Beacon - A Decentralized Asset Exchange Protocol

Charles Cooper and Mandara Gabriel

info@beaconexchange.io

November 19, 2018
v1.0

Abstract

Decentralized exchanges currently suffer from a number of design issues. While decentralized asset settlement and transfer has been enabled by existing blockchain designs, a decentralized matching engine exposes participants to counterparty risk in the form of front-running and information leakage by liquidity takers, or reneging on obligations by liquidity providers. We propose a peer-to-peer protocol for broadcasting, discovering and executing on liquidity, along with on-blockchain arbitration mechanisms to level the playing field between liquidity providers and liquidity takers. By removing the requirement for an external matching engine and instead introducing opt-in forfeiture fees, we create a risk market for liquidity which allows market participants to measure and monetize liquidity risk.

Contents

1 Introduction

People have traded goods and services since the dawn of civilization. In the modern day and age, trading is institutionalized in the securities exchange marketplace. The marketplace serves as a place to discover potential trading opportunities, intermediated by a system of auctioneers or brokers. Perhaps the most well-known of these institutions is the New York Stock Exchange, or NYSE.

The NYSE's trading format can be traced back to its roots over 200 years ago. On May 17, 1792, 24 stockbrokers signed what is now known as the Buttonwood Agreement, agreeing to trade securities only among themselves and to maintain a commission rate floor. Named after the buttonwood tree standing on Wall Street in New York City, this agreement established the "specialist" system wherein an appointed person oversees trading in each security, matching customer orders and trading on a principal or agency basis as circumstances require to maintain an orderly market. Although not technically the first stock exchange in the United States, the NYSE remains the largest stock exchange in the world by market capitalization and maintained a monopoly on stock trading until the NASDAQ began gathering trade volume two centuries later.

In 1997, an electronic quoting and matching system called Island ECN (today wholly owned by NASDAQ) executed its first trades and quickly gained popularity. From the perspective of this paper, Island's primary innovation was to avoid trading on a principal basis at all. Instead, it served solely as a matching engine for its customers, using its computers to match customer order flow on an agency basis through a predetermined set of rules. Enabled by technology, this regime change had a profound effect on reducing trading costs over the next decades.

While the cost of trading has been significantly reduced, customers still need to rely on a set of central institutions to act as agents to match orders and settle trades. This leads to monopolistic practices, potential for fraud or manipulation (though disincentivized through regulation), and barriers to entry for traders. This paper explores the possibility of decentralized exchange: how to enable traders to trade on a peer-to-peer basis without central institutions acting as intermediating agents, while still retaining the benefits of liquidity discovery, order matching and trade settlement.

1.1 Distributed Ledgers

In 2009, the Bitcoin paper [1] proposed a peer-to-peer electronic cash system which did not rely on any trusted clearing house. Prior to Bitcoin, the primary obstacle in implementing such a system was that participants with the ability to write to the ledger could rewrite it to their advantage and mint money at the expense of other users.

In short, the solution proposed in the Bitcoin paper was an incentive system better known today as mining. Nodes propose a block to append to the shared ledger (also known as the blockchain), incurring an up-front cost in return for

an economic gain should the block be accepted by other nodes. The mining system incentivizes block production through payment and disincentivizes so-called "double-spend" attacks (wherein the attacker mints money for themselves at the expense of other users) by making it extremely costly to rewrite history. As a result, the Bitcoin blockchain has the property of being trustless. No central authority or existing relationship between users is needed in order for transactions to safely take place.

Blockchain technology can be extended to a decentralized exchange of assets by using smart contracts as defined in the Ethereum network [2] for trustless settlement (including delivery). However, matching engines are subject to additional trust considerations beyond double-spend attacks.

1.2 Matching Engines

Modern-day stock markets typically use price-time priority matching engines to determine the ordering of transactions.[1] When an order is sent to the marketplace, if there are any orders resting on its books which have a price better than or equal to the price indicated by the incoming order, the marketplace matches these resting orders with the incoming order. To determine which resting orders are matched with the incoming order, the resting orders are ranked first by price; if multiple resting orders have the same price then they are ranked according to which order was entered into the system first. A naive implementation of a decentralized exchange would encode the same matching engine logic into a smart contract, providing functions like `submitOrder` and `cancelOrder` and delegating price-time priority matching to the blockchain miners who execute the smart contract.

However, unlike ordinary electronic cash transfers, which rely minimally on ordering,[2] the ordering of transactions on exchanges nearly always has economic consequences for traders. A trader responding to offers (a liquidity-taker) will have an economic advantage if they trade first; later traders may find the liquidity supply exhausted and only higher priced assets remaining.

