WerkCoin

Business Platform and Decentralized Exchange

Werkcoin Team, werkcoin@gmail.cz

April 1, 2017

Abstract

WerkCoin - WER is a new type of decentralized exchange, uses liquidity mechanism of business companies and provides supported and add-on blockchain gateway functions.

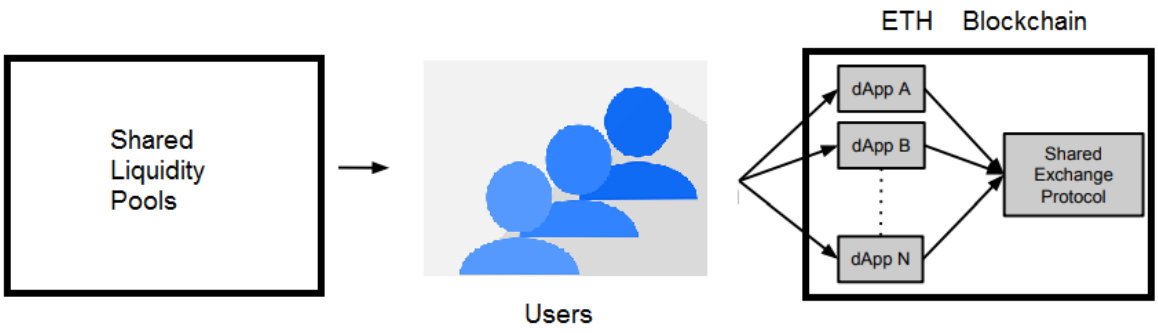
1. Introduction

Our blockchain platform uses the protocol token mechanism (ERC-20 protocol token style and native cryptocurrencies). His purpose is create a blockchain agreement with a shared certificate that allows market to maintain market activity among participants.

The blockchain platform protocol facilitates low friction peer-to-peer exchange of ERC20 tokens on the Ethereum blockchain. The protocol is intended to serve as an open standard and common building block, driving interoperability among decentralized applications (dApps) that incorporate exchange functionality.

Trades are executed by a system of Ethereum smart contracts .DApps built on top of the protocol can access public liquidity pools or create their own liquidity pool and charge transaction fees on the resulting volume. The protocol is fair and it extracts same value from one group of users to benefit another.

Decentralized governance is used to continuouslyand securely integrate updates into the base protocol without disrupting dApps or end users.



2. Role of WerkCoin platform

The primary role of our blockchain platform are to solve coordination problems among multilateral agreements between a network of participants. By ensuring transparency, assurance, and enforcement, we can enable multilateral agreements where they were not previously possible.

