ENGLISH

WHITEPAPER

PRIZM

The initial concept of digital currency

Prizm Whitepaper Revision — June, 2020

PZM.SPACE

INTRODUCTION

Bitcoin is the world's first decentralized digital currency, allowing you to store and transfer cryptographic coins easily using the P2P network to transmit information, hashing as a synchronization signal to prevent double spending, as well as a powerful scripting system to determine the owner of coins. This demonstates a growing technology and business infrastructure. According to the original design, bitcoins are interchangeable and seen as a neutral means of exchange. Bitcoins may have special properties supported by either the Issuer or a public agreement, and have a value independent of the underlying nominal value. Bitcoin has proven that the P2P electronic payment system can really work and process payments without any third party involvement.

However, for the entire e-economy to be based on a fully decentralized peer-to-peer solution, the system must be able to do the following:

1 - Process transactions securely, quickly and efficiently, in the amount of thousands per hour or more;
2 - Encourage people to participate in network security;
3 - Scale at the global level with the minimum consumption of resources;
4 - And to be able to work on a wide range of devices including mobile.

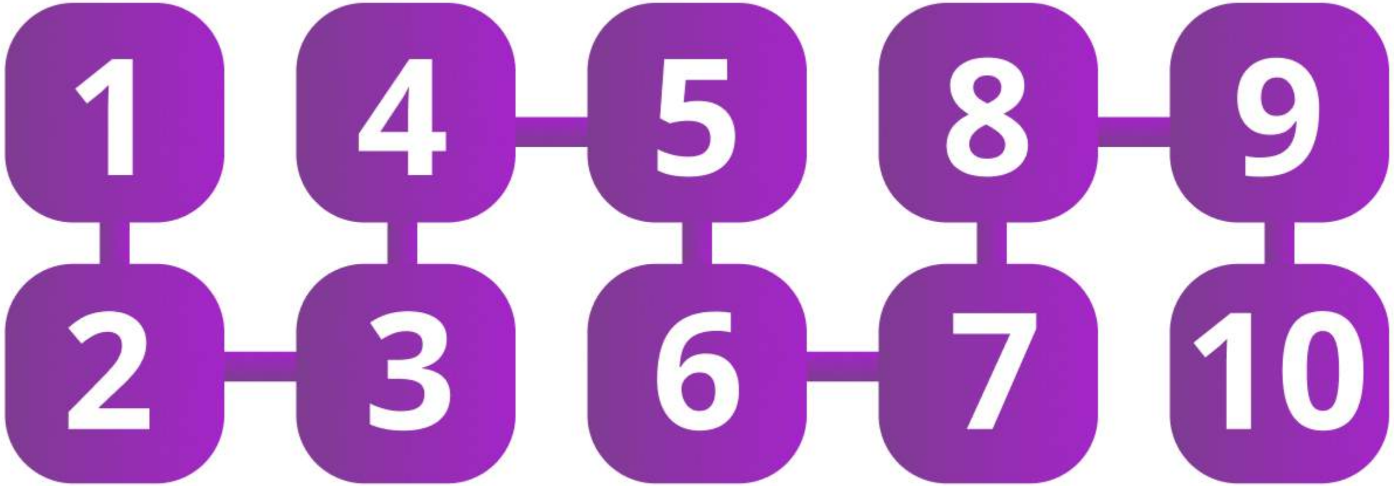
PZM (pronounced as "Prizm") meets all these conditions. Another advantage, the unique one, is **PARAMINING**, which is not presented in any other existing cryptocurrencies.

But more on that later.

PRIZM is a 100% proof-of-stake cryptocurrency based on the NEXT-kernel which is built on Java with an open source code. The unique PRIZM proof-of-stake algorithm does not depend on any implementation of the "coin age" concept used by other proof-of-stake cryptocurrencies, and is resistant to the so-called "nothing at stake" attacks. The total number of coins available was distributed in the Genesis block. Curve25519 cryptography is used to provide a balance of security and the required processing power along with the more commonly used SHA256 hashing algorithms.

60sec

Blocks are generated every **60** seconds, on average, by accounts that are not blocked on network nodes.

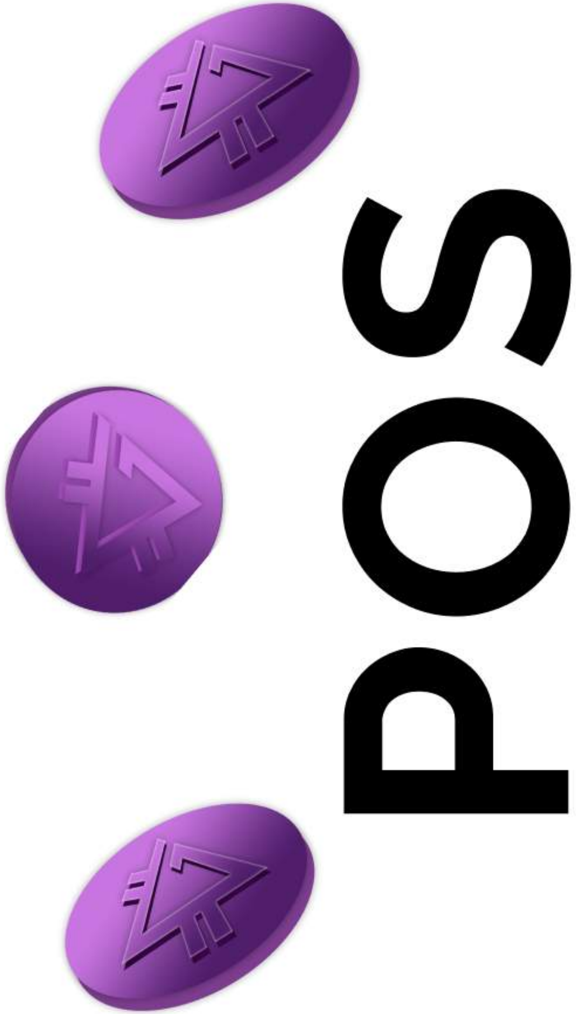


PZM are redistributed by incorporating transaction fees that are awarded to an account when it successfully creates a block. This process is known as forging and is akin to the notion of "mining" used by other cryptocurrencies. Transactions are considered secure after **10** block confirmations, and the current architecture and block size of the **PZM** allow processing up to **367,200** transactions per day.

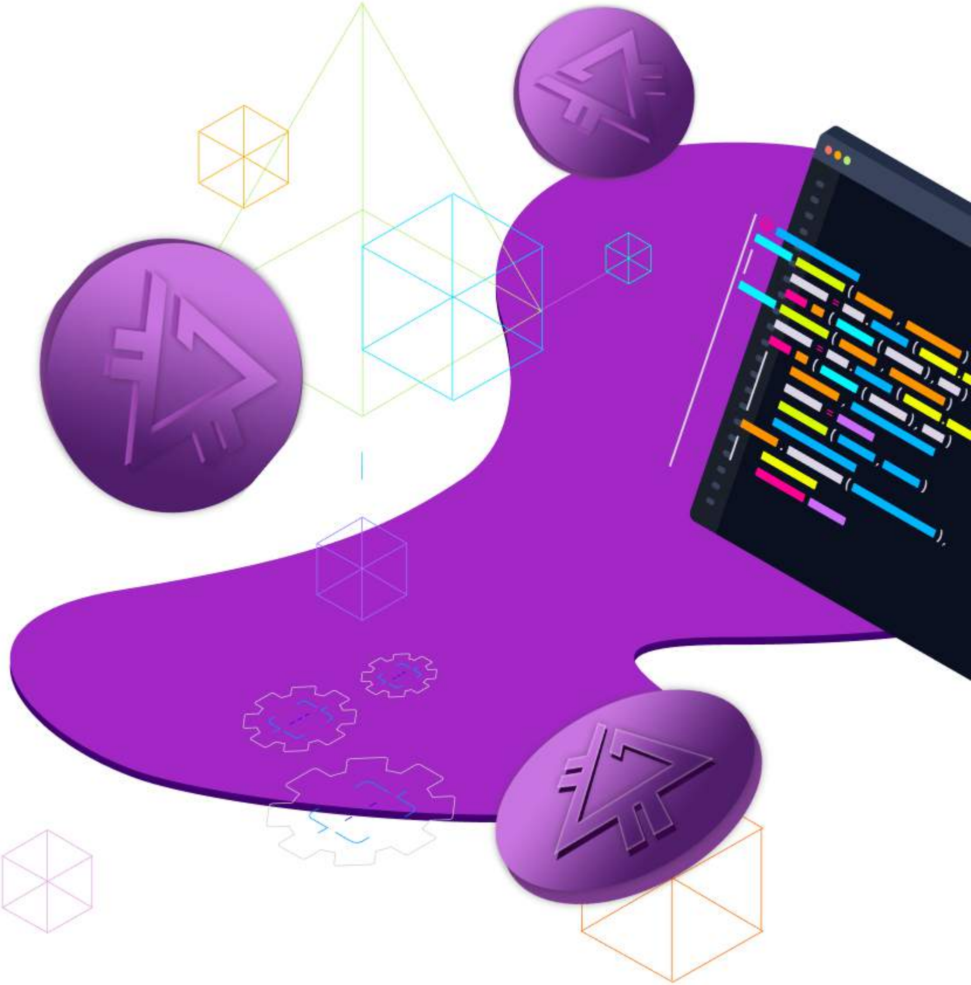
PZM includes the implementation of **Transparent Forging** that would allow you to increase the performance of transaction processing by two orders of magnitude by using the generation algorithm is deterministic block, in combination with additional security mechanisms of the network.

