NEX: A High Performance Decentralized Trade and Payment Platform

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. Unfortunately, today's DEXs are slow, hard to use, and are restricted to trades on a single blockchain. Neon Exchange (NEX) is a new decentralized exchange that solves these problems, leveraging a high performance off-chain matching engine built with Elixir to handle massive order volume, allow cross-chain exchange, and support more complex trading APIs. Many people do not use DEXs because they cannot trade with their national currencies. NEX is the first DEX that will allow its clients to enter the exchange with fiat currencies such as USD, powered by a global network of licensed third parties.

Exchange services are critical not just for users of cryptocurrencies, but also for many other applications built upon blockchain technology. NEX envisions a vibrant ecosystem of web-based decentralized applications (dApps) running across public blockchains such as NEO, Bitcoin, and Ethereum. These applications require tools to interact with, make transactions upon, and send data across chains. To achieve this vision, NEX is developing the first cross-chain browser extension. Web applications can use this extension to collect payments for goods in any digital currency, make trades and transactions across blockchains, and interact with smart contracts across NEO and Ethereum. All of this cross-chain functionality is powered and made possible by the NEX off-chain matching engine.

NEX is issuing 50 million tokens to fund development and future expansion, of which 25 million will be sold to the public. NEX tokens are in the process of being registered as a security (first in Europe), and will provide holders with a share of all revenue generated by the exchange and related services. Revenue per user will be determined by the number of NEX tokens they hold and the length of time they commit to staking their tokens. The minimum fee share rate is 25%, increasing linearly to a maximum of 75% when tokens are staked for two years.

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

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 [2]. 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 [29, 21].

Recently, decentralized exchanges have emerged to allow users to trade without giving up control of their funds [28, 3]. 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].

Decentralized exchanges also tend to have problems trading cryptocurrencies across chains or against national currencies. This is because smart contracts operating on one chain have no means of reasoning about transactions on another. While mechanisms such as atomic swaps have been proposed to allow a DEX to trade assets between independent chains (for example, trading Bitcoin for Ethereum), none of these mechanisms have achieved widespread use [18]. Today, users who want to trade currencies across chains or against national currencies must use centralized exchanges.

If centralized exchanges provide speed and flexibility, and decentralized exchanges provide security, then it is natural to ask: can a hybrid system provide the best of both worlds? In this paper, we propose that the optimal mix of these properties is provided by *a decentralized exchange with an off-chain matching engine*. Order matching is by far the most computationally expensive operation when running an exchange. By encapsulating this component in an off-chain service, we can reap enormous improvements in speed, and also support complex trades such as limit or market orders. Similarly, an off-chain engine can act as a coordinator of transactions across chains, enabling a straightforward approach to cross-chain exchange. By committing orders on-chain as they are matched—with provable deterministic behavior—we can also retain the security benefits of traditional DEXs.

Exchange is a key component and enabler of a broader ecosystem of decentralized applications. By creating the first high performance API for decentralized cross-chain exchange, NEX enables many new possibilities for and interactions with such applications. For example, if a website takes payment in GAS and a user holds only ETH in their wallet, NEX will allow the user to convert some portion of their holdings and send it to the website, directly from their existing address. Further, this interaction can occur seamlessly in a user's browser. The user simply clicks a button on the website: this opens a pre-populated transaction window that will make the necessary conversion then send the transaction. To support such an ecosystem, we have developed a cross-chain browser extension that allows websites to communicate with user accounts and the NEX matching engine.

Neon Exchange (NEX) is a new decentralized exchange that embodies these ideas. This white paper presents our vision for the NEX platform, the performance benefits of our technical approach, and how NEX will shape the broader cryptocurrency 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 [26]. 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 [30]. 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 [5, 4]. 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 [23, 25]. 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 [22, 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 [28, 3]. In these systems, market makers broadcast an order off-chain, which can then be picked up by an interested counter party 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. 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 Cross-chain Exchange

Cross-chain exchange is the process of trading or transferring assets that exist on two different blockchains (for example, trading NEO for ETH). This kind of exchange is trivial when two parties trust one another or an independent third party (e.g., a centralized exchange). However, exchange becomes much more difficult in the general case, where trust is absent. As a simple example, if one person sends ETH to another in expectation that the other will send them back NEO, the second party may simply keep the ETH transferred by the first without fulfilling the exchange.

Atomic swaps offer one approach to facilitating cross-chain trades without trusting a third party [18]. These swaps make use of hash time-locked contracts (HTLC), a multisignature transaction scheme that requires cooperation from both parties for a trade to be successful. While this approach has seen several successful implementations [19], it is in practice a slow and cumbersome process that is not suitable for a high volume exchange. Atomic swaps require potential traders to first find each other to coordinate the swap, then engage in multiple transactions on two different blockchains.

