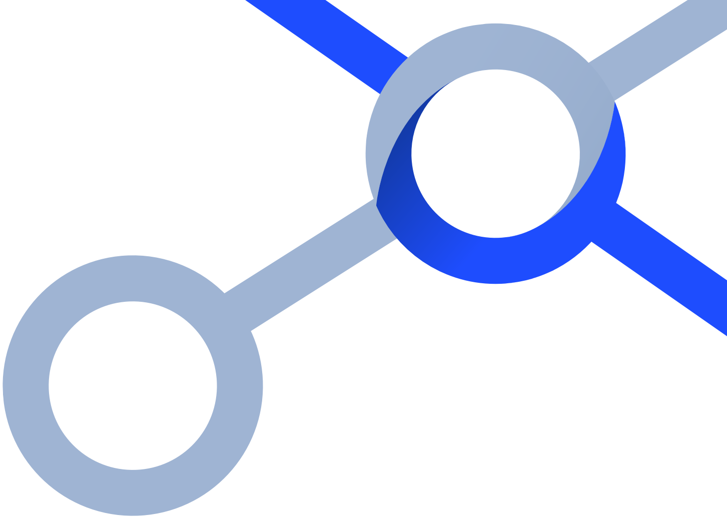
BlockStocks

Liquid Stocks Protocol Whitepaper

Technical Whitepaper
Alpha Version 0.4

May 2018



1. Introduction

Blockchain technology has become an incredibly disruptive trend1 in financial technology. It's combined security2 with the resilience of its underlying P2P network, have led to large organic growth in the broader ecosystem around it. This has led to proven and sustainable markets, a growing industry and expanding technology infrastructure.

Both the technology and the industry are very much in an early stage of development and market demand has shown a strong appetite for innovation and investment opportunities in the space.

DISTRIBUTED LEDGER TECHNOLOGY

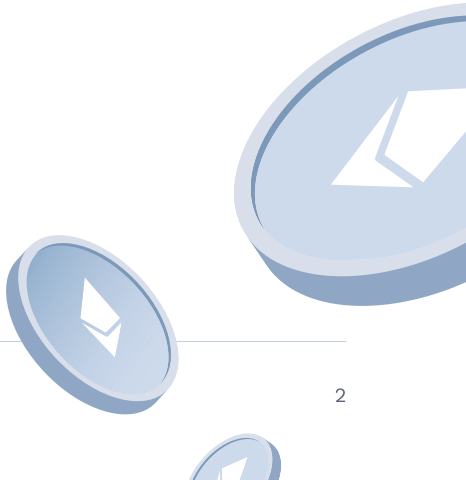
Distributed Ledger Technology (DLT) enables a special form of electronic data processing and storage. A Distributed Ledger is a decentralized database or data processing platform in which network users may share "read" and "write" permissions. Unlike a centrally managed database, this network does not require a central instance that makes new entries in the database. New records can be added at any time by the participants themselves. A subsequent update process ensures that all participants have the latest version of the database. A special form of the DLT is a Blockchain. Yet, other DLT expressions exist, e.g., a Tangle network used in IOTA.

Configurations of DLT systems are dependent on the access possibilities of the participants in a network. Distributed Ledgers can be subdivided into "permissioned" and "unpermissioned" ledgers. While the latter are openly accessible to anyone (such as in the Bitcoin network), access to the ledger is regulated at the former. Participants in networks with permissioned ledgers are generally registered and meet certain requirements in order to access data or to get involved in the consensus computation. The choice of the circle of authorized users (open or limited circle of participants) also involves the choice of the consensus mechanism. For example, Proof of Work (PoW) algorithms are primarily used for unpermissioned ledgers (yet not exclusively, as we will see in the XAIN framework), since the validation of entries requires no trust among the participants. On the other hand, permissioned ledgers use Proof of Stake (PoS) or Probabilistic Byzantine Fault Tolerance (PBFT) consensus mechanisms that require less computational power. The establishment of a basis of trust in this case already takes place through the admission of the participants to the network.

ETHEREUM

After the initial use of Bitcoin as a virtual currency and a startin point for many alternative coins, a second generation of Blockchain applications became possible when smart-contract implementations became simpler with the introduction of Ethereum in 2014 [5]. A developer can create a smart contract and deploy it to the network, every node on the network will receive the byte code of the contract, and make it available in its virtual machine.