Proof of Stake

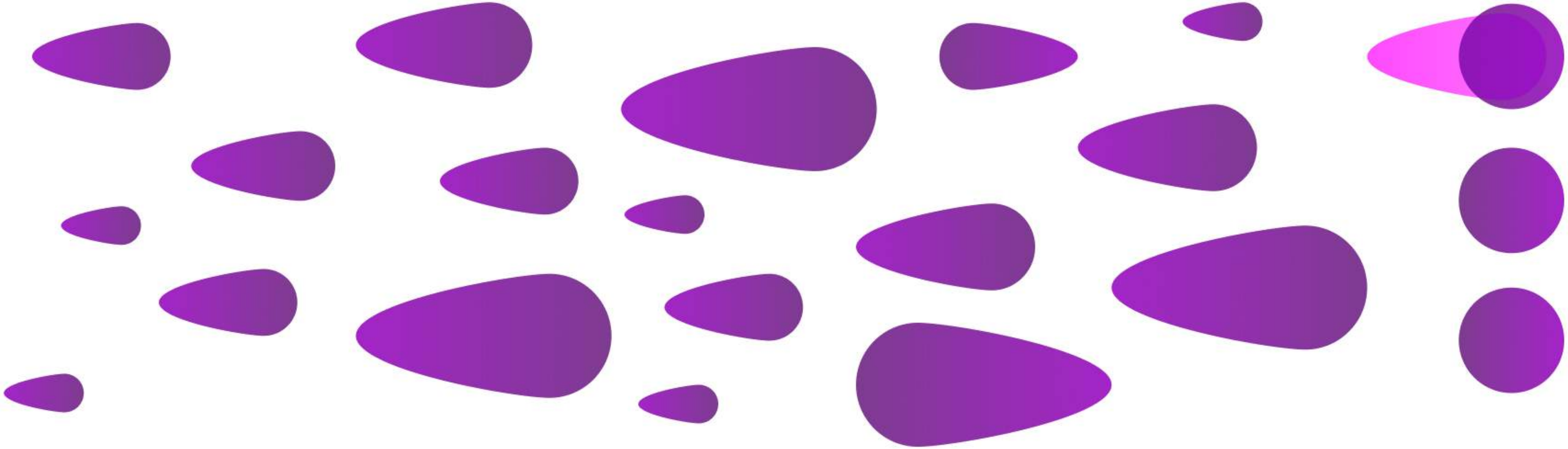
In the traditional "Proof of Work" model used by the majority of cryptocurrencies, network security is ensured by participants performing "work". They use their resources (calculation/processing time) to reconcile transactions with double costs and in order to impose extraordinary costs on those who attempt to collapse transactions. For this work, participants are awarded with PZM, and their frequency and amount vary depending on the working parameters of the cryptocurrency. This process is known as mining. The frequency of block generation, which determines each available reward for mining cryptocurrencies, as a rule, should remain constant.



As a result, the labor intensity of the work required to obtain rewards should increase as the network becomes more efficient. As the **Proof of Work** network develops, the individual user has less incentive to support the network, since their potential reward is distributed among more colleagues. In search of profitability miners continue to invest resources in the form of specialized, patented equipment that requires significant investment and high current energy costs. Over time, the network becomes more centralized as smaller partners (those who can do less work) drop out or pool their resources into pools. The Creator of bitcoin Satoshi Nakamoto, intended bitcoin network to be completely decentralized. But no one could predict that the incentives provided by **Proof of Work** systems would lead to the centralization of the mining process. This leads to potential vulnerabilities.



PROOF OF STAKE



GHash. The bitcoin IO pool has reached **51%** of bitcoin mining power in the past, and the top five bitcoin mining pools make up **70%** of the hashing power of the network. The concept of decentralization is at risk of total loss. In the **Proof of Stake** model used by **Prizm**, network security is regulated by partners who have a stake in the network.

The incentives provided by this algorithm are not conducive to centralization as **Proof of Work** algorithms,and data show that the **Prizm** network remains highly decentralized since its inception: a large (and growing) number of unique accounts contributing blocks to the network, and five top accounts generate **35%** of the total number of blocks.

PROOF OF STAKE IN PRIZM

Prizm uses a system in which each “coin” in the account can be considered as a miniature mining farm. The more coins are contained in the account, the more the account is likely to receive the right to create a block. The total "reward" received as a result of creating the block is the amount of commissions for transactions located inside the block. **PRIZM** does not create any new coins as a result of creating blocks. **PRIZM** does not create any new coins as a result of building blocks. PZM redistribution occurs as a result of block generators receiving transaction fees, so the term "forging" is used in this context instead of "mining" and implies to "create relationships or new conditions”. Subsequent blocks are generated based on verifiable, unique, and almost unpredictable information from the previous block. Blocks are linked by virtue of these links, creating a chain of blocks (and transactions) that can be traced back to the Genesis block. Block generation time is approximately **59** seconds, but changes in the probabilities have led to the fact that the average generation time of the block can be **80** seconds, there are longer intervals of blocks. The security of the Blockchain is always set in the Proof-of-Stake system.

.

The basic principles applying to the Prizm Proof of Stake algorithm:

- The cumulative complexity value is stored as a parameter in each block, and each subsequent block receives its new "complexity" from the value of the previous block. In the case of ambiguity, the network achieves consensus by choosing a block or chain fragment with the highest cumulative complexity.

- In order for account holders not to move their funds from one account to another as a means of manipulating in order to be able to generate blocks, coins must be stationary within the account for 1,440 blocks before they can contribute to the block generation process. Coins that meet this criterion contribute to an efficient account balance, and that balance is used to determine the probability of forging.

- To prevent an attacker from creating a new chain all the way from the Genesis block, the network only allows the restructuring of the chain of 720 blocks located behind the current block. Any block below this threshold shall be rejected. This move threshold can be considered as the only fixed PZM checkpoint.

- Due to the extremely low probability that any account will take over the Blockchain management by creating its own chain of blocks, transactions are considered safe if they are encoded into a block that is located 10 blocks behind the current block.

COMPARSION WITH PEERCOIN PROOF OF STAKE

Peercoin uses the setting of the age of the coin as part of the algorithm of the mining probability. In this system, the longer your **Peercoins** have been on your account (up to **90** days), the more power (coin age) they have to create a block. The act of "Meeting" the block requires the consumption of the dignity of the coin age, and the network determines the consensus by selecting the chain with the greatest total consumed coin age. When the **Peercoin** blocks are separated, the consumed coin age is returned back to the original block account.

As a result, the cost to attack the **Peercoin** network is low, since the intruders can continue to try to generate blocks (called grinding the stake) until they are successful. **Peercoin** minimizes these and other risks by centrally publishing blockchain checkpoints several times a day to "freeze" the blockchain and block transactions. **Prizm** does not use the coin age as part of the forging algorithm. The chance of creating a block by any account depends only on its current balance (which is the advantage of each account), the time since the last block (which is shared by all forging accounts) and the base target value (which is also common for all accounts).

PRIZM

TOKENS

10 MLN
PZM

INITIAL EMISSION

6 BLN
PZM

FINAL EMISSION

The initial emission is **10** million PZM and the final amount is **6** billion **PZM**. The coins were issued with the creation of the Genesis block (the first block in the blockchain). Paramining implementes in all countries of the world, at nominal costs and in limited quantities in order to achieve the initial **Prizm** decentralization. The total amount of **PZM** will be 6 billion tokens. Account Genesis generates anti-coin signals of Paramining (signal of sending the coins to a certain wallet) to the limit of minus 6 billion PZM.

The existence of anti-coins in Genesis has several interesting side effects:

All tokens sent to the Genesis account are effectively destroyed, as the negative account balance cancels them. The main function of Prizm is the traditional payment system, but it was created to do much more. The goals of the CWT community can be achieved under the condition of PZM parity with the main Fiat currencies.

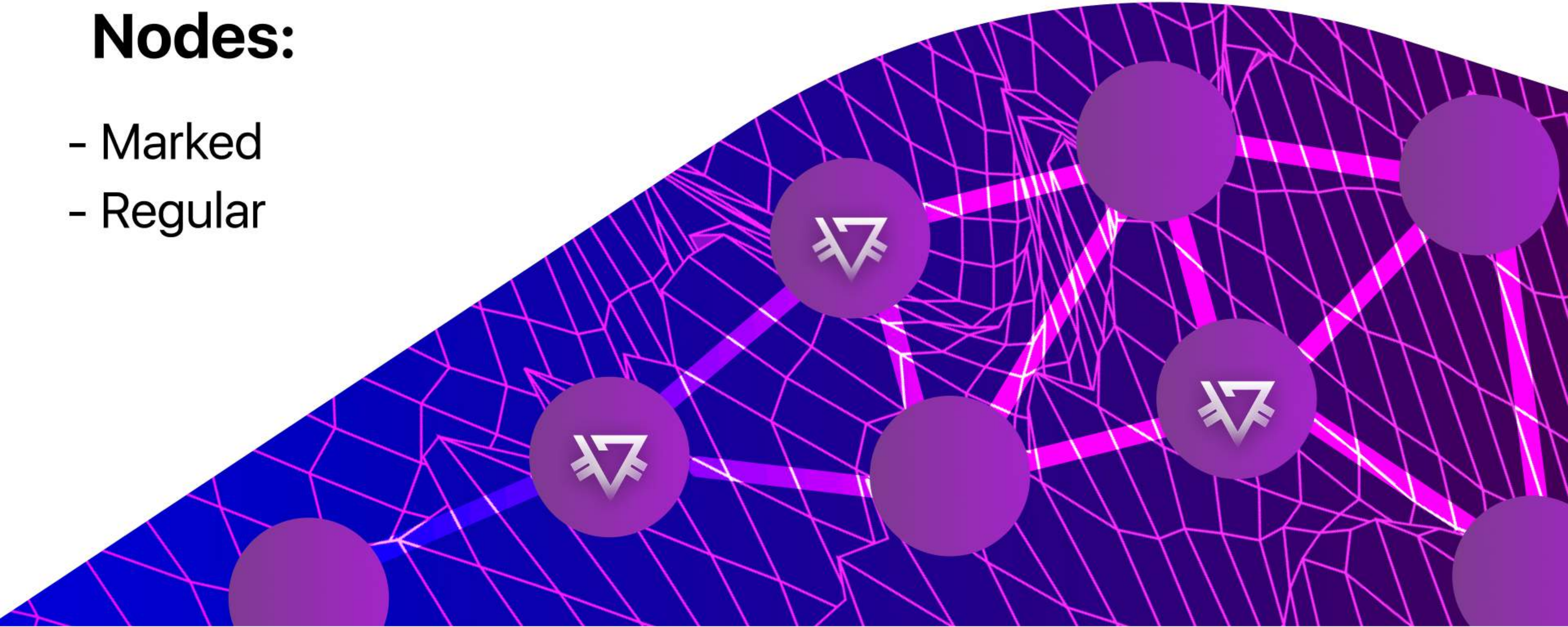
PRIZM

NETWORK NODE

Prizm network node is any device that makes a transaction or block data into the network. Any device with the PZM software is treated as a node. Nodes can be divided into two types: marked and regular.

