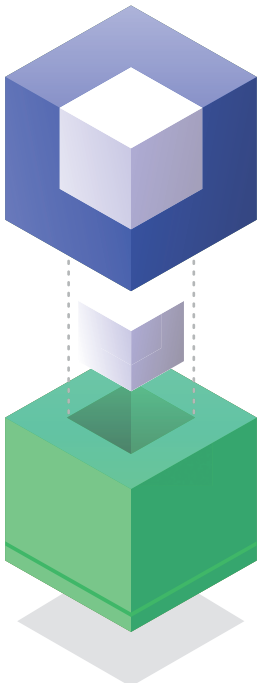
NEX

A platform for decentralized cryptographic trade and payment service creation

v 1.1



Latest publication date: Nov. 23 2017

http://www.neonexchange.org

NEX: a platform for decentralized cryptographic trade and payment service creation

Ethan Fast, PhD*
Neon Exchange
Stanford, CA. USA
ethan@neonexchange.org

Fábio C. Canesin, MSc
Neon Exchange
Cambridge, MA. USA
canesin@neonexchange.org

Luciano Engel, Eng
Neon Exchange
Florianópolis, Brazil
luciano@neonexchange.org

Fabian Wahle, PhD
Neon Exchange
Zürich, CH
fabian@neonexchange.org

Thomas Saunders
Neon Exchange
Minneapolis, MN. USA
tom@neonexchange.org

Abstract

Today, cryptocurrencies are primarily traded on centralized exchanges where user funds are at risk to hackers and platform managers. Decentralized exchanges (DEXs) allow users to retain control of their funds as trades are mediated by smart contracts on a blockchain, but on-chain computation is generally too slow to keep up with high volume order books. This paper describes Neon Exchange (NEX), a new decentralized exchange on the NEO blockchain that applies a publicly verifiable off-chain matching engine to handle massive trading volume and support complex orders (such as limit orders) that are not possible on existing DEXs. NEX also introduces a payment service and funds management layer that enables third-party smart contracts on NEO to send and receive global assets as part of their computation. To fund developments and future expansion into related services, NEX will issue 50 million tokens that give holders a share of profits in its services.

1 Introduction

Cryptocurrency markets have grown enormously in recent years, from a daily trade volume of $60 million in January of 2015 to more than $8 billion in November of 2017 [1]. Despite the fact that most cryptocurrencies are secured by decentralized architectures, almost all trades between currencies take place on centralized exchanges, where funds must be deposited under the control of the entity facilitating exchange. This layer of centralization puts user funds at risk to hackers and platform managers. Most famously, millions of dollars worth of Bitcoin were stolen from Mt. Gox in 2011, and again from Bitfinex in 2016 [27, 20].

Recently, decentralized exchanges have emerged to allow users to trade without giving up control of their funds [26, 2]. Under these systems, trades are executed by smart contracts on a blockchain, removing the need for a centralized third-party to control user accounts. While these exchanges succeed at their primary goal of decreasing third-party risk, their success comes at the cost of a huge loss of trading performance. Smart contracts are far too slow to execute the complex matching logic of order books on high-volume, centralized exchanges. In practice, this means that users cannot execute complex trades, and presents opportunities for arbitrage on stale orders [11].

If centralized exchanges provide speed, and decentralized exchanges provide security, then it seems natural to ask: can a hybrid system provide the best of both worlds? In this paper, we propose that

*Computer Science PhD to be granted in May 2018 by Stanford University

the optimal mix of speed and security is provided by *a decentralized exchange with a fully off-chain matching engine*. Order matching is by far the most computationally expensive operation when running an exchange. By encapsulating this component in a trusted off-chain service, we can reap enormous improvements in speed, and also support complex orders such as limits or margins. At the same time, by committing orders on-chain as they are matched—with provable deterministic behavior—we can retain the security benefits of traditional decentralized exchanges.

Neon Exchange (NEX) is a new decentralized exchange that embodies these ideas, built on the NEO blockchain. This white paper presents our vision for the NEX platform, the performance benefits of our technical approach, and how NEX fits into the broader NEO ecosystem. We also discuss our roadmap over the coming months and plans for a public token sale.

