# White Paper: NFTAMA

## 1. Core Concept and Vision

NFTAMA.io is a Web3 gaming platform where users can earn and use the AMA2 (Amatu) token through a system of NFT safes.

The core concept of the project:

- users receive NFT safes;
- safes generate AM weight;
- AM weight determines the share of participation in AMA2 distribution;
- AMA2 can be spent on safe upgrades, account upgrades, minting new safes, and internal operations;
- this creates a closed growth cycle of gameplay and economics.

NFTAMA is a gaming crypto platform in which NFT safes function as assets with parameters, history, and a yield function. A user does not simply buy an NFT image, but receives a developing digital asset that provides AM weight, affects the user’s share in the project economy, and can strengthen over time through age, upgrades, and minting of new safes.

### What Problem the Project Solves

NFTAMA solves several tasks at once:

- it turns NFTs from static images into assets with practical utility;
- it gives the user a game-based way to participate in token economics;
- it creates a clear reinvestment mechanism through AMA2;
- it combines NFT ownership, upgrading, and token rewards in a single system;
- it builds long-term motivation to hold and develop the asset rather than quickly resell it.

### Target Audience

NFTAMA is aimed at:

- investors interested in a long-term model of accumulating a position through NFTs and a scarce token;
- users who want to participate in the platform economy through clear actions;
- gamers who are drawn to mechanics of growth, level, rarity, and age;
- DeFi enthusiasts using BNB Smart Chain, DEX, liquidity, and tokens;
- NFT collectors for whom rarity, class, series, and asset history matter.

### Long-Term Project Goal

The strategic goal of the project over a 1-3 year horizon is to increase the utility and circulation of AMA2 within the platform, expand the number of users, strengthen the scarce token model, and form a sustainable ecosystem of NFT safes.

The team’s target reference point for AMA2 price over a horizon of about 3 years is approximately `0.3 USD` per `1 AMA2`.

Note:

- this is a strategic target, not a guaranteed return;
- the actual AMA2 price will depend on demand, liquidity, user activity, overall market conditions, and the pace of project development.

## 2. How NFTAMA Differs from Regular NFTs, GameFi, and DeFi

### Compared to Regular NFTs

Regular NFTs are often limited to collectible or visual value.

In NFTAMA, a safe is an asset with practical utility:

- it provides AM weight;
- AM weight participates in AMA2 distribution;
- the safe can be upgraded;
- the safe ages and can become stronger;
- the safe can be used to mint a new NFT;
- the safe can be sold to another user.

### Compared to Typical GameFi

In many GameFi projects, game items exist only inside the game.

In NFTAMA:

- the safe is an NFT asset;
- the project operates on BNB Smart Chain;
- the user can deposit and withdraw assets;
- there is a secondary market between users;
- there is internal financial infrastructure.

### Compared to Typical DeFi

In classical DeFi, the user mainly interacts with a token, pool, or staking.

In NFTAMA, the user works not only with a token but also with a portfolio of gaming NFT assets:

- every decision regarding safes affects AM weight;
- AM weight affects the user’s share in AMA2;
- AMA2 affects the speed of further portfolio growth.

## 3. How the Project Works as a Whole

The basic NFTAMA cycle is as follows:

1. The user registers, passes captcha, and receives an internal IN wallet.
2. The user confirms email to fully participate in the AMA2 economy. Until the email is confirmed the account is frozen: no AMA2 for weight and no referral bonuses, no buying safes (auto-buy included), no market deals, minting, upgrades, modifier keys, swaps or withdrawals, and challenges and the safe of luck do not count. Safes and balance stay and work again right after confirmation. If letters do not reach the address (it does not exist, the mailbox is full), mailing to it stops until it is confirmed or changed.
3. The user tops up the internal balance with BNB and, if needed, AMA2.
4. The user buys a safe when opened, uses auto-buy, buys a safe from another user, or mints a new one.
5. The user receives an NFT safe that is assigned in the system and then minted to the internal wallet.
6. The user increases total AM weight through new safes, their age, upgrades, and minting.
7. The user receives AMA2 proportionally to the share of weight.
8. The user uses AMA2 for further growth, exchange, withdrawal, or sale.

