FUDA

Fuel for Digital Assets (FUDA)

Version 1.0

16 May 2018

Contents

Introduction

Fuel for Digital Assets (FUDA) is an enterprise-grade blockchain platform developed to promote the digitisation and exchange of assets on the blockchain. By leveraging on the FUDA platform, businesses are able to quickly develop and scale a blockchain based solution or Blockchain Application (bApp) for their own customers. This will enable asset owners to realise the value of their assets, improve efficiency, reduce the cost of processing and managing assets and facilitate the trading of assets.

The future development roadmap will also support:

1. **free form data** to be stored and shared on the blockchain
2. **Smart Automata**, self-contained bots that provide services to users

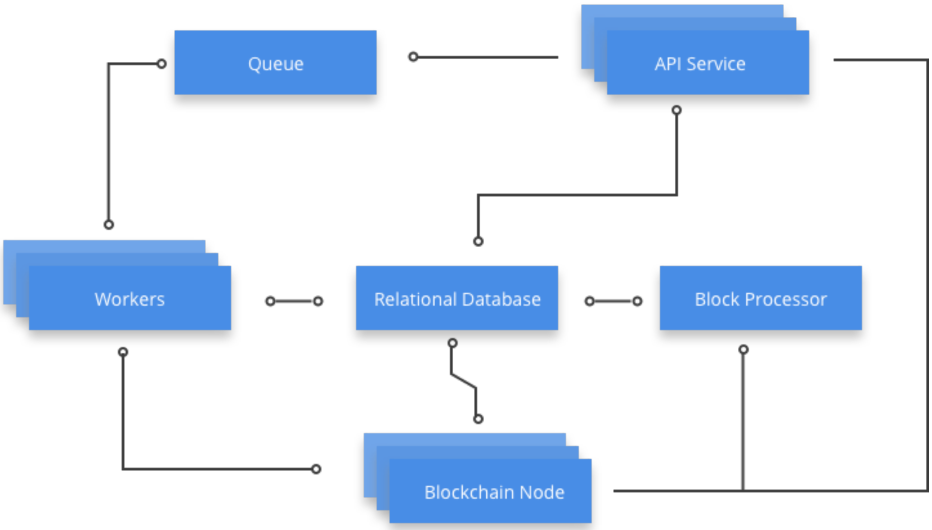
One of the first use cases for FUDA is FundPlaces, a real estate investment platform that allows property developers to easily issue TILES (assets), distribute the economic benefits and redeem those TILES at the conclusion of the investment. TILES owners can also make decisions regarding their investment through the blockchain voting mechanism.

FUDA Architecture

Overview

The FUDA stack is built on permissioned, Proof-Of-Authority (POA), blockchain technology and exposes the FUDA RESTful API so enterprises can easily integrate it into their existing web and mobile applications.

Each FUDA stack also integrates its own relational database, to process and store blockchain information. The database organizes and caches blockchain data so that it can be queried efficiently. Each enterprise (also known as a partner) runs their own FUDA stack. The API, datastore and blockchain nodes are horizontally scalable.



FUDA TECHNOLOGY STACK

Every "action" conducted on FUDA is a transaction that requires transaction fees. Validators who validate transactions and secure the blockchain, collect these fees which must be paid in ? tokens. Therefore, ? tokens derive intrinsic value from being critical to validating transactions.

Safety and Permissions

One of the great promises of public blockchains was to let everyone transact anonymously in a peer-to-peer and completely decentralized fashion. Unfortunately, every Garden of Eden has its snakes.

From the hacks of exchanges, to scam phishing scams, crypto currency has come to be known in the mainstream as a dangerous place. As such, financial regulators are clamping hard down on crypto currencies. Even the giant internet advertisers such as Facebook, Google and Twitter are banning crypto related ads on their platforms.

This issue is so important that the Winklevoss Twins, that run Gemini, have submitted a proposal to create the Virtual Commodity Association, a self-regulatory organization meant to police digital-currency markets and custodians.

