Imperial College
London

IMPERIAL COLLEGE LONDON

DEPARTMENT OF COMPUTING

An Analysis of Decentralised Cryptocurrency Exchanges (DEX)

Author:
Mohammed Hussan

Supervisor:
Prof. William Knottenbelt

Submitted in partial fulfillment of the requirements for the MSc degree in Computing Science of Imperial College London

September 2018

Abstract

There has been an explosion in popularity of cryptocurrencies in past years leading to significant trading activity most of which has been concentrated on centralised exchanges. Centralised exchanges (CEX) are vulnerable to attacks leading to theft of crytocurrency placed in their custodian by users. Decentralised exchanges (DEX) have emerged, promising to protect users from theft and operate transparently.

This thesis presents a comprehensive and critical analysis of the major DEX projects. The landscape of DEX projects are surveyed to select the projects with the most development, adoption and variation of approaches. A total of 13 projects are analysed. Literature available of these projects, including whitepapers, websites and codebases are analysed. Primary information is gathered from engineering teams where required. Logical categories are defined in detail; Order Book, On-Chain Reserves and Peer-to-Peer Negotiations.

Analysis is focused on DEX security & decentralisation, performance & cost and usability. Security analysis considers liveness and safety failures, points of centralisation are determined and attacks such as front-running and maker griefing are demonstrated. Performance analysis outlines messaging overhead in the designs and where possible transactions costs have been presented. Software requirements of users and 3rd party platform operators are detailed. A discussion presents a summary of the defined DEX categories offering an overall assessment of the degree of decentralisation.

Acknowledgments

I acknowledge Prof. William Knottenbelt for offering this project and for his support.

A special acknowledgement goes to Alexei Zamyatin for his guidance and continued support throughout the project.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

The advent of Bitcoin in 2008, a peer-to-peer payments systems [6] introduced the blockchain, an append only list of records collected in blocks that are secured and linked cryptographically [7]. Altcoins similar to the bitcoin protocol have since been developed including Litecoin and Dash etc. Ethereum followed introducing a distributed computing platform featuring smart contract scripting. The platform serves many types of decentralised applications including tokens. At the time of writing 115704 ERC20 tokens are listed on Etherscan Token Tracker [8]

A secondary market has emerged to facilitate the exchange of cryptocurrencies and tokens for fiat currency. This market is dominated by centralised liquidity providers referred to as exchanges. Examples include OKEx, Binance, Bitfinex etc. These exchanges pose significant counterparty risk to users, there have been notable security breaches leading the theft of user funds including Mt. Gox [9] and Bitfinex [10]

There has been an emergence of so called decentralised exchanges (DEX) promising token exchanging services and cross-chain transactions whilst limiting counterparty risk. To date, no study and comparison of these protocols/systems has been conducted. This project will examine the veracity of the claims made by the most developed platforms with a focus on the trust models. The approaches taken will be categorised into logical groupings and a framework to compare the platforms on technical merit will also be presented. The analysis will be split into sections, a general sections will outline the categorisation, security and decentralisation will be considered followed by performance and cost, finally usability will be analysed.

Interoperability between separate blockchains is essential for cross-chain asset exchange, techniques to achieve interoperability will discussed. Technical aspects of the Bitcoin and Ethereum blockchains, relevant to DEX, will be presented for background. Centralised exchange models and their issues will be outlined.

Chapter 2

Background

2.1 Blockchain Interoperability

Blockchain interoperability is a field of research concerned with enabling disparate blockchain to communicate systematically without the involvement of a central party. In the context of decentralised cross-chain exchanges i.e a DEX that facilitates the exchange of BTC-ETH, interoperability between the Bitcoin and Ethereum blockchain is necessary. Below techniques to achieve interoperability have been summarised.

2.1.1 Atomic Swaps

Atomic swaps in the context of cryptocurrency exchange are under pined by the property of atomicity. Whereby one of two outcomes are only possible from the swap process, either both parties perform their transaction or neither do and the state of both parties is unchanged [11]. The atomic swap is attributed to TierNolan, a Bitcoin forum user, who proposed a protocol for a swap in 2013 [12]. The protocol is described by way of an example where Alice would like to exchange a number of Litecoin (LTC) for Bob's b number of Bitcoin (BTC). Both parties have the ability to verify the state of both chains respectively and have an off-chain medium to send each other prepared transactions to sign.

1. Alice is the initiator of the atomic swap, firstly Alice generates a transaction T_{x1}, which pays Bob a LTC if he can reveal a secret s. s is randomly generated by Alice and is kept secret. T_{x1} is encumbered by two conditions; 1) knowledge of Bob's private key and 2) knowledge of a secret which when input to a cryptographic hash function $H()$ is equal to $H(s)$. Alice does not broadcast T_{x1} to the network yet.

2. Alice also generates T_{x2}, a refund transaction. T_{x2} pays Alice a LTC using T_{x1} as an input but it cannot be spent for a time period t_1. T_{x1} is sent to Bob who signs and sends it to Alice. Alice now has a means to refund her original transaction T_{x1} once it is broadcast but can only do so after a time period t_1.

3. Alice broadcasts T_{x1}, committing her side of the atomic swap

4. Bob can inspect the Litcoin network to observe T_{x1}, once he is satisfied that it has received sufficient confirmations he generates a transaction T_{x3}. T_{x3} pays Alice b bitcoin if she can reveal a secret s. Notice that bob is not required to have knowledge of s but is aware of $H(s)$ which can be observed from T_{x1}.

5. Bob also generates a refund transaction T_{x4} similar to Alice but has a time period of t_2. t_2 is set to be less than t_1, the time difference between t_2 and t_1 must be such that it allows Bob enough time to spend T_{x1} on Bitcoin. T_{x4} is sent to Alice to sign and is returned to Bob.

6. Bob broadcasts T_{x3} to the network.

7. Alice generates T_{x5} spending T_{x3} in the process revealing s, Alice must do so before t_2.

8. Bob now can generate a transaction T_{x6} spending T_{x1} as the secret s has been revealed, he must do so before t_1 expires.

Each stage of the process is carefully constructed to avoid any race conditions and an outcome that favours one of the participant over the other. At stage 3, Alice has committed a LTC in T_{x1}, Bob may be unresponsive after this point. Alice has to wait till t_1 expires before she can broadcast the T_{x2} and effectively refund her initial expenditure once the transaction has been mined. Apart from the transactions fees and the time for which her coins were unavailable for use, Alice had suffered no significant loss. Similarly after 4, Bob can refund his original transaction T_{x3} using T_{x4} after a time period t_2. The most prominent issue with the protocol is that once Alice spends T_{x3}, Bob has a finite time to claim his coins on Litcoin. If he cannot do so he is at a loss and Alice can claim the LTC.

Atomic swaps are implemented using Hashed Timelock contracts in Bitcoin (see 2.2.3) and an implementation in Ethereum is trivial due to the its Turing complete smart contract programming language. As shown by the example, an atomic swap is an interactive process due to the timelocks involved. Transactions in the process are time sensitive and participants have to be available for the during of the process.

2.1.2 Chain Relays

Chain relays can effectively run a light client of a primary blockchain on a secondary blockchain, if the secondary has capabilities to validate the consensus algorithm of the primary. This gives the secondary blockchain the ability to inspect the state of the primary blockchain. An example implementation is BTCRelay [13]. BTC Relay allows Ethereum contracts to verify Bitcoin transactions without intermediaries. The relay contract stores Bitcoin block headers and uses the SPV method [14] to build a mini-version of the Bitcoin blockchain [15]. Chain relays can be used to perform non interactive exchanges demonstrated by XClaim, a protocol for issuing, trading and redeeming cryptocurrency backed tokens [16].

2.2 Bitcoin

Bitcoin is a peer-to-peer network of nodes where an append only blockchain of transactions is maintained. Bitcoins are generated and the blockchain is extended through block creation and mining where miners implement a Proof-of-Work algorithm [14]. Below relevant aspects of Bitcoin in relation to this thesis are detailed.

2.2.1 Unspent Transaction Output (UTXO)

An unspent transaction output is a fundamental building block of a bitcoin transaction, it is an indivisible chunk of bitcoin currency encumbered to a specific public key by a locking script. The bitcoin network tracks all UTXO. The concept of a user balance in Bitcoin is derived, it is calculated by scanning the blockchain and aggregating the UTXO belonging to the user. One or more UTXO are consumed in a transaction and in most cases created as outputs in a transaction [14].

2.2.2 Script

Bitcoin uses a traction script language, named Script. It is a Forth-like stack-based execution language. It was designed to be limited in scope as a deliberate security feature. There are no loops or complex flow control capabilities other than conditional flow control [14].

2.2.3 Bitcoin Opcodes

Operation codes or Opcodes are the Script language words used to generate bitcoin script transactions. Opcodes useful to interoperability are described.

OP_CheckLockTimeVerify (OP_CLTV) OP_CheckLockTimeVerify allows transaction outputs to be encumbered by a timelock. Transactions can define a parameter nLockTime which mandates a minimal time specified in either unix time or block height before a transaction can be included in a block. When the CLTV opcode is called it will only pass so long as the nLockTime of the transaction is less than the time parameter provided to the CLTV opcode.

Typical Transaction Scripts Popular transactions on the bitcoin network include Pay-to-Public-Key-Hash (P2PKH) and Pay-to-Script-Hash (P2SH) [14].

2-of-2 MultiSig Multi-signature scripts set a condition where N public keys are provided in the locking script and at least M must provide signatures to release the encumbrance. A 2-of-2 multisig is a script where 2 public keys are listed as signers and both must provide signatures to unlock the script [14].

Hashed TimeLock Contract (HTLC) HTLC are contracts that require the recipient of a payment to reveal the pre-image of a hash function, a secret, in order to spend the output before the expiration of a time-lock [17]. After the specified time-lock the output can no longer be spent by the recipient and is effectively refunded to the sender. This contract is central to achieving atomic swaps.

2.3 Ethereum

Ethereum can be described as deterministic state machine consisting of a globally accessible singleton state and a virtual machine called the EVM that applies changes to this state. It is effectively a decentralised computing infrastructure that executes programs called smart contracts. It uses a blockchain to store and synchronise the state of these smart contacts [18].

2.3.1 Turing Completeness

The EVM is Turing complete, allowing smart contracts that can be programmed to execute complicated logic including loops and complex flow control. This property provides flexibility that is useful for implementing interoperable systems. However, it does introduce complexity. For Turing complete systems you cannot predict whether a program will terminate without running it. Therefore they can run in infinite loops. As every node on the network must validate every transaction, this exposes the network to potential denial of service attacks. Ethereum uses gas 2.3.5, a mechanism to deal with this issue.

2.3.2 Smart Contracts

In the context of Ethereum, smart contracts are immutable and deterministic programs written in a high-level language compiled to EVM bytecode. Compiled contracts are deployed to the network via special contract creation transactions. Contracts on the Ethereum blockchain have their own addresses and are considered a type of account (2.3.3). Smart contracts are used to implement tokens on Ethereum (2.3.6). [18].

2.3.3 Accounts

Ethereum supports two types of accounts, both of which have an address and state.

Externally Owned Account (EOA) An EOA is an account that is held by a user, the state of the EOA is it's ether balance. An EOA can create transactions transferring ether or calling functions on smart contracts on the blockchain [19].

Contract Account Contracts are a collection of code (functions) that define how the contract state can be altered. The state of the contract is formed by its ether balance and storage; holding contract variables and data structures. EOAs can call the contract functions and contract accounts are able to pass messages between themselves [20].

2.3.4 Ether

Ether is the Ethereum blockchains native currency. it is used to pay for computation within the EVM indirectly by gas.

2.3.5 Gas

As the EVM executes a smart contract, all instructions such as computations and data access are accounted. Each instruction has a pre-defined units of gas based on its complexity. When a transaction triggers an execution of a smart contract function, it defines an upper limit of gas cost that it can consume. If this limit is reached before the completion of the transaction, the EVM will terminate execution and revert any state changes made. The gas cost does have an ether value which is paid by the transaction originator to the miner of the block where the transaction is included. Gas cost is calculated by multiplying the units of gas consumed by a transaction and a gas price. The gas price is a variable set by the originator when initiating a transaction, a higher gas price will encourage a miner to include the transaction in the block so to receive a higher reward [18].

2.3.6 ERC20 Tokens

The ERC20 token standard defines a common interface for contracts implementing a token so that any compatible token can be accessed and used is a common way. The interface expresses common functionality that must be included in every implementation of the standard and has optional functions that can be added by developers [18]. The interface can be viewed in the EIP [21]. Important functions are defined.

transferFrom(A, B, X) This function transfers ownership of X amount of the token from address A to address B. The originator of this transaction must be given approval to make this transaction by address A [21].

approve(A, X) The originator of this transaction provides address A permission to to call transferFrom for their tokens up to an amount X [21].

2.3.7 W-ETH

W-ETH or wrapped ETH is a representation of eth, the native Ethereum currency, that fits the ERC20 token standard. The act of 'wrapping' is effectively trading eth

for the W-ETH token. W-ETH is necessary where users want to trade eth against ERC20 tokens on some DEX designs [22].

2.4 Centralised Exchanges (CEX)

Centralised exchanges are the most common way to trade cryptocurrencies, this includes buying and selling cryptocurrencies for fiat as well as one cryptocurrency for another. Typically, they are online or mobile platforms that require users to sign up as customers.