Like Bitcoin Ethereum currently uses Proof of Work as a consensus mechanism, but will migrate to Proof of Stake in 2018.



ERC20 STANDARD

ERC-20 defines a common interface for tokens implemented on Ethereum to follow, meaning that developers of software, whether smart contracts or traditional applications handling these tokens, can be sure that their software will operate with all tokens following this standard.

The ERC20 standard specifies that a compliant token will implement the following functions on the right.

It is proposed that the tokens implemented by Liquid Stocks, both the BlockStocks token and the branded tokens issued as part of the ICO creation process will follow this interface.

```
function totalSupply() public constant returns (uint);

function balanceOf(address tokenOwner) public
constant returns (uint balance);

function allowance(address tokenOwner, address
spender) public constant returns (uint remaining);

function transfer(address to, uint tokens) public
returns
(bool success);

function approve(address spender, uint tokens)
public returns (bool success);

function transferFrom(address from, address to, uint
tokens)public returns (bool success);
```

XAIN TECHNOLOGY STACK

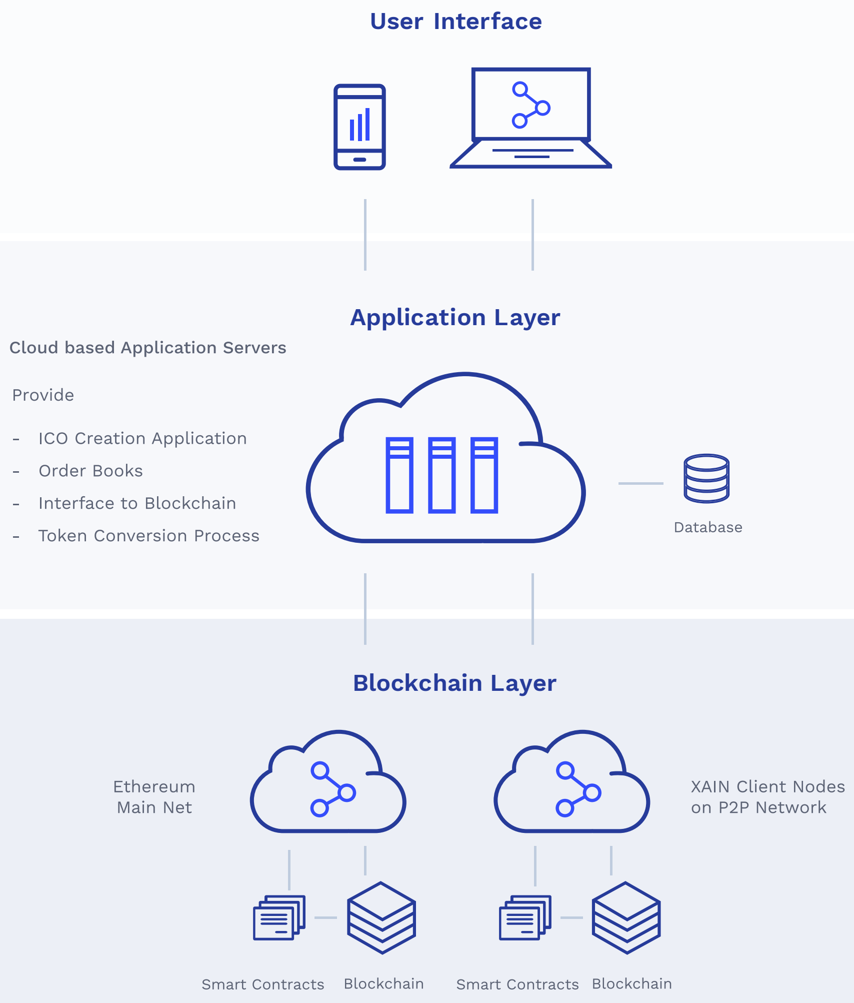
The Xain Technology stack follows the traditional split into 3 layers:

- User Interface
- Application
- Storage layers

in this case the storage layer being a blockchain network, which can hold application logic.

The Xain blockchain is a private blockchain network , though with links to the main Ethereum net as required, in this paper we represent this as a token transfer process.

An optional storage component using IPFS to store large data items is not used here, instead a traditional database will be used, this will be managed by the application layer.



