PRELIMINARY DRAFT - ideas and specifications proposed in this draft are highly conceptual at this stage. These may be subject to significant revisions based on further discussions with partners, advisors and the wider openANX community after Token Launch

openANX Platform Specifications - Draft V4.0

16th June, 2017

Authors:
Lionello Lunesu Founder Enuma Technologies
Hugh Madden CTO, Co-Founder ANX International
Bok Khoo Founder, Bok Consulting

Notes to Reader:

1. This document focuses specifically on technical matters and must be read in conjunction with the Information Memorandum. In all instances, where there is a discrepancy or conflict between this document and the Information Memorandum, the Information Memorandum shall always prevail. All terms defined in the Information Memorandum which are not otherwise defined in this document have the same meaning.

2. This document outlines the openANX Platform as it is currently envisaged to operate in future. Nothing in this document should be taken to imply that any particular feature will necessarily become or continue to be available to any OAX purchaser of Member of the openANX Platform. As at the date of this document,the features outlined in this document either do not yet exist or remain in prototype only. Additionally, all legal rights or privileges pertaining to OAX are as described in, and subject to the terms of, the Information Memorandum.

3. The project as envisaged in this document is under development and is being constantly updated, including but not limited to key governance and technical features. Accordingly, if and when the project is completed, it may differ significantly from the project set out in this document.

4. The views and opinions expressed in this document are those of the authors and do not necessarily reflect the official policy or position of any government, quasi-government, authority or public body (including but not limited to any regulatory body of any jurisdiction) in any jurisdiction. information contained in this document is based on sources considered reliable by the authors but there is no assurance as to their accuracy or completeness.

Background

Introduction

The openANX project is concerned with merging the best aspects of decentralized exchanges with those of the current, incumbent centralized exchanges. It aims to provide an open platform to allow centralized exchanges to focus on their core competency, regulatory compliance and integration with traditional off chain assets such as fiat currencies. These core competencies will then be integrated with the new breed of decentralized exchanges to migrate the current critical mass of trading activity away from centralized exchanges and into the public, open domain.

It also aims to supplement both models with:

- consumer protection through:
    - collateralization

    - dispute resolution
    - reduction of funds held under custody (Ethereum and Ethereum tokens will not be held under custody)
- credit risk markets to provide price discovery on credit risk
- liquidity aggregation to ensure necessary levels of liquidity
- lowering the barrier of entry to new market participants through cost savings through shared open source software, and liquidity critical mass through liquidity aggregation

A token sale is used to finance the project and mobilize a incentivized community of users and suppliers through the sales of tokens termed “OAX”. OAX tokens are used to redeem memberships in the platform and register tokens and order books. There are several types of memberships, voting memberships, founding memberships, gateway memberships, and service provider memberships.

Tokens are used to exchange for membership, and destroyed after such exchange. Memberships themselves are transferable.

Outside of technical developments, there is also a significant legal and compliance effort underway; however this paper focuses specifically on technical matters.

The openANX technology deliverables as currently formulated include:

- A number of Ethereum smart contracts forming the backbone of the project, co-ordinating:
    - Memberships
    - Collateral management and dispute resolution
    - Governance and upgrades
    - Compliance and identity services (Know your Customer and Anti Money Laundering)
    - Token and order book registries
    - Settlement for asset exchange; both over the counter (OTC) and order book based
- Enhancements and integration of off-chain asset swaps protocols
- A reference implementation of a user interface that allows openANX users and service providers to interact with the smart contracts and off-chain matching engines
- A reference implementation of an API that allows openANX users to integration automated trading and extract market data streams

Note that off-chain functions such as banking integration and KYC data collection is not within scope. It is expect that service providers will use their existing centralized infrastructure for these purposes, with points of integration to the Ethereum contracts.

Functional overview

Membership

OAX token holders may be able to exchange their tokens for differing types of memberships. The OAX tokens are destroyed upon exchange for memberships.

Memberships will be linked to an address and will be able to be transferred.

Memberships may eventually be quite a complex hierarchy of sub users, and it can be expected that users may wish to keep memberships cold and nominate lower privilege accounts for day-to-day operations; however this is beyond the scope of this prototype.