2.4.1 CEX Architecture

Internal architectures of these exchanges are not readily available however a simple model can be inferred [23]. Figure 2.1 shows a generic architecture overview of a centralised exchange. Firstly, when users sign up with an exchange, they can either purchase some cryptocurrency with fiat or transfer ownership of existing cryptocurrency to the exchange, this is done by a transaction on-chain. Users interact with the exchange via a web or mobile app, where they can trade their cryptocurrencies, under the management of the exchange operators, against liquidity provided by the operator or other traders. Buy/sell orders placed by traders and changes in balances are maintained in a database. The on-chain transactions in the system typically represent users withdrawing funds from the exchange, the operator will transfer cryptocurrency from their address to one which is owned by the trader.

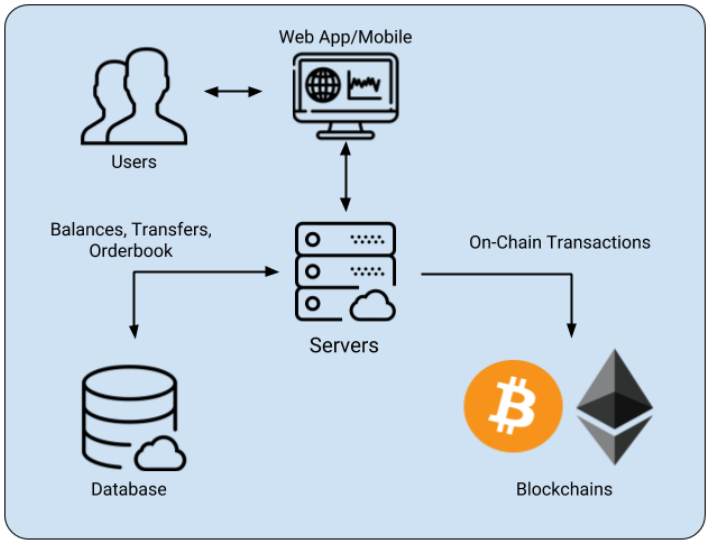


Figure 2.1: Centralised Exchange Architecture

2.4.2 Exchange Model

Centralised cryptocurrency exchanges follow the model of traditional exchanges in financial markets. Below, key terms related to exchanges are defined.

Order An order from a trader signals a commitment to buy or sell an asset at a given price and volume. Orders fall in to different types that affect the price and the time that the order can be executed [24].

Limit Order A limit order is a buy order with a price ceiling, therefore the trader is willing to buy an asset for a price below a threshold [25].

Maker A maker generates a buy or sell order. They offer to either buy or sell a given asset at a given price and volume [26].

Taker A taker opts to accept an order generated by a maker by either buying or selling a given asset at offered price and volume.

Order Book Order books maintains a list of buy and sell orders for a given asset, traders can view the order book and decide to fill orders. [27]

Liquidity Liquidity describes the degree to which an asset can be readily bough or sold in a market without affecting its price. In the context of an order book, a liquid asset would have many sell orders with respect to buy orders making it easy for a trader to buy the asset [28].

2.4.3 CEX Issues

Custodian of funds Typically centralised exchanges require users to transfer over possession of cryptocurrencies they wish to exchange to the exchange operator. In this state, users are not in control of their assets and must trust the operator to behave fairly and return their assets when they choose to withdraw. There have been numerous cases where exchanges have claimed to hacked and user funds held in exchange accounts have been stolen [29]. In many of these cases, users have no recourse and have to rely on good faith from the exchange to make up their lost funds where possible.

Restricted Listing The major exchanges typically focus on a handful of the major cryptocurrencies and tokens, offering high and concentrated liquidity in those markets. CEX have relatively slow processes for listing tokens, this has been apparent with the popularity of ERC20 tokens.

Privacy Due to regulation in many regions, exchanges requires customers to do KYC. In this process, customers may be asked to provide a photo of a government ID e.g. a passport or a driver's license and other personal information.

Chapter 3

Related Work

There is a lack of academic research focused on surveying the landscape of the projects on the decentralised exchanges from a technical perspective. There have been a handful of attempts, Michael Borkowski et al. presented an overview of projects implementing atomic cross-chain token transfers, a descriptive approach is taken with a lack of detailed and thoughtful categorisation [30]. Peter Bennink presents a study analysing methods that execute atomic swaps on the Ethereum blockchain, both single and cross-chain swaps are considered. A brief analysis of two DEX projects is presented [31].

Academic research into specific approaches to achieve interoperability leading to potential application in decentralised exchanges can be found in the literature. Tesseract is presented a secure real-time cryptocurrency exchange service. Issues with theft of funds, trade latency and frontrunning are tackled using Intel SGX as a trusted execution environment [32]. XClaim is presented as a protocol for issuing, trading and redeeming Bitcoin-backed tokens on Ethereum [16].

There is however a rich source of non-academic writing and analysis on DEX from the cryptocurrency and blockchain community [33] [34] [35]. The Web3 Foundation have produced a report after a workshop involving developers from notable projects in the DEX space [36]. The workshop focused on challenges in the space including front running prevention, order book unlinkability, throughput, dark pools and cross-chain exchanges. The developers present, analysed their projects against a descriptive framework and presented an unstructured and incomplete table [37]. Mansi Prakash presents the most comprehensive community analysis in a blog post [38]. DEX projects have also produced blogs analysing the DEX space with a focus why their projects compare favourably to competition [39] [40].

Chapter 4

DEX Analysis: General

Decentralised Exchanges (DEX) are a class of platforms that enable traders to buy and sell cryptocurrencies whilst limiting trust assumptions required in the platform. DEX aim to tackle the issues of centralised exchanges outlined 2.4.3. A range of projects have attempted to develop DEX, taking differing approaches. In this chapter a general analysis of the DEX is presented. The different DEX categories are described in detail with examples of notable projects provided. In general, the projects selected for analysis fulfil the following requirements. The projects have detailed explanation and documentation supplied in a whitepaper or on a website, an open source codebase and have been deployed to live blockchain networks.

Many of the projects that have been proposed and are discussed have focused on DEX that allow for the trading of ERC20 tokens on Ethereum (see 2.3.6), projects have focused on this problem space as it creates a market that is not served by the major exchanges and it is a simpler problem when compared to a cross-chain exchange. Nevertheless, there are projects that have attempted a cross-chain exchange, these two classes of exchanges will be considered separately.

4.1 DEX Protocol Roles

The DEX approaches define roles within their systems that are taken up by individual users or 3rd parties. Each project defines their own terms to represent these roles, below general terms and an abbreviations that are representative of such roles are defined.

Operator (O) Most of the DEX have a party that is hosting part of the DEX infrastructure, typically this is the organisation behind the DEX. The parties are responsible for keeping ths system live.

Maker (M) A maker is a participant on a DEX that has an Order book (2.4.2), they generate either buy or sell orders (2.4.2)

Taker (T) A taker moves to fill an order generated by a maker. In DEX that do not have an order book, a taker is considered the party who wants to buy or sell an asset to the DEX.

Relayer (RL) A Relayer is responsible for relaying signed orders off-chain, making them available to be filled.

Signing Relayer (SRL) Signing relayers are similar to relayers, except they are required to sign the orders.

Liquidity Provider (LP) Liquidity providers participate with DEX by providing liquidity to the platform, they are market makers. They serve a similar function to a maker except a significant volumes.

4.2 DEX Category: Order Books

This class of DEX implements an order book similar to those of traditional exchange platforms described in 2.4.2. They typically include makers that create buy or sell orders that populate order books, takers then can decide to the fill the orders and initiate the trade. Figure 4.1 displays a categorisation of order book DEX.

In this context, an off-chain order represents order books that are maintained off-chain, typically on servers managed by the DEX operator or distributed amongst nodes on a peer-to-peer network. An on-chain order book is one maintained in a smart contract on the Ethereum blockchain.

Settlement relates the balance transfer finality, on-chain settlement occurs as a transaction transferring ether or ownership of token on the Ethereum blockchain. Off-chain settlement occurs by either balance management on DEX operator servers or by trade execution on sidechains.

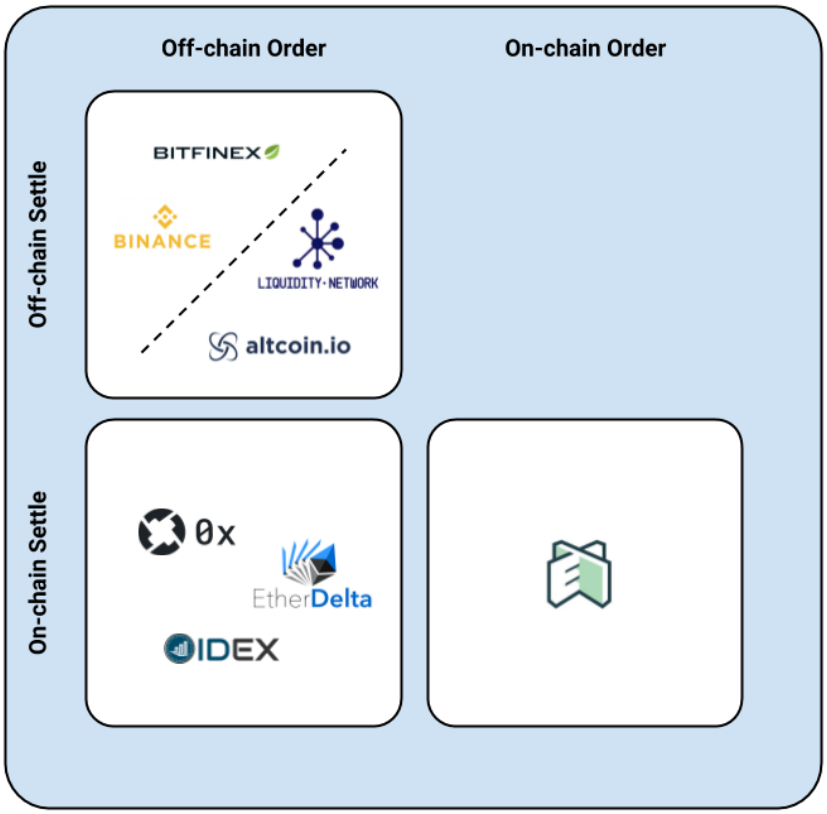


Figure 4.1: Order book DEX comparison, projects categorised by on or off chain order book versus on or off chain settlement

4.2.1 Off-Chain Order/ On-Chain Settlement

This group of order book DEX take advantage of an off-chain order book. Order book management when maintained on-chain requires blockchain transactions to make and cancel orders. Transactions on the network require a finite time to be mined resulting in a delay and poor trading experience. The on-chain transactions also have an associated gas cost (2.3.5), on-chain order books therefore place a cost barrier for market makers which is undesirable for maintaining liquidity. Moving the order book off-chain however results in additional trust requirements for the users, DEX operators have a measure of control over the order book which can be exploited to the benefit of the operator (see front running 5.3). Below the key DEX in this category are described.

0x Protocol 0x is an open permissionless protocol allowing ERC20 tokens to be traded on the Ethereum blockchain. The 0x team have developed smart contracts where a taker can fill signed orders from a maker, the orders are authenticated on-chain [1].

0x defines a message format for an order containing order parameters that are concatenated and hashed to 32 bytes via the Keccak SHA3 function. The order is then signed by the maker using their private key to produce a ECDSA signature [1].

The protocol defines the position of a Relayer, tasked with broadcasting orders and maintaining order books off-chain whilst taking a fee for facilitated transactions, any party can adopt the 0x protocol and become a relayer [1]. Figure 4.2 shows the generic architecture of the 0x protocol, the steps in the protocol are numbered and described.

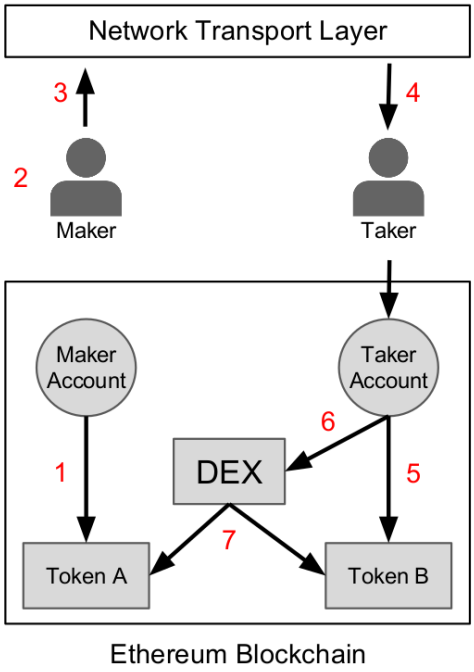


Figure 4.2: 0x Protocol Architecture as presented in [1]

1. Maker approves transferFrom (2.3.6) method for DEX contract
2. Maker creates and signs an order exchanging Token A for B
3. Maker broadcasts order through communication medium
4. Taker discovers the order and decides to fill it
5. Taker approves the transferFrom method for DEX contract
6. Taker submits the signed order to DEX contract

7. DEX contract checks order is valid, all requirements are met and transfers tokens

The network transport layer shown in Figure 4.2 can be any communication layer where a maker and taker can agree on a trade and produce an order that satisfies the 0x order message format. As mentioned, this role is filled by a relayer. 0x allows relayers some flexibility over their implementation. Relayer strategies including Open Order Books and Order Matching are described.

Open Order Books An open order book uses limit orders that are not assigned to a specific counter party i.e when an order is generated a taker is not set. This means that the Exchange smart contract accepts these orders from any Ethereum address acting as a taker, on a first come first serve basis [41]. Figure 4.3 demonstrates the open order book strategy with stages in the design numbered and described.