2 Background

2.1 Blockchain and Smart Contracts

A blockchain is a decentralized ledger that can record transactions between two parties in a verifiable and permanent way without the need for a central authority [24]. In 2008, Bitcoin emerged as the first public blockchain with large-scale adoption as a digital currency. Other chains have since attempted to improve on this technology. Most notably, Ethereum launched in 2015 as the first blockchain with programmable, Turing complete smart contracts [28]. Smart contracts allow developers to publish programs on a blockchain that anyone can inspect, and that will deterministically execute to accomplish complex goals in a way verifiable to all involved third parties. For example, a smart contract might accept incoming funds from a user, then release them at a certain date, or collect funds from a series of users and split them evenly. These smart contracts are what make possible more sophisticated distributed on-chain applications such as decentralized exchanges.

2.2 Decentralized Exchanges

Many decentralized exchanges have emerged over the years. In this section, we lay out the trade-offs in this design space, and how NEX contributes over existing systems. In summary, NEX trades a small degree of user trust for vastly improved performance and usability.

The earliest decentralized exchanges placed order books directly on the blockchain [4, 3]. In these systems, market makers must perform on-chain transactions every time they want to place, modify, or cancel an order. Further, as new orders are placed, a smart contract must execute matching logic that runs slowly (and redundantly) on all virtual machines in the network. In general, these exchanges take up a large amount of network bandwidth and operate very slowly, so very few decentralized exchanges operate under this scheme today.

A second class of systems uses automated market maker (AMM) smart contracts, as opposed to an on-chain order book [22, 23]. These systems adopt a price adjustment model, where all parties trade with the AMM, and the spot price of an asset is determined by the resulting market forces. While AMMs provide increased availability and performance over on-chain order books, they are still much slower than centralized exchanges and must place artificial constraints on supply to prevent their working capital from being depleted by potential arbitragers.

State channels have been proposed to reduce network overhead for the more general problem of asset exchange [21, 6], allowing two parties to iterate on a transfer off-chain before ultimately committing to it on-chain. However, state channels are expensive to open and close, usually requiring a security deposit and a series of on-chain transactions. For this reason, they are most useful among known parties who want to manage a series of interactions (e.g., a "bar tab"), not a single party conducting one transaction with a broader market.

Building from state channels, a more recent class of DEX is based on off-chain relay [26, 2]. In these systems, market makers broadcast an order off-chain, which can then be picked up by an interested counterparty and passed to a smart contract for fulfillment. These systems require far fewer on-chain transactions to perform a trade, but still suffer from performance issues in comparison to centralized exchanges. Notably, order matching is not automatic in these systems, presenting arbitrage opportunities against users who are slow to cancel their orders. Similarly, the absence of matching means that users cannot place more complex orders such as limit buys or market sells.

In contrast to these approaches, we introduce a new kind of decentralized exchange, NEX, based on a trusted, off-chain matching engine [5]. This matching engine works exactly like its equivalent in a centralized exchange, but only has control over active orders, and commits trades on-chain without access to the full balance of a user account. Like exchanges based on off-chain relay, NEX orders are matched off-chain and fulfilled on-chain, but NEX's automatic matching engine reduces opportunities for arbitrage and allows for more complex orders. To ensure trust, NEX provides a public record of orders and a deterministic specification of behavior, so that users can verify orders matched off-chain and claim an award in the event of incorrect behavior. Taken together, this makes NEX the first decentralized exchange with performance comparable to today's centralized exchanges.

2.3 The NEO Blockchain

NEO was launched in 2015 as China's first public blockchain. Recent improvements to the network have made it a compelling alternative to Ethereum for smart contracts and distributed applications [29]. NEX will run first on NEO, before later expanding to support exchange on Ethereum. While most of the ideas behind NEX apply to both platforms, there are several major differences between NEO and Ethereum that are relevant to decentralized exchanges.

2.3.1 Modeling User Balance

Ethereum is based on an *account model*, where a user's balance of ETH is stored as a number in the Ethereum Virtual Machine (EVM) and can be easily modified (e.g., sent or received by smart contract logic) [12]. In contrast, global assets in NEO such as NEO and GAS are based on a *UTXO model*, where funds are sent and received through a chain of spent transaction ids on the network [10]. Notably, these differences only apply to global assets on the NEO network and not tokens created through smart contracts, which behave similarly to ETH [13].

Each system design has trade-offs. In Ethereum, for example, it is easy for a smart contract to interact with a user's balance of ETH, but difficult for a node to prove that a transaction has taken place without syncing the full chain and running the EVM. In contrast, it is easy for third parties in NEO to verify that a transaction has taken place on the chain (e.g., through SPV [9]), but much more difficult for smart contracts to program interactions with a user's NEO or GAS balance.

