CrypticOcean

BitPayer Decentralized Interoperability Exchange

SATOSHI CENTRE

DISCOVER | SUPPORT | INNOVATE

https://www.crypticocean.com

INDEX

7. Governance model

- Elements of blockchain governances
- Types of governance
- Governance tokens

8.Technical aspects of DEX

- Peer – to – peer trading

9.Market research

10. Competitor analysis

- Major challenges
- An Ideal DEX should have the following features

11. DevOps engineering

- Public and private subnets
- Interfacing components
- External services that are part of the program
- Public and private load balancers
- Web servers
- Service types
- Databases
- Docker hub and ansible
- Types of web traffic
- Public API
- Audit service
- Graph service
- Ledger service

Introduction

A decentralized exchange [DEX] is a cryptocurrency exchange where trading of cryptocurrencies and other digital assets takes place in a decentralized way without a central agency or intermediary. Decentralized exchanges allow peer-to-peer trading of cryptocurrencies directly between users through an automated process. They are generally established by creating proxy tokens which represent a certain fiat or cryptocurrency or through a decentralized multi-signature escrow system. The absence of a centralized server, which acts as a single point of failure in a traditional cryptocurrency exchange increases the security of the trading process.

Decentralized exchanges are generally non-custodial , offer great level of privacy and no risk of server downtime. DEXes never take custody of funds. They allow users to retain control of their funds and no central authorities can freeze the assets. Because users do not need to transfer their assets to the exchange, they reduce the risk of theft from hacking , offering a great level of trust. With respect to privacy, Dexes do not require KYC (Know your customer) or registration requirements for using the exchange beyond having a wallet address. Since the hosting of DEXes is distributed throughout the nodes involved in the network, they reduce the risk of server downtime.

Types of decentralized exchanges

At present, DEXes are developed in one of the three modes:

- On-chain order books and settlements
- Off-chain order books with on-chain settlement
- Smart contract-managed reserves

1.On-chain order books and settlements

These are entirely blockchain based and represent the first generation of DEXes. Every new order updates the state of the blockchain. Although it protects user privacy and security this form of DEX makes exchanges illiquid, slow, expensive and unable to operate with other DEXes.

2.Off-chain order books with on-chain settlement

Execution of the trades happens on the blockchain, where users have control of their funds until the exchange takes place. The order books are hosted by third party-services called relayers . This enables the exchange to maintain liquidity and create a more robust infrastructure for traders.

3.Smart contract-managed reserves

This model connects the buyer and seller function when there is low liquidity. With smart contract-managed reserves, instead of having to find a buyer for the bitcoin, a user can trade with an external reserve, depositing bitcoin into the reserve and receiving ether in return.

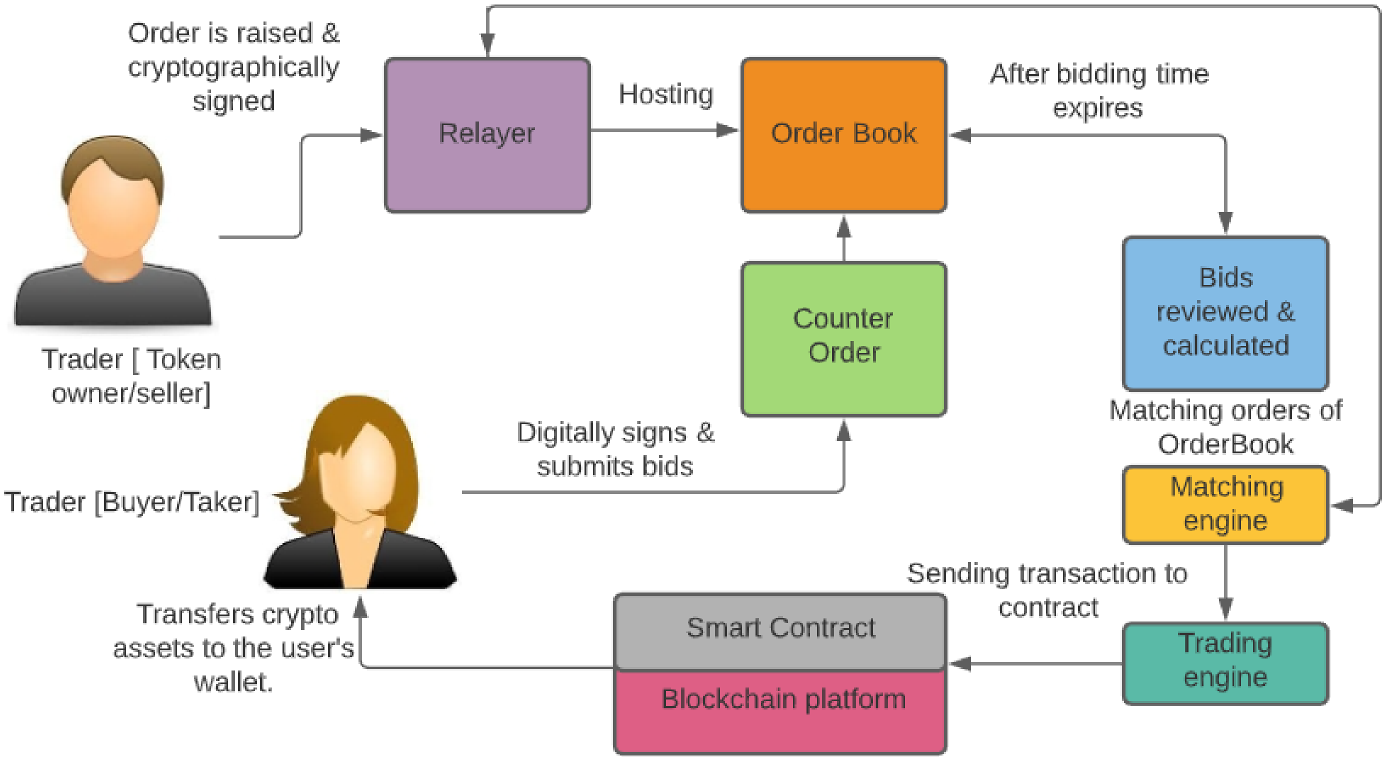
0X Protocol

The 0X protocol is an open protocol that offers low friction peer-to-peer exchange of ERC 20/ERC-721 tokens and serves as an open standard among decentralized applications. It is built on the Ethereum blockchain. The Ox protocol is an example of both off-chain order relay and on-chain settlement modes. The users compile the parameters of a trade into a block of data and then cryptographically sign with their private key but do not broadcast to the blockchain. Rather, they sent the block of data to a specified counterparty directly or send it to a relayer who include in the orderbook. Although relayers have access to cryptographically signed messages, they cannot have control of user assets.

Working of a decentralized exchange

A DEX functions as a decentralized peer-to-peer network. Any node in the network can function both as a trader and a server at the same time.

1. A token owner raises an order to exchange his/her assets/funds with another node available on the DEX. The owner specifies the number of units they want to sell, the cost of each token, and the time for bidding of their assets as a request to the relayer, who is responsible for hosting the order book.
2. Once the selling order is established , other users in the network can digitally sign and submit bids to create a counterorder.
3. Once the time assigned by the seller for bidding expires, all the bids are reviewed and calculated.
4. The orders are then sent to a smart contract in the blockchain, which carries out the transaction and then transfers the crypto assets to the users' wallet.


Order is raised & cryptographically signed
Relayer
Hosting
Order Book
After bidding time expires
Trader [Token owner/seller]
Counter Order
Bids reviewed & calculated
Matching orders of OrderBook
Digitally signs & submits bids
Trader [Buyer/Taker]
Matching engine
Sending transaction to contract
Smart Contract
Blockchain platform
Trading engine
Transfers crypto assets to the user's wallet.

2.1 - WORKING OF A DECENTRALIZED EXCHANGE

Entities involved in a DEX

1.**Traders/Users**

Traders/users are the main players in a DEX. They interact with others users and the DEX through a trading screen , where the user can buy or sell an order. Each trader/user is considered as a node in the network.

2.**Relayer**

Relayer hosts the database of cryptographically signed trade orders and is responsible for collecting them. The collected orders are then represented in the form of an orderbook, which is hosted off-chain.

3.**Matching Engine**

Matching Engine helps in matching orders of the orderbook. Buyers and sellers are paired together on basis of mutual agreements. Filling of orders generally through an automatic matching process ,where an underlying algorithms helps in pairing and executing of orders.

4.**Trading Engine**

Trading engine helps in transfer of transactions to exchange contract.

5.**Smart Contract**

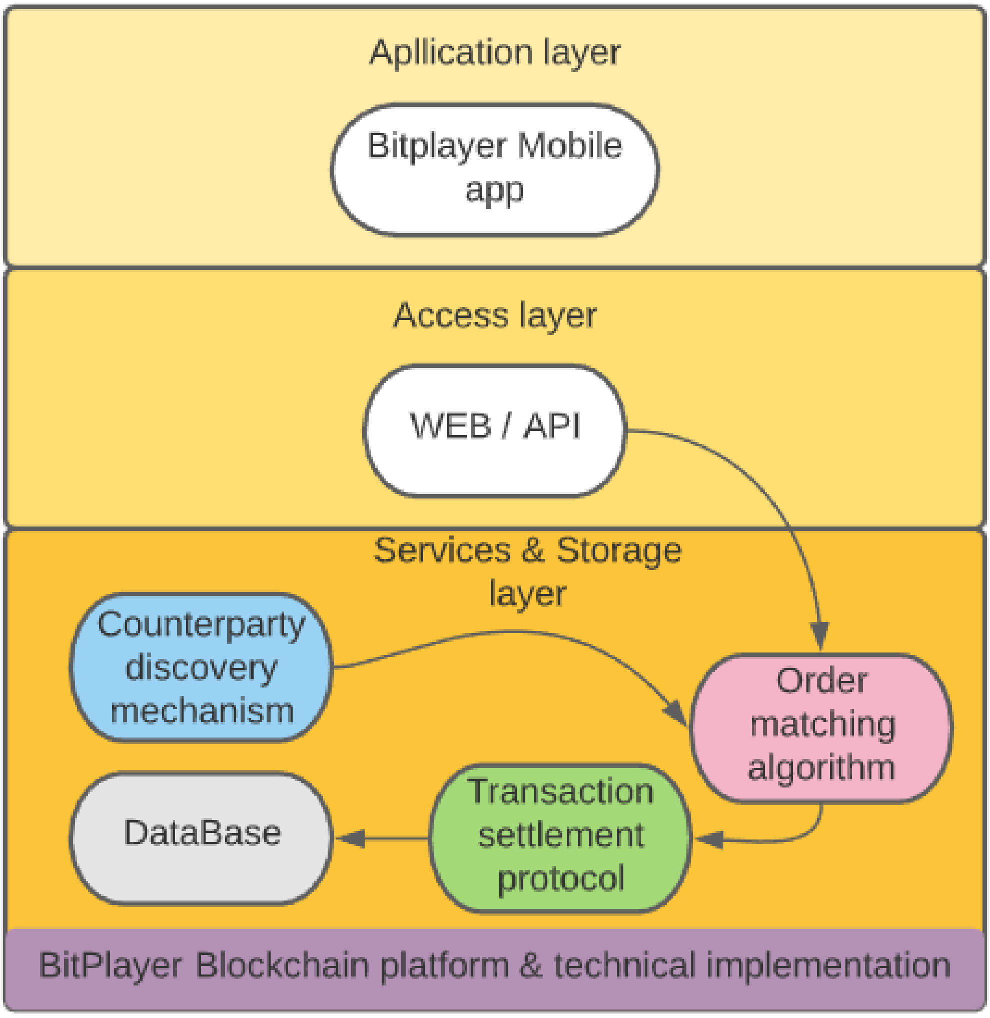
A smart contract is a transaction protocol or the underlying - logic with the terms of agreements between the buyer and seller . It helps in swapping tokens after validating signature and deposits assets, from which the users making the trade can withdraw them.

Architecture of a DEX

A DEX application builds on top of a decentralized exchange protocol, and adds an on-chain or off-chain order book database and a graphic user interface (GUI) and APIs so that the information is easily accessible. Overall, a decentralized exchange application can be broken down into the following components:

1.The blockchain platform & technical implementation
2.The counterparty discovery mechanism

3.The order matching algorithm
4.The transaction settlement protocol


Apllication layer
Bitplayer Mobile app
Access layer
WEB / API
Services & Storage layer
Counterparty discovery mechanism
Order matching algorithm
DataBase
Transaction settlement protocol
BitPlayer Blockchain platform & technical implementation

2.2 - ARCHITECTURE OF DEX

A decentralized exchange protocol generally describes a software program, hosted on or integrated into one or more distributed ledgers like Ethereum, thatenables peer-to-peer transactions that are automatically settled on the distributed ledger. process. Users retain sole custody of their private keys throughout the transaction process.

Modules of bitpayer exchange

Entities that will interact
with Bit Player DEX :

- Users/traders
- Admins

User module

Users can:

- Place orders
  1. Select tokens
  2. Set price
  3. Sign orders
  4. Set bidding time period
- Manage accounts
- Manage wallet details
- Manage open orders
  1. Modify open orders
  2. Cancel open orders
  3. Status of openorders

The liquidity and market depth of decentralized exchanges are quite low. In order to increase the liquidity, users should be able to interact with DEXes seamlessly. A good user experience and user interface design is required.