Cloud based Application Servers

Provide

- ICO Creation Application
- Order Books
- Interface to Blockchain
- Token Conversion Process

PRACTICAL PROOF OF KERNEL WORK & DISTRIBUTED ADAPTIVENESS

Practical Proof of Kernel Work: PPoKW

Proof of Work, as currently used in Bitcoin and Ethereum, has been very successful as a consensus mechanism for cryptocurrencies and Blockchain systems. However, it consumes a lot of energy and leads to centralization of mining in standard incentivization structures. Therefore, it seems desirable to retain the advantages of PoW while also containing its energy consumption and mitigating, if not eliminating, centralization of mining. The work in [31] already developed means of minimizing energy consumption of PoW in the "governed Blockchain" setting [32], at a guaranteed level of security. We now want to scale up these abilities.

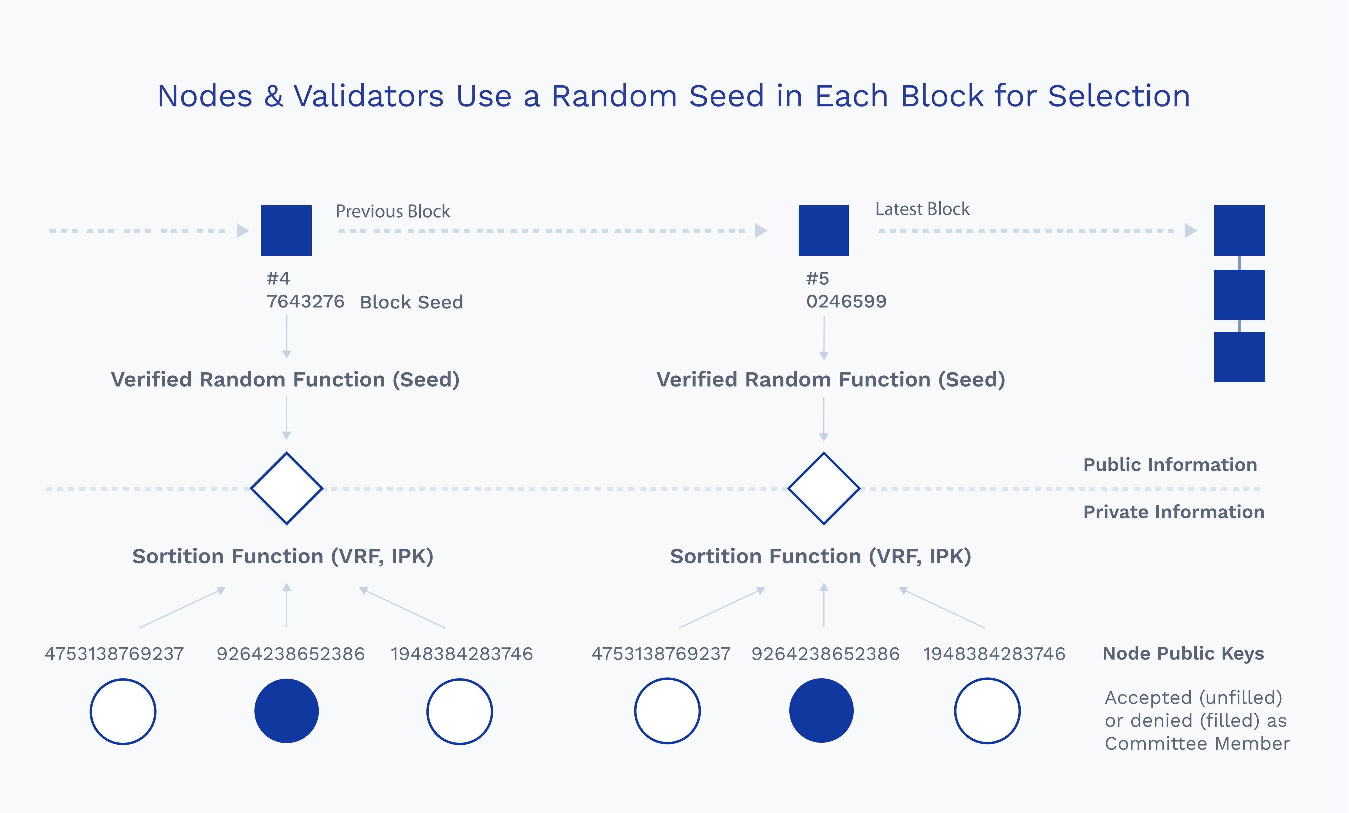
A Blockchain B 0 , B 1 , . . . , B r-1 consists of a linearly ordered list of blocks, where B i has block number i and block height i - 1, the number of blocks that precede B i in that chain. Block B 0 is the Genesis Block. Each block B r with r > 0 is determined by a mining race that also makes B r depend on B r-1 .

