Abstract. We design and build Kardin, an on-chain protocol which allows instant exchange and conversion of digital assets (e.g. crypto tokens) and cryptocurrencies (e.g. Ether, Bitcoin, ZCash) with high liquidity. Kardin will be the first system that implements several ideal operating properties of an exchange including trustless, decentralized execution, instant trade and high liquidity. Besides serving as an exchange, Kardin also provides payment APIs that will allow Ethereum accounts to easily receive payments from any crypto tokens. As an example, any merchant can now use Kardin APIs to allow users to pay in any crypto tokens, but the merchant will receive payments in Ether (ETH) or other preferred tokens. Although running on the Ethereum network, Kardin roadmap includes supporting cross-chain trades between different cryptocurrencies using relays and future protocols like Polkadot and Cosmos. Ethereum accounts will be able to safely receive payment from Bitcoin, ZCash and other cryptocurrencies via our payment APIs, through this trustless payment service. Derivatives will be introduced to mitigate the exposure to the risk of volatilities for the users of Kardin Token (KDN) and selected cryptocurrencies. This will allow users to participate in the price movements synthetically.

Table of contents

1. Introduction

Emerging cryptocurrencies such as Bitcoin, Ethereum and others have been gaining tractions of late because they allow users to transact, manage their digital assets in a decentralized and trustless model without relying on a third party. More interestingly, Ethereum network, with its turing complete scripting language and trustless smart contracts, makes its easier for people to issue and digitalize their own crypto tokens which either represents some real-world asset (e.g. Digix Gold token) or has values in some platform (GolemNetwork token, Gnosis token, Augur token and so on). To date, the total market capitalization of the most popular cryptocurrency assets is 72 Billion USD[1]. This total market cap has tripled in the last 5 months and is still growing.

1.1. Motivation

1.1.1. Risk of centralization

As the Blockchain market grows and more crypto assets are being introduced, the need to convert and exchange between crypto tokens is ever increasing. The trade volume between, for example, ETH and Bitcoin is worth hundreds of million of dollars per day on major exchanges. The total trade volume between ETH and other crypto tokens on its network, most of which are less than 2 years-old, is also in the order of millions of dollars. However, despite the decentralized and trustless natures of cryptocurrencies and crypto tokens, most of the trades happening on centralized exchanges are vulnerable to internal fraud and external hacking. This is an ongoing concern and a number of hacking incidents has been reported at various exchanges[2] affecting thousands of users and loss of hundreds of million of dollars.

1.1.2. Lack of instant exchanges

Existing exchanges, including centralized and decentralized ones, often require user to wait for several minutes before allowing them to withdraw their funds.

1.1.3. Problem of existing decentralized exchanges

The quests to build decentralized exchanges have been initiated by several parties on the Ethereum network[3]. Although these parties build decentralized and trustless exchanges, they are still vulnerable to external manipulation since there is a delay when an order is created and when it is accepted in a block (read more here).

---

[1] https://coinmarketcap.com/charts/

[2] For example, MtGox, Bitfinex, Shapeshift.

[3] See 0xProject, OasisIndex and EtherDelta.

There are other possible reasons that existing decentralized exchanges are not as popular as expected despite having better security features. These exchanges keep an orderbook of users on the chain. As a result, adjustment or cancellation of bid orders can be expensive to regular users. Repeated revisions of orders will compound the issues as the cost will escalate until a match between buy and sell order is found.

Some exchanges[4] hope to resolve this issue by making the price discovery and negotiation process done offline via intermediate parties. A trade is done on-chain only after the two parties have agreed on the rate. This raises the issues of trust in the role of the intermediate party in finding the best counterparty for the trade. We also note that no-fees orders are susceptible to adversarial sybil or denial-of-service attacks.

1.1.4. The problem of having many digital assets

As the number of ICOs increases, so does the introduction of new crypto tokens. It is logical to assume that investors will acquire a variety of desired crypto tokens as part of their investment strategy. The convertibility of one crypto token to another represents a new challenge for both investors and operators alike. For example, it may be a challenge for any party to allow an already deployed contract to accept new crypto tokens as a form of payment.