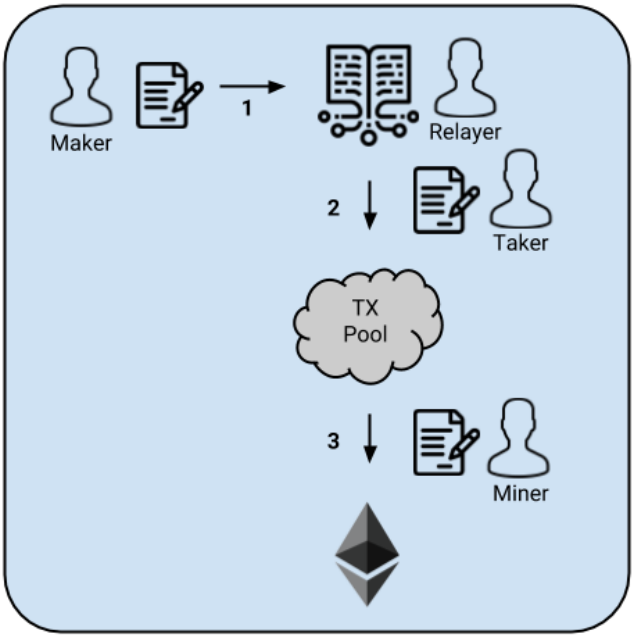


Figure 4.3: Open Order Book Relayer Strategy

1. Maker generates a signed order and places it with a relayer who adds it to the order book
2. Taker selects the order, filling it and submitting it to the 0x contract
3. Miner selects transaction mining it and adding it to the blockchain

The relayer in this case will monitor transactions filling order that are submitted and mined, so to prune the order book of filled orders.

Order Matching In this strategy, a relayer hosts a publicly viewable order book of signed orders. However, the taker parameter for each order is set to an EOA that is controlled by the relayer. Therefore, the relayer is the only party capable of filling the orders, eliminating the potential of front-running. When orders on both sides of the market overlap in price, the relayer batch fills these orders simultaneously, matching two or more parties together [41]. Figure 4.4 shows the order matching strategy with stages in the design numbered and described.

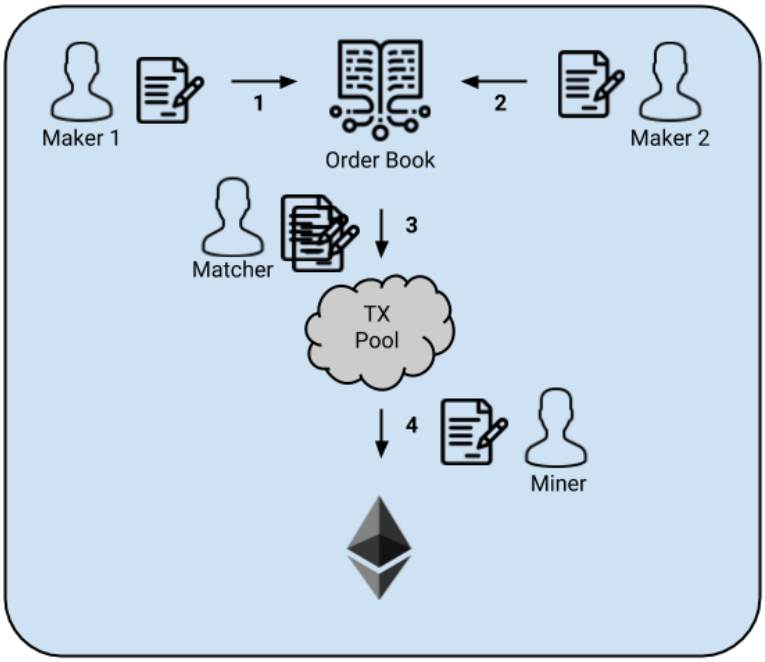


Figure 4.4: Order matching relayer Strategy

1. Maker 1 submits a buy order.
2. Maker 2 submits a sell order on the opposite side to the Maker 1 order, overlapping in price.
3. Matcher batch fills orders simultaneously
4. Miner selects transaction mining it and adding it to the blockchain

In effect, there are no takers in this strategy and the relayer is required to place a transaction on the network. This results in the relayer having to pay the gas cost and then charge the the users later resulting in difficulty.

Below, some of the largest 0x relayers are described in more detail.

DDEX The DDEX relay has the largest market share of all 0x relayers according to Etherscan at 4.9%. It follows the Order Matching relayer strategy described above [42].

Radar Relay Radar Relay was one of the first relays to be implemented using the 0x protocol, it follows the Open order book relayer strategy [43].

IDEX IDEX has the largest market share of all the ERC20 Token DEX. IDEX is made of a smart contract, trading engine and a transaction processing arbiter. Users are required to transfer ether or tokens they wish to exchange to the IDEX contract for management via a deposit transaction [2].

An order book along with user's balance are maintained off-chain, both are updated in real-time. Only the IDEX operator submits trades to the smart contract allowing IDEX to control the order that transactions are processed avoiding issues with race condition and front running described here 5.4 and here 5.3.

Trades are signed by users private keys and are verified on-chain. The authorisation prevents users form rescinding trades whilst also preventing IDEX from generating unauthorised orders [2]. However, this gives the IDEX operator the ability to censor, delay or manipulated the order of orders agreed by makers and takers. Figure 4.5 shows the IDEX ecosystem in detail.

1. Maker and Taker deposit tokens into IDEX contract
2. IDEX off-chain database is updated to include addresses and balances
3. Maker creates and submits a signed order
4. IDEX confirms sufficient funds
5. Order is added to order book
6. Taker submits matching order to fill the makers order at the same price level and a suitable volume.
7. IDEX confirms sufficient funds
8. The order book is updated and the order is marked as matched
9. IDEX updates database to reflect new balances, trader can continue to trade based on this. The trade transaction is added to the queue to be broadcast on Ethereum
10. After all dependent transactions have been mined, the transaction is broadcast to the network
11. After the transactions is mined, the maker and taker can withdraw funds

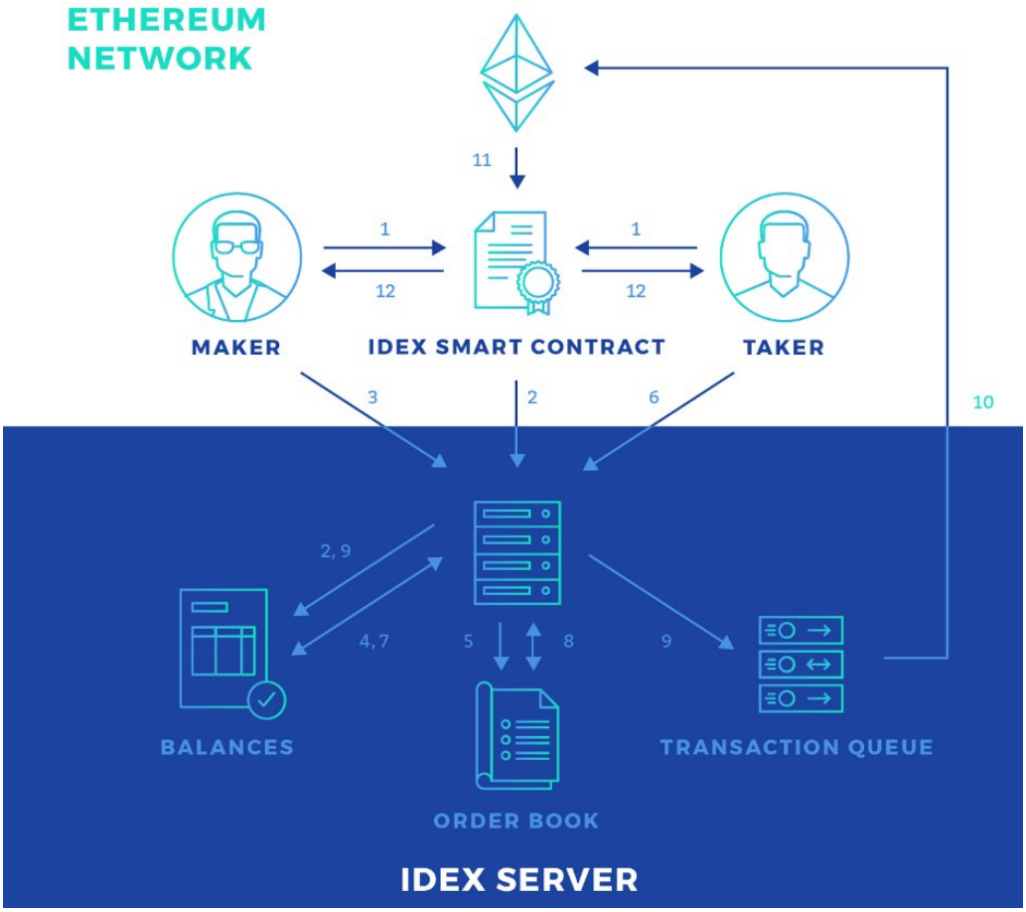


Figure 4.5: IDEX Architecture shown in the IDEX Whitepaper [2]

DEx.top DEx.top follows a near identical model to IDEX, users however must register an account in order to make trades [44].

EtherDelta EtherDelta are one of the first exchanges that were branded as decentralised and it opened up a market for ERC20 tokens. It operates as a open order book style DEX equivalent to the open order book relayer strategy [45].

4.2.2 On-Chain Order/ On-Chain Settlement

In general, on chain order books result in poor user experience as all orders have to made on-chain. This results in significant operational latency. Furthermore, this discourages makers as there is a gas cost associated to making markets having an adverse effect on liquidity. Nevertheless, there are projects that are attempting to implement on-chain order books.

Ethex The Ethex project believes that the latency related to an on-chain order book is a price worth paying due to its benefits. With an op-chain order book, a point-of-centralisation is removed from the system as the blockchain is entrusted to maintain

the order book infrastructure as opposed to a DEX operator. In some cases the number of on-chain transactions required to settle a position is less in an on-chain order book design [46].

4.2.3 Off-Chain Order/ Off-Chain Settlement

The majority of centralised exchanges fall into this category. They benefit from not having to initiate any on chain transactions for every trade allowing for very quick and cheap trading, however they have custodian of user funds which puts their funds at risk. For completeness, large centralised exchanges Binance and Bitfinex are shown in Figure 4.1.

There is another emerging group of DEX designs where trade settlement can be classed as off-chain. Altcoin present an idea to achieve settlement off-chain using sidechains based on the Plasma whitepaper [47]. An explanation is presented in a blog post [48]. The Liquidity Network present an idea based on off-chain settlement in NOCUST [49]. Both offerings are conceptual and require significant development before implementation.

4.3 DEX Category: On-Chain Reserves

This category of DEX aims to hold liquidity within a smart contract and make it available to users who want to place a trade against the liquidity. Trades require a single transaction and there are no order books. Mechanisms are in place for maintaining liquidity and for fairly pricing trades. A detailed explanation for this category of DEX is presented by way of example. KyberSwap and Bancor Network are described below.

4.3.1 KyberSwap (KyberNetwork)

KyberSwap is an on-chain decentralised exchange where users can instantly convert tokens as a means of exchange or to pay another user in a different token. There are no order books and users need to complete a single on-chain transaction to make a trade.

There are several important participants in the system described below.

1. User: participants who send and receive tokens to and from the network
2. Reserve: Provides liquidity to the network. Kyber Network have their own reserve and third party reserves can register. These third parties are classed as liquidity providers 4.1 described before. Reserves can be public or private, placing restriction on contributors.
   (a) Reserve Manager: Tasked with managing a reserve, setting an exchange for token pairs in management, feed the rates to the KyberNetwork
   (b) Reserve Contributor: Can provide capital to public reserves and share in profits.
3. KyberNetwork Operator: Responsible for adding and removing reserves from the network, listing and delisting token pairs. This role is currently taken by the KyberNetwork.

Figure 4.6 shows the architecture of the KyberSwap system. The KyberNetwork contract is central to the system, users can query the contract to receive the exchange rate for a given token pair. The KyberNetwork contract will fetch all exchange rates for the token pair from reserve managers and return the best. If users are satisfied with the rate on offer, they can execute a trade, the KyberContract will execute the trade based on the best rate at the time.

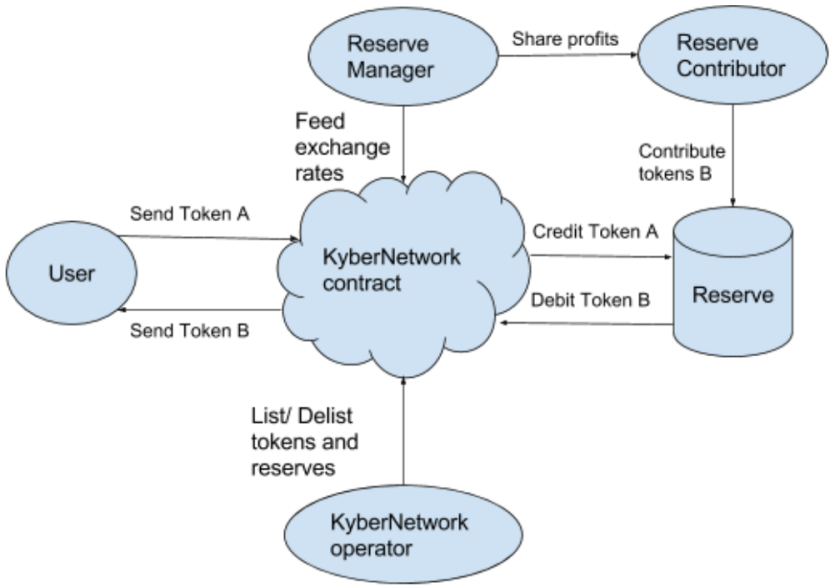


Figure 4.6: KyberSwap Architecture introduced in the KyberNetwork whitepaper [3]

4.3.2 Bancor Network

The Bancor protocol outlines automatic price determination and liquidity mechanism for tokens on smart contract blockchains. The protocol introduces Smart Tokens that are connected to one or more reserves of other tokens. Users are able to instantly purchase or liquidate a supported smart token against any of its connected tokens via the smart contract. The price is determined by a formula that continuously balances buy and sell volumes [50].