For NEX to succeed, smart contracts on NEO require some way to programmatically interact with global assets like NEO and GAS. To solve this problem, we introduce a novel *payment service layer*, that converts global assets into smart contract tokens, which can then be easily interacted with by smart contracts on the NEO network. Users can convert their global assets into tokens by depositing them at the payment service address, then later withdraw them from that contract address (perhaps with a different balance), whenever their interactions with a third-party smart contract have been completed. We believe this solution will generalize to other networks that combine UTXO models with independent smart contracts, such as Cardano [7].

2.3.2 Calls Between Smart Contracts

A second major difference between NEO and Ethereum is how and when smart contracts are allowed to call each other in the course of execution. In Ethereum, a smart contract can dynamically call any other smart contract, given an address passed at run-time. In contrast, NEO enforces that all calls between smart contracts must be declared statically in advance [14]. This constraint makes it much easier for NEO to implement sharding optimizations across VM state, but means that NEX smart contracts must hard-code all token pairs that are supported by the exchange.

2.3.3 Consensus

NEO and Ethereum also operate under very different consensus models. NEO uses delegated Byzantine Fault Tolerance (dBFT) for consensus [19, 25], whereas Ethereum uses Proof of Work (PoW) [8]. NEO's consensus model allows for much higher theoretical transaction throughput (up to 10,000 tps) which has a huge positive impact on the performance of a decentralized exchange. As Ethereum moves to a Proof of Stake (PoS) model in 2018, NEO's comparative advantage may diminish, but many details have yet to be worked out before that transition occurs [15].

3 Neon Exchange

Neon Exchange (NEX) aims to combine the performance of centralized exchanges with the trust and security properties of decentralized exchanges. The system consists of three main components: an off-chain trade matching engine, a smart contract where trades are executed, and a payment service where global assets such as NEO and GAS can be converted to tokens that can be transfered directly by smart contracts, making them compatible with the exchange. In the following sections, we describe each component in more depth.

3.1 Off-chain Matching Engine

An off-chain matching engine allows NEX to benefit from the performance characteristics of centralized exchanges, while maintaining a decentralized user account model based on the blockchain (Figure 1). Orders are signed and sent from user addresses to the matching engine, where they are quickly and deterministically processed using high-performance hardware. Matched orders are then signed off-chain and committed back to user accounts on the blockchain.

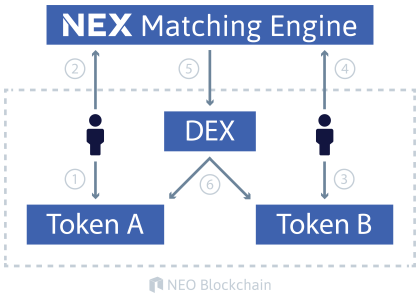


Figure 1: The NEX architecture provides fast, decentralized exchange using an off-chain matching engine. Here we illustrate an example user interaction with NEX exchange. First, one user authorizes a trade to exchange Token A for Token B (1) and sends the order to the matching engine (2). Next, a second user authorizes and submits a trade for Token B in exchange for Token A (3-4). The engine matches the orders (5) and submits them to a smart contract for execution (6). Note that steps (1-2) and (3-4) can be initiated either via API call or the NEX exchange website.

To trade on NEX, a user must first authorize NEX to access the amount of token to be traded by calling the NEP-5 *approve* method on the token's smart contract. The user can then submit a signed JSON request to the NEX matching engine API. Once the order is matched off-chain, the engine will call the NEX smart contract to execute the order. Because a single invocation transaction on NEO can contain many smart contract calls, the engine can batch a set of matched orders in one on-chain transaction to minimize computation. Assuming 1,000 transactions per second, NEX could potentially execute more than 100 thousand trades per second on the chain. In the future, such batches could adopt match-rings to further enhance liquidity [18].

3.2 Trusted Off-chain Matching

While an off-chain matching engine brings enormous performance benefits, it also opens the door to potential trust issues between users and the exchange. How do users know that the engine is matching orders fairly, for example, and not manipulating the order books to its own benefit? To address this problem, we propose the idea of *provable fair off-chain matching*. Under this scheme, the off-chain matching engine follows a publicly specified deterministic algorithm. By combining this knowledge with a public ledger of the order in which trades have been sent to the exchange and fulfilled on the blockchain, any user can verify that the exchange is operating fairly. To make this trust in NEX even more explicit, in the future we plan to build a smart contract where users can submit evidence of unfair exchange behavior in return for a large reward.