Admin module

- Manage smart contracts
    1. Deploy new exchange contract
    2. Remove exchange contract
    3. Deploy New registry contract
    4. Remove registry contract
- Manage token listing
    1. Introduce and create new tokens
    2. Remove existingtokens
- Manage users
    1. Verify KYC [formanual filling]
    2. Invite new users
- Check statistics
    1. Successful orders·
    2. Unsuccessful orders
    3. Transaction fees collected
    4. No of daily users·
    5. Volume of trade
    6. Liquidity
- Manage fee structure
- Manage fee wallet

User flow

- Users and traders are initially required to register on the DEX platform and all the basic details like mail-id, contact number and National ID are collected.
- Once registration is done, Admin will verify KYC documents either manually or through a third-party service.
- After KYC verification the user will be able to sign in to the trading platform.
- The user will be provided with a manual which has all the guidelines and Terms & conditions which include creation of order, cancellation, displaying of order history and access to open orders. Tutorials for “how to connect the wallet with DEX” is also provided. **An exclusive online classroom for Bit Player coins will be provide where users can access different learning materials, discussideas and get connected with mentors.**

Order creation

- After connecting with the wallet ,the user will approve an XXXX contract to spend BPY tokens which will be collected in the form of fees or traded.
- An order is created by filling in parameters like price , amount and bidding time. DEX will automatically verify if a user/trader has enough balance to trade tokens. The order is then cryptographically signed and sent to matching engine. The order is created and sent to the order book by the relayer.
- If the order gets matched, it is sent to the trading engine where the orders are further sent to exchange smart contracts to fill in orders, else it remains in the order queue.
- YYYY function of exchange smart contract will be the entry point of exchange smart contract, actual transfer of assets will be take place at XXXXX smart contract by decoding asset specific metadata contained within an order.
- YYYY function of exchange smart contract will emit an event if transaction is executed or revert, Transaction success message will correspondingly notify counter parties by email, desktop notification and message on registered phone no with transaction details and updated balance of asset.

Order cancellation

- User will be able to cancel all open orders from his/her own account, by simply selecting order from open orders and 'confirm' cancel order.
- This will remove the order from order book only and will not send cancel order transaction to exchange smart contracts because the signature is not exposed.

AuBlock - Gamified auction platform

- Besides creation and cancellation of orders, a gamified auction platform is also provided allowing users and BPT token holders to enter raffle pools and bid with other users to buy cryptocurrencies cheaper than the market price.

Investment platform for agriculture

- The platform will also support farmers and small communal economies dependent on agriculture through Initial Exchange Offerings , wherein coin holders can submit a project. The exchange's fundraising platform will directly organize a fund raising event .The community members are provided with an option to vote. Selected projects receive the requested financial support from the community budget and pledge part of their revenue back to the budget with an interest rate.

Admin flow

Manage smart contracts & token listing

Admin can:

- Deploy the exchange smart contract.
- Deploy ERC20 Proxy contract, Deploy ERC721Proxy contract (XXX Contract)
- Call function XXXX of Exchange contract and input address of ERC721Proxy Contract, similarly call function YYYY of Exchange contract and input address of ERC20 Proxy Contract.
- Call function XXX of ERC20 Proxy contract and input address of exchange contract, similarly call function XXXX of ERC721Proxy contract and input address of exchange contract.
- Every time XXXX function of exchange smart contract is called by admin, exchange will internally call XXXX contract to call transfer from function of corresponding asset.

Manage users

- Once a user registers with the DEX and fills in KYC details, the admin has to verify and approve the registration.

Check statistics

- The admin can keep an account of the statistics of successful and unsuccessful orders, the transaction fees collected, the number of daily users and the volume of trade.

Promoting members of the community

- A page will be created on our website where members of the community, their stories ,and future aspirations are promoted. The admin's role is to identify and promote the stories of token holders.

DeFi

Decentralized finance refers to a wide variety of financial applications in blockchain geared towards disrupting financial intermediaries. DeFi is one of the fastest-growing sectors in blockchain technology. With a huge upsurge in the adoption of DeFi with regards to cryptocurrency, it has established itself as a complete game-changer. In comparison to centralized banks, DeFi has been projected as a secure, confidential, transparent, and permission less solution to help communities earn higher profits and accelerate their monetary growth. The decentralized nature of this ecosystem empowers investors to make deals without the intervention of central authority, corporations, or agencies that monitor and approve the business functions. Smart contracts are employed to make sure the deals are executed appropriately.

DeFi draws inspiration from blockchain, the technology behind the digital currency bitcoin, which allows several entities to hold a copy of a history of transactions, meaning it isn't controlled by a single, central source. That's important because centralized systems and human gatekeepers can limit the speed and sophistication of transactions while offering users less direct control over their money. DeFi is distinct because it expands the use of blockchain from simple value transfer to more complex financial use cases.

Direct purchases aren't the only type of transaction or contract overseen by big companies; financial applications such as loans, insurance, crowdfunding, derivatives, betting and more are also in their control. Cutting out middlemen from all kinds of transactions is one of the primary advantages of DeFi.

Benefits of DeFi

Decentralized finance leverages key principles of the Ethereum blockchain to increase financial security and transparency, unlock liquidity and growth opportunities, and support an integrated and standardized economic system.

- **Programmability-**Highly programmable smart contracts automate execution and enable the creation of new financial instruments and digital assets.
- **Immutability-**Tamper-proof data coordination across a blockchain's decentralized architecture increases security and auditability.

- **Interoperability-**Ethereum's composable software stack ensures that DeFi protocols and applications are built to integrate and complement one another. With DeFi , developers and product teams have the flexibility to build on top of existing protocols, customize interfaces, and integrate third-party applications. For this reason, people often call DeFi protocols "money legos."

- **Transparency-**On the public Ethereum blockchain, every transaction is broadcast to and verified by other users on the network. This level of transparency around transaction data not only allows for rich data analysis but also ensures that network activity is available to any user. Ethereum and the DeFi protocols running on it are also built with open source code that is available for anyone to view, audit, and build upon.

- **Permissionless-**Unlike traditional finance, DeFi is defined by its open, permissionless access:anyone with a crypto wallet and an Internet connection, regardless of their geography and often without any minimum amount of funds required, can access DeFi applications built on Ethereum.

- **Self-Custody-**By using Web3 wallets like MetaMask to interact with permissionless financial applications and protocols, DeFi market participants always keep custody of their assets and control of their personal data.

Use cases of DeFi applications

- **Decentralized exchanges (DEXs):** Online exchanges help users exchange currencies for other currencies, whether U.S. dollars for bitcoin or ether for DAI. DEXs are a hot type of exchange, which connects users directly so they can trade cryptocurrencies with one another without trusting an intermediary with their money.

- **StableCoins:** Another form of DeFi is the stablecoin. A cryptocurrency that's tied to an asset outside of cryptocurrency to stabilize the price. Cryptocurrencies often experience sharper price fluctuations than fiat, which isn't a good quality for traders who want to know how much their money will be worth a week from now. Stable Coins peg cryptocurrencies to non-cryptocurrencies, such as the U.S. dollars, in order to keep the price under control.

- **Lending platforms:** These platforms use smart contracts to replace intermediaries such as banks that manage lending in the middle.

- **Yield farming:** For knowledgeable traders who are willing to take on risk, there's yield farming, where users scan through various DeFi tokens in search of opportunities for larger returns.

- **Liquidity mining:** When DeFi applications entice users to their platform by giving them free tokens. This has been the buzziest form of yield farming yet.

Governance model

To ensure transparency in the system, DeFi platforms follow a democratic governance model based on a liquidity pool. In such a model, the power to make decisions is distributed among the nodes. DeFi platforms provide governance tokens to investors. While the platform is developed by a specialized team, the ultimate aim is to pass over the authority to the token holders.

“Governance” is a structure that every user or node agrees to follow. Its core purpose is to meet the user's needs with available resources as efficiently as possible and achieve the long-term sustainability of the digital structure.

Blockchain governance usually involves four central communities, though to what degree each is involved varies from blockchain to blockchain.

These communities include:

- **Core developers –** They are responsible for maintaining the main code underpinning the blockchain. Though they can add or remove code to modify the central code, they can’t put it into effect network-wide.
- **Node operators –** Node operators have a full copy of the blockchain ledger and runs it on their computers. They can decide whether to implement a feature on their nodes or not. Code developers are dependent on node operators to agree on their provided features.
- **Token Holders –** These are the user and entities who hold the blockchain token. Depending on the various blockchains they have various degrees of voting rights on what feature to implement, set prices, etc. Generally, investors form the major part of the main token holder community.
- **The Blockchain Team –** It can be a firm or a non-profit organization that takes on various roles. The primary role is to steer the fund and project development. They also represent the broader communities of investors and supporters to negotiate with code developers and node operators, as well as often take on a marketing role.

Elements of blockchain governances

There are multiple ways governance methods can be categorized. In the case of blockchain, identifying the primary categories is crucial to evaluate and develop an effective blockchain governance structure.
The elements of blockchain include:

- Consensus
- Incentives
- Information
- Governing Structure

Types of governance

Governance can be categorized into two types –

- **Off-Chain Governance:** Off-Chain governance usually promotes a balance between a blockchain community, e.g. its core developers, miners, users, and business organizations. Bitcoin and Ethereum both follow this governance model. This type of governance model resembles the traditional structure of governance, but there are some similarities and also some dissimilarities with traditional governance models.

- **On-Chain Governance**: This is another governance structure explicitly created for blockchain, unlike other Off-Chain methods. It is far more democratic in nature. The direct democracy in On-Chain governance is achieved because of blockchain's built-in voting mechanism, which can be optimized as per the specific requirement of a network.

Governance tokens

Governance tokens give holders the right to influence the existing strategies and create new ones. One of the biggest advantages of these tokens is that they give the holder a direct stake in decentralized finance platforms. Those who receive tokens have the influence to decide how platform economics transform over time.

So far, DeFi governance tokens have been used to vote on various proposals like which assets are supported, collateralization levels for certain assets, and where protocol fees should be directed. Concurrently, governance tokens are used as a method of increasing the incentives available to investors. A strong, comprehensive, and progressive governance model is a prerequisite requirement.

In the first phase, developers will put together the code to create result-oriented strategies and will carefully monitor for time-sensitive incidents. At the same time, to begin the process of democratization, the developers will launch a liquidity mining reward program for the community [Farming community]. This means that a significant part of the BPY tokens goes to the community. Once this phase is completed, by distributing BPY token, the developers will move forward towards the process of progressive decentralization. This would help developers to achieve their visionary goal of a 100% community-owned platform.

The BPY governance token will give the community complete empowerment and will get the holder’s right to initiate a proposal on platform parameters, such as strategy risk score changes or even yield allocation, and vote on the same.

The community can propose the following changes:

1. adding new assets to the platform,
2. remove an asset,
3. changing an asset’s interest rate.

The final decision will be taken based on the majority of the votes. This true democratic spirit will be reflected in each decision taken collectively by the pooled community.

The community will have the power to evaluate risk scores for each strategy and distribute funds based on risk rating.

The community can control the following economic resources:

1. proposing and voting on risk scores for strategies.
2. determining risk tolerance levels.
3. adding new yield strategies.
4. allowing changes to the strategies.
5. incentive structure

Technical aspects of DEX

A Dex has the following features:

- Off-chain order book
- On chain settlement
- Off chain order matching, order cancel
- Efficient matching engine
- Secure trading engine
- User friendly UI [in the form of a mobile appln]
- Admin dashboard with all the features listed in admin module
- Customised graphs and statistics
- Customised incentive programs for users
- Community spotlight
- Gamified auction platform
- Mentorship portal
- Investment platform
- Additions features for crowd sale using DEX(using native token of DEX or any other token pair listed)