Smart Token Smart Tokens operate as regular ERC20 tokens but with additional logic allowing users to buy and sell the token through its own contract. An example is the BNT Smart Token which has a single connection to a reserve of ETH. Buyers can purchase BNT by sending ETH to the contract, this ETH is added to the reserve and BNT is issued to the sender, both the supply of BNT and the ETH reserve has increased. Sellers of BNT will send BNT to the contract, the BNT will be burned and ETH will be sent to the sender, in this case both the supply of BNT and reserve ETH balance has decreased [50]

To determine the amount to issue a buyer or withdraw for a seller, the Smart Token continuously recalculates its price against each of its connected tokens. The Bancor Formula does so by maintaining a fixed ratio, weight, between the value of the Smart Token and the value of its connector balances [50]. Smart Tokens therefore have a built-in liquidity mechanism and can be considered automated market makers.

Pricing Algorithm Bancor presents a formula for algorithmic pricing, it is built on the idea that each Smart Token maintains a ratio between its total value and its con-

nector balance. This ratio is a the connector weight or CW

$$CW = \frac{connector\ balance}{Smart\ Token's\ total\ value}$$

The Smart Tokens's total value is define as

$$Smart\ Token's\ total\ value = price \times Smart\ Token\ supply$$

The price is then determined as

$$price = \frac{connector\ balance}{Smart\ Token's\ total\ value \times CW}$$

For more detail on supply demand analysis at different connector weights and price slippage handling, refer to the Bancor whitepaper [50].

4.4 DEX Category: Peer-to-Peer

The Peer-to-Peer DEX design aims to pair makers and takers on a network based on intent to trade a token pair. Once peers are paired, a negotiation protocol is executed where a maker and taker decide on a price and volume. If an agreement is reached, an on chain transaction is placed to settle the trade. The most substantial implementation of a Peer-to-Peer DEX is AirSwap, which is described below.

4.4.1 AirSwap

AirSwap's Peer-to-Peer DEX is based on the swap protocol presented by Micheal Oved and Don Mosites. The Peer-to-Peer design is favoured over order book style DEX due to key advantages; scalability and privacy. Peer-to-Peer trading scales in the sense that orders are negotiated and filled in a 'one and done' fashion. Order book typically have many unfilled orders which expire or are cancelled. Peers negotiate price and volume in private, this removes any opportunity for other parties to act on order request behaviour [4].

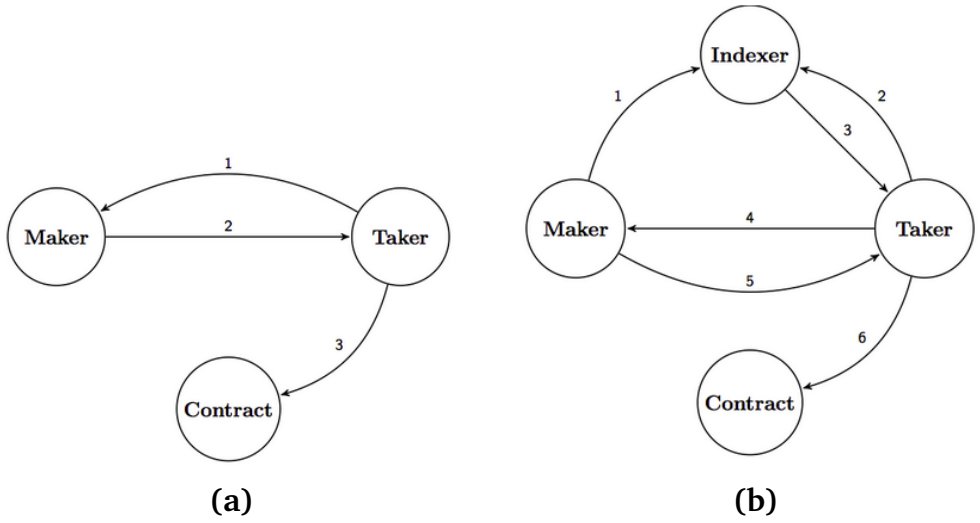


Figure 4.7: (a) AirSwap Peer Protocol diagram, **(b)** AirSwap Indexer Protocol diagram sourced from Swap white paper [4]

The Swap protocol defines two sub protocols; the Peer Protocol and the Indexer Protocol.

Peer Protocol The Peer protocol is shown in Figure 4.7 (a).

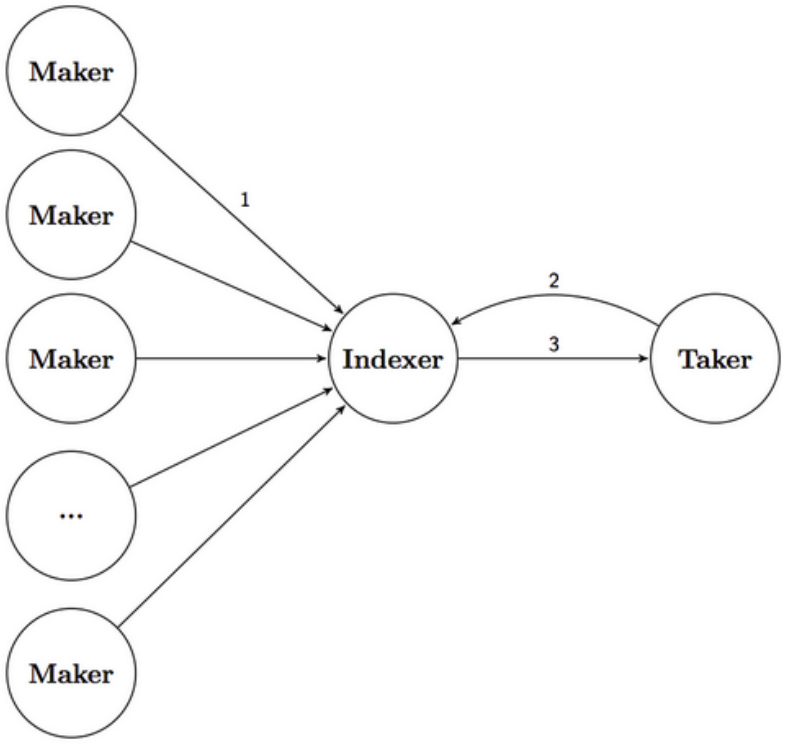
1. A Taker requests an order from a maker, specifying a token pair and the volume of the token they would like to receive
2. A Maker replies with a signed order specifying the opposing token volume and order expiration
3. A Taker can fill order on the contract if they are happy with rate

Indexer Protocol An Indexer is an off-chain service that matches peers based on intent to trade a token pair. AirSwap provides this Indexer service to peers on the exchange. Prospective makers signal an intent to trade a token pair to the indexer, takers ask the Indexer to be paired with an appropriate maker, there is likely to be multiple results. A taker can then proceed to negotiate with each maker using the Peer protocol described above a place and trade on-chain if they wish. The Indexer protocol for the simplest case is shown in Figure 4.7 (b).

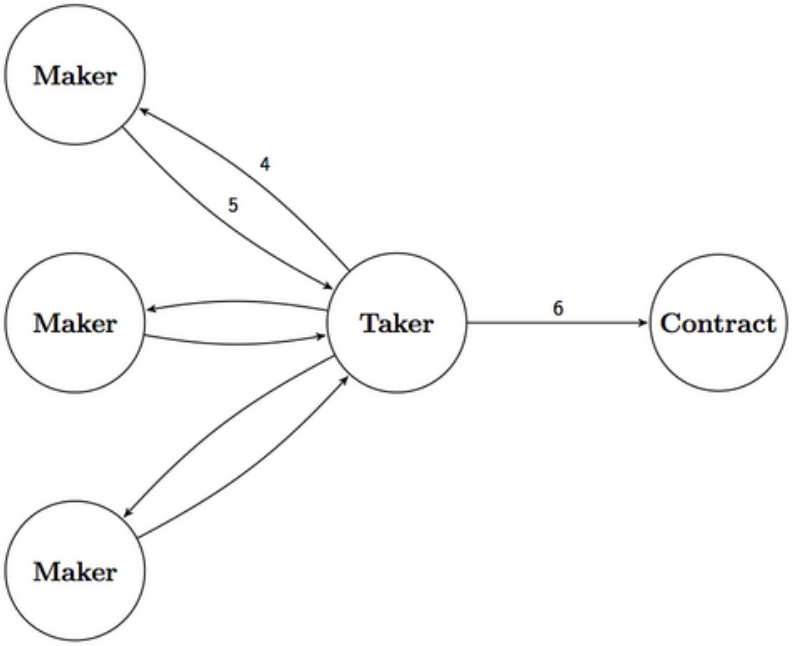
1. A Maker signals intent to trade a token pair to the Indexer
2. A Taker asks the Indexer for suitable peers to trade with
3. The Indexer returns a suitable Maker to the Taker
4. The Taker request an order from the Maker
5. The Maker replies with an order
6. The Taker places the order on-chain

Interactions where there are several Makers and a single Taker are shown in Figure 4.8 (a) and (b).

1. Several Makers signal intent to trade a token pair
2. A Taker asks the Indexer for suitable peers to trade with
3. The Indexer returns suitable Makers by address to the Taker
4. The Taker requests orders from several Makers
5. Makers reply with orders
6. The Taker selects an order and places it on-chain



(a)



(b)

Figure 4.8: **(a)** AirSwap Indexer Protocol diagram, with multiple Makers **(b)** AirSwap Peer Protocol diagram with multiple Makers sourced from Swap white paper [4]

4.5 DEX Category: Cross-Chain

The Cross-Chain category is not an exclusive DEX category. Operationally, DEX that are Cross-Chain can behave as an Order Book, On-chain Reserve or Peer-to-Peer DEX. However, a Cross-Chain DEX must facilitate a cross-chain exchange, where crypto assets can be traded across multiple different blockchains i.e a trade between Bitcoin and Ether or ERC20 tokens. barterDEX and BlockDX are the two cross-chain DEX that this study will analyse, the two projects have the most product development and documentation outlining their designs.

4.5.1 barterDEX (Komodo)

barterDEX is a DEX project developed by the Komodo team. It consists of decentralised order matching, trade clearing and settlement processes. Order matching is done via a low-level pubkey-to-pubkey messaging protocol and final settlement in executed through an atomic swap protocol [51].

Users are required to install the barterDEX wallet. Users have the ability to exchange whitelisted tokens, which include BTC-like coins that are based on UTXOs and the Bitcoin Core protocol and ETH/ERC20 tokens [52].

Komodo The Komodo technology is a fork of ZCash, integrating a delayed-proof-of-work consensus algorithm [51]. In delayed-proof-of-work, the Komodo ecosystem is notarised on the Bitcoin blockchain via a Bitcoin transaction. Komodo define notary nodes responsible for securing the chain, a voting process is in place to delegate the notarisation responsibilities. The delayed-proof-of-work whitepaper provides more detail [53].

Komodo is designed to allow users to spin up and manage their own blockchains which they can choose to connect with the Komodo ecosystem.

Order Matching The order matching is done via custom peer-to-peer network that employs two separate types of nodes: a full-relay node and a non-relay node

1. Full-relay Node: high- volume trader who provides liquidity to the network in exchange for being a trading hub on the network, considered an **LP** in relation to DEX roles

2. Non-relay Node: common user, who engages with barterDEX when trading

A Non-relay nodes signal their intent to trade a token pair, this signal is sent to a random selection of active Full-relay nodes on the network. Once connected, the two nodes negotiate a price and volume for the trade based on their UTXOs. If an agreement is reached i.e both parties are in possessions of UTXOs that satisfy the price and volume for the trade, the execution proceeds to trade settlement [52].

Trade Settlement Up to this stage the trade protocol has remained consistent for a BTC-like to BTC-like and BTC-like to ETH trades. The settlement processes differ and will be explained.

BTC-like to BTC-like The atomic swap process is similar to the process defined in 2.1.1. The process use hashed timelock contracts and refund transactions. Alice and Bob are used to demonstrate the execution of the swap.

1. Alice sends dexfee
2. Bob sends bobdeposit
3. Alice sends alicepayment
4. Bob sends bobpayment
5. Alice spends the bobpayment
6. Bob spends the alicepayment
7. Bob refunds his own deposit

BTC-like to ETH The BTC-like to ETH is supported by using a Komodo based ETOMIC token which act as a proxy for ETH/ERC20 tokens. A smart contract is deployed on the ethereum mainnet to hold ETH deposits and enable payments between the parties [54]. A BTC-like to ETOMIC swap occurs using the barterDEX wallet, the tx information of the ETOMIC deposit transactions and the deposit parameters are used to deposit the tokens on the Ethereum blockchain. Throughout the process the ETOMIC transaction are mirrored on the Ethereum blockchain with the same parameters. Converstions with Artem Pikulin, a developer at Komodo has helped in providing detail [55]. The process is desribed below.