Our solution to restoring mainstream trust in blockchain technology and crypto currencies are permissioned chains where every entity on the chain has some basic level of vetting. Addresses and/or accounts cannot be created willy-nilly and used to hide the origin of stolen funds. Scammers and thieves cannot just exit their funds through fly-by-night exchanges in questionable jurisdictions.

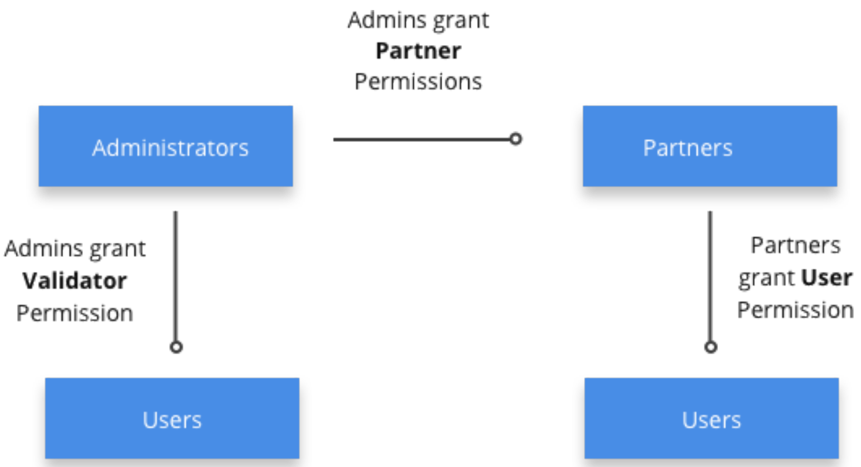
There are 4 groups of participants on the chain.

1. **Administrators** -- They have the ability to grant or revoke all permissions, including admin permissions. The granting or

revoking of certain important permissions is consensus driven and must be agreed upon by a majority of the administrators.

2. **Validators** -- They maintain the blockchain by validating transactions and collecting any transaction fees. Only Administrators can grant or revoke Validator permissions and Partner permissions and this is subject to Administrator consensus.
3. **Partners (Enterprises)** -- They provide the applications that the Users will use and act as gatekeepers for their users. For example, KYC services, real estate investment opportunities, hotel booking etc. They have permission to create addresses and have the responsibility to do some some basic level of verification depending on what they allow their Users to do. Partners are granted their permissions from Administrators and are allowed to create new assets, streams (a type of datastore) and addresses on the chain.
4. **4. Users** -- They are the end users of the applications. Each User has an address on the chain which allows them to send and receive assets. Their limited permissions are granted by the Partners. Users cannot grant any permissions.

https://www.bloomberg.com/news/articles/2018-03-13/winklevoss-twins-have-a-plan-to-police-cryptocurrency-trading


Admins grant **Partner** Permissions
Administrators
Partners
Admins grant **Validator** Permission
Partners grant **User** Permission
Users
Users

Fuda Permissions Waterfall

To summarize, Administrators grant access to the chain to Partners and give Validators permission to validate and collect transaction fees. Partners grant access to the chain to their Users and are responsible to appropriately verify their Users depending on what their applications allow Users to do. User verification is local to that Partner and Application, sensitive personal data will NEVER BE STORED on the blockchain.

An application that sells financial products to a User will have to do a much more in depth User verification than an application that allows you to just book hotel rooms. This ensures that potential bad actors cannot leverage the decentralized nature of the blockchain to hide and exit their ill-gotten gains.

Blockchain

The Multichain (https://www.multichain.com/) permissioned blockchain is the basic technology underlying the platform and the Crypto Assets API. It is a fork of bitcoin core but with added support for:

1. Permission management
2. Asset Issuance
3. Streams which are like key-value storage

Rich metadata with each "operation" (also known as transactions in blockchain parlance)

As the blockchain is permissioned, only entities which have been given explicit access can connect to the blockchain and engage in activities such as: sending and receiving assets, issuing assets, writing to streams and

mining.
Since all participants are "trusted", a much less resource intensive mechanism, Proof-Of-Authority and round-robin validation. This means that the throughput of the chain can be a lot higher than that of the public blockchains.

