far short of what is achievable with centralized exchanges. The TRS Protocol sidesteps this by putting the responsibility for monitoring and remargining contracts in the hands of the counterparties, where off-chain monitoring costs are minimal. Oracles only need to be called on-chain when either counterparty decides to run the contract's `remargin()` function on-chain.

4.3 Jump Risk and Margin Stop Outs

Any financial derivative that uses margin or leverage has some risk of a violent, unexpected price move quickly depleting the margin of a counterparty: we call this *jump risk*. Traditionally, the solution to this was to rely on some trusted reputation framework (like the legal system) to ensure that margin calls were met; a trustless derivative system purposely avoids this.

The TRS Protocol allows counterparties to self-manage jump risk by specifying the required margin, default procedure, and default penalties at the creation of the contract. Counterparties are naturally incentivized to set higher margin requirements on contracts with more volatile NPVs, and to set higher default penalties for contracts where defaults are costly (*i.e.* contracts with risk that is difficult to hedge or to recreate). These terms can be set dynamically too: contracts could specify the margin requirements by querying an oracle for the current volatility of the underlying asset, thereby allowing margin terms to adjust to changing market conditions.

Although the trustless nature of the protocol removes any need for a preexisting reputation for any counterparty, positive reputation can be useful to repeat users of TRS. Since the history of every TRS transaction is permanently recorded on the blockchain, some market participants may prefer counterparties with demonstrated "good" behavior. For example, a taker may choose to trade with the marker maker who has a history of not defaulting on their margin requirements and reliably contributing additional margin when needed. Similarly, a market maker may choose to offer lower margin requirements to a counterparty with a solid track record of avoiding defaults. The value of positive reputation creates another economic incentive for users to make good on the terms of their contract.

4.4 Third Party Exploitation of Transparency

The economic terms, margin requirements, and current margin and escrow balances of every TRS trade are freely observable to any third party on the public blockchain. Conceptually, a profit seeking third party may look to exploit this transparency by identifying points where a large number of trades with similar economics could deplete their margin accounts and “stop-out”. The exploiter could buy or sell the underlying asset and attempt to quickly move the price through these “stop-out” points, hoping to profit from a wave of one-way price action.

This exploit will fail as other market participants can also analyze the blockchain looking for stop-out points and can position and margin themselves appropriately. Since all information is public, no participant (including the market makers) can benefit from inside information on market supply and demand. Any attempts to exploit the public record will be defended by market participants operating in their own economic best interest.

5 Future Work

5.1 Implementing the TRS Network

The TRS Networks consists of three interwoven components: the *TRS Protocol*, the generalized, open-source specification for trustless derivatives described in this paper; the *TRS Token*, an ERC20 token for governance of the TRS Network and a mechanism to create economic incentives around adoption of the TRS Protocol (details to be published separately); and the *TRS Foundation*, a not-for-profit foundation with a dual mandate of supporting the development of the TRS Protocol and promoting efficient markets on the network. The market side of the Foundation will be modeled after financial service industry associations like SIMFA [11] and ISDA [12]: like those organizations, the Foundation will act to coordinate best practices between market participants and foster the development of deep, liquid, and efficient markets.

5.2 Margin Netting

As the TRS Network becomes more liquid, market makers will seek to reduce the total amount of margin required for their portfolio of contracts. Consider the instance where Alice has two offsetting trades, one that pays the total return of $1mm BTC to Bob, and another that receives the total return of $1mm BTC from Charlie. Without some form of margin netting, Alice would be required to post required margin to both these contracts—even though she has no risk between the two positions.

Forthcoming research will detail mechanisms for market makers and other frequent users of the TRS Protocol to *net* contracts of similar risk, vastly reducing the total amount of capital required by market makers with offsetting trades. The risk exposure, margin requirements, remargining rules, and governance of these netted trades are publicly visible on the blockchain, maintaining all the benefits of the decentralized and trustless protocol.

5.3 Future Proofing: Currency and Blockchain Agnostic Approach

Althought the TRS Protocol is currently implemented using the Ethereum blockchain with Ether (ETH) as a common currency for all margin transfers and settlement payments, the specification is designed to be blockchain and currency agnostic. The protocol is easily portable to other blockchains, and it is the Foundation's intention to port both the implementation and the best practices, standards, and incentive mechanisms to every backbone used by market participants. This supports our vision of a single, unified global marketplace, supported by a common, unified, protocol.

6 Conclusion

The TRS Protocol is a decentralized specification to enable the creation, purchase, and settlement of financial derivatives for any underlying asset. The protocol uses novel systems to securely maintain collateral, allowing market participants to trade without counterparty or settlement risk. The computational limits of continuous on-chain margining are avoiding by enabling market participants to monitor their own contracts off-chain and triggering the smart contract's remargin() function when economically beneficial.

Most of all, the TRS Protocol reveals a plan to build a new open, trustless market operating system—laying the foundation for what could grow to be deepest, most liquid, and most accessible exchange in the world.

References

[1] Ethereum: A Next-Generation Smart Contract and Decentralized Application Platform

https://github.com/ethereum/wiki/wiki/White-Paper

[2] OTC derivatives statistics at end-June 2017
https://www.bis.org/publ/otc_hy1711.htm

[3] A. Bomfim, Understanding Credit Derivatives and Related Instruments. UK: Elsevier, 2004, ch. 7.

[4] TrustToken project website.
https://www.trusttoken.com

[5] A. Juliano, dYdX whitepaper, 2017.
https://whitepaper.dydx.exchange

[6] Risk Management Lessons from the Global Banking Crisis of 2008, SSG, 2009, pg. 6.
https://www.sec.gov/news/press/2009/report102109.pdf

[7] V. Buterin, SchellingCoin: A Minimal-Trust Universal Data Feed, 2014.
https://blog.ethereum.org
/2014/03/28/schellingcoin-a-minimal-trust-universal-data-feed/

[8] F. Zhang et al, Town Crier: An Authenticated Data Feed for Smart Contracts, 2016.
https://eprint.iacr.org/2016/168.pdf

[9] TLSnotary - a mechanism for independently audited https sessions, 2014.
https://tlsnotary.org/TLSNotary.pdf

[10] R. Brodetski, Introducing Oracul: Decentralized Oracle Data Feed Solution for Ethereum, 2017.
https://medium.com/@roman.brodetski/introducing-oracul-decentralized-oracle-data-feed-solution-for-ethereum-5cab1ca8bb64

[11] Securities Industry and Financial Markets Association Website.
https://www.sifma.org/about/

[12] International Swaps and Derivatives Association Website.
https://www.isda.org/about-isda/

[13] Shayan Eskandari, Jeremy Clark, Vignesh Sundaresan, Moe Adham: On the feasibility of decentralized derivatives markets, 2017.
https://users.encs.concordia.ca/ clark/papers/2017_wtsc.pdf

[14] VariabL project website.
https://variabl.io/