Miners or any other third party choosing the ordering of events can profit at the expense of other participants. They may put their favored liquidity providers at the front of the queue, execute their favored liquidity takers in front of other traders, or even front-run traders themselves. Moving the matching logic to an off-chain system of miners who specialize in matching orders simply moves these problems off-chain. Even if miners are all honest, the accuracy of time-stamping is limited by the network traversal time, and it is impossible to objectively determine who sent an order 'first'. Therefore, we consider it untenable for a decentralized exchange to be premised on delegating order flow handling to any third party, including specialized off-chain miners.

[1] There are other mechanics that marketplaces operate by. While this paper focuses on price-time priority matching engines because they are the most prevalent, the discussions generalize to other matching algorithms.

[2] Protection against double-spending requires only an ordered history of a particular coin, as opposed to system-wide chronological ordering.

1.3 Trustless Negotiation

One of the primary benefits of having a 'disinterested' third party (such as a blockchain miner or traditional trading facility) handling orders is to provide safety to the process by intermediating the trade negotiation, holding the buyer and seller to their respective ends of the bargain. In an unmediated negotiation, any party that extends an offer first gives the other party a "free option" [3] and an advantage in the trade:

- If the liquidity taker (the party acting on a bid/offer they have seen) has the final say in executing a trade, the liquidity provider gives the potential liquidity taker a free option by posting their liquidity. The liquidity taker is likely to hold on to the order—disregarding any requests to cancel the order—and only execute it if the exchange rate moves in the liquidity taker's favor.
- If the liquidity provider (the party posting their bid/offer first) has the final say in executing a trade, the liquidity taker gives the liquidity provider a free option when they try to act on the liquidity provider's quote. The liquidity provider is likely to renege on their quote unless the exchange rate moves in the liquidity provider's favor.

To reiterate, the issue with having a third party oversee trade execution is that they can be incentivized to favor either party (or another party, including themselves).[3] The design of a decentralized exchange must provide a solution to the negotiation problem outlined above without creating an opportunity for a third party to take advantage of traders.

1.4 Design Goals

Beacon's primary design goal is to enable decentralized exchange, providing safety to both parties during liquidity discovery, negotiation of terms, and settlement of assets. We define safety as follows:

- Safety to the liquidity provider. The liquidity provider faces a kind of information asymmetry when posting liquidity, since the liquidity taker can see their intent before making a decision. Beacon allows liquidity providers to cancel their orders without relying on an intermediary to carry out their instructions.
- Safety to the liquidity taker. The liquidity taker wants the liquidity provider to honor their quote, even if the market is moving against them. In Beacon, liquidity takers can verify that liquidity providers have a financial incentive to go through with the trade.

[3]The rules of execution may also implicitly favor one party over another. For instance, in the case of a price-time priority matching engine, liquidity takers are implicitly granted a free option because the liquidity provider needs to budget at least the latency of the matching engine in order to cancel a quote. The matching engine operator is also incentivized to ignore (or delay acting on) cancel requests because they are compensated based on volume transacted.

- Safety in settlement. Any exchange of assets happens atomically.

Beacon's secondary design goal is to allow traders to transact at a lower cost to attract liquidity and competition, which will ensure use of the system as well as accessibility to a wider audience. Beacon does not impose fees - the only fees which participants pay are transaction fees charged by the parent blockchain (and any fees agreed on by and between parties to a transaction). Additionally, Beacon allows parties to discover liquidity and negotiate off-chain, further saving on transaction costs.

2 System

The Beacon protocol provides a method for parties to mitigate the free option problem and negotiate without an intermediary. If at any point one party grants an option, they may simultaneously demand compensation for granting that option (making it no longer free).

More concretely, Beacon grants liquidity providers the final say over the execution of their quote, and allows the liquidity taker to receive compensation for granting the liquidity provider this option.[4] When a liquidity provider posts a quote (which we will refer to as an "Intent-to-Trade"), they set a forfeiture fee and a challenge period. If a liquidity provider makes an offer on the Intent-to-Trade (which we will refer to as a "Proposal-on-Intent"), and does not receive a response, they may challenge the liquidity provider. If the liquidity provider does not go through with the trade within the challenge period, they must compensate the liquidity taker by paying them the forfeiture fee. All terms are enforceable by a smart contract running on the settlement blockchain.

As we will see in Section 3.1 ("Forfeiture Fee and Challenge Period Selection"), forfeiture fee rates are influenced by supply and demand. Liquidity takers generally demand higher forfeiture fee rates, so quotations with low-or-zero forfeiture fees are rendered uncompetitive.

