MESH+

– Sustainable –

– Self-Governed –

– Tokenomics–

Dr. Oleksandr Letychevsky
Prof. Volodymyr Peshanenko
Timothy Kravchunovsky
Mircea-Teodor Raduta

Revision 4

Preface

Today, the whole world experiences the beginning of a true mass adoption of blockchain technologies, as new projects and services evolve by the day, creating a brand new ecosystem of crypto-related platforms and products that utilize cryptocurrencies and tokens.

Nevertheless, we haven't ventured far from the Stone Age of the Blockchain-ruled world with just a few real benchmarks or examples of long-lasting, self-sustaining token economies. The main reason is that creation of a self-governing token economy is highly difficult and comprehensive task that requires not just effort, but deep knowledge and foresight.

It's easy to see that most creators, even if they have a profound whitepaper and appear to have quite a thoughtful token economy, intuitively build pyramid-scheme-like structures with a lack of real token utility and stakeholders motivation to hold, and most importantly - **use** their tokens in daily activities within the platform. As a result we have a majority of projects that will never reach a system balance and become much too dependent on speculation in the shadow of "whale" investors.

Is there a simple and efficient way to help the blockchain project creators avoid mistakes and create self-sustainable token economies right at the stage of the white-paper and MVP development?

The key is math and formal methods. Modeling is a mandatory activity that shall be provided during the conception of Tokenomics. Algebraic modeling may assist in building and proving the economy equilibrium, as well as analyse different undesirable properties like centralization, or prevent malicious actions of unscrupulous stakeholders.

The current Tokenomics paper presents the Mesh+ service on the IOTA platform that is prepared using formal methods and algebraic approach.

It demonstrates the equilibrium of the token life cycle and initial parameters of Tokenomics that were defined using algebraic modeling.

Scope

The following document presents specific tokenomics information for the MESH+ Project such as token distribution, reward structure and plans for the organization of financing rounds along with an in-depth modeling and simulation work realized using a multi-objective algebraic approach to establish a forecasted performance of the project over a medium and long timeframe under conservative conditions and based on a set of justified assumption.

The Algebraic modeling approach was implemented using the Insertion Modeling System IMS, a bespoke piece of software developed by the Glushkov Institute of Cybernetics of the National Academy of Ukraine. It functions on the basis of the principle of modeling the interactions between agents and their environments.

The model was used to establish a forecasted scenario for which graphs can be seen on the final 5 pages, including conservative values for parameters of interest such as Token Price, Liquidity, amount of Burned Tokens, Distribution of tokens towards traders, miners, investors and cryptocurrency exchanges.

It must be remembered that all modeling and simulation work is assuming a neutral case, not including considerations outside the scope of the model, and is therefore not necessarily representative of the real-life evolution of the token price or performance, which can be impacted by unforeseen events or unexpected patterns of action from stakeholders.

It is thus not to be considered as financial advice, but rather as an essential pursuit aimed at assuring a balanced/equilibrium case without undesirable emerging properties as a result of the chosen structure, such as decentralization or inherent vulnerabilities.

Reward Structure – the essentials

Most retail transactions with cryptocurrencies in the last couple of years have been recognised, and often self-reported, as speculative in nature. The inherent untapped value and potential to generate unique value through this novel asset class has been the main driver of investments in the field, with remarkable success in most cases. Larger investors are much more conservative and prioritise the protection of principal as well as the pursuit of further appreciation, whilst actual users benefitting from the infrastructure laid out through Mesh+ require predictability as a core property of the system.

Due to our function as the providers of a utility coin, linked through the inherent economic value of LoRa coverage, we provide incentives to all stakeholders involved with our network, and the quantum as well as the nature of the rewards have been thoroughly analysed, modelled, simulated and evaluated on both historical data from similar projects, an in-depth forecasting of demand for the technology and trends regarding the wider adoption potential from small businesses and service providers.

1. Coin allocation

The philosophy of our coin allocation process was designed to reward participation in the early development of the project through a small barrier for entry with high potential long-term rewards for potential users of the network as well as traders and miners.

The decreasing relative size of the Financing Rounds (seed, TGE1 & TGE2) provide opportunities to participate in the building of the network by purchasing discounted tokens. The Seed Round provides the most generous discounts of all by ensuring the fixed sale price of 0.1$/MESH, as well as a very large potential long-term share of the token supply, given the 300M maximum tokens in circulation and the 17% size of the seed round. This is half of the price expected for subsequent rounds, assuring great potential for long-term holders to secure very significant returns from the opportunity cost incurred, as well as rewarding early support. Essentially, the 17% of tokens sold at 0.1$/token are valued at 5,1 million, compared to the possible market cap of the project and the strategy for emissions and burns (token supply reduction) giving rise to the performance simulated in the graph on the last pages of the report.

The first people with investments above 1000$ gain the privilege of taking part in the test-net and are, thus, rewarded with the privilege to mine additional tokens to increase their participation while also benefitting from an improve trust score on the network, being able to cement their long-term mining rewards with the expectations for much higher rewards on network launch and expansion. Nodes without traffic will be rewarded, although these rewards decrease with time whilst traffic will generate increasing rewards.

2. Benefits for Early investors

An essential reward available for participants of early investment stages only, is the prospect of deciding to maintain the tokens in possession locked, in exchange for receiving a fraction of all operating proceeds from the provision of coverage to businesses. In essence, this acts as a sort of token dividend that benefits from, not just speculative action on the price of the increasing demand for the token, but also from the fees paid by large-scale business Users.

3. Locking coins

In a manner similar to the token locking reward mechanism provided for early investors, a small and predefined quantum of coins will be available for locking by later investors. This will be dependent on the freeing up of slots in a quantity proportional to the tokens unlocked by early investors.

4. Inherent properties of the Burn-Equilibrium equilibrium model

The Burn-Equilibrium model assures a limited supply of coins in circulation at any given point. In addition, it prevents emissions during downturns, thus preventing large fluctuations, whilst also providing liquidity in periods of immense network growth. In essence, this has a net positive impact on assuring a stable growing price for the tokens in a predictable manner that assures availability for businesses to join the network as well as by encouraging the growth of the network through the installation of devices.

CONTENTS

MESH+ PROJECT

DECENTRALIZED NETWORK FOR THE IoT
USING THE BURN-EQUILIBRIUM TOKEN MODEL

Brief information about the project

The project involves the deployment of a network of antenna devices to enable connectivity for the Internet-of-Things (IoT). This service functions on the IOTA Tangle using the MESH+ token. This token is used to pay for traffic by Users. The service also introduces an additional NUC token, which is linked to the dollar exchange rate and is used to determine the cost of traffic. Owners (miners) of purchased antenna devices (nodes) receive rewards for providing coverage and using the service.