It also introduces more room for implementation bugs and security flaws. As an example, recently, in the DAO Token ICO, there was a major bug that distributed more tokens to SNGLS contributors than to ETH contributors, although they contributed the same amount. Thus, there is a need to simplify the payment procedure for both token holders, merchants and users in the network.

1.2. The Kardin

We introduce Kardin, an on-chain decentralized exchange providing several useful applications, including building a practical exchange and providing payment APIs for merchants and users to instantly convert tokens effortlessly and “trustlessly”. There is no orderbook. Users will know the conversion rate before sending the transaction and receive the corresponding amount. Users don’t pay any extra fees (other than the gas fees for the transaction). Kardin benefits through pricing a reasonable spread in the conversion rate.

Our users can also send their existing token A, by converting to a different type of token B and sending it to another user, who only accepts payment in B all in one transaction. More interestingly, Kardin introduces a new standard contract wallet to allow existing contracts, which only accepts few tokens, to receive payments from any future tokens without

[4] Swap.tech and 0xproject

any modification to the contract code. This allows contracts or merchants to access to a wider class of users, receives payments and contributions in any tokens that Kardin supports.

Kardin's design has several novel constructions to support all these applications.

- Instead of maintaining a global order book, we maintain a reserve warehouse which holds an appropriate amount of crypto tokens for purposes of maintaining exchange liquidity. The reserve is directly controlled by the Kyber contract, and the contract has a conversion rate for each exchange pair of tokens by fetching from all the reserves. The rates are frequently updated by the reserve managers, and Kyber contract will select the best rate for the users. When a request to convert from token A to token B arrives, the Kyber contract checks if the correct amount of token A has been credited to the contract, then sends the corresponding amount of token B to the sender's specified address. The amount of token A, after the fees, is credited to the reserve that provides the token B.
- We introduce a new standard contract wallet to enable some of our interesting applications. Specifically, our new standard contract wallet allows the Kyber contract to send a user's newly converted tokens to his/ her destination address on the user's behalf. The destination address will receive the converted tokens as if the tokens were sent from the sender, not the Kyber contract.
- Our long-term plan also includes employing future features of the EVM language to build an efficient ZCash-Relay on Ethereum. A ZCash-Relay on Ethereum will allow us to support cross-chain trades between ETH and ZEC. We also leverage future platforms like Polkadot and Cosmos to enable more cross-chain trading and payments.
- The Kyber contract is designed with extensibility-focus which has well modularized components. Specifically, we allow dynamically adding any new tokens or delisting existing tokens. Thus, we are able to work with any tokens or digital assets in the future.

2. Kardin's Design

2.1. Actors in the Kardin

There are 5 roles for the actors in the network:

1. Users who send and receive token[5] to and from the network. Users in Kardin includes individual users, smart contract accounts and merchants.
2. A reserve entity(ies) provides liquidity to the platform. This can be our own reserve or other third party reserves that are registered by other market makers. Reserves can also be classified into public and private reserves which do and do not take contributions from the public.
3. Reserve contributors who provide capital to the reserve entity and share the platform profit. This actor only exists in public reserves which accept contributions from public to build up the reserve.

[5] For the rest of this paper a token also refers to the Ether currency.

4. Reserve manager who maintains the reserve, determines exchange rates and feeds the rates to the Kardin.
5. Kardin operator who is responsible to add and remove reserve entities, list/delist pairs of tokens in the network. Initially the Kyber team will act as the Kardin operators to bootstrap the platform in the early phases. Later on, a proper decentralized governance will be set up to take over the task.

Each of the actors interacts with the smart contract independently in a different way. The users send and receive tokens within a single transaction, without waiting for any response from the reserve or the Kardin operator. The Kardin operator is responsible for adding and removing reserves, while the reserve manager determines and feeds the exchange rates to the contract for a fixed period (several seconds basis). The main contract relies on the reserve entity to guarantee high liquidity.