1. Alice sends dexfee
2. Bob sends bobdeposit (ETOMIC) and bob calls bobMakesEthDeposit(...) (ETH)
3. Alice sends alicepayment (BTC) (alice will have to inspect the ethereum blockchain to confirm the Eth deposit first)
4. Bob sends bobpayment (ETOMIC) and bob calls bobMakesEthPayment() (ETH)
5. Alice spends the bobpayment (ETOMIC) and calls aliceClaimsPayment() (ETH)
6. Bob spends the alicepayment (BTC)
7. Bob refunds his own deposit (ETOMIC) and calls bobClaimsDeposit() (ETH)

ETOMIC Token Komodo support Assetchains, a spin up of a komodo blockchain that manages the distribution and security of token on Komodo. The Komodo team have created an ETOMIC assetchain and distribute ETOMIC to users via a faucet. An assetchain is notarised to the main komodo ecosystem which is notarised to Bitcoin.

4.5.2 BlockDX (BlockNet)

BlockDX is a DEX developed by BlockNET. It allows users to execute trades for BTC-like coins. BlockDX uses order books that are cached and stored by Service nodes. A peer-to-peer messaging protocol is implemented between service nodes, makers and takers [56].

Service Node BlockDX defines the role of a service node, a third party operator required to enable trading. Service nodes facilitate mechanisms for accepting but not spending trading fees (anti-spam and anti-DoS fees) to discourage a party from constantly cancelling orders before they are accepted [56].

Order Matching A high level overview of the order match process is defined, more detail can be found in the blockDX whitepaper [56].

1. A maker privately sends the order and an anti-spam fee to a Service Node. The service node can broadcast the latter, spending the coins as a network fee and costing the maker if it acts maliciously. The Service Node then signs the order, which the potential taker uses to validate the order.

2. When an order is broadcast, traders may verify that the order is backed by real coins and that it has been signed by a Service Node

3. When one or more takers attempt to accept an order, the maker decides between these requests (typically accepting the first valid request) and selects its counterparty.

4. When a counterparty is selected, it is in the makers interest that the counterparty will not DOS the trade. Therefore, it awaits verification by a Service Node that a trade fee was paid.

5. Once a Service Node broadcasts a signed acceptance-message, the rest of the market updates its order books by removing the order from the book

An atomic swap protocol is similar to 2.1.1 is implemented to settle the trade.

4.6 Etherscan Market Share

Etherscan is an Ethereum Block explorer [57], they offer a DEX order tracker product which tracks transactions made to 21 projects listed, many of which are analysed in this study. Figure 4.9 shows a pie chart of the market share of the listed DEX. The market share has been calculated based on the percentage of orders placed on each DEX in the past 7 days dated on 04/09/18 [5].

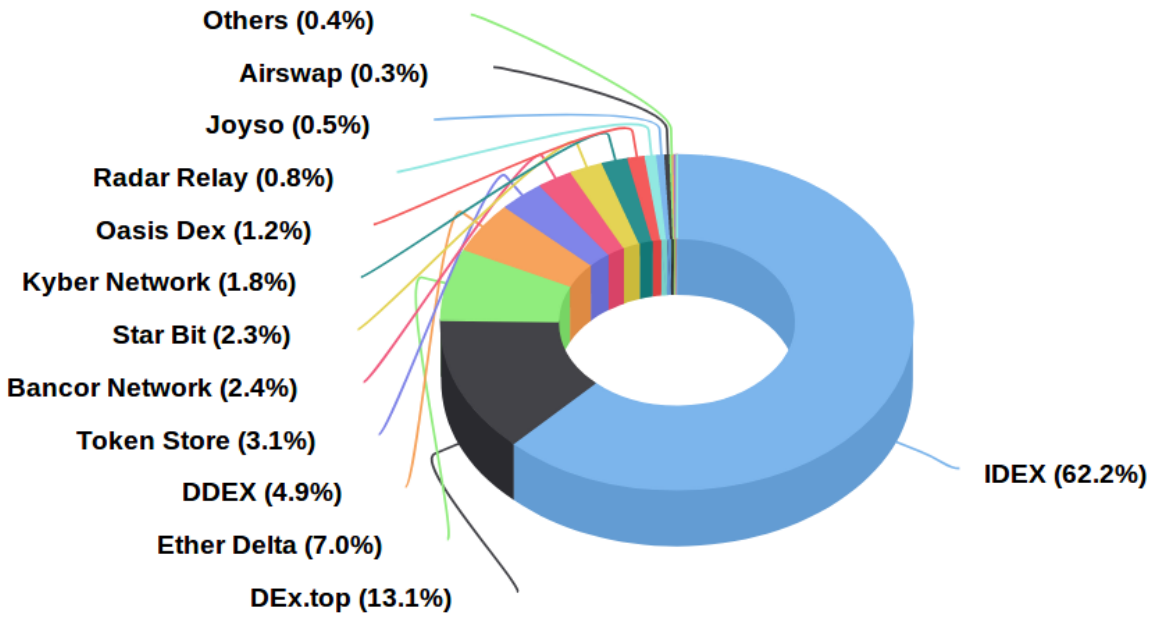


Figure 4.9: Etherscan DEX Pie chart showing marker share of the listed DEX, based on the number of order transaction place in the past 7 days dated on the 04/09/18 [5]

4.7 General Summary

To summarise, a Tables 4.1 and 4.2 of general analysis are presented. The categories for general analysis are described.

Paper Provides a reference to the whitepaper if available.

Code Provides a reference to the codebase if available.

Live Describes the stage of development for the project, all projects described have some form of deployment to a mainnet blockchain network. Options include alpha, beta and live.

ETH Volume Where possible ETH volumes of DEX are given, all volumes are taken from CoinMarketCap's Exchange Index on 01/09/18 [58].

DEX Category Categorisation of DEX have been defined in detail in this chapter, the categories are Order book, On-Chain Reserves, and Peer-to-Peer Negotiation. The categories capture key design features of these DEX, which result in security, performance and usability gains and losses.

Chain Support Chain Support describes whether the DEX is build for a native blockchain or whether is operates cross-chain. As mentioned the majority of DEX are build for Ethereum ERC20 tokens. Bitcoin-based refers to DEX that are designed to trade assets that support the Bitcoin Core API (2.2.3) e.g. Bitcoin Cash, Litecoin, Dash, Zcash etc.

Blockchain Requirements Blockchain Requirements describes the technical requirements needed on native chains so that the swap protocols can run. Examples include Bitcoin opcodes such as OP_CheckLockTimeVerify (2.2.3) or script contracts such as 2-of-2 multisig (2.2.3).

Below a general analysis of the DEX are presented.

Table 4.1: General DEX analysis

Exchange		Paper	Code	Live	Deploy Date	Volume ($USD)[1]
0x Protocol		[1]	[59]	Live	08/17	n/a
0x	**DDEX (Hydro)**	[42]	[60]	Live	01/18	209,435
	Radar Relay	n/a	[43]	Live	10/17	491,767
EtherDelta		n/a	[45]	Live	02/17	140,950
IDEX (AuroraDao)		[2]	[61]	Live	10/17	2,438,237
DEx.top		[44]	[62]	Live	05/18	440,392
OasisDex (MakerDao)		[63]	[64]	Alpha	12/17	440,784
AirSwap		[4]	[65]	Live	04/18	3,740
KyberSwap (KyberNetwork)		[3]	[66]	Live	07/18	155,944
Bancor		[50]	[67]	Live	10/17	1,893,573
barterDEX (Komodo)	**BTC-like <-> BTC-like**	[51]	[68]	Beta	03/18	n/a
	BTC-like <-> ETH					
	ETH <-> ETH					
BlockDX (BlockNet)		[56]	[69]	Live	?	n/a

[1]Volume sourced from CoinMarketCap [58] on the 01/09/18

Table 4.2: General DEX analysis

Exchange		Category	Technology	Chain Support	Blockchain Requirements	
					Chain A	Chain B
0x Protocol		Order Book	Single On-Chain Swaps	Ethereum	Turing Complete Scripting	n/a
0x	**DDEX (Hydro)**	Order Book	Single On-Chain Swaps	Ethereum	Turing Complete Scripting	n/a
	Radar Relay	Order Book				
EtherDelta		Order Book	Single On-Chain Swaps	Ethereum	Turing Complete Scripting	n/a
IDEX (AuroraDao)		Order Book	Single On-Chain Swaps	Ethereum	Turing Complete Scripting	n/a
DEx.top		Order Book	Single On-Chain Swaps	Ethereum	Turing Complete Scripting	n/a
OasisDex (MakerDao)		Order Book	Single On-Chain Swaps	Ethereum	Turing Complete Scripting	n/a
AirSwap		Peer-to-Peer	Single On-Chain Swaps	Ethereum	Turing Complete Scripting	n/a
KyberSwap (KyberNetwork)		On-Chain Reserves	Smart Contract	Ethereum	Turing Complete Scripting	n/a
Bancor		On-Chain Reserves	Smart Contract	Ethereum	Turing Complete Scripting	n/a
barterDEX (Komodo)	**BTC-like <-> BTC-like**	Order Book	Single On-Chain Swaps	BTC Based	OP_CLTV	OP_CLTV OR 2-of-2 multisig
	BTC-like <-> ETH		Cross-chain	Cross-chain[1]	OP_CLTV	Turing Complete Scripting
	ETH <-> ETH		Single On-Chain Swaps	Ethereum[2]	Turing Complete Scripting	Turing Complete Scripting
BlockDX (BlockNet)		Order Book	On-Chain Swaps	BTC Based	OP_CLTV	OP_CLTV

[1] A barterDEX assetchain token ETOMIC is needed to enable BTC-like-ETH/ERC20 tokens on Ethereum

[2] A barterDEX assetchain token ETOMIC is needed to enable ETH/ERC20-ETH/ERC20 tokens on Ethereum

Chapter 5

DEX Analysis: Security & Decentralisation

This chapter will closely look at the security and decentralisation properties of the DEX. Security analysis will examine to what degree user funds are at risk of being lost due any failure and will examine whether the DEX are vulnerable to attacks or manipulation including front-running, maker grieving, wash trading etc. Decentralisation analysis with examine the trust model of the design focusing on points of failure due to trusting parties to behave fairly.

5.1 Trust Model

A trust model outlines the points in a system where trust in placed in a participant to ensure a protocol failure does not occur. The trust model can be considered from two view points, liveness and safety.

5.1.1 Liveness

Liveness describes the property that the protocol proceeds to an eventual conclusion once it is initiated [70]. A liveness failure occurs when the protocol is halted in execution at any stage, this may or may not lead to any participant incurring a financial loss. Liveness failure typically occurs when a participant in the system is not live or does not respond or behave as expected. Liveness failure need to be avoided so that DEX can operate and successfully complete trades.

All the DEX analysed are prone to liveness failure as they rely on the availability and liveness of a public blockchains such as Bitcoin and Ethereum. Specifically Ethereum, as most of the DEX rely on smart contacts on the blockchain to execute the trades.

In many cases, liveness failure is due to the potential of the DEX operator going off-line or not behaving fairly. Off-chain order book DEX design rely on the operator to manage the order book infrastructure. There is always a possibility that the operator's servers are taken off-line. The following DEX are susceptible to this form of liveness failure; **DDEX**, **Radar Relay EtherDelta IDEX Oasis DEX** and **DEx.top**.

AirSwap is also susceptible to a similar liveness failure as the Indexer is maintained by AirSwap, if it is unresponsive users fail to find peers to trade with. Both maker and taker can choose to unresponsive in the negotiation process which also causes a liveness failure.

Both **barterDEX** and **BlockDX** have interactive atomic swap protocols where both maker and taker must participate therefore if one is unresponsive, this will lead to a liveness failure.

5.1.2 Safety

Safety can be partially viewed as an extension to liveness, if a liveness failure leads a participant paying a financial penalty or incurring some loss, this is a safety failure [70]. Furthermore, if user funds are under the custodian of the DEX this could lead to a safety failure where user funds are stolen.

The majority of the DEX complete an on-chain swap executed by a smart contract. The contract could potentially be written in a way that leads to user funds being stolen. However, the projects considered in this analysis have public contracts that have been scrutinised and the likelihood of a safety failure due to this vector is small.

There are some DEX, (**EtherDelta**, **IDEX**, and **DEx.top**) that require users to place a deposit, this transfers ownership of their token to the DEX smart contract. This exposes users further as a bug, malicious or not, may result in their tokens being stolen. However, community scrutiny has not raised any concerns with the DEX mentioned.

Both **barterDEX** and **BlockDX** have interactive atomic swap protocols prone to liveness failure. However, the protocol are designed such that a safety failure is minimised (see 2.1.1 for more details).

5.1.3 Censorship & Manipulation

In most of the DEX analysed, the DEX operators have a central role to an extent. In all order book DEX, the order book manager; the relayers on 0x, EtherDelta, IDEX, DEx.top, Oasis Dex are tasked with standing up the infrastructure for the off-chain order book. Therefore they have the ability to censor any orders placed by makers. In the case of order matching style DEX (IDEX, DEx.top and DDEX), all transactions that settle trades are placed by the DEX operator. The DEX operator can therefore manipulate the order of these transactions. Correspondence with IDEX developers clarifies that IDEX can censor makers on the platform [71]. Censorship is not limited to order book DEX, AirSwap indexer is tasked with aiding peer discovery and has the ability to censor makers and takers attempting to discover peers. Correspondence from AirSwap engineers has revealed that AirSwap operate a blacklist where some peers a blocked for using the indexer service [72].

5.2 Attack & Security Breaches

To date the majority of the DEX analysed have not suffered from publicised attacks or security breaches.

