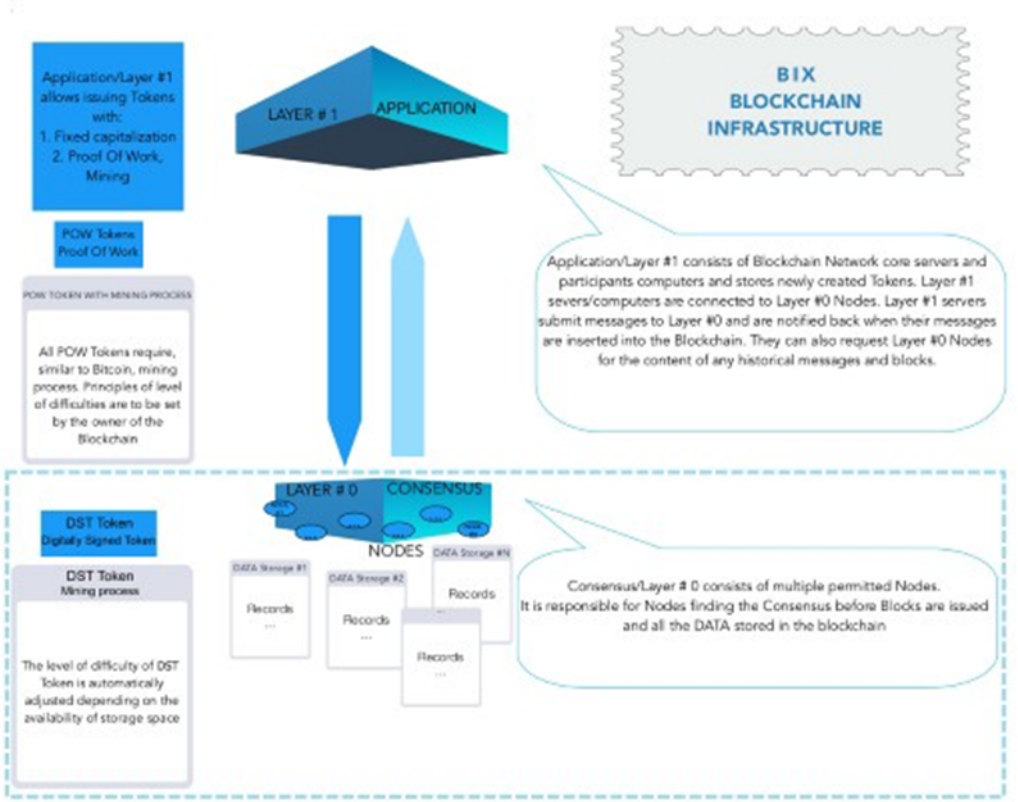
	BLOCKCHAIN SPECIFICATION	BIX Blockchain	Notes
1	**Permissioned Blockchain**	YES	Essentially distributed and privately owned by governments, authorities, agencies, associations, banking/ non-banking financial institutions, business entities, IoT, universities, colleges, schools and other organizations, BIX Blockchain is enabling its members, employees, students using BIX network for business, financial, payment, security, educational, gaming and other purposes.
2	**Public Blockchain**	YES (with privately permitted nodes)	Any AML/ KYC compliant individual, entity or group, permitted by the board of BIX Blockchain Network, can set up a server node(s) and participate in the consensus mechanism by operating servers having in their possessions and participating in validating transactions process of the network.
