Contents

VDex White Paper v0.1.2

The Volentix Labs Team
info@volentixlabs.com

September 4, 2018

Abstract

There is consensus among cryptocurrency users that the handling of currency should be simplified and enhanced. Because of the complexity of decentalized systems, users and investors have had little recourse securing enough knowledge to properly quantify their investments. The interfaces offered by tokenized environments and their dApps will promote better user implication by presenting the information in more ergonomic, familiar and pedagogical ways. As a response to this, VDex is a decentralized exchange with the user and community in mind. Using some of the most recent paradigms and established protocols for security, ease of use and multi asset support, this low friction peer-to-peer exchange abides by open standards and ensures a harmonious and seamless flow among decentralized applications. Through the use of Verto, a versatile and highly customizable crypto wallet, VDex provides easy to use options for security, anonymity and speed of payment. Open order books support integration with other decentralized exchanges, in effect producing a massive decentralized exchange of exchanges, contributing to the liquidity and effectiveness of all the connected exchanges. VDex is also a pillar in the Volentix ecosystem, a network of DApps whose synergy promotes greater liquidity for users while simplifying and enriching user experience, providing comfort for users, and maximizing security of user transactions.

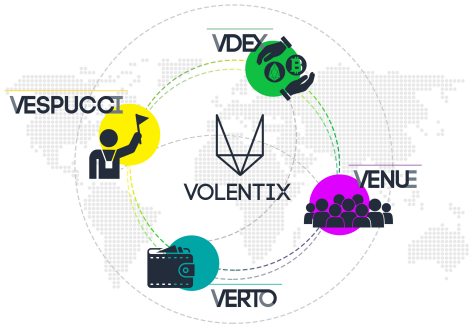

Figure 1:

1 Introduction

VDex is a distributed exchange that provides a highly customizable environment for speed, cost, anonymity, security, and scalability. Power users are enabled with the freedom to choose and thrive, while new users feel welcomed and free from the risks inherent in a centralized system. The growth of the product resides in a flexible architecture able to adopt the best practices of 2nd generation blockchain applications. The task of building VDex mainly resides in successfully merging and integrating today's best protocols, paradigms and patterns to match the Volentix requirements on top of the 3rd generation blockchain, EOS.IO decentralized operating system.

2 Volentix

The Volentix ecosystem which consists of DApps which improve the effectiveness of each other. The "four pillars" are an initial set of DApps which support the entire Volentix ecosystem, each in a way specific to their own needs. From the perspective of VDex, the purpose of Volentix DApps is to grow the user base of VDex and thus increase the liquidity available to all users.

- **Venue**
  Grows the Volentix community
- **Verto**
  Enables funds to be continually maintained by the user while using VDex
- **Vespucci**
  Increases trust in the tokens available on VDex
- **VDex**
  Provides liquidity between all crytocurrencies

2.1 Venue dynamic community platform

Venue is planned as a dynamic community platform that recruits and aligns members of the Volentix community to facilitate distribution of VTX, the native token of Volentix, and to promote awareness of Volentix initiatives. Recently launched in beta testing, Venue enables users to earn VTX tokens in exchange for participating in developing dedicated communities, submitting bug fixes, and claiming bounties. Leaderboards and live metrics reflect user participation. The first signature campaign was launched on the https://bitcointalk.org/ forum on July 13, 2018. Please visit https://venue.volentix.io for more information.

2.2 Verto wallet

Verto is a multi-currency wallet for use with the VDex exchange, and intends to facilitate custody of private keys for use in peer-to-peer transactions. Both private and public keys will be locally managed, with the goal of eliminating the risks of devastating losses of stake associated with failures of central operators. Verto plans to employ a system of smart contracts to maintain the state between two trading clients, the simplest operations being accomplished with atomic swaps.[15]

2.3 Vespucci analytical engine

Vespucci is an analytics engine accessible via a user-friendly interface with real-time market data such as cryptocurrency ratings and sentiment analyses, empowering users with tools to graph and compare tradeable digital assets, to access and parse historical trading records, to plot trends and patterns, and to monitor and assess open-source software developments. Vespucci seeks to make available comprehensive market-relevant data by aggregating the information currently scattered throughout many different blockchains, websites, chat rooms, and exchanges.