Only critical information is stored on the blockchain. Since anyone who is allowed to connect to a blockchain can see all the information on it, no personal identifying information is stored. Other than asset ownership data, the only other things stored in streams on the blockchain are: address relationships, asset information, asset booking data and trading order data

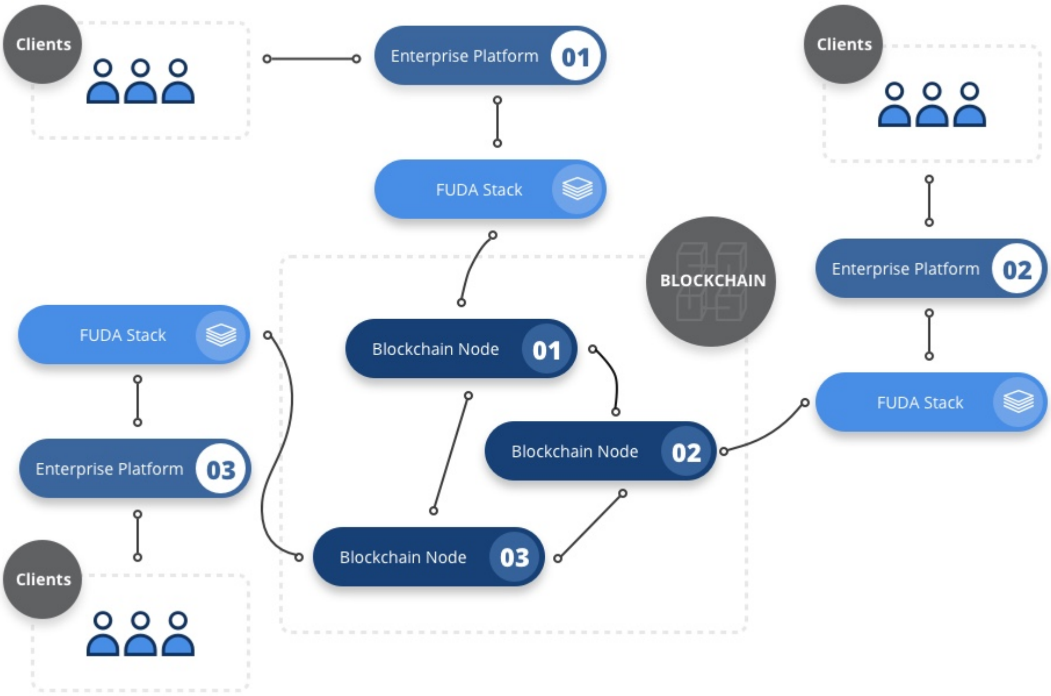
FUDA API

The interface to the blockchain is powerful, but low-level and allows all sorts of applications to be built on top. Hence, we have created the FUDA API. This is a micro-service which presents the programmer with a friendly Restful API interface for managing assets, taking trades etc. Programmers only need to be familiar with RESTful APIs to do the integration, no blockchain development experience is necessary.

There are 2 ways in which an enterprise can use the FUDA API, they can connect to it through an existing partner or they can run a separate FUDA technology stack which includes their own blockchain node.

Connecting through an existing partner is the easiest way and there is no need to run any infrastructure. However, some customer information like names, email addresses and mobile numbers will need to be shared.

If a partner runs their own blockchain node and version of the FUDA stack, they can achieve complete customer data segregation from other entities utilizing the same chain. This allows them to more easily conform to local data privacy and banking secrecy regulations, while implementing jurisdiction appropriate KYC measures for their customers.