3	**Publicly available Data Storage**	YES	
4	**Privately available Data Storage**	YES	
5	**Blockchain purpose & architecture**	The canonical Blockchain is stored on "consensus Layer0" servers, where each server has a copy of the whole Blockchain. Initially BIX Blockchain's main purpose was using it as the decentralized database for record storage, while customers can connect to any of our Core Servers to login. In case of using BIX (Blockchain Information eXchange) API, they can continuously send requests for insertion of the data into the Blockchain, immediately followed by acknowledgement for each request. **<u>Two main Layers:</u>** **Consensus** (**Layer0**) – customized **DPoS** based; **Application** (**Layer1**) with optional **PoW** for new tokens As it is displayed in Figure1, **customized DPoS (Delegated Proof of Stake)** consensus algorithm is operational at **"Consensus Layer0"** and Decentralized Applications DApps can be created at "**Application Layer1"**, including PoW green mining, token exchange facility, multi token wallets, uploading & managing records in Blockchain, etc. Any functionality desired by the owner of permitted blockchain can be implemented at Layer1.	 **Figure 1.**

6	**Capability of issuing New Tokens**	YES (On a fly)	No programming experience required
7	**Types of Tokens with or without mining**	1. Digitally Signed (DST) Token. -> 2. New Tokens w/o Mining -> 3. New Coins with Proof of Work Mining ->	The only one token (GBX), that is generated at Consensus Layer 0. Set and implemented with "Fixed Capitalization" at the Application Layer 1. Set and implemented based on PoW protocol at the Application Layer 1.
8	**Blockchain advantages**	Blockchain, in general, is a linear storage of historical information/events that contains a suite of blocks; once created it cannot be changed. Oldest blocks can be disregarded, if there is no need of them, but cannot be changed. It's impossible adding any content to an accepted historical block.	The program may keep and maintain members data keys and it can point to the places in the Blockchain, where they are defined and/or redefined. It can keep a pointer to the last block, where the key was redefined for example. The historical Blockchain blocks are never changed (they are recording the whole history of transactions).
9	**Consensus Layer 0**	**Layer0 (core) servers** are servers responsible for the Blockchain operation.	They receive messages to be inserted and stored in BIX canonical version of the Blockchain and are responsible for finding a consensus of accepting new blocks. When requested, they send the information (like the content of a given block) to Layer1 servers.
10	**Application Layer 1**	**Application Layer1** provides a GUI for the Blockchain and, based on the data stored inside of the Blockchain at "**Consensus Layer0",** allows implementing any desired application (Smart Contract).	Layer1 servers get requests from users and submit corresponding messages to Layer0 servers to be inserted into the Blockchain. These messages belong to new account creation, money transfer, new token creation, etc. They can also request Layer0 servers for the content of any accepted block and/or the current state of internal databases. For example, they can request current balance of an account, content of currently mined block for Proof of Work (PoW) mining,
11	**Network architecture (types of nodes, etc.)**	Nodes are running on regular computers using either Linux, Windows or Mac OSX operating system(s).	GUIs are implemented via www (HTML5) protocol.
12	**Programming language for core, tokens and dApps**	The system is primarily implemented in "C", a highly portable procedural programming language, that is used for system applications, which form a major part of Windows, UNIX, Linux, Android iOS and OSX (the "Apple OS") operating systems.	Comparing to C#, Java, Python and other programming languages, programs written in "C" have much better performances. It might seem to be more difficult to develop software programs in "C", but Comparing to C++, "C" is better mainly because of its simplicity. Software program written in "C" usually performs better, works faster and requires significantly less hardware (machine) resources. Software program written in "C" is usually easier to maintain and extend than software written in other languages (OOP in particular). A program in "C" is easier to understand for 3rd party developers / code reviewers.
13	**Hardware and software requirements for running blockchain servers**	The software has very low hardware requirements running on MS-Windows and Linux system without requiring any particular version. The only requirement is the installation of OpenSsl library.	The memory (around 100 MB) is generally used only for pending messages, waiting to be inserted into the Blockchain. CPU is mainly spent on verification of signatures of received messages; large elliptic curve verifications may be a bit expensive.
14	**Crypto algorithms used for closure blocks, transactions, file encryption**	SHA512 Secp521r1	SHA512 is used everywhere, where a hash of data is required. "secp521r1" elliptic curve keys & signatures for signing messages and blocks.