Actor/ Use Case	**Functionality**
User Exchange OAX tokens for membership	Register for membership by invoking smart contract which will cause debit to ERC20 OAX balance. (there will no credit, thus reducing the overall supply of tokens)
User Transfer membership	Ownership of membership transferred to another user through smart contract.

Collateral Management & Dispute Resolution

Asset gateways may register collateral in the form of Ethereum by sending ETH to a payable method on the collateral management smart contract.

This collateral is locked for a period of time. It may additionally be locked by pending dispute cases.

End users can raise a dispute case against an asset gateway. They must pay a relatively small amount of ETH(the price amount or method of calculation for this is yet to be determined) which will be locked pending the outcomes of the dispute resolution, to prevent frivolous/malicious attacks on asset gateway businesses.

Dispute facilitators will be selected annually with the ability to vote “M-of-N” style for the rejection or approval of outstanding disputes.

Dispute facilitators are also responsible for the real world facilitation of dispute cases through pre-determined dispute tribunal.

The methodology for sizing collateral returns through this process is a subject for more formal legal framework and outside of the scope of this functional overview.

Memberships are linked to an address and can be transferred.

Collateral is pledged against specific ERC20 tokens, such as for example an ANXUSD token, however there is no minimum or maximum level of collateral to be pledged. Users and agencies may examine the collateral pledged, its current value, and the value of gateway tokens (such as

ANXUSD) to assist in the quantification of an asset gateway's credit risk.

Some interesting opportunities arise if the expected Proof-of-Stake protocol Casper allows contracts to stake, however this seems unlikely/ highly speculative at the current time.

Actor/ Use Case	**Functionality**
Asset Gateway Pledge collateral	Pledge collateral by sending ETH to a payable method on the collateral management contract, specifying the relevant tokens against which it is pledged.
Asset Gateway Redeem collateral	Seek return of collateral by invoking collateral management contract, which can return ETH after specified time lock periods and pending dispute queues.
User Raise dispute	Initiate a dispute resolution case by pledging an amount of ETH to the collateral contract and specifying their membership and Asset Gateway, ERC20 token of relevance, and original transaction size.
User Cancel dispute	Close a dispute resolution and return ETH to user by invoking method on smart contract.
User Claim dispute	In the event of a positive dispute outcome, invoke the collateral contract to claim a dispute in ETH.
Dispute Facilitator "M-of-N" vote on dispute	The collateral smart contract will support the voting of platform registered dispute referees with a configurable threshold, most likely requiring high numbers of registered referees on a scale. A successful vote would see a claim either denied or accepted. It is not anticipated denied claims would see the return of the plaintiff's original amount.

Governance & Upgrades

Certain types of memberships will entail rights to vote, and rights to table agenda items for voting.

Business processes, governance rules, and static data (such as membership costs) are managed through the deployment of new smart contracts, and setting static data in the openANX contracts.

It is intended to modularise openANX components and user interfaces, and a service locator pattern is offered to allow modules (contracts) to be swapped out through a governance and voting system.

It is anticipated that the mechanism for most of these changes is as follows:

- An openANX improvement proposal (OAXIP) is created to document and facilitate peer review of a system change
- A developer volunteers (or is engaged by the Foundation) to create a new implementation of a module. The code is made available, and tested by the ecosystem, Foundation appointees, and suppliers to the openANX Platform
- The contract is deployed to Mainnet, and a voting proposal raised by a founding member
- The service locator contract reference is updated to new module implementations by successfully voted items
- Major contract updates that involve Application Binary Interface (ABI) changes are achieved by deploying the new module contracts and permanently halting the old version. This requires support of both the Voting Members and Dispute Facilitator (see below.)

Certain modules, such as those holding data stores, require special migration processes after contract deployment, migrating earlier contracts' state to the new version of the modules' contracts.