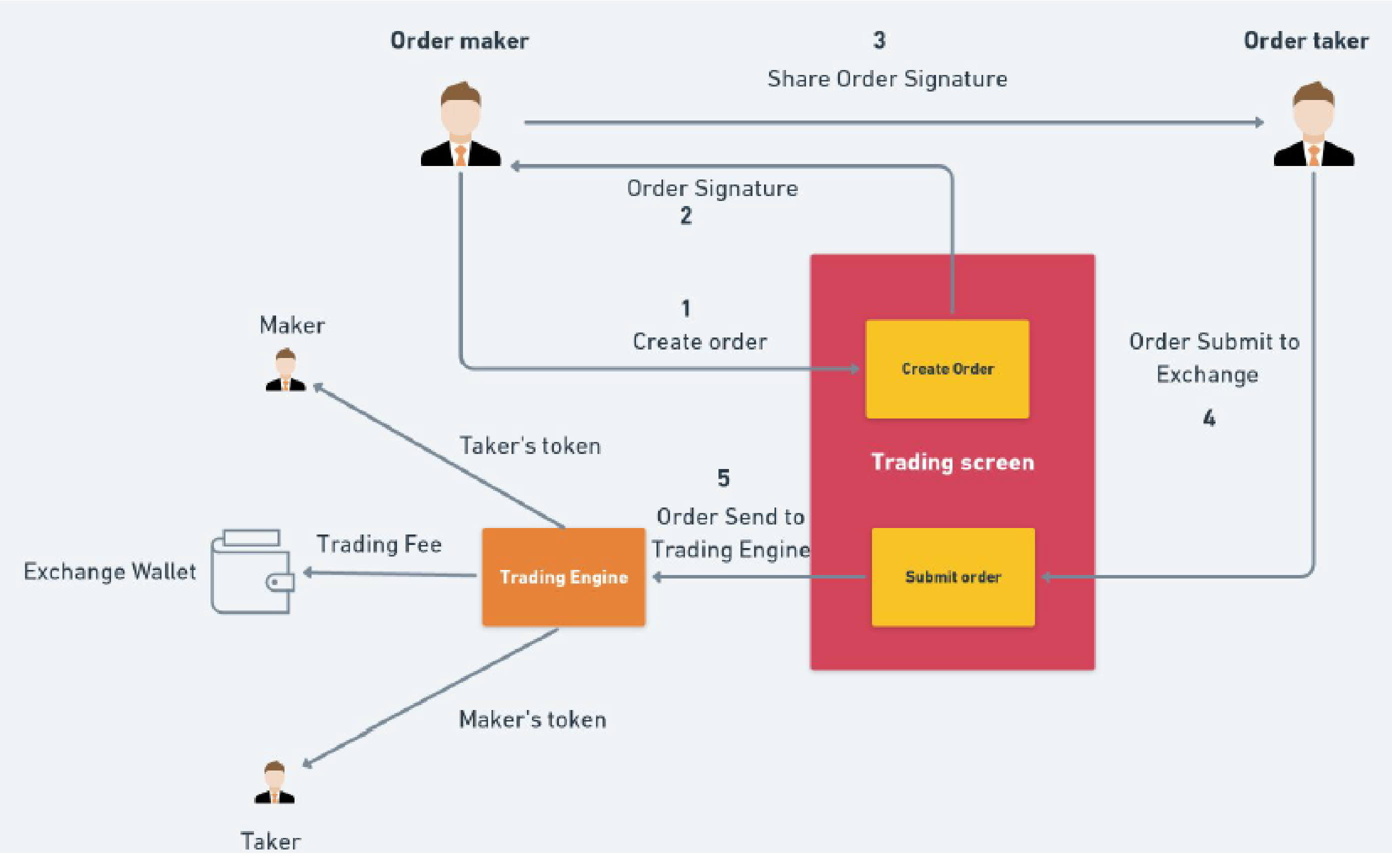
Peer – to – peer trading

Peer to peer trading will be possible with DEX, where anyone can make an offer (maker) and send directly to taker also while making offer for peer to peer trading specific taker address should be mentioned while creating an offer hash and signing it, and afterwards offer will be submitted by taker to DEX for trading, to fill order using 0X protocol.

- Maker will create an offer.
- Maker got the signature and details of order through mail and will be also saved in order history.
- Maker will share order signature with any source of medium.
- Taker will submit order to DEX and order signature will be validated before sending to trading engine, only valid open orders will be sent to trading engine.

- Once the order is validated, offer will be sent to trading engine to fill the order, taker will get makers assets and maker will get takers assets and fees will be sent to exchange wallet.

- All the necessary precautions and validations will be considered before sending order to trading engine.



8.1 - PEER-TO-PEER TRADING

Market research

Normally the centralized exchange uses the concept of hot wallets (The wallets which are directly connected with the internet) and cold wallets(The wallets which are not connected with the internet).The hot wallets are more prone to hacking since it is connected with the internet. If the centralized exchange is hacked then simultaneously the hacker will get access to all the keys of the wallets of the users and the exchange funds are hacked within a second.

To overcome this phenomenon, the concept of decentralized exchange came into the picture. In the decentralized exchange environment, user has their own key and it is stored in a mush safer environment.

An analysis of some of the top grossing decentralized exchanges which are currently ruling the market has been done. A synopsis on what protocol they are using, what kind of blockchain has been used in those platforms and what are the advantages and disadvantages associated with it are listed below.

1. Bartardex

Built on: Komodo Blockchain

Technology/Protocol: Iguna core

Advantage: Supports atomic swaps/ Cross chain trading[Atomic swap is a service where one user can exchange their cryptocurrencies with other users without any third party exchange. They can be executed between different blockchains on any native coins or they can be executed on off chain channels of the main blockchain].

Disadvantage: The UI (User Interface) of the exchange is not that user-friendly.

2. Nash

Built on: Neo Blockchain

Technology/Protocol: Off-chain matching

Advantage: One of the main disadvantage associated with the decentralized exchange is the speed. Nash is comparatively faster than its competitors.

Disadvantage: It is relatively new in the market so not quite tried and tested.

3. Waves DEX

Built on: Waves Blockchain

Technology/Protocol: Waves Protocol

Advantage: One of the main advantages associated with this decentralized exchange is the addition of the custom tokens through plugins. This makes it one of the flexible decentralized exchanges available in the market.

Disadvantage: Since Waves DEX allows custom token creation and addition, there are chances of drawbacks when someone is using low liquidity and highly volatile tokens.

4. IDEX

Built on: Ethereum Blockchain

Technology/Protocol: IDEX Protocol

Advantage: The exchange balance updates in real time -The exchange's architecture design allows users to trade continuously across multiple markets without waiting for transactions to mine. It helps users to place true market orders and fill multiple orders in one shot and cancel orders without any gas costs.

Disadvantage: Only supports ERC 20 tokens.

5. Open shares DEX

Built on: Bitshares Web wallet & Open shares DEX

Technology/Protocol: Bitshares

Advantage: -The exchange allows access to deposit, withdrawal and storing of both crypto and fiat funds.
The exchange support storing, sending and trading of around 125 cryptocurrencies with 63 pairs. Allows withdrawing of new currency to other exchanges.

Disadvantage: Fees structure is a bit high.

Reason why decentralized exchanges have got a competitive edge over centralized exchanges :

- **Top notch Security:** One of the biggest advantages of Decentralized exchange over centralized ones is the security which it posses. The Decentralized exchange does not depend on any third party services. The entire control of the wallet remains in the hands of the users.

- **No Risk of Identity and information theft:** Users who want to trade in decentralized exchanges don’t have to submit the government proofs like (e.g.: Passport, Driving License) etc. So the users don’t feel the headache of losing the confidential info while registering for trading in decentralized exchanges.

- **No Infrastructure Risk:** The decentralized exchange doesn't have any kind of infrastructural risk while executing trade orders. This uniqueness allows the user the flexibility to trade anytime without any major headache.

- **No Risk of banking information theft:** As decentralized exchanges only work through cryptos so there is no risk of sharing banking information over decentralized exchanges.

- **No Risk in Government shutdown:** The government can't interfere in Decentralized exchanges so the government can't take any drastic action against any decentralized exchanges.

- **Anonymous transaction:** Every transaction associated with the decentralized exchange is entirely anonymous and every transaction is being verified and validated by the almighty Blockchain "smart contract". So the privacy of the transactions treated in the most deserving way.

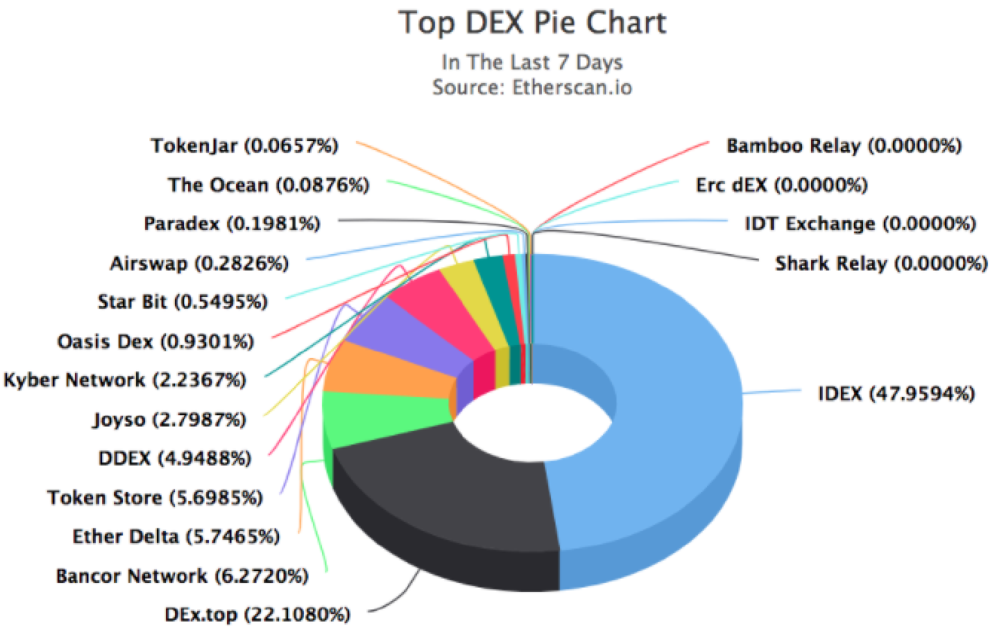
Research analysis

Decentralized exchange is a new phenomenon under the Blockchain but still, it has to go a long way before it can stamp its entire authority over the trading market. There is no denying fact that it is highly popular among the crypto enthusiasts and a lot of top-notch centralized exchanges and market leaders in the trading market are launching their own decentralized exchanges

Competitor analysis

Centralized exchanges have serious drawbacks, perhaps most notably the exposure of users funds to theft. But the trends of creation of decentralized exchanges that place users funds in their control is growing rapidly. Decentralized exchanges will most likely become a key piece of infrastructure for the crypto industry in the future.

There are currently over 250 DEXs and over 30 DEX protocols. A DEX protocol is not exactly a DEX itself, but rather provides teams with the tools they need to build a DEX, The most popular DEX protocol is 0x. The standard feature that makes DEXs “decentralized” is that they do not custody customer assets. Rather, users are responsible forholding assets in their respective wallets, which should reduce the risk of being hacked.



10.1 - (Source: https://github.com/distribuyed/index)

The main differences lies in where the orderbook is hosted, how orders are created, modified, matched, and cancelled, and how transactions are executed. This determines the architecture and where DEXs lie on the decentralization spectrum. For example, a DEX could be fully decentralized and submit every order creation, modification, cancellation, and settlement to the blockchain.

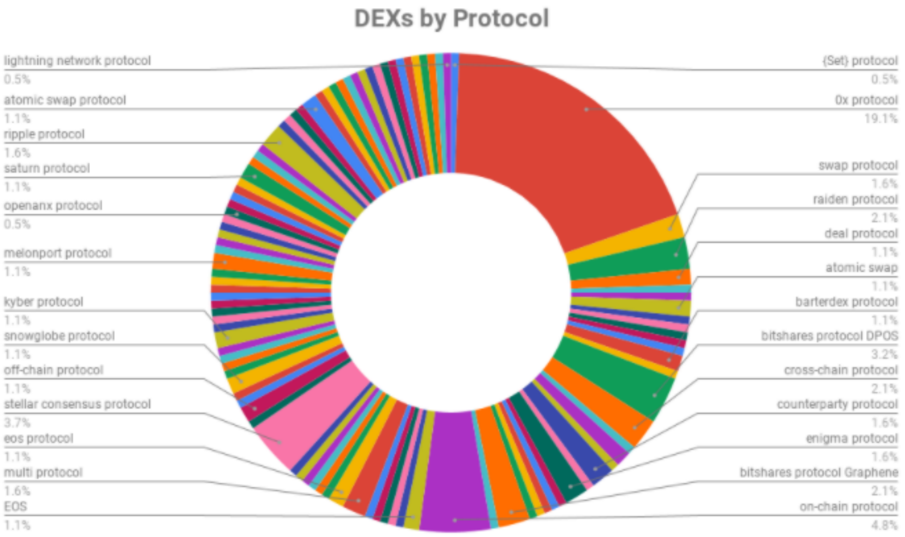
However, this is problematic due to the scalability limitations of existing blockchains because users would have to submit multiple entries to the blockchain, even for theoretically non-critical actions. As a result, very few DEXs are fully decentralized. Most DEXs have chosen to use a hybrid centralized/decentralized approach by keeping non-critical actions off-chain and critical actions on-chain.

DEX appears to be a magic bullet that can solve security problems and possibly pave the way for mass adoption. The promising potential of a DEX motivated us to look into Binance DEX, Resistance DEX, eFin, and Bisq and also highlight some features of INDEX and Ethfinex to seewhat the future may hold.

Major challenges

The two major challenges with most DEX’s today are liquidity and UX. The problem is that there is a lack of standardization across DEXs — no common trading and liquidity protocol. This is the reason that most DEXs have low liquidity and volume — they simply do not share a common liquidity pool. There are many projects trying to solve this protocol problem today. The leader by number of DEX integration partners in this space currently is the 0x protocol and has a great chance of becoming that standard in 2019.

After analysing a Github database of around 258 known DEXs in existence today in the market , we have tried visualising the lack of a standardized protocol for decentralized exchanges. The result is staggering. There are over two dozen different protocols. The leader is 0x which powers over 19% of these DEXs today:



10.2 - (Source: https://github.com/distribuyed/index)

We have analysed each Dex and made comparisons using these criteria:

- User Experience
- Security
- Liquidity
- Trading Pairs

User Experience

This feature is crucial because it determines the amount of effort required for a new trader to learn or get a feel of the interface of the DEX.