This cycle is already reflected in the product structure:

- `Dashboard`
- `Account`
- `Wallet`
- `Virtual Safes`
- `AMA Tokens`
- `Rules`
- `Finances`
- `Referral Program`
- `Blog`

# 4. NFT Safe Concept

## What an NFT Safe Is

An NFT safe in NFTAMA is an NFT asset on BNB Smart Chain that participates in the internal gaming and economic logic of the project.

A safe has:

- AM weight;
- level;
- age;
- traits: capacity, growth, rarity gene, mint resource;
- class;
- series;
- image;
- value history;
- ownership history;
- mint participation limit.

From a practical point of view, a safe is simultaneously:

- an asset;
- an income tool;
- a game object for development.

## Why a Safe Has Value

A safe is valuable not as an image by itself, but because it:

- provides AM weight;
- increases the user’s share in the system;
- participates in AMA2 distribution;
- can be upgraded with AMA2;
- becomes stronger over time due to age;
- can participate in minting a new safe;
- can be sold to another user.

## Main Safe Parameters

## 1. AM Weight

AM weight is the main characteristic of a safe.

AM weight determines:

- the user’s share among all participants;
- the size of participation in AMA2 distributions;
- the strategic value of the safe inside the portfolio.

## 2. Level

The level of a safe is increased using AMA2.

Each new level:

- increases the safe’s weight;
- strengthens the role of the safe in the portfolio;
- increases the user’s future share in distributions.

A level adds $\Delta A = A_n * 0.08 * q$. The price of level **n** depends on its number and on the safe's weight **A** (`A0` is the start weight of a safe in the current epoch):

$$U_n = \max(30, 100 * 1.5^{(n-1)} * \sqrt{(A / A0)})$$

A safe of the epoch start weight pays the table: `100` AMA2 for the first level, every next one 1.5 times dearer. A heavier safe pays more, but not in proportion to its weight: four times heavier — twice as dear; a lighter one pays less, but never below `30 AMA2`. A level cut by the capacity, which adds only part of its gain, costs the same part of the price. High levels are a goal to save for: they bring weight, extra mint slots (levels 5 and 10) and a stronger rarity gene.

Prices in an epoch with a start weight of 35 AM (not counting that the safe gets heavier with every level):

| Safe weight | Level 1 | Level 5 | Level 10 | 0 → 10 total |
|---|---|---|---|---|
| 15 AM | 65 | 331 | 2,516 | ≈ 7,400 |
| 35 AM | 100 | 506 | 3,844 | ≈ 11,300 |
| 140 AM | 200 | 1,012 | 7,688 | ≈ 22,700 |
| 1,000 AM | 534 | 2,706 | 20,548 | ≈ 60,600 |

Levels 15 and 20 of a safe of the start weight cost `29,193` and `221,683 AMA2`: a Gold safe to level 20 costs several hundred thousand.

The class limits the level: Regular 10, Silver 15, Gold 20. Levels 5 and 10 give one extra mint slot each (mint resource **F** → `F + 1` → `F + 2`). A level never lifts the weight above the safe capacity (see "Safe Traits").

Formula for weight increase on upgrade:

$$\text{NewWeight} = \text{CurrentWeight} + \text{CurrentWeight} * 0.08 * \text{QualityModifier}$$

or in compact form:

$$A_{n+1} = A_n * (1 + g_a * q)$$

where:

- `0.08` = base increase of 8%;
- `QualityModifier` = quality modifier of the safe.

If the quality coefficient does not change, then after **m** consecutive upgrades:

$$A_{n+m} = A_n * (1 + g_a * q)^m$$

This means that strong high-quality safes receive a greater effective effect from upgrading.

## 3. Rarity, Class, and Quality

The class of a safe determines:

- the series;
- the class name;
- the drop chance;
- the range of the weight modifier.

In practice, this means:

- two safes with the same base weight may have different final strength;
- a rarer safe may receive a higher weight multiplier;
- rarity affects not only perception, but also economics.