2.4 VDex

The fourth pillar of Volentix, the VDex exchange, is the tradable digital assets platform introduced in detail in this white paper. VDex provides crypto currency exchange services directly from the user's Verto wallet where both public and private keys are locally managed. VDex supports trading in many cryptocurrencies and hosts multiple liquidity pools to accomodate different types of markets. VDex distinguishes itself with a unique order book decentralization system multiple liquidity pools and extensive security measures and user awareness mechanisms with regards to trading in these pools/networks.

2.5 VTX

VTX is an enabling currency that flows through the pillars to enables them to interact.

3 Architecture

3.0.1 Overview

Cross-chain, inter-wallet transactions are done by providing wallets with the contracts or scripts to transact in any cryptocurrency. The transaction between Bob and Alice involves each sending funds toward the other's accounts using these provided contracts best described in the original atomic swap paper.[15] These contracts have guarantees toward each other by the means of shared secrets within timeouts or a refund occurs. The various means of ensuring collateral for a particular transaction or augmenting liquidity of a network will be described later in this document.

3.0.2 Operating system

EOS.IO is an operating system-like framework upon which decentralized applications can be built. The software provides accounts, authentication, databases, asynchronous communication and scheduling across clusters. Components and protocols are already built into the platform, and just a subset can be used to satisfy VDex requirements. VDex initially benefits from the standard features offered by EOS.IO such as account and wallet creation and the recovery of stolen keys, but will subsequently implement the protocols for the creation of decentralized exchanges through its contracts and tools [10]
Here is a summary of features and functionality :

1. **Context Free Actions**
   A Context Free Action involves computations that depend only on transaction data, but not upon the blockchain state. ex: Parallel processed signature verifications
   Most of the scalability techniques proposed by Ethereum (Sharding, Raiden, Plasma, State Channels) become much more efficient, parallelizable, and practical while also ensuring speedy inter-blockchain communication and unlimited scalability.

2. **Binary/JSON conversion**
   EOS contracts combine the human readability of JSON with the efficiency of binary.

3. **Parallelisation and optimisation**
   Separating authentification from application allows faster transaction times and increases bandwidth. EOS.IO blocks are produced every 500 ms.

4. **Web Assembly(WASM)**
   Web Assembly enables high-performance Web applications and also secures each application in its own sandbox, through which functionalities VDex can regulate network access, filesystem namespace restrictions, enforced rule-based execution and application spawning.

5. **Rust/C++ contracts**
   At the time of this writing, C++ has best tooling for and execution speeds for WASM. C++ is a much more mature language than for instance Solidity used for Ethereum contracts. It also has better debugging support as well as libraries that have been tested over the years and provide reliable functionality. The EOS codebase has also very heavy usage of templates. For example, C++ allows the use of templates and operator overloading to define a runtime cost-free validation of units. Managing memory is much easier building smart contracts because the program reinitializes to clean state at the start of every message. Furthermore, there is rarely a need to implement dynamic memory allocation. The WebAssembly framework automatically rejects any transaction addressing memory inaccurately. Since EOS.IO contracts use C++14 one can resort to smart pointers if dynamic memory allocation is needed. It is noteworthy that the first implementation of PARSEC Directed Acyclic Graph (DAG) technology is expected to be in Rust.[20]

6. **Schema defined messages and database**
   Service contracts are standardized to guarantee a baseline measure of interoperability associated with the harmonization of data models. The *Standardized Service Contract* design principle advocates that service contracts be based on standardized data models. Analysis is done on the service inventory blueprint to find out the commonly occurring business documents that are exchanged between services. These business documents are then modeled in a standardized manner. The Canonical Schema pattern reduces the need for application of the data model transformation design pattern. [9]

3.0.3 Inter Contract Communication

Data is shared between contracts via an oracle which creates a transaction embedding the data in the chain. "An oracle, in the context of blockchains and smart contracts, is an agent that finds and verifies real-world occurrences and submits this information to a blockchain to be used by smart contracts." [3] Every node has an identical copy of this data, so it can be safely used in a smart contract computation. Oracles push the data onto the blockchain rather than the smart contract pulling the information. Most reading of the data is done via polling **nodeos** (the blockchain instance) to monitors the blockchain's state and performs certain responsive actions.