Nodes:

- Marked
- Regular



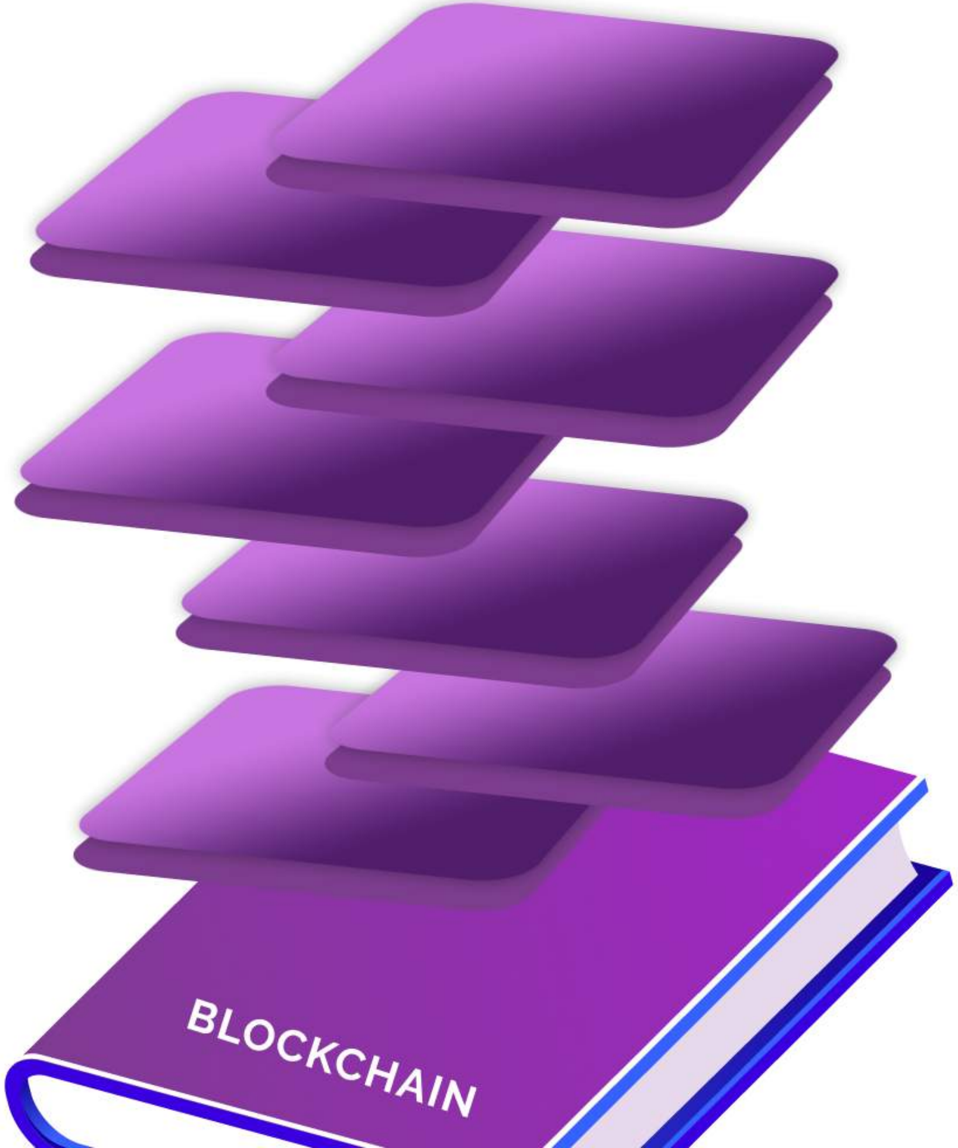
NETWORK NODE

A marked node is simply a node that is marked with an encrypted token received from the account's private key; this token can be decoded to show the specific **PZM** account address and balance that are associated with the node. The label placement act on a node adds a layer of accountability and trust, so that marked nodes are more reliable than those that do not have markings on the network. The more the balance of account is linked to a marked node, the more confidence is given to this node. While an attacker might want to mark a node to gain trust on the network and then use that trust for malicious purposes, the barrier to entry (the cost of **PZM** needed to build adequate trust) prevents such abuse.

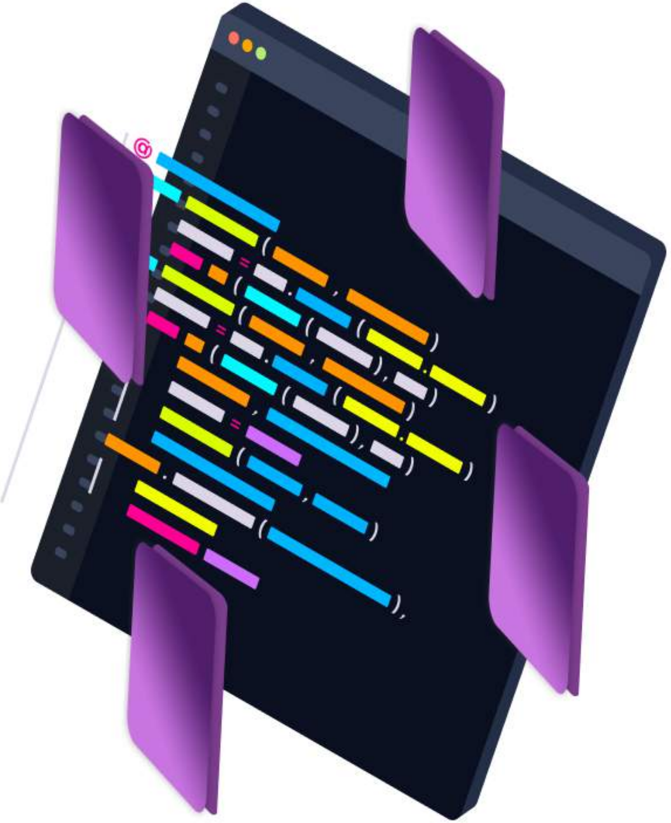
Each node in the **PZM** network has the ability to process and transmit both transactions and block information. Blocks are scanned as they are received from other nodes, and in cases where a block check is not performed, the nodes can be "blacklisted" temporarily to prevent the dissemination of invalid block data.

Each node has built-in **DDOS** protection mechanisms (**Distributed Denial of services)** that limit the number of network requests from any user to **30** per second.

As with other cryptocurrencies, the **PZM** Ledger (Ledger of transactions) is built and stored in a linked series of blocks known as the **blockchain**. This workbook provides a permanent record of the transactions that have occurred, and it also establishes the order in which the transactions were made. A copy of the Blockchain is stored on each node in the **Prizm** network, and each account that is not blocked on the node (by providing the private key of this account) has the ability to generate blocks, providing that at least one incoming transaction in the account has been confirmed **1,440** times. Any account that meets these criteria is called an active account. In **PZM**, each block contains up to **255** transactions, all of which are preceded by a **192**-byte Header that contains identifying parameters. Each transaction in a block is represented by a maximum of **160** bytes, and the maximum block size is **32** KB.



All blocks contain the following parameters:

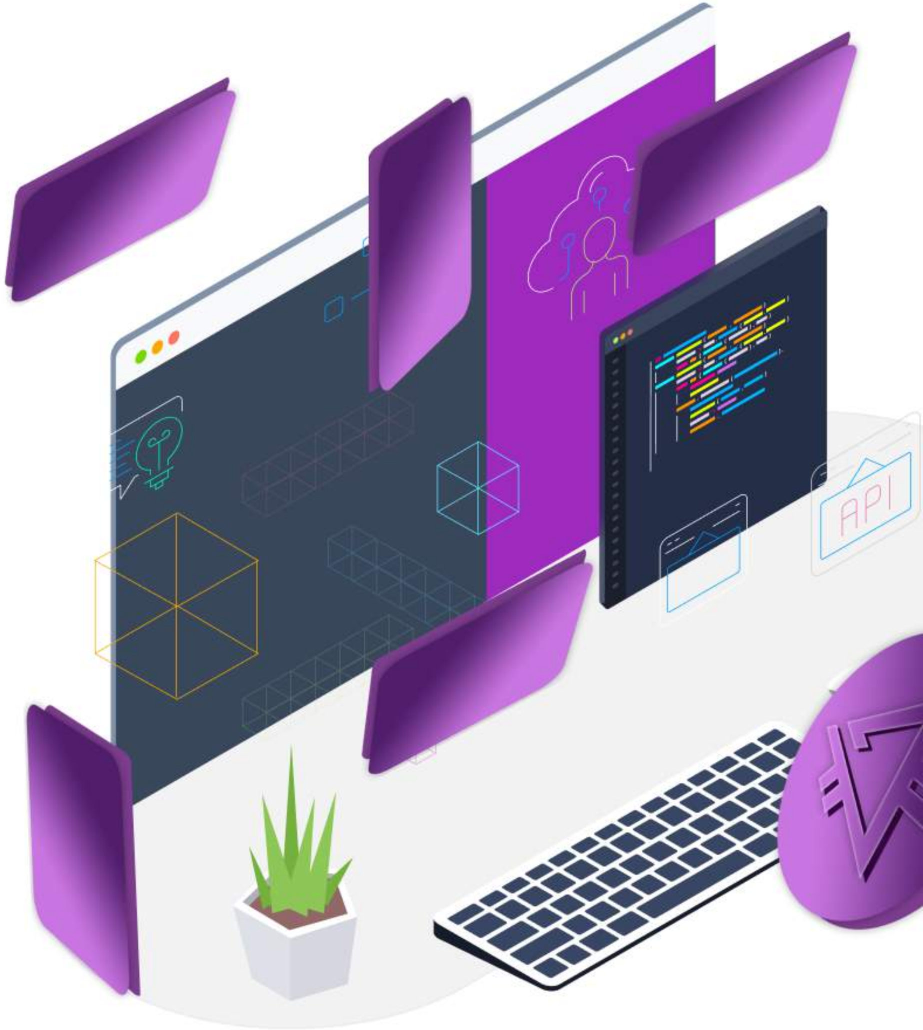


- The version of the block, the height of the block and the block ID
- Block timestamp expressed in seconds from the Genesis block
- The **ID** of the account that created the block, as well as the public key of the account
- ID and hash of the previous block
- Number of transactions stored in the block
- The total amount of **PZM** represented by transactions and commissions in the block
- Transaction data for all transactions included in the block, along with transaction IDs
- Length of the payload block and a value of a hash function of the payload block
- The base target value and cumulative difficulty for the block

There are three values that are key to determine which account has the right to generate a block, which account is entitled to create a block and which block is considered to be authoritative in the time of a conflict: base target value, target value and cumulative difficulty.