TOKEN MESH+

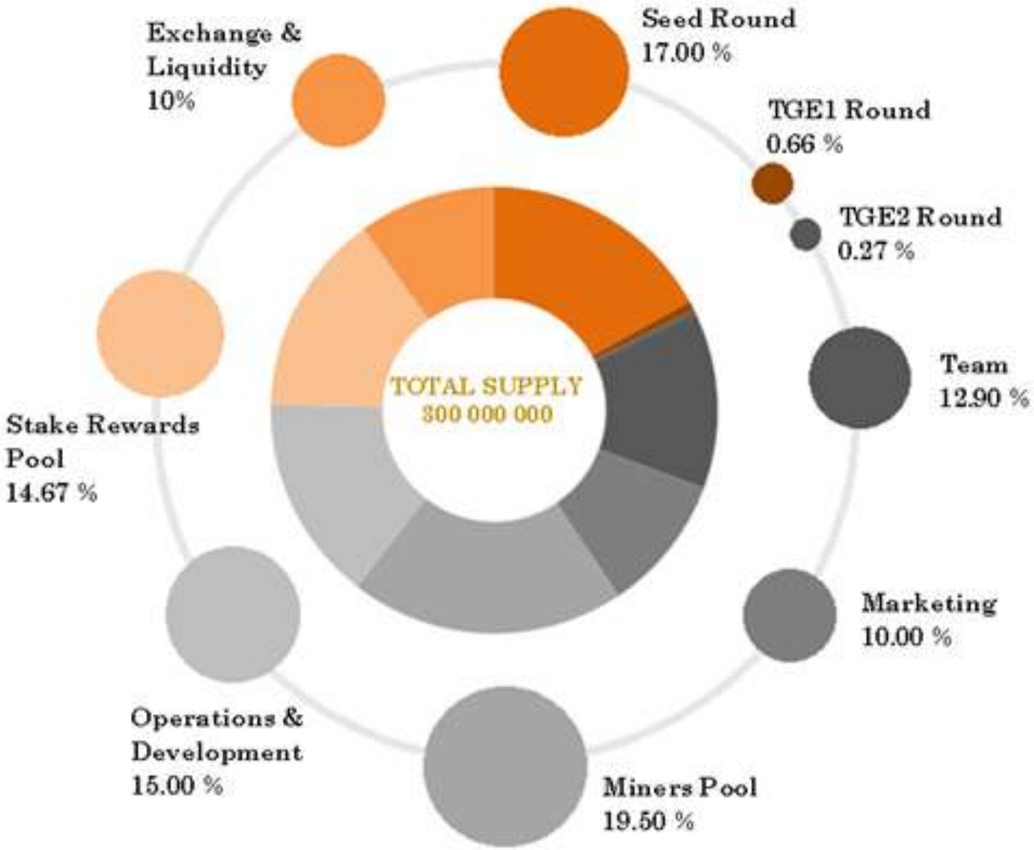
TOKEN NUC

300 M TOTAL SUPPLY

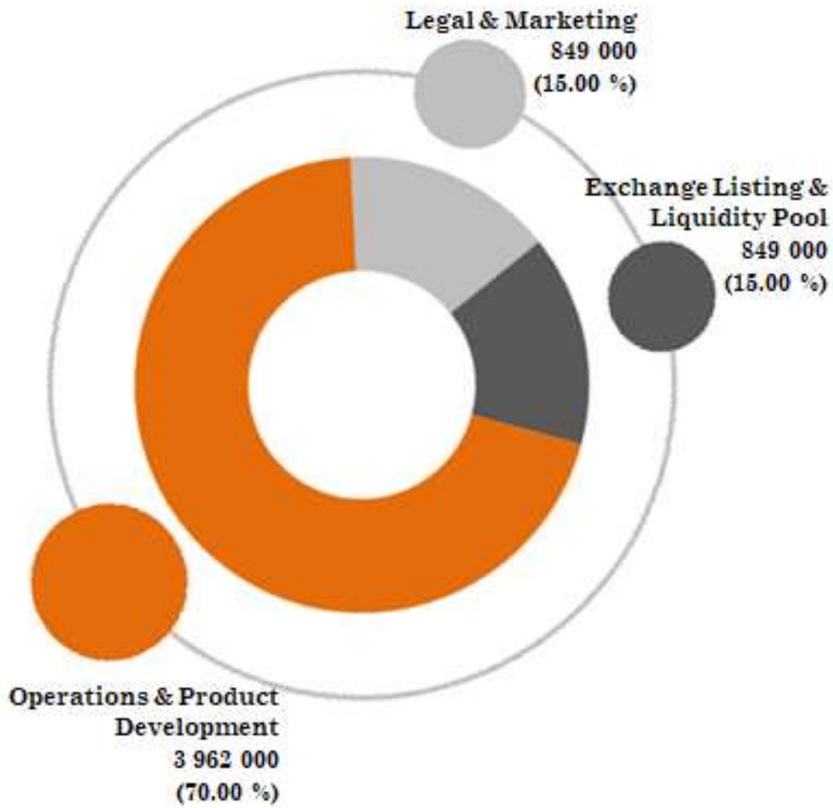
This paper presents the economics of the Mesh token, its life cycle, and scenarios for possible behavior under different conditions.

MESH Token Distribution

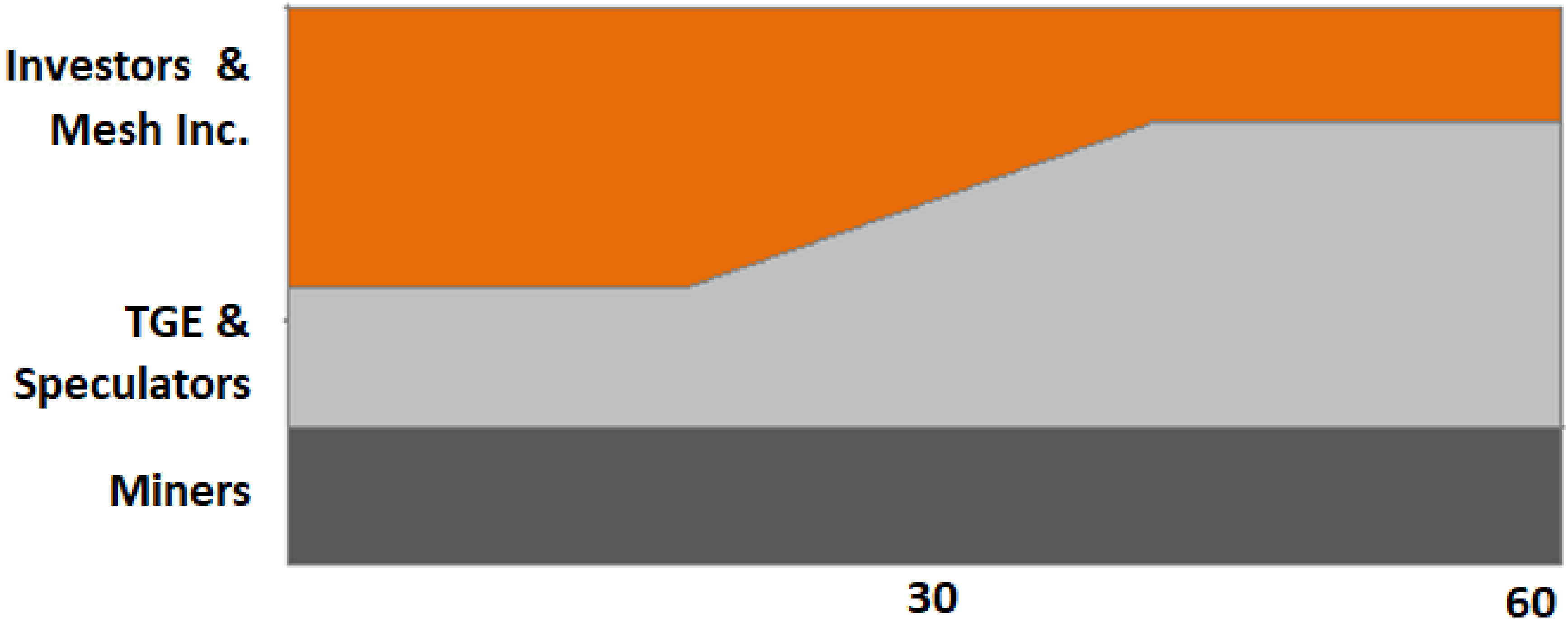
Token Allocation



Use of Sale Proceeds



Final distribution (60 years)



Introduction

The tokenomics of a project consists of two parts/stages. The first part - the **pre-production stage**, during which service development takes place, involves the open sale of tokens along with marketing activities for the operation. During this period, the sale of nodes begins. Purchased devices are involved in the test period and the owners of antenna devices (miners) are rewarded.

The product stage is defined by the functioning of the product and forms a self-managed sustainable system of tokenomics. During the product stage, IoT devices are sold and connected to the service network. Device owners buy tokens on a cryptocurrency exchange to pay for traffic. Investors, the team and miners receive tokens as rewards. Tokens shall be listed on the cryptocurrency exchange at the beginning of the product period.

PREPRODUCT STAGE 12 MONTH

EXCHANGE LISTING

PRODUCT STAGE 60 YEARS

Pre-production stage.

This stage is a public sale of tokens open for all with interest in supporting the project consisting of a sale with an initial set price of **$0.1.** The purpose of this stage is to raise funds for the listing on the cryptocurrency exchange, for further product development, as well as to establish appropriate fiat and token reserves on the exchange for their further sale to network users and traders.

The required number of tokens expected to be sold on the cryptocurrency exchange:

$$T = (U * B * D / P) * I$$

Where:

T - the required number of tokens to sell,

U - planned number of network users in the first month,

B - the estimated average amount of traffic per user (in bytes),

D - the cost per unit of traffic,

P - the planned initial price of the token at listing,

I - coefficient, which determines the additional number of tokens to be sold to traders in the first month. It depends on the participation rate of traders relative to token buyers who are users of IoT devices.

The pre-production stage consists of three rounds:

- Seed Round;
- TGE1 Round;
- TGE2 Round.

Token distribution for each of these rounds is equal 17% of the 300M Mesh tokens for the Seed round, 0.66% - for TGE1 and 0.27% for TGE1 rounds.

Token price for Seed Round is equal to \$0.1 and for the TGE Round – \$0.2.

Unlocking Schedule

Seed Round	
Start Unlocking	**3** months from exchange listing, **1%** unlock.
End Unlocking	In **5** years. Token shall be unlocked **daily**.
TGE1 & TGE2 Rounds	
Start Unlocking	**8** months from exchange listing.
End Unlocking	In **5** years. Token shall be unlocked **daily**.

Seed and TGE Investors automatically get a share in the stake of income (45% and 5% respectively).