3.0.4 Side Chains

In EOS.IO, issuing tokens is a process which creates a sidechain. Sidechains are emerging mechanisms that allow tokens and other digital assets from one blockchain to be securely used in a separate blockchain and then be moved back to the original blockchain if needed. There can be multiple side chains where different tasks are distributed accordingly for improving the efficiency of processing. For inter-blockchain communication, the EOS.IO protocol creates a TCP like communication channel between chains to evaluate proofs. For each shard (a unit of parallelizable execution in a cycle), a balanced merkle tree is constructed of these action commitments to generate a temporary shared merkle root; this is done for speed of parallel computation. The block header contains the root of a balanced merkle tree whose leaves are the roots of these individual sharded merkle trees. [10]

3.0.5 Liquidity

I A cryptocurrency token is liquid if it is easily sold or purchased in ordinary trading volumes without a significant short-term impact on its prevailing market price. In order to achieve such a status, traditionally any tradable asset must surmount a trading volume threshold sufficient to support stability. In order for a token to effectively partake in the global token economy, its trading volume must cross a critical barrier where the matches between buyers and sellers become frequent enough to ensure a stable pool of "coincidences of wants" [11]. Specifically, the following protocols and methodologies for managing liquidity are to be implented:

1. Implementation of the Loopring protocol with the use of EOS.IO contracts acting as nodes.[25]
2. Implementation of the Bancor algorithm used to bring stability to the token.[11]
3. Toggles between these protocols and atomic swaps(HTLC) according to the Vespucci analysis on the VDex network.

3.0.6 Hashed timelock contracts (Atomic Swaps)

A Hashed Timelock Contract (HTLC)[15] is a type of smart contract enabling the implementation of time-bound transactions. Users are offered a variable lock-in period for their transations, with a discount on the transaction fee in exchange for choosing a slightly greater lock-in period.

3.1 Network Topology

3.1.1 Nodes

Nodes are the endpoints of the VDex network. Their function is to:

1. Act as a portal to VDex trough the Verto wallet.
2. Merge order book information with others.
3. Settle order book.
4. Manage order cancellation.
5. Assign timeouts for the **Raft** Protocol.
6. Initiate contract for orders that have been filled.

Nodes earn a portion of the fee for each transaction. If a user has sufficient funds and posessing a good track record, their Verto wallet can act as a node.

3.1.2 Aggregators

The VDex aggregators are dedicated Volentix servers for simulator and security purposes. One of their functions is to pull logs and order book data from nodes into sparse distributed representations for hierarchical temporal memory as intrusion [16] analysis for detecting anomalies in the system such. The aggregators also are host to other components such as metachain ledgers[4] blockchain scrapers and Vespucci components.

3.1.3 Latency

EOS.IO has low latency block confirmation (0.5 seconds).[10] This latency can be provided if the currencies being traded are issued from blockchains that are equally fast, otherwise the transaction is as fast as the slowest block chain(for example a bitcoin block takes 9 minutes to mine at time of this writing). On completion of the transaction, a transaction receipt is generated. Receiving a transaction hash does not mean that the transaction has been confirmed; it means only that a node accepted it without error, although there is also a high probability other producers will accept it.

3.2 Order Book

An order book is a list of orders that VDex uses to record the interest of buyers and sellers. VDex's matching engine uses this book to determine which orders can be fulfilled. Order books can be tailored according to cost, security and speed according to which protocol is used to settle the book to cater to these requirements. For instance the Loopring protocol already allows for its orderbook to be modified in order to work with other orderbooks.[25] The same can be done for other non restrictive but efficient protocols such as PARSEC, RAFT or Bancor which each offer their own very distinct advantages. Loopring for instance has very good front-running protection steps by checking for sub-rings in its ringbuffer(FIFO) data structure, while the Bancor formula can be used to ensure coin stability or create liquidity. On the other hand, PARSEC or RAFT offer the simplest and most efficient decentralization solutions research has yet to offer. The ability to modify order book settlement methods on the fly according to rules based on models built by oracles residing on aggregators provides another precautionary measure to ensure the homeostatis of VDex. With this in mind, containers provided by EOS.IO provide best performance.[17] and using these containers to unify the order book morphology is optimal.