Base target value

To win the right to forge (generate) a block, all active Prizm accounts "compete" by trying to create a hash value that is lower than the specified base target value. This base target value changes from block to block and is derived from the base target value of the previous block multiplied by the amount of time it took to generate that block.



Target value

Each account calculates its own target value based on the current effective rate.

This value is equal:

$$T = Tb \times S \times Be$$

There the following is:

T - the new target value
Tb - the reference target value
S - the elapsed time since the last block in seconds
Be - the effective account balance

As you can see from the formula, the target value increases with every second that has passed since the previous block.

The maximum target value is 1,53722867 x 1017, and the minimum target value is half the base target value of the previous block. This target value and the base target value are the same for all accounts that are trying to forge on top of a particular block. The only defined account parameter is an effective balance parameter.

FORGING (CREATION OF BLOCKS)

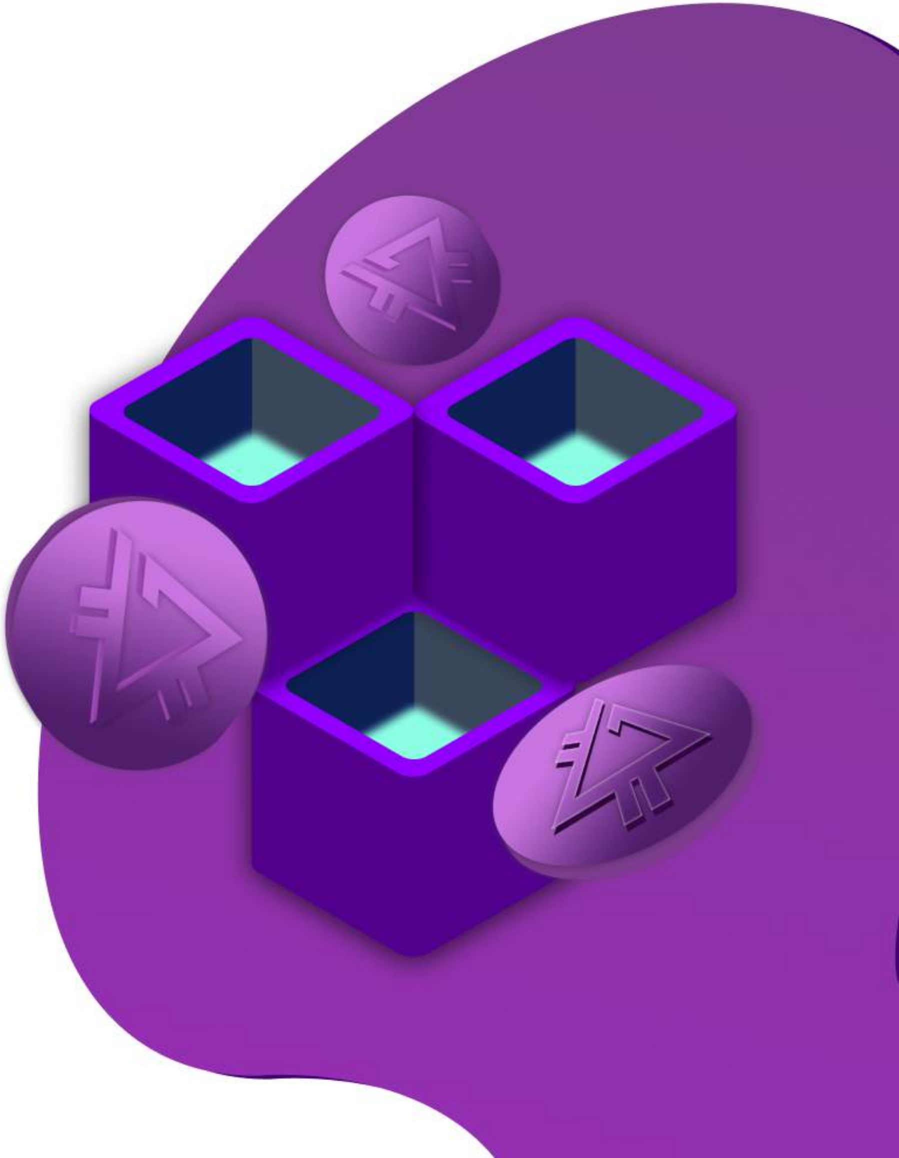
The total complexity

The total value of complexity obtained from the base target value according to the formula:

$$Dcb = Dpb + 264 / Tb$$

Where the following is:

Dcb - the complexity of the current block
Dpb - the complexity of the previous block
Tb - the base target value of the current block

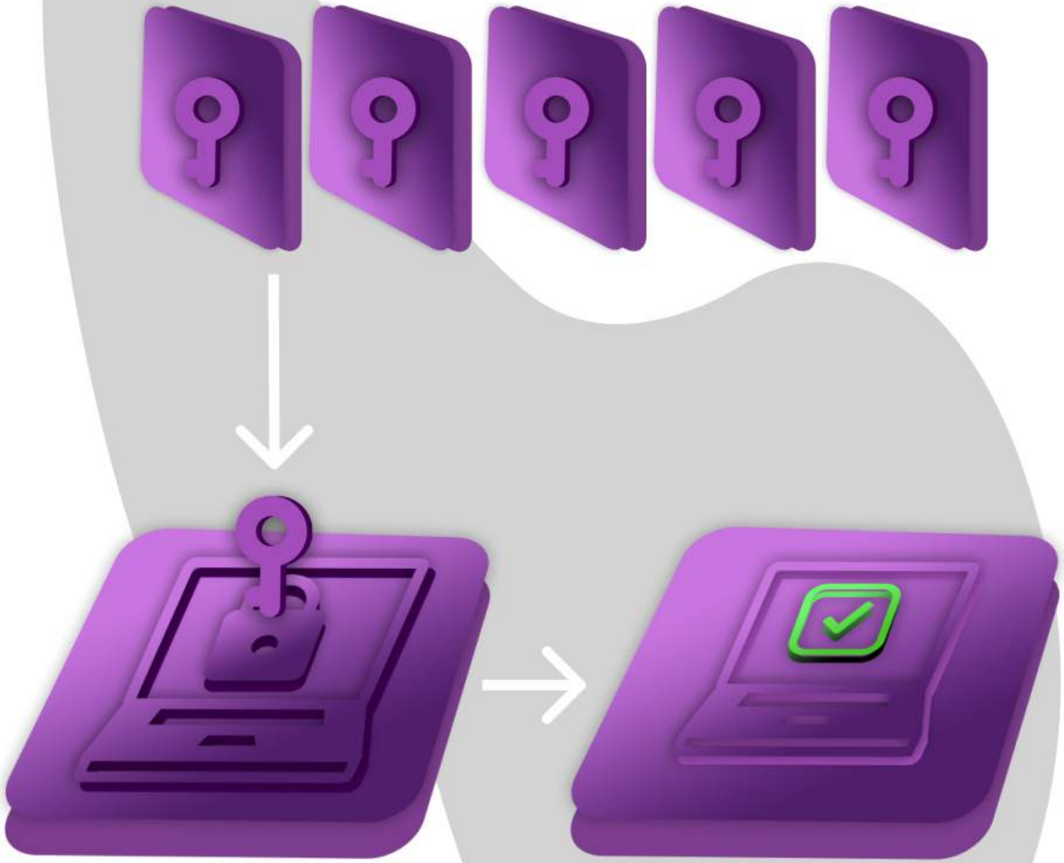


FORGING ALGORITHM

Each block in the chain has a signature generation parameter. To participate in the forging process of a block, the active account cryptologically signs the previous generated block with its own public key. This creates a 64-byte signature that is then hashed using SHA256. The first **8** bytes of the resulting hash give a number called the hit of the account. The hit is compared to the current target value. If the calculated hit is lower than the target, the next block can be generated. As noted in the target value formula, the target value increases with each second. Even if there are only a few active accounts on the network, one of them will eventually generate a block because the target value will become very large. The consequence of this is that you can estimate the time it will take for any account to force the block by comparing the hit value of that account to the target value. The last point is of great importance. Since any node can request an effective balance for any active account, it is possible to go through all active accounts to determine their individual hit value. This means that with reasonable accuracy, you can predict what the following account wins right to block counterfeit. A shuffle attack can be triggered by moving a stake into an account that will generate the next block, which is another reason why **PZM** bet must be stationary for **1,440** blocks before it can contribute to the forging (through an effective balance value). Interestingly, the new base target for the next block cannot be reasonably predicted, so a virtually deterministic process of determining who will force the next block becomes more and more stochastic as attempts are made to predict future blocks.

FORGING ALGORITHM

This feature of the forging **PRIZM** algorithm helps to form the basis for the development and implementation of the **Transparent Forging** algorithm. When an active account is granted the right to create a block, it combines up to **255** available unconfirmed transactions into a new block and populates the block with all its necessary parameters. This block is then transmitted to the network as a Blockchain candidate. The payload that generates the account and all signatures on each block can be checked by all the network nodes that receive it. In a situation where multiple blocks are generated, nodes select the block with the highest accumulated complexity as the authoritative block. Because the block data is distributed among the members (peers), forks (unauthorized chain fragments) are detected and dismantled by examining the cumulative complexity of the chains stored in each fork.