15	**Data type and encryption**	Binary data OpenSsl algorithm called "EVP_aes_256_ctr"	Blockchain users can insert any binary data into the Blockchain. I.e. they can insert a file unencrypted or encrypted by their own algorithm. In case the user uploads a file with the basic encryption, we provide as default an OpenSsl algorithm called "EVP_aes_256_ctr" with a user entered symmetric key and a randomly generated initialization vector.

16	**Key generation algorithms**	OpenSSL Transport Layer Security (TLS) & Secure Sockets Layer (SSL) protocols Linux package named "haveged"	OpenSSL (robust, commercial-grade, and full-featured toolkit for the Transport Layer Security (TLS) and Secure Sockets Layer (SSL) protocols) is used to generate keys. Linux package named "haveged" is used to improve entropy of generated keys. This is used for everything related to security, hashes, signing messages, wallets, blocks, keys
17	**Communication between servers**	Node servers are communicating via freely inspired FIX (Financial Information eXchange) based, in-house developed proprietary messaging exchange BIX (Blockchain Information eXchange) protocol using standard TCP/IP transport	Communicating servers are using standard SSL connections, confirming their identity and protecting their communication channels. Moreover, for the security reasons, each message server nodes exchange (within the SSL connection) has an additional signature, generated with server's private key using elliptic curve signatures for this purpose. Well documented BIX (Blockchain Information eXchange) protocol allows members to develop their applications connected to the consensus Layer0 and participate in the network.
18	**Consensus of the Main Core**	Enhanced **DPoS** with additional confirmation rounds. Proprietary customized consensus algorithm used by Layer0 servers, is designed in a way allowing max data flow. Consensus protocol is minimizing the amount of data exchanged between servers.	Consensus is based on principals of accepting a block, when it is digitally signed by 2/3 of Layer0 permitted servers. This is the solution, which does not require a particular computation power and prevents forking of the blockchain as well. Once 2/3 of signatures for block number “N” are reached, there is no way to collect 2/3 of signatures for a different block for a simple reason – Layer0 servers will not ever sign 2 different blocks with the same “N” number. There are several rounds. To speed up finding a consensus, in the first round only 1/2 of servers is sufficient to pass to the next round. In the final round 2/3 of signatures are required.
19	**Forking risk**	There is no actual risk. The only critical moment is between the moment, when a server signs a block and the moment when the block is accepted and inserted into the Blockchain.	At the moment, when a server signs a block, it does not accept any other block than the one he had signed. In case the network, for any reason, accepts some other block, the server will stay blocked (frozen). However, this possibility is rather theoretical and still would not cause fork of the Blockchain.
20	**Consensus (Layer0) customized DPoS based**	**Layer0** is ensuring storage of the data into the Blockchain (you can imagine it as a file system in a computer).	Except the minor technical details, the main difference compared with the standard DPoS is that we are collecting signatures from 2/3rd of servers. Each block must be signed by large majority of participating servers before being inserted into the Blockchain. It makes the Blockchain particularly safe. An attacker would need to take full control of 1/3rd of servers to fork the Blockchain in comparison to Bitcoin, where a fork can occur naturally or standard DPoS, where an attacker needs to hack a single server only.
21	**Advantage of customized DPOS mechanism**	The main advantage of BIX consensus algorithm compared to others is the speed of block being accepted into the Blockchain. It doesn’t rely ambiguity on participants and instead requires secure signings of all participants during consensus process. To speed up finding the consensus, in the first round only 1/2 of servers is sufficient to pass to the next round. In the final round 2/3 of signatures is required.	After having briefly evaluated aBFT's DPoS algorithm, it looks very similar to our approach. From what we have seen DPoS BFT is using two rounds to find consensus, while our approach has refined the consensus further into several rounds, we are successively checking more and more requirements in successive rounds. So, if a block is invalid in the round “x”, then the information necessary for round “x+1” is not broadcasted in the network. Another important point if that there is no necessity checking the *entire* ledger with each latest block, because the latest block included in Blockchain is containing hash of all previous blocks, so re-checking of the entire ledger on each new block is not required.
