# Whitepaper
# dPNM ecosystem

v 1.1

The dPNM token is a 100% BUSD backed token with an always increasing price on the Binance Smart Chain blockchain.
The project consists of 3 immutable smart contracts.

# dPNM

This contract records, issues and burns dPNM tokens.
Allows you to activate new users and manages income limits, as well as accrues marketing payments.

## Prestart

At the time of the start of the project, the prestart mode is activated.
In this mode, users are activated without the possibility of buying and selling a dPNM token. The prestart mode will be deactivated when 10,000 participants are reached, or after 3 weeks.

## User activation

User activation is a one-time event. The activation fee is 10 BUSD.

When a user is activated, a card is created for him in the dPNM contract and the user is located in the Phenomenal Tree.

At the time of the first dPNM purchase, a person is compensated for activation in the amount of 10 GWT.

**One address cannot be reactivated** but one person can **create multiple addresses** to go through the activation at each address and positioning in the tree.

To be able to buy/sell a dPNM token **one-time activation required**. Activation allows you to buy / sell a dPNM token without time limits.

Upon activation, the user is assigned **active tree status for 30 days**, this allows you to get **marketing** rewards.

## dPNM purchase

The dPNM token purchase is made using **BUSD**.
**Terms of dPNM purchase:**
1. You cannot buy a dPNM token for less than **20 BUSD** per transaction
2. The maximum limit for the dPNM token purchase per day starts at 50 BUSD and will continue to increase as the token's liquidity pool in BUSD increases
3. When purchasing dPNM 20 % **fee** is charged
   a. 10% - distributed to the tree for **marketing rewards**
   b. 10% - remains in **liquidity pool** of the contract, thereby increasing the price of the dPNM token
4. All fees paid when buying a dPNM token are **compensated by GWT token** in a ratio of 1:1

When buying a dPNM token, a person is credited with an income **limit in a ratio of 1:2.**
*For example, when buying tokens for 50 BUSD, an income limit of 100 BUSD is credited.*

# Daily dPNM purchase limit calculation for the user

This limit was created to **exclude token price manipulation** by big players. The limit allows you to ensure a gradual increase in price while developing the community.

The maximum limit for the token daily purchase cannot be less than 50 BUSD. As the BUSD liquidity pool on the smart contract increases, dPNM will **automatically increase the limit** for the token daily purchase.

The formula for calculating the maximum daily buy limit is:

$b -$ $max\ buy\ limit$
$L -$ $BUSD\ liquidity\ pool$
$b = \frac{L}{1000}$

*For example, if the token liquidity pool is 1,000,000 BUSD, the limit will be 1,000,000 / 1000 = 1,000 BUSD*

The user can **use the limit in parts**. For example, buy a token for 20 BUSD, and then buy another 30 BUSD within 24 hours.
After 24 hours from the first purchase of 20 BUSD **the limit is reset** to 50 BUSD again (or 0.1% of the pool size).

It is possible to **sell any amount of token** with its subsequent purchase. During **48 hours** from the moment of sale, the amount of the limit will be increased by the amount of the sale.
***Example***: *the user's purchase limit is 50 BUSD, the user sells a token for 100 BUSD, respectively, within 48 hours he will be able to buy a token for 50+100=150 BUSD*

**The formula for calculating the final purchase limit for a user is:**

$p -$ $user\ dPNM\ purchase\ amount\ with\ BUSD$
$b -$ $max\ buy\ limit,\ calculated\ from\ liquidity\ pool\ size$
$B -$ $user\ BUSD\ amount\ of\ dPNM\ purchased\ in\ last\ 24\ hours$
$S -$ $user\ BUSD\ amount\ of\ dPNM\ sold\ in\ last\ 48\ hours$
$p = b - B + s$

# Income limit

This limit was created to **increase community activity**, get rid of long-term holders and provide **maximum** efficient algorithmically calculated **price increase** of dPNM token.