Actor/ Use Case	**Functionality**
Founding Member Raise vote on module replacement	A smart contract address is submitted for deployment depending on a voting outcome.
Voting Member Vote on the deployment of an updated module	Invoke a vote method for a specific proposal, a critical threshold will entail an update to the persistent store of service addresses.
Dispute Facilitator Halt System	Is it anticipated that a Dispute Facilitator can halt all transactional activities if needed. A formal vote would be required to resume activities.

Compliance and Identity

It is anticipated that many asset gateways holding fiat funds under custody will have KYC/AML obligations in order to legally operate.

Upon registration, and asset gateway will nominate a KYC/AML service along with supported custom ERC20 token contracts.

Gateway tokens generally require one of the following:

- No KYC at all (tokens can be transferred to or from any valid Ethereum address);
- Boundary KYC (tokens can only be minted to, and exchanged against, addresses that have passed KYC, however minted tokens can be freely transferred between addresses); or

- Full KYC (tokens can only be minted to, exchanged against and transferred to and from addresses that maintain the stated KYC service approval)

The custom ERC20 token implementations will contain KYC controls as implemented by the asset gateway, or it may call out to an external, potentially shared KYC service.

This current prototype is focussed on simply recording KYC approvals directly against accounts, howeverit is anticipated that this may evolve into a system of bearer tokens, where a member signs off-chain a token for one of many accounts, allowing some degree of privacy through all interactions with openANX.

Actor/ Use Case	**Functionality**
Non KYC Token Exchange Claim Approve & Transfer	Tokens issued by gateways without any KYC restrictions can be freely exchanged back to the asset gateway, claimed as a result of minting by a gateway, or transferred to other unregistered addresses.
Boundary KYC Token Exchange Claim Approve & Transfer	Tokens issued by gateways with boundary KYC restrictions will require KYC status checks for minting and exchange, but otherwise can be freely transferred to or from other unregistered addresses.
Full KYC Token Exchange Claim Approve & Transfer	Tokens issued by gateways with full KYC restrictions will require KYC status checks for minting, exchanges and transfers.

Token and Order Book Registry

The token and order book registry allows for the location, static data, and discovery of openANX supported ERC20 tokens, as well as establishing trading pairs.

Registration of tokens and order books may require a one-time fee paid in ETH, OAX or other cryptographic tokens. Duplicate registrations are not supported.

Actor/ Use Case	**Functionality**
Voting Member Register token	Register an ERC20 compatible token in the registry, with a sum of OAX tokens which are subsequently destroyed.
Voting Member Register orderbook	Token pairs are registered as an order book, along with nominated exchange channel, with a sum of OAX tokens which are subsequently destroyed.

Settlement for Asset Exchange

openANX asset exchange contracts will be integrated with decentralized exchange matching engines. At least one-to-one OTC settlement, and order book matching will be supported.

OTC settlement refers to two parties meeting off-band (chat, whisper, voice) and agree to a transaction. Both parties ensure their on-chain token balances are adequate for settlement, and both parties submit a signed transaction which commits to the atomic swap of assets between the two parties.

Order book matching refers to placing orders to buy or sell with certain parameters, to any party. Centralised order book rules have a “fairness” system where the matching of oldest and better priced orders is enforced.

In most decentralised order book implementations, the selection of the best priced orders is up to the user (or their software), and there is no concept of “oldest” orders having priority when other parameters are equal.

Actor/ Use Case	**Functionality**
User OTC Settlement	Submission of a one-to-one asset exchange match for settlement through openANX (ERC20 compatible tokens).
User Opening a state channel for asset swap (TBC)	
User Closing a state channel for asset swap (TBC)	
User Submission of anti-cheat transaction (TBC)	