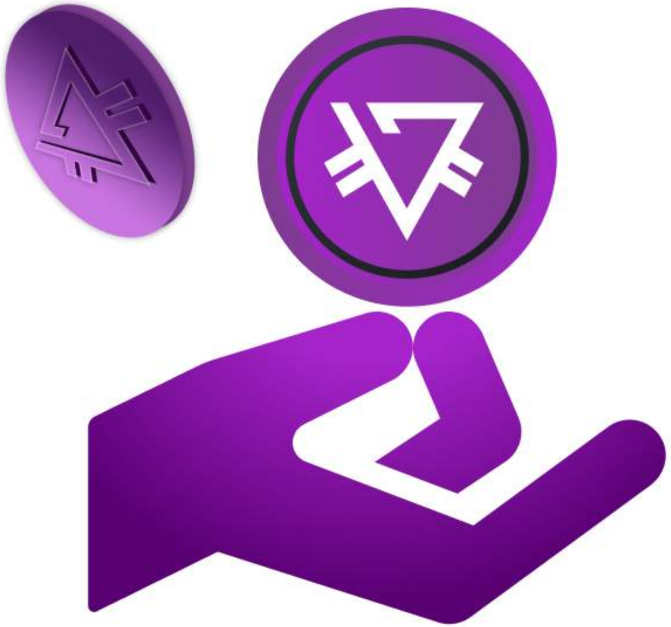


PARAMINING

PARAMINING — is PRIZM's key advantage over the other cryptocurrencies. PRIZM developers added a unique, linear-retrograde mechanism for determining the reward for funds storage aimed at economic attractiveness and gradual replacement of all existing financial instruments of the world by the mass of PZM to the basic mechanism of forming.

That is an addition to the basic forging mechanism, which does not increase the amount of funds in the system. In PZM this additional mechanism is ParaMining, which creates new coins, according to metrics of standard mathematics development of normalized financial system in the slice of the world economy. According to our calculations, only such a format of coin weight growth can provide a gradual and confident replacement of all existing economic instruments.

PARAMINING



The rate of new coins mining using ParaMining is calculated from three main parameters, which are **(1) the number of coins in a personal wallet, (2) the number of coins in followers' wallets up to 88 levels, and (3) the mining difficulty factor.** The difficulty factor is calculated as a percentage, and is proportional to the total number of coins issued. The maximum difficulty level for standard accounts is **98%**, which corresponds to **3 billion generated PZM**. For accounts in the **HOLD - MODE**, the maximum difficulty level is **97%.** According to the characteristics, Paramining is an MLM 2.0 system that excludes everything that pushes a simple person from the network business, but at the same time involves him in the development of the network to increase the speed of coin mining on his personal wallet.

When making any transaction in the wallet, the ParaMining system writes a blockchain containing the value of the number of coins of the wallet owner and the number of coins in the wallets of its followers, at this moment new coins are generated to the wallet balance.

HOLD - keep coins in your personal wallet and not make any transactions, due to this, you can reduce the coefficient of difficulty in mining coins and increase the profitability of your wallet.

PARAMINING SYSTEM is the most advanced tool for promotion and popularization, as it has no analogues in any modern cryptocurrency. The main advantage of Paramining is that no network user can interfere with this mechanism and falsify new coins, all users can monitor the number of coins issued by the system in real time. Paramining works on any wallet with a balance of over 1 PZM and automatically stops when a balance of 1 million PZM is reached.

Also, for the first time, the system of establishing referral links without the use of any links was applied. After creating a new wallet, the system captures in the blockchain who the first transaction arrives from and permanently establishes a referral chain that cannot be changed, this makes it easy to build global MLM networks and increase the speed of new coin mining.

The technical implementation is currently not described in detail due to the fact that for all of us, the main thing is to create not 100 “dead” tools, but the one with good support and good work. If our know-how is revealed, then someone will definitely try to repeat it and this will inadvertently lead to a scattering of attention and the use of this idea not for noble and significant goals for our planet, but for goals that we don’t know and are not always positive coloring intent.

P R I Z M

PARAMINING OPTIONS

To start mining a new PZM, just a single coin is needed in an electronic wallet that automatically starts ParaMining. This is a process that allows you to increase the number of coins in the wallet without any costs of electricity.

Paramining starts with 1 coin and stops automatically when you reach 1 million coins in your wallet.

Paramaning — is a unique method of creating new coins by all users simultaneously, regulated by three parameters:

1 - The number of coins in your personal wallet
2 - The number of coins of the followers structure
3 - Mining difficulty called Paratax

1 - Number of coins in your personal wallet

Number of coins in personal wallet (PZM)	Daily profit	Monthly profit
from 500.000 to 1.000.000	0,33%	9,9%
from 100.000 to 499.999	0,28%	8,4%
from 50.000 to 99.999	0,25%	7,5%
from 10.000 to 49.999	0,21%	6,3%
from 1.000 to 9.999	0,18%	5,4%
from 100 to 999	0,14%	4,2%
from 1 to 99	0,12%	3,6%

2 - Number of coins of the followers structure

Total Volume	Multiplier
from 1000 to 9999	2.18
from 10.000 to 99.999	2.36
from 100.000 to 999.999	2.77
from 1.000.000 to 9.999.999	3.05
from 10.000.000 to 99.999.999	3.36
from 100.000.000 to 999.999.999	3.88
1.000.000.000	4.37

The Paramining principle is based on the fundamental laws of physics, from the "Visible Radiation" section. Like the model of our Universe, the system is constantly expanding, gaining speed, thanks to a complex recount of interest with a period of 55 seconds.

PRIZM

PARAMINING OPTIONS

Any user with a balance

from 1000 to 110 000 PZM

can increase the period for calculating compound interest, provided that his balance is involved in forging. **To start the HOLD period**, you need to generate at least one block with target transactions. The **HOLD** period cannot be interrupted by an incoming transaction or by receiving a forging fee. The duration of the **HOLD** period is not limited, provided that the forger generates at least one block out of **100,000.**

The HOLD period is interrupted by an outgoing transaction.

PRIZM

PARAMINING OPTIONS

3 - Mining difficulty

PARATAX

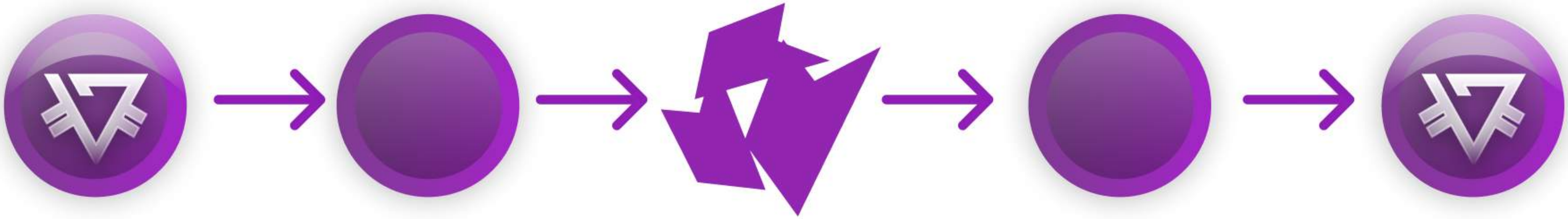
is a linear increase in the difficulty of coins generation, expressed as a percentage of the number of coins already mined by all users.

The maximum limit of PARATAX will be 98% at the time of production of 3 billion PZM

For FORGERS whose balance at the time of generating the block does not exceed 110,000 PZM, the maximum value of PARATAX is 97%

PRIZM ACCOUNTS

Prizm implements a smart wallet as part of its design: all accounts are stored on the network with personal keys for each possible account address, directly derived from the code phrase of each account using a combination of operations **SHA256** and **Curve25519**. Each account is represented by a **64**-bit number, and this number is expressed as the account address using the error Correction of Solomon-Code, which allows you to detect up to four errors in the address of the account or correct up to two errors. This format was implemented in response to concerns that an incorrect account address could result in coins, aliases, or assets being irreversibly transferred to erroneous target accounts. Account addresses are always preceded by **"PRIZM-..."**, which makes Prizm account addresses easily recognizable and distinct from the address formats used by other cryptocurrencies.



Address account, coded Solomon-Code associated with a secret passphrase is generated in the following way:

- **The secret passphrase is hashed using SHA256 to retrieve the account's private key.**
- **The private key is encrypted using Curve25519 to obtain the public key of the account.**
- **The public key is hashed with SHA256 to obtain the account ID.**
- **The first 64 bits of the account ID are the visible account number.**
- **Encoding Solomon-Code, the visible account number with the prefix "PRIZM -" generates the address of the account.**

When an account is accessed with a secret passphrase for the first time, it is not protected by a public key. When the first outgoing transaction is made from the account, the 256-bit public key received from the passphrase is stored in the blockchain, and this protects the account. The address space for public keys (2256) is larger than the address space for account numbers (264), so there is no one-to-one matching of code words to account numbers and possible collisions. These collisions are detected and prevented as follows: after a certain passphrase is used to access the account, and the account is protected with a 256-bit public key, no other public-private key pair can access this account number.

The properties of the account's balance:

1 An effective account balance is used as the basis for billing your account. An effective account balance consists of all the coins that were stationary on that account for **1,440** blocks. In addition, the "Account Leasing" function allows you to set an effective balance on another account for a temporary period.

2 The guaranteed account balance consists of all tokens that were stationary on the account for **1,440** units. Unlike an efficient balance sheet, this balance cannot be assigned to any other account.

3 The base account balance accounts for all transactions that have at least one confirmation.
The boost account balance shows the total amount of **PZM** received as a result of the successful forcing blocks.

4 Unconfirmed account balance is the one that is displayed in **Prizm** clients. It represents the current account balance, net of the coins involved in unconfirmed, sent transactions.

5 Guaranteed asset balances list (make a list) guaranteed balances of all assets associated with a particular account.

6 Unconfirmed balances and unconfirmed asset balance list of all assets associated with a specific account.

WALLET.DAT

Bitcoin and related currencies often use an encrypted file, under the name and wallet, to store generated addresses for receiving coins. **The Next** core used in the **Prizm** neither simulate this functionality, neither rule it out. Client developers can implement a system in which a private key group for **Prizm** accounts is stored in an encrypted stand-alone file.

Confirmation of transactions

All PZM transactions are considered unconfirmed until they are included in a valid network block. The newly created blocks are distributed to the network by the node (and the associated account) that creates them, and the transaction that is included in the block is considered to be received by one confirmation. As subsequent blocks are added to an existing blockchain, each additional block adds another confirmation to the number of transaction confirmations. If a transaction is not included in the block before it expires, it burns and is deleted from the transaction pool.

The timing of the transaction

Each transaction contains a deadline parameter set to the number of minutes since the transaction was sent to the network. By default, the deadline is 1440 minutes (24 hours). A transaction that was sent to the network but was not included in the block is called an unconfirmed transaction.