2.1 Topology

For the purposes of discussion, we will assume that messages are broadcast off-chain in a Kademlia [4] style network[5] (the "Beacon network") which will be used for message passing, routing, and node discovery. We will assume that

[4] We are investigating the possibility of allowing for alternative dynamics, such as giving the liquidity taker final say and having them conversely compensate the liquidity provider.

[5] A pure Kademlia implementation may introduce some risks; a node may not want to broadcast information until it becomes profitable to trade against it. Ideally, neighboring honest nodes would eventually pass along the information, but a coordinated malicious attack (such as an eclipse attack [5]) could partition the network. Initially, this can be addressed by creating a set of trusted 'anchor' nodes that any node can connect to, and which are guaranteed to transmit information honestly. Eventually, this system can be extended by creating a decentralized market for message passing, allowing node operators to receive compensation for passing information in a timely, honest manner.

the assets being traded are ERC20 tokens,[6] and that custody, settlement, and enforcement of transaction terms are governed by the "Beacon contract" run on the Ethereum network (the "parent blockchain"). Theoretically, it could run on any blockchain that supports tokens and enough scripting capability to implement the requisite escrow and arbitration functionality.[7]

2.2 Deposits

Users deposit tokens they wish to trade from their address on the parent blockchain into the Beacon contract by calling the `deposit` function. The Beacon contract maintains custody of user balances, and Beacon network participants are responsible for tracking counterparty off-chain transactions and on-chain account balances.

2.3 Standard Executions

Let us consider two traders on Beacon named Alice and Bob. Alice is a liquidity provider, and Bob is a liquidity taker. Both are assumed to be running a Beacon protocol compliant node, making them participants in the Beacon network. For the sake of simplicity, both are also assumed to have accounts on the same parent blockchain and settle their transactions on this shared chain.

In this example, Alice wishes to purchase 10 Token B at 123 Token A apiece (for a total of 1,230 Token A). To find a counterparty for this trade, Alice broadcasts an Intent-to-Trade (ITT) to the network with the following information:

Table 1: Intent-to-Trade (ITT)

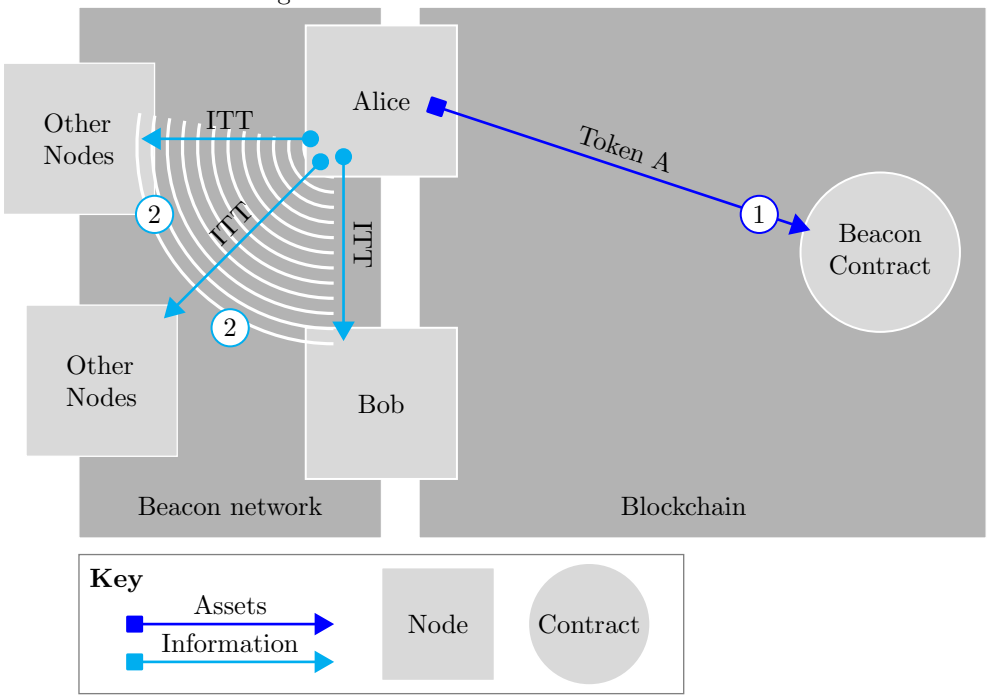
Section	**Information**
Party Information	Beacon contract address[8]
	Alice's network node ID
	Alice's Ethereum address
Terms	Have 1,230 Token A
	Want 10 Token B
	Willing to forfeit 1.23 Token A
	Challenge period is 317 blocks
	Partial fills not allowed[9]
Binder	Alice's nonce for this ITT
	Alice's digital signature

[6] The Beacon protocol and mechanics extend to non-fungible tokens like the ERC-721 token standard, but in this paper we will focus on the mechanics of trading fungible tokens.