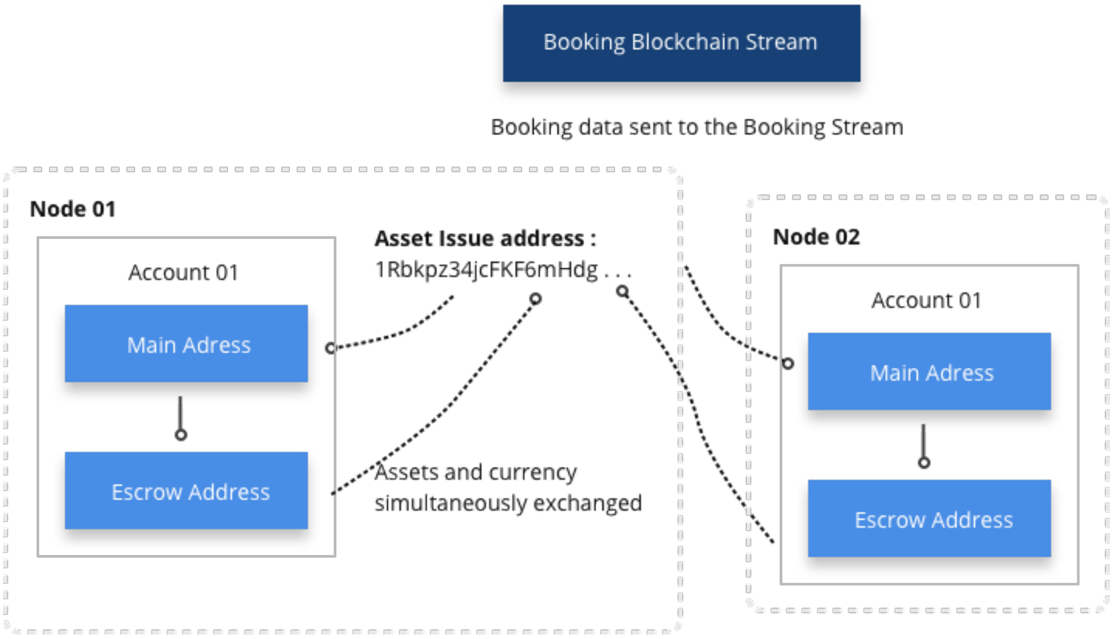
In this example, Enterprise 01, Enterprise 02 and Enterprise 03 run their own separate stacks.

Blockchain Operations

Issuing Assets

Each node has the ability to issue assets (in FundPlaces case, we call them TILES) and grant that ability to any addresses it generates. Every unique asset has a unique asset address that is allowed to issue it. No other address will be able to issue that asset.

To issue an Asset, a node will first "reserve" the Asset by creating a single genesis token and then "burning" it (sending it to an address whose private key is not known). Then, it will broadcast the presence of this new token over a blockchain stream for all other nodes to pick up. Other nodes can then take bookings for the token from their members.



There is a 4 step process in the booking and issuing process.

1. The account transfers currency tokens (currency), to their Escrow address. Anything in this Escrow address still "belongs" to them but is earmarked for a future transaction. Metadata containing some of the Booking data is also recorded with the transfer to tie them together.

2. The account then prepares an atomic exchange instruction. This is an instruction to transfer x currency for y asset. This instruction is incomplete as it only has one leg, which has been authorized by the

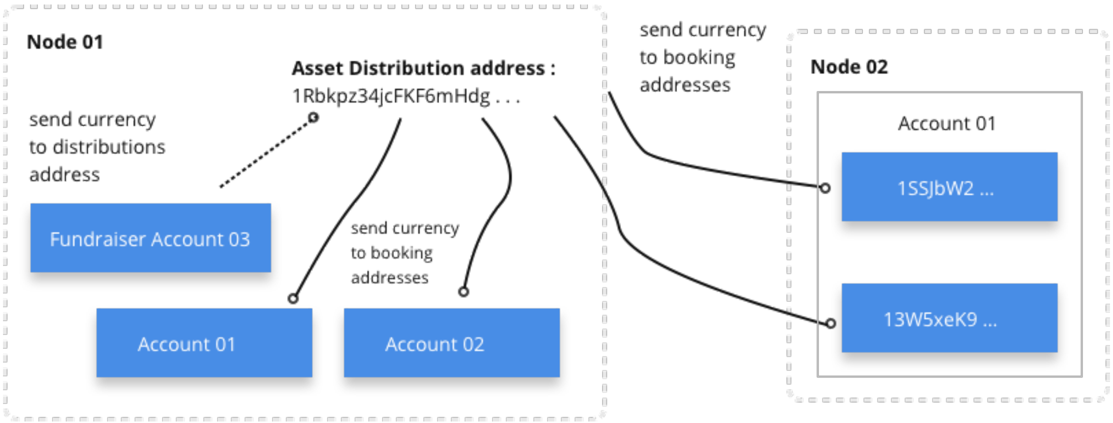
account. Only when the issuer provides the asset issuance leg and authorizes it will the instruction be complete and can be sent to the blockchain.