Concretely, matching on NEX occurs deterministically based on price and time, commonly known as FIFO [16]. Lower priced orders will be matched first, with preference given to orders placed earlier in time at a given price level. Any modifications to an order will reset its placement time.

3.3 Centralized User Accounts

The security problems of centralized exchanges are not simply a technical challenge to be overcome, but also a social consequence of the common user desire to hold assets in exchange accounts. This desire is largely due to the familiarity of the bank-like user experience provided by these centralized platforms when managing funds. NEX aims to bring a similar user experience to decentralized exchange by storing a user's encrypted private key client-side in a user's browser. This preserves the security guarantees of a decentralized account model while allowing users to login into NEX through a traditional web form that asks for a username and password.

3.4 Types of Orders

Unlike existing decentralized exchanges, which only support point-to-point orders that allow tokens to be traded at a fixed price, NEX supports more complex trades such as limit and market orders. Below we describe the types of trades available in NEX (Table 1):

Table 1: Order types supported by NEX

Type	Description
Limit	Exchange tokens above or below a given price ratio
Market	Exchange one token for another at the current market price
Margin[2]	Borrow with leverage to go long or short on a token

NEX is able to support these complex order types due to the speed and flexibility of its matching engine, which is not limited by slow computation cycles on the blockchain.

3.5 Exchange API

NEX exposes a public JSON API that third-party applications can use to trade tokens. This API allows users to place, modify, and cancel orders on the matching engine. Because these transactions take place off-chain, the NEX API can handle tens of thousands of requests per second, in-line with popular centralized exchanges [17].

To submit an order to the matching engine, a client must make a JSON request that is signed with the private key associated with the address placing the order. This ensures that a user cannot submit a trade on an address they do not control. Before attempting to match an order, the engine will also verify that the user has granted NEX's smart contract enough of the asset such that the order can successfully be executed. If the user has not authorized enough funds, the order will be rejected.

To modify or cancel an order, a user must similarly submit a JSON request signed with the correct private key. An order will then be canceled or modified if it has not already been matched. If an order has been partially matched, then only the unmatched portion of the order will be affected.

Table 2: NEX initial fee structure

User 30 days volume	Taker fee	Maker fee
0%	0.25%	0%
1%	0.22%	0%
2.5%	0.19%	0%
5%	0.19%	0%
10%	0.16%	0%
20%	0.13%	0%

3.6 Fee Structure

NEX follows the maker/taker fee structure common to other exchanges. Market *makers* who place new limit orders on the order books will pay no fee, while *takers* who place an order at market place, or a limit order below the current market price will pay a small fee (Table 2). Fees will be deducted

[2] this order type is on our roadmap, to be supported some time after the initial launch

from the taker in the token denomination of their trade. NEX computes a user's 30 days moving average volume using the volume of trades associated with their public key, as a percentage of total exchange volume.

3.7 Implementation

The NEX off-chain matching engine will be built in Elixir, a functional programming language designed to build scalable, distributed, and fault-tolerant applications. Elixir builds on top of Erlang, a language originally intended for development of telecommunication systems, which is now used by modern web developers to manage the challenges of dealing with high availability. Elixir will help NEX realize an off-chain matching engine that provides service to users from all over the world, while functioning continuously and without downtime.

3.8 Smart Contract for Token Exchange

The NEX matching engine communicates with a smart contract that commits trades between users. This smart contract contains logic powered by the NEP-5 token standard, which allows it hold to user tokens involved in active trades. Once the matching engine computes a match, it sends this smart contract all involved user addresses and the types and amounts of tokens to trade between them, and the contract completes the trade. Calls to this smart contract can be batched in a single invocation transaction to increase performance and reduce network volume.

3.8.1 Trade Method Signature

The NEX exchange smart contract (SC) accepts two parameters: a *string* indicating the operation to be performed and a *bytearray* containing serialized data for usage in the method. The output of any call will be returned as a *bytearray*, with the first byte indicating the success or failure of the call and any resulting data serialized in the remainder of the *bytearray*.

The central interface between the off-chain matching engine and the blockchain will be the *trade* method of the exchange SC. This method will take the parameters *currency_maker*, *currency_taker*, *amount_maker*, *amount_taker*, *address_maker*, and *address_taker*. With these data the exchange SC delegates the trade of each currency to a corresponding SC via the NEP5 *transferFrom* method. In the case that the transfer of currency from the maker to taker or vice versa fails, any non-failed transfers will be reversed and the method will return *false*, otherwise the method will return *true*.