Under certain conditions, such as - if the number of tokens was collected before the product period, it is possible to **increase the price of token**, as well as - an earlier listing on the cryptocurrency exchange.

Product stage

At this stage, the product begins to function and participants pay for the service with an internal NUC payment token, which is automatically generated when payment is made. The traffic participants themselves buy the MESH token on the exchange.

The exchange enables trade activity of owners who may buy, hold and sell tokens, thereby changing the liquidity of the token.

Traffic fees generate the profits of the service, which consist of team profits, rewards for investors and owners who block their tokens, and rewards for miners. A portion is allocated to manufacturers whose devices have joined the Mesh+ network.

Internal NUC token

The NUC internal token (Network Usage Coin) is designed to pay for data transfer and device communication on the network and is a stable token. The price is determined by the cost of sending a unit of information in the network (24 bytes). In the model we consider this cost to be constant and the change of its value is considered to depend only on market conditions of the external economy, on dollar liquidity.

Below is an example of converting a NUC to a basic Mesh+ token.

The user of an IoT device registers with the “connect portal” for device registration and monitoring and is provided with a certain number of bonus NUC tokens. In the future, the NUC conversion will be performed according to the calculation below.

Since users pay proportional to the number of bytes sent, assume that the cost of some unit of traffic (24 bytes) is $.00001.

Suppose that the cost per traffic unit P is 0.001 NUC, the price of NUC (Pn) is $.00001 and the price of token (Pt) is $2. Let's use the formula to calculate the number of tokens that are deducted from the wallet to generate NUC for 100 traffic units:

$$T = (Traffic * P/Pn) / Pt$$

Results in:

$$\mathbf{T = (100 * 0,001 / 0,00001) / 2 = 5000}\, tokens$$

Calculation of profits, rewards and token burn

The amount of tokens received for traffic is distributed among the project participants as follows:

- investors and team – 75%
- miners - 20%

The remaining tokens (5%), which are obtained as a result of traffic, are to be burned.

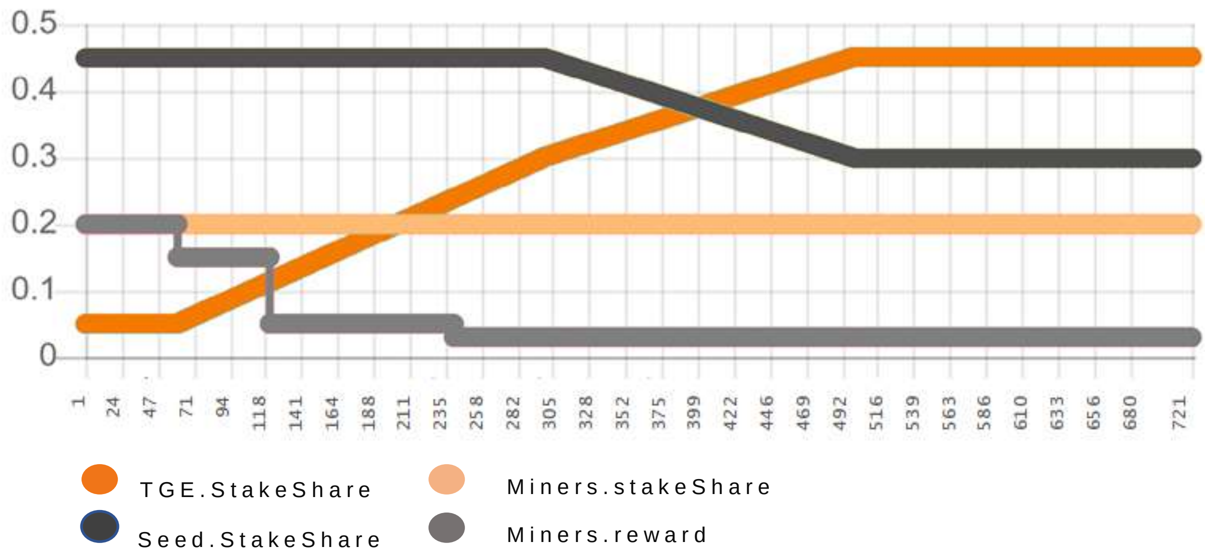


Fig.3. Burning and Rewards

The calculation of profit for miners is as follows:

$$P = \text{MCoef} * (X/T)$$

where:

P - the size of the token reward;

X - a value of the total fiat income;

T - current token price;

MCoef - miners Stake Share ratio.

Trading

Trading may be exercised by any person with the will and ability to buy and sell tokens on the cryptocurrency exchange. To model tokenomics, different ratios of traders' activity are considered, which are expressed as a percentage of the total number of tokens on the exchange.

Marketing activities

Marketing activities are employed as a procedure that increases the number of network **Users** as well as the number of **Miners**.

To model tokenomics, we use Helium's historical data on the growth of the number of miners, as well as historical data on the purchase of IoT devices.

We also don't consider a failed marketing outcome for modeling, but we do model tokenomics scenarios at different intensities as well as variability in intensities.

Agents in Mesh+ Tokenomics

In the process of creating the Mesh+ project model, two types of agents were identified – EXCHANGE and STAKEHOLDER.

We define an agent to be an entity, an individual or a group of individuals, a company or a group of companies (and any entity of individuals or companies).

Each agent has its own behavior in tokenomics cycles and significantly affects the entire tokenomics of the project as a whole. The mathematical model of the agents is presented in the corresponding section of this document.

A table describing the types of agents as well as the actions of the agents with respect to the main Mesh token is shown below.

Table 1. List of agents

№	Agents	Description of actions
2	**miners**	An agent of STAKEHOLDER type. They buys hardware in order to provide an IoT node and receives remuneration for transmitting data. He also sells his tokens to be exchanged for fiat.
3	**exchange**	An agent of EXCHANGE type It is an agent who represents the exchange on which the token is registered. He performs the purchase and sale of tokens to agents.
4	**tge and seed**	Agents of STAKEHOLDER type. A groups of individuals or companies investing in the purchase of tokens at the pre-production stage (Investors). Investors make a profit during the operation of the product - a % of transaction fees as well as receiving the privileged right to buy tokens at a significantly discounted price than on the exchange.
6	**team**	An agent of STAKEHOLDER type. The project team receives tokens from the general tokens pool, as profit and sells on the exchange.
7	**speculant**	An agent of STAKEHOLDER type. An agent, who trades, buys and sells tokens on an exchange for the purpose of speculating and making a profit by increasing the price. Other agents may be part of this group if they bought tokens on the exchange for the purpose of speculation.
8	**marketing**	An agent of STAKEHOLDER type. Receives tokens from the general tokens pool. These tokens are used to organize the different processes for the increase in the number of network Users as well as the number of Miners.
9	**devNoperations**	An agent of STAKEHOLDER type. Receives tokens from the general tokens pool and use it for the operational needs of the project.