3. The Booking is broadcast on the Booking Stream and can be picked up by the issuing node. The issuing node will decrypt the booking and make sure that the exchange instructions are correct. If not, it will reject the booking and execute the refund instruction.

4. Once the booking period is over, then the issuing node can allocate the Assets to be issued according to various algorithms. For each booking, an asset is allocated and the exchange instructions are completed and broadcast to the blockchain.

Distributing Returns

Distributing returns is a relatively straightforward process. The fundraiser will send currency tokens to the Token Distribution Address and then those funds will be distributed to the addresses which contain the Tokens.

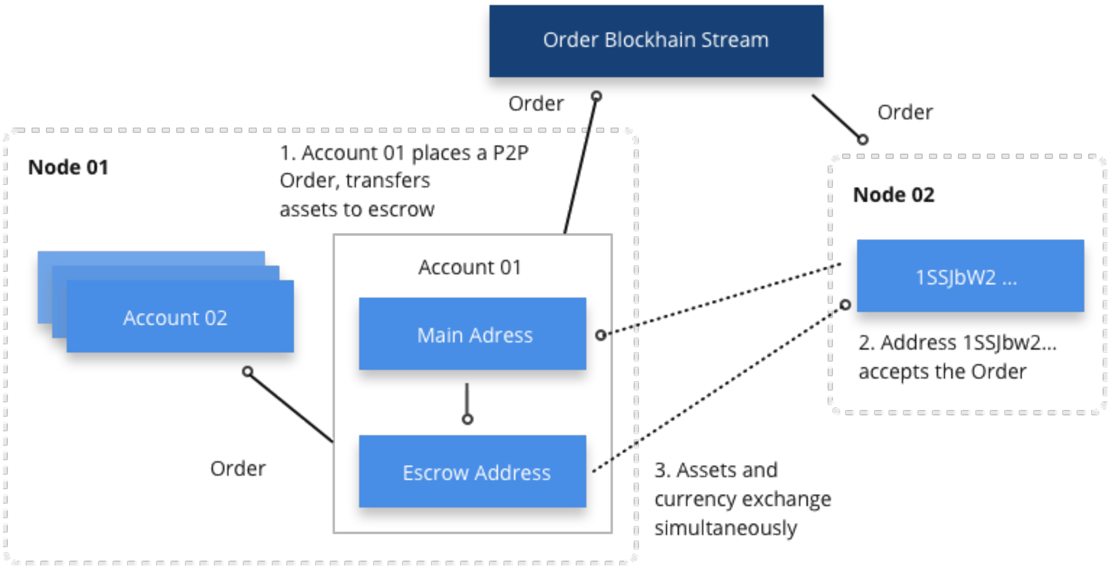


Trading

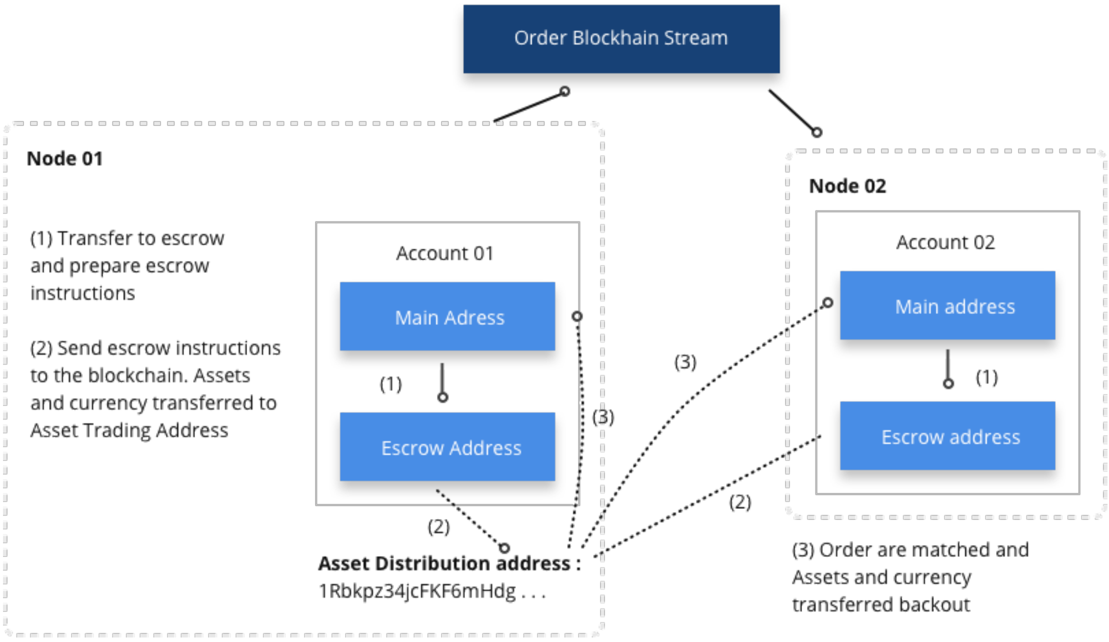
There are 2 types of trading, Peer-to-Peer (P2P) trading and automated trading.

In **P2P trading,** an investor makes an offer to sell x of an Asset for y currency (or vice-versa). The seller will transfer their asset to their escrow address and prepare an atomic exchange transaction instruction.

The incomplete instruction is sent with the order to the Order Stream. All the nodes will receive it and can offer it to their users. Any potential buyer can view the instructions and if they agree with the trade, add their leg of the instructions. They will then send this to the blockchain to be confirmed as a transaction.


Order Blockhain Stream
Order
Order
Node 01
1. Account 01 places a P2P Order, transfers assets to escrow
Account 01
Account 02
Main Adress
Escrow Address