3.2.1 Example Data structures

Using the Loopring Protocol FIFO (first-in first-out) circular buffer, nodes can design their order books to display and match a user's order. This method follows an OTC model, where limit orders are positioned based on price only. [25] Referencing the EOS.IO persistence API, the order book is able to take advantage of the powerful multi-index container shared among nodes through the same EOS.IO account.

3.2.2 On-Chain order book

An on-chain order book is a record of offers residing on the wallet (node) chosen to settle the order book. It resides in a persistent database on each node subscribing to the same account as all the other nodes.

3.2.3 Off-Chain order book

On the aggregator, offline order books serve for simulator and security purposes. Built within the period between two order book settlements, this books is gathering logs from as many nodes as possible, building its own versionof the orderbook. At the moment a node settles the on-chain order book, both order books should be identical.

3.2.4 Decentalization process of order book settlement

For decentralization purposes, nodes will take turns to settle the order book. The settling node must be designated by the protocol and all order book entries from all nodes must be available to the settling nodes. The RAFT[7] and PARSEC[20] protocols offer elegant and simple solutions. RAFT is a well-established algorithm and is easy to implement. PARSEC is more recent and more efficient, using Directed Acyclic Graph (DAG) technology and eliminating the need for copying logs.

3.3 Order settlement

Order settlement contains familiar elements of conventional financial market transactions. Utilizing FIFO technology to design the order book, VDex intends to check order, inventory, and fill rate, as well as limit orders and cancellations.

3.4 VTX

VtX is the native cryptocurrency token to be used on the VDex exchange. A variant of the eosio.token contract from the EOS.IO framework will be used to issue 2.1 billion EOS.IO-compliant VtX tokens with a supply of 1.3 billion. As a utility token, VtX will have a diverse array of uses, for example:

1. To reward participants in the consensus process and in Venue campaigns.
2. To pay and redistribute transaction fees on the VDex exchange.
3. To submit and vote on proposals to the Volentix network, using the voting rights allocated to VTX holders.
4. To incentivize users to participate in order book settlement by becoming nodes via their Verto wallets.
5. To incentivize users to lock funds in for ¿24 hours by HTLC time-bound transactions.
   item To stake support for reviewing proposals and implementing projects.

VtX Distribution [under revision]
Distribution classes:

1. The founders
   the foundational promoters and designers of the Volentix project.
2. Prior work
   Core team, board of advisors, software engineers, blockchain analysts, legal experts, business development specialists, and the Volentix board of advisors.
3. Decentalized treasury
   Publicly controlled funds for community projects, business operations, and salaries for ongoing development. A small percentage of every transaction amount is expected to be earmarked to be returned to the treasury.
4. Ongoing core development
   Incentives for core team. These are bonuses given throughout the project after development milestones have been reached.

3.4.3 Planned VtX distribution [under revision]

1. 35% Decentralized Treasury
2. 5% Initial funding
3. 28% Distribution
4. 12% Founders
5. 10% Prior Work, Core Team and Board of Advisors
6. 10% Ongoing Core Development

3.4.4 VtX Crowdsale [under revision]
The Volentix founders, core team, and advisors are currently considering the question of whether or not to conduct a crowdsale and, if the answer is yes, then based on what terms and conditions.

EOS.IO The following considerations are applicable for deploying the exchange on the EOS.IO platform

1. Deploying a contract has a cost but is free to use.
2. Developers have to stake EOS.IO tokens to deploy contract. After the contract is destroyed, the locked tokens are returned.
3. Decentralized applications must allocate resources to their contracts, memory, cpu, bandwidth.
4. The choice of who pays the resources is up to the DApp.
5. Multiple messages in one transaction and multiple accounts can be asigned to the same thread.