[7] Or a pair of blockchains, so long as atomic swaps between these two chains are feasible and the aforementioned conditions are met.

Figure 1 below illustrates Alice broadcasting an ITT to the network.

Figure 1: Alice broadcasts her ITT.



1. Alice deposits some amount of Token A from her parent blockchain address into the Beacon contract.

2. Alice broadcasts an ITT on the Beacon network. The ITT tells prospective trading partners what exchange she would like to make, as well as the details of the forfeiture fee and challenge period. Bob and other nodes on the Beacon network receive the ITT.

Bob is interested in Alice's ITT, so he checks the parent blockchain to verify that Alice has enough funds to complete the trade. For now, let us assume that Alice is an honest participant and she has enough assets on deposit with the Beacon contract to cover the ITT. Bob sends Alice a Proposal-on-Intent (POI)[10] with the following information:

[8]This might not be a universal constant. For instance, the contract may be upgraded or the parties may want to use a custom smart contract to govern the negotiation.

[9]For the sake of simplicity, we will only consider ITTs which do not allow partial fills in the following examples. If Alice allows partial fills, all the operations described below can be executed against her ITT in a pro-rata manner. In the case of partial execution of a trade, the ITT will remain active and the forfeiture fee will remain in proportion to the remaining funds.

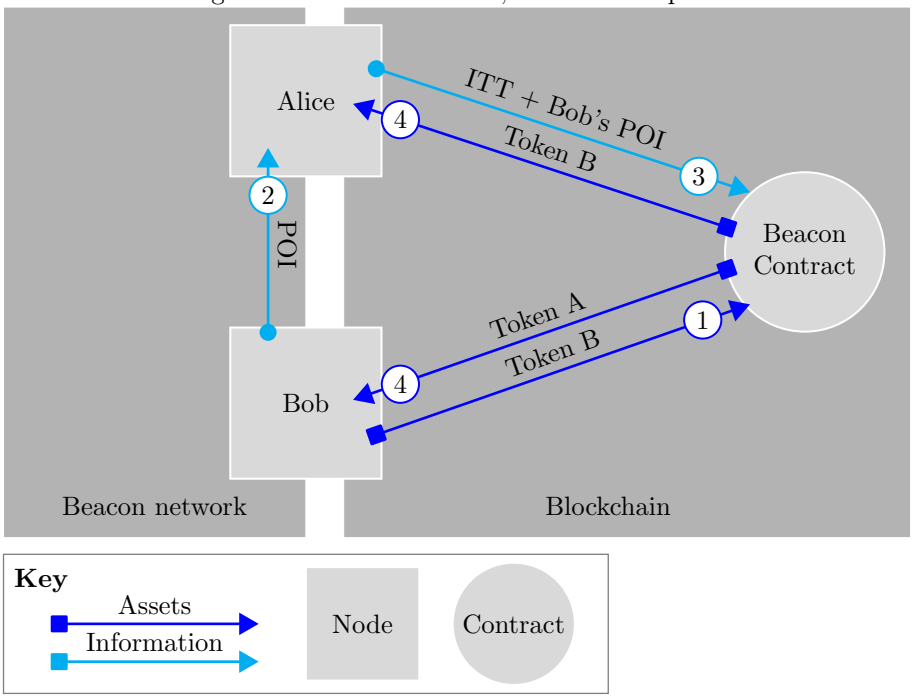
[10]To avoid being front-run, he should send this over a private, encrypted channel.

Table 2: Proposal-on-Intent (POI)

Section	**Information**
Party Information	Beacon contract address
	Bob's network node ID
	Bob's Ethereum address
Terms	Want 1,230 Token A
	Offer 10 Token B
Binder	Hash of Alice's ITT
	Bob's nonce for this POI
	Bob's digital signature

Figure 2 shows Bob offering Alice a POI and Alice accepting it, causing the transaction to go through.

Figure 2: Bob sends a POI, which is accepted.



1. Bob deposits an amount of Token B from his account (on the parent blockchain) into the Beacon contract.
2. Bob locates Alice's node on the Beacon network (using the node ID in her

ITT) and sends her a POI.

3. Alice accepts Bob's offer by submitting Bob's POI along with her ITT to the Beacon contract, calling the `acceptPOI` function.[11]

4. The assets (Token A and Token B) are atomically swapped. Bob's newly acquired Token A and Alice's newly acquired Token B remain in the Beacon contract and are available for withdrawal or use in other trades.