3.8.2 Security

To ensure that no third-party can execute a fraudulent trade between users, the *trade* method of the exchange smart contract will only accept orders signed by a private key held by the matching engine.

3.8.3 Withdrawal

The NEX smart contract only has access to funds involved in active trades, and not the full balance of tokens at a user's address. To retrieve tokens held for an active trade, a user can submit a cancel order to the NEX matching engine, which will then submit an immediate withdraw order to the exchange SC, transferring tokens back to the user. Delegating this request through the matching engine ensures that the engine will not match an order that is no longer backed by user funds.

However, to enhance user trust in NEX, the smart contract also supports a slower direct withdrawal that requires no communication with the off-chain matching engine. This second withdrawal process ensures that (1) users can withdraw active trade funds in the event of a broken or compromised matching engine (2) the matching engine has enough time to notice and cancel orders invalidated through any direct withdrawal of funds. To withdraw directly, a user first calls a public *withdrawal* method on the exchange SC, specifying the token type and amount. Ten blocks later, the user can then call a *complete_withdrawal* method, and the tokens will be transfered.

Users can always retrieve funds immediately by submitting an order cancellation to the matching engine, so we expect that the slower direct withdrawal method will not often be used. It exists simply to prove that users control any funds involved in active trades.

3.9 Payment Service

The NEX payment service allows NEO smart contracts to interact with assets that live outside of the NEO virtual machine. For example, a user might use the payment service to make transactions across chains, sending ETH to a NEO smart contract that then distributes it among their friends' NEO addresses. In the future, the payment service will support assets on other blockchains or non-digital assets such as USD, serving as a gateway for the trade of off-chain assets on NEX. In the near term, however, the payment service is designed to provide a similar function for global assets on the NEO blockchain such as NEO and GAS. More broadly, the payment service provides a starting point for reasoning about how a decentralized exchange can interact with assets on other chains.

3.9.1 Motivation for NEO and GAS

While it might seem surprising that NEO smart contracts cannot send NEO and GAS directly through computation, there are good reasons for this involving network scalability (see Section 2.3.1). As a result, the NEX payment service is necessary for complex interactions—such as decentralized exchange—between smart contracts and these global assets.

3.9.2 Interacting with the Payment Service

At a high level, the payment service converts global assets such as NEO and GAS into token assets that can be more easily sent and received by smart contracts (Figure 2). When you deposit NEO into the payment service, an equivalent amount of XNEO is created for you, which can be sent and received by smart contracts using the NEP-5 standard. For example, when you then withdraw your NEO from the payment service, your balance of XNEO is adjusted down accordingly. Any address that receives XNEO can withdraw the equivalent amount of NEO from the payment service.

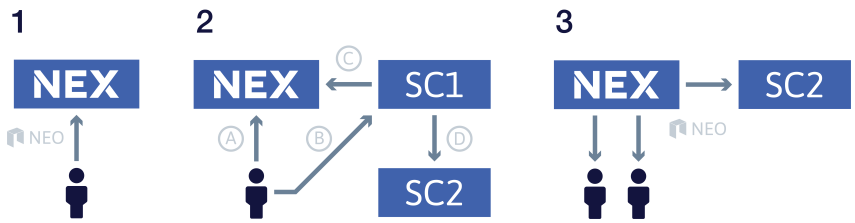


Figure 2: NEX's payment service layer allows NEO smart contracts to interact with UTXO based global assets such as NEO and GAS. Above, a user deposits NEO in the NEX payment service smart contract, creating a balance of XNEO that can be transferred between smart contracts (1). The user authorizes a third-party smart contract to access their XNEO (2a). The user then calls that contract (2b), which sends some of the XNEO to another user and authorizes a second smart contract to use the rest (2c). The first smart contract then calls the second smart contract (2d). Finally, the users and second smart contract withdraw NEO from the payment service, zeroing their XNEO balances (3).

3.9.3 Network-Assisted Global Asset Withdrawal procedure

The NEX payment service is powered by a novel withdrawal procedure we have developed for global assets on the NEO blockchain. While smart contracts are not able to send NEO or GAS to a user address directly, they are able to execute logic that decides whether a user is authorized to withdraw a certain amount of these assets through a normal contract transaction. We use this idea to allow users to withdraw funds from the NEX payment service.