The income limit is calculated during **each** dPNM token purchase by a user in **a ratio of 1:2**. *When buying a token for 50 BUSD, 100 BUSD of the income limit will be credited.*

This limit is **burned on dPNM sales**.
*For example, when selling a token for 70 BUSD, the income limit is reduced from 100 BUSD to 30 BUSD.*

When re-purchasing tokens, the income limit is **added to the remainder**.
*For example, with a further purchase of tokens for 50 BUSD, the income limit will be 30 + 100 = 130 BUSD*

**The value of your tokens is calculated according to the formula:**

$V\ -\ user\ dPNM\ token\ value$
$T\ -\ user\ total\ dPNM\ tokens\ amount$
$P\ -\ 1\ dPNM\ token\ price\ in\ BUSD$
$V = T \times P$

There are **two rules for burning the income limit** and they depend on the value of your dPNM tokens:

1. If the value of your tokens is **less** than your income limit, then there is a **1:1** burn

   *Example: You have 40 dPNM tokens, price of 1 token = 2 BUSD.*
   *Income limit = 100 BUSD.*
   *Token value = 40*2 = 80 BUSD*
   *Selling 20 dPNM tokens will burn 40 BUSD of income limit*

2. In case the value of your tokens is **greater** than your income limit, then the income limit is burned **proportionately to token amount**

   *Example: You have 40 dPNM tokens, price of 1 token = 3 BUSD.*
   *Income limit = 100 BUSD.*
   *Token value = 40*3 = 120 BUSD, which is more than the income limit*
   *50 BUSD = 50% of your income limit.*
   *When selling dPNM tokens for 50 BUSD, 50% of dPNM tokens and 50 BUSD of income limit will be burned*

Proportional burning of the tokens allows for a constant circulation of tokens, motivating participants not to keep tokens on their balance for years, since the growth of the ecosystem is possible only with the activity of all participants.

It is possible to buy an additional income limit.

## Buying an additional income limit

The purchase of an additional income limit occurs using the GWT token.
When you purchase, you get 1.25 BUSD income limit per 1 GWT.

The amount of additional income limit available for purchase is limited to **10% of the maximum income limit for all time**.

**This limitation is calculated by the formula:**

$I\ -\ maximum\ possible\ income\ limit$
$D\ -\ total\ cost\ of\ all\ user\ dPNM\ purchases\ for\ all\ time\ in\ BUSD$

$I = \frac{D}{100} \times 220$

*Example:*
*The user bought dPNM tokens for 50 BUSD, the income limit is 100 BUSD.*
*The user bought dPNM tokens for 70 BUSD, the income limit is 100+140 BUSD.*
*The user bought dPNM tokens for 100 BUSD, the income limit is 100+140+200 BUSD.*
*Next, the user decided to buy an income limit for GWT*
*the maximum income limit is = (50+70+100)/100*220 = 484 BUSD*

*The current income limit is 440 BUSD, the maximum is 484 BUSD. The user can buy an additional 44 BUSD income limit.*

To purchase an income limit **dPNM token one-time purchase is required**.

The fee for purchasing the income limit is 2 BUSD:
1. 1 BUSD is sent to BUSD **liquidity** of dPNM token, increasing its price
2. 1 BUSD is sent to the fee collector, used for **maintaining and developing the project**

# Extending the tree life

At the time of activation, a position in the tree is activated for the user. **Activation is a one-time event**. The tree is considered active for **30 days** from the moment of activation.

The active tree allows you to receive **marketing rewards** from tree payments by other users and their dPNM token purchase.
The ability to buy and **sell dPNM does not depend on whether** the tree is active or not, a one-time activation is enough.