Algorithmic Introduction to PPoKW

The process of electing a leader – who can propose the next block on the chain relies on Proof of Work (PoW). We restrict the PoW mining race for the next block by two mechanisms:

- A dynamic White List L which is authenticated on the Blockchain and maintains those public keys that are, in principle, eligible to participate in a PoW mining race.
- An adaptive node selection mechanism, illustrated is based on Cryptographic Sortition as introduced in Algorand, this determines the superset of nodes that may be eligible to participate in specific tasks of Blockchain construction and management. These tasks include mining, machine learning, and management of the white list L.

The above cryptographic eligibility is necessary but not sufficient for engaging in a task: white list L or other security state may override such eligibility.


Nodes & Validators Use a Random Seed in Each Block for Selection
Previous Block
Latest Block
#4
7643276
Block Seed
#5
0246599
Verified Random Function (Seed)
Verified Random Function (Seed)
Public Information
Private Information
Sortition Function (VRF, IPK)
Sortition Function (VRF, IPK)
4753138769237
9264238652386
1948384283746
4753138769237
9264238652386
1948384283746
Node Public Keys
Accepted (unfilled) or denied (filled) as Committee Member

CRYPTOGRAPHIC SORTITION

Distributed Adaptiveness

A crucial element of our engine is a machine-learning model that informs decisions on how to adapt run-time parameters.

Given a budget, an engine optimizes for stability by adjusting internal parameters of a Permissioned Blockchain at run-time. Optimization decisions taken by an engine are based on a continually evolving machine-learning model. We use reinforcement learning (RL) to train machine-learning engines. RL is positioned in between supervised learning and unsupervised learning: Whereas supervised learning has labels for each training example and unsupervised learning has no labels at all, reinforcement learning has sparse and time-delayed labels.

Model We define the set of all possible states the system can be in as B (i.e. the last k blocks from the chain including operational meta-data concerning those blocks). The set of all possible run-time parameters an engine can control is defined as C. With that an engine can be defined as the function

$$engine: B > C$$

mapping from Blockchain state to control parameters.

We define such engines by appeal to a set S of neural network parameters and a set A of actions.

ENVIRONMENT AGENT MODEL FOR PPOKW

State

We represent state S using k triplets each holding the mining time, difficulty, and approximated hash rate (at the time of mining block Bk) of the top k blocks of the chain.

$$S = (hash\_rate; mining\_time; difficulty)^k$$

Following our optimization objective (i.e. a maximum mining time of target seconds), we define robust_target = 0:8 target to determine rewards. We assign positive rewards if the agent is close to the robust_target,and decreasing (even negative) rewards the further an agent diverges from robust_target.

$$reward = - m \; (robust\_ \; target - \; mining\_ \; time_k)^2 + 1$$

PUBLIC AND PRIVATE NETWORKS, VALUE AND UTILITY CHAINS

Public and private networks

The peer to peer network underpinning a blockchain system can be open for anyone to join in the case of a public network, or restricted in the case of a private network. The restriction for private networks can be implemented at both network level using a firewall, or at the peer to peer protocol level, implemented in the client software running a node.

Permissioned Blockchain Systems

Network access control functionality itself, however, can differ in various ways

First of all, it can mean whether users in the system have the means to actually participate in the Blockchain by being a full node or having the ability to create and operate new smart contracts as a programmatic logic. The permissioning of node participation is thereby relatively easy to achieve. Secondly, permissioning the Blockchain network can also mean to influence the mining architecture itself, including the possibility of nodes to participate in the mining game. This control functionality is much harder to achieve, as it requires encapsulated functionalities of the running Blockchain clients, in our case a bespoke client based on the Ethereum clients. The Xain client has been adapted to increase security

This requirements mainly results from the possible ability of nodes to impersonate miner addresses, such that we need to change the actual structure of the block headers to include additional encryption and validation procedures of further parameters, which we define more closely in the optimization sections of this paper.

Value and Utility Chains

In the context of tokens issued by public and private networks we can form a hybrid system allowing token transfer between a public network, here the Ethereum main net, and a private blockchain, here the Xain network. These blockchains are then referred to as the value chain and the utility chain respectively.