Concretely, to withdraw assets from the payment service, a user calls the *withdrawal* method on the service SC, specifying an unspent TXID, asset type, and the amount to withdraw. The smart contract checks this information, and if a user is cleared for withdrawal, adds the TXID, output address, and amount to a white list in VM storage. This white list is consulted in any attempted transfer of funds from the smart contract. The user can then withdraw the appropriate amount from the smart contract using a normal contract transaction on the network.

By default, this kind of withdrawal is a *two-step* process. In the first step, a user registers a TXID and amount in the system for withdrawal, and in the second step, they perform a contract transaction to

execute the withdrawal. However, it is possible to make withdrawals a *one-step* process from the perspective of an end-user by incentivizing third parties to perform the second step of the process, allowing them to take a small gas fee. In this scenario, a user would authorize a withdrawal with the SC, and a number of bots monitoring public events on the chain would compete to execute the contract transaction in return for the fee.

3.9.4 Generalization to Assets on Other Blockchains

In the future, the concepts applied to NEO and GAS on the NEX payment service can also be applied to assets on other chains. One additional challenge when dealing with off-chain assets, however, is that a NEO smart contract cannot observe them directly. To solve this, we propose to create hubs that monitor events that occur on one chain, such as a user depositing funds in an address controlled by NEX, and push them to a smart contract on another. Beyond this important difference, similar principals hold: a NEO smart contract can create virtual NEP-5 representations of an asset under NEX control, allowing users to, for example, send Ethereum to a NEO address or smart contract. The owner of the receiving address or smart contract can then withdraw the funds on the outside chain following a procedure through which they prove ownership of the NEO address in question.

4 Decentralized Banking

Beyond enabling decentralized exchange, NEX offers a long term vision of decentralized banking: a smart contract based funds management service for assets on the blockchain. As with traditional banking, this management service would be designed to provide both fund security and investment opportunities (Figure 3). For example, this funds management smart contract can provide advanced security features through a dual key system consisting of a frequent use key and a large funds key. A user moving funds in NEX with the frequent key would be limited to a daily amount, and to trade above this threshold would need to provide their large funds key. If a frequent use key is compromised the user can lock their account using both keys and transfer the funds to a new account. Following the success of core exchange features, we envision this smart contract will integrate with and manage other NEX services, such as peer-to-peer lending or indexed investment accounts.

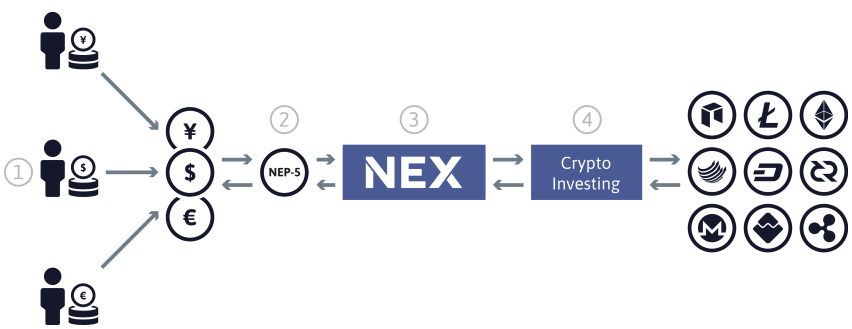


Figure 3: NEX enables a long term vision of decentralized banking through a smart contract based funds management service. (1) Users buy assets from conventional services, (2) then interact with them on the NEO blockchain through the NEX smart contract. (3) These assets can also be traded on NEX through NEP-5 pairs, to enable cross-chain exchange. (4) Funds stored in the management contract have access to other services such as indexed investment accounts or peer-to-peer lending.

5 NEX Token

The NEX token allows holders to claim a share of fees generated by the payment service and exchange. In total, 50 million tokens will be issued that entitle holders to a share of the fees taken by the exchange and payment service. NEX holders can claim their profits through a staking process, where claims on the staked NEX operate similar to GAS claim calculations on the NEO network. In this way, token holders who stake NEX benefit directly from the success of the exchange services: as more fees are generated, holders will receive larger rewards.

Please note: legal and regulatory policy may require changes in this token model. We aim to be as transparent as possible with the NEX community, and will share any updates as they occur.