Token life cycle

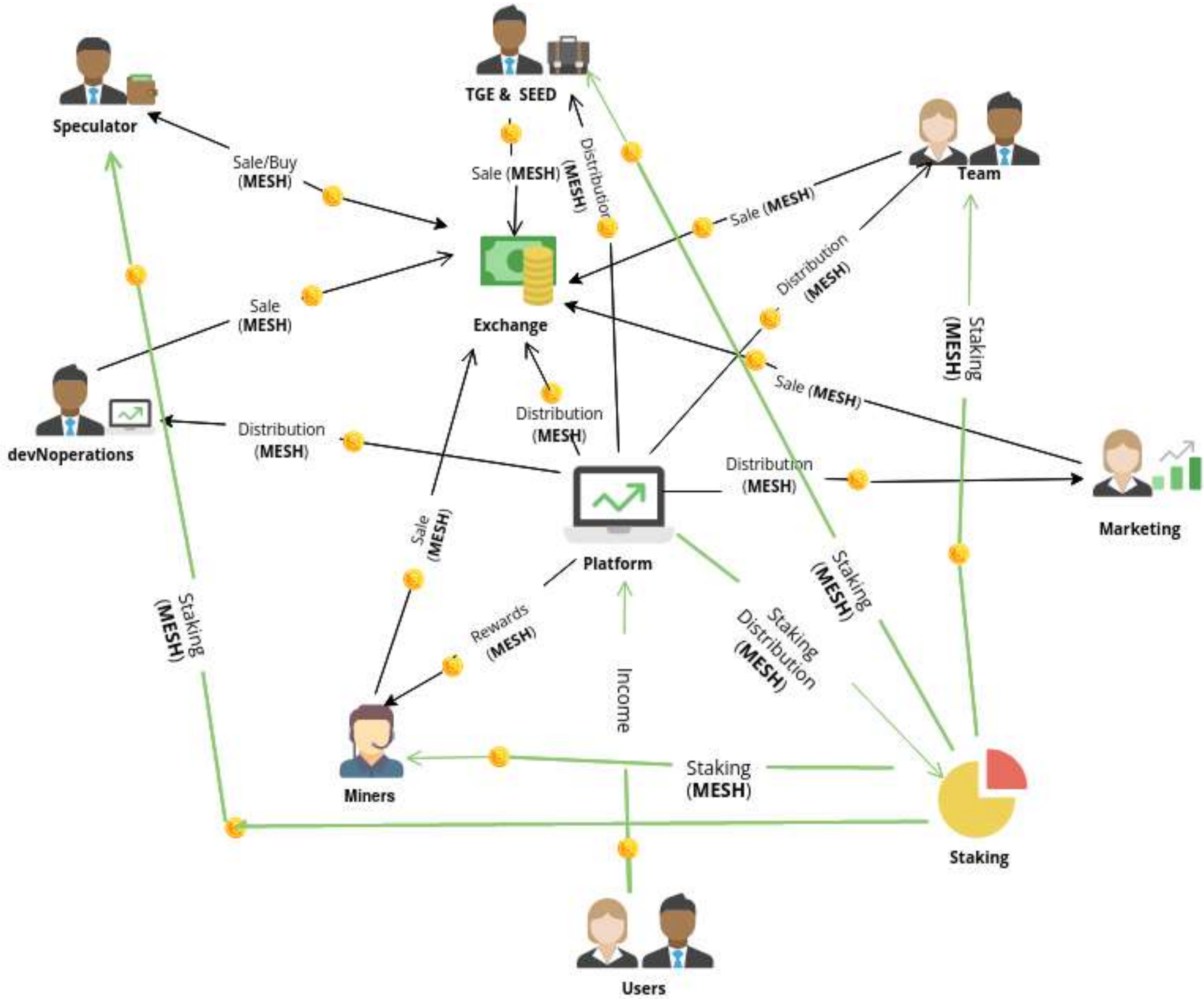


Figure 4. Token life cycle in Mesh+ tokenomics

1 Investor (**tge** and **seed** agents) buys tokens. The fiat proceeds from the sale of tokens to the Investor goes into the budget for development, marketing, and other expenses. Investors can sell tokens on the cryptocurrency exchange, receiving fiat as well as receiving tokens as profit for blocked tokens.

2 **Team** receives tokens as profit from the project's service from Platform (general tokens pool). The team sells tokens on the cryptocurrency **exchange**.

3 **Miners** receive a reward from the Platform (Reward Pool) for covering the network, as well as a reward for transmitting data. **Miners** sell tokens for the **Exchange**.

4 **Exchange** sells **Speculant** tokens and buys tokens from **Speculant**, **Team**, and **Investor**.

5 **Speculant** buys and sells tokens from the cryptocurrency **Exchange**.

6 **Marketing** receives tokens from the general tokens pool. **Marketing** sells tokens for the **Exchange**.

7 **devNoperations** receive tokens from the general tokens pool. **devNoperations** sells tokens for the **Exchange**.

Mathematical model of tokenomics

Our approach is to use an algebraic modeling approach, implemented within the framework of the Insertion Modeling System IMS (http://apsystems.org.ua/node/7).

The insertion modeling system IMS was developed at the Glushkov Institute of Cybernetics of the National Academy of Sciences of Ukraine under the guidance of Academician of the National Academy of Sciences of Ukraine, Professor A.A. Letichevsky.

Insertion modeling is an approach for modeling complex distributed systems, which is based on the theory of interaction between agents and environments. In the last decade, this theory has been successfully used for the verification of software system specifications.

The Basic Protocol Specification Language (BPSL) is used to represent requirements specifications. The basic concept of the language is that of a basic protocol. The basic protocol represents the transition of some system from one state to another. Basic protocol is a three-component entity, which includes - transition precondition, postcondition and action or process of basic protocol. We speak about the applicability of the basic protocol, if the system can make the transition representing it. After application of the basic protocol, a change of the system state occurs, which is described in the postcondition of the basic protocol with the help of imperative or declarative operators of the basic language.

An agent represents a system or component for which a requirement is presented in the underlying protocol describing its transition. Immersing an agent in the environment means modeling its behavior according to the behavior of the environment. The environment affects the agent with trigger events and changes its states. In essence, the environment is also some higher-level agent interacting with the immersed agents. The semantics of BPSL allows the creation of concrete and symbolic models at different levels of abstraction. An agent and an environment have typed attributes, which are formalized in corresponding theories.

For mathematical refinements, we use a transitory system for the agent, which is the most abstract mathematical concept that models a system that evolves over time.

Within the method of insertion modeling we use specifications of behavior algebra for formalization. The behavior algebra is a bipartite algebra, the first component of which is the set of behaviors, and the second component represents the set of actions. The signature of the behavior algebra consists of two operations, one relation, and three constants.

The first operation is called prefixing. Its arguments are action and behavior. The result of the operation is a new behavior. The second operation is a nondeterministic choice operation. It is a binary operation defined on a set of behaviors.

The constants of the behavior algebra are successful completion Δ, indeterminate behavior $\perp$ and deadlock behavior, which is zero (a neutral element) of nondeterministic choice. On the set of behaviors a binary approximation relation is defined, which is a partial-order relation with the smallest element $\perp$.

The set of first-order logic formulas over polynomial arithmetic is used as a basic logical language. Below is the experience of using the algebraic modeling system in modeling and analyzing tokenomics models.

Experience using IMS in tokenomics modeling:

- Letychevsky, O., Peschanenko, V., Radchenko, V., Poltoratzkyi, M., Kovalenko, P., & Mogylko, S. (2019, May). Formal Verification of Token Economy Models. In 2019 IEEE International Conference on Blockchain and Cryptocurrency (ICBC) (pp. 201-204). IEEE.
- Letychevskyi, O., Peschanenko, V., Poltoratskyi, M., & Tarasich, Y. (2019, June). Our Approach to Formal Verification of Token Economy Models. In International Conference on Information and Communication Technologies in Education, Research, and Industrial Applications (pp. 348-363). Springer, Cham.
- Letychevskyi, O., Peschanenko, V., Radchenko, V., Poltoratskyi, M., & Tarasich, Y. (2019). Formalization and algebraic modeling of tokenomics projects. In CEUR Workshop Proceedings (pp. 577-584).

Tokenomics modeling tools:

1 Tokenomic constructor

The tokenomic constructor had been developed by our team members, Prof. Volodymyr Peshanenko and Maksym Poltoratskyi, and a group of scientists and programmers from the algebraic school of academician Victor Glushkov, all with over 20 years of industry experience and scientific research in the use of formal methods.

It is based on the AVM (Algebraic Virtual Machine) system and it presents a special Web interface for modeling and evaluating tokenomics projects.

The tokenomic constructor features the following functions:

1. Creation of multiple scenarios and token lifecycles
2. Aid in model creation with different levels of abstraction
3. Use of external scenarios for liquidity & market activity changes
4. Behaviour algebra specifications for token economy modelling
5. Verification of the model for undesirable properties such as centralisation or token leakage

2 Model Creator

Model Creator is a system that uses symbolic modeling techniques, including algebraic and deductive-formal methods for solving complex problems.

Key features of the platform:

- ✓ testing technology;
- ✓ model-based development;
- ✓ supporting the development process of a critical system or quality of service (QoS) system;
- ✓ verification and validation;
- ✓ cybersecurity.

The Model Creator includes a number of systems and libraries for implementing algebraic formal methods and integrating with other software systems.

An example of the implementation of the model in the system is shown in Fig.6-7.