- **Binance DEX**

Binance DEX features the same user interface as Binance.com. It also integrates TradingView charts. On top of that, users have the option to create wallets and keep their own private keys. If that’s not secure enough for you, Binance DEX also incorporates software and hardware wallets such as the Ledger Nano S and Trust Wallet.

- **ResDEX(Resistance)**

ResDEX caters to the novice user. Users need to first download the ResDEX client before they can begin trading. Once the platform is open, they can easily choose which coins to trade. They can even buy Bitcoin (BTC) or Ethereum (ETH) with their Visa credit card.

- **Bisq**

Bisq also caters to the novice user. Just like ResDEX, you need to download the desktop client to begin trading. It also comes with a special wallet where you control the private keys.

- **eFin**

eFin’s user interface may be complex but it is easier to navigate than the interface of Binance DEX. Experienced traders will have no problem using eFin because it mirrors most trading platforms. There might be some learning curve for new users but it is not as overwhelming as Binance DEX. Among the four decentralized exchanges, Bisq offers the most basic interface. It is very easy to navigate because it comes with limited features.

Security

Traders maintaining control of their private keys is only one of the aspects of DEX security. Another big issue is front-running. This occurs when someone else can insert orders ahead of pending orders, all other security concerns are discussed in security analysis section separately.

- **Binance DEX**

Binance DEX implements several measures to combat front-running, also they are planning to use proof stake technique to reduce block time but all the measurements are in development.

- **ResDEX**

ResDEX relies on InstantSwaps to combat front-running. This feature allows trades to be executed in less than a minute without waiting for block confirmations. To do this, users must deposit a certain amount of RES (native gas/token) as collateral.

- **Bisq**

Bisq protects user transactions by employing three basic measures. When an offer to buy or sell is matched, an escrow transaction requires 2-of-3 multi-signature addresses. Both parties must sign the transaction before the escrow is unlocked. This usually happens when one user confirms the payment of fiat currency while the other user confirms receipt of payment.

With this system of locking funds in escrow, frontrunning is never an issue. As long as there’s no dispute, both parties will receive the funds as agreed. The downside is that trades can take time and users go through the hassle of depositing funds for escrow before trading.

- **eFin**

eFin solves the two issues (transaction speed and escrow deposit) faced by Bisq users with the implementation of the Multi-signature Transaction Engine (MTE). Just like Bisq, the MTE requires the signatures of two participants before the shared wallet is unlocked. Once funds are unlocked, the data is published on the eFin blockchain, which facilitates the trade by using features such as Atomic Swaps or RSK Smart Contracts. These features provide near-instant transaction times. This mechanism solves the slow transaction speed that plagues Bisq users.

Liquidity

Liquidity is another problem that prevents existing decentralized exchanges from attracting users. A DEX is as good as dead if trading activity is anaemic.

- **Binance DEX**

Binance (centralized) is the largest cryptocurrency exchange in terms of volume with an average daily trading volume of $664.4 million. They also have an average of 313,000 daily active users to put them in second place in this category just behind CoinBase. Therefore, Binance can leverage these numbers to provide liquidity on Binance DEX

- **ResDEX**

Unlike Binance, ResDEX doesn’t have a huge liquidity pool to rely on. So, they partnered with someone who does. They signed a contract with Huobi the company agreed to provide liquidity on ResDEX.

- **Bisq**

Unfortunately, Bisq has very little to offer in terms of liquidity. There’s little to no trading activity on the platform. In fact, Bisq has an average of 52 trades per day according to March2019 numbers. So far, Bisq offers no real solution to boost its liquidity pool.

- **eFin**

eFin provides a two-pronged approach to combat liquidity issues. The first method is to motivate users to stake, store, and trade their cryptocurrencies on eFin by providing incentives through a distributed rewards system. The second approach is to integrate order books with other decentralized exchanges. The idea is to create one massive decentralized exchange that helps users find a suitable trade as fast as possible..

An Ideal DEX should have the following features

- Aggregate liquidity.
- User friendly UX.
- Be trustless.
- Have low to no fees.
- Have automatic order matching, instant trades, update traders with all available resources.

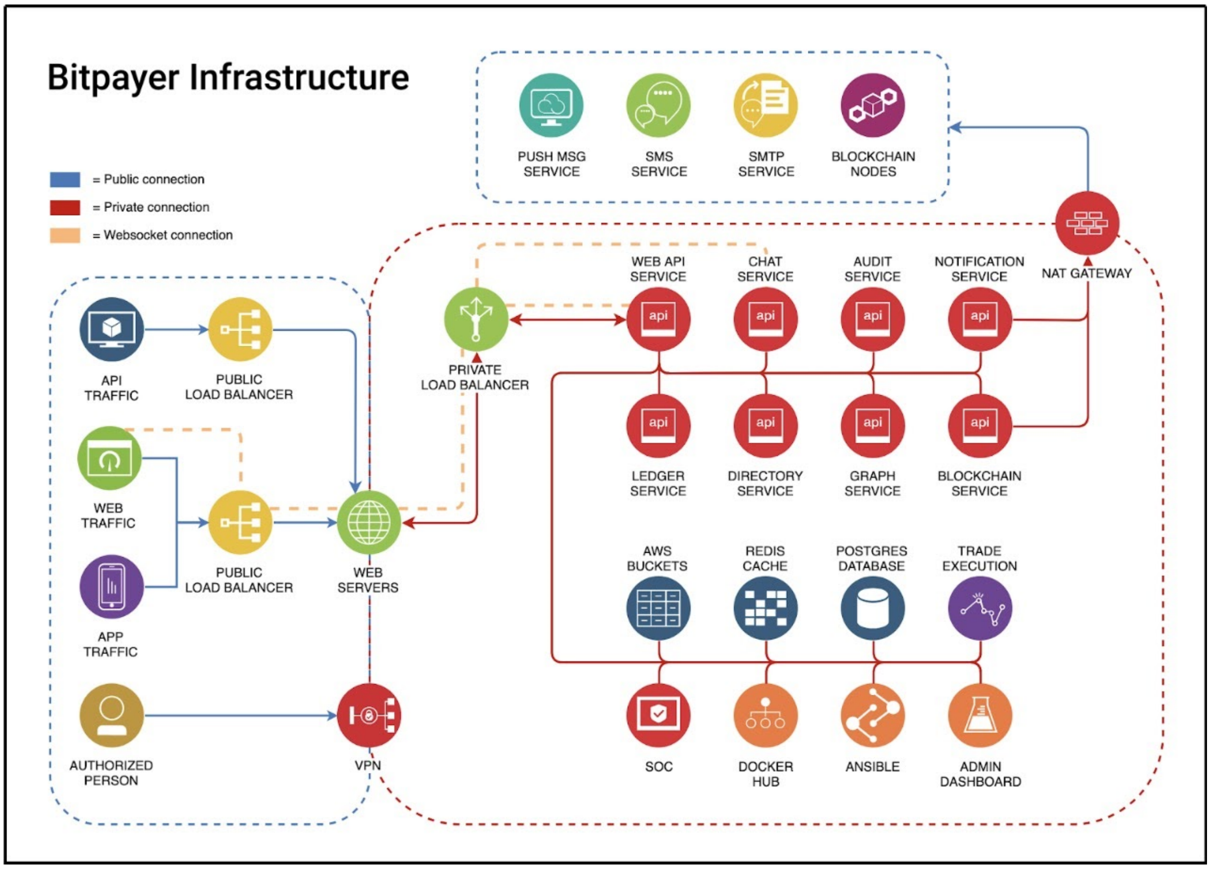
Initially DEX should not impose any fees to increase crowd and liquidity, instead DEX should give rewards of DEX native tokens, this will also help in increase liquidity of native tokens as well, once DEX reach the high amount of users and high liquidity, new regulations can be imposed to users with minimum amount of fees.

DevOps engineering

Important to note is that the system has been designed to be platform agnostic, which means it can be taken into operation on self-hosted hardware and all major cloud-based hosting solutions like Google Cloud, Microsoft Azure and AWS (Amazon Web Services). In our examples we reference several AWS services like “ec2” and “buckets”.

The infrastructure consists of several containerized micro services to ensure maximum security, flexibility and scalability. We recommend using Docker to containerize services since Docker is supported on all major operating systems and very easy to set up. Running services inside a docker container does not mean each container should live on its own instance/server. One could choose to run most containers on one dedicated server, as long as this has no complications in terms of security.

The following pages highlight each part of the infrastructure and take us deep into the world of micro services and their data flows.



11.1 - BITPAYER INFRASTRUCTURE

Public and private subnets

As you can see in the image - 11.1, there is a clear distinction between publicly connected components (blueline) and privately connected components (red line).

Making it impossible to directly reach components handling sensitive information from the internet requires the creation of public and private subnets. Components connected with a blue line live in public subnets, able to connect to and from the internet. Components connected with a red line live in private subnets and can only connect to each other. Separate internal/private subnets might be introduced at a later stage to segregate services. Public and private subnets can both have security groups assigned to allow/deny traffic. So, even if public components have an external IP address, access to the components can be denied by setting specific security rules. Private components are unable to connect to the internet by default. They have to use a NAT Gateway or Internet Gateway to relay the requests.

Interfacing components

There are five (3) components living in both a public subnet as well as in a private subnet:

1. Web Server
2. Blockchain Service
3. Notification Service

Web Server - Important to note is that web servers don’t require an external IP address even though they are part of apublic subnet, as long as they are registered in a “Target Group”. To streamline incoming web traffic, we utilize public load balancers that will forward traffic to a desired target group, balancing the traffic between multiple web server instances. Web servers are also used to establish a websocket connection between a web client and for example the chat service.

Blockchain Service - The Blockchain Service is always able to connect to the internet through a NAT gateway to reach its blockchain nodes to fetch blockchain-related data or process.

Notification Service - Several parts of the system need to notify customers and/or BitPayer DEX staff members. The notification service is able to reach external services like mobile push message providers.

External services that are part of the program

1. Google Firebase
2. Mandrillapp
3. Blockchain Nodes

Google Firebase - There are several companies providing a solution for push notifications to reach mobile app users, but Google Firebase is one of the most reliable services out there.

Mandrillapp - A project of company Mailchimp, widely known as one of the best HTML newsletter providers. Mandrillapp offers an API that enables programmatically sending emails from several sending domains, including your own custom domain.

Blockchain Nodes - Every supported cryptocurrency in the trading platform has a set of dedicated blockchain nodes so we don’t rely on external services which (at this time) don’t provide all the needed functionalities to safely create and manage customer and system blockchain wallets. The blockchain nodes live on the internet and are connected to thousands of other nodes across the world. These nodes are marked insecure which is why we recommend running the blockchain nodes on dedicated servers across the world and making them only available for programmatic access for our Blockchain Service.

Public and private load balancers

Public and private load balancers have one main job: load balancing traffic. The most important difference between the two is to what kind of subnet/network they are connected.

The public load balancers are used to balance the web traffic coming from the internet between multiple web servers that will process the requests. An additional job for the public load balancer is offering an SSL certificate to encrypt traffic between a client (web/app) and the platform. In the MVP setup, we plan to use three public load balancers:

1. Web traffic (web + app)
2. API traffic (programmatic)

Private load balancers are used to balance traffic between internal/private services only. For example, if the load of the WEB API SERVICE consistently is too high, one can introduce a private load balancer which would balance traffic between multiple WEB API services in one target group.

Web servers

Multiple web servers are set up as part of a target group behind a public load balancer to serve static files to web clients and proxying requests for dynamic data to the underlying services. We recommend using NGINX, a solid industry standard software package. Static files will change less often, and NGINX is built to efficiently deliver static files to clients, for example using the HTTP/2 protocol. HTTP/2 is fully multiplexed, so it is able to send multiple requests for data in parallel over a single TCP connection.

Service types

There are two main service types: stand-alone, and API-enabled.
An important example of a stand-alone service is the Trade Execution. It won’t serve an API to other services and will perform its job without the interaction with other services. The other services offer an API so other services can “talk” to them, requesting or providing information about internal transactions.

Databases

PostgreSQL - PostgreSQL is an object-relational database management system (ORDBMS) with an emphasis on extensibility and standards compliance. It has become the preferred open source relational database for many enterprise developers and start-ups, powering leading business and mobile applications. PostgreSQL is offered as an industry standard with all major cloud-based hosting providers.

Redis - Redis is an open source, in-memory data structure store, used as a database, cache and message broker. It is mainly used as a caching mechanism in the trading platform and made available for all services depending on caching to improve on speed.

Docker hub and ansible

Two important automation tools are Docker Hub and Ansible. As we mentioned at the beginning of this document, all services are containerized using Docker. The Docker Hub is basically a repository for all Docker images which offers version tags to quickly rollback a system if needed.

Ansible is an open-source software provisioning, configuration management, and application-deployment tool. It uses playbooks to record system operations which you can then play on a particular instance/server.

Types of web traffic

There are four different kinds of web traffic flowing through this platform:

1. Regular Web Traffic
2. Mobile App Traffic
3. Public API Traffic
4. Websocket Traffic

Regular Web Traffic - By this we mean traffic from the web browser to our platform.