3.5 Inter blockchain communication

EOS.IO is designed to make Inter-Blockchain-Communication proofs lightweight. For chains with an insufficient capacity for processing the IBC proofs and establishing validity, there is the option to degrade to trusted oracles/escrows. To directly control other currency transactions with an EOS.IO based smart contract, a trusted mutisig wallet holding the currency in escrow is used to persuade the signing/publishing of the currency transaction based on IBC proofs from the originating chain.

3.6 Security

3.6.1 Introduction

To shake out certain assumptions, we intend to commence security testing following the prototyping phase. Security concerns are of paramount importance to users and must be addressed. Threats include, for example, an attacker executing malicious code within a transaction or manipulating the order of transactions or the timestamps of blocks. In the following sections, we address certain security measures and specific security threats and remedies.

3.6.2 Contract security

1. Retain vast majority of funds in a time-delayed, multi-signature-controlled account.
2. Use multi-signatures on a hot wallet with several independent processes/servers double-checking all withdrawals, with the concomitant benefit of creating a trusted list of accounts.
3. Deploy a custom contract that allows withdrawals only to accounts verified by KYC/AML.
4. Deploy a custom contract that accepts only deposits of known tokens from accounts verified by KYC/AML.
5. Deploy a custom contract that requires a mandatory 24-hour waiting period for all withdrawals.
6. Utilize contracts with hardware wallets for all signing, including for automated withdrawals.
7. Upgrade broken contracts.
8. Include ability to pause the functionality of a contract.
9. Include ability to delay an action of a contract.

3.6.3 Malware detection by auditing processes

The system provides insights on rogue processes during the transaction period with AI analysis residing on the aggregators.

3.6.4 Random diversification

By using the RAFT protocol in the election process, a certain level of randomization is aquired with varying length of timeouts. The toggling of protocols is a level of complexity

3.6.5 Multiple factor identification

As in many existing applications, this measure is efficient and already known to the public at large.

3.6.6 Logs

Ensure inspection of logs as control.

1. Raft.
2. Anomaly detection with AI(Numenta).
3. Script investigations of certain non token purchases related addresses.

3.6.7 Transaction as Proof of Stake (TaPoS)

1. Prevents a replay of a transaction on forks that do not include the referenced block
2. Signals the network that a particular user and their stake are on a specific fork.

3.6.8 Double spend

A double spend is an attack where a given set of coins is spent in more than one transaction.

1. Send two conflicting transactions in rapid succession into the network. This is called a race attack.

2. Pre-mine one transaction into a block and spend the same tokens before releasing the block to invalidate that transaction. This is called a *Finney* attack.

3. Own 51+% of the total computing power of the network to reverse any transaction, as well as have total control of which transactions appear in blocks. This is called a 51% attack. This is impossible according to EOS.IO, Loopring or Raft. If a block producer takes an unreasonable amount of runtime or is not profitable enough, the process is blacklisted.[25]

3.6.9 Front running

To prevent someone from copying another node's trade solution and have it mined before the next supposed transaction in the pool, a higher fee per transaction is charged.
The major scheme of front-running in any protocol for order-matching is order-filch: when a front-runner steals one or more orders from a pending order book settlement transaction. EOS.IO and loopring both have remedies to this. In both cases keys are not part of the on-chain transaction and thus remain unknown to parties other than the order book settling node.

3.6.10 Forged identities

Malicious users create forged identities to send a large number of small orders to attack Loopring nodes. However, most of these orders will be rejected for not yielding satisfying profit when matched.

3.6.11 Insufficient Balance

Malicious users sign and spread orders the value of which is non-zero but the address of which has a zero balance. Nodes monitor actual balances, update these order states accordingly, and then discard them.

3.6.12 Timing attack

Timing attacks are a class of cryptographic attacks through which a third-party observer can deduce the content of encrypted data by recording and analyzing the time taken to execute cryptographic algorithms. The randomness of timeouts in the raft algorithm prevents this, for instance, at the orderbook level.