5.1 Calculating Fees

Fees are calculated in terms of each asset traded or transfered on NEX. For example, if a user places a market price order trading 1000 NEX for NEO, then the exchange will collect a fee of $1000 * 0.0025 = 2.5$ NEX. Similarly, if a user transfers 1000 NEO on the payment service for a fee of 0.001 GAS, that fee will be added to the GAS fee total. Total NEX fees are calculated by simply computing the fees taken for each asset on the exchange. As fees are taken, a proportion of them are moved to an independent smart contract that manages the claiming process.

5.2 Claiming Fees via Staking NEX Tokens

Users can stake their NEX tokens in a smart contract that pays out a proportion of exchange and payment service fees. To stake their tokens, users send their NEX tokens to the smart contract via a *stake* method that records the starting block and the amount sent by the user. The user can then make periodic claims on the contract to retrieve their share of NEX profits since staking began. Users can commit to staking their tokens for longer periods of time to receive a larger proportion of fees.

5.2.1 Claim Example

A user owns 1000 NEX, and NEX has generated fees in tokens equivalent to 100 million dollars at market value since they last made a claim. Assuming the user has staked NEX at a rate of 75%, they would be eligible for a claim worth $\$100,000,000 * \frac{1000}{50,000,000} * 0.75 = \$1500$. This rate is hypothetical, and we will release true rates at a later time. The claim can be received:

- The user claims a direct cut of fees across each token on the exchange, so if NEX is trading NEO, GAS, NEX, and RPX, the user would receive a share of each of these assets.
- The user claims an equivalent amount in one preferred asset type. Here NEX will do the conversion automatically using its trade features and corresponding fee structure.

5.3 Token Sale

NEX will hold a token sale in Q1 2018. We plan to sell 25 million tokens to the public, of a total pool of 50 million. We will announce more details for the sale as soon as they have been worked out with our legal advisors. We aim to be maximally compliant with regulators, and will attempt to include as many countries as possible in the sale.

6 Current Progress and Roadmap

Here we describe the current state of development on NEX.

6.1 Incorporation

NEX will be incorporated in Zug, Switzerland. We are currently cooperating closely with Swiss financial authorities to ensure compliance in order to protect investors.

6.2 Technology

We have implemented alpha prototypes of all the smart contracts described in this paper, and will release these publicly in Q4 2017. We have also begun work on prototypes for the trading interface and matching engine. We strongly believe in a transparent development process where possible, and will be open sourcing much of our early work to benefit the NEO community. See our Github account for updates and more information: `https://github.com/neonexchange`.

6.3 Roadmap

NEX plans to have trading of NEO tokens operational in mid-2018. We propose the following as a preliminary release schedule:

- **Q4 2017**: Smart contract examples for payment service and decentralized exchange, demo of profit distribution mechanism on TestNet.
- **Q1 2018**: NEX token sale and release of open source platform for token sales on NEO.
- **Q2 2018**: Payment service launch on MainNet for NEO and GAS. Also, matching engine launch on TestNet, supported by accompanying CLI.
- **Q2 2018**: Smart wallet and API for integration with tethered tokens in MainNet.
- **Q3 2018**: Trading interface and matching engine launch on MainNet: begin trading NEO, GAS, and NEX. Also, cross-chain demo on ETH and NEO TestNets.
- **Q4 2018**: Cross-chain launch to support trading of ETH and ETH tokens. Also, support for margin trading on MainNet.
- **2019+**: Decentralized banking: smart contract asset management across chains.

References

[1] Coinmarketcap.com. http://coinmarketcap.com, 2017.

[2] Etherdelta. https://etherdelta.com/, Accessed 2017.

[3] EtherOpt. https://github.com/etheropt, Accessed 2017.

[4] Maker Market. https://oasisdex.com, Accessed 2017.

[5] NEO White Paper: Superconducting Exchange. https://github.com/neo-project/docs/blob/e01d268426a8b5f9b3676cfd03d0b8b83d7711a1/en-us/white-paper.md#highly-scalable-architecture-design, Accessed 2017.

[6] Raiden Network. https://raiden.network, Accessed 2017.

[7] Why we are building Cardano. https://whycardano.com, Accessed 2017.

[8] Bitcoin Developer Guide: Proof of Work. https://bitcoin.org/en/developer-guide#proof-of-work, Accessed 2017, Archived at https://www.webcitation.org/6v7DUj9JJ on November 20th, 2017.

[9] Bitcoin Developer Guide: Simplified Payment Verification. https://bitcoin.org/en/developer-guide#simplified-payment-verification-spv, Accessed 2017, Archived at https://www.webcitation.org/6v7DUj9JJ on November 20th, 2017.