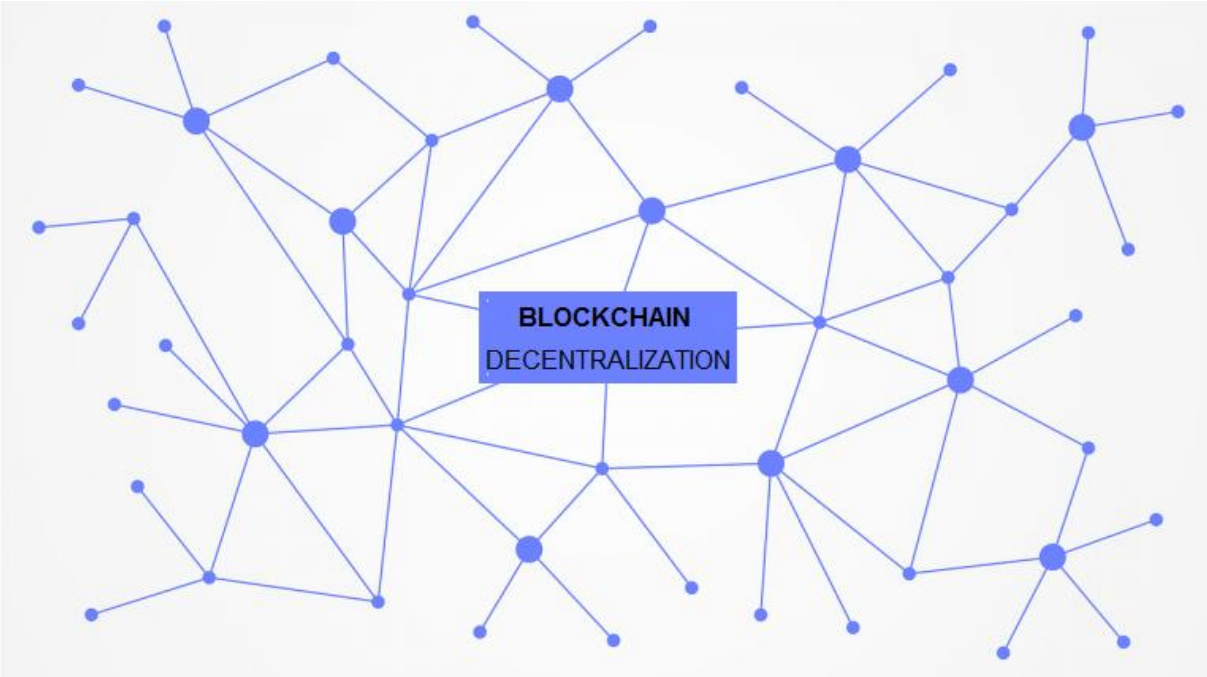
When all parties are assured that the operations are not only transparent, but also the mechanisms are guaranteed to not change without significant effort, parties are more willing to coordinate their businees process. All participants have the same guarantees.

Business processes now is much easier. In other words, any single participant is more willing to use systems where the business processes and mechanisms itself are not owned by any other single participant.

Blockchains have been revolutionary by allowing anyone to own and transfer assets across an open financial network without the need for a trusted third party. Now that there are hundreds [of blockchainbased assets, and more being added every month, the need to exchange these assets is compounding.

With the advent of smart contracts, it is possible for two or more parties to exchange blockchain assets without the need for a trusted third party. Decentralized exchange is an important progression from the ecosystem of centralized exchanges for a several key reasons. Decentralized exchanges can provide stronger security guarantees to end users since there is no longer a central party which can be hacked, run away with customer funds or be subjected to government regulations. Hacks of Mt. Gox, Shapeshift, Bitfinex, Thether and Binance have demonstrated that these types of systemic risks are palpable.

Decentralized exchange will eliminate these risks by allowing users to transact trustlessly - without a middleman - and by placing the burden of security onto individual users rather than onto a single custodian.

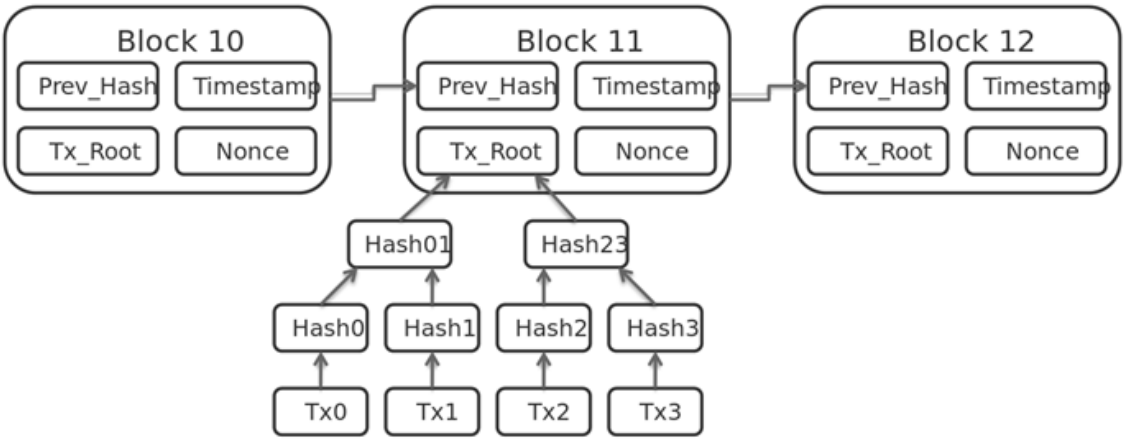


3. Blockchain overview

A blockchain, originally block chain is a growing list of records, called blocks, that are linked using cryptography. Each block contains a cryptographic hash of the previous block a timestamp, and transaction data (generally represented as a Merkle tree).