It is possible to extend the activity of the tree.
The cost of extending the activity of the tree is **10 BUSD** and it increases tree activity by additional **30 days**.
Period of paid tree activity **cannot exceed 90 days**.
*Example:*
*The user has been activated in the project and the period of activity of his tree is 30 days.*
*The user pays to extend the activation of the tree for another 30 days and now the tree will be active for the next 60 days.*
*The user once again pays for the activity of the tree and receives 90 days.*
*Subsequent payment for the tree is not possible due to exceeding the limit of 90 days.*
*After 30 days, the user will be able to pay for tree activity again.*

**The payment of 10 BUSD per tree activity is distributed as follows:**
1. 5 BUSD distributed in **marketing** payments
2. 1 BUSD is sent to dPNM token **liquidity in BUSD**, increasing its price
3. 4 BUSD sent to commission collector, used for **maintaining and developing the project**

The payment amount of 10 BUSD is fixed.

# dPNM Token Sale

You can sell the token at **any moment**. At the time of the token sale, the dPNM token is **burned** and the person gets an **instant transfer of BUSD from the token liquidity pool**.

To sell a token **income limit is required**. You can sell for any amount less than or equal to the income limit and less than or equal to the value of the tokens.

Income limit **decreases** for the amount of the token sale, or **proportionately**, if the income limit is less than the value of the token.

**When selling a token, a commission of 10% is charged, commission distribution:**

1. 5% is sent to dPNM token **liquidity in BUSD**, increasing its price
2. 5% sent to commission collector, used for **maintaining and developing of the project**

All commission paid is compensated by the token **GWT** in a ratio of **1:1**

# Phenomenal Tree

The tree of project participants is implemented using a separate smart contract.

The purposes of this tree are: **allow participants to get paid for referrals,** creating additional sources of income, and **increase the price of the dPNM token** thanks to tree activity payments.

The tree is made up of **3 branches** and has a depth of **10 levels for each user**.

| # | | |
|---|---|---|
| 1 | | 3 |
| 2 | | 9 |
| 3 | | 27 |
| 4 | | 81 |
| 5 | | 243 |
| 6 | | 729 |
| 7 | | 2 187 |
| 8 | | 6 561 |
| 9 | | 19 683 |
| 10 | | 59 049 |
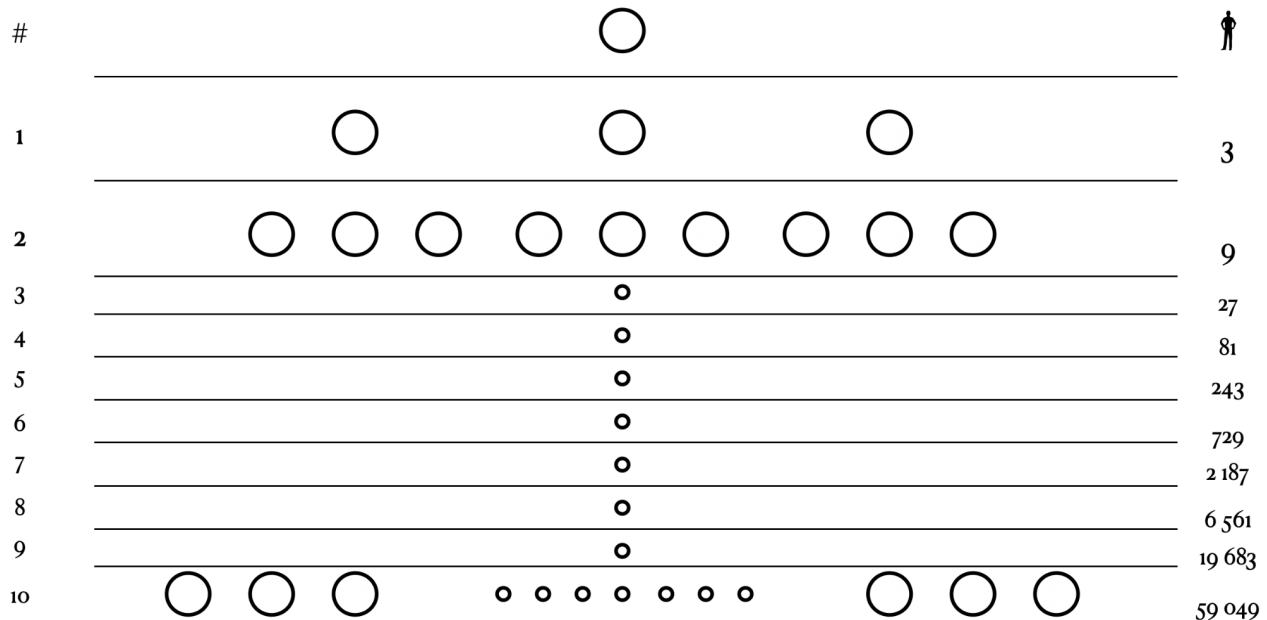
At the moment of activation, the user takes his position in the tree **taking into account the address of the user who invited him**.