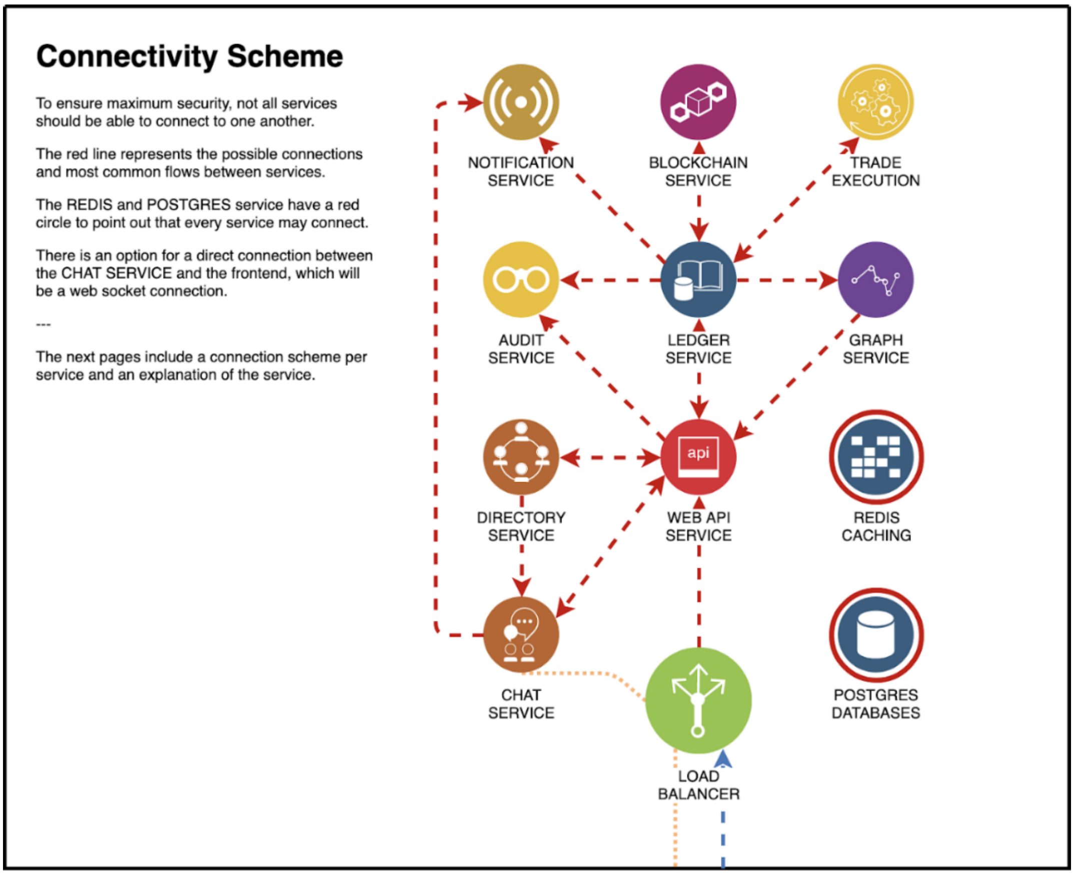
Mobile App Traffic - Basically, the same kind of traffic as Regular Web Traffic, but coming from mobile apps.

Public API Traffic - High velocity traffic coming from clients running trading software or trading bots. We might offer a websocket connection at a later stage for more efficiency.

Websocket Traffic - High velocity traffic between clients and the chat service. Instead of having to open a separate TCP connection for every update on the chat conversation, we use a single websocket connection that will remain open to improve on velocity.

Public API

As mentioned in the previous topic, the public API is an interface that can be used for high velocitytraffic originating from trading software and trading bots to automate trades and write monitoring tools. Users will need to generate an API key which is used to authenticate and encrypt data between the client and the platform.



Connectivity Scheme

To ensure maximum security, not all services should be able to connect to one another.

The red line represents the possible connections and most common flows between services.

The REDIS and POSTGRES service have a red circle to point out that every service may connect.

There is an option for a direct connection between the CHAT SERVICE and the frontend, which will be a web socket connection.

The next pages include a connection scheme per service and an explanation of the service.

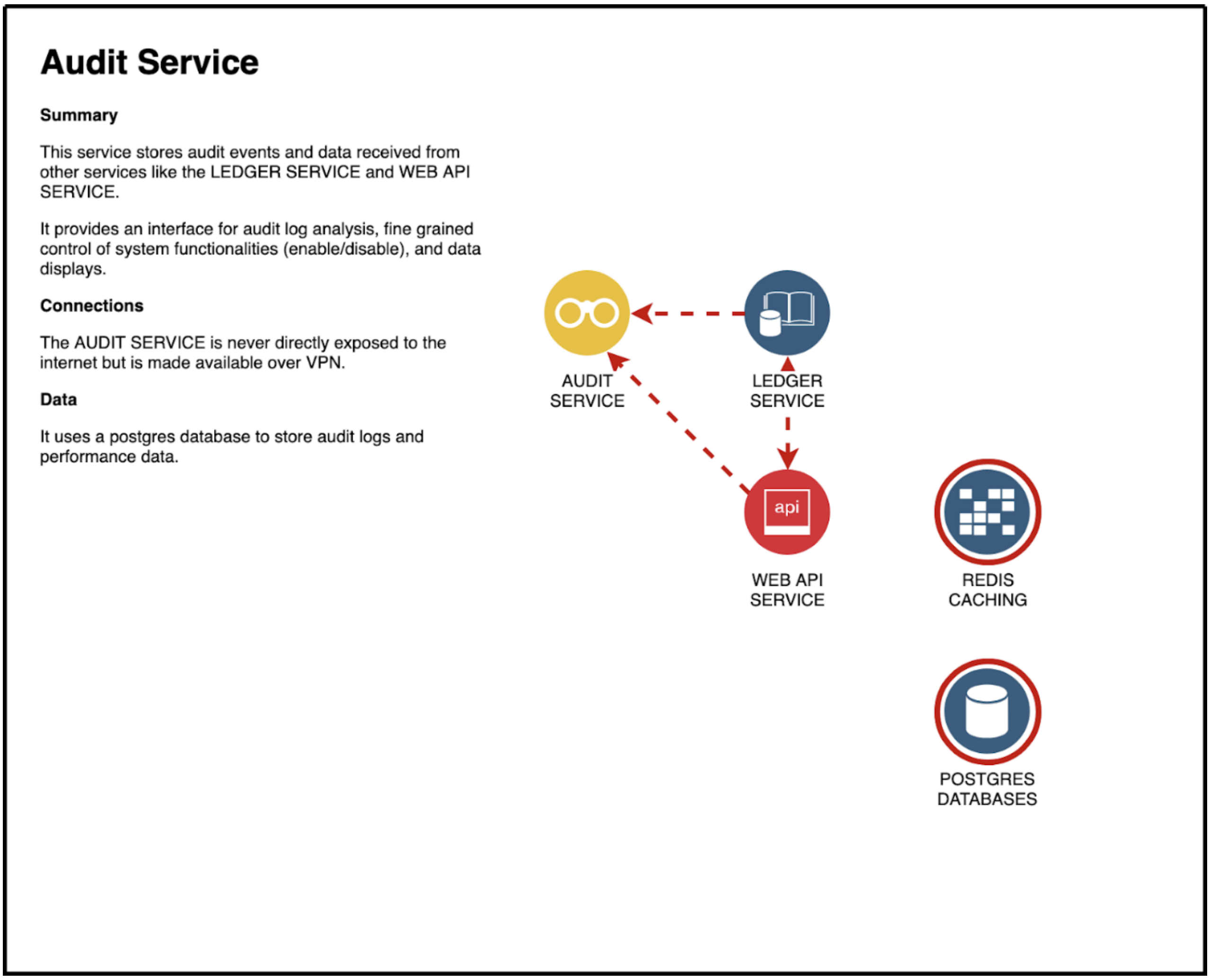
11.2 - CONNECTIVITY SCHEME

Audit service

Image - 11.3 shows a stripped-down version of image - 11.2, highlighting the connections between the audit service and other services.

The audit service stores events and data received from other services like the ledger service and web API service. It provides an interface for audit log analysis, fine grained control of system functionalities (enable/disable), and data displays.

It stores audit logs and performance data in a postgreSQL database. The audit service is never directly exposed to the internet.



Audit Service

Summary

This service stores audit events and data received from other services like the LEDGER SERVICE and WEB API SERVICE.

It provides an interface for audit log analysis, fine grained control of system functionalities (enable/disable), and data displays.

Connections

The AUDIT SERVICE is never directly exposed to the internet but is made available over VPN.

Data

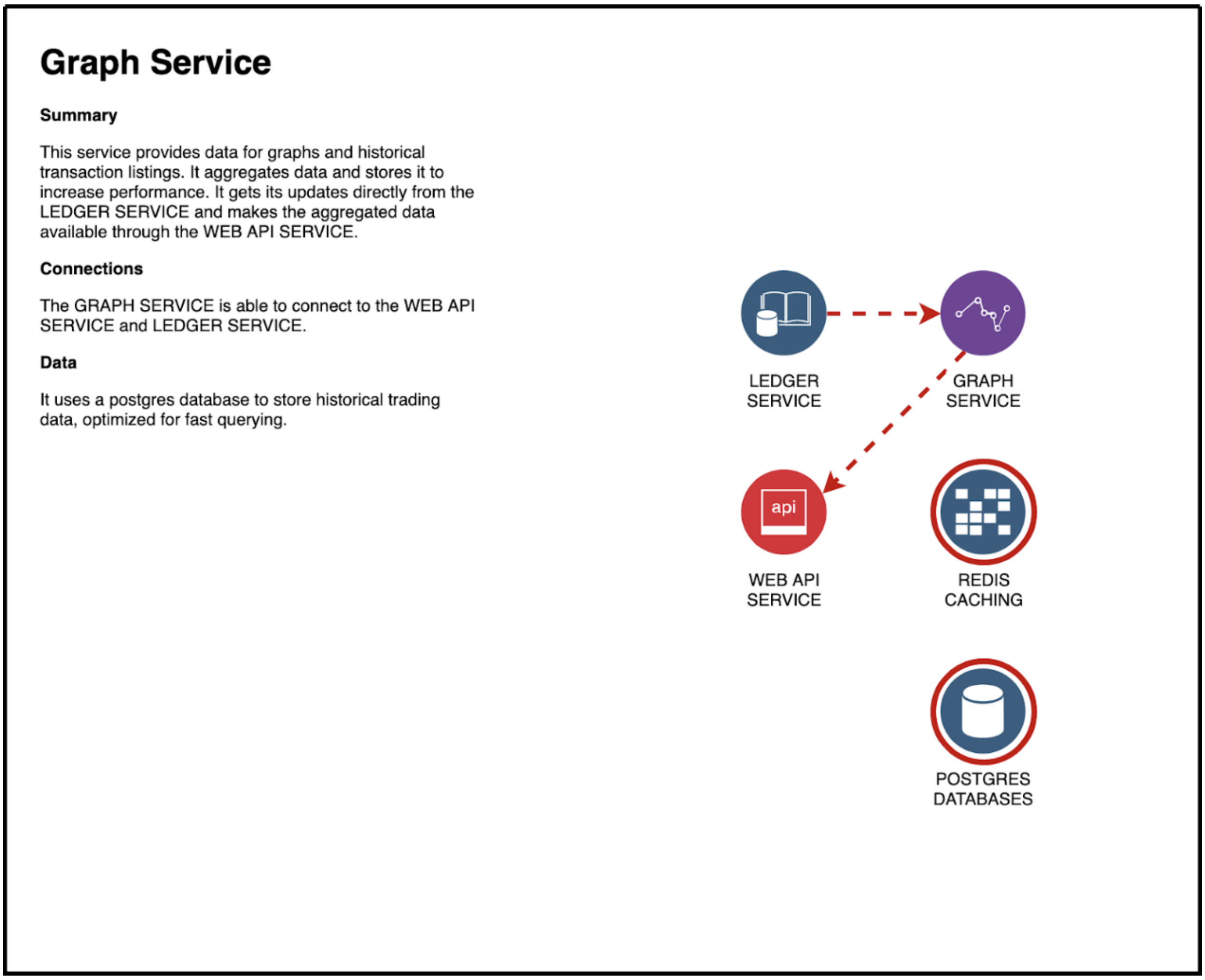
It uses a postgres database to store audit logs and performance data.

11.3 - AUDIT SERVICE

Graph service

Image - 11.4 shows a stripped-down version of image 11.2 , highlighting the connections between the graph service and other services.

The graph service provides data for graphs/charts and historical transaction listings. It aggregates data and stores it in a postgreSQL database to increase performance. It gets its updates directly from the ledger service and makes the aggregated data available through the web API service.



Graph Service

Summary

This service provides data for graphs and historical transaction listings. It aggregates data and stores it to increase performance. It gets its updates directly from the LEDGER SERVICE and makes the aggregated data available through the WEB API SERVICE.

Connections

The GRAPH SERVICE is able to connect to the WEB API SERVICE and LEDGER SERVICE.

Data

It uses a postgres database to store historical trading data, optimized for fast querying.