Other projects have proposed cross-chain relay as a more scalable approach to decentralized cross-chain exchange [24]. In this approach, smart contracts on one chain are given the ability to verify transactions on another. For example, NEO smart contracts might be provided with a stream of block headers that would allow them to verify transactions on Bitcoin. Given such a setup, a smart contract could tell when a transaction has occurred on another chain and so unlock funds on its own chain. Unfortunately, this approach requires trust in an oracle that provides information about the external chain and cannot be fully implemented on chains without smart contracts, such as Bitcoin.

In contrast, we introduce a new approach that simplifies cross-chain transactions though a small trade off in trust. Because the NEX matching engine is off-chain, it can easily act as a coordinator and intermediary between many separate chains. For example, if user A wants to trade 100 NEO to user B in return for 10 ETH, the matching engine can delegate fulfillment between smart contracts on both NEO and Ethereum: on NEO, one smart contract will move 100 NEO from A's address to B's address; while simultaneously on Ethereum, a similar smart contract will move 10 ETH from B's address to A's address. As with all trades matched by the engine, NEX provides a public record of orders and a deterministic specification of matching behavior, so users can verify the correctness of a cross-chain match and claim a bounty in the event to incorrect behavior.

2.4 The NEO Blockchain

NEO was launched in 2015 as China's first public blockchain [31]. NEX will run first on NEO, before later expanding to support exchange on Ethereum and other blockchains. 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.

2.4.1 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 *token representation smart contract*, 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 smart contract address, then later withdraw them from (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].

2.4.2 Consensus

NEO and Ethereum also operate under very different consensus models. NEO uses delegated Byzantine Fault Tolerance (dBFT) for consensus [20, 27], whereas Ethereum uses Proof of Work (PoW) [8]. NEO's consensus model allows for higher theoretical transaction throughput, which has a positive impact on the performance of a decentralized exchange. NEO consensus also provides *finality*: once a block has been generated by consensus, it is not possible to have competing version of the chain within the network. 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 [14].

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.

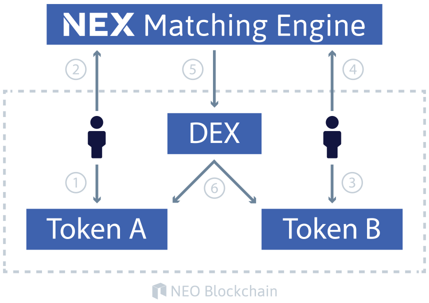


Figure 1: The NEX architecture provides fast, decentralized exchange using an off-chain matching engine. Here we illustrate an example user interaction with NEX exchange. First, one user authorizes a trade to exchange Token A for Token B (1) and sends the order to the matching engine (2). Next, a second 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 [17].

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 [15]. 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 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 National Currencies

One major problem with today's decentralized exchanges is lack of support for national currencies such as USD. NEX is solving this project through banking partnerships with companies around the world. Through this network of partners, users will be able to buy cryptocurrencies directly using national currencies to store in their wallets and trade on the exchange.

3.5 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

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.

3.6 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 [16].

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.

3.7 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 from the taker in the token denomination of their trade. NEX computes a user's 30 days moving

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%

average volume using the volume of trades associated with their public key, as a percentage of total exchange volume.

3.8 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.

3.9 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.

3.9.1 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*.

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.

3.9.2 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.

3.9.3 On-chain Settlement

Even with batched trades and a blockchain with high TPS, on-chain settlement will occasionally be too slow to keep up with the off-chain matching engine. To combat this problem, we adopt a multi-user state channel approach. The matching engine keeps track of user account balances in memory as these balances are updated via trades that come in over the API. Periodically, these balances will be committed back to the chain in settlement.

However, because funds committed back to the chain may need to be reused for new trades that come in over the API, it is not sufficient to simply settle these balances directly back to a user's account. For example, suppose a bot trades some Ethereum for NEO. In a naive approach to settlement, as soon as the NEO is settled in the bot's account, the bot would have to reauthorize the funds to the matching engine to commit a new trade, which would require a least one new block to be created on the chain before the funds would be available again. This would prevent the bot from quickly reacting to new market conditions and, for example, trading the NEO back to Ethereum at a higher price.