2.2. Dynamic Reserve Pool

Kardin guarantees high liquidity by leveraging the existing reserves in the network.
Different reserves are directly managed by different reserve managers, which may and may not be associated to Kardin operator. Kardin allows multiple reserves to co-exist to enable better prices (by eliminating monopoly of reserve), guarantee better liquidity (by utilising

other sources). Furthermore, allowing different people, apart from Kardin operator, to manage their own reserves permits Kardin to support low-trading-volume tokens by off-loading the management efforts of those tokens to corresponding reserve managers. Thus, different parties who wish to take the risk of trading/ converting low-trading-volume tokens can create their own reserve of those tokens and register with Kardin. Note that Kardin does not hold any funds of the reserves that register with it. Their funds are stored on their reserve contracts which will follow Kardin ground principles.

When a trade/ conversion request arrives, Kardin will fetch the conversion rates from all reserves that can process the request. Kardin then selects the best rates and executes the request. We guarantee that both the reserves and the users are safe, namely we do not keep any party's funds and all transactions are atomic.

We note that when we launch Kardin, it is likely to have only a single reserve provided by us in the network. This reserve will be the main source of liquidity for the system before other reserves are registered.

Why other reserves should join Kardin? Kardin creates a platform for reserve managers to monetize their otherwise idle assets. By serving trade requests from users, reserves earn profit from the spread, which they can decide on their own. Of course the reserves can always do the trading without joining Kardin, however they will get higher volume due to network effects in Kardin. We will bring more users to Kardin by having collaborations with wallet providers and other token projects.
In addition, Kardin also provides a reserve dashboard software to help reserve managers manage their reserve portfolio. The reserve dashboard will include standard and popular trading algorithms/ strategies to allow reserve managers to automatically make prices and rebalance their portfolio. Our reserve dashboard is flexible enough that reserve managers can always implement and deploy their own strategies when and where they see fit.

How to keep the reserves safe? The security of reserves becomes a major concern in Kardin, especially for public reserves that take contributions from other members in the network. One of the primary concerns is that a bad/unethical reserve manager may quote and trade bad prices to him/herself to drain all coins from the reserve.
Let us categorize the reserves into two types: (1) private reserves which do not accept contributions and (2) *public* reserves which take external contributions and share profits with contributors. Whilst still a valid concern, if reserve managers of private reserves follow good security practices, the risk exposure of private reserves can be confined to an acceptable range, especially since the reserves are handled locally and other parties cannot interfere without permission. On the other hand, public reserves are subject to greater risk exposure due to its open nature. To mitigate the security risks of public reserves, we will employ a transparent fund management model, for example MelonFund (developed by MelonPort), so that contributors of the reserve can track all trading activities done by reserve managers. On top of that, we also plan to introduce restrictions to protect open reserves. For example, the funds of the reserves

can only be transferred to predefined addresses in the contracts, such as the reserve contract itself, and other exchanges that the reserves interact with. Hence, the risk of unwarranted extraction of funds out of the system is removed. Also, to prevent reserve managers from deliberately setting up false and unreasonable exchange rates, e.g. one million Golem Network Token (GNT) per Ether when the spot rate is only five hundred GNT to one Ether, just so that the manager can buy GNTs at a cheap price, we employ both on-chain mechanisms (e.g., prevent unreasonable changes in price without special authorization) and by off-chain mechanisms. For example, a background monitor that will halt transactions when the system detects dubious activities that undermine the integrity of the network can watch and flag out suspicious behaviours from any reserve manager in the network.

2.3. Main System Components

Kardin consists of the following major components in its system.

- Smart contracts: Kardin contains several contracts, including the main contract which serves as the main entrance to the system for users and reserve managers. We also have different contracts to maintain the reserves, and a contract wallet which provides convenient interface to all features that Kyber supports.
- User's wallet: Wallet apps with friendly interfaces to support users. Integrations with existing wallet apps like Status, Token, Metamask and so on will help improve the adoption of Kardin.
- Reserve manager portal: aids the management of the reserve by displaying their performance, network stats, supporting different strategies and algorithms to make prices/ rebalance. Reserve managers interact with the network (or the Kyber contract) via this portal.
- Operator dashboard: Helps Kardin operator manage the entire system. Operator can add and remove new reserves, change network parameters via this dashboard.

A minimum-viable-product has been released in August 2017. The readers can find more details in our release blog post [6].