2.4 Challenging an ITT

Alice may not respond to Bob's initial POI for any number of reasons. (For instance, she might have been unaware of the POI, or may have been holding out for better opportunities.) Let us continue to assume that Alice is an honest participant and was simply unaware of Bob's POI, and she had every intention to follow through with her ITT had she been aware of Bob's POI.

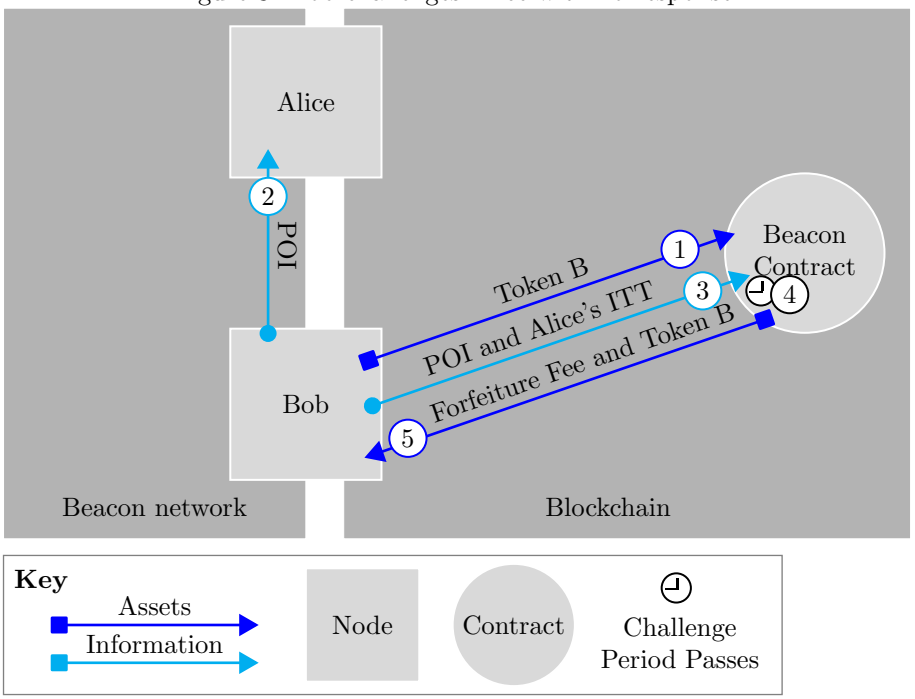
Bob may challenge Alice's ITT on-chain at any time.[12] Alice will then have until the end of the challenge period specified in her ITT to accept the POI. If Alice does not respond within the challenge period, Bob receives the forfeiture fee from Alice. To challenge Alice, Bob sends his POI and Alice's ITT to the Beacon contract, calling the `challengeITT` function.

In Figure 3, Bob challenges Alice and does not receive a response.

[11] To minimize the risk of a failed transaction, Alice should check that Bob has sufficient Token B on deposit prior to doing so.

[12] Since there is no way to prove that Alice received his off-chain POI, he can even skip sending a POI and move directly to challenging her on-chain. However, sending the POI off-chain will minimize transaction costs and expedite trade execution.

Figure 3: Bob challenges Alice with no response.



1. Bob deposits an amount of Token B from his account (on the parent blockchain) into the Beacon contract.

2. Optionally, Bob locates Alice's node on the Beacon network (using the node ID in her ITT) and sends her a POI.

3. Bob challenges Alice by submitting his POI with Alice's ITT to the Beacon contract and calling `challengeITT`. This locks up all of the assets relevant to the challenge for the duration of the challenge period specified in the ITT.