22	**Consensus Layer0 block producers**	In the current version of the software producers hold the whole Blockchain. Layer0 servers are DPoS algorithm based “Block producers.	The main reason is that before joining the network each server is checking the consistency of the whole Blockchain. This is done by starting from genesis block (1st block of the Blockchain) and passing through all blocks until the most recent one. In the current version of the software the producers hold the whole Blockchain. **Future implementation** - implementing specific optimization for saving disk space. There will be some special "quasi-genesis" blocks, which will be generated within regular intervals and will be used by new servers to "start" their copy of the Blockchain.

23	**Block and message retrieval**	Layer0 is basically something like a special and safe file system in a computer.	Concerning efficiency of retrieval, blocks are stored in separate files in separate directories. I.e. retrieving block is as efficient as opening a file. To get a particular message from the block one doesn't have to traverse the given block and to find the message, but use a message ID instead
24	**Block producers rewards**	All producing servers are getting the same reward. There is no waiting list for block producers and/or any other governance mechanisms-built in.	We call this process mining; however, it is very different from mining as known from Bitcoin; there is no need of any hash power or hash rate, same as there is no waste of CPU power for searching a unique nonce. Blocks are validated by digital signatures since every server is signing blocks by its secret private key and the block is accepted if it collects signatures from enough servers. Owners of servers are equally rewarded every day and their reward does not depend on the number of blocks signed.
25	**Layer0 DST token mining**	Each server is rewarded for storing the blockchain, participating in the consensus process and signing blocks. Proposing a new block to the consensus does not require any cost.	Pre-built "mining method" for the Core Servers represents "Digitally Signed (GBX) Token" that is used for "Safe Storage" payments. Production Core Servers, for their participation and actual work in the Blockchain, are regularly compensated by obtaining a certain number of crypto units (GBX tokens) at the same time for each mining period. Since the storage space is a critical resource (comparing to the CPU consumption, that is negligible in BIX Blockchain), the GBX mining level of difficulty is constantly increased within the increased usage of the storage resources.
26	**Application (Layer1) with optional PoW mining for new tokens**	**Layer1** supports all programs, using Layer0 storing their data. Such applications are designed for different purposes. Example: Separate from Consensus Layer PoW mining process does not slow down the Blockchain Network.	The owner of the token or his volunteers have to install and run (or create their own) mining software. Basically, anyone can install mining software and provide PoW mining for any token. If there is nobody mining for the given token, no transactions of such token can be performed and the token is in fact "frozen". For avoiding such outcome there is a built-in, "low priority, low CPU consumption universal miner" in the blockchain, that insures the consistency of mining process of all PoW tokens. However, it's usage is limited.
27	**PoW protocol at Layer1 (available at the request)**	Means that transactions (sending/receiving money) on such tokens are performed only if some mathematical puzzle (PoW) is resolved.	Used by token owners looking to have an additional level of security or believe that the mining capability will add an additional value to their businesses. In case of payment transactions based on those tokens, they are performed in 3 steps: a. Request for the transaction is inserted into the Blockchain; b. Solution for the mathematical puzzle for the transactions is found inserted by some miner; c. Blockchain validates such solution before inserting it into the Blockchain and applies the transaction only if both (a. and b.) steps were performed successfully.
28	**Identification of the data stored into the blockchain**	Identified by two parameters: "**data type**" & "**data itself**"	"**data type**" - any string, similar to file system (like ".exe", ".pdf" or ".doc") "**data" itself** - an array of bytes, which corresponds to content of the file
29	**Data exchange**	A request to insert a **data** into blockchain **is considered as a "message". Each "message" has a "message ID"**.	Each **message is inserted into a particular block.** The data can be retrieved from the blockchain by entering the **block number and the message ID.** In case if false message is issued and insert it into a block, other servers will not approve such message preventing it from being included into the blockchain. The latest block included in blockchain contains hash of all previous blocks, so re-checking of the entire ledger within each new block issued is not required.