2.4. Kardin APIs

Kardin supports different API commands for users, reserve and reserve contributors.

2.4.1. User API

User API can be called by any Ethereum account, including normal account and contract ones.

Transfer(amount, source tokens, destination token name, destination address)
Transfer function converts **amount** of source tokens (token A) to destination tokens (token B) and sends type B tokens to destination address. For example, users can call

[6] https://blog.kyber.network/kybernetwork-mvp-release-e8440a79346f

Transfer(100, "DGD", "Melon", "0xb794f5ea0ba39494ce839613fffba74279579268"*)*
to convert 100 DigixDao tokens to Melonport tokens and transfer all converted Melonport tokens to "0xb794f5ea0ba39494ce839613fffba74279579268".

GetExchangeRate(token A, token B)
Returns the conversion rate between token A and token B. In the future we can support different exchange rates for different trade volumes.

2.4.2. Reserve Contributor API

Reserve Contributor APIs can be called by any account in the Ethereum network, though some API only works if the account already contributed. There will be two different reserve types in Kardin: private ones which do not take public contributions and public ones which allow others to contribute funds. The APIs for public reserves highly resemble ones from MelonFund (decentralized hedge fund platform built by MelonPort). Here we list the main ones.

ContributeReserve(token type, amount)
Contribute some amount of tokens of a certain token type to the reserve. For every contribution, the contributor will receive some amount of reserve tokens/ shares to represent their contribution to the platform. We refer the readers to Melonport's greenpaper for more technical details.

WithdrawProfits()
Profits are distributed proportionally to the contributions of the contributors. The exact formula to distribute the platform profits will depend on the implementation of the reserve.

WithdrawContribution(KDN amount, token type)
An existing contributor can withdraw their contribution from the reserve. The contributor can specify in which token type that he wishes to receive for his withdrawn contribution, we do the conversion in the background.

2.4.3. Reserve Manager API

SetRate(token A , token B, rate)
To set a conversion rate between an existing pair of token A and token B. In the real deployment, this API will be replaced by a different API which updates the rates of all existing pairs in one transaction. The purpose of batch-update is mainly to reduce the gas cost.

2.4.4. Kardin Operator API

ListPair (token A, token B, initial rate)
To introduce a new pair of tokens that Kardin supports.

DelistPair (token A, token B)
To stop accepting trade between a pair of tokens.

AddReserve (reserveAddress)
Add a new reserve to the network. The reserve is managed by its own manager.

RemoveReserve (reserveAddress)
Remove an existing reserve from Kardin. The removal is due to low liquidity, bad price and other reasons.

2.5. Support trustless trading cross-chain

Chain relays, e.g. BTCRlay, enables communication between different blockchains. The launches of protocols like Polkadot and Cosmos will make cross-chain interactions even easier. Kardin will leverage these technologies to allow Ethereum accounts to receive payments from different cryptocurrencies.

3. System Properties

3.1. Trustless and secure

The Kardin operator does not hold the tokens of the users. Hence, by design, user's tokens are secured from theft losses. Users need not trust the intentions of the reserve entity and the KDN token holders, as the integrity of the operator is enforced/ensured by the smart contract.

3.2. Instant trade

An exchange or convert request is executed immediately within a single transaction. Users get their exchanged token at the exact moment they transferred their original token. No deposit or confirmation or waiting time is needed. This efficient and user friendly feature distinguishes Kardin from most other existing and future exchanges.

3.3. On-chain exchange

The exchange runs on chain and is accessible for all accounts, including normal accounts and smart contracts. That allows smart contracts to directly interact with the exchange without a third party intervention to receive funds/ payments from different tokens that they do not support

originally. This feature enables us Kardin to be an on-chain proxy payment platform for all accounts, including normal accounts and smart contracts.

3.4. Compatibility

Kardin does not require any modification in the underlying protocol of Ethereum and existing smart contracts to function. Our payment API can communicate with existing contracts without any change on their side.

That said, we also introduce a new contract wallet that holds all user Ether and tokens. The wallet allows the user to pay with token A to a contract that expects token B, where the conversion from A to B is seamlessly done by the Kardin. The receiver will receive the payment as if it was sent by the original user.