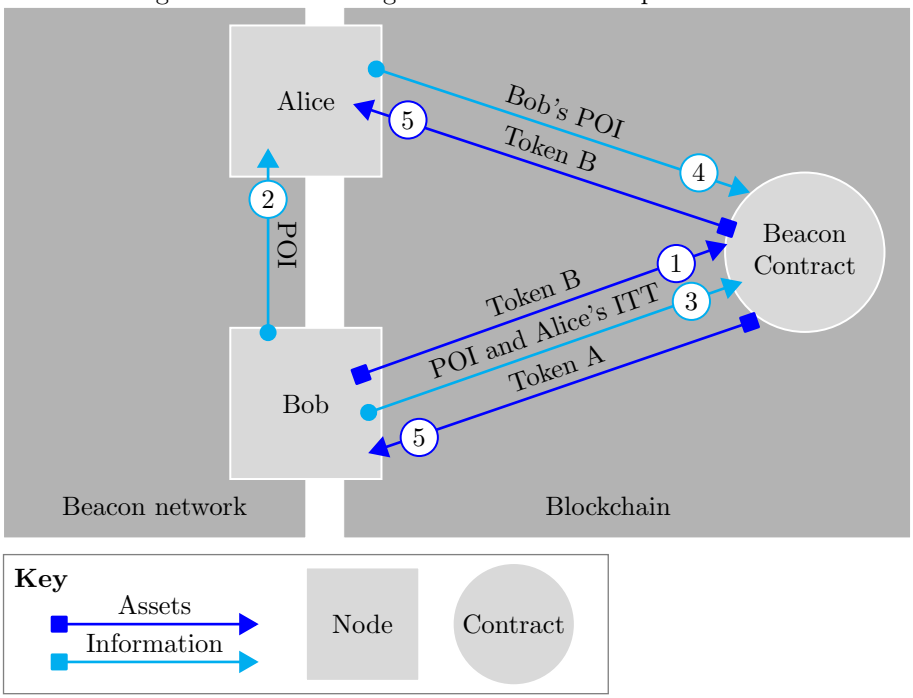
4. Alice doesn't respond within the challenge period. Locks on assets relevant to the ITT are lifted for both parties and the ITT is considered canceled.

5. The forfeiture fee amount is earmarked as being payable from Alice to Bob. Bob retains his Token B and Alice retains her Token A (less the forfeiture fee). All assets remain on deposit with the Beacon contract for withdrawal or use in other trades.

In the previous scenario, Alice might have been aware of the challenge, but allowed it to expire because she thought the benefits of intervening did not outweigh the transaction cost to do so.

Alternatively, Bob's challenge could make Alice decide to accept Bob's POI. In Figure 4, Alice accepts Bob's POI during the challenge period.

Figure 4: Bob challenges Alice and she accepts his POI.



1. Bob deposits an amount of Token B from his account (on the parent blockchain) into the Beacon contract.

2. Optionally, Bob locates Alice's node on the Beacon network (using the node ID in her ITT) and sends her a POI.

3. Bob challenges Alice by submitting his POI with Alice's ITT to the Beacon contract and calling `challengeITT`. This locks up all of the assets relevant to the challenge for the duration of the challenge period specified in the ITT.

4. Alice accepts Bob's offer by calling the `acceptPOI` function. Locks on assets relevant to the ITT are lifted for both parties.

5. The assets (Token A and Token B) are atomically swapped. Bob's newly acquired Token A and Alice's newly acquired Token B remain on deposit with the Beacon contract and are available for withdrawal or use in other trades.

2.5 Cancellations and Rejections

If Alice is present during the challenge period and doesn't like Bob's offer, she may cancel her ITT by submitting the ITT with her cancellation to the Beacon contract and calling `cancelITT`. This would effectively reject Bob's POI, award him the forfeiture fee, and free up Alice's remaining assets immediately. Alice's cancellation message contains the following information:

Table 3: ITT Cancellation

Section	**Information**
Binder	Cancel ITT with hash <Hash of Alice's ITT>
	Alice's digital signature

Meanwhile, the Beacon contract will not release Bob's assets; Bob is committed to waiting out the challenge period in order to 'earn' the forfeiture fee.

If Alice knows (prior to being challenged) that she no longer intends to trade on her ITT, she may broadcast a cancel message to the Beacon network. Any node can then claim Alice's forfeiture fee by submitting the ITT with Alice's cancellation to the Beacon contract and calling the `cancelITT` function.[13] By canceling in this way, Alice avoids two costs of being challenged: her Token A being locked up and the potential transaction cost of canceling during the challenge period to free up her locked assets sooner. Bob's benefit from submitting the cancellation instead of challenging is that although he must lock up his Token B for the duration of the challenge period, he is not subject to any exchange rate risk.

In the scenarios discussed thus far, Alice has become committed to losing her forfeiture fee as soon as she broadcasts her ITT. There are actually several ways Alice could try to liberate her forfeiture fee (although none are guaranteed):

- Create a sister account with sufficient Token B on deposit to submit Alice's ITT with her cancellation to the Beacon contract and collect the forfeiture fee. This account must execute ahead of any competition and pay the associated transaction costs. This strategy has the additional downside of locking up her Token B for the duration of the challenge period.

- Create an insufficient assets scenario by locking up enough of her assets in other ITTs that there isn't enough remaining in her account to cover the forfeiture fee. For this reason, well-behaving nodes should consider any ITTs which could overdraw Alice's account as invalid.