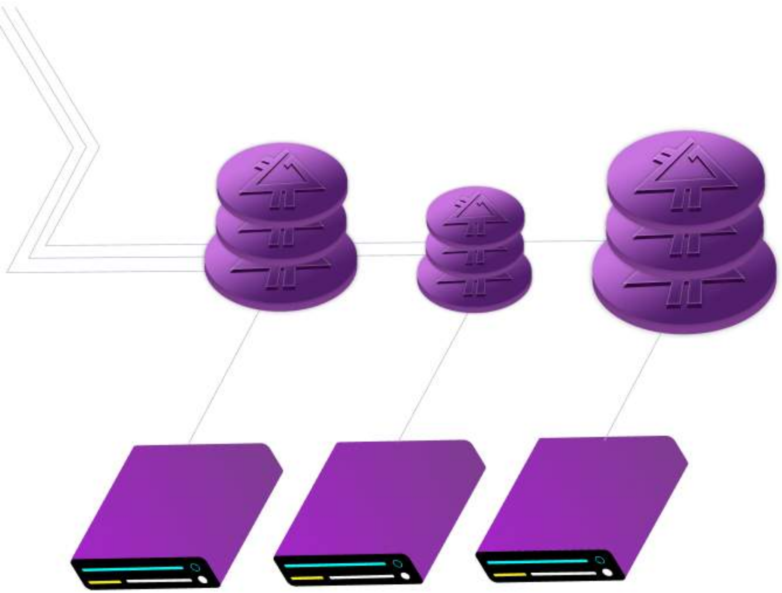
If the transaction was not included in the block before the transaction deadline, the transaction is removed from the network. Transactions can be left unconfirmed because they are invalid or distorted, or because blocks are filled with transactions that offer to pay a higher Commission. In the future, features such as multi-signature transactions can use deadlines as a means of enforcing expiration.

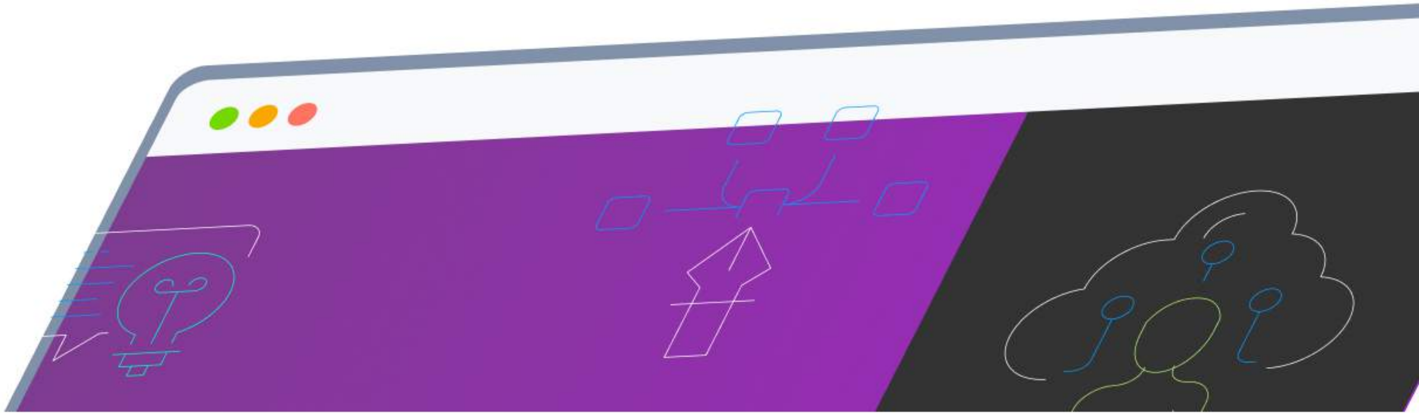
Creating and processing of transactions

Detailed information about creating and processing of a PZM transaction is as follows: - The sender specifies the transaction parameters.

Transaction types change, and you specify the desired type when you create the transaction, but for all transactions you should specify multiple parameters:

- **The private key for the sending account**
- **The transaction deadline**
- **Optional transaction reference**





BASICS OF PRIZM CRYPTOGRAPHY

The key exchange in **Prizm** is based on the **Curve25519** algorithm, which generates a shared secret using **Diffie-Hellman**'s fast efficient elliptic curve with a high degree of protection. The algorithm was first demonstrated by Daniel J. Bernstein in **2006**. **Next** implementations on **Java** were reviewed by **Doctor Evil** in March **2014**. The signing of messages in **Prizm** is carried out using the **Elliptic-Curve** digital signature algorithm (**EC-KCDSA**), which was defined by the **IEEE P1363a** group in **1998** by the **KCDSA** Task force team. Both algorithms were chosen to balance speed and security for a key size of only **32** bytes.

Main features

Advanced JavaScript client

Convenient client application of the Second generation embedded in the distribution of the basic software Prizm, and which can be accessed via a local web browser. The client provides full support for all major Prizm features implemented so that users ' private keys are never available online. It also includes an enhanced administrative interface and built-in Javadoc documentation for the Prizm low-priority application programming interface.

Portable device

Thanks to its cross platform based on Java roots, the hashing of Proof of Stake and its future ability to reduce block chain size, Prizm is extremely well suited for use on small low-power low-resource devices. Android and iPhone apps and software have been ported to low-power ARM devices such as the RaspberryPi and CubieTruck platforms. The ability to implement Prizm on low-power, always-connected devices such as smartphones allows us to present a scenario in which most Prizm networks are supported on mobile devices. Low cost and resource consumption of these devices significantly reduce network costs compared to traditional cryptocurrency Proof of Work.

Basic payments

The most fundamental feature of any cryptocurrency is the ability to transfer coins from one account to another. This is the most fundamental type of Prizm transactions, and it allows you to use basic payment functions.

P R I Z M

PRIZM KEY FEATURES

POS — forging typing

Mixing two technologies at the same time : paramining + forging. Source codes are closed (not lined), up to a certain time, as protection against clones as the guarantee that the system will be liquid.

- **Partnership program of 88 levels in the structure**
- **NEXT / Proof of stake core of the cryptosystem**
- **User-friendly interface for mobile devices**
- **The user password is not being sent to the server**

Nothing at stake

In the "nothing is at stake" attack, forgers try to build blocks on top of all the forks they see because it costs them almost nothing, and because ignoring any fork can mean losing on the block the rewards that would be earned if that fork were designed to become the chain with the most cumulative difficulty. Although this attack is theoretically possible, it is currently impractical. Prizm network does not experience long blockchain forks, and the reward for low blocks does not give a strong incentive for profit; In addition, compromising network security and trust for such a small profit could make any victory hollow.

Attacks on history

In "attack on history," someone acquires a large number of coins, sells them, and then tries to create a successful fork just before their coins have been sold or exchanged. If the attack fails, the attempt is worthless because the coins are already sold or transferred; If the attack succeeds, the attacker gets his tokens back. Extreme forms of this attack include obtaining private keys from old accounts and using them to build a successful chain directly from the Genesis block. In Prizms, the main history attack usually fails because all bets must be fixed at 1,440 blocks before they can be used for forging; In addition, the effective account balance that each block generates is verified as part of the block check. The extreme form of this attack usually fails because the PRIZM blockchain cannot be reorganized by more than 720 blocks behind the current block height. This limits the time frame in which a bad actor could establish this form of attack.

APPLICATION

The Bitcoin problems, considered in Prizm.

Prizm was created as a cryptocurrency 2.0 - response to Bitcoin. Prizm uses functions that are well established in Bitcoin, and consider the aspects of concern. This application addresses issues with the Bitcoin Protocol and network that are smoothed out by Prizm technology.

About of transactions per day

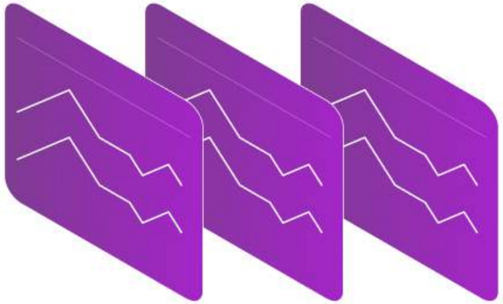
By the end of 2013, the number of transactions processed in the Bitcoin network reached a maximum of 70,000 per day, which is about 0.8 transactions per second (tps). The current standard Bitcoin block size of one megabyte, generated (on average) every ten minutes on the "full" site of customers, limits the maximum bandwidth of the existing Bitcoin network to about 7 TPS. Compare this with the bandwidth of the VISA network to handle 10,000 TPS, and you'll see that Bitcoin can't compete as it exists today.

Blockchain Size

The Bitcoin Blockchain is a complete sequential collection of the generated data blocks containing the e-book registers all Bitcoin transactions that have taken place since its launch in January 2009th. Four years later, in January 2013th, Bitcoin's blockchain size was 4 gigabytes (GB) - the approximate amount of data needed to store a two-hour movie on a DVD. Eighteen months later, in July 2014th, the Bitcoin blockchain size increased by almost five-to 19 gigabytes (GB) 37. Bitcoin blockchain is undergoing exponential growth, and modifications to the original Bitcoin Protocol will require a solution to this.

Answer of PRIZM

In its current state, Prizm can handle up to 367,200 transactions per day - more than nine times the current Bitcoin peak. The Transparent Forging implementation allows transactions to be processed almost instantly, significantly increasing this limit.



Time to confirm transaction

Transaction confirmation time for Bitcoin ranged from 5 to 10 minutes for the most part during 2013. At the end of 2013 after the announcement that Chinese banks would not be allowed to process Bitcoins, the average Bitcoin transaction time increased significantly, to 8-13 minutes, with periodic peaks of 19 minutes. Since then, the confirmation time has shifted from 8 to 10 minutes. However, since several checks (usually six preferred confirmations) are required to complete a Bitcoin transaction, one hour can easily pass before the sale of assets paid by Bitcoin is completed.

Answer of PRIZM

The average block generation time for PZM has historically been 80 seconds, and the average transaction processing time was the same. Transactions are considered safe after ten confirmations, which means that transactions become permanent in less than 14 minutes. The implementation of Transparent Forging allows you to make almost instant transactions, which will further reduce this time.