Tree positioning is **impossible without the address of the inviter**.

# Finding a position for user in the tree

The position in the tree starts from the **inviter** address. Smart contract checks positions **in each branch of the tree from left to right**.

Search position algorithm:

1. Check 3 positions on the **first level** of **inviter** address starting **from left to right**
2. If **free position** is found then **use** it
3. If **no free positions** then **select branch that has lowest number of positions** filled 10 levels deep from the inviter's address
4. Check **3 positions on the selected branch** starting from left to right. **Proceed to step 2**

In the graphic illustration below, you can see where the position will be located, highlighted in orange in relation to the blue position from which the search begins. Already filled positions are highlighted in green
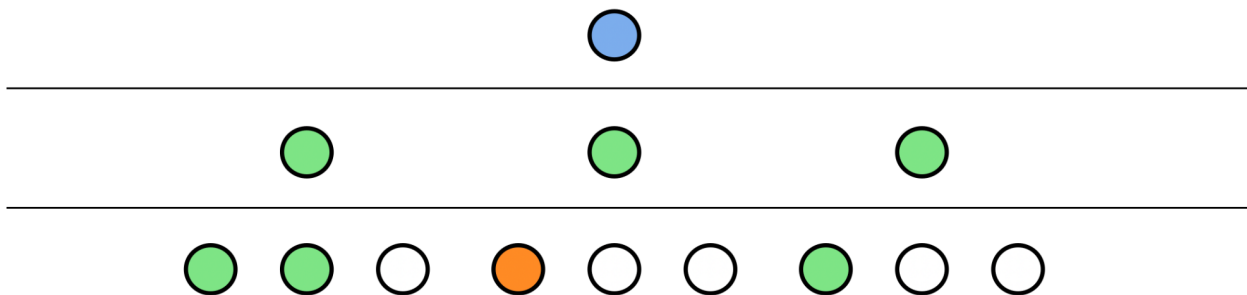


This algorithm allows to completely fill the user tree, which can accommodate **88 572 addresses**

With this algorithm for arranging addresses in the tree **so-called overflows are possible** (this is a situation where the invited addresses from **superior** tree members will be located in **your** tree). This allows you to have marketing rewards without **making any recommendations**, only **active tree is required**.

# Marketing rewards

In the case of positioned addresses in your tree, you claim to receive rewards for the activity of this address.

There are **2 types of rewards** in a project.

## Reward for paying tree activity

Each time a user pays 10 BUSD to keep the tree active, a **5 BUSD in marketing** rewards distribution occurs.

Distribution occurs according to the location of addresses in the tree on **10 levels up from the address that paid the tree**.
You can see the amount of your reward in the table below, the level is indicated in relation to the reward **recipient**.
***Example***: *if at the 8th level of your tree there was a payment for the activity of the tree, then you instantly receive 0.8 BUSD to your address.*

The only **condition** to receive these rewards is an **active** tree (payment 10 BUSD for 30 days)

| Level | Your reward |
|:---:|:---:|
| 1 | 0.1 BUSD |
| 2 | 0.1 BUSD |
| 3 | 0.1 BUSD |
| 4 | 0.5 BUSD |
| 5 | 0.5 BUSD |
| 6 | 0.5 BUSD |
| 7 | 0.8 BUSD |
| 8 | 0.8 BUSD |
| 9 | 0.8 BUSD |
| 10 | 0.8 BUSD |

## Reward for purchasing a dPNM token

In case of purchasing a dPNM token in your tree, you can also receive a **reward depending on the size of the purchase**.