- Create an insufficient assets scenario by initiating a withdrawal on enough of her assets that there isn't enough remaining in her account to cover the

[13] Bob may still try to negotiate with Alice, but it is unlikely that she will accept since she suspects he already has her cancellation in hand. If she accepts, she risks Bob skipping out on the trade and collecting the forfeiture fee instead.

forfeiture fee, and make it through the withdrawal period before someone tries to collect the forfeiture fee. (Discussed further in Section 2.6.)

Because of these insufficient funds scenarios, Bob needs to remain attentive. If Bob does not see sufficient assets in Alice's account with the Beacon contract available for executing the ITT in full and he wants such a trade to go through, he should decide against submitting a POI on Alice's ITT and look for a similar ITT by someone more honest. On the other hand, if he sees sufficient assets for the ITT's forfeiture fee (but not for executing the ITT in full), he may decide to challenge Alice's ITT to claim the forfeiture fee. At worst, if Bob does not see sufficient assets even for the forfeiture fee, then there is no potential economic gain to be had by interacting with Alice.

2.6 Withdrawals

When Alice decides to remove her assets, she must first initiate a withdrawal period with the Beacon contract by calling `initiateWithdrawal`, earmarking the amount she wishes to withdraw. This function begins a span of time in which liquidity takers can collect forfeiture fees on outstanding ITTs. Once the withdrawal period is over,[14] Alice may call `completeWithdrawal` to withdraw whatever remains of the assets originally earmarked for withdrawal, after accounting for payouts.

3 Implications of the Beacon Model

Trading on Beacon has a number of properties which differ from traditional trading facilities. Here we discuss some of the salient features of trading on Beacon which might not be intuitive.

3.1 Forfeiture Fee and Challenge Period Selection

On a price-time priority exchange, speed has significant economic importance to traders. Liquidity providers can profit by buying at the bid and selling at the offer ahead of other market makers. Conversely, liquidity takers usually act on an information advantage, and need to submit their orders before potential counterparties can withdraw their liquidity.

Since Beacon ITTs are matched by liquidity takers (rather than a third party), posting liquidity first no longer has the same economic value. Submitting orders (or cancellations) quickly to act on an information advantage is similarly reduced in value. Liquidity providers have final say in the execution of their ITT, so they can take some extra time to avoid becoming a victim of *adverse selection* (the liquidity taker having access to more or newer information than

[14] The specific duration of the withdrawal period has not been established, but it should be related to the throughput of the parent blockchain to establish a fair opportunity for liquidity providers to hold Alice accountable on any open ITTs.

they do). In return, the liquidity taker is compensated for liquidity risk (the risk that they will not be able to execute at the price that they see) by the possibility of collecting the forfeiture fee. This liquidity risk can be decomposed into opportunity cost and exchange rate risk. A liquidity taker challenging an ITT takes on the opportunity cost of tying up their assets for the duration of the challenge period, as well as the exchange rate risk of the collateral changing in value during the challenge period. Therefore, to attract liquidity takers, a liquidity provider should post higher forfeiture fees. A higher fee signals that the liquidity provider is less likely to renege on their quote (and in the case that they do, Beacon turns this signal into economic reality by awarding the forfeiture fee to the liquidity taker), making a liquidity taker more likely to assume these risks.

In other words, all things being equal, a liquidity taker would prefer ITTs with a higher forfeiture fee and shorter challenge period, while a liquidity provider would prefer to supply a lower forfeiture fee and longer challenge period. In this way, the Beacon model incentivizes accurately setting forfeiture fees and challenge periods—as opposed to speed—as a primary element of competition between liquidity providers.[15] In the next section, we explore a framework for setting forfeiture fees.

3.2 Comparison with Standardized Options

To better understand the nature of forfeiture fees, let us look at the challenge procedure more closely. Suppose Bob challenges Alice. If the exchange rate moves in Alice's favor (this is a subjective judgment on Alice's part but might be based on a reference exchange rate from another marketplace) during the challenge period, she is likely to accept Bob's POI. However, if the exchange rate moves in Bob's favor, Alice is likely to allow the challenge period to expire and pay Bob the forfeiture fee.

As shown in the table below, by challenging Alice, Bob is effectively selling her a covered[16] American option. The option premium is equal to the forfeiture fee, and the option expiry is equal to the current time plus the challenge period. The strike price can be thought of as the quoted price less the premium (forfeiture fee). The table below shows Alice's balance sheet if she buys a standardized stock option, compared with an equivalent challenge scenario.

[15] As an example of the benefits of this kind of competition, this naturally disincentivizes spoofing. Spoofing occurs when traders submit quotes which are significantly worse than the best existing quotes, with the intention of canceling their quote once it is approaching execution in order to manipulate an asset's price. Spoofers are easily detected because of their low forfeiture fees, making it harder for them to manipulate the price of an asset.