The interchange between the networks is handled by a token conversion process, this process runs in the application layer, allowing transfer between the chains and proving the veracity of the transfers.

For this application users will be allowed to purchase BlockStocks token on the Ethereum main net, and use these to buy branded tokens, or alternatively spend branded tokens to buy BlockStocks tokens.

When creating an ICO, branded tokens will be minted, and an equivalent value of BlockStocks tokens will be held in escrow on the main net.

CENTRALISED, DE CENTRALISED AND HYBRID EXCHANGES

Centralised exchanges dominate the cryptocurrency space, they control of funds on behalf of traders when executing market functions. All trade processing is carried out in a centralised and often closed manner. Such exchanges are seen as less robust and less secure than alternative approaches.

At the other extreme, decentralised exchanges allow users full control of their funds and decentralise all processing of transactions. Although more secure and transparent, decentralised exchanges are typically less performant, and have poor liquidity since they rely on an ad hoc set of users to act as market makers.

With a hybrid approach we attempt to de-risk the benefits of a centralised exchange while maintaining the scalability and performance. The limited risk of operating through blockchain based accounts, and the added transparency for the reconciled state of the market in the blockchain, allows for an otherwise centralised system to operate in a strictly controlled and transparent capacity.

OFF CHAIN PROCESSING AND PAYMENT CHANNELS

With the evolution of blockchain technology, multi-asset trading and exchange will increasingly become a larger and growing part of the technology.

One solution to the problem of scalability in such systems is to use off chain processing to implement payment channels. Micro payments for example may be un economic to implement due to relatively high transaction fees involved in the blockchain.

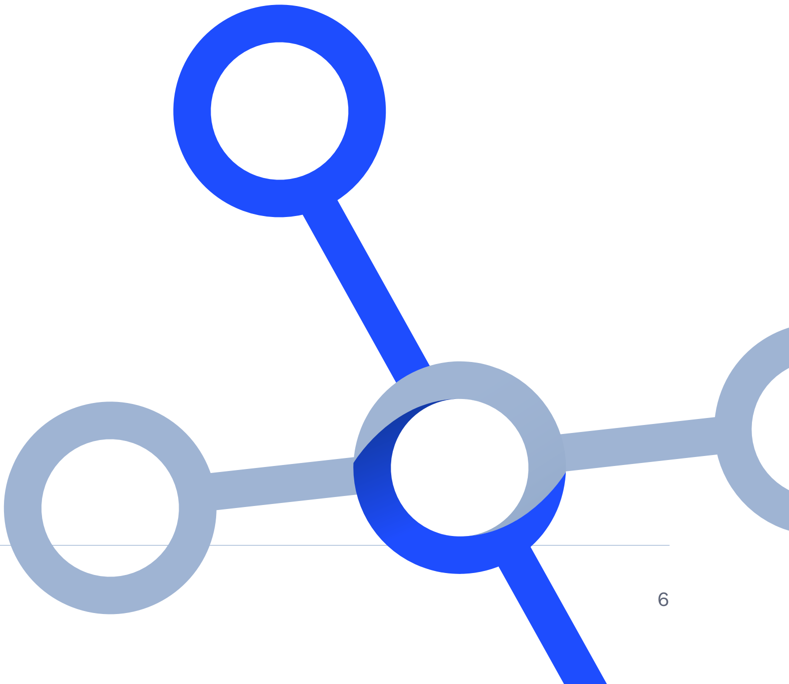
Payment channels solve this by moving the majority transactions off the blockchain, and only reporting the starting and end points of the interaction. The intermediate transactions are still cryptographically secure and either party to the process can close the channel at any point by posting the latest transaction to the blockchain.

COMPLIANCE

In the grey areas around blockchain, of which there are many pitfalls, the future support of mass market adoption must include the support regulatory compliance.

Regardless of the nature of either the blockchain industry, or of the authorities who are surely less than perfect in their application or enforcement of them, regulations are intended to protect citizenry and should be respected as such if blockchain is to have wider acceptance in the market.

Many ongoing reviews of technology be regulators and governing bodies have resulted in varying initial attempts to address blockchain technology, with more sure to follow.



2. User Interactions

CREATING AN ICO

The ICO creation process involves both the value chain (the Ethereum main net) and the utility chain (the private network).

On initiation of the process, the company details are stored in the database. BlockStocks tokens are put into an escrow account on the main net and an equivalent value amount of branded tokens are created on the utility chain.