To solve this problem, we have created a smart contract vault that can be accessed by both the bot and the matching engine for future trading. So, when a trade is committed, a bot can elect to have the funds move back to the vault instead of its own address. There, the funds can be traded again immediately by the matching engine with no delay, even before the funds have been committed to the vault. At any future time the bot can remove the funds from the vault with several blocks of delay to warn the matching engine and avoid committing impossible trades. These vaults can be created on both Ethereum and NEO to support trades between any combination of NEP5 and ERC20 tokens.

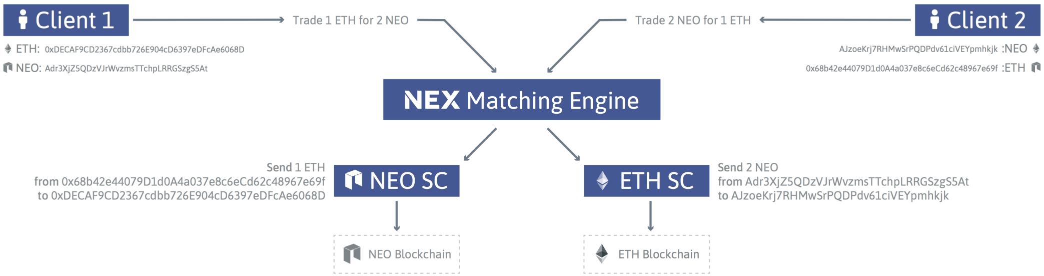

Client 1
ETH: 0xDECAF9CD2367cdbb726E904cD6397eDFcAe6068D
NEO: Adr3XjZ5QDzVJrWvzmsTTchpLRRGSzgS5At
Trade 1 ETH for 2 NEO
Trade 2 NEO for 1 ETH
Client 2
AJzoeKrj7RHMwSrPQDPdv61ciVEYpmhkjk :NEO
0x68b42e44079D1d0A4a037e8c6eCd62c48967e69f :ETH
NEX Matching Engine
Send 1 ETH
from 0x68b42e44079D1d0A4a037e8c6eCd62c48967e69f
to 0xDECAF9CD2367cdbb726E904cD6397eDFcAe6068D
NEO SC
ETH SC
Send 2 NEO
from Adr3XjZ5QDzVJrWvzmsTTchpLRRGSzgS5At
to AJzoeKrj7RHMwSrPQDPdv61ciVEYpmhkjk
NEO Blockchain
ETH Blockchain

Figure 2: Cross-chain exchange on NEO and Ethereum.

3.10 Cross-chain Exchange

To enable cross-chain exchange, the matching engine coordinates settlement across two chains simultaneously following the procedure outlined above. To do this, the engine simply tells smart contracts on each chain where to move the funds that have been allocated for trading (Figure 2). The matching engine is off-chain and so can communicate with smart contracts on any number of chains. Further, every NEX user will own an address on each chain operated on by the exchange.

Example: Suppose user A wants to trade 100 NEO for 5 ETH, and user B wants to trade 5 ETH for 100 NEO. Each user places a limit order on NEX, moving their funds into smart contract vaults on NEO and Ethereum respectively. When the matching engine matches the trade, it sends a signal to the vaults on each chain. On Ethereum the smart contract moves 5 ETH into A's address, while on NEO the smart contract moves 100 NEO into B's address.

3.11 Withdrawal for UTXO-based Assets

While smart contracts are not able to directly transfer UTXO-based assets such as NEO or GAS to a user address, 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 token smart contract.

Concretely, to withdraw UTXO-based assets 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.

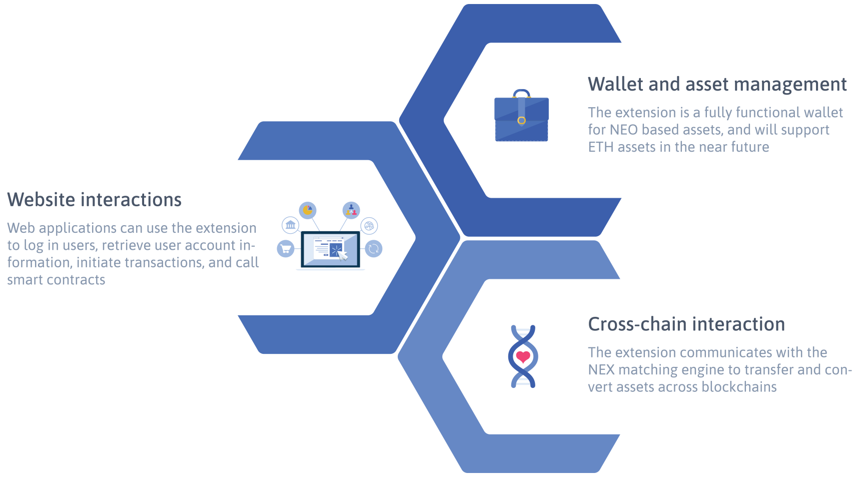

Wallet and asset management
The extension is a fully functional wallet for NEO based assets, and will support ETH assets in the near future
Website interactions
Web applications can use the extension to log in users, retrieve user account information, initiate transactions, and call smart contracts
Cross-chain interaction
The extension communicates with the NEX matching engine to transfer and convert assets across blockchains