These rewards are generated in the same way **from 10 tree levels below you**.

**To receive this rewards, you should have:**
1. Active tree
2. **Unlock** corresponding tree **level**

Tree levels are unlocked **automatically** and depend on the amount of **turnover** of the addresses in your tree on **10 levels deep**. Turnover accrues on each dPNM buy.
*For example, purchasing 50 BUSD by dPNM address at level 7 of your tree increases your turnover by 50 BUSD.*

Levels **1-3 are unlocked** by default and do not require turnover.

You can see the amount of your bonus in the table below.

| Level | Bonus % from dPNM purchase | Required turnover of BUSD from 10 levels to unlock a level |
|-------|----------------------------|-----------------------------------------------------------|
| 1 | 0,2 | 0 |
| 2 | 0,2 | 0 |
| 3 | 0,2 | 0 |
| 4 | 1 | 40 000 |
| 5 | 1 | 100 000 |
| 6 | 1 | 200 000 |
| 7 | 1,6 | 500 000 |
| 8 | 1,6 | 1 000 000 |
| 9 | 1,6 | 2 000 000 |
| 10 | 1,6 | 5 000 000 |

*Example:*
*An address in your tree at level 3 buys dPNM for 50 BUSD, your reward will be 50\*0.2% = 0.1 BUSD Reward is **instantly** credited to your address and you get 50 BUSD of **turnover**.*

In a situation where you have a lack **of turnover** to open a new level, you can **buy additional one using GWT.**

In cases where a purchase of a dPNM token occurs in your tree at **locked** level, you **get turnover**, but **you don't get a reward.**

# Buying turnover using GWT

You can buy the required amount of turnover (without limits) using GWT. Per **1 GWT you will receive 200 BUSD** of turnover.

If the required amount of turnover is reached to open the level, it will be **unlocked instantly**.

The commission for the purchase of turnover is 2 BUSD:
1. 1 BUSD is sent to dPNM token **liquidity in BUSD**, increasing its price
2. 1 BUSD is sent to the commission collector, used for **maintaining and developing the project**

# Grow Token (GWT)

The GWT token stands for **Grow Token** and it allows you to get maximum **benefit** from participating in the project.

The token allows you to buy turnover and income limit, as well as unlock tree levels faster.

**The GWT token is credited every time you make:**
1. **dPNM purchase** - GWT accrual in the amount of the paid commission 20%
2. **dPNM sale** - GWT accrual in the amount of the paid commission 10%
3. **activation payment and tree activity renewal** - accrual of **10 GWT**

The GWT token has been created in accordance with the ERC20 standard and can be transferred to any address.

**When transferring the GWT token, a commission of 2 BUSD is paid:**
1. 1 BUSD is sent to dPNM token **liquidity in BUSD**, increasing its price
2. 1 BUSD is sent to the commission collector, used for **maintaining and developing the project**

## GWT token staking

The GWT token can be staked. Staking allows one to earn a GWT **token,** locking it for a while. This helps to **balance project tokenomics** and **improve dPNM growth rate**.

When staking a GWT token, it **is locked for a selected period of time** and **daily fixed** rewards are credited with GWT token

You can see GWT staking conditions in the table below.

| Lock period | Daily yield in GWT token |
| --- | --- |
| 14 days | 0.1% |
| 30 days | 0.2% |
| 60 days | 0.25% |
| 90 days | 0.3% |
| 180 days | 0.4% |
| 365 days | 0.5% |

You can withdraw your reward from staking **at any time**.
The main locked amount of the GWT token will only be available **after the end of the lock period**. Early return of locked tokens from staking is not possible.

You can stake **only one period** of lock at the same time.