Order
Node 02
1SSJbW2 ...
2. Address 1SSJbw2... accepts the Order
3. Assets and currency exchange simultaneously

For Automated trading, an investor makes an offer to sell x of an Asset for y currency (or vice-versa) with the possibility of partial fulfillment but with the promise of more liquidity and quicker execution of the trade. The issuer of the asset will facilitate trading by acting as a market maker or matching orders directly.


Order Blockhain Stream
Node 01
(1) Transfer to escrow and prepare escrow instructions
(2) Send escrow instructions to the blockchain. Assets and currency transferred to Asset Trading Address
Account 01
Main Adress
(1)
Escrow Address
(2)
(3)
Asset Distribution address :
1Rbkpz34jcFKF6mHdg . . .
(3)
(2)
Node 02
Account 02
Main address
(1)
Escrow address
(3) Order are matched and Assets and currency transferred backout

The seller will transfer their asset from their main address to their escrow address and then prepare an escrow instruction that will transfer the assets from their escrow address to the Asset Trading address. This instruction is signed but not broadcast to the blockchain yet. The order is generated and the instruction is added to it. Then the entire order is encrypted using Asset Trading address' public key, after which the order is sent to the Order Stream.

The order can be picked up from the Order Stream by any node, but it can only be decrypted by the node that manages the Asset Trading address. Other nodes will just ignore this particular order. Once sufficient orders are received, an order matching algorithm is run for that particular Asset.