[10] Bitcoin Developer Guide: UTXO. https://bitcoin.org/en/developer-guide#term-utxo, Accessed 2017, Archived at https://www.webcitation.org/6v7DUj9JJ on November 20th, 2017.

[11] The Cost of Decentralization in 0x and EtherDelta. http://hackingdistributed.com/2017/08/13/cost-of-decent/, Accessed 2017, Archived at http://www.webcitation.org/6v7Ff8r7D on November 20th, 2017.

[12] Ethereum Wiki: Design Rationale. https://github.com/ethereum/wiki/wiki/Design-Rationale#accounts-and-not-utxos, Accessed 2017, Archived at http://www.webcitation.org/6v7FswqI2 on November 20th, 2017.

[13] NEO NEP-5: Token Standard. https://github.com/neo-project/proposals/blob/master/nep-5.mediawiki, Accessed 2017, Archived at http://www.webcitation.org/6v7FuuPv2 on November 20th, 2017.

[14] Reconstructing Smart Contracts, Part II: Scalability. https://themerkle.com/reconstructing-smart-contracts-part-ii-parallel-universes-and-unlimited-scalability/, Accessed 2017, Archived at http://www.webcitation.org/6v7FxFbHA on November 20th, 2017.

[15] Ethereum Wiki: Proof of Stake. https://github.com/ethereum/wiki/wiki/Proof-of-Stake-FAQ, Accessed 2017, Archived at http://www.webcitation.org/6v7G0YAQH on November 20th, 2017.

[16] CME Matching Algorithm: FIFO. https://www.cmegroup.com/confluence/display/EPICSANDBOX/Matching+Algorithms#MatchingAlgorithms-FIFO, Accessed 2017, Archived at http://www.webcitation.org/6v7GArrCz on November 20th, 2017.

[17] High Frequency Trading on the Coinbase Exchange. https://www.coindesk.com/high-frequency-trading-on-the-coinbase-exchange/, Accessed 2017, Archived at http://www.webcitation.org/6v7GHNIhG on November 20th, 2017.

[18] Loopring White Paper. https://github.com/Loopring/whitepaper/blob/master/en_whitepaper.pdf, Accessed 2017, Archived at http://www.webcitation.org/6v7GNu8z7 on November 20th, 2017.

[19] Castro, M., Liskov, B., et al. Practical byzantine fault tolerance. In *OSDI*, vol. 99 (1999), 173–186.

[20] Coindesk. The Bitfinex Bitcoin Hack: What We Know (And Don't Know). https://www.coindesk.com/bitfinex-bitcoin-hack-know-dont-know/, 2016, Archived at http://www.webcitation.org/6v7FW2mc9 on November 20th, 2017.

[21] Coleman, J. State Channels. http://www.jeffcoleman.ca/state-channels/, Accessed 2017, Archived at http://www.webcitation.org/6v7Fi0GbQ on November 20th, 2017.

[22] Hertzog, E., Benartzi, G., and Benartzi, G. Bancor protocol: A hierarchical monetary system and the foundation of a global decentralized autonomous exchange, 2017.

[23] Othman, A., Pennock, D. M., Reeves, D. M., and Sandholm, T. A practical liquidity-sensitive automated market maker. *ACM Transactions on Economics and Computation 1*, 3 (2013), 14.

[24] Swan, M. *Blockchain: Blueprint for a new economy*. " O'Reilly Media, Inc.", 2015.

[25] Vukolić, M. The quest for scalable blockchain fabric: Proof-of-work vs. bft replication. In *International Workshop on Open Problems in Network Security*, Springer (2015), 112–125.

[26] Warren, W., and Bandeali, A. 0x: An open protocol for decentralized exchange on the ethereum blockchain, 2017.

[27] Wired Magazine. The Inside Story of Mt. Gox, Bitcoin's $460 Million Disaster. https://www.wired.com/2014/03/bitcoin-exchange/, 2014, Archived at http://www.webcitation.org/6v7FULQZa on November 20th, 2017.

[28] Wood, G. Ethereum: A secure decentralised generalised transaction ledger. *Ethereum Project Yellow Paper 151* (2014).

[29] Zhang, E., and Hongfei, D. NEO White Paper. http://docs.neo.org/en-us/index.html, Accessed 2017, Archived at http://www.webcitation.org/6v7FpGOHZ on November 20th, 2017.