3.6.13 Other EOS.IO security attributes

1. No uses of mutex or locks for on-chain parallellisation

2. All accounts must only read and write in their own private database

3.7 Inter chain security

Transactions sent to a foreign chain require some facilities on the foreign chain to be trustless. In the case of two EOS.IO based chains, the foreign blockchain runs a smart contract which accepts block headers and incoming transactions from untrusted sources and is able to establish trust in the incoming transactions if they are provably from the originating chain. For chains with an insufficient capacity for processing the IBC proofs and establishing validity, the options degrade to trusted oracles/escrows.

3.8 Multi blockchain

Multi-blockchain information can be obtained by aggregating blockchain timelines in parallel order (with variance in the frequency of when the state is changed) into a comprehensible data structure. This enables a system to trigger multichain load balancers, transfer states, draw data outputs from smart contracts, and trigger execution of transactions on foreign blockchains. Relative block distance, relative global state, and timestamped events are recorded on a global ledger to optimize and confirm transactions before they actually happen on the native chain. This could be used to determine block production coincidence between chains to choose optimal liquidity.[4]

3.9 User experience

As highlighted by the Statement of Purpose, our focus on user experience is primary. We wish to make VtX and the four pillars of Volentix – Venue, Verto, Vespucci, and VDex – easily accessible to and useable by all those who wish to join the community. The user interface makes available relevant market data as well as account information. The experience of VDex shouldcontinually to be educational l, with templates and simulators to support a superior UX/UI relationship.

Comprehensive customizable and detailed interface

1. Shows the entire market and fluctuations

2. Shows wallet: balance and previous transactions.

3. Shows detailed history with built in tax calculator.

4. Contains toggles for advanced features.

3.10 True decentralization

EOS.IO is an open-source, scalable infrastructure for decentralized applications. Its goal is a fair and transparent block producer (BP) election process utilizing a democratic delegated proof of stake (DPoS) consensus. Particularly as such a system just begins to proliferate, there will be glitches. Therefore, some degree of retained centralization is inevitable and necessary. But our guiding philosophy is one of decentralization, and our ongoing efforts are targeted to promoting a reduction in dependence on central authority. For example, initially we plan to erect a system for electing nodes (when solving order books) that will not use a shared central clock or DPoS but instead will be based either on random timeouts for the determination of leaders in an election (RAFT) or on Directed Acyclic Graph (DAG) in the PARSEC protocol.

3.11 System recovery

The RAFT and PARSEC protocols provide a robust system for recovery in the case of node failure. Security measures are also provided for trading between and among native blockchains. If a chain defies identification, the system defaults to the next block or a short time lock.

3.12 Scalable and modular architecture

To secure the potential for innovation, the principles, concepts and paradigms proposed by components of the system must favour decoupling of technologies. Since creating and maintaining distibuted and decentralized systems is very complex we must use different strategies:

3.12.1 Problem decomposition

Problem solving strategy of breaking a problem up into a set of subproblems, solving each of the subproblems, and then composing a solution to the original problem from the subproblem solutions.

3.12.2 Minimize state space

Dynamic programming and templating are hard because of complexity and debugging challenges. Nesting conditions can also seem unimportant for normal program execution. Special attention to these patterns and details in the initial design will maximize efficiency while allowing for easy replacement or addition of components. In the context where CPU, bandwidth and RAM are monetized, this is even more important.

4 Contributions

The public software repository for Volentix is https://github.com/Volentix. All contributions and suggestions to these repositories will be reviewed and integrated.

5 Risk

"The risks associated with the cryptocurrency/blockchain space remain patent. It is still a nascent and controversial area. Substantial time and money can be committed, consumed, and ultimately come to nothing. It is irresponsible to devote capital or other resources, including your most precious asset – time – unless you can afford to expend those resources. Thousands of scam cryptocurrency coins have come and gone, and are yet to come, offering nothing but vaporware and fraudulent guarantees of huge returns on investment. It is small wonder that regulators speak with aggressive words.