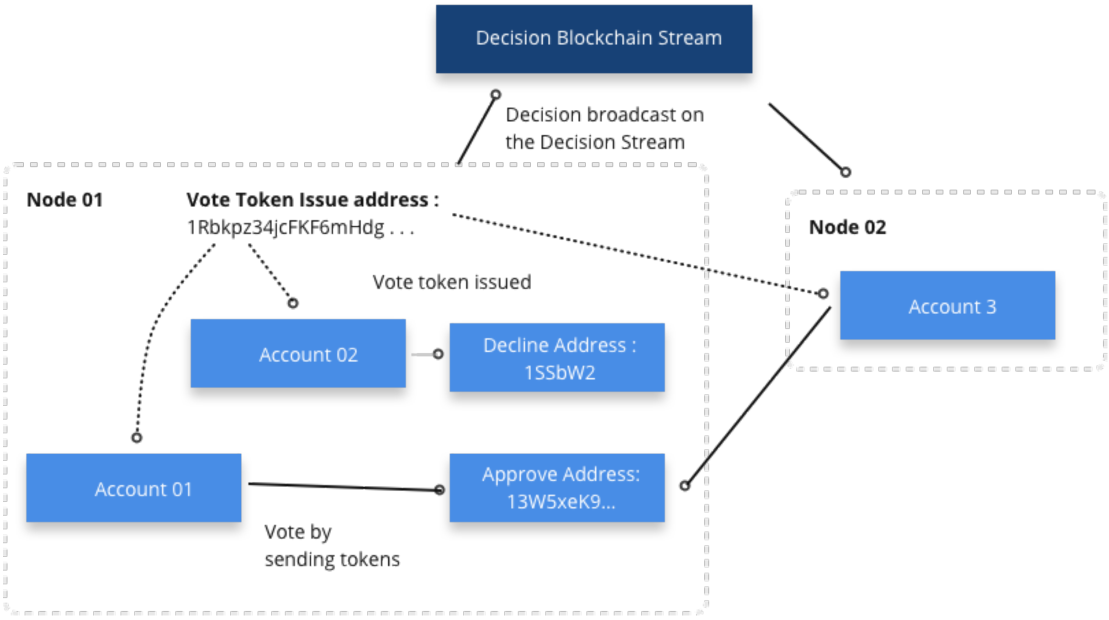
Once the algorithm has matched the orders, the matching orders' escrow instructions are broadcast to the blockchain. This results in currency and tokens being transferred from the escrow addresses of accounts that have placed orders to the Asset Trading address. Then those assets and currency are transferred out to fulfill the orders.

Voting on Decisions

Periodically, token holders can vote on whether various decisions will affect the investment underlying their token. The votes are basically tokens and each decision will have its own unique vote token. For the vote to be conducted, vote tokens will be issued in a 1:1 ratio to those addresses which hold the assets.

Voters can either approve or decline the decision by sending their tokens to either an **Approve** address or a **Decline** address. Voting will be open for a certain time period and when either an Approve or Decline address contains 50% + 1 of the voting tokens by the end of the voting period, then that choice "wins". If no choice has a majority, then the decision is considered to be declined.

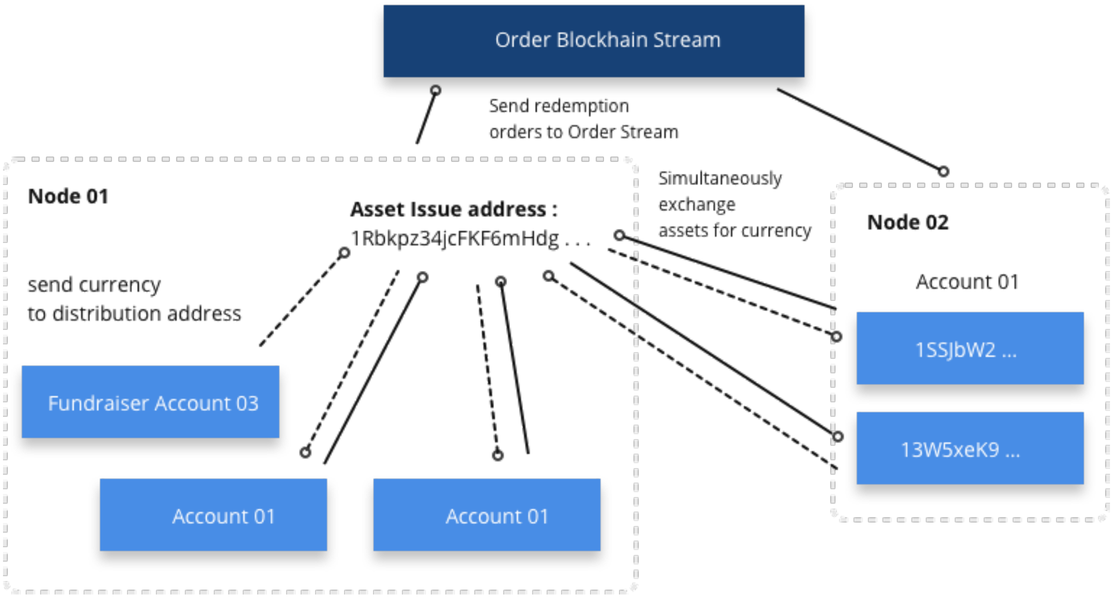
Additionally, once vote tokens are issued, they can be traded like any other token.