***Example***:
*The user has staked 100 GWT for a period of 90 days.*
*The daily reward amount is 100 * 0.25% = 0.25 GWT*

*After 10 days, the user claimed 2.5 GWT*
*After another 20 days, the user claimed 5 GWT*
*After 60 days, the user has collected 15 GWT rewards + 100 GWT of unlocked tokens*
*Further, if desired, he will again be able to start staking for 90 days.*

# How the price of the dPNM token is growing

The dPNM token is an algorithmic deflationary token, which **always** grows in price. The principle of its growth can be explained in one sentence

*"At **each** operation with the token, the token becomes **less,** than its **collateral**"*

**The formula for token price calculating is as follows:**

$P\ -\ 1\ dPNM\ price\ in\ BUSD$
$L\ -\ dPNM\ liquidity\ pool\ in\ BUSD$
$T\ -\ total\ dPNM\ tokens\ supply\ in\ circulation$
$P\ =\ \frac{L}{T}$

*For example, if there are 20 dPNM tokens and 10 BUSD in collateral, then the price of 1 dPNM = 10/20 = 0.5 BUSD*

*Further, let's say we remove 1 dPNM token and at the same time remove 0.4 BUSD, then the price of 1 dPNM = 9.6 BUSD / 19 dPNM = 0.505 BUSD. As we see **price has increased by 1%***

Accordingly, every time when **reducing the** dPNM tokens **number slower than the amount of BUSD collateral**, algorithmically **the price for 1 dPNM token increases**, while always remaining **100% secured by BUSD**

**Which operations in the dPNM ecosystem increase the token price:**
1. Each **token BUY** upon payment of 20% commission creates a token for 80% of the purchase amount, while 90% of the purchase amount goes as security (another 10% goes to marketing payments)
2. Each dPNM **token SALE** upon payment of 10% commission burns 100% of the dPNM token, while reducing liquidity by 95% (5% goes to the development of the project)
3. Each **tree activity payment** at 10 BUSD sends 1 BUSD to token liquidity without creating new tokens
4. Every **GWT token transfer** transfers 1 BUSD to dPNM token liquidity without creating new tokens
5. Each **income limit purchase** transfers 1 BUSD to dPNM token liquidity without creating new tokens
6. Each **turnover purchase** transfers 1 BUSD to dPNM token liquidity without creating new tokens

There are also partners as an **exchange** and additional **products** and **services** which add liquidity to the dPNM token for its growth. The number of services may be added in the future which will bring even more liquidity to dPNM.

In this way, **with sufficient community activity,** which simply uses the ecosystem, actions that **increase the price of the token** are constantly taking place. The BUSD liquidity pool is filled with liquidity from **more than 10 sources**, which guarantees its growth under any circumstances.

Imagine that **every** community member wants to **sell** own dPNM tokens, the smart contract will ensure every sale **at 100%** and **there will still be funds left** in the liquidity balance.

**It is important to understand some points:**
1. **Less** tokens **liquidity** in BUSD, faster **its price rises**. Thus, the more people sell the token, the

easier its price increases.
2. Token **price can't stop growing ever**, only growth speed can decrease
3. Token BUSD **collateral can never be less than 100%** regardless of the number of sellers

Precisely for **balancing** the system introduced limits on the maximum daily token purchase by users. The system is as balanced and healthy as possible when there are many participants who share tokens among themselves.

**At the same time, no one has access to the liquidity of the token in BUSD, except for the smart contract itself.**

# Who manages the project?

This project has been created by a group of enthusiasts and blockchain technology and smart contracts were chosen for its implementation. There is no legal entity, company, fund or bank behind the project. Everything is coordinated in a decentralized way.

A smart contract is an agreement, or a set of conditions, that is recorded on the blockchain using cryptographic keys, and at the time of publication, they can no longer be changed.

Some smart contracts are renewable, which means that the owner of the cryptographic key can change their original terms.

In our project, it was decided to use non-updatable smart contracts. This means that after a smart contract is published on the blockchain network, no one else ever has the opportunity to change the conditions, even those who have a cryptographic key.

This gives confidence to the project participants that becoming part of the community, the conditions on which they became a participant will not change over time. It also ensures that the **only one who has access to providing liquidity to assets is the smart contract**.