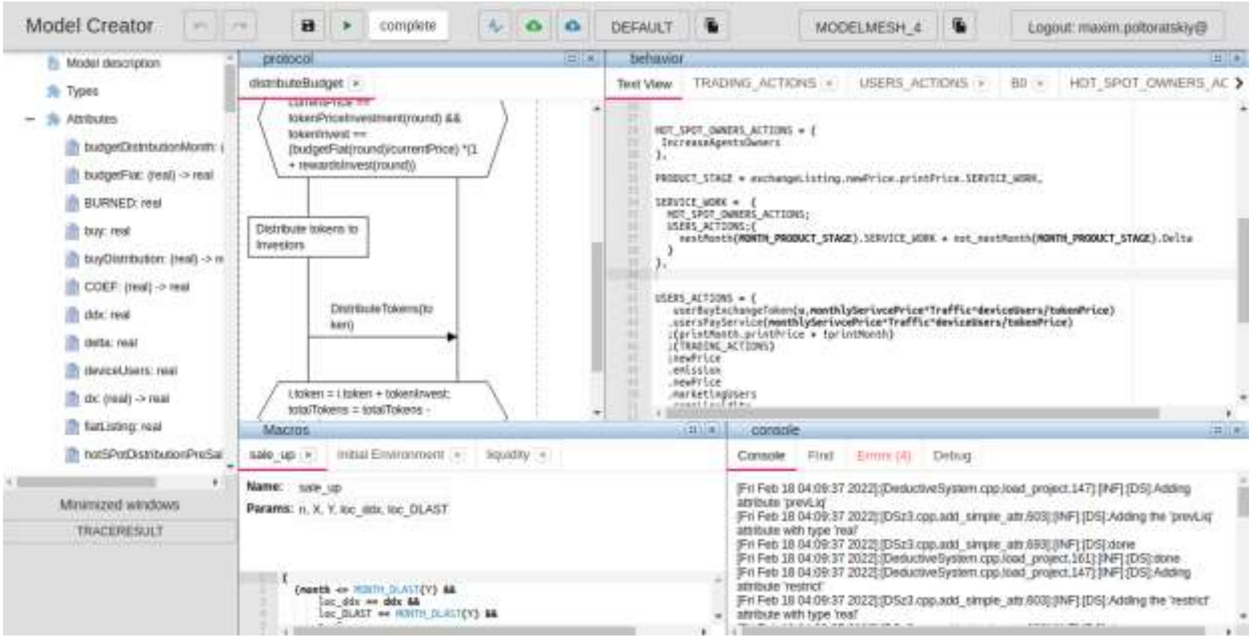


Fig.5. Web-interface of the Model Creator. An example of the representation of actions, behaviors, macros.

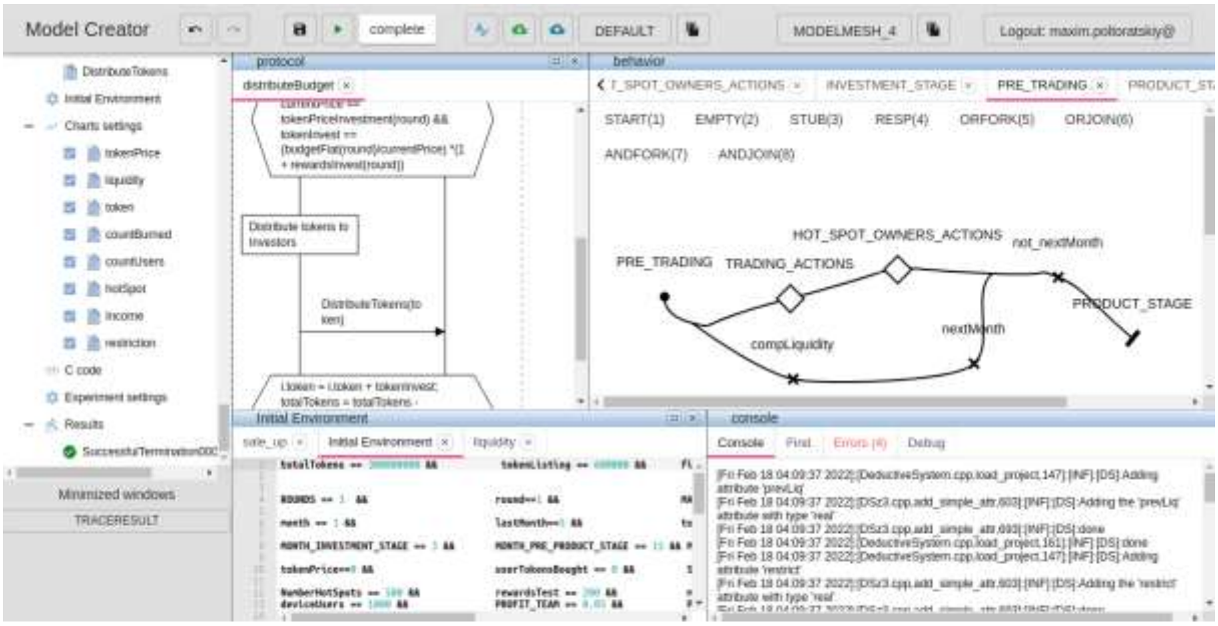


Fig.6. The Web-interface of the Model Creator. An example of the representation of actions, behavior (UCM), the formula of the initial state of the environment.

Formalizing the Mesh+ tokenomics model in terms of behavioral algebra

This section presents a formalization of the Mesh+ project tokenomics in the form of behavioral algebra formulas (behavioral equations) and a set of basic protocols that specify the semantics of agent actions. This formalization is used for further modeling.

Description of environment attributes & agent types.

This section describes the basic attributes of the environment, agent types and their attributes. We describe only the main ones, which are essential for understanding the basic protocols. We omit additional and technical attributes.

Environmental Attributes:

month:int

Current month

tokenPrice:real

Current token price

LIQUIDITY:real

Token liquidity

SOLT_TOKEN:real

Number of sold tokens

BOUGHT_TOKEN:real

Number of tokens purchased by users

COEF:(int)->real

Coefficient determining the angle of inclination for a linear increase in sales

totalFiatIncome:real

The total amount of profit from sales

DyExponent:(int)->real

A function describing the exponential growth in token sales

DyLinear:(int)->real

A function describing the linear growth of token sales

salesMaximalUSD:(int)->real and **salesMinimalUSD:(int)->real**

Maximum and minimum profit thresholds

rewardsPool:real

Amount of tokens allocated for rewards

tokenSpeculant:real

The number of tokens owned by speculators

Types of agents and their attributes

agent_types:obj(

EXCHANGE:obj(

fiat:real,

token:real

),

STAKEHOLDER:obj(

token:real,
tokenLocked:real,
tokenStaked:real,
tokenStakedLocked:real,
startUnlocking:real,
endUnlocking:real,
initialUnlock:real,
stakeShare:real,
tokenShare:real,
dUnlock:real,
dVesting:real,
startVesting:real,
endVesting:real,
saleFactor:real,
stakeChangeStart:real,
stakeChangeEnd:real,
DstakeShare:real,
stakeShareMin:real,
reward:real),

);

Agent names (seen in parentheses after obj)

```
agents:obj(
        EXCHANGE:obj(exchange),
        STAKEHOLDER:obj(seed),
        STAKEHOLDER:obj(tge),
        STAKEHOLDER:obj( minners),
        STAKEHOLDER :obj(speculant),
        STAKEHOLDER:obj(team),
        STAKEHOLDER :obj(marketing),
        STAKEHOLDER :obj(devNoperations)
    );
```

Equations of behavior

This section describes a tokenomics model that consists of behavioral equations.

The top-level main equation as a sequential composition of the five parts of tokenomics:

```
B1 = (    (UNLOCKING); (VESTING);
    (SALES);
    (TOKEN_CIRCULATION);
    (STAKING);
    (nextMonth.B1 + !nextMonth.Delta)
),
```

Equation representing the UNLOCKING part:

```
UNLOCKING = (
    (unlockInitial(devNoperations) + !unlockInitial(devNoperations));
    (unlockInitial(marketing) + !unlockInitial(marketing));
    (unlockStakeInitial(team) + !unlockStakeInitial(team));
    (unlockStakeInitial(seed) + !unlockStakeInitial(seed));
    (unlockStakeInitial(tge) + !unlockStakeInitial(tge));
    (unlock(devNoperations) + !unlock(devNoperations));
    (unlock(marketing) + !unlock(marketing));
    (unlockStake(team) + !unlockStake(team));
    (unlockStake(seed) + !unlockStake(seed));
    (unlockStake(tge) + !unlockStake(tge)) ),
```

Behavior describes the tokens unlocking for each type of agents.

Equation representing the VESTING part:

```
VESTING = (
    (vesting(team) + !vesting(team));
    (vesting(seed) + !vesting(seed));
    (vesting(tge) + !vesting(tge))
),
```

Equation representing the SALES part:

```
SALES = (
    (salesLinear(1) + !salesLinear(1));
    (salesExponent(2) + salesExponentInter(2) + noSales(2));
    (buyBurnToken + !buyBurnToken);
    (salesLinear(3) + !salesLinear(3));
    (salesExponent(4) + salesExponentInter(4) + noSales(4));
    (salesExponent(5) + salesExponentInter(5) + noSales(5));
    (minersReward);
    (minersRewardsChanging + !minersRewardsChanging)
),
```

The behavior describes several types of token sales - linear and exponential - actions salesLinear(n) and salesExponent(n), n=1,…,5. Thus, in order to model the possibility of transitions between types of sales, 5 different sales time intervals are considered. Behavior also includes actions of tokens burning and rewarding miners.

Equation representing the TOKEN_CIRCULATION part:

```
TOKEN_CIRCULATION = (
   (saleToken(team) + !saleToken(team));
   (saleToken(tge) + !saleToken(tge));
   (saleToken(marketing) + !saleToken(marketing));
   (saleToken(devNoperations) + !saleToken(devNoperations));
   (saleToken(seed) + !saleToken(seed));
   (saleToken(miners) + !saleToken(miners));
   (print);
   (newLiquidity + !newLiquidity);
   (newPriceDelta + !newPriceDelta)
),
```

Behavior describes the processes of selling tokens by various agents, including the actions of recalculating the price change and liquidity of the token.