## 4. Safe Traits: Capacity, Growth, Gene, Forge Resource

Every paid safe is born with four traits. The ranges depend on the class:

| Trait | Regular | Silver | Gold |
|---|---|---|---|
| Capacity **C**, × base weight | 4–8 | 6–12 | 10–20 |
| Growth **g**, % of base weight a day | 0.02–0.06 | 0.04–0.08 | 0.06–0.12 |
| Rarity gene γ, p.p. | 0.5–4 | 2–6 | 4–10 |
| Mint resource **F**, mints | 2–4 | 3–5 | 4–6 |

**The lighter a safe is born, the better its traits.** At birth one number $u \in [0; 1]$ is rolled (0 is the lightest weight of its range, 1 the heaviest), and every trait equals

$$X = X_{min} + (X_{max} - X_{min}) \cdot (1 - u)$$

For a showcase safe **u** is where its quality modifier sits in its class range (a Regular without a range gets a random number); for a forged safe it is where the rolled weight share sits in the 8–22% range. A forged safe inherits 20% of its parents capacity and growth: $\mathbf{X = X_{min} + (X_{max} - X_{min}) \cdot (0.8 \cdot (1 - u) + 0.2 \cdot \bar{s})}$, where $\bar{s}$ is the average place of the parents traits in their ranges.

**Base weight $A_{base}$** is the start weight of the safe: its weight when bought on the showcase (with class, sale, first purchase bonus and coupon) or minted. For safes bought before traits existed, the base weight is their weight when the traits were given out.

**Capacity. $Cap = A_{base} \cdot C \cdot (1 + 0.1 \cdot e) \cdot (1 + \min(0.2; 0.01 \cdot L_{acc}))$**, where **e** is the number of expansions and $L_{acc}$ the owner account level. No level, key, growth or collection bonus lifts a safe above its capacity; weight already gained is never taken away. A full safe takes no new level until it is expanded.

**Capacity expansion:** +10% of the base capacity per step, up to 10 steps. A step costs the full showcase price of the added AM, at least 400 AMA2 (both are project settings): a player who needs more room is a big player.

$$E = \max(400; 0.1 \cdot A_{base} \cdot C \cdot C_0)$$

Example: a Regular 35 AM safe with **C = 6** holds up to 210 AM; a +21 AM step at $\mathbf{C_0 = 14.57}$ **AMA2** costs **max(400; 306) = 400 AMA2**. A Gold 70 AM safe with **C = 15** holds 1,050 AM, and its +105 AM step costs about 1,530 AMA2.

**Growth.** A safe grows every day, linearly, by a percent of its base weight until it is full:

$$A(d) = \min(Cap; A(d - 1) + A_{base} \cdot (g + 0.002 \cdot \min(10, \lfloor L_{acc} / 5 \rfloor)) / 100)$$

| Growth a day | In 1 year | In 3 years |
|---|---|---|
| 0.02% | ×1.07 | ×1.22 |
| 0.04% | ×1.15 | ×1.44 |
| 0.06% | ×1.22 | ×1.66 |
| 0.12% (Gold) | ×1.44 | ×2.31 |

Example: a Regular 35 AM safe growing 0.04% adds 0.014 AM a day and weighs about 50.3 AM after 3 years (without levels and keys). Growth does not depend on levels and keys, so it never compounds and upgraded safes do not run away by time alone.

**A growth key** (lootbox, challenge rewards) adds +0.01% a day forever to any safe with traits; it takes no key slot and never burns; a safe growth never goes above the class maximum +0.05%.

**Rarity gene.** In the forge the parents genes add to the top class chance (see "Minting"). Every 5 safe levels add +0.25 p.p. to its gene.

**Mint resource** is how many mints a safe can take part in: $\mathbf{L_m}$ **= F + level slots** (one at level 5 and one at level 10).

**Series collections.** Different designs of one series held by a player give every safe of that series a share of its base weight as extra weight in the distribution: 3 designs +2%, 5 +4%, 10 +7% (never above the safe capacity; levels and keys do not multiply it; the safe own weight does not change).

**Sale.** All traits, expansions and growth from keys belong to the safe and go to the buyer with it; account level bonuses follow the new owner.

The interface shows +xx% next to the safe weight: how much the weight has grown from the start through levels, keys and growth. The daily gain is shown in the daily summary (e-mail, push notification, Telegram bot), on the cabinet home page and in the daily Telegram group report.

Safes bought before traits existed keep all their weight: they get traits once, their current weight becomes the base weight, and their mint resource is never below the former limit of 3 mints or the mints already made.

## 5. Lifetime

For demo/free safes, a limited lifetime may be set.