Problems of centralization

The complexity increase along with the hash rate for Bitcoin have created a high barrier for entry of newcomers, and lower profits for existing mining installations. The incentive to encourage blocks used by Bitcoin has led to the creation of large single-tier installations of specialized mining equipment 44, as well as reliance on a small set of large mining pools 45. This led to the effect of "centralization", where Large volumes of mining are concentrated in the control of a decreasing number of people. This not only creates the kind of power structure that Bitcoin has designed to bypass, but it also presents the real possibility that a single mining operation or pool can gain 51% of the total mining capacity in the 46 network and perform a 51% attack. There are also attacks that require only 25% of the total network hashing power. In early January 2014 GHash.io began to voluntarily reduce the power of its own mining, as it approached the level of 51%. A few days later, the power in the pool decreased to 34% of the total capacity of the network, but the speed immediately began to increase, and in June 2014 again reached dangerous levels.

Answer of PRIZM

The incentives provided by the Proof of Stake algorithm used in Prizm provide a low return on investment of about 0.1%. Since new coins are not generated with each block, there is no additional "reward for mining", which stimulates the joint efforts to create blocks. The data shows that the Prizm network has remained very decentralized since its inception: a large (and growing) number of unique accounts adds blocks to the network.

Proof of Work - maintenance costs

Confirmation of transactions on existing bitcoin and creation of new bitcoins for entering into circulation require enormous computing power, which has to work constantly. This computing power is provided by the so-called mining rigs, which are managed by miners. Bitcoin miners compete with each other to add the next block of transactions to the overall bitcoin chain. This is done by "hashing" - combining all Bitcoin transactions occurring within the last ten minutes, and trying to encrypt them into a block of data, which also coincidentally has a certain number of consecutive zeros in it. Most trial blocks generated by hashing miners do not have this target number of zeros, so they make small changes and try again. A billion attempts to find this "winning" block is called GH, and Mining rig is estimated by how many GH it can perform per second, denoted by GH / sec. The winning miner, who was the first to create a cryptologically correct block of Bitcoin, immediately receives a reward of 25 new bitcoins - the reward at the time of writing was about 15 750 US dollars. This competition among miners with the award is repeated again and again every ten minutes or so. By the beginning of 2014, more than 3,500 bitcoins a day, equal to about $ 2.2 million a day, had been generated. With so much money on the bet, miners supported the rapid arms race in mining rig technology to improve their chances of winning. Initially, bitcoins were mined using a Central processor (CPU), a typical desktop computer. Then in order to increase the speed the chip of a specialized graphics processing unit (GPU) in high-end graphics cards was used. Then the microprocessor with programmable gate array (FPGA) and then the chip of specialized applied integrated circuits (ASIC) were used. ASIC technology is the pinnacle of the line for bitcoin miners, but the arms race continues with the advent of different generations of ASIC chips.

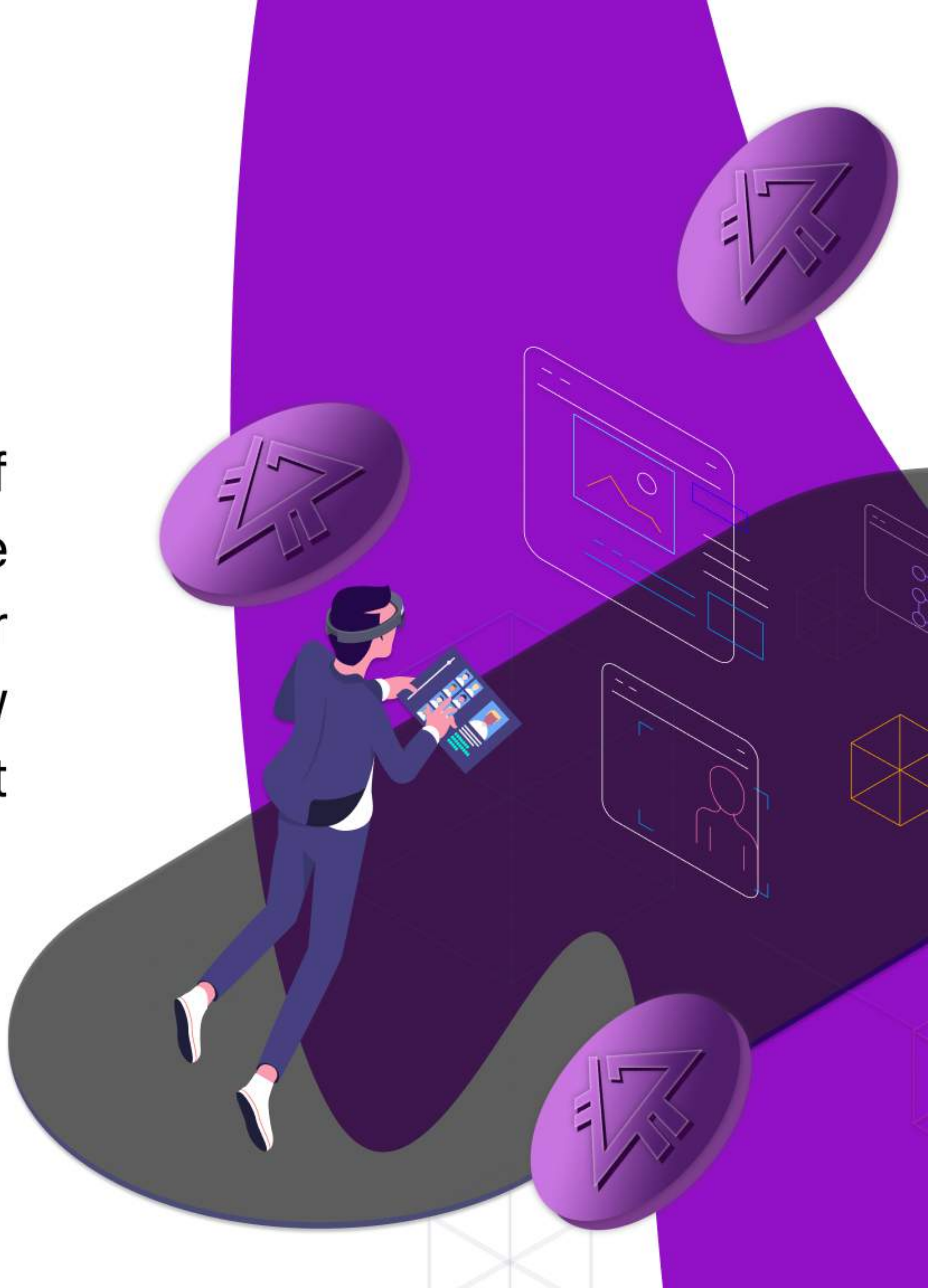
Proof of Work - maintenance costs

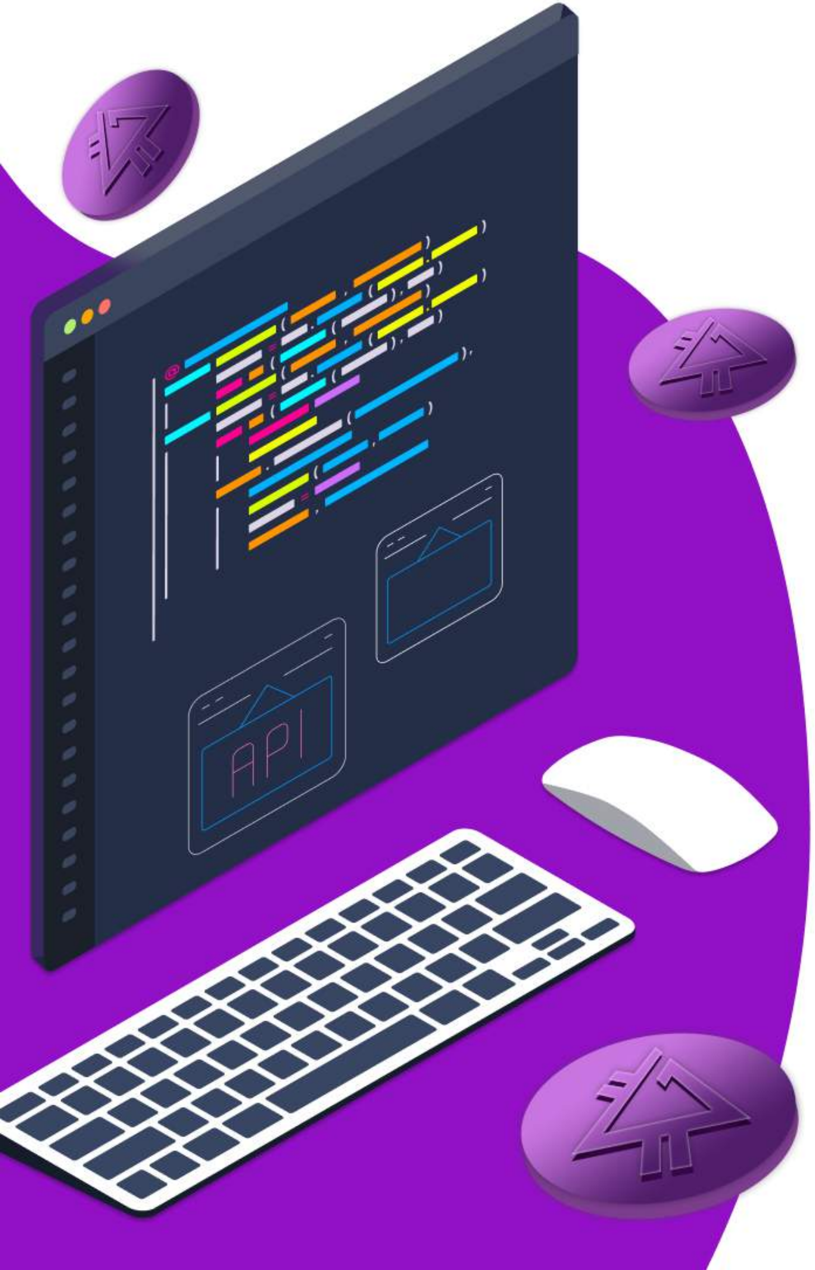
The current generation of ASIC chips is the so-called 28 nm devices based on the size of their microscopic transistors in nanometers. They have been replaced by 20nm ASIC modules by the end of 2014. An example of a new state of the art mining rig would be a 28nm ASIC card "the Monarch" from Butterfly Labs, which is to provide 600GH / sec for an electricity consumption of 350 watts and a price of 2,200 USD. The mining rig infrastructure, which is currently being used to support Bitcoin's current operations, is striking. Bitcoin ASIC is similar to autistic scientists - they can only perform the calculation of a block of bitcoins and nothing more, but they can do it with one calculation at the speeds of a supercomputer. In November 2013, Forbes magazine published an article titled " global bitcoin computing power is 256 times faster than 500 combined supercomputers!". In mid-January 2014, the statistics stored on the site blockchain.info, showed that the continuous support of Bitcoin operations requires a continuous hash rate of about 18 million GH / s. Within one day, this hashing power produced 1.5 trillion trial blocks, which were generated and rejected by Bitcoin's mayonnaise, in search of one - magical 144 blocks that will cover them $ 2.2 million. Almost all Bitcoin calculations are not aimed at fixing the disaster by modeling DNA or searching for radio signals from E. T.; Instead, they are completely wasted. The power and costs associated with this wasteful Bitcoin background support are enormous. If all Bitcoin mining rigs had "Monarch" levels, as described above - and they will not be until they are upgraded - they will represent a pool of 30,000 machines worth more than $ 63 million. It consumes more than 10 megawatts of continuous power during operation and the electricity Bill is more than $ 3.5 million per day. The real figures are much higher for the current, less efficient mining rig pool of machines that actually support Bitcoin today. And these numbers are now going up the exponential growth curve as bitcoin marches from its current one transaction per second to its current maximum of seven transactions per second.