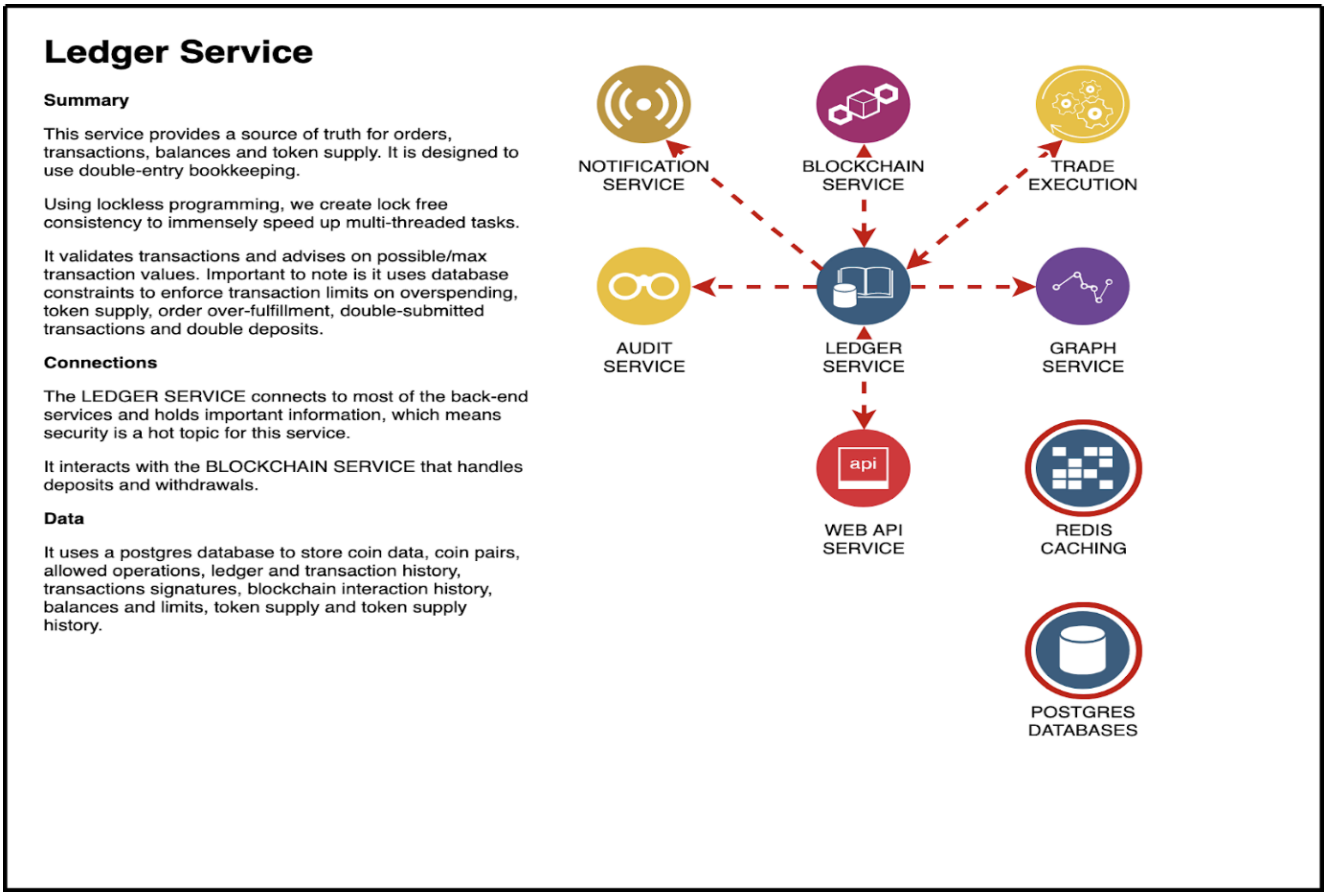
11.4 - GRAPH SERVICE

Ledger service

Image - 11.5 shows a stripped-down version of image - 11.2, highlighting the connections between the ledger service and other services.

The ledger service is probably the most well-connected service in the system. Because of its importance and availability to so many system components, security is of the essence. It provides a source of truth for orders, transactions, balances and token supply.

It validates transactions and advises on possible/maximum transaction values. Using database constraints, it enforces transaction limits on overspending, token supply, order over-fulfilment, double-submitted transactions and double deposits. Using lockless programming, we create a lock free consistency to immensely speed up multi-threaded tasks.



Ledger Service

Summary

This service provides a source of truth for orders, transactions, balances and token supply. It is designed to use double-entry bookkeeping.

Using lockless programming, we create lock free consistency to immensely speed up multi-threaded tasks.

It validates transactions and advises on possible/max transaction values. Important to note is it uses database constraints to enforce transaction limits on overspending, token supply, order over-fulfillment, double-submitted transactions and double deposits.

Connections

The LEDGER SERVICE connects to most of the back-end services and holds important information, which means security is a hot topic for this service.

It interacts with the BLOCKCHAIN SERVICE that handles deposits and withdrawals.

Data

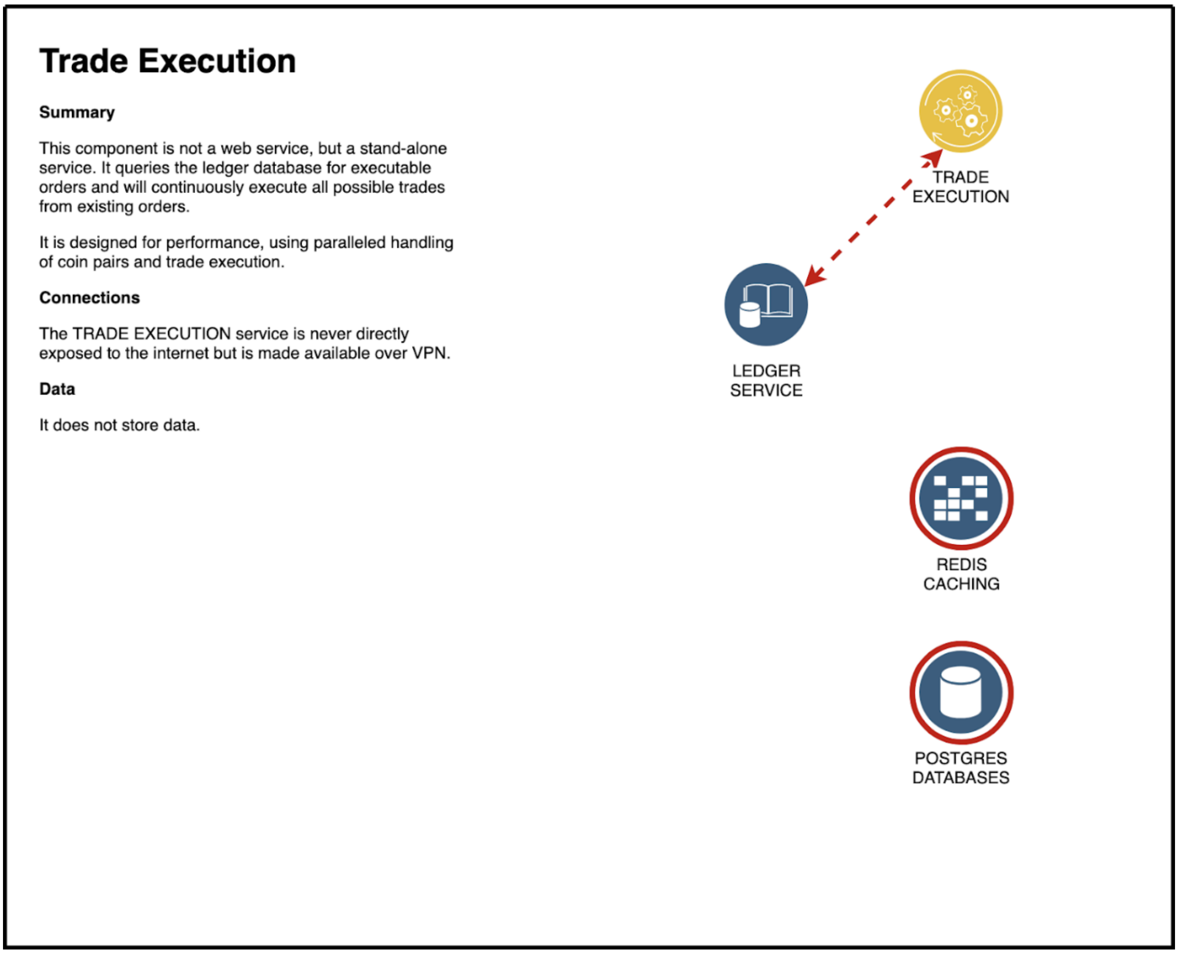
It uses a postgres database to store coin data, coin pairs, allowed operations, ledger and transaction history, transactions signatures, blockchain interaction history, balances and limits, token supply and token supply history.

11.5 - LEDGER SERVICE

Trade execution

Image - 11.6 shows a stripped-down version of image - 11.2, highlighting the connections between the notification service and other services.

It is designed for performance, using parallel coin pair processing and transaction execution. No data is saved.



Trade Execution

Summary

This component is not a web service, but a stand-alone service. It queries the ledger database for executable orders and will continuously execute all possible trades from existing orders.

It is designed for performance, using paralleled handling of coin pairs and trade execution.

Connections

The TRADE EXECUTION service is never directly exposed to the internet but is made available over VPN.

Data

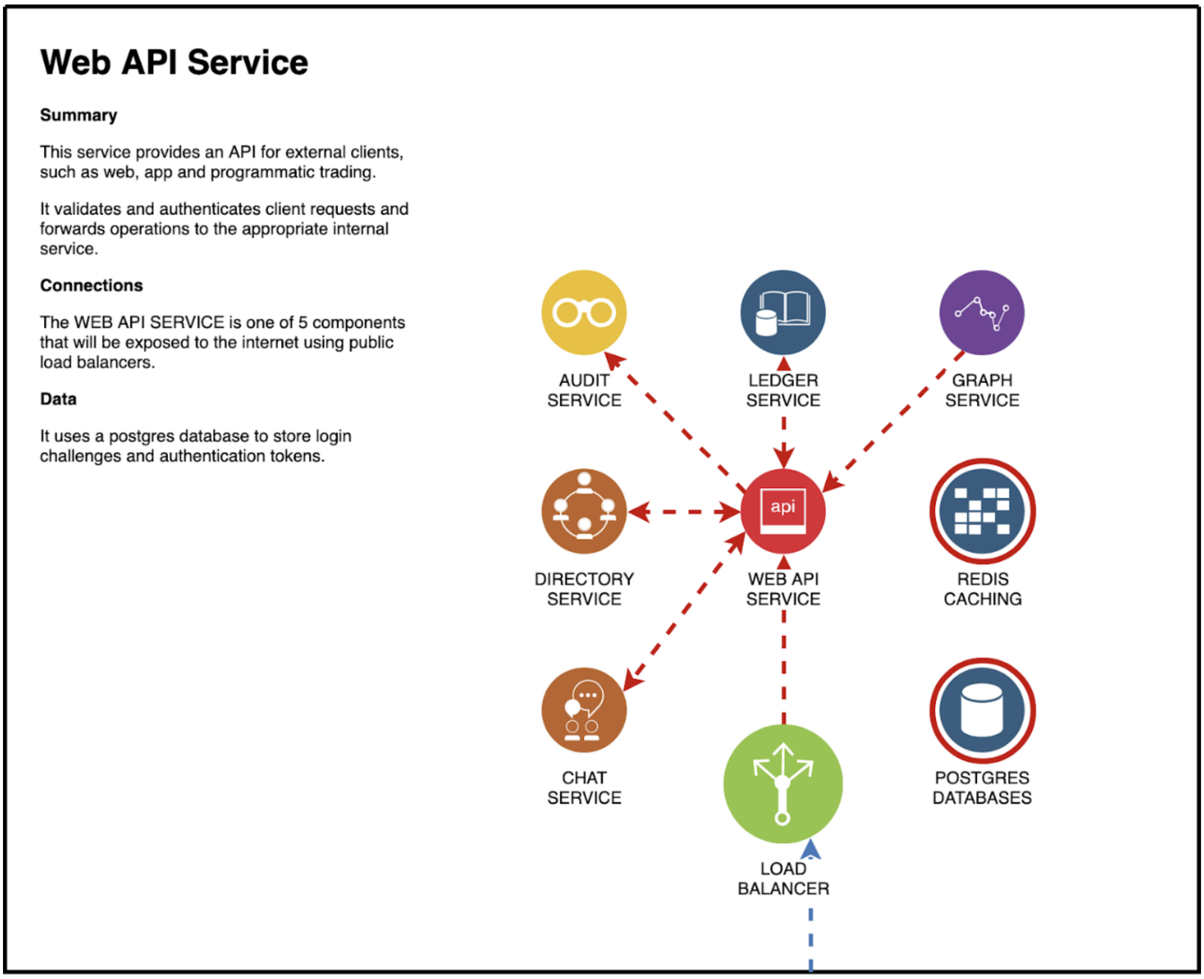
It does not store data.

11.6 - TRADE EXECUTION

Web API service

Image - 11.7 shows a stripped-down version of image 11.2, highlighting the connections between the web API service and other services.