Architecture overview

Components

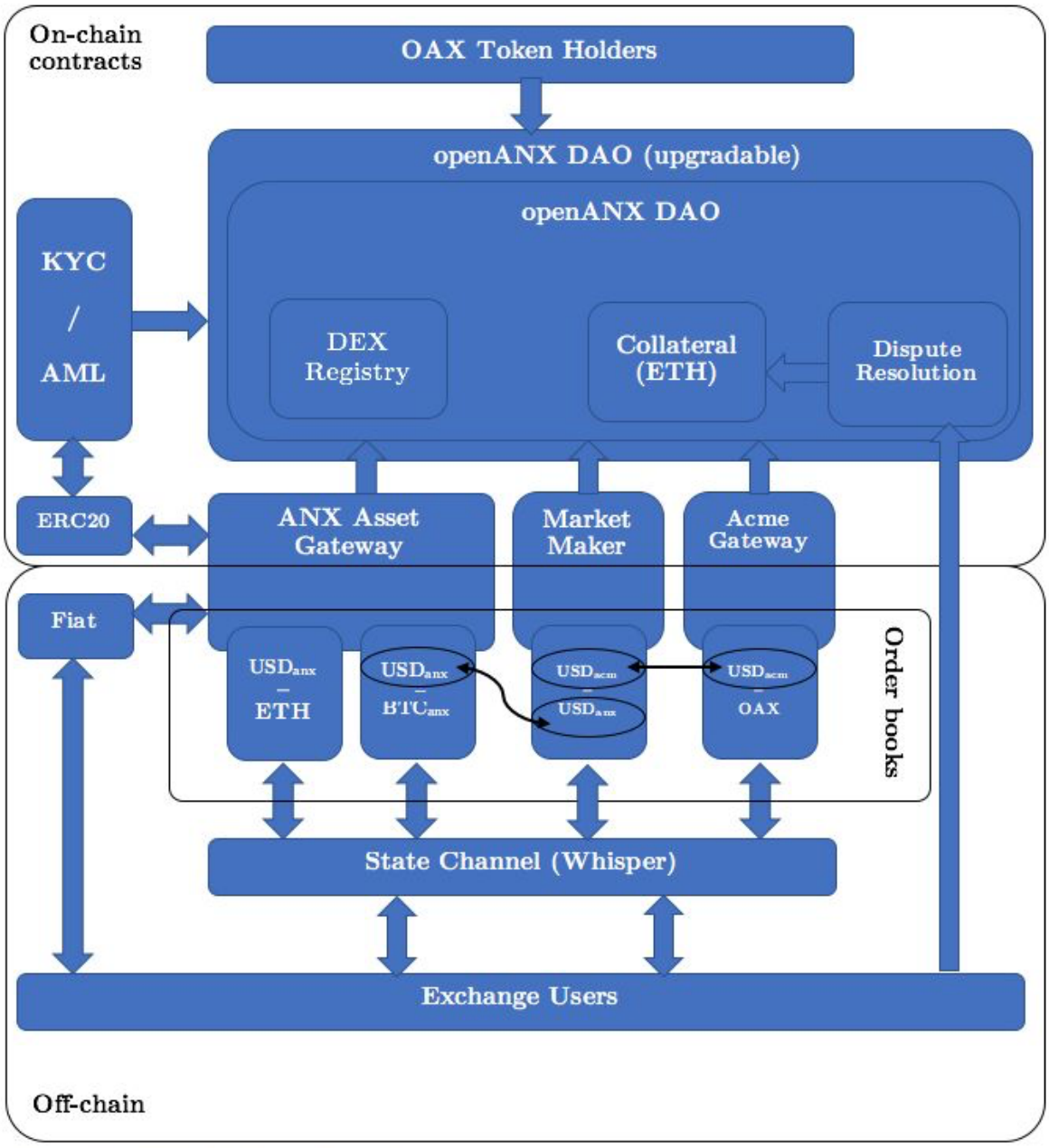

Figure 2 openANX Architecture diagram

OTC Order Matching

The over-the-counter experience for an asset exchange is essentially an atomic token swap from two ERC20 token contracts: in a single transaction, the 'transferFrom' function is invoked on both tokens. On error, the transactions are rolled back and the state of each token is reverted. (Note that a failed trade still consumes gas).

openANX will be on-chain OTC compatible with the 0x Project (https://0xproject.com) and Consensys' Swap (https://swap.tech/whitepaper).