3.5. Comparison to existing systems

We compare Kardin to existing systems in the table below. We left out Bancor intentionally as they claim (from our private conversation) to be a platform that focus on community tokens, rather than general purpose exchange.

Exchange	Trading Cost [7]	Trustless	Instant Trades	On-chain	Guaranteed Liquidity	SecureAgainst Attacks
Kraken/Poloniex	Low	No	No	No	Yes	No
Shapeshift	Low	No	Yes	No	Yes	No
Coinbase	Low	No	Yes	No	Yes	No
EtherDelta Oasis Index	High	Yes	No	Yes	No	Yes
Swap.tech 0xProject	Low Low	Somewh-at[8]	No	Hybrid	No	Not sure[9]
Kardin	Low	Yes	Yes	Yes	Yes	Yes

[7] Cost to execute a trade, apart from the trading fees.
[8] Users need to trust the relays to match them the best counterparties
[9] Attackers can create fake orders without any cost. No guarantee that a trade can be settled.

4. Applications

4.1. Instant and secure exchange

First and foremost, Kardin is an exchange. Unlike most exchanges, however, Kardin performs trade requests instantly. Moreover, Kardin does not hold users' tokens, thus any theft or loss of tokens is prevented by design.

This contrasts sharply to most exchanges where confirmation time of several minutes is typically needed. Any malfunction during that period could potentially result in inconvenience or in the worst case scenario, loss of funds.

4.2. Generic payment APIs with any token

Conducting an exchange over a smart contract allows user to pay for any service or product with any crypto token they prefer. The contract will provide instant conversion to Ether and securely pay on behalf of the user to any contract he wishes. The figure below describes how a user could participate in an ICO that accepts only Ether with any token. The entire process occurs within a single transaction, and the Kardin never has a possession on the user tokens (neither token A nor token B).

4.3. Trusted on-chain source for rate quotes

Kardin exchange rates are visible to other smart contracts. Hence, it enables the implementation of advanced financial instruments such as swap contracts. The quotes provided by Kardin are secure as they reflect the real rates which are being used to trade between pairs of tokens.

4.4. Mitigate the risks of price fluctuations

Due to the illiquidity of crypto assets, the exchange rates often seem too volatile due to irregular demand and supply. This issue is aggravated further due to lack of parties that are willing to warehouse crypto-assets. The lack of options means now that it is almost impossible for users of crypto assets to hedge themselves for future requirements. The Kardin will be addressing this challenge by introducing derivatives in the forms of forwards and options to provide more alternatives to users.

4.5. Forwards

A forward is a contract whereby parties agree to trade an asset at a later date at a price specified in the present. One of the common problems as ICOs become mainstream is the need for some users to convert between tokens, such as from Melon to ETH, in preparation to participate in an upcoming ICO. The user could either acquire ETH at current market rate or commit to a forward contract to negate the risk of the price fluctuations in the ETH as an viable alternative.

4.6. Options

Options contracts allow users to hedge against adverse price movement for a fee called premium. A call option gives the owner of the contract the right to purchase the crypto asset at an agreed price. A put option is an opposite. The premium is calculated using the implied volatility of the underlying crypto asset.
The user of crypto assets that need to prepare for a future purchase or sale commitments can pay a premium to buy a call or put option. As an example, holders of iced tokens are able to write call options to earn premiums while forgoing the upside of the price.

5. Road map

The road map of Kardin includes several phases.

5.1. Phase 0: Testnet deployment

Est. delivery: August 2017
Develop an MVP version of our platform, including the Kardin wallet, the main Kardin contract and our reserve dashboard. The purpose of this phase is to create a basic and functional version of Kardin with all the main functionalities and applications.
The MVP will be released publicly, and the related contracts will be deployed and tested on the Ethereum testnet.

5.2. Phase 1: Basic mainnet deployment

Est. delivery: Q1 2018

We deploy the first version of Kardin on the mainnet. We start off with supporting trades and proxy payments between any tokens to and from Ether. It is likely that our reserve will be the the main one that serves all the trades, though we plan to partner with big token holders and other market makers to introduce their reserve in Kardin. The tokens that we support will be the popular tokens that have high demand and high trading volumes in the market.