By design, a blockchain is resistant to modification of the data. It is "an open, distributed ledger that can record transactions between two parties efficiently and in a verifiable and permanent way". For use as a distributed ledger, a blockchain is typically managed by a peer-to-peer network collectively adhering to a protocol for inter-node communication and validating new blocks.

Once recorded, the data in any given block cannot be altered retroactively without alteration of all subsequent blocks, which requires consensus of the network majority. Although blockchain records are not unalterable, blockchains may be considered secure by design and exemplify a distributed computing system with high Byzantine fault tolerance. Decentralized consensus has therefore been claimed with a blockchain.



4.Specification

Main state channels are proposed as a means of scaling the Ethereum blockchain and reducing costs for a variety of applications - including exchange - by moving transactions off of the blockchain.

Participants in a state channel pass cryptographically signed messages back and forth, accumulating intermediate state changes without publishing them to the canonical chain until the channel is closed.
It follows that channel participants must always be online to challenge a dishonest counterparty and the participants are therefore vulnerable to DDOS attacks.

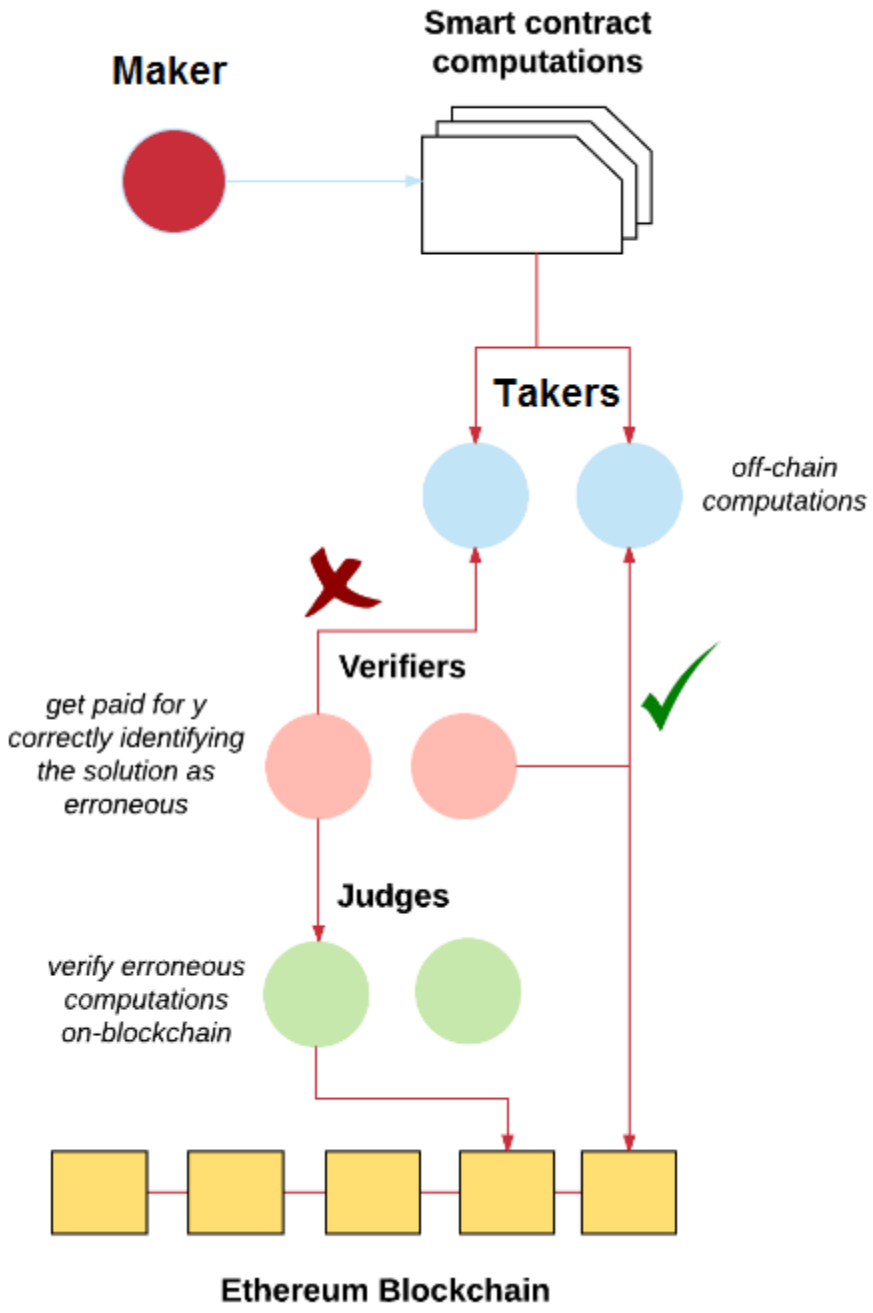
While state channels drastically reduce the number of on-chain transactions for specific use cases, the numerous on-chain transactions and security deposit required to open and safely close a state channel make them inefficient for one-time transactions.

