iZ³

Fully customizable private blockchain platform
izzz.io

General description of the blockchain platform IZZZIO

Technical Summary

Blockchain

Limited Confidence Proof of Activity (LCPoA)

Cross-chains

Cross-chain with limited length

WebSocket

The ability to work with partial or without synchronization with the network

Smart contracts

Messaging bus (Starwave)

Operation in browsers without extensions and plugins

Certain possibilities of the main and cross blockchains

Comparison with existing platforms

Version dated 12/09/2018

Technical Summary

Method for block confirmation	**Basic:** own development based on a hybrid PoW + PoA method - Limited Confidence Proof of Activity with a time confirmation binding. **Any other method is available.**
Software platform	Node.js > 8
Hashing	Filtered SHA256
Structure of the main circuit of blocks	JSON blocks with arbitrary content of up to 15 megabytes (depends on the configuration of the node)
Cross chains	JSON blocks of arbitrary content with adjustable maximum block volume
Data base	LevelDB, Sqlite3, Memory
Smart contracts	ECMAScript 6 (JS ES6), turing complete, with extended functionality, WASM
Primary network interface	WebSocket
API	REST

- Absence of mining in the main chain

- There is no big expense of computing resources due to the Limited Confidence Proof of Activity (LCPoA) algorithm with time-bound

- The possibility of using a single blockchain circuit in any project

- High-speed data transfer protocol in the blockchain network (Starwave)

Blockchain

The main chain consists of JSON blocks with a structure

- SHA256 hash correspondence to network filter conditions
- Block number
- The start time of block generation
- The end time of the block generation
- Block data

Limited Confidence Proof of Activity (LCPoA)

IZZZIO blockchain uses its own algorithm for creation and confirmation of blocks bound to work on time, and not requiring large computing resources. The security of the blockchain network itself and the consensus is provided by the mechanism of the "limited confidence zone" with automatic creation of control points and also with checking the block generation time and checking the total network time.

At any point in time, it is possible to recheck the time spent on generating the block due to the constant minimum generation time of the block.

Detailed description of the LCPoA algorithm: https://goo.gl/Gtb5ge

Cross-chains

Cross-chain is a mechanism of creating additional chains of blocks, with special parameters. In this case, additional chains work on top of the main network block. Additional block chains are created using special blocks and smart contracts.

Cross-chain with limited length

For cross-chains, the chain length limitation function is available. In the event that the amount of data exceeds the recordable, two scenarios are available:

- Intentional "forgetting" of old blocks that extend beyond the length of the chain
- Ban on writing data in a chain with subsequent freezing of block generation

In the first case, the circuit can be used to stream information between network clients (eg VPN, events notification, chat rooms, etc.)
In the second case, it's like an immutable data store.
Recording in a cross-chain can also be limited to the rules of a smart contract.

WebSocket

Due to the use of WebSocket as the main way of interaction with the network, it becomes possible to draft entire browser clients of our blockchain network. The WebSocket protocol itself is very common, and it has implementations in many programming languages and in all modern browsers.

The ability to work with partial or without synchronization with the network

Clients of the network can work with the network without full synchronization, provided that several nodes with a fully synchronized chain are used as data sources for the network. At the moment, this mechanism is used by one of our tools - Candy.

Smart contracts

Blockchain supports smart contracts written in JavaScript ES6, which are executed in an isolated environment. Contracts can use the functionality of the main and secondary blockchains, tokens, additional checks of external factors. Also for smart contracts writing it is suggested to use an alternative dialect of ECMAScript - STAR Script.

Main features:

1. JavaScript ES6. There is support for working with random numbers (Math.random) and time (Date).
2. Events.
3. Interaction with other contracts in the network.
4. Support for large numbers (BigNumber) for secure data calculations including tokens.
5. Built-in iterable and non-iterable data structures: KeyValue, BlockchainArraySafe, BlockchainArray, TokenRegister.
6. Built-in functionality of recommended contracts.

Messaging bus (Starwave)

The messaging bus allows to send random data to any member of the network, while the message delivery route will be generated dynamically at the first time you try to send a message. When you change the route, the system rebuilds the connections automatically.

The protocol supports the transmission of broadcast and personal messages. Messages can be public, open with a cryptographic signature, and encrypted. The encrypted channel is organized by using keys exchange between participants of the Diffie-Hellman (DH) protocol.

The data transfer mechanism is similar to the functionality of UDP (User Datagram Protocol) broadcast messages, and does not provide information about message delivery status.

Operation in browsers without extensions and plugins

Since the block computer uses the WebSocket protocol to exchange information, it becomes possible to work with the network directly, using browser tools. For this we have developed a special open source script - Candy.

Using blockchain to store and load any information allows you to evenly distribute the load among the nodes, which makes iZ^3 work as a load distributor, and protect against DDoS attacks.

Certain possibilities of the main and cross blockchains

- Storage of any information. Blocks of a certain type are processed only by those clients that can use them. The remaining blocks are simply transferred unchanged.
- Using blockchain to verify digital signatures as a trusted data source
- Using blockchain to organize information exchange networks
- Creation of trusted data sources with limited storage time
- Archiving data
- Creation of digital assets (tokens, assets)
- Exchange of information between the client and the server

and others.

Comparison with existing platforms

Characteristics	Waves	Bitcoin based	Ethereum	Universa	BitShares	EOS	IZZZIO
Main concept	Easily generated tokens	Crypto currency	Smart contracts performed in a common environment	A chain of smart contracts	Blockchain for the exchanges	Infrastructure for DApp	Maximum universal platform
Any projects implementation	-	-	-	+	-	+	+
Tokens creation based on code	-	-	+	+	+	+	+
Turing full smart contracts	+	-	+	+	-	+	+
Cross-blockchain (network of blockchain with intersections)	-	-	-	-	-	-	+
The ability to use without full synchronization with the network	+ thin client	In some implementations or a thin client	+ thin client	+	-	+	+
Chains with length limitation	-	-	-	-	-	-	+
Transmission bus	-	-	-	n/a	-	-	+
Interaction with the browser without plug-ins and external services	-	-	-	-	-	-	+
Shardig	-	-	-	-	-	+	+
Block Validation Algorithm	PoS	PoW and others	PoW	-	DPoS	DPoS	LCPoA
Programming language	Scala, Java	C++	C++, Go	Java	C++	C++ & WASM	Node.js & WASM