5.2.1 Bancor Network Hack

On the 9th of July 2018, Bancor suffered an attack where $23.5 million of cryptocurrency tokens were stolen. In a statement, Bancor say that ”a wallet used to upgrade some smart contracts was compromised. The attackers made off with $12.5 million in Ether, $1 million in Pundi Xs NPXS token and $10 million in Bancors BNT. [73]. The ether and tokens were stolen from connector contracts that hold the reserves. This suggest that Bancor has the ability to access the tokens held in the reserves of the connectors to these Smart Tokens. Further, Bancor announced that then BNC tokens were ”frozen”. Bancor have defended the actions claiming these powers are part of a strategic decision to have a 3 year pilot period with safety measures to protect the community. However, this is clear evidence of centralisation.

5.3 Front Running

In the case of an open order book style DEX, the threat of front running is apparent. Front-running is the act of a party stepping ahead of orders that are about to filled. A front runner is a party that has a view of orders being placed by a maker and then a taker attempts to fill that order, at this point the front runner can step ahead of the taker and fill the order, therefore griefing the taker. The taker has spent time and paid a fee to attempt to fill this order, they will lose out.

Figure 5.1 shows front running occurring on an open order book DEX on Ethereum. The stages are described below.

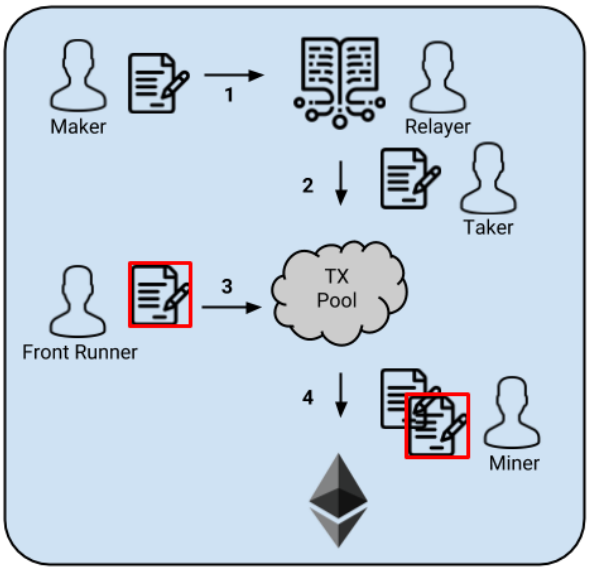


Figure 5.1: Open Order Book Front-running, the front runner steps ahead of the taker to fill the order

1. Maker places order that is included in the order book
2. Taker selects the order and fills it, placing a transaction on-chain, the transactions is held in the tx pool waiting to be mined
3. The front runner observes the taker's transaction in the tx pool before creating a transaction to fill the same order but places a higher gas price for the transaction.
4. The miner decides to mine the front runner's transaction first as it has a higher reward

Another front running situation can occur where a miner them self acts as a front-runner. This would mean the gas price for the transaction will not be required to be higher than the original transaction.

Radar Relay and **EtherDelta** are the DEX that are susceptible to the front-running defined. Order book DEX that implement the Order Matcher style order book (4.2.1) are not susceptible to this kind of front running, as the operator is responsible for placing trades on-chain. **IDEX**, **DDEX** and **DEx.top** fall into this category.

The Peer-to-Peer Negotiation and On-chain Reserve DEX are not susceptible to front running as there is no order book and maker/taker interaction.

For both **barterDEX** and **BlockDX**, the interactive nature of the atomic swap protocol makes front-running impossible.

5.4 Race Conditions

On-chain transactions have a time delay between being broadcast to the network and being mined into a block. During the waiting period, the transactions sit in the pending transaction pool waiting to be included in a block. The length of time transactions must wait depends on the frequency at which new transaction are created, if this is greater than the blockchain's maximum throughput, the backlog of pending transactions will grow. This results in blockchain transaction being vulnerable to race conditions and manifest in collisions related on-chain settlement.

Trade Collision A trade collision is a process similar to front-running. In an open order book DEX, multiple parties can choose to take an order and broadcast their transactions within a single block time. When one of these transactions are included in a block all the others will fail to verify.

Radar Relay and **EtherDelta** are vulnerable to this race condition.

Cancel Collisions A cancel collision can occur when a maker attempts to cancel an order on-chain within the block time of a taker who attempt to fill it. **Radar Relay**, **Ether Delta** and **AirSwap** are vulnerable to this race condition.

5.5 Maker Griefing

Maker griefing is an attack where a maker of an order moves funds that are made available for a trade before a taker or the DEX operator can execute the trade on-chain.

In the case of **EtherDelta**, **IDEX** and **DEx.top**, users must deposit tokens in the

contract. On **EtherDelta**, a maker may be able to withdraw tokens via an on-chain transaction before a taker can fill it on-chain. However, on **IDEX** and **DEx.top**, the DEX operator is tasked with placing transactions that match orders therefore they are able to track the order of transactions off-chain and ensure fairness.

5.6 Wash Trading

Wash trading is process where a trader chooses to sell a token to themselves on an exchange with the purpose of feeding misleading information to the market [74] i.e. boosting the volume of a token or to grief the DEX operator if they absorb the trading cost.

All the DEX in this analysis place a transaction cost on either the maker or taker which will discourage wash trading for the purpose of griefing the DEX operator. However, eliminating wash trading for the purpose of misleading the market is more challenging, where a user is willing to absorb the trading costs. Ethereum addresses are pseudo anonymous and identifying whether any two addresses relate to the same user is technically challenging.

5.7 False Flagging

False Flagging is a situation where individuals or groups of traders create buy or sell walls. A buy wall is where there are significantly more buy orders than sell for a token pair and the sell wall is vice versa. These walls can encourage traders to either buy or sell a token pair [75]. In the case of a buy wall, sentiment suggest buying would be a good idea and the reverse is true for a sell wall. False flaggers create these walls, as other traders react, they cancel their orders and suspend buying or selling. The trade movement will have moved the price point of the token pair and the false flaggers aim to benefit from this movement. This is a form of market manipulation [75].

All DEX suffer from finite time delays when completing on-chain transaction where takers fill orders and makers cancel orders. Responsive trading strategies of this nature are therefore difficult to implement.

5.8 Security & Decentralisation Summary

Trust Model

Liveness Outlines the parties who actions can lead to a liveness failure. The following abbreviations are used; O for operator, SC for smart contract, M for maker, T for taker, PC for platform chain.

Safety Outlines the parties and elements that may lead to a safety failure. SC is an abbreviation for smart contracts

Custodian of funds States whether the DEX is a custodian off user funds, SC is an abbreviation for smart contracts

Availability Requirements Availability requirements relate to Liveness. It outlines whether the matching and swap processes are interactive or non-interactive. The abbreviations I and NI are used respectfully.

Tokenised Incentives States whether users are incentivised by a platform token when using the DEX.

Table 5.1: Security & Decentralisation DEX analysis

Exchange		Trust Model		Custodian of Funds	Availability Requirements		Tokenised Incentives
		Liveness	Safety		Matching	Swap	
0x	**DDEX (Hydro)**	Relayer [O], SC	None	No	NI	NI	Yes
	Radar Relay	Relayer [O], SC					Yes
EtherDelta		O, SC	SC	SC	NI	NI	No
IDEX (AuroraDao)		O, SC	SC	SC	NI	NI	No
DEx.top		O, SC	SC	SC	NI	NI	No
OasisDex (MakerDao)		O, SC	None	No	NI	NI	Yes
AirSwap		Indexer [O], T, M, SC	None	No	I	NI	Yes
KyberSwap (KyberNetwork)		SC	None	No	NI	NI	Yes
Bancor		SC	None	No	NI	NI	Yes
barterDEX (Komodo)	**BTC-like <-> BTC-like**	Komodo [PC], BTC, M, T	None	No	NI	I	No
	BTC-like <-> ETH	Komodo [PC], SC, M, T					Yes
	ETH <-> ETH	Komodo [PC], SC, M, T					Yes
BlockDX (BlockNet)		Service Nodes , M, T	None	No	NI	I	Yes

Chapter 6

DEX Analysis: Performance & Cost

This chapter will analyse the performance of the DEX by considering the latency in trading. A measure of latency is derived from the number of transaction that are required to complete a trade. A distinction is made between on-chain transactions and off-chain messages, as on-chain transactions have a more significant time delay when compared to off-chain messages. A registration requirement will also be considered as an effect on performance.

The cost analysis will consider cost of on-chain transactions along with fees charged by the DEX. The performance and cost are analysed separably for makers, takers and operators or 3rd party operators.

6.1 Transactions & Messages

The typical transactions and messages for the DEX are profiled.

6.1.1 On-chain Transaction

The standard on chain transaction that occur within DEX designs are described.

Approve transferFrom For some Ethereum based DEX, approving the transferFrom (2.3.6) transaction is required. This applies to the maker and taker for 0x relayers (DDEX and Radar Relay), Oasis DEX and AirSwap. These transactions are included in Tables 6.1 and 6.2.

Wrapping ETH Wrapping ETH transaction are required for 0x relayers (DDEX and Radar Relay) and Oasis DEX, this allows users to use ether on these DEX (2.3.7). This transaction is dependant on whether users are trading ether and therefore are not included in the final Table summary.

Trade Execution The trade execution is typically done by the taker reflected in Table 6.2. However, for order matcher order book DEX such as IDEX, DEx.top and DDEX, the trade is placed by the operator shown in Table 6.3.

Deposit & Withdraw For EtherDelta, IDEX and DEx.top, makers and takers are required to transfer their token to the DEX contract before they can trade. They will also need to withdraw their tokens if they wish to spend them elsewhere. Therefore these transaction are added to the Tables 6.1 and 6.2.

Cross-Chain Both cross-chain DEX have more involved protocols that include native token blockchain transactions (e.g. Ethereum and Bitcoin) but also platform chain transactions such as Komodo and BlockNet transactions. For barterDEX refer to 4.5.1 for detailed protocol breakdown showing transactions, see 4.5.2 for a similar breakdown for BlockDX.

6.1.2 Off-Chain Messages

Order Placement Orders are typically placed by the maker.

Negotiation Protocol In the case of AirSwap, barterDEX and blockDX there are bespoke messaging protocol that have been described in detail previously.

6.2 Transaction Fees

All on-chain transactions described have an associated fee due to gas. DEX operators have provided estimated costs for some of these transaction. Etherscan is another source for investigating the cost of these transactions. Where possible, the cost of these transactions are provide. Where the cost is provided as eth, an exchange rate of 229.72 has been applied to convert the cost to $(USD).

Approve transferFrom The approve transaction is called on the token contract of the token that is to be traded. However implementation of this method can vary therefore an exact transaction cost is not possible.

Trade Execution Trade execution cost for many of the DEX vary depending on the tokens that are traded. where possible, estimations are provided.

6.3 Exchange Fees

The DEX operator can choose to charge an exchange fee, this is typically percentage of the trade.

6.4 Performance & Cost Summary

Tables 6.1, 6.2, 6.3 summarise the findings of the performance and cost analysis.

No. TX: On-chain Number of on-chain transaction, these transaction include on-chain transaction for the platform blockchains and the native coin/token blockchains.

No. TX: Off-chain messages Outlines the number of off-chain messages require to execute a trade.

TX fees / Gas cost ($USD) Where possible, transaction costs have been provided in $USD. var is used as an abbreviation for variable where the cost is dependant on the tokens being traded. A question mark (?) is placed where information is not available.

Exchange fee (%) An exchanges as a percentage of a trade is provided

Registration States whether participants are required to register with the DEX.

3rd party Outlines the 3rd parties who participate in the matching and trade stages. The following abbreviations are defined; RL for relayer, O for DEX operator, C for reserve contributor on KyberNetwork, LP for a reserve manager on KyberNetwork and sRL for a service node (signed relayer).

Table 6.1: Maker Performance and Cost analysis

Exchange		No. TX		TX fee Gas Cost ($USD)	Exchange fee (%)	Registration
		On-Chain TX	Off-Chain Msg			
0x	DDEX (Hydro)	1	1	var	0.05-0.01	No
	Radar Relay	1	1	var	0	No
EtherDelta		2	1	0.09	0	No
IDEX (AuroraDao)		2	1	0.46	0.1	No
DEx.top		2	1	0.34	0.1	Yes
OasisDex (MakerDao)		1	1	var	0	No
AirSwap		1	2	var	0	No
KyberSwap (KyberNetwork)		n/a				
Bancor		n/a				
barterDEX (Komodo)	BTC-like <-> BTC-like	2	3	?	0	No
	BTC-like <-> ETH	3	3	?		
	ETH <-> ETH	7	3	0.16		
BlockDX (BlockNet)		3	5	?	0.05	No

Table 6.2: Taker Performance and Cost analysis

Exchange		No. TX		TX fee Gas Cost	Exchange fee (%)	Registration
		On-Chain TX	Off-Chain Msg			
0x	**DDEX (Hydro)**	n/a				
	Radar Relay	2	0	var	0	No
EtherDelta		3	0	0.16	0.3	No
IDEX (AuroraDao)		1	1	var	0.2	No
DEx.top		1	1	var	0.2	Yes
OasisDex (MakerDao)		2	0	var	0	No
AirSwap		1	1+n[1]	0.07	0	No
KyberSwap (KyberNetwork)		1	0	var	0	Yes
Bancor		1	0	var	0	No
barterDEX (Komodo)	**BTC-like <-> BTC-like**	4	1	?	0	Yes
	BTC-like <-> ETH	7	1	0.05		
	ETH <-> ETH	7	1	?		
BlockDX (BlockNet)		3	3	?	0.05	No

[1] n represents the number of maker that the taker negotiates with before coming to an agreement

Table 6.3: 3rd party Performance and Cost analysis

Exchange		**3rd Party**	**No. TX**		**TX fee Gas Cost**	**Registration**
			On-Chain TX	**Off-Chain Msg**		
0x	**DDEX (Hydro)**	RL	1	?	var	n/a
	Radar Relay		0	?	n/a	n/a
EtherDelta		n/a				
IDEX (AuroraDao)		O	1	0	var	n/a
DEx.top		O	1	0	var	n/a
OasisDex (MakerDao)		n/a				
AirSwap		n/a				
KyberSwap (KyberNetwork)		C	?	?	?	No
		LP	?	?	?	Yes
Bancor		n/a				
barterDEX (Komodo)	**BTC-like <-> BTC-like**	n/a				
	BTC-like <-> ETH					
	ETH <-> ETH					
BlockDX (BlockNet)		sRL	4	0	?	Yes

Chapter 7

DEX Analysis: Usability

A usability analysis will consider how usable the DEX is to the end user and any third party operator. The software and client requirements are considered for the participants of the DEX. In some cases, users require a platform token when using the DEX adding a layer of complexity, these requirements will be analysed to assess utility. Liquidity on exchange platforms are important to allow efficient markets and competitive pricing, where DEX have made specific liquidity provisions, these plans will be discussed.

