



KRONOBIT BLOCKCHAIN ChainPaper

V 2.1b

KRONOBIT BLOCKCHAIN [KNB]

Abstract:

This whitepaper provides an overview of Kronobit Blockchain (KNB), a Proof of Authority (PoA) EVM blockchain designed for Web3 implementations. Kronobit offers high transaction speed, low transaction costs, and compatibility with popular wallets and development tools. This document outlines the key features, ecosystem components, and social media channels associated with Kronobit Blockchain.

Official Project Logo



1. Introduction

KRONOBIT Blockchain (KNB) is a Proof of Authority (PoA) EVM blockchain that combines high transaction speed with low transaction costs, making it suitable for various Web3 implementations. This whitepaper explores the technical specifications, ecosystem components, and development resources associated with KRONOBIT Blockchain.

Why is Kronobit Blockchain unique?

Kronobit Networks Blockchain is developing technology to address current issues of security, scalability, and transparency present in existing blockchains. It aims to enhance security through a Proof of Authority (PoA) system for projects and transactions, preventing the creation of Dapps and tokens without passing the KYC verification process. This grants confidence and security to investors, users, and projects in general. For these reasons, Kronobit Networks is developing the KdT (Know Development Team) system.

Educational Blockchain

Kronobit Networks will have a decentralized online educational platform (Kronobit Academy) where qualified facilitators can upload audio, visual, and written content, generating rewards based on requests and payments from teachers. Payments and rewards can be made using the platform's coins, and upon completion, each course will be backed by an official certification from Kronobit Academy.

Content

All facilitators can provide any type of educational content and set a reasonable price for those seeking knowledge. There are no content restrictions, as long as good manners and respect for others are maintained. Within Kronobit Academy as a part of KronoTV, free basic training courses, talks, and documentaries will be broadcasted, ranging from literacy to various vocational technical courses and workshops, with the aim of promoting healthy personal development. Users may also receive random rewards in some of these courses and workshops.

The protocol includes:

- An ad-free audiovisual platform that respects good manners and respect for others, regardless of race, creed, or sexual preference. (KronoTV - Academy)
- Multilanguage support.

2. Technical Specifications

- Kronobit Blockchain operates on a Proof of Authority consensus mechanism.
- The blockchain achieves a transaction speed of over 100,000 transactions per second (TPS).
- Block time is set at 5 seconds, ensuring fast confirmation of transactions.
- Transaction costs are less than 0.000002\$, making Kronobit cost-effective for Web3 applications.
- Kronobit is compatible with popular wallets such as Metamask, TrustWallet (on Test), Safepal, Nabox, and more to come
- The native symbol for Kronobit is KNB.

Official Kronobit Blockchain parameters to add any EVM compatible Wallet or Development:

Mainnet: <https://explorer.qbitscan.com>

ChainID: 13600

Testnet: <https://testnet-explorer.qbitscan.com/>

ChainID: 7622

Faucet: <https://faucet.qbitscan.com>

3. Native Coin Details

- The maximum token supply for KNB is 650 Million KNB Coins
- KNB also has a Token version on the Polygon Chain & Binance Smart Chain. 89% of other chains are locked
- a total of 50 million of KNB coins will be Burn per year until the total supply is reduced to 250 million coins.

4. Ecosystem Components

- <https://kronosend.com> : Multi coins token sender
- <https://kronoswap.finance> : Official Dex, Bridge, Swap, Farms
- <https://kronotv.com> : Decentralized Multimedia educative Streaming
- <https://kronostake.com> : Official Staking
- <https://kronopad.org> : A platform where users can launch, lock coins & LP tokens.
- <https://kronolist.org> : Official Multi chain wallet aggregator
- <https://bytedetective.tech> : Official Audit service

5. Social Media Channels

Stay connected with Kronobit Blockchain through official social media channels:

- Twitter Official: https://twitter.com/Kronobit_Knb
- Telegram: <https://t.me/kronobit> Spanish
- Telegram 2: https://t.me/KronobitNetworks_support English
- Telegram 3: <https://t.me/knbchain> Global
- Telegram 4: <https://t.me/knbiran> Persia
- Instagram: <https://www.instagram.com/kronobit/>
- Medium: <https://kronobitnetworks.medium.com/>
- GitHub: <https://github.com/QbitNetworks/go-ethereum>