As a consequence, it must be understood that this project belongs exclusively to the community, which is **collectively responsible for the success of the development of the project** - this is called **DAO** (Decentralized Autonomous Organization).

Over time, new smart contracts will be added to the project, which will allow each participant to vote and decide which decisions to make and which not.

To ensure the security of the smart contract code and its compliance with the stated conditions, the smart contract has been **audited by a third-party company**, which is the market leader in smart contract audits for the blockchain.

If you wish, you can conduct your own audit of the smart contract code, since the code is **public** and available for viewing to every person in the world.

The immutability of the conditions also imposes some inconveniences, in particular, sometimes it is required to change some conditions in order to direct the development of the project in a more efficient direction.

For easy adjustments in the direction of development, some conditions are created that can be changed to a **limited** extent by participants who have the required cryptographic key.

# Conditions subject to change

Parameters that can change to adjust the direction of the project development and more stable and predictable growth.

| Condition | Description | Range |
|---|---|---|
| buyFeeToLiquidity | Percentage of the **dPNM purchase amount,** which goes into liquidity | Default: 10%<br>Variable: 0-10% |
| sellFeeToLiquidity | Percentage of the **dPNM sale amount,** which goes into liquidity | Default: 5%<br>Variable: 0-5% |
| gwtTransFeeLiquidity | Commission that goes to liquidity at **purchase of income limit, turnover, GWT transfer** | Default: 1 BUSD<br>Variable: 0-2 BUSD |
| turnoverForOneGWT | Amount of **turnover** credited **for 1 GWT** | Default: 200 BUSD<br>Variable: 200-250 BUSD |
| dPNMbuyTurnoverIncrease | **Turnover** increase when **buying** dPNM | Default: 0%<br>Variable: 0-25% |
| dPNMsellTurnoverIncrease | **Turnover** accrue from dPNM **sale** amount | Default: 0%<br>Variable: 0-25% |
| gwtBuyIncrease | Increase in accrued **GWT** from **dPNM purchase** | Default: 0%<br>Variable: 0-25% |
| gwtSellIncrease | Increase in accrued **GWT** from **dPNM sale** | Default: 0%<br>Variable: 0-25% |
| gwtForTreeActivation | **GWT deposited** for **activation** and **tree** payment | Default: 10 GWT<br>Variable: 5-15 GWT |
| earnLimitDepositedPerc | **Turnover** accrued from the **dPNM purchase amount** | Default: 200%<br>Variable: 200-250% |
| maxDailyBuy | Maximum **daily purchase** of dPNM 0 - calculated as 0.1% of liquidity | Default: 0<br>Variable: more than 50 BUSD |
| treePaymentPeriod | Amount of activity **days** accrued when **paying for tree** | Default: 30<br>Variable: 30-60 |
| treeMaxPaymentPeriod | **Maximum** number of **tree activity days** | Default: 90<br>Variable: 90-180 |

| isLocked | Blocking the purchase of dPNM, activation, tree extension | Default: off Variable: On |
|---|---|---|

All other conditions are written in the smart contract code are immutable

# Prospects for the project development

As the community develops and grows, a number of other products are planned to be launched. It is important to understand that all products will use the existing tree structure. All proposals for the introduction of new products will be voted on by the community.

# Participation Risks and Frequently Asked Questions

It is very important when making decisions to make your **personal analysis** of the project and evaluate your personal financial prospects.

Implementation using a smart contract gives confidence that the conditions will not change, but this does not mean that the price of the token will rise in the absence of activity.

The growth of the token occurs according to the algorithmic principle. The price of the token does not grow just because time passes, the token is locked in a smart contract and there are no traders and other legends about the increase in its price in the project.

For example, you can take bitcoin, it is also a token that has value in proportion to its development and acceptance by the community. If bitcoin had 5 holders, then it would not have much value.

The token collateral consists of BUSD, which is backed by the Binance exchange.