7.1 Roles

Usability analysis will be considered for two separate participant on a DEX.

User A user is either a maker or taker in the DEX.

3rd party operator A 3rd party operator takes up a required role in the DEX. The actual DEX operator who has developed the product is not considered a 3rd party operator. Examples of 3rd party operators are reserve managers in for KyberSwap or service nodes in blockDX.

7.2 System Specific Requirements

To makes trades on all the DEX, users are required to run a client on the blockchain for native crypto asset that is being traded, this includes Bitcoin and Ethereum. Users have two options for clients on these blockchain.

Full Client A full client implementation downloads, installs and synchronises a wallet of a cryptocurrency. The initial installation is time consuming and running the client requires significant storage. However, a full client is typically faster and more stable than SPV. Running a full client has minimum requirements with including 145 gigabytes of free disk space, a broadband internet connection and a minimum of 6 hours a day where the client is left running [76].

SPV or Remote Client An SPV client uses simple payment verification where the full copy of the blockchain is not maintained. In the context of Bitcoin, an SPV node only downloads block headers and not the transactions in each block. SPV nodes do not have a full view of all UTXOs and rely on peers to provide partial views of the blockchain to verify transactions [14]. An examples is the Electrum wallet [77].

In the case of Ethereum, remote clients are a popular choice for most users. A remote clients does not store the full Ethereum blockchain but relies on a full node elsewhere to keep updated [18]. A popular example is MetaMask [78].

The majority of the DEX in this analysis require users to run a remote client for Ethereum as the DEX are based on the Ethereum blockchain. KyberSwap defines two 3rd party roles; the reserve manager who requires a full Etheruem client and reserve contributor who requires a remote client.

For the cross-chain exchanges, an SPV or remote client is required for the native blockchains. With barterDEX, users a required to install and run a Komodo client and a barterDEX wallet client. In the case of blockDX, users require a Blocknet client.

7.3 Platform Token Requirements

Some DEX requires users and 3rd party operators to utilise a platform token outside an added incentivisation scheme. This requirement places an added barrier to usability as users and 3rd party operators need to source the token and purchase it to use the platform.

0x requires users (makers and takers) to pay fees to relayers using ZRX tokens. The token is also used for voting in its proposed decentralised governance mechanism.

AST tokens are required for makers and takers to signal intent to the indexer and trade a token pair on AirSwap.

Reserve managers, a 3rd party operator on the KyberNetwork require the KNC token, to operate a reserve.

On barterDEX, users require the ETOMIC token making any exchange involving ether or ERC20 tokens. BlockDX users and Service nodes require the BLOCK coin to use the DEX.

7.4 Listing Restrictions

All the DEX in this analysis place restrictions on tokens that can be traded on their platforms. All DEX except AirSwap, operate a whitelist or token registry, limit the token pairs that can be traded on the platform.

AirSwap is the only DEX that operates a blacklist, where tokens that have been flagged as fraudulent are not allowed to be traded. All other ERC20 tokens can be traded on the platform.

7.5 Liquidity

Some of the DEX make provisions to ensure liquidity on their networks. Both KyberSwap and Bancor have liquidity at the heart of their designs. 0x has outlined plans for networked liquidity on their platform, where relayers following the same strategy can share order books. However, there a significant difficulties in creating a fee sharing mechanism amongst the relayers [79]. The AirSwap team have made agreements with professional liquidity providers who will make markets on the platform [80].

7.6 Usability Summary

Table 7.1 presents the results of the usability analysis.

Specific System Requirements Defines the type of blockchain clients required to use the DEX for a user and 3rd party operator. Options include SPV (also represents remote) or a full client. Any other clients or wallet software required for separate blockchain systems are included

Platform Tokens Requirements Defines whether users and 3rd party operators are required to use a platform token to complete a trade.

Order Cancellation Defines whether users have the ability to cancel an order

Listing Restrictions Outlines what listing restrictions are placed by the DEX operator. Typically DEX have a whitelist or a token registry of traded token pairs and some operate a blacklist of banned tokens. The following abbreviations are used W for whitelist and B for blacklist.

Table 7.1: DEX Usability Analysis

Exchange		System Specific Requirements		Platform Token Requirement		Order Cancellation	Listing Restriction
		User	**3rd party Operator**	**User**	**3rd party Operator**		
0x	**DDEX (Hydro)**	SPV Client <Eth>	n/a	Yes	Yes	Yes	W
	Radar Relay	SPV Client <Eth>					W
EtherDelta		SPV Client <Eth>	n/a	No	n/a	Yes	W
IDEX (AuroraDao)		SPV Client <Eth>	n/a	No	n/a	Yes	W
DEx.top		SPV Client <Eth>	n/a	No	n/a	Yes	W
OasisDex (MakerDao)		SPV Client <Eth>	n/a	No	n/a	Yes	W
AirSwap		SPV Client <Eth>	n/a	Yes	n/a	Yes	B
KyberSwap (KyberNetwork)		SPV Client <Eth>	Full Client <Eth>[1]	No	Yes	n/a	W
			SPV Client <Eth>[2]		No		
Bancor		SPV Client <Eth>	n/a	No	n/a	n/a	W
barterDEX (Komodo)	**BTC-like <-> BTC-like**	SPV Client <BTC-like> Komodo Client, barterDEX Wallet	n/a	No	n/a	Yes	W
	BTC-like <-> ETH	SPV Client <BTC-like, Eth> Komodo Client, barterDEX Wallet		Yes			
	ETH <-> ETH	SPV Client <Eth> Komodo Client, barterDEXWallet		Yes			
BlockDX (BlockNet)		SPV Client <BTC-like> Blocknet Client	SPV Client <BTC-like> Blocknet Client[3]	Yes	Yes	Yes	W

[1] KyberNetwork Reserve manager
[2] Reserve contributor
[3] BlockNet Service Node

Chapter 8

Discussion

An overall discussion is presented incorporating the analysis presented so far, the single chain and cross-chain DEX will be discussed separately.

8.1 Order Book

The order book category is the most populous category of the projects analysed. It suffers from a degree of centralisation, when off-chain order books are deployed to improve performance. Off-chain order books reduce the number of on-chain transactions required reducing costs and the latency associated with trading. However, off-chain management of order books exposes users to the potential of censorship.

Unmatched orders leads to many potential attacks such as front running and maker griefing. Accidental collisions with regards to trades and cancellation can occur due to race conditions associated with filling order on-chain, these collision are likely to be exacerbated as trading volumes increase. Matching orders via a DEX operator to avoid the attacks and collisions leads to more centralisation, allowing the DEX operator to manipulate the order of which transactions are broadcast to the blockchain.

None of the order book DEX analysed have a clear mechanism in place to provide liquidity to the platforms.

8.2 On-Chain Reserve

On-chain reserves address the front running and race conditions associated with order book DEX but the cost of added complexity of maintaining liquidity and pricing mechanisms.

KyberNetwork addresses liquidity and pricing by allowing liquidity providers to set competitive spreads and make markets. The KyberOperator role has significant importance in the system, listing/delisting token pairs and adding/removing reserves. This role is taken by Kyber Network resulting in centralisation. There are plans

outlined for a decentralised governance mechanism, implementation is yet to be completed. The entire ecosystem is on-chain introducing a operational latency to the 3rd party operators involved such as the reserve managers and reserve contributors. However, there is reduced messaging complexity to the end user who requires a single on-chain transaction to execute a trade.

Bancor Network introduces smart token that deterministically sets supply and price of a token. However the project has demonstrated centralisation after a wallet with control of platform contracts was breached leading to a loss of funds.

8.3 Peer-to-Peer Negotiation

Peer-to-Peer negotiation, namely AirSwap, replaces order books with a light weight messaging protocol that facilitates a negotiation between peers. It also eliminates front running attacks and collisions are associated with unmatched order.

This introduces more messaging overhead in the trade however most of this messaging is off-chain. AirSwap does rely on an indexer, tasked with enabling peer discover. The indexer is a point of centralisation, the AirSwap team implement the indexer and therefore have the capability to censor users.

AirSwap have announced agreements with professional liquidity providers to make markets on the platform, the nature of these agreements are not clear.

8.4 Cross-Chain

Cross-chain exchanges have posed a technical challenge that yet to be fully mastered by a project in this space. Both barterDEX and BlockDX have implemented DEX that use atomic cross-chain swaps. In both cases, an additional requirement related to the security of the platform blockchains has been added. In both cases, messaging complexity is comparatively high in comparison single chain DEX, additional security waiting periods are required to transactions on blockchains have sufficient block confirmations. Both projects are in beta testing and are being continually developed.

Chapter 9

Future Work & Conclusion

9.1 Future Work

Transaction Analysis Transaction analysis can be conducted to determine whether attacks such as front-running and maker griefing are prevalent on order book DEX.

DEX Governance Some of the DEX including KyberSwap and 0x have plans to implement decentralised governance mechanisms replacing centralised components of their designs. A study of these designs and future implementations would be valuable. Governance is the root of centralisation in the DEX analysed in this study.

9.2 Conclusion

In conclusion, a complete analysis of the most developed decentralised exchange projects has been presented. Where possible, white papers, documentation and code has been analysed. In some cases, primary information has been gathered from engineering teams of the projects to clarify understanding. The analysis has presented a strict and well defined categorisation of the projects; Order Book, On-Chain Reserves and Peer-to-Peer Negotiation.

Security analysis is conducted, where liveness and safety failures are identified and explained. Points of centralisation are established and potential failures due to the centralisation are defined. Attacks such as front-running and maker griefing and on-chain trade collision due to race conditions are demonstrated.

Performance is analysed focused towards on and off chain messaging overhead for participants, the overhead correlates to trade execution latency. Where possible exchange fees and transaction costs are presented. Usability, from the perspective of users and 3rd party operators are considered focusing on software and platform token requirements.

All results are collected in clear tables. An overall discussion summarising the degree of decentralisation of projects in each category is presented.

Bibliography

[1] 0x: An open protocol for decentralized exchange on the ethereum blockchain. https://0xproject.com/pdfs/0x_white_paper.pdf. Accessed: 2018-06-08. pages viii, 14, 31

[2] Idex: A real-time and high-throughput ethereum smart contract exchange. https://idex.market/static/IDEX-Whitepaper-V0.7.5.pdf. Accessed: 2018-06-08. pages viii, 17, 18, 31

[3] Kybernetwork: White paper. https://home.kyber.network/assets/KyberNetworkWhitepaper.pdf. Accessed: 2018-06-08. pages viii, 21, 31

[4] Swap: A peer-to-peer protocol for trading ethereum tokens. https://swap.tech/whitepaper/. Accessed: 2018-08-08. pages viii, 23, 25, 31

[5] Dex tracker statistics. https://etherscan.io/stat/dextracker?range=7. Accessed: 2018-09-04. pages viii, 29

[6] Satoshi Nakamoto. Bitcoin: A peer-to-peer electronic cash system, http://bitcoin.org/bitcoin.pdf, 2008. pages 1

[7] Bitcoin and cryptocurrency technologies arvind narayanan, joseph bonneau, edward felten, andrew miller and steven goldfeder. *Network Security*, 2016(8):4, 2016. pages 1

[8] Etherscan: Token tracker. https://etherscan.io/tokens. Accessed: 2018-09-02. pages 1

[9] Darryn Pollock. The mess that was mt. gox: Four years on. https://cointelegraph.com/news/the-mess-that-was-mt-gox-four-years-on, March 2018. pages 1

[10] Izabella Kaminska. Bitcoin bitfinex exchange hacked: the unanswered questions. https://www.ft.com/content/1ea8baf8-5a11-11e6-8d05-4eaa66292c32, August 2016. pages 1

[11] M. Herlihy. Atomic cross-chain swaps. *arXiv:1801.09515*. pages 2

[12] TierNolan. Alt chains and atomic transfers. https://bitcointalk.org/index.php?topic=193281.msg2003765#msg2003765, 2013. pages 2

[13] Btc relay. https://github.com/ethereum/btcrelay. Accessed: 2018-06-08. pages 3

[14] Andreas M. Antonopoulos. *Mastering Bitcoin: Unlocking Digital Crypto-Currencies*. O'Reilly Media, Inc., 1st edition, 2014. pages 3, 4, 48

[15] BTC Relay. http://btc-relay.readthedocs.io/en/latest/frequently-asked-questions.html#how-does-btc-relay-work, 2016. pages 3