"For those who would eschew outright gambling, and instead wish to consider carefully how to proceed prudently, however, the mandate is to become well educated. These are technical topics, to be sure, but one's most important armament is knowledge. There is a sound future for brilliant digital application use-case incursions into forming innovative economies and functionalities. In perhaps 5-10 years from now, all major businesses will adopt some form of private blockchain. All professionals in law, finance, medicine, engineering, and education will become schooled in digital applications and smart contracts. Ontological harmonization is inevitable as the diverse dialects of humans converge to become the standard global language of artificial intelligence as lightning-fast micro- and macro-transactions come to dominate existence."[26]

All of us at Volentix are dedicating sizable quantities of work and insights to developing a program premised on empowerment and independence. If you are of a mind to join us, in whatever capacity, then please do so only if you are willing to become educated on the topics contained in this white paper and additional Volentix publications as we share them with our community.

6 Aknowledgements

Thanks to professor Yiannis Emiris from the Department of Informatics and Telecommunications at the National and Kapodistrian University of Athens for comments on this document.

7 Conclusion

Constructs, concepts and protocols that stand out by their simplicity while retaining their effectiveness helps implement a modular design which enhances the system with the capacity to easily add or replace components with the prospects of carefully advancing functionality as micro services are created. Although certain assumptions made in this paper still remain to be verified, a very distinctive direction for VDex architecture has been distilled as a highly flexible and modular MVP capable of adaptation and reaction to the changing technological ecosystem. For this, provisions of the EOS.IO operating system, Loopring, Bancor, RAFT and PARSEC protocols have been retained.

Bibliography

References

[1] Aelf, *A multi-chain parallel computing blockchain framework*, (2018).

[2] ARK, *A platform for consumer adoption*, (2018).

[3] blockchainhub.net, *blockchain-oracles*, (2017).

[4] BlockColliderTeam, *Block collider white paper*, (2018).

[5] V. Buterin, *Ethereum: a next generation smart contract and decentralized application platform*, (2013).

[6] S. Cormier, *A machine based societal model for curbing citizen cynicism*, (2017).

[7] J. O. Diego Ongaro, *In search of an understandable consensus algorithm*, (2018).

[8] M. Duncan, Quale, *Halo platform*, (2018).

[9] T. Earl, *Soa principles of service design*, (2016).

[10] EOS.IO, *Eos.io technical white paper v2*, (2018).

[11] G. B. Eyal Hertzog, Guy Benartzi, *Bancor protocol: Continuous liquidity for cryptographic tokens through their smart contracts*, (2018).

[12] S. D. K. M. T. S. H. Garcia-Molina, *The eigentrust algorithm for reputation management in p2p networks*, (2018).

[13] M. R. Garrick Hileman, *Global cryptocurrency benchmarking study*, (2017).

[14] Komodo, *An advanced blockchain technology, focused on freedom*, (2018).

[15] K. Kurokawa, *Atomic cross chain transfer, an overview*, (2015).

[16] L. Lamport, *The part time parliament*, (1998).

[17] D. Larimer, *eosio.boot telegram chat*, (2018).

[18] Q. Liquid, *Providing liquidity to the non-liquid crypto economy*, (2018).

[19] S. R. M.P.M-S, Aniket Kate Matteo Maffei, *Concurrency and privacy with payment-channel networks*, (2017).

[20] F. H. Q. M. S. S. Pierre Chevalier, Bart lomiej Kami´nski, *Protocol for asynchronous, reliable, secure and efficient consensus (parsec)*, (2018).

[21] SingularityNET, *A decentralized, open market and inter-network for ais*, (2018).

[22] M. M. Timo Hanke and D. Williams, *Dfinity technology overview series consensus system*, (2018).

[23] A. B. Will Warren, *0x: An open protocol for decentralized exchange on the ethereum blockchain*, (2017).

[24] G. Wood, *Ethereum: A secure decentralised generalised transaction ledger.ethereum project yellow paper*, (2014).

[25] F. Zhou, Wang, *Loopring: A decentralized token exchange protocol*, (2018).

[26] D. K. Rutledge, Esq., *The Future of Cryptocurrency: The Vital Importance of User Experience and Knowledgel*, (2018).

[9] [3] [10] [5] [24] [13] [25] [23] [19] [11] [12] [14] [2] [8] [18] [21] [22] [7] [1] [4] [6] [15] [20] [16] [17] [26]