A hybrid implementation, which we refer to as “off-chain order relay with on-chain settlement,” combines the efficiency of state channels with the near instant settlement of on-chain order books.

In this approach, cryptographically signed orders are broadcast off of the blockchain; an interested counterparty may inject one or more of these orders into a smart contract to execute trades trustlessly, directly on the blockchain. Friction costs are minimized for market makers because they can signal intent offchain and transactions only occur when value is being transferred. We extend this approach by allowing anyone to act as the exchange and by making the protocol application.

4.1 Off-chain order

Maker approves the decentralized exchange (DEX) contract to access their balance of Token A.



Maker creates an order to exchange Token A for Token B, specifying a desired exchange rate, expiration time (beyond which the order cannot be filled), and signs the order with their private key.

Maker broadcasts the order over any arbitrary communication medium.

Taker intercepts the order and decides that they would like to fill it, approves the DEX contract to access their balance of Token B. Taker submits the makers signed order to the DEX contract. The DEX contract authenticates makers signature, verifies that the order has not expired, verifies that the order has not already been filled, then transfers tokens between the two parties at the specified exchange rate.

4.2 Message Format

Each order is a data packet containing order parameters and an associated signature. Order parameters are concatenated and hashed to 32 bytes via the SHA3 Keccak function. The order originator signs the order hash with their private key to produce an ECDSA signature.

Point-to-point orders allow two parties to directly exchange tokens between each other using just about any communication medium they prefer to relay messages. The packet of data that makes up the order is a few hundred bytes of hex that may be sent through email, a Facebook message, whisper or any similar service. The order can only be filled by the specified taker address, rendering the order useless for eavesdroppers.

4.3 Smart Contract

The exchange protocol is implemented within an Ethereum smart contract the entire contract need ETH to fill an order. ETH smart contract may be used as a proxy for ERC20 ether and includes a proposal to change ether to follow the ERC20 token standard.

4.3.1 Signature Authentication

The exchange smart contract is able to authenticate the order originator's (Maker's) signature using the ecrecover function, which takes a hash and a signature of the hash as arguments and returns the public key that produced the signature. If the public key returned by ecrecover is equal to the maker address, the signature is authentic.

```
address publicKey = ecrecover( hash, signature( hash) );
if ( publicKey != maker ) throw;
```

4.3.2 Fills & Partial Fills

The exchange smart contract stores a reference to each previously filled order to prevent a single order from being filled multiple times. These references are stored within a mapping; a data structure that, in this case, maps a 32-byte chunk of data to a 256-bit unsigned integer.

Passing the parameters associated with an order into the Keccak SHA3 function produces a unique 32-byte hash that may be used to uniquely identify that order (the odds of a hash collision, finding two different orders with an identical hash, are practically zero). Each time an order is filled, the mapping stores the order hash and the cumulative value filled.