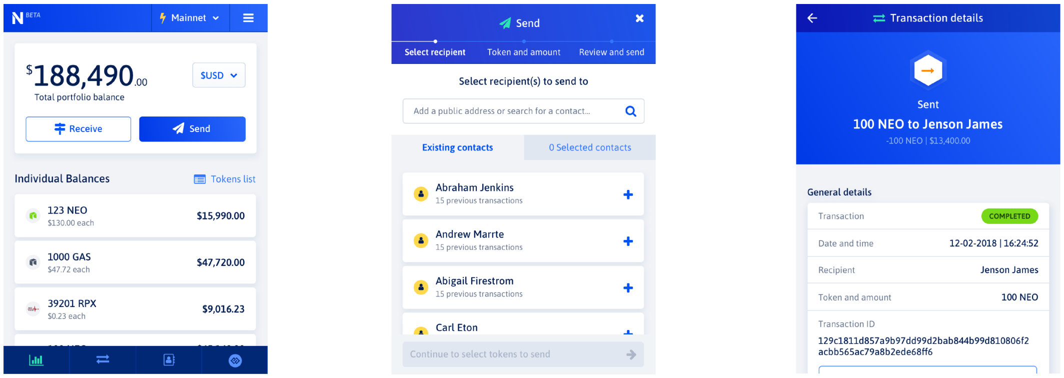
Figure 3: The NEX web extension allows browsers to communicate with and across blockchains.

3.12 NEX Web Extension

Exchange is a key component of user interactions with an ecosystem of decentralized applications. For example, users may want to make payments to a website using blockchain based assets, convert assets on demand to meet the needs of a specific service, or interact with smart contracts on one blockchain using assets from another. To make possible these interactions and many others, we are developing the NEX web extension: the first browser extension to manage assets across multiple chains, allowing web based dApps to interact with assets through a consistent API (Figures 3, 4).

3.12.1 Asset Management

The NEX web extension provides users with a full featured cryptocurrency wallet with the ability to view asset balances and transaction history, and send and receive assets. The wallet will initially support NEO and NEP5 tokens, before expanding to ETH and ERC20 tokens. In addition to these standard features, the extension will allow users to monitor ongoing trades on the NEX platform and convert balances between cryptocurrencies. The extension will also allow users to purchase cryptocurrencies with national currencies (e.g., USD) from partnering banks.



$188,490.00
Total portfolio balance

Individual Balances

123 NEO $130.00 each	$15,990.00
1000 GAS $47.72 each	$47,720.00
39201 RPX $0.23 each	$9,016.23

Select recipient(s) to send to

Abraham Jenkins
15 previous transactions

Andrew Marrte
15 previous transactions

Abigail Firestrom
15 previous transactions

Carl Eton

Sent
100 NEO to Jenson James
-100 NEO | $13,400.00

General details

Transaction	COMPLETED
Date and time	12-02-2018 | 16:24:52
Recipient	Jenson James
Token and amount	100 NEO

Transaction ID
129c1811d857a9b97dd99d2bab844b99d810806f2
acbb565ac79a8b2ede68ff6

Figure 4: The NEX web extension provides asset management software that can interact with webpages within a user's browser.

3.12.2 Extension Web API

By building tools for blockchain and cross-chain interaction into a single web extension, NEX will allow decentralized websites to interact more seamlessly with users through several APIs:

User Identity: Websites can leverage the web extension to pull and populate relevant user information into a page (e.g., a user's NEO address, or the address of a contact). For security reasons, this information will only be available after a user has specifically approved a domain to access it.

Transaction Initiation: Websites can initiate transactions through API calls to the web extension, resulting in a new extension window pre-filled with the relevant transaction data. This is useful, for example, in shopping websites: allowing for one-click purchases after user approval of a transaction.

Smart Contract Invocation: Some websites may integrate information from smart contracts. This API allows such websites to initiate a call to a smart contract in a new transaction window.

3.12.3 Security

Security is vital when building any asset management platform. The NEX extension has undergone a full security audit by the Cure53 team [1]. The extension also integrates security features to prevent spoofing, such as a secret color and code word presented to the user in each popup window. Beyond adding new functionality to a user's browser, the NEX extension actively protects users from known scammers by forcing a redirect when it encounters a known malicious website.