The web API service provides an API for external clients such as web, app and programmatic trading. It validates and authenticates client requests and forwards operations to the appropriate internal service. Several web API services are part of a load balanced target group to increase performance. The web API service uses a postgreSQL database to store login challenges and authentication tokens.



Web API Service

Summary

This service provides an API for external clients, such as web, app and programmatic trading.

It validates and authenticates client requests and forwards operations to the appropriate internal service.

Connections

The WEB API SERVICE is one of 5 components that will be exposed to the internet using public load balancers.

Data

It uses a postgres database to store login challenges and authentication tokens.

11.7 - WEB API SERVICE

Front-end applications

This topic shines a (technical) light on the available front-end applications and the differences between them related to security and communications. In-depth information about topics like design, flow diagrams, and UX are covered in the Design Documentation.

Web application

For the web application, we use a combination of HTML, CSS, Javascript, and Python. All communications between front-end and back-end servers will be handled by python, a high-level programming language used throughout the system.

Traffic: All client traffic is encrypted using HTTPS whether in web or app form. Our end of these communications is handled by NGINX as explained before in topic Infrastructure > Web Servers.

We split types of traffic by using different subdomains to balance traffic through different load balancers.
For example, chat-related traffic would use the API listening on chat.Bitpayer DEX.io where programmatic traffic related to trading software would use api.Bitpayer DEX.io.

At the time of writing, we plan to use four public load balancers:
1.Web traffic originating from web browsers;
2.App traffic originating from the mobile application;
3.API traffic originating from trading software and bots;
4.Chat traffic originating from web browsers.

In the event the web API service is overused because of highly volatile market updates, an extra load balancer might be introduced to streamline market traffic to an API listening on market.Bitpayer DEX.io.

Mobile application

The mobile application will be developed for iOS and Android devices. The application will be written in either native languages such as Kotlin for Android and Swift for iOS or using Flutter, an open-source mobile application development framework created by Google which compiles for both iOS and Android.

Traffic: Traffic from mobile apps will be sent through a separate load balancer (app.Bitpayer DEX.io) to distribute the load more evenly and give an extra point for control. This can be useful when traffic from mobile apps needs to be interrupted in the event of a major update and at the same time not affecting regular web traffic.

Public API

A public API will be made available to enable trading and market monitoring using trading software and trading bots. Users will be able to generate a public API key via the web application. Every request between the client (trading software) and the platform has to be signed using that key to properly authenticate.

To improve on performance, the public API will support websocket traffic via its own load balancer (api.Bitpayer DEX.io). More about websockets in topic Infrastructure > Types of Web Traffic.

Technology stack

Blockchain	Ethereum
Admin Portal front end	React.js
Database	MongoDB , PostgreSQL
Message brokers	Rabbit Mq, Kafka
APIs	Node.js, Golang
Cloud Services	AWS (EC2, S3, EBS, ELB, IAM)
Operating System	Linux (Ubuntu 18, 20)
Caching	Redis
Containerisation Tools	Docker, Kubernetes
CI/CD & Infrastructure tools	Jenkins, terraform
Configuration management	Ansible
Monitoring tools	Grafana, Nagios
Build tools	Maven
Version control	git
Code Quality tools	sonarQube
Web Servers	Nginx
Load balancers	Nginx, Netscaler
Smart Contract	Solidity
Smart Contract test	Chai-Mocha
Smart contract Platform	Truffle, Remix ide
Testnet	Rinkeby, Ropsten, Ganache

Testing and quality assurance

Testing

❑ **Smoke Testing:**

When the application gets ready for testing, the first thing is to confirm that a given application is ready for further testing or not. To confirm this, smoke testing is required, which can be done at the start of any major testing when we release anything major and needs to confirm release done properly.

❑ **Functional Testing:**

Once the application is confirmed ready for the test, functionality testing of all features needs to be performed. Functional testing is very important to confirm if all developed functionalities are working or not. Functional testing will be based on analysis of the specifications of the functionalities. Test cases creation and execution will be based on functional testing.

❑ **Regression Testing**:

When we post issues as part of execution, we may find multiple issues, so confirmation testing requires that other modules are not impacted after making changes in the application for issues found during execution.

❑ **System Testing:**

It includes complete testing of fully integrated system components based on defined scope.

❑ **System Integration Testing:**

Testing to be done for the areas where software is integrated with any third party application or software.

❑ **Performance Testing**:

Site can be accessible by multiple users at the same time and multiple data can be transferred for any single request. In this case, performance testing needs to be carried out with subtype i.e. Load Testing and Stress testing.

Once the testing is performed, the QA Team would share a Sign-Off email to the respective team members stating the details about the testing performed, open issues (if any) and applicability of the build in-order to release on the Staging / Production Environment.

Architecture of smart contract

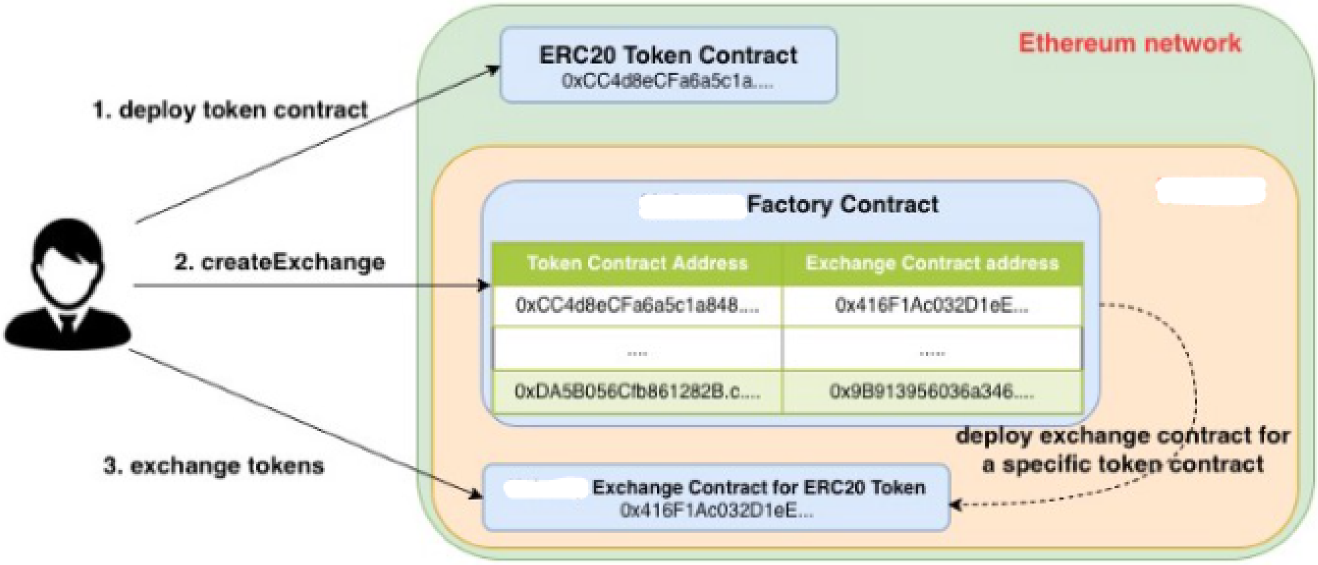
Smart contracts are self-executing contracts containing the terms and conditions of an agreement among peers. The terms and conditions of the agreement are written into code. The smart contract executes on the Ethereum blockchain's decentralized platform.

The agreements facilitate the exchange of money, shares, property, or any asset. BitPayer implements a decentralized token exchange for cryptocurrencies in the Ethereum network, which provides a simple interface for token holders to swap one kind of token to the other with low gas cost. The BPY tokens are added to the BitPayer in order to boost the token liquidity.

Architecture overview for p-2-p trading

The smart contract of decentralised exchange is made up of:

- **Factory Contract:** it is the manager of exchange contracts, which creates new exchange contracts and maintains a mapping from token contract address to its exchange contract address.
- **Exchange Contract:** each token contract has its own exchange contract, which executes the transaction to swap tokens.



Token Contract Address	Exchange Contract address
0xCC4d8eCFa6a5c1a848....	0x416F1Ac032D1eE...
....	.....
0xDA5B056Cfb861282B.c....	0x9B913956036a346....

Architecture Overview of Uniswap

14.1 - ARCHITECTURE OVERVIEW

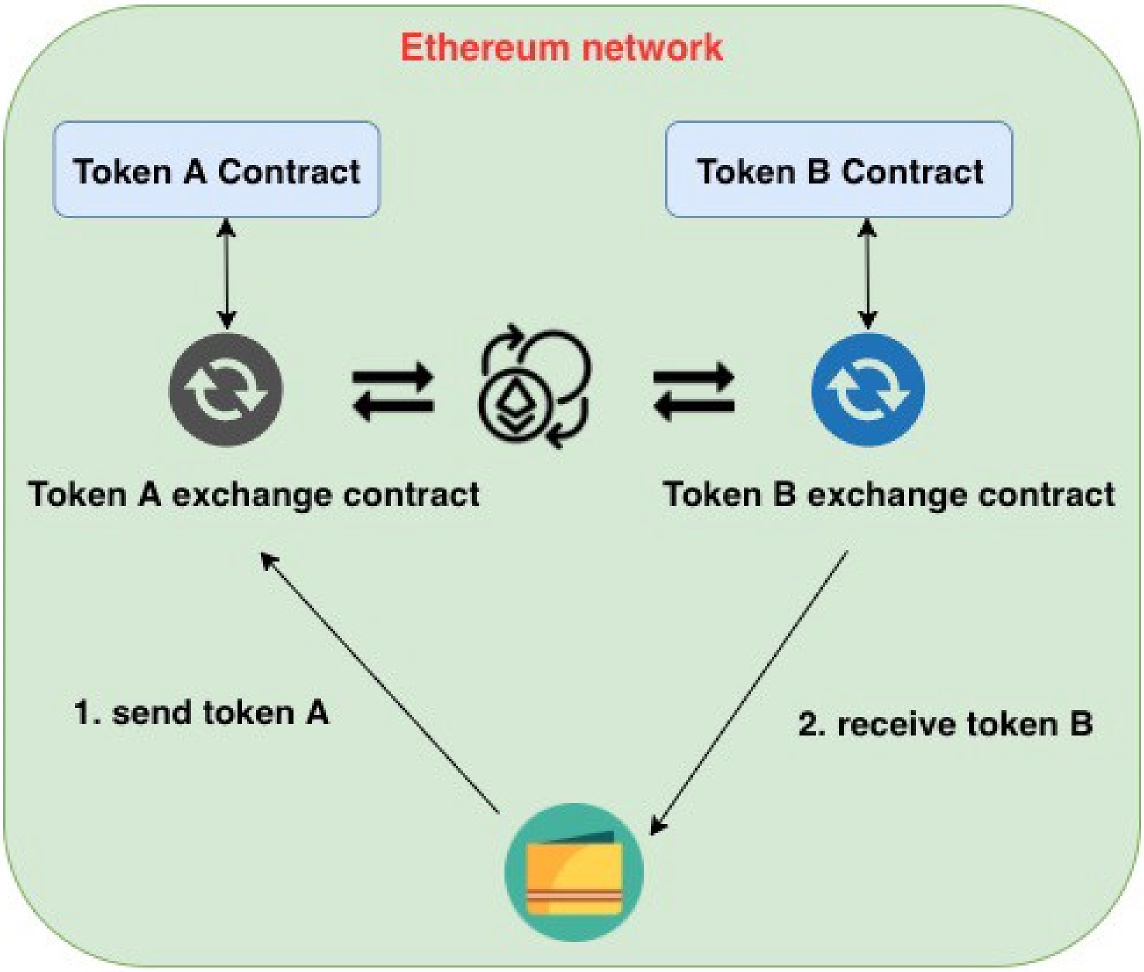
A typical developer’s workflow using BitPayer is:

1.Deploy BPY token contract to Ethereum network;

2.Create an exchange contract through factory contract for BPY token.

3.Swap BPY token with other tokens using the exchange contract.

Each token exchange contract can only swap BPY token with BPY by itself. To swap between different BPY tokens, BitPayer uses BPY as the “ bridge” or intermediary token (i.e., sell token A into BPY and buy token B using BPY).



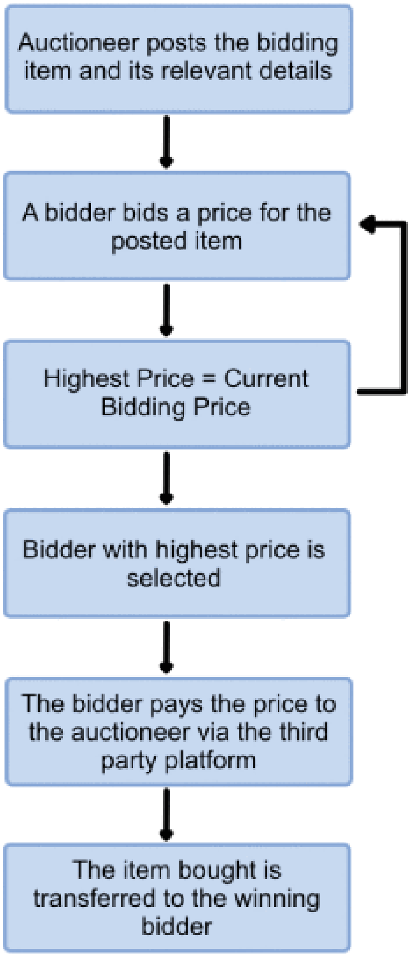
14.2 – TOKEN SWAP WITH BITPAYER

The architecture is limited to Ethereum network and token swap in the “same” Ethereum network.