One of the best companies was hired to audit the smart contract to minimize possible risks, but nevertheless, each risk lies individually on each member of the community.

# FAQ

### How to start?

In order to get started you need:
1. Create a crypto wallet, we recommend to use [MetaMask](MetaMask) with Binance smart chain network.
2. Transfer the required amount of BUSD to the address in the wallet, you can use the Binance exchange
3. Transfer a small amount of BNB, this is a token that will allow you to pay a fee for signing transactions on the blockchain
4. Go to the dPNM token website at [www.dpnmdefi.com](www.dpnmdefi.com) and complete the activation for 10 BUSD
5. Buy the desired number of dPNM tokens with BUSD

### What is the guarantee of the dPNM token price growth?

An increase in the price of the token occurs from each operation due to commissions that fill the BUSD liquidity pool or burn the token. The price of the token can only grow. The growth rate of the token price depends solely on the size and activity of the community.

### Can the project close?

Once a smart contract has been published in the blockchain, it is impossible to delete or change its terms. This means that it is not possible to delete/close the project.

### Can the token security disappear?

The provision of the token in BUSD is on the balance of the smart contract and the actions that can be performed with it are unchanged after the publication of the smart contract on the blockchain. Accordingly, no one can transfer it, withdraw it, etc.
The only option to reduce the collateral – sale of the dPNM token by participants, but the collateral for each token is always 100%.

### Can I lose my invited people?

Once an address is placed in the tree, it is no longer possible to rearrange/delete/change it. Accordingly, unlike traditional projects, you can be sure that your structure will always be yours.

### Can I be blocked or terminated?

There is no functionality in the smart contract code that allows to limit the rights of an address or change the balances of its assets. The code is public and if you wish, you can conduct a personal audit.

### What happens if the site is down?

The site is just a frontend for a more convenient interaction with the contract, so if the site is closed, you can always interact directly with the blockchain, for example, through the site [https://bscscan.com/](https://bscscan.com/) and perform any actions, sell/buy dPNM, activate/pay for the tree, etc.

### Can blockchain disappear?

The blockchain supports a number of decentralized validators. In order for the blockchain to disappear,

all validators must stop supporting the network, which is almost impossible. We recommend you to study what blockchain is in order to understand the likelihood of such an event occurring.

### How do I buy/sell dPNM?

You can make any interaction with the smart contract using the official website of the project, or by directly interacting with blockchain. The dPNM token cannot be transferred to another user, so there are no P2P sales.

### How do I buy/sell BUSD?

This can be done on the Binance exchange, on P2P platforms, or when interacting with other users.

### How do I buy/sell GWT?

GWT cannot be bought/sold directly. This token does not have a collateral pool and its actual price/value is measured in its need. This token is necessary for all project participants to expand the income limit or increase marketing income.

You receive a GWT token as compensation for all commissions spent. If you wish, you can transfer GWT and sell it to another user. Similarly, you can buy it from another user (if there is someone who wants to sell it). Please note that the GWT transfer fee is 2 BUSD.

### How can I restore access to my address in case of loss/theft?

To create a crypto wallet with different addresses, we recommend using MetaMask.
At the time of creating a wallet, you need to remember the secret phrase from a set of words. If you lose access to addresses, knowing the secret phrase, you can restore all your addresses.
If the secret phrase is lost, the addresses cannot be restored. Accordingly, it is impossible to access your assets in the smart contract that you connected using this wallet.
Please **pay maximum attention to ensuring the safety and security of your secret phrase** from the wallet. Please note that if your secret phrase is stolen, then the attacker will have access to all your assets and it will be impossible to restrict his access

### Can my balances disappear?

Noone has access to your balances other than the smart contract itself. Accordingly, the only way to change balances is to sign transactions in your wallet. A smart contract can store your balances indefinitely.

### Can I create multiple addresses?

Yes, you can create any number of addresses in your crypto wallet. Accordingly, you can interact with the smart contract on behalf of any address, which will act as a separate account. For each address, if desired, you can go through activation and own assets.