[16] With standardized exchange-traded options, the option holder only needs to put up enough money to pay for the premium. This can result in the option holder and option seller each being leveraged, with the seller exposed to theoretically unlimited losses. In Beacon, each party to a challenge is holding the analog of a covered option because Beacon requires that both parties have the full underlying collateral on deposit. A clearing house is therefore not required because there is no delivery risk.

Table 4: Comparison to an American Option

Forfeiture Fee		**Buying an Option**	
Bob challenges Alice on an ITT offering 100 Token A for 20 Token B. Alice's forfeiture fee is 1 Token B.		Bob sells Alice an option allowing Alice to sell 100 Stock X for $21. The option costs Alice $1.	
a) Alice accepts Bob's offer.		a) Alice exercises her option.	
Token A: 0	**Token B: 0**	**Stock X: 0**	**Dollars: -1**
-100	**+20**	**-100**	**+21**
a) Total: -100	**Total: 20**	**b) Total: -100**	**Total: 20**
OR		OR	
b) Alice declines Bob's offer and pays the forfeiture fee.		b) Alice does not exercise her option.	
Token A: 0	**Token B: 0**	**Stock X: 0**	**Dollars: -1**
+0	**-1**	**+0**	**+0**
a) Total: 0	**Total: -1**	**b) Total: 0**	**Total: -1**

The parallel with standardized options means that the setting of forfeiture fees and challenge periods is amenable to using options pricing models. For instance, the Black-Scholes pricing model[17] implies that given the other parameters of an ITT, a prevailing interest rate, a reference exchange rate, and exchange rate volatility, a fair forfeiture fee can be calculated for that ITT.

This leads us to note an important property of trading on the Beacon network. On traditional exchanges, trading costs related to volatility can only be inferred based on aggregate behavior.[18] In Beacon, trading costs related to volatility are encoded directly in each ITT, as measured by the volatility implied by the forfeiture fee and challenge period.

4 Summary

We began by analyzing matching engines and determining that requiring transactions to be decided by a third party invariably risks the third party favoring one party to the transaction over the other. By framing the problem of decentralized exchange in terms of a negotiation framework, we identified

[17] While the Black-Scholes model has some overly simplified assumptions, it is a well-known model and the source of standard terminology such as *implied volatility*. Therefore, it provides a useful starting point for discussion.

[18] For example, [6] argues that bid-ask spreads are determined by volatility and the quote interval (the amount of time required before a quote can be amended). Determining bid-ask spreads requires multiple quotes, and on modern exchanges are usually set by multiple competing parties.

the root of the problem as what is known as the free option problem. By removing the third party and introducing competitive payment for options into the equation, we created a protocol for broadcasting liquidity and negotiating trades which is trustless and enables fully decentralized exchange of assets from quotation to settlement. Finally, by comparison to traditional options, we concluded that forfeiture fees are set competitively, and transaction costs related to volatility can be encoded directly in each transaction rather than inferred from aggregate behavior.

References

[1] Satoshi Nakamoto, 2009, *Bitcoin: A Peer-to-Peer Electronic Cash System*. Retrieved from `https://bitcoin.org/bitcoin.pdf` on May 6, 2018.

[2] Gavin Wood, 2014, *Ethereum: A Secure Decentralized Generalised Transaction Ledger*. Retrieved from `https://ethereum.github.io/yellowpaper/paper.pdf` on May 6, 2018, Byzantium Version *f72032b - 2018-05-04*.

[3] Joseph Poon and Vitalik Buterin, 2017, *Plasma: Scalable Autonomous Smart Contracts*. Retrieved from `http://plasma.io/plasma.pdf` on May 6, 2018.

[4] Petar Maymounkov and David Mazières, 2002, *Kademlia: A Peer-to-Peer Information System Based on the XOR Metric*. Retrieved from `https://pdos.csail.mit.edu/~petar/papers/maymounkov-kademlia-lncs.pdf` on May 6, 2018.

[5] Yuval Marcus, Ethan Heilman and Sharon Goldberg 2018, *Low-Resource Eclipse Attacks on Ethereum's Peer-to-Peer Network*. Retrieved from `https://eprint.iacr.org/2018/236.pdf` on October 3, 2018.

[6] Thomas Copeland and Dan Galai 1983, *Information Effects on the Bid-Ask Spread*. The Journal of Finance, 38(5), 1457-1469. doi:10.2307/2327580. Retrieved from `https://www.jstor.org/stable/2327580` on October 5, 2018.