[16] Alexei Zamyatin, Dominik Harz, Joshua Lind, Panayiotis Panayiotou, Arthur Gervais, and William J. Knottenbelt. Xclaim: Interoperability with cryptocurrency-backed tokens. Cryptology ePrint Archive, Report 2018/643, 2018. https://eprint.iacr.org/2018/643. pages 3, 10

[17] Christian Decker and Roger Wattenhofer. A fast and scalable payment network with bitcoin duplex micropayment channels. In *SSS*, 2015. pages 5

[18] A.M. Antonopoulos and G. Wood. *Mastering Ethereum: Building Smart Contracts and Dapps*. O'Reilly Media, Incorporated, 2019. pages 5, 6, 48

[19] Account management. http://ethdocs.org/en/latest/account-management.html. Accessed: 2018-08-30. pages 5

[20] Contracts. http://ethdocs.org/en/latest/contracts-and-transactions/contracts.html#contracts. Accessed: 2018-08-30. pages 6

[21] eip-20. https://github.com/ethereum/EIPs/blob/master/EIPS/eip-20.md. Accessed: 2018-08-30. pages 6

[22] W-eth. https://weth.io/. Accessed: 2018-08-30. pages 7

[23] Saturn Network. Architecture comparison of decentralized exchanges. https://rados.io/architecture-comparison-of-decentralized-exchanges/, February 2018. pages 7

[24] Investopedia. Order. https://www.investopedia.com/terms/o/order.asp. Accessed: 2018-08-31. pages 8

[25] Investopedia. Limit order. https://www.investopedia.com/terms/l/limitorder.asp. Accessed: 2018-08-31. pages 8

[26] Investopedia. Maket maker. https://www.investopedia.com/terms/m/marketmaker.asp. Accessed: 2018-08-31. pages 8

[27] Investopedia. Order book. https://www.investopedia.com/terms/o/order-book.asp. Accessed: 2018-08-31. pages 8

[28] Investopedia. Liquidity. https://www.investopedia.com/terms/l/liquidity.asp. Accessed: 2018-08-31. pages 8

[29] Alex Munkachy. 30+ hacks cryptocurrency exchange hacks a comprehensive list. https://coiniq.com/cryptocurrency-exchange-hacks/. Accessed: 2018-06-08. pages 8

[30] Michael Borkowski, Daniel McDonald, Christoph Ritzer, and Stefan Schulte. Towards atomic cross-chain token transfers : State of the art and open questions within tast. 2018. pages 10

[31] Peter Bennink, Lennart van Gijtenbeek, Oskar van Deventer, and Maarten Everts. An analysis of atomic swaps on and between ethereum blockchains using smart contracts. 2018. pages 10

[32] Iddo Bentov, Yan Ji, Fan Zhang, Yunqi Li, Xueyuan Zhao, Lorenz Breidenbach, Philip Daian, and Ari Juels. Tesseract: Real-time cryptocurrency exchange using trusted hardware. Cryptology ePrint Archive, Report 2017/1153, 2017. Accessed:2018-08-04. pages 10

[33] Phil Glazer. Decentralized cryptocurrency exchanges. https://hackernoon.com/decentralized-cryptocurrency-exchanges-93039613eeb7. Accessed: 2018-08-14. pages 10

[34] Nathan Sexer. State of decentralized exchanges, 2018. https://media.consensys.net/state-of-decentralized-exchanges-2018-276dad340c79. Accessed: 2018-08-14. pages 10

[35] Gary Basin. The state of decentralized exchanges. https://hackernoon.com/the-state-of-decentralized-exchanges-235064446ab0. Accessed: 2018-08-07. pages 10

[36] Web 3 Foundation. Decentralized exchanges workshop outcomes. https://medium.com/web3foundation/decentralized-exchanges-workshop-outcomes-4753dbd86f2b. Accessed: 2018-06-08. pages 10

[37] Web 3 Foundation. Decentralized exchanges workshop outcomes. https://docs.google.com/spreadsheets/d/1H7_w7kazjFmXzeo6nU0TTRSpF9mvQI8dP1X3WFfIb_Q/edit#gid=0. Accessed: 2018-06-08. pages 10

[38] Mansi Prakash. Ecosystem of decentralized exchanges. https://medium.com/@mansiprakash/ecosystem-of-decentralized-exchanges-88ba89f10d64. Accessed: 2018-06-23. pages 10

[39] Deepa Sathaye. The lay of the land in decentralized exchange protocols. https://blog.airswap.io/the-lay-of-the-land-in-decentralized-exchange-protocols-55ed00feb3df. Accessed: 2018-08-14. pages 10

[40] Altcoin.io Exchange. Decentralized exchanges explainedŁŁand why we need them more than ever. https://blog.altcoin.io/decentralized-exchanges-explained-and-why-we-need-them-more-than-ever-4ef9fbb9192d. Accessed: 2018-06-23. pages 10

[41] 0xprotocol wiki. https://0xproject.com/wiki. Accessed: 2018-06-08. pages 15, 16

[42] Hydro a coordination layer for decentralized exchanges. https://thehydrofoundation.com/Hydro-Whitepaper-v0116-en.pdf. Accessed: 2018-08-31. pages 17, 31

[43] Radar relay. https://github.com/RadarRelay/. Accessed: 2018-08-31. pages 17, 31

[44] Dex technical white paper. https://github.com/dexDev/DEx.top/blob/master/whitepaper/DEx-Whitepaper-Short-Version.pdf. Accessed: 2018-08-08. pages 18, 31

[45] Etherdelta. https://github.com/etherdelta. Accessed: 2018-08-31. pages 18, 31

[46] Advantages of the ethex on-chain order book. https://medium.com/ethex-market/advantages-of-the-ethex-on-chain-order-book-3d94f91bf24f. Accessed: 2018-06-08. pages 19

[47] Joseph Poon and Vitalik Buterin. Plasma: Scalable autonomous smart contracts. *White paper*, 2017. pages 19

[48] Introducing our plasma dex v1. https://blog.altcoin.io/plasma-dex-v1-launching-next-month-4cb5e5ea56f6. Accessed: 2018-09-05. pages 19

[49] Rami Khalil and Arthur Gervais. Nocust–a non-custodial 2 nd-layer financial intermediary. pages 19

[50] Bancor protocol. https://storage.googleapis.com/website-bancor/2018/04/01ba8253-bancor_protocol_whitepaper_en.pdf. Accessed: 2018-08-31. pages 21, 22, 31

[51] Komodo advanced blockchain technology, focused on freedom. https://komodoplatform.com/wp-content/uploads/2018/06/Komodo-Whitepaper-June-3.pdf. Accessed: 2018-08-08. pages 26, 31

[52] barterdex - atomic swap decentralized exchange of native coins. https://github.com/SuperNETorg/komodo/wiki/barterDEX-Whitepaper-v2. Accessed: 2018-08-08. pages 26

[53] Delayed proof of work (dpow) whitepaper. https://github.com/SuperNETorg/komodo/wiki/Delayed-Proof-of-Work-(dPoW)-Whitepaper/. Accessed: 2018-08-08. pages 26

[54] Etomic swap smart contracts for barterdex platform. https://github.com/artemii235/etomic-swap. Accessed: 2018-08-08. pages 27

[55] Personal correspondence with A. Pikulin, Discord Messaging, July 2018. pages 27

[56] The blocknet design specification. https://www.blocknet.co/wp-content/uploads/2018/04/whitepaper.pdf. Accessed: 2018-08-08. pages 28, 31

[57] Etherscan. https://etherscan.io/. Accessed: 2018-09-04. pages 29

[58] Coinmarketcap. https://coinmarketcap.com/. Accessed: 2018-09-01. pages 30, 31

[59] 0x. https://github.com/0xProject. Accessed: 2018-08-31. pages 31

[60] Ddex (digital data exchange). https://github.com/ddexnet. Accessed: 2018-08-31. pages 31

[61] idex-api-docs. https://github.com/AuroraDAO/idex-api-docs. Accessed: 2018-08-31. pages 31

[62] Dex.top - instant trading on chain. https://github.com/dexDev/DEx.top. Accessed: 2018-08-31. pages 31

[63] Oasisdex. https://github.com/OasisDEX/oasis/wiki. Accessed: 2018-08-31. pages 31

[64] Oasisdex open trading interfaces. https://github.com/OasisDEX. Accessed: 2018-08-31. pages 31

[65] Airswap. https://github.com/airswap. Accessed: 2018-08-31. pages 31

[66] Kybernetwork. https://github.com/kybernetwork. Accessed: 2018-08-31. pages 31

[67] Bancor protocol contracts v0.4 (alpha). https://github.com/bancorprotocol/contracts. Accessed: 2018-08-31. pages 31

[68] Barterdex: Decentralised exchange and cryptocurrency market. https://github.com/KomodoPlatform/BarterDEX. Accessed: 2018-08-31. pages 31

[69] blockdx. https://github.com/BlocknetDX/BlockDX. Accessed: 2018-08-31. pages 31

[70] Rachele Fuzzati. A formal approach to fault tolerant distributed consensus. 2009. pages 34

[71] Personal correspondence with F. Whaling, Discord Messaging, August 2018. pages 35

[72] Personal correspondence with D. Mosites, Telegram Messaging, August 2018. pages 35

[73] The crypto worlds latest hack sees bancor lose $23.5m. https://techcrunch.com/2018/07/10/bancor-loses-23-5m/. Accessed: 2018-09-06. pages 35

[74] Wash trading. https://www.investopedia.com/terms/w/washtrading.asp. Accessed: 2018-08-23. pages 38

[75] How to read buy and sell walls in crypto. https://smartoptions.io/read-buy-sell-walls-crypto/#5. Accessed: 2018-08-14. pages 38

[76] Running a full node. https://bitcoin.org/en/full-node#minimum-requirements. Accessed: 2018-09-02. pages 48

[77] Electrum bitcoin wallet. https://electrum.org/#home. Accessed: 2018-08-31. pages 48

[78] Metamask. https://metamask.io/. Accessed: 2018-09-04. pages 48

[79] Fee sharing and networked liquidity in the open orderbook model. https://forum.0xproject.com/t/fee-sharing-and-networked-liquidity-in-the-open-orderbook-model/31. Accessed: 2018-09-04. pages 49

[80] Introducing the airswap partner network. https://blog.airswap.io/introducing-the-airswap-partner-network-a96a4119338. Accessed: 2018-09-04. pages 49

Appendix A

Ethical Considerations

The ethics considerations of this project have been msde against a checklist presented below. Answers to all questions are no and there are not other ethical consideration outside of this list.

Table A.1: Ethics Checklist 1

HUMAN EMBRYOS/FOETUSES	
Does your project involve Human Embryonic Stem Cells?	No
Does your project involve the use of human embryos?	No
Does your project involve the use of human foetal tissues / cells?	No
HUMANS	
Does your project involve human participants?	No
HUMAN CELLS / TISSUES	
Does your project involve human cells or tissues? (Other than from Human Embryos/Foetuses i.e. Section 1)?	No
PROTECTION OF PERSONAL DATA	
Does your project involve personal data collection and/or processing?	No
Does it involve the collection and/or processing of sensitive personal data (e.g. health, sexual lifestyle, ethnicity, political opinion, religious or philosophical conviction)?	No
Does it involve processing of genetic information?	No
Does it involve tracking or observation of participants? It should be noted that this issue is not limited to surveillance or localization data. It also applies to Wan data such as IP address, MACs, cookies etc.	No
Does your project involve further processing of previously collected personal data (secondary use)? For example Does your project involve merging existing data sets?	No
ANIMALS	
Does your project involve animals?	No

Table A.2: Ethics Checklist 2

DEVELOPING COUNTRIES	
Does your project involve developing countries?	No
If your project involves low and/or lower-middle income countries, are any benefit-sharing actions planned?	No
Could the situation in the country put the individuals taking part in the project at risk?	No
ENVIRONMENTAL PROTECTION AND SAFETY	
Does your project involve the use of elements that may cause harm to the environment, animals or plants?	No
Does your project deal with endangered fauna and/or flora /protected areas?	No
Does your project involve the use of elements that may cause harm to humans, including project staff?	No
Does your project involve other harmful materials or equipment, e.g. high-powered laser systems?	No
DUAL USE	
Does your project have the potential for military applications?	No
Does your project have an exclusive civilian application focus?	No
Will your project use or produce goods or information that will require export licenses in accordance with legislation on dual use items?	No
Does your project affect current standards in military ethics e.g., global ban on weapons of mass destruction, issues of proportionality, discrimination of combatants and accountability in drone and autonomous robotics developments, incendiary or laser weapons?	No
MISUSE	
Does your project have the potential for malevolent/criminal/terrorist abuse?	No
Does your project involve information on/or the use of biological-, chemical-, nuclear/radiological-security sensitive materials and explosives, and means of their delivery?	No
Does your project involve the development of technologies or the creation of information that could have severe negative impacts on human rights standards (e.g. privacy, stigmatization, discrimination), if misapplied?	No
Does your project have the potential for terrorist or criminal abuse e.g. infrastructural vulnerability studies, cybersecurity related project?	No
LEGAL ISSUES	
Will your project use or produce software for which there are copyright licensing implications?	No
Will your project use or produce goods or information for which there are data protection, or other legal implications?	No
OTHER ETHICS ISSUES	
Are there any other ethics issues that should be taken into consideration?	No

Appendix B

Professional Considerations

Professional considerations have been given importance through the duration of this project. The BCS code of conduct was followed. The project involved direct communication with engineering teams of the DEX projects researched over messaging mediums such as telegram and discord. In all these communications, a highly professional approach was taken. A transparent approach was taken, where relevant information surrounding the need for direct communication was disclosed. Where results about the projects are presented, significant effort has been placed into ensuring accuracy.