6. under Developments

Kronobit Blockchain has ongoing development plans to enhance its ecosystem:

- **Krono Magic & NFT Marketplace:** A fusion of 2 Dapps into better, bigger, NFT marketplace + NFT Staking reward
- **Krono Miner:** A mining game simulator with great features & rewards
- **Android Mobie Wallet:** Official android wallet project

6a. Kronobit Compatibility

Kronobit Networks blockchain is fully compatible with the Ethereum Virtual Machine (EVM). It also supports Web3 JS API and RPC.

6b. Kronobit Blockchain Nodes

Validator Nodes: Are responsible for validating transactions, maintaining the Kronobit Networks chain.

6c. Virtual Nodes

These nodes provide support and speed to the network.

6d. Key Management

This article serves as a guide for implementing a key management strategy on the client side of your Decentralized Application (DApp) on Kronobit Networks Blockchain.

Setup Web3

Web3.js is a JavaScript library that enables communication between our client-side application and the blockchain. We configure Web3 to interact via Metamask.

Documentation for web3.js can be found [\[here\]\(#\)](#).

Connect to KNB Network

```
```javascript
// Mainnet
const web3 = new Web3('https://mainnet-rpc.qbitscan.com');
// Testnet
const web3 = new Web3('https://testnet-rpc.qbitscan.com');
```
```

Set Up Account

If the installation and instantiation of Web3 were successful, the following code should successfully return a random account:

```
```javascript
const account = web3.eth.accounts.create();
```
```

Recover Account

If you have backed up the private key of your account, you can use it to restore your account.

```
```javascript
const account =
web3.eth.accounts.privateKeyToAccount("$private-key");
```
```

Full Example

```
```javascript
const Web3 = require('web3');

async function main() {
 const web3 = new Web3('https://mainnet-rpc.qbitscan.com');
 const loader = setupLoader({ provider: web3 }).web3;

 const account = web3.eth.accounts.create();
 console.log(account);
}
```
```

6e. Wallet Provider for DApps

A wallet provider is a service that enables DApps to offer wallets to their end-users without handling the security concerns of key management. It's an excellent way to prepare your DApp for a broader audience.

Arkane: allows easy integration of your app with Kronobit Chain, whether you already have a DApp integrated with web3 or are building a new application from scratch. Arkane provides a seamless and enjoyable experience for you and your users on both web and mobile platforms.

Arkane facilitates interaction with Kronobit Networks, allowing you to create blockchain wallets, various asset types such as fungible (QBN20, ERC20), and non-fungible tokens (NFTs) like QBN721 and ERC1155, and interact with smart contracts. Alongside superior developer experience, Arkane offers users a user-friendly interface.

Each application is unique with different needs, so Arkane provides various ways of interaction. Applications supporting Web3 are recommended to integrate the Arkane Web3 provider, while others are suggested to use the Arkane Widget.

Key Features

- Support for web and mobile applications
- Offers social logins
- Provides a fiat-on-ramp
- Only wallet supporting NFTs (QBN721 and ERC1155) on Kronobit Networks
- Multichain support for both KNB and Ethereum
- Easy integration using web3
- Tailored for a mainstream audience
- Offers in-app customer support
- Provides a complete test environment
- Wallet creation via APIs

Getting Started

If you already support Web3 technology, you can enhance the UX within your application by integrating the Arkane Web3 provider, a smart wrapper around the existing Web3 Ethereum JavaScript API.

By utilizing our Web3 provider, you can leverage the full potential of Arkane with minimal effort and onboard less tech-savvy users without redirecting them from your application or requiring them to download third-party plugins. Integration takes just 2 steps and 5 minutes.

Don't support Web3 yet?

Don't worry! We've got you covered with our  Widget - Arkane Connect.

Step 1: Add the library to your project

Install the library by downloading it to your project via NPM:

```
```bash
npm i @arkane-network/web3-arkane-provider
```
```

Then, add the script to the head of your page:

```
```html
<script src="/node_modules/@arkane-network/web3-arkane-
provider/dist/web3-arkane-provider.js"></script>
```
```

After adding the JavaScript file, a global Arkane object is added to your window, serving as the gateway for creating the web3 wrapper and fully integrating the widget - Arkane Connect.