30	**Transactions**	For using Tokens and/or receiving/sending transactions users do not need to download and hold the whole Blockchain.	For sending a transaction user just needs to connect to one of the Layer0 servers and submit transaction. For receiving a transaction user connects to one of Layer0 servers and sends the request for particular block, followed by server responding back by sending the block to the user.

			User can also subscribe for the newly generated blocks and Layer0 servers will continuously send blocks as they are inserted into the blockchain. User can use or disregard such retrieved content in any way depending on the necessity factor.
31	**Handling impostors**	Each message must be signed by private key of the account holder and the private key of the server.	Also, the identity of servers is checked by usual SSL authentication. The only way how to hack this network is to take full control over the whole server and (get its IP address and private keys). However, due to the requirement of 2/3rd of signatures hackers would need to take full control over 1/3 of all participating servers to be able forking the blockchain.
32	**Scalability, performance, security, & encryption** **Transactional flow rate limit**	We have chosen SECP521r1 for digital signatures because currently it is by far the safest available option.	As of now realistic stable transactional flow rate equals to 1,500 messages per second with the pick of 5,000/ per sec. If there is a need for increasing speed, a weaker algorithm for signing less important messages can be used for better performance. NOTE: efficiency calculation, does not only estimate the cost of finding the consensus, but evaluates the whole implementation that includes the cost of finding a consensus, distribution of messages, cost of signing and verifying signatures of each message stored in the block.
33	**Transactional file type and size limit**	Currently permitted maximum size of a single message block is set to 10 MB. Maximum size configuration depends on settings and can be changed to up to 100 MB. In case of larger size, more memory capabilities for buffering purposes will be needed.	The blockchain can store various types of messages. The size of a single message is limited by the maximal size of the block to 10MB. The number of messages stored inside a single block is limited only by the total size of messages. The total size of messages must be lower than 10MB. In an extreme case of 10MB message size, each block will contain one message only. A typical use case is secured storage of important files like digitally signed contracts, documents, video files & etc. Any file types including raw data files are supported. Capability of storing large files is only one of BIX product features.
34	**Hash rate factor**	There is no blind generation and checking of hashes in the Blockchain consensus and mining approach.	Each server just generates its digital signature based on its secret private key. A block is accepted, when it collects enough digital signatures from Layer0 servers/nodes.
35	**Peer - to - peer decentralized token exchange capabilities.**	Pre-built decentralized exchange facility allows participants exchanging issued tokens using “Order Matching System” by placing and executing Limit & Market orders.	The Blockchain can be connected to 3rd party exchanges or other “Liquidity Sources” providing ECN (Electronic Communication Network) type operation with additional, outside of our Blockchain, tradable instruments provided by 3rd party Liquidity Providers (LPs).
36	**Testnet**		BIX “Pilot” blockchain is constantly undergoing through the test when running under high speed constantly performed transactions for few months. The last measured length of the “Testnet” blockchain few months ago was 745664 blocks and the current disk space usage was 6.2GB.
37	**BIX API**	BIX (Blockchain Information eXchange) API core engine is based on freely inspired FIX (Financial Information eXchange) protocol standards.	All additional future Applications, including applications developed by 3rd parties can become functional using BIX API. Any authorized application, developed by 3rd parties, can be connected to the blockchain using BIX API.
38	**Pre - built AML & KYC management for main Blockchain & New Tokens**	YES	KYC/AML compliance process can be controlled by central department or be connected to the 3rd KYC/AML party. In the current version, wallets can hold multiple token accounts and are not related to a particular token. Each member account holds multiple balances in multiple tokens with a separate balance for each existing token.