4 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. NEX tokens will be regulated as registered European securities with plans for expansion to other jurisdictions.

4.1 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.

4.2 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. The base rate of fee share will be 25%, if the user stakes their NEX tokens for one day, increasing linearly up to 75% if a user is willing to stake for two years.

4.2.1 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 the two year staking rate of 75%, they would be eligible for a claim worth $\$100,000,000 * \frac{1000}{50,000,000} * 0.75 = \$1500$. 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.

4.3 Token Sale

NEX will hold a token sale in April of 2018. We plan to sell 25 million tokens to the public, of a total pool of 50 million. As payment methods we accept NEO and GAS only. More details are available on our website: `https://neonexchange.org`.

5 Progress and Roadmap

Here we describe the current state of development on NEX.

5.1 Development

We have completed and deployed smart contract prototypes for trading and withdrawal features on the exchange. Development of the matching engine has begun and a full prototype will be available in early 2018. We have also released an alpha version of the NEX web extension, which will connect with the exchange for asset conversion, and begun work on the trading user interface. We have created and shared template code with our banking partners that will allow them to interface with the NEX extension to buy cryptocurrencies and send them to a user's account.

5.2 Partners

We'd like to thank our partners for their continuous support in legal matters: Brown Rudnick LLP, Nägele Rechtsanwälte GmbH, Ernst & Young AG, IdentityMind Global, GN Treuhand Establishment, and one undisclosed partner.

5.3 Incorporation

NEX is a limited company registered in Vaduz, Liechtenstein ("Neon Exchange AG"). We are currently cooperating with financial authorities in pursuit of acquiring an Organized Trading Facility (OTF) license.

5.4 Roadmap

Q1 2018:

- Launch of web extension for dApp integration with NEO blockchain and NEX exchange

- Registration, lottery, and KYC for NEX token sale
- Release of an open source framework for conducting such sales, including smart contracts, wallet and extension integration, KYC, and backend website
- Open source framework and template for interaction with NEX banking partner APIs
- Announcement of partnerships

Q2 2018:

- Participation for lottery winners in the NEX token sale
- Matching engine launch on NEO TestNet, along with v1 of exchange APIs for testing with our liquidity creation partners (e.g., trading bots)
- Trading interface launch on NEO TestNet: usability research and information gathering from alpha testers
- NEX web extension support for ETH and ERC20 tokens
- Buy and sell NEO, GAS, and RPX through banking partners
- Release of v1 of web extension dApp APIs, for integration on client websites and smart contracts on ETH and NEO (e.g., APIs that allow websites to interact with smart contracts, or request payments in NEO or ETH or constituent tokens from a user)

Q3 2018:

- Beginning of trading operations: Matching engine and trading user interface launched on MainNet with support for NEO, ETH, NEP5, and ERC20 tokens
- Cross-chain token conversion support in NEX extension wallet
- Launch of staking contract on NEO for holders of NEX to receive exchange revenue

Q4 2018:

- Once required licenses are granted, NEX and other security token trading starts on NEX
- Advanced trading features
- Cross-chain support for trading BTC, LTC, and RPX on NEX

References

[1] Cure53. https://cure53.de.

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

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

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

[5] Maker Market. https://oasisdex.com, 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] Ethereum Wiki: Proof of Stake. https://github.com/ethereum/wiki/wiki/Proof-of-Stake-FAQ, Accessed 2017, Archived at http://www.webcitation.org/6v7GOYAQH on November 20th, 2017.

[15] 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.

[16] 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.

[17] 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.

[18] Atomic swaps on the lightning network. https://bitcoinmagazine.com/articles/atomic-swaps-how-the-lightning-network-extends-to-altcoins-1484157052/, Accessed 2018, Archived at http://www.webcitation.org/6yEeo5Vak on March 27th, 2018.

[19] Lightning network. https://lightning.network, Accessed 2018, Archived at http://www.webcitation.org/6yEeqsbkP on March 27th, 2018.

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

[21] 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.

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

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

[24] Luu, L. Chain relays for cross-chain trades. https://blog.kyber.network/chain-relays-or-a-practical-approach-for-cross-chain-trades-d0d7003f266b, Accessed 2018, Archived at http://www.webcitation.org/6v7Ff8r7D on March 27th, 2018.

[25] 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.

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

[27] 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.

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

[29] 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.

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

[31] Zhang, E., and Hongfei, D. NEO White Paper. http://docs.neo.org/en-us/index.html, Accessed 2017, Archived at http://www.webcitation.org/6v7FpG0HZ on November 20th, 2017.