Step 2: Initialize the web3 provider

Add the following lines of code to your project to load the Arkane web3 provider:

```
```javascript
Arkane.createArkaneProviderEngine({clientId:
'Arketype'}).then(provider => {
 web3 = new Web3(provider);
});
```
```

The web3 instance now functions as if it were injected by Parity or Metamask. You can fetch wallets, sign transactions, and messages.

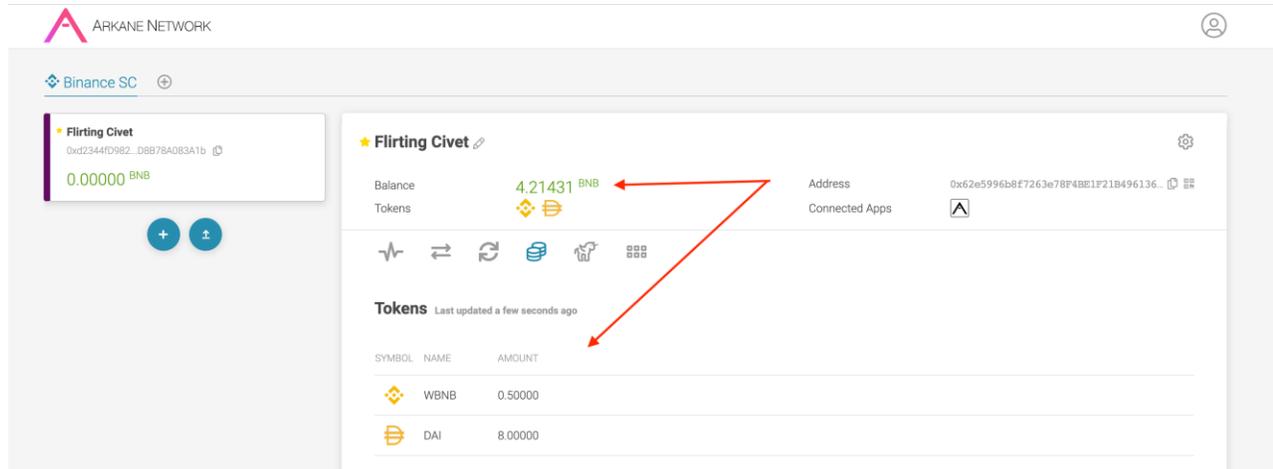
Congratulations! Your DApp now supports Arkane 🎉

To connect to Arkane's production environment and mainnet, you'll need to register your app and request your Client ID.

To learn more about the fantastic world Arkane offers, check out their documentation.

Receive KNB and QBN20 Tokens

Every KNB wallet in Arkane can send and receive KNB or QBN20 tokens, which will automatically appear in your wallet.



6f. Build QBN20 Token on Kronobit Blockchain

A QBN20 token must implement the interface IQBN20 in IQBN20.sol. This is a template contract QBN20Token.template. Users just need to fill in `_name`, `_symbol`, `_decimals`, and `_totalSupply` according to their own requirements:

```

```solidity
constructor() public {
 _name = {{TOKEN_NAME}};
 _symbol = {{TOKEN_SYMBOL}};
 _decimals = {{DECIMALS}};
 _totalSupply = {{TOTAL_SUPPLY}};
 _balances[msg.sender] = _totalSupply;

 emit Transfer(address(0), msg.sender, _totalSupply);
}
```

```

Then users can use Remix IDE and Metamask to compile and deploy the QBN20 contract to KNB.

Interact with Contract with Web3 and NodeJS.