Equation representing the STAKING part:

```
STAKING = (
  (stakeDistribution);
  (stakeChange(seed,tge) + !stakeChange(seed,tge));
  (stakeChangeBurn(tge) + !stakeChangeBurn(tge));
  (stakeSpeculant + !stakeSpeculant)
)
```

Behavior includes the actions of allocating the share of the stake and changing the investors' share of the stake.

Basic agent protocols

Each agent's base protocols describe the transition in the tokenomics system from one state to another when the token distribution changes.

The basic protocols for the UNLOCKING process

```
unlockinitial = (( month == x.startUnlocking)->
        ("Environment#env:action 'Unlock Token' ")
        (x.token = x.tokenLocked*x.initialUnlock);
        x.tokenLocked = x.tokenLocked*(1 - x.initialUnlock))
        ) ,
```

```
unlock = (( x.endUnlocking >= month > x.startUnlocking)->
        ("Environment#env:action 'Unlock Token' ")
        (x.token = x.token + x.dUnlock;
        x.tokenLocked = x.tokenLocked - x.dUnlock)
        )
```

These protocols describe the process of tokens unlocking for such types of agents as devNoperations, marketing, team, seed and tge. Unlocking of tokens occurs daily at a specific time interval for each type of agent. The number of unlocked tokens is calculated by the formula:

$$X = N * (1 - k)/(d_2 - d_1)$$

where
X – number of unlocked tokens,
N – total number of locked tokens (for agent type),
k – the percentage of tokens that are unlocked,
d_1 – unlock start time, d_2 – unlock end time.

The basic protocols for the VESTING process

```
vesting = ((x.startVesting <= month < x.endVesting) ->
          ("Environment#env:action 'Vesting' ")
          (x.token = x.token + x.dVesting*x.tokenStaked;
          rewardsPool = rewardsPool - x.dVesting*x.tokenStaked)
),
```

This protocol describes the process of vesting for such types of agents as team, seed and tge. Vesting continues from exchange listing month until end of low income period. The vesting coefficient is determined for each agent individually. The corresponding number of tokens is credited to the agent and deducted from the reward pool.

The basic protocols for the SALES and TOKEN_CIRCULATION

```
buyBurnToken =((exchange.token > 0) ->
          ("Environment#env:action 'Buy and Burn Token' ")
          (exchange.token = exchange.token - totalFiatIncome/tokenPrice;
          exchange.fiat = exchange.fiat + totalFiatIncome;
          BOUGHT_TOKEN = BOUGHT_TOKEN + totalFiatIncome;
          rewardsPool = rewardsPool + totalFiatIncome/tokenPrice;
          totalFiatIncome = 0)
),
```

The protocol describes the processes of buying and burning tokens.

The number of tokens purchased by agents is deducted from the exchange and added to the reward pool. The amount of exchange's fiat received from sales increases.

```
minersReward = ((1) ->
          ("Environment#env:action 'Miners Rewards' ")
          (miners.token = miners.token + miners.reward*(totalFiatIncome/tokenPrice);
          rewardsPool = rewardsPool - miners.reward*(totalFiatIncome/tokenPrice);
          totalFiatIncome = 0)
),
```

Miners receive a reward from the Platform (Reward Pool) for covering the network, as well as a reward for transmitting data. The required number of tokens is deducted from the reward pool and added to the total number of miner tokens.

Recalculation of token price and liquidity:

```
newPriceDelta =(
        (month > 1)->
        ("Exchange#e:action 'Recalculation of the new price on the exchange ';")
        (tokenPrice = exchange.fiat /exchange.token)),
```

```
newPriceRatio =(
        (month > 1)->
        ("Exchange#e:action 'Recalculation of the new price on the exchange ';")
        (tokenPrice = tokenPrice*LIQUIDITY)),
```

```
compLiquidity =(Exist (liq:real)(

        (month > 1)->
        ("Environment#env:action 'Token Liquidity Calculation ';")
        (LIQUIDITY = BOUGHT_TOKEN/SOLT_TOKEN)),
```

The price of a token is determined by the ratio of the amount of fiat on the exchange to the number of tokens. Liquidity is defined as the ratio of all bought tokens to sold tokens over a certain period (in this case, a month).

Basic protocols governing month sequences

```
nextMonth=(
        (month <= lastMonth)->
        ("Environment#env:action 'Next month';")
        (BOUGHT_TOKEN = 0;
        SOLT_TOKEN = 0;
        month = month + 1),
```

Not considered are some technical and contrarian protocols that are triggered if the precondition is not met, for example, the month in which the protocol is executed does not match.

Simulation results

This system was simulated with the following conditions:

- ✓ the effect of trading activity on token liquidity 10-40%;
- ✓ a positive trend in marketing with regard to market saturation and coverage was considered. An unsuccessful outcome was not considered;
- ✓ volatility was considered with different starting number of tokens when listing the exchange, with different distribution of marketing, with different initial number of buyers;
- ✓ sensitivity is observed in more than 5 years in the case of a shortage of tokens for the exchange settlements with the increase in consumers. To eliminate this, the required (larger) number of tokens for the listing is calculated.

Initial suggested values of tokenomics to obtain consensus:

rewardsPool == 44010000

Amount of tokens distributed for Rewards Pool

exchange.token == 30000000

Amount of tokens distributed for Exchange

devNoperations.tokenLocked == 45000000

Amount of tokens distributed for Development & Operations

marketing.tokenLocked == 30000000

Amount of tokens distributed for Marketing

seed.tokenStakedLocked == 51000000

Amount of tokens distributed for Seed Round

team.tokenStakedLocked == 38700000

Amount of tokens distributed for Team

tge.tokenStakedLocked == 2800000

Amount of tokens distributed for TGE Round

miners.token == 58490000

Amount of tokens distributed for Miners

month == 1 && **lastMonth == 720**

time range covering all stages of tokenomics

tokenSpeculant == 2000

the number of tokens held by speculators

tokenPrice == 0.4

initial token price

seed.initialUnlock == 0.01

the percentage of tokens that are unlocked

tge.dVesting == 0.001 && team.dVesting == 0.0005 && seed.dVesting == 0.003…

vesting coefficient for different types of agents

Chart 1. Tokens Unlocking.

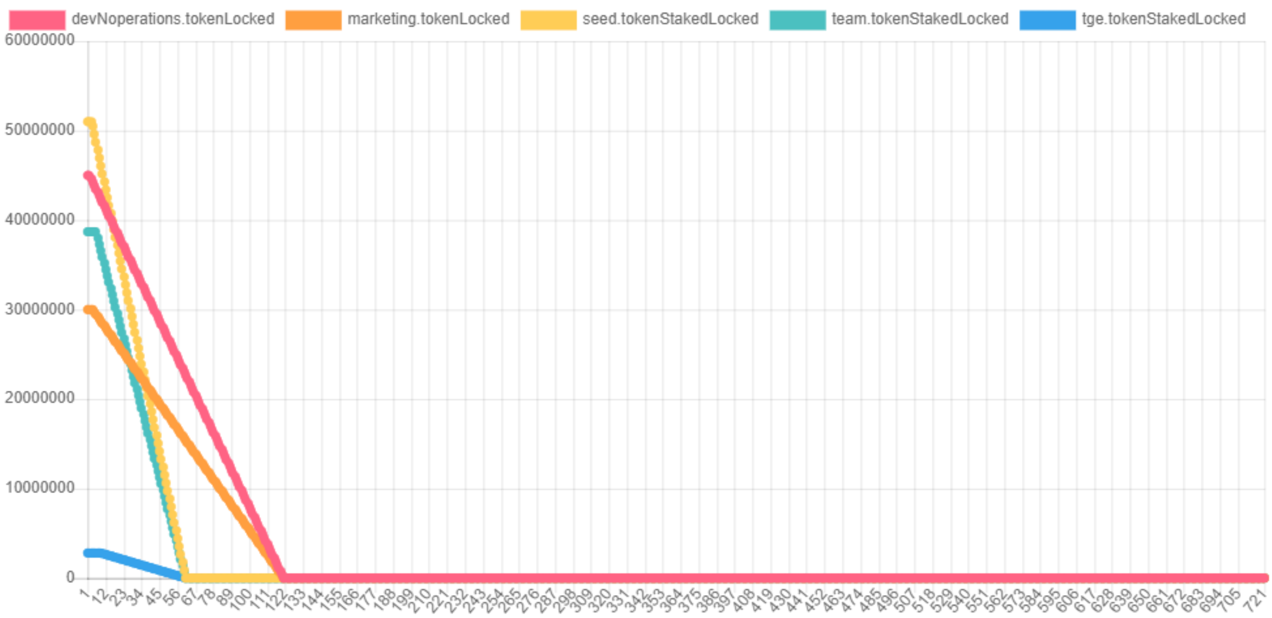


Chart 2. Burning and Rewards.