A Taker may partially fill an order by specifying an additional argument, valueFill, when calling the exchange smart contract's fill function. Multiple partial fills may be executed on a single order so long as the sum of the partial fills does not exceed the total value of the order.

4.3.3 Expiration Time

An order's expiration time is specified by the Maker at the time the order is signed. The expiration time is an unsigned integer value that represents the absolute number of seconds since the unix epoch. This value cannot be changed once it has been signed.

Time within the Ethereum virtual machine is given by block timestamps that are set each time a new block is mined. Therefore, the expiration status of an order does not depend upon the time at which a Taker broadcasts their intention to fill an order, instead it depends upon the time at which the fill function is being executed in the EVM by a miner. A miner cannot set the block timestamp of the current block to be earlier than the timestamp of the previous block.

4.3.4 Cancelling Orders

An unfilled and unexpired order may be cancelled by the associated Maker via the exchange smart contract's cancel function. The cancel function maps an order's hash to the order's maximum value (valueA), preventing subsequent fills. Cancelling an order costs gas and, therefore, the cancel function is only intended to serve as a fallback mechanism. Typically, Makers are expected to avoid on-chain transactions by setting their order expiration times to match the frequency with which they intend to update their orders.

One issue with this approach is that it can create situations where a Maker attempts to cancel their order at roughly the same time a Taker is attempting to fill that same order. One of the two parties

transactions will fail, wasting gas, depending upon the sequence in which the two transactions are mined. Uncertainty regarding the sequence in which transactions are mined could lead to undesirable outcomes at times. This uncertainty could increase if the Ethereum blockchain were to experience a significant backlog of pending transactions.

5. Summary

Blockchains allows society to externalize the world's business processes from single centralized corporations into open, decentralized computing networks. Wercoin is a network which decentralizes market liquidity, orderbook matching and execution, clearinghouse custodianship, and high-scalability payments to help resolve payments across these emerging called eWallet payment networks.

By shifting these business processes traditionally placed into a single corporation, it is possible to provide eWallet providers an entire interchange process in a decentralized high-performant open network.

The endstate requirement is a construction of a decentralized mechanism for eWallet platforms holding fiat-backed value. The eWallet fiat tokens will have the ability to use Ether on the decentralized, public Ethereum chain (or any other decentralized cryptocurrency) as the interchange/intermediary cross for maximum efficiency. We believe that this allows for significant more activity and value in decentralized cryptocurrencies, as it will serve as a useful venue for many eWallet business platforms.

The mechanism must also allow to trade these assets/commodities. In order to perform interchange, it requires an order to be placed across many different pairs on an open public market. This requires a decentralized orderbook and trading engine.

The trading engine is built into the WER blockchain, orders are published and matches are performed as part of every block when a matched order has reached sufficient number of validation

confirmations. This results in a non-custodial decentralized exchange held by a single party where the eWallet platforms may exchange onto other eWallet platforms without centralized trust on a single entity.

However, direct crosses between eWallet fiat tokens may not be desirable, as there may be too many. It would be necessary to use cryptocurrency for a liquid market without

single preference. By bonding Ethereum into a smart contract (or Bitcoin-like tokens into bonded clearinghouses), it is possible to lock up Ether onto the activity of the WER chain to allow for eWallet pairs to occur over Ether or other cryptocurrencies, creating a liquid market (if every pair cross es with ETH, spreads would be much smaller provided low currency volatility). For activity requiring very small spreads, it may emerge that some eWallet tokens will be used as interchange crossing; however, there's strong incentive
to use decentralized tokens for settlement due to coordination/trust advantages rel ate d to programmatic adjudication. eWallet fiat tokens may also cross using other eWallet tokens if necessary, but bonding which don't affect short-term exchange rate fluctuations of smart contract activity will be primarily in ETH (e.g. HTLC clearinghouse, liquidity providing, and WER chain enforcement). By allowing for cryptocurrencies to be the backing for eWallet platforms, the platforms can be assured of an even playing field between eWallet interchange activities.