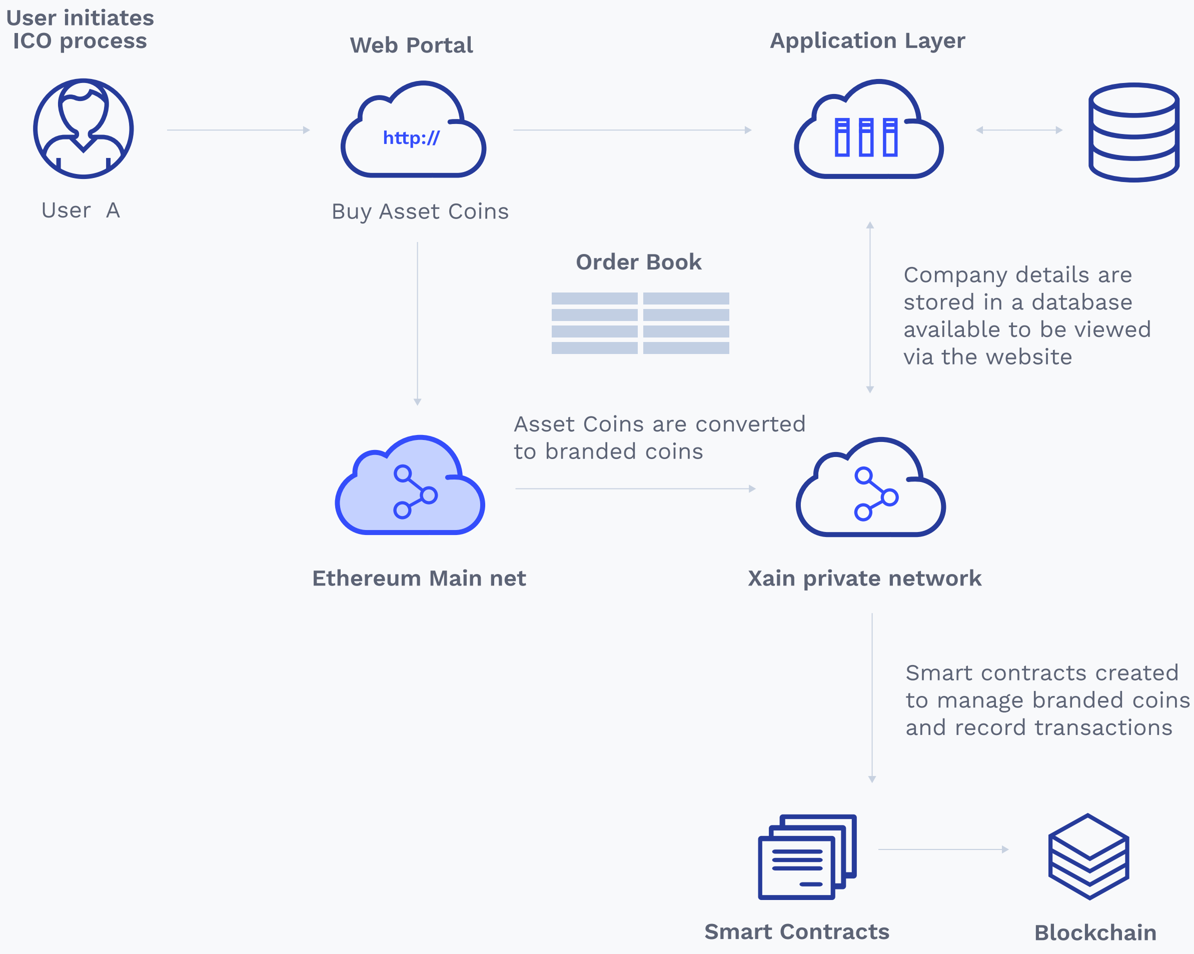
An order book is created to allow trading in the branded token.

TOKENISATION OF ASSETS

If the branded token represents, links to or gives rights to a real world asset, the data representing the asset will be stored in the smart contract used to track the token. For flexibility this data is held in JSON format as an array of strings.