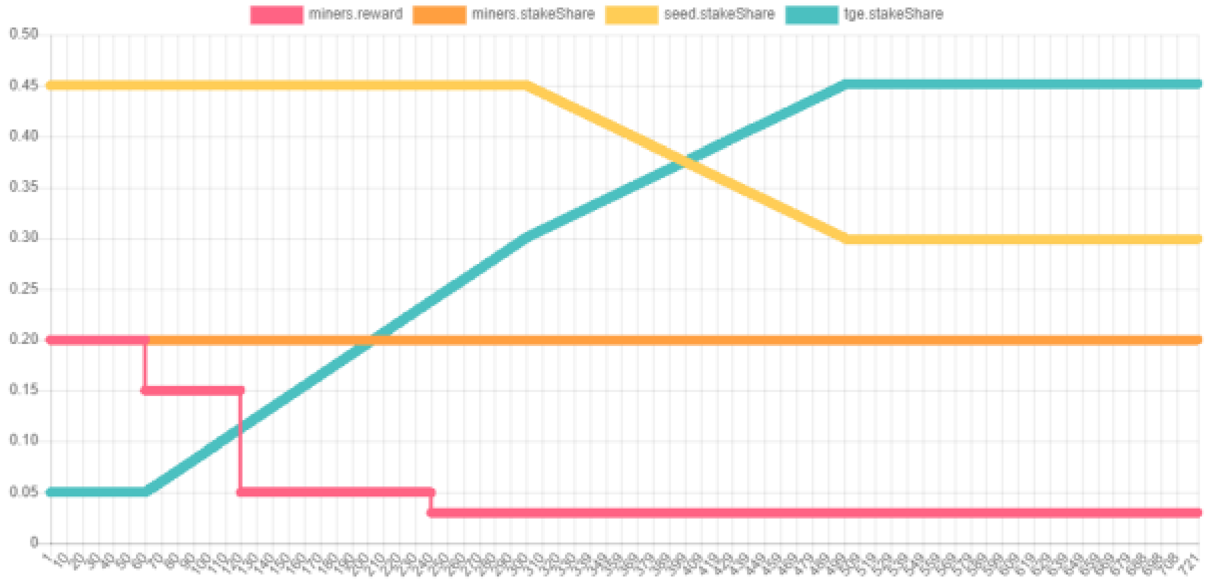


Chart 3. Distribution of tokens on the exchange.

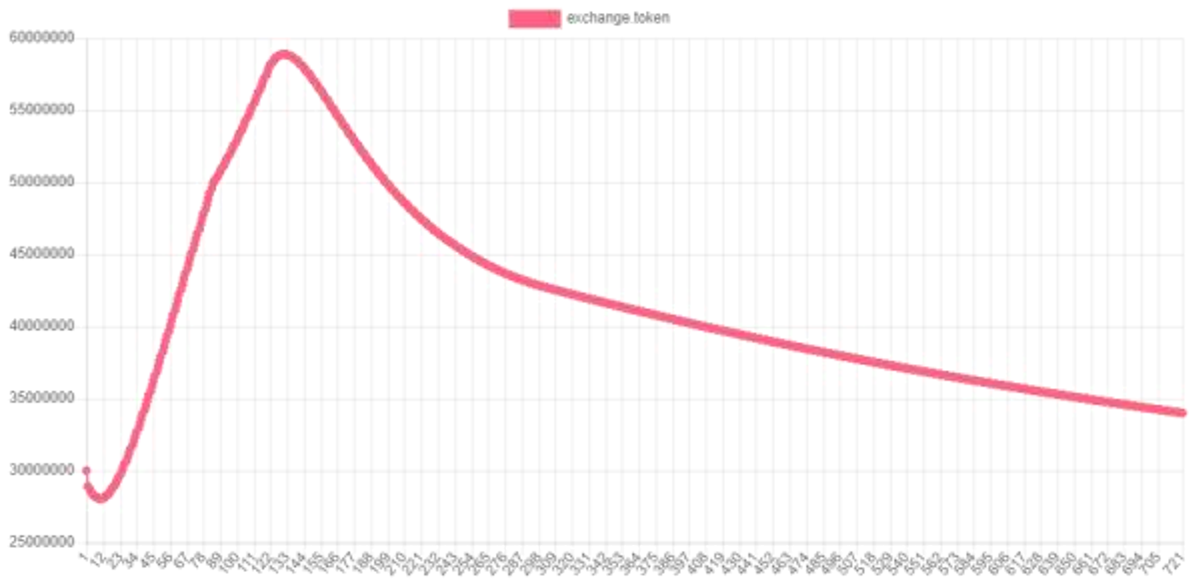
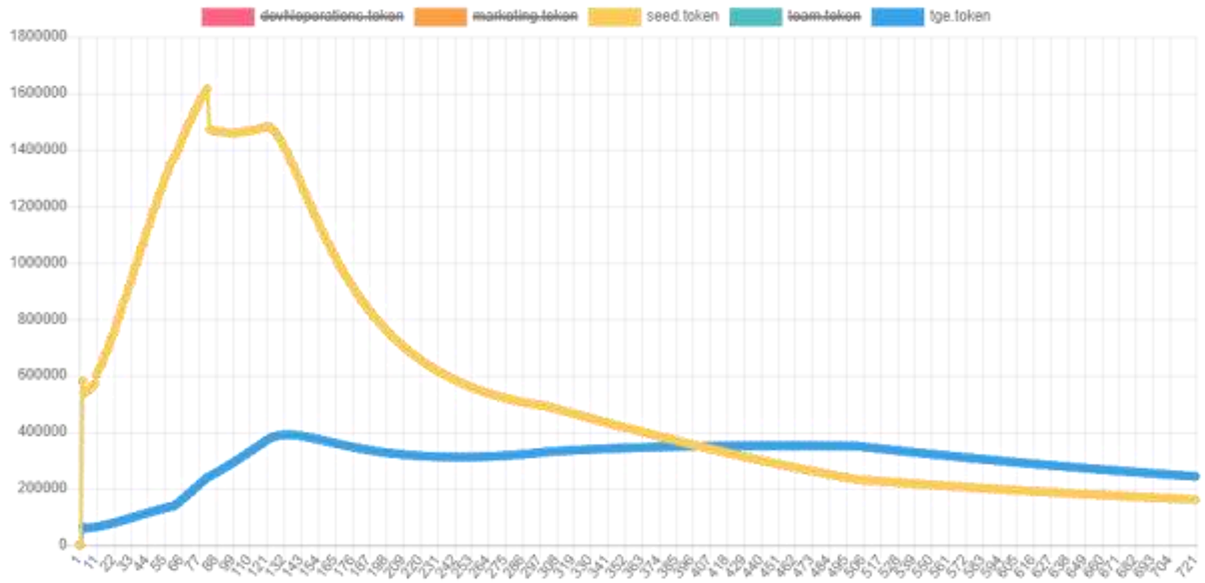
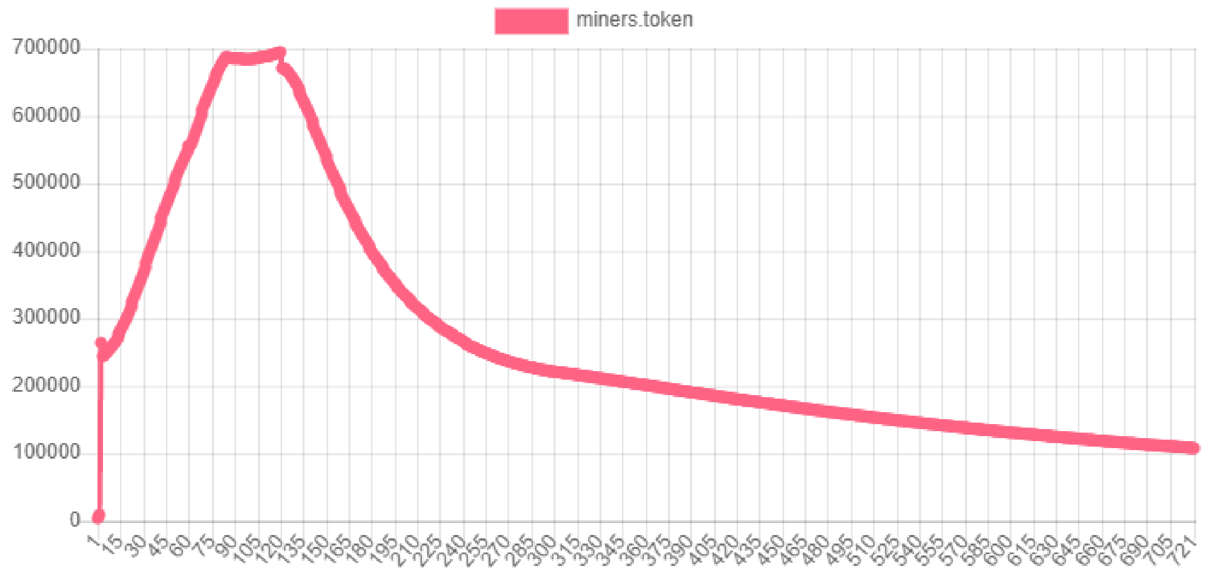


Chart 4. Distribution of tokens among investors.