Connect to Kronobit networks's public RPC endpoint

```

```javascript
const Web3 = require('web3');
// mainnet
const web3 = new Web3('https://mainnet-rpc.qbitscan.com');

// testnet
const web3 = new Web3('https://testnet-rpc.qbitscan.com');
```

```

Create a wallet

```
```javascript
web3.eth.accounts.create([entropy]);
```
```

Output:

```
```plaintext
{
 address: '0x926605D0729a968266f1BB299d8Df0471C4F5367',
 privateKey:
'0x6b4618539d95f205f33e916e89404b301dde545c0c4acc181fd0c0b4
2708bad3',
 signTransaction: [Function: signTransaction],
 sign: [Function: sign],
 encrypt: [Function: encrypt]
}
```
```

Recover a wallet

```
```javascript
const account =
web3.eth.accounts.privateKeyToAccount("0xe500f5754d761d74c3e
b6c2566f4c568b81379bf5ce9c1ecd475d40efe23c577")
```
```

Check balance

```
```javascript
web3.eth.getBalance (holder).then(console.log);
```
```

Output:

The balance will be bumped by e18 for KNB.

```
```plaintext
6249621999900000000
```
```

Create transaction

Parameters:

- Object - The transaction object to send:
- from - String|Number: The address for the sending account. Uses the web3.eth.defaultAccount property if not specified. Or an address or index of a local wallet in web3.eth.accounts.wallet.
- to - String: (optional) The destination address of the message, left undefined for a contract-creation transaction.
- value - Number|String|BN|BigNumber: (optional) The value transferred for the transaction in wei, also the endowment if it's a contract-creation transaction.
- gas - Number: (optional, default: To-Be-Determined) The amount of gas to use for the transaction (unused gas is refunded).
- gasPrice - Number|String|BN|BigNumber: (optional) The price of gas for this transaction in wei, defaults to web3.eth.gasPrice.
- data - String: (optional) Either an ABI byte string containing the data of the function call on a contract, or in the case of a contract-creation transaction, the initialization code.
- nonce - Number: (optional) Integer of a nonce. This allows overwriting your own pending transactions that use the same nonce.

```
```javascript
```

```
// Make a transaction using the promise
```

```
web3.eth.sendTransaction({
 from: holder,
 to: '0xB75fbeB0BC7CC0e9F9880f78a245046eCBDBB0D',
 value: '10000000000000000000',
 gas: 5000000,
 gasPrice: 18e9,
}, function(err, transactionHash) {
 if (err) {
 console.log(err);
 } else {
 console.log(transactionHash);
 }
});
```
```

How to Run a Full Node on Kronobit Networks

Full Node Functions:

- Stores the full blockchain history on disk and can answer data requests from the network.
- Receives and validates new blocks and transactions.
- Verifies the states of every account.

Supported Platforms:

We support running a full node on Mac OS X and Linux.

Suggested Requirements:

Full Node:

- VPS running recent versions of Mac OS X or Linux.
- 4 cores of CPU and 8 gigabytes of memory (RAM).
- A broadband Internet connection with upload/download speeds of 5 megabytes per second.

Steps to Run a Full Node:

1. Download `knb_mainnet.json` and `static-nodes.json` from <https://github.com/Qbitnetworks/go-ethereum>

```
```bash
wget https://raw.githubusercontent.com/Qbitnetworks/go-ethereum/master/knb_mainnet.json
wget https://raw.githubusercontent.com/Qbitnetworks/go-ethereum/master/static-nodes.json
```
```

2. Make a node folder

```
```bash
mkdir node
```
```

3. Initialize the Node

```
```bash
./geth --datadir ./node init knb_mainnet.json
```
```

4. Copy the `static-nodes.json` to `node/geth`

5. Run the Nodes

```
```bash
./geth --datadir node --syncmode 'full' --gcmode=archive --port
40605 --http --http.port 3545 --http.api
'personal,eth,net,web3,personal,admin,miner,txpool,debug' --
bootnodes
enode://e8b960268a7f9644e2a886dbbd211b237eb770b3961b260c5df
75def555711a90b132caaedf58579bffc6660d3360fc653b3210fb5af3f8
948ae55dc7850cbbf@202.143.110.107:0?discport=40606 --
networkid 13600 --allow-insecure-unlock
```
```

Ensure to adjust the parameters according to your system requirements and network settings.

Where to Find Support:

For technical assistance and discussions regarding running our software:

- Telegram: <https://t.me/Kronobit>)

For reporting bugs or making technical contributions:

- GitHub: <https://github.com/Qbitnetworks>

Official contact for Kronobit Networks: admin@kronobit.org

Wallet Support for Kronobit Networks:

Supported wallets:

- MetaMask
- Nabox Wallet
- Safepal