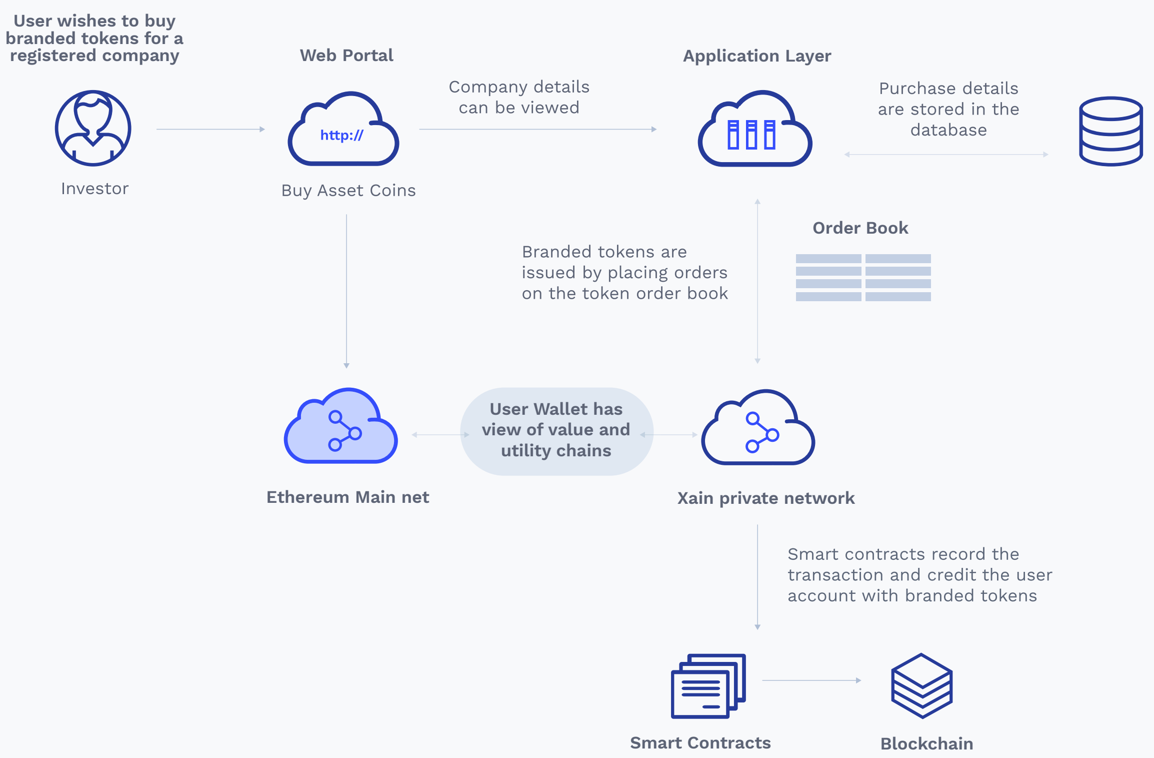
Links to external services, such as multilateral trading facilities, or financial data providers to manage the link between token and asset will be accomplished at the application level, with data flows up to the user interface, and down to the blockchain.

ICO Creation Process


User initiates ICO process
User A
Web Portal
http://
Buy Asset Coins
Application Layer
Order Book
Company details are stored in a database available to be viewed via the website
Asset Coins are converted to branded coins
Ethereum Main net
Xain private network
Smart contracts created to manage branded coins and record transactions
Smart Contracts
Blockchain

INVESTMENT PROCESS

ICO Investment Process


User wishes to buy branded tokens for a registered company
Investor
Web Portal
http://
Buy Asset Coins
Company details can be viewed
Application Layer
Purchase details are stored in the database
Order Book
Branded tokens are issued by placing orders on the token order book
User Wallet has view of value and utility chains
Ethereum Main net
Xain private network
Smart contracts record the transaction and credit the user account with branded tokens
Smart Contracts
Blockchain

Investors can participate in the ICO by buying branded tokens with BlockStocks tokens.

The trade details are recorded both on the utility chain and in the off chain database. The trade is handled by the order book created for trading in the branded token.

3. Liquid Stocks Order Book

SUBMITTING AN ORDER

A standard order book and matching are used to allow tokens to be transferred between users.

The order book is kept off chain so that matching orders can be efficient, but a hash of the order details are stored on the utility blockchain.

An order is defined as

$$O = (User\ ,\ Token_1,\ Token_2,\ Token_1\ Amount\ ,Token_2\ Amount)$$

with

User representing the Ethereum address of the User originating the order

Token being the 3 letter identifier of the token

Token Amount being the amount offered or required. This should be specified in the lowest denomination for that token.