0x

An orderbook can choose to list 0x-compatible orders. An OTC transaction using 0x protocol takes place as follows:

1. A taker uses its client (dApp) to generate a signed 0x *order*, specifying which token to buy and which to sell and their values.
2. The client queries the openANX registry and publishes the intent to an exchange that lists the tokens in the order.
3. The exchange verifies the order signature and, if applicable, checks whether the required KYC/AML requirements for the tokens have been met. 0x-compatible orders must list their 0x exchange contract address.
4. A market maker can settle the order on-chain by invoking the *fill* method on the 0x Exchange contract.

Swap

Alternatively, trades can be settled using Swap. Similar to 0x, Swap-compatible orders are identified as such on the exchange order book.

1. A taker creates a Swap *intent*, specifying which token to buy or sell and how much.
2. The client queries the openANX registry and submits the intent to a Swap-compatible exchange ('indexer') that lists the tokens in the order by calling *intent.add*.
3. It verifies the order signature and, if applicable, checks whether the required KYC/AML requirements for the tokens have been met. 0x-compatible orders must list their 0x exchange contract address.
4. A taker uses its client (dApp) to call order.request on the found makers, specifying which token to buy or sell and how much, and invokes the Swap *order.request* API on the listed makers.
5. Several makers call order.provide on the taker.
6. Taker selects an order and calls *fill* on the Swap contract.

Off-Chain Order Book Matching

Many proposals for a "decentralized exchange", like 0x and Swap, have off-chain orders, but on-chain settlement. Having to pay gas to write each settled trade to a single ledger (albeit a decentralized one), and wait for blocks diminishes the liquidity of the market. Instead, openANX will leverage state channels, established by the exchanges, to efficiently trade assets without having to invoke an on-chain contract for every match.

In order to establish a payment channel, a trader must make a deposit of the asset that is being sold. This ensures that a receiver can trust the payment without having to immediately settle the trade on the chain. openANX is investigating a novel payment channel algorithm "Sprites" that scales better with the length of the path through the network (https://arxiv.org/pdf/1702.05812). The described payment channel algorithm will be extended to support multi-currency asset exchange along the path. Furthermore, openANX is considering using Whisper (https://github.com/ethereum/wiki/wiki/Whisper) for all channel communications.

Note that the Whisper protocol only allows for small sized messages. It is expected that larger messages will be built with an algorithm to stitch together the small Whisper messages (e.g. like using small network packets to build a TCP packet, with some checksums and retries).

Initially there will not be a centralized matching engine. openANX is investigating whether the off-chain matching and routing can be completely done in the client: the client would identify the required order-books, create off-chain order, and submit the order to the channel. Only after all order books along the identified path have completed the transfers will the client receive the updated state from the exchange. A similar pathfinding algorithm has been used successfully in Ripple.(https://ripple.com/build/paths/)

Final settlement can be achieved by closing the channel with the exchange. This happens by the client submitting its last state to the contract. During an anti-fraud period, the exchange, being the counterparty in the channel, can provide its state of the channel, in case the state provided by the user is not the latest known state. When the anti-fraud period times out without dispute, the balances in the on-chain token contract are updated and the channel is closed.

Order Book Aggregation

Small disjoint order book for multiple asset gateway tokens can be aggregated to provide a single functional pool of liquidity.

Consider the situation where there are three order books:

1. ETH/ANXUSD
2. ETH/ACMEUSD
3. ANXUSD/ACMEUSD

By themselves order books 1 and 2 reflect individual fragmented pools of liquidity.

If there is an active credit risk order (i.e. book 3), matching logic can match trades by combining the

three order books into a single order book with simple graph. This implicitly requires the ability to settle 3 different matches atomically.

In this fashion, an active credit risk order book can transform small individual exchange pools of liquidity into a single large order book. This approach may at last bring about the liquidity network effect to turn a decentralized, open exchange into the dominant source of exchange liquidity.

Note that intermediate order books (i.e. 3 above) can be cross currency, allowing the pooling together of liquidity across different sovereign fiat currencies. A similar pathfinding algorithm has been used successfully in Ripple.(https://ripple.com/build/paths/)

openANX UI and API Reference Implementation

openANX will implement a:

- fully functional reference implementation user interface (most likely Mist however Coinbase token will be considered)
- API services for automated trading and market data integration