Prizm Solutions

Analysis of the cost and energy efficiency of the Prizm network shows that the entire PRIZM ecosystem can be maintained for about **$ 60,000** per year, which is now almost **2,200 times cheaper** than the cost of operating the Bitcoin network.

PRIZM.SPACE





The cost of POW maintaining relating to the coins' holders

In addition to the huge costs of electricity, there is a hidden fee for the simple storage of bitcoins. For each block found, the one who generates the block receives a reward. At the time of writing, it's approximately a 12,5 BTC reward (for now), which is 10% inflation in the total Bitcoin supply. For every $ 1,000 bitcoin that it belongs to, a person pays $100 for bitcoin in order to "pay" miners for network security.

May the force be with you

PRIZM Integration

PRIZM payment system is the easiest way to receive and send crypto payments.

You can easily integrate PRIZM into your project, online store, exchanger and etc

Step-by-step tutorial:

https://pzm.space/en/pzm-integration/

Prizm payment system integration

To begin working with PRIZM you will need to launch the **network node** (Node) and **API_Servlet.**

Network Node

The software can run on one server as well as on different servers. However it's better to launch it on the one for your convenience.

Firstly, you should launch the node and wait while it syncs. The next step is configuration of the PrizmAPIServlet module.

PrizmCore wallet

https://github.com/prizmspace/PrizmCore#prizmcore-wallet-download-v1103-windows-osx-linux

Easy API Gateway

https://github.com/prizmspace/PrizmCore#easy-api-gateway-prizmapiservlet

Configuration
PrizmAPIServlet

Inside the archive there is a file called

PrizmAPIServlet.properties

After you have filled in the fields, you should launch the servlet through

run-servlet.sh

in the line

passphrase: NONE

instead of NONE you should write the private key of the wallet that will be used by your project.

in the line

sendkey: NONE

instead of NONE you should write the password (it will be used by the function of coin sending as an additional protection from unauthorized transactions).

The example of implementation in PHP

The description of work with receiving and sending coins, with examples of ready-made functions and description of the principles of work. The Mysql database is used to store the transaction list, there is a dump of the storage table below, along with examples of code to work with the table (if you apply QueryBuilder, it won't be a problem).

The main principle of work

There is a script in the Cron-task that makes a request to the servlet every 2-5 minutes so it could receive new transactions on the wallet of the store. Having received the list of transactions, you should save them to the local database. If there are no operations in the database, you should run the command without any parameter. However if you wish to receive new transactions, you should send the number of the last transaction that you have as a parameter.

The example of the function:

```
<?php
function historyPZM($last_id = 0)
{
 if ($last_id) {
  $url = 'http://localhost:8888/history?fromid=' . $last_id;
 } else {
  $url = 'http://localhost:8888/history';
 }
 $page = '';
 $result = get_web_page($url);
 if (($result['errno'] != 0) || ($result['http_code'] != 200)) {
  $error = $result['errmsg'];
 } else {
  $page = $result['content'];
 }
 $array_new = array();
 $xcmorewrite = explode("\n", str_replace("\r", '', $page));
 foreach ($xcmorewrite as $value) {
  if ($value) {
   $array_new[] = explode(";", $value);
  }
 }
 return $array_new;
}
?>
```

The function for retrieving page content:

```php
<?php

function get_web_page($url)
{
 $uagent = "Opera/9.80 (Windows NT 6.1; WOW64) Presto/2.12.388 Version/12.14";
 $ch = curl_init($url);
           curl_setopt($ch, CURLOPT_RETURNTRANSFER, 1); // recovers the web page
          curl_setopt($ch, CURLOPT_HEADER, 0); // doesn’t recover headers
          curl_setopt($ch, CURLOPT_FOLLOWLOCATION, 1); // follows redirects
          curl_setopt($ch, CURLOPT_ENCODING, ""); // handles all encodings
         curl_setopt($ch, CURLOPT_USERAGENT, $uagent); // useragent
         curl_setopt($ch, CURLOPT_CONNECTTIMEOUT, 20); // time-out of the connection
         curl_setopt($ch, CURLOPT_TIMEOUT, 20); // time-out of the answer
         curl_setopt($ch, CURLOPT_MAXREDIRS, 2); // stops after the 10th redirect

 $content = curl_exec($ch);
 $err = curl_errno($ch);
 $errmsg = curl_error($ch);
 $header = curl_getinfo($ch);
 curl_close($ch);

 $header['errno'] = $err;
 $header['errmsg'] = $errmsg;
 $header['content'] = $content;
 return $header;
}

?>
```

The function for retrieving page content:

You can test it through the console, for example: curl http://localhost:8888/history

The example of the Cron-task handler script for receiving new transactions and the table structure

```
CREATE TABLE `pzm_history` (
  `id` bigint(20) NOT NULL,
  `tarif_id` int(1) NOT NULL,
  `tr_id` varchar(255) NOT NULL,
  `tr_date` varchar(255) NOT NULL,
  `tr_timestamp` int(11) NOT NULL,
  `pzm` varchar(50) NOT NULL,
  `summa` decimal(16,2) NOT NULL,
  `mess` varchar(255) NOT NULL,
  `status` int(1) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
```

** All necessary keys and autoincrement for ID should be added to the table

Handler:

In this example you receive the list of new transactions that should be saved to the local database.

Therefore, you keep a history of all transactions on the wallet and in the future you will search for them in our local database using key data.

```
<?php
$nomer = getLastPrmHistory();
$historys = historyPZM($nomer);

foreach ($historys as $item) {
  if ($item['0'] != "No transactions!") {

// this line adds data to the ‘pzm_history’ table using
INSERT IGNORE

PzmHistory::find()->insertIgnore([
      'tr_id' => $item['0'],
      'tr_date' => $item['1'],
      'tr_timestamp' => $item['2'],
      'pzm' => $item['3'],
      'summa' => $item['4'],
      'mess' => $item['5'],
      'status' => 0
    ]);
  }
}
```

```
function getLastPrmHistory()
{
//  this line searches for the last row in the table to get the last ID of the transactions which are in the table

if (!empty($pzmHistory = PzmHistory::find()->orderBy('id', "DESC")->first())) {
  return $pzmHistory->tr_id;
 };
 return 0;
}

?>
```

Your project must work with the same Prizm Wallet, that is why all clients will be given the same requisites to replenish the internal account and the same hash ID of the operation. Be sure to inform the client that he must make a transaction strictly on the requisites indicating the hash identifier in the payment comment.

Thus, there should be another process that will analyze new incoming transactions and deposit coins to the internal account if the payment comment has a hash identifier of the client. Also you need to make a separate **"I PAID"** button for the client which could search and record new transactions for this user after making click on it.

Secondary functions and functions of coin sending

Getting public key for the wallet (works only for activated wallets having balance).

```php
<?php

 function destinationPZM($pzm)
   {
      $url = 'http://localhost:8888/publickey?destination=' . $pzm;
      $page = '';
      $result = get_web_page($url);
      if (($result['errno'] != 0) || ($result['http_code'] != 200)) {
         $error = $result['errmsg'];
         return '';
      } else {
         $page = $result['content'];
         $haystack = "Public key absent";
         $haystack2 = "Send error!";
         $pos = strripos($page, $haystack);
         $pos2 = strripos($page, $haystack2);
         if ($pos === false AND $pos2 === false) {
            $xcmorewrite = explode(' ', $page);
            $page = trim($xcmorewrite[0]);
            return $page;
         } else {
            return '';
         }
      }
      return $page;
   }

?>
```

Receiving current balance of the wallet:

```
<?php

 function getBalancePZM($pzm)
    {
        $ip = '*******';  // пример 192.168.1.1:9976  with port
  $url = 'http://'.$ip.'/prizm?requestType=getAccount&account=' . $pzm;
        $page = '';
        $result = get_web_page($url);
  //print_r($result); die;
        if (($result['errno'] != 0) || ($result['http_code'] != 200)) {
            $error = $result['errmsg'];
            return '';
        } else {
            $page = $result['content'];
            $page = json_decode($page, true);
            if ( isset($page['balanceNQT']) ) {
              return $page['balanceNQT'] / 100;
            } else {
              return 0;
            }
        }
    }

?>
```

Method of coin sending:

```
<?php

public function payPZM($summa, $pzm, $public_key, $text)
{
 $p2 = SENDKEY;   // this is the password that you specified during setup
 $return = false;
 $url = 'http://localhost:8888/send?sendkey=' . $p2 . '&amount=' . $summa .
'&comment=' . urlencode($text) . '&destination=' . $pzm . '&publickey=' . $public_key;
 $page = '';
 $result = get_web_page($url);

 if (($result['errno'] != 0) || ($result['http_code'] != 200)) {
  $error = $result['errmsg'];
 } else {
  $page = $result['content'];
 }

 if (preg_match('/^\+?\d+$/', $page)) {
  $return = true;
 } else {
  $return = false;
 }
 return $return;
}

?>
```

WHITEPAPER

PRIZM

The initial concept of digital currency

Prizm Whitepaper Revision

June, 2020

PZM.SPACE