Architecture overview for AuBlock gamified auction platform

An e-auction system has several elements that are in common with the stock exchange platform. It consists of bidders, auctioneers, and third-party intermediaries who provide the platform that connects bidders to auctioneers and allows posting products, checking the highest bidding price, and declaring the winner with the highest bidding price.

First, the seller posts the bidding information and the starting price. The bidders bid the price in the sealed envelope, and when it is received by the auctioneer, the sealed envelope price that is the highest is announced as the current highest price. If no price received higher than the current bidder's highest price or the ending time is due, it is announced as the winning price, and the auctioneer can send the product and receive the money from the winning bidder. By applying the proposed blockchain based e-auction platform , we conclude that the smart contract can enforce confidentiality, non-repudiation, and prevention of unauthorized alteration of entered bidding orders.


Auctioneer posts the bidding item and its relevant details
A bidder bids a price for the posted item
Highest Price = Current Bidding Price
Bidder with highest price is selected
The bidder pays the price to the auctioneer via the third party platform
The item bought is transferred to the winning bidder

14.3 - AuBLOCK AUCTION PLATFORM

Liquidity pools

Liquidity pools offer a new standard for efficiently trading assets while allowing investors to earn a yield on their holdings. They are pools of tokens locked in a smart contract to facilitate liquidity. Typically, the function of liquidity pools is to allow traders to trade their digital assets while earning rewards on their asset holdings.

There are currently two types of decentralized exchanges on Ethereum:

1. Order book peer-to-peer exchanges
2. Liquidity pool exchanges

ORDER BOOK EXCHANGES - rely on a bid/ask system to fulfil trades. When traders place a buy or sell order at their chosen price for a token, the exchange's matching engine only executes the trade once an opposite order at that price is available. Traders who place (limit) orders on the order book are called market makers and traders who execute their orders against orders already on the order book are called takers. A token's price is therefore determined by the traders who choose at what price level to place orders. This system works fairly well when there are enough buyers and sellers in the market, but there are a few unavoidable issues: tokens that lack liquidity due to low volume or interest not only become difficult to buy and sell, but are also susceptible to unpredictable price swings caused by large individual transactions. Consequently, tokens that are characterised by high price volatility and inefficient conversions are unlikely to be adopted.

LIQUIDITY POOLS - directly address this problem by removing the dependence of tokens on trade volume and ensuring constant liquidity.

Compared to the traditional order book model, liquidity pools have four main advantages:

1. **Guaranteed liquidity at every price level :**

Behind the scenes, the "liquidity pool" is just an automated market maker in the form of a smart contract that automatically matches traders' buy and sell orders based on predefined parameters. Traders do not need to be matched directly with other traders, so as long as investors have deposited assets into the pool, liquidity is constant.

2. **Automated pricing enables passive market making :**
On order book exchanges, market makers need to constantly adjust their bids and asks as asset prices move. Liquidity pools don't need to aggregate information across exchanges to determine the price of assets. Liquidity providers simply deposit their assets into the pool and the smart contract takes care of the pricing.
3. **Anyone can become a liquidity provider and earn:**
Liquidity pools require no listing fees, KYC, or other barriers characteristic of centralized exchanges. Anyone can invest in an existing liquidity pool or create a new exchange pair for any token, at any time.
4. **Lower gas fees:**
Decentralized exchanges created have a minimalist smart contract design that reduces gas costs. Efficient price calculations and fee distributions within the pool means less friction between transactions. Most smart contracts can only send traded funds back to the same wallet.

Liquidity pool returns depend on three factors:

1. Asset prices when supplied and withdrawn
2. Liquidity pool size
3. Trading volumes

Working of liquidity pools

Liquidity pools are pools of tokens that are locked in a smart contract. They are used to facilitate trading by providing liquidity and are extensively used by some of the decentralized exchanges.

One of the first projects that introduced liquidity pools was UniSwap. In its basic form, a single liquidity pool holds 2 tokens and each pool creates a new market for that particular pair of tokens.

When a new pool is created, the first liquidity provider is the one that sets the initial price of the assets in the pool. The liquidity provider is incentivised to supply an equal value of both tokens to the pool. If the initial price of the tokens in the pool diverges from the current global market price, it creates an instant arbitrage opportunity that can result in lost capital for the liquidity provider. This concept of supplying tokens in a correct ratio remains the same for all the other liquidity providers that are willing to add more funds to the pool later.

When liquidity is supplied to a pool, the liquidity provider (LP) receives special tokens called LP tokens in proportion to how much liquidity they supplied to the pool. When a trade is facilitated by the pool a 0.3% fee is proportionally distributed amongst all the LP token holders. If the liquidity provider wants to get their underlying liquidity back, plus any accrued fees, they must burn their LP tokens.

Each token swap that a liquidity pool facilitates results in a price adjustment according to a deterministic pricing algorithm. This mechanism is also called an **automated market maker (AMM)** and liquidity pools across different protocols may use a slightly different algorithm.

Basic liquidity pools use **a constant product market maker algorithm** that makes sure that the product of the quantities of the 2 supplied tokens always remains the same. On top of that, because of the algorithm, a pool can always provide liquidity, no matter how large a trade is. The main reason for this is that the algorithm asymptotically increases the price of the token as the desired quantity increases

The bigger the pool is in comparison to a trade, the lesser the price impact or slippage occurs, so large pools can accommodate bigger trades without moving the price too much.

Because larger liquidity pools create less slippage and result in a better trading experience, some protocols like Balancer started incentivising liquidity providers with extra tokens for supplying liquidity to certain pools.

Modules with Timeline

Modules

1.UI and UX:

Description: This module includes designing the UI with designing tools and writing the corresponding html and css. This section also includes user experience requirement gathering from client and implementation.Admin dashboard, Trading screen, Order pool

2.Admin

Description: This module contains both backend and frontend integration for admin dashboard.

2.1 Admin Authentication

- Login with email
- Google Captcha
- 2FA Integration
- Password Recovery

2.2 Manage smart contracts

- Deploy smart contracts
- Interact with smart contracts

2.3 Manage token listing

- Add new token pair to DEX
- Remove existing token pair
- Analytics of token pairs

2.4 Manage fee wallets, fee structure and incentive program

- Add fee wallet to collect fees of trading
- Change fee wallet
- Change fee structure
- Add incentive program of native tokens

2.5 Manage users

- Verify KYC(if manual)Invite new users
- Block users
- View full transaction history of users (open/closed/executed orders)
- All statistics of particular user

2.6 Customised graph and statistic

- Interactive graphs
- Bar charts and statistics

3.Trading screen (traders/users)

Description: This module contains both backend and frontend integration for admin dashboard.

3.1 Trader/Users Authentication

- Signup
- Login
- Google Captcha
- 2FA Integration
- Profile updation

3.2 KYC Integration (manual or third party if Needed)

3.3 Token pair list

- Token pair table (market price, volume, 24 hrs change %)

3.4 Relayer (Order book)

- Each and every token pair have separate order book
- Buy /Sell Order book

3.5 Integrate wallets (metamask,trezor etc) Integrate all wallets

- Select wallet
- Select account
- Change wallet
- Change account

3.6 Matching engine

- Algorithm implementation for matching order

3.7 Trading engine

- Optimised trading engine with security

3.8 Create order offline

- Create buy/sell limit/market order

3.9 Modify/Cancel offer

- Cancel existing order
- Modify existing order

3.10 Graphs and analytics

- Customised graphs

3.11 Peer to peer trading

- Maker create offer offline and sign it
- Taker submit order to DEX (only valid taker)

3.12 Initial exchange offering

- Startup Register panel
- Smart contract creation by filling details only
- Deployment of token contract for IEO
- Adding token pair to DEX if token contract already created
- Listing token after IEO

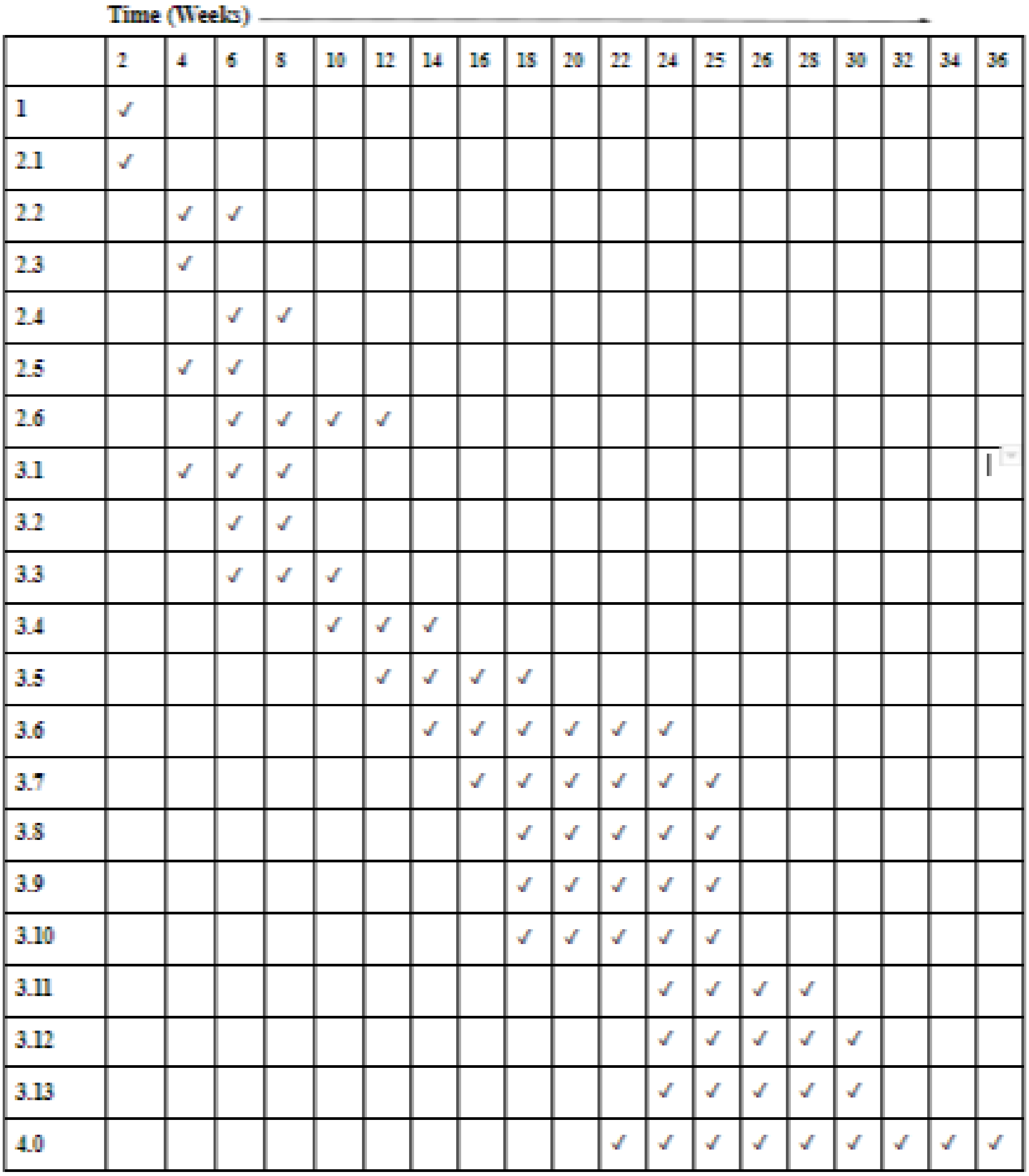
3.13 Margin trading

- dy dx protocol integration
- Separate screen for margin trading (wireframes, design)
- Order book for margin trading
- Trading engine for dy dx protocol

4.Testing

- Testing Module

Module - week breakdown



Time (Weeks) →

	2	4	6	8	10	12	14	16	18	20	22	24	25	26	28	30	32	34	36
1	✓																		
2.1	✓																		
2.2		✓	✓																
2.3		✓																	
2.4			✓	✓															
2.5		✓	✓																
2.6			✓	✓	✓	✓													
3.1		✓	✓	✓															
3.2			✓	✓															
3.3			✓	✓	✓														
3.4					✓	✓	✓												
3.5						✓	✓	✓	✓										
3.6							✓	✓	✓	✓	✓	✓							
3.7								✓	✓	✓	✓	✓	✓						
3.8									✓	✓	✓	✓	✓						
3.9									✓	✓	✓	✓	✓						
3.10									✓	✓	✓	✓	✓						
3.11												✓	✓	✓	✓				
3.12												✓	✓	✓	✓	✓			
3.13												✓	✓	✓	✓	✓			
4.0											✓	✓	✓	✓	✓	✓	✓	✓	✓

Total Timeline : 30 Weeks

Testing : + 2 weeks Alpha testing, + 2 weeks Beta testing, Production live

Cost : USD 80,000