We will also partner with wallet providers like MyEtherWallet, Status, Jaxx and others to implement the core features of Kardin. Since most users stay with their favourite wallet, bringing our features to the wallets is the best way to increase the adoption of Kardin.

5.3. Phase 2: Supporting arbitrary pairs of tokens

Est. delivery: Q2 2018

This phase can be easily achieved following the smooth implementation of phase 1. By then, we expect to have more reserves (i.e. market makers) to join Kardin. The number of supporting tokens will increase as we can get more reserves in our platform.

Kardin will also work with other strategic partners to build APIs to allow users in their platforms to efficiently withdraw tokens/ shared fees in prefered tokens. For example, many platforms, projects are employing the fees sharing model in which token holders share all the platform fees (which may be spread in many tokens) collected from platform users. Token holders in these platforms may get their shared fees in, for example, ETH seamlessly via Kardin if these platforms use our related APIs.

5.4. Phase 3: Trading advanced financial instruments

Est. delivery: Q4 2018

Once our development and operations stabilizes, we will deploy phase 3 of Kardin, in which we support trading advanced financial instruments as discussed in Section 4.

We plan to work with decentralized hedge fund platforms (e.g. provided by Melonport) that allow people to invest in trustless hedge fund and get the profit share from efficient fund managements. Our team needs to discuss, exchange and build APIs between the related platforms to enable us to do what we aim to do in a secure way. Similarly, collaborations with ICOs projects that have vesting schemes for their founders, advisors are also important.

5.5. Phase 4: Support cross-chain trades

Est. delivery: End 2018/ Early 2019

The deployment in this phase allows users to trade between Ether/ tokens to Bitcoin, ZCash, ETC and so on. There are two ways to enable this goal: using chain relays (e.g. BTCRelay and ZecRelay) or using interchain communication protocols (e.g. Cosmos, Polkadot). We will watch

the development of these protocols and relays closely to decide which solution will be employed in Kardin.

6. Crowdsale and the Kardin

A fixed number of Kardin tokens (KDN) will be distributed to the public in exchange for Ether contribution. The details of how many KDN are distributed, and how the sale is conducted will be publicly available in our blog posts and website.

6.1. Use of tokens

Kardin tokens (CDN) are required for reserves to participate in the network and to reward various parties who will help generate more trading activities in the platform. Kardin will rely on various partners, including both software and hardware wallets, blockchain explorers, and on-chain smart contracts to direct users to the platform. These partners will be paid in KDN for every trade that they introduce to Kardin.

Before operating, Kardin reserves need to pre-purchase and store KDN tokens. In every trade, a small fraction (exact numbers are TBD) of the trade volume will be paid by the reserve to Kardin platform in KDN. This small fee represents the reserve's payment in return for the right to be able to operate and earn profits from trading activities in Kardin. The collected KDN tokens from the fees, after paying for the operation expenses and to the supporting partners, will be ***burned***, i.e. taken out of circulation. The burning of tokens could potentially increase the appreciation of the remaining KDN tokens as the total supply in circulation reduces. In order to determine the network fees, the conversion rate between KDN and ETH will be updated frequently to the Kyber contract by KDN operators, based on the trading rates on various exchanges.

As an example, for a trade volume of 10 ETH with a 0.01% fee, a corresponding 0.001 ETH worth of KDN will be paid by the chosen reserve to Kardin as a fee for the use of the reserve dashboard and access to network users. Suppose the rate of KDN at the trading time is 1 KDN for 0.1 ETH, the reserve needs to pay 0.01 KDN to the Kyber platform. The wallet/ website that helped the user initiate the trade will get, supposedly, 5% of the fees, or 0.0005 KDN. The remaining 95% of the fees, or 0.0095 KDN will be burned forever.

This approach would increase the demand of existing KDN tokens as the trading volume happening on Kardin increases. The approach also properly rewards all participants who help grow the ecosystem. KDN token holders can easily track the total supply by reading from the contract, without relying on any off-chain accounting firm.