The submit order function will return an order identifier I

$$I = H\,(O, \mathit{timestamp}, \mathit{nonce})$$

with

timestamp being a unix format timestamp

nonce being a strictly increasing integer applied from the originating user account

H being the standard SHA-3 hash function.

When an order is added, the tokens required to pay for the order are added to an escrow account maintained on the blockchain. The matching process runs, and tries to match the new order with existing orders.

If a match occurs, token transfers take place between the originators of the matched orders, and fees are sent to the Liquid Stocks Fee wallet.

The identifier of the order will be stored on the utility blockchain

CANCELLING AN ORDER

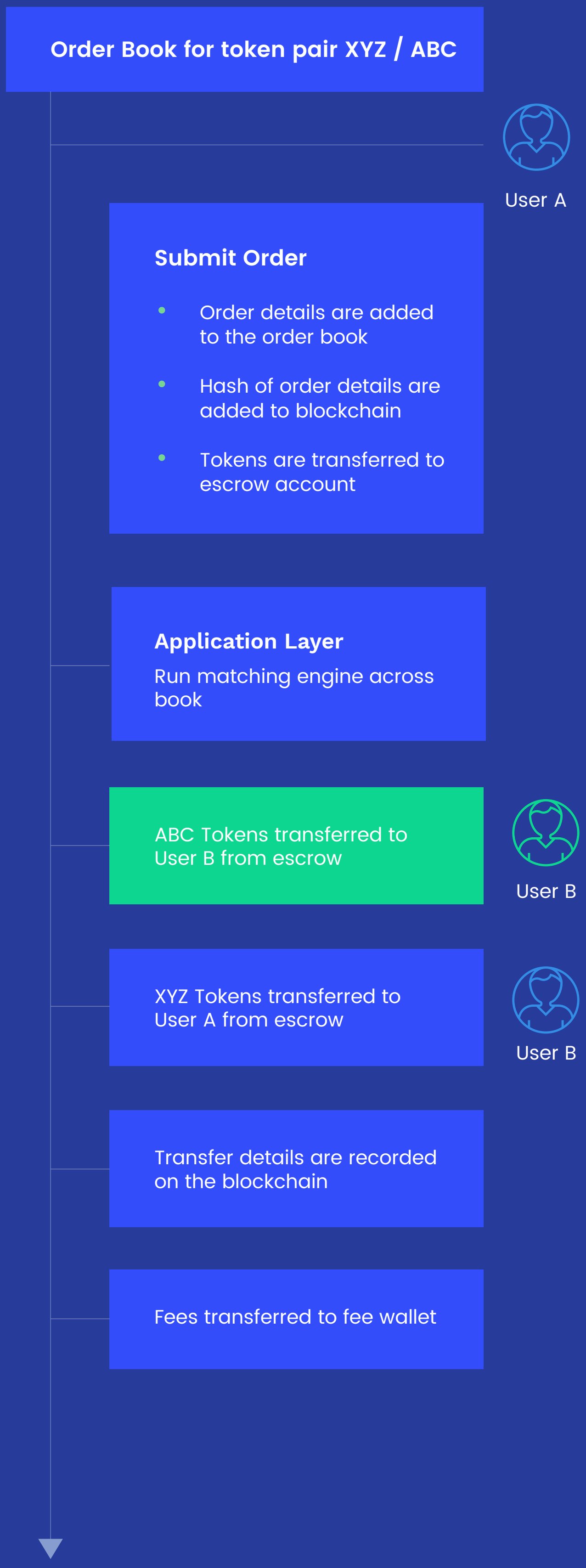
If a user wishes to cancel the order, the tokens can be returned from the escrow account to the user's wallet and the order removed from the order book. The user will need to specify the order identifier I as defined above.

4. Creating a Portfolio and Social Trading

Services that enhance the user experience such as trade analytics or portfolio management which do not require the trust layer provided by the blockchain, are better carried out off chain using the application layer, and using a traditional database for data storage.

If this involves reading data from a blockchain, data reads are transaction fee free and can be relatively fast.

TRADING TOKENS TIMELINE



Order Book for token pair XYZ / ABC

User A

Submit Order

- Order details are added to the order book
- Hash of order details are added to blockchain
- Tokens are transferred to escrow account

Application Layer

Run matching engine across book

ABC Tokens transferred to User B from escrow

User B

XYZ Tokens transferred to User A from escrow

User B

Transfer details are recorded on the blockchain

Fees transferred to fee wallet