The above graph signified the investor profits (in tokens). Investors may sell the received tokens with a certain intensity.

Chart 5. Distribution of tokens among miners.



The purchaser of the equipment profits from the traffic (in tokens), and can sell the received tokens with a certain intensity.

Chart 6. The ratio of sold to buy tokens.

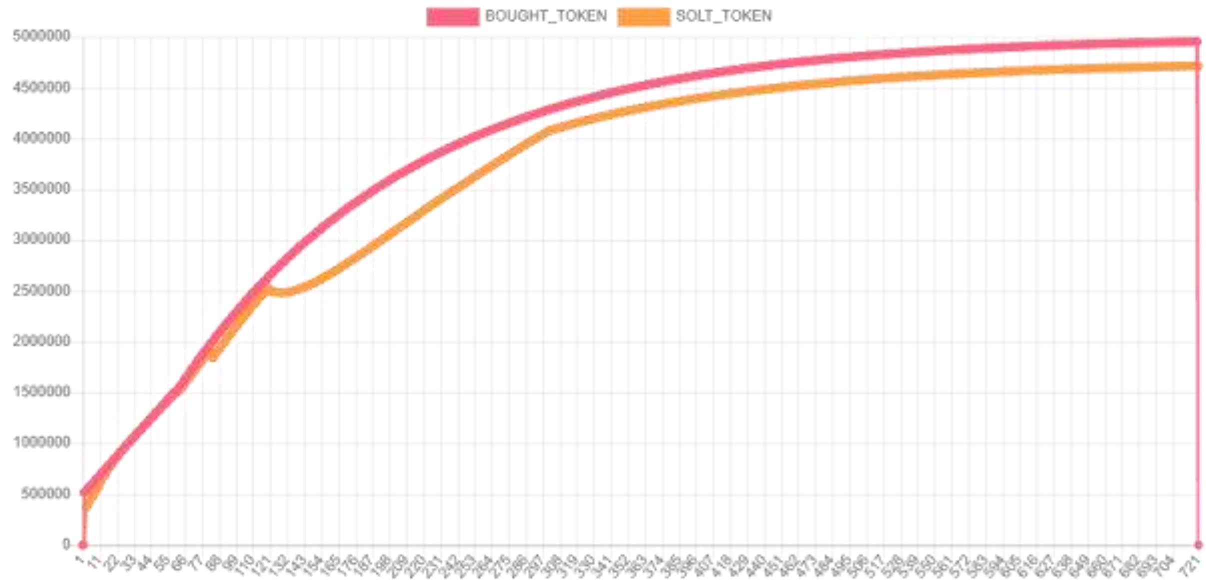


Chart 7. Token liquidity

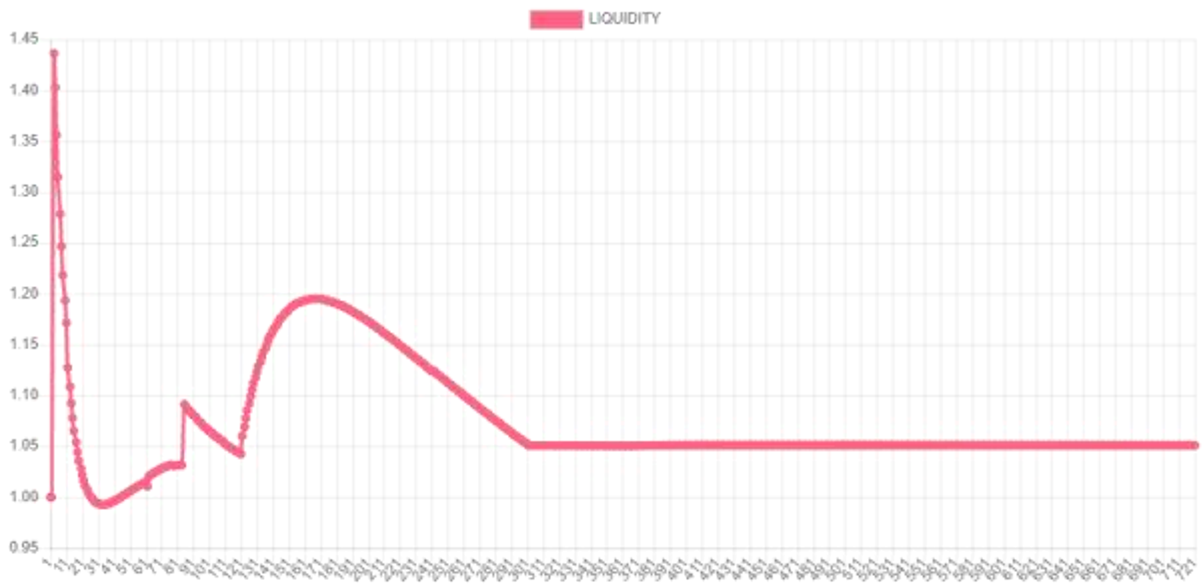
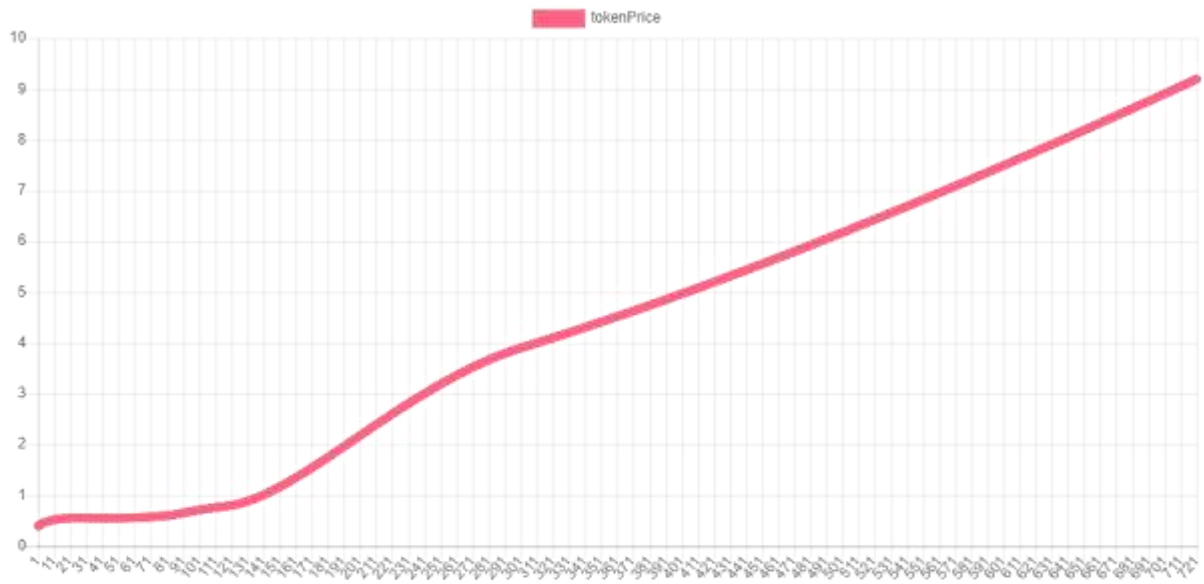


Chart 8. Token price



The growth of the token price is uniform.

Comments on the model

1. The model works with given sales volumes of the service. As the volume decreases, the price of the token may fall but the balance will be restored according to the number of tokens that have been burned.
2. The model considers the worst factor of the sales incentives. It is implied that all tokens earned by passive income or distributed to teams are immediately sold by fiat on the exchange. With the participation of speculators who will hold tokens and buy them to hold tokens, the token's liquidity will increase.
3. Significant pressure on tokenomics is exerted by the outflow of tokens for development and marketing - this is more than a quarter of all tokens. Therefore, for balance, it was advisable to extend the unlocking period. The second condition for the model is the monthly outlay (sale) of tokens for development - no more than 5 percent.
4. Note that in the model, half of the burned tokens are offset in the reward pool, and the other half - as an emission on the market.

info@meshplus.io

https://meshplus.io/