This requires a greater degree of liquidity in funds locked up, and the Wercoin decentralized exchange may not be desirable to transact for low-value interchange activity (e.g.for high- volume micropayments). Not every payment between two distinct eWallets must be performed using a trade on the decentralized exchange.

There is an expectation, that eWallets will hold some reserve of fiat tokens of other eWallets, ready to be used for smaller transfer s in popular directions. And it can solve the problem with transaction for low-value interchange activity.
Constructions such as Lightning Network allow for payments to occur off- chain when eWallets hold balances to facilitate rapid payments.

Decentralized Liquidity hub for channels: The construction has the additional benefit of allowing for a decentralized liquidity pool to

be created for use with payment channels on various cryptocurrencies, such as Bitcoin (and to some extent Ethereum).

For individual token payments on blockchains, there is a need to scale the underlying blockchain activity which does not affect the underlying chain to reduce computational pressure of validating/mining nodes. It is therefore necessary to conduct

Lightning Network activities (or similar constructions using channels). However, Lightning Network faces significant pressure around network effects with capital, it's desirable to prevent liquidity pools from centralizing to a singletrusted entity. By using the same mechanisms of the decentralized clearinghouse, we can create a

Lightning Network hub which is not owned by any single individualon tokens which support more complex smart contracts (e.g. Ethereum, ERC-20 like tokens, etc.). For currencies with simple smart contracts, any node on the network (e.g. Bitcoin network) can act as a gateway into the WER chain pool and crossback with any other participant. This allows the Wercoin chain to offload a lot of on-chain activity, while encouraging decentralization.

We believe that the natural network effects of liquidity centralization can be mitigated by decentralized stake-chains with deterministic/known consensus rules.

For Ethereum in particular (and other full-feature d smart contract scripting blockchains), all participants setup channels into an ETH smart contract operating as a single pool of funds. The chain state of the WER chain reflects the current balance of participants. This allows for any participant to supply liquidity onto this networkwhich can be allocated in accordance to the Wer-chain consensus rules (limits may be inplace early on to prevent this blockchain from sucking up all the spare liquidity from the cryptocurrency space if this construction is

successful before robust testing/validation overtime). These funds can thereby be used for any liquidity activity on the WER chain.

The above mechanisms require significant volume of activity (with a large amount of state) and is not at this time suitable for all activity to occur off the Ethereum mainchain, however we working on the project Giga for construction to bond trading activity with contract execution which will be provide only by the WER chain.

We are building a blockchain which hooks into other blockchains to allow for trading across token/asset classes, largely backed by ether . From the perspective of any individual chain, we are building a scalable blockchain whose contract state is bonded by the activities of the WER chain itself. Activity on other chains can interlink with this chain viainterchain committed proofs similar (but constructed differently) to BTC on the Werchain which can be submitted on Ethereum.

Apendix

ERC20 Token ERC20 establishes a standard contract ABI for tokens on the Ethereum blockchain and has become representation for all types of digital assets. ERC20 tokens share the same contract interface, simplifying integration with external contracts.

Core ERC20 functions include:

• transfer (to, value)

• balance of (owner)

• approve (spender, value)

• allowance (owner, spender)

• transfer from (from, to, value)

References:

[1] Coinmarketcap. https://coinmarketcap.com/all/views/all/

[2] Wikipedia: https://en.wikipedia.org/wiki/blockchain/

[3] Coindesk: Hacking Incident. https://coindesk /blog/Mt.Gox/

[4] Bitcoinist: https://bitcoinist.com/ Bitfinex/Thether/

[5] Coinmarketcap: https://coinmarketcap.com/altcoins/views/all/

[6] EtherOpt. https://etheropt.github.io/

[7] Google. https://google.com/decentralize/ crypto/token/coin/

[8] Tokens. https://www.reddit.com/tokens/erc20ú

[9] Photos https://www.pixabay.com/

[10] BBC http://www.bbc.com/new/blog/binance /

[11] IDEX, Decentralized Capital. http://www.idex.market/