Decision Blockchain Stream
Decision broadcast on the Decision Stream
Node 01
Vote Token Issue address : 1Rbkpz34jcFKF6mHdg . . .
Vote token issued
Account 02
Decline Address : 1SSbW2
Account 01
Approve Address: 13W5xeK9...
Vote by sending tokens
Node 02
Account 3

Redeeming Tokens

At the end of the lifespan of the investment, the Assets can be redeemed by the issuing address. Automated trading is shut down and all tokens and currency are returned to the originating addresses.

Currency for the redemptions are placed in the Asset Distribution Address and atomic exchange transaction instructions are prepared for each address that contains the Asset. Each instruction is encrypted with destination address' public key and broadcast on the Order Stream. The order can then be picked up, decrypted and accepted.



Future Roadmap

Data Storage

Nodes on the chain can store and share data through the use of streams. Partners can create streams and allow other partners to write to these streams to collaborate. Given the additive nature for blockchain data, each stream will an have associated parser defined so that stream data can be read in a coherent manner.

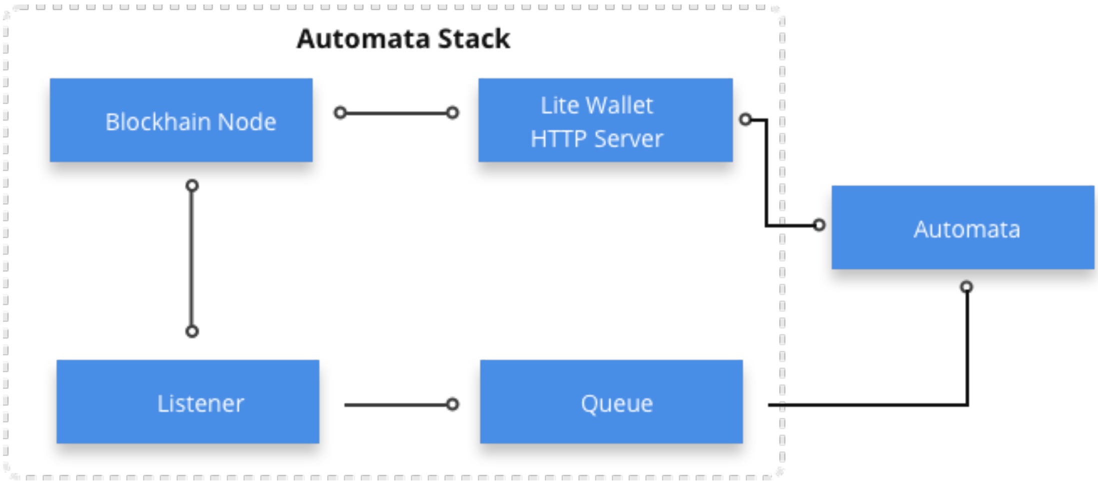
Lite-Wallet Server

Users can directly access the blockchain without going through a partner. However, they will still need a partner to grant them the appropriate permissions to conduct transactions and they will need their own pool of native currency to pay for transaction fees. This will allow Lite-Wallets to be used especially for mobile devices.

Smart Automata

Smart Automata are services that are built out of the accounts and data storage primitives that we have defined on the chain. Smart automata are autonomous services that store their critical state on the blockchain.

These services "belong" to an account and are located at an "address". Users on the blockchain interact with these services by sending transactions to that address with attached metadata. Like users, automata need to be given permission by an existing partner to access the chain.



Automata can use any internal architecture, however, they must be able to speak HTTP, consume and produce JSON as well as read incoming transactions (jobs) from the queue.