Such safes:

- may be issued by the system;
- usually have low weight;
- stop participating in account weight calculation after expiration;
- are used as a user onboarding tool rather than the foundation of the main economy.

## What Can Be Done with a Safe

A safe can:

- be upgraded;
- be sold to another user;
- participate in minting a new safe;
- be held in the portfolio as a long-term asset.

Combining several safes into a new safe is considered as a separate future development scenario for the minting mechanic. Even now, minting a new safe for AMA2 serves as a reinvestment mechanism.

# 5. How Users Obtain Safes

In the current system there are several basic scenarios:

- purchase of an opened safe on the platform;
- auto-buy according to rules;
- purchase of a safe from another user;
- minting a new safe for AMA2;
- receiving a demo/free safe in specific scenarios.

The basic mechanics of the current project economy:

- purchase at opening;
- purchase from another user;
- minting a new safe for AMA2.

# 6. Epochs

## What an Epoch Is

An epoch is a development period of the project within which the parameters of issuance and life of new safes are defined.

An epoch determines:

- the range of total project weight;
- the starting and final weight of a new safe;
- the starting and final price of a new safe;
- the lifetime of an opened safe;
- the interval between safe openings;
- the share of AMA2 directed to distribution;
- the distribution fee.

An epoch is defined by the interval:

$$E_j = [W_j^- ; W_j^+)$$

where:

- $E_j$ is epoch number **j**;
- $W_j^-$ is the lower bound of total system weight;
- $W_j^+$ is the upper bound of total system weight.

## Interpretation of an Epoch

An epoch is a range of overall project development measured by total AM weight.

Epoch boundaries double:

- `40,001–80,000 AM` is epoch 19;
- `80,001–160,000 AM` is epoch 20;
- and so on.

The upper boundary of each next epoch is twice the previous one:

$$W_{j+1}^{+} = 2 * W_{j}^{+}$$

When moving to a new epoch, the following may change:

- frequency of appearance of new safes;
- price parameters;
- weight parameters of new safes;
- conditions of part of the AMA2 economy.

## Why Epochs Matter

Epochs make NFTAMA a dynamic system:

- early participants enter earlier project phases;
- conditions of new safes change over time;
- development of the overall system affects new entry points;
- this supports the scarce model and interest in early participation.

If within an epoch the starting price of a safe is `P0`, and the final price is `P1`, then:

$$P1 \leq P(t) \leq P0$$

And for weight:

$$A1 \leq A(t) \leq A0$$

An epoch defines a range of parameters within which a new safe changes.

## How New Epochs Are Created

Epochs are not set by hand in advance: the next epoch is generated from the previous one when the network weight approaches the current boundary (three epochs ahead are always ready). Rules from epoch **j** to `j + 1`:

- the weight range doubles: $\mathbf{W_{j+1}^{-} = W_j^{+} + 1}$, $\mathbf{W_{j+1}^{+} = 2 * W_j^{+}}$;
- the start weight of a new safe grows 1.5x: $\mathbf{A0_{j+1} = round(1.5 * A0_j)}$;
- the start price grows 1.3x: $\mathbf{P0_{j+1} = floor(1.3 * P0_j)}$;
- the finish weight is 10% of the start: **A1 = max(1, round(0.1 * A0))**;
- the finish price makes 1 AM at the end of a safe's life cost 1.5 times the start price of 1 AM: **P1 = P0 * A1 / A0 * 1.5**;
- the safe life time and the pause between safes grow by 10% (at most 1 hour and 4 hours);
- the daily AMA2 emission share of the pool shrinks by 10%: $\mathbf{e_{j+1} = 0.9 * e_j}$ (at least `0.0001%`).

The point: within a safe's life an early purchase is the better deal per weight; one AM gets slightly cheaper in a new epoch (safes stay affordable), but a share of the network gets about **2 * 1.3 / 1.5 ≈ 1.7** times dearer per epoch, so early players keep their advantage. Emission goes down like a halving and AMA2 becomes scarcer.

| Epoch | Network weight, AM | Safe weight, AM | Price, AMA2 | AMA2 per 1 AM | Life / pause | Emission, AMA2/day* |
|---|---|---|---|---|---|---|
| 19 | 40,001–80,000 | 35 → 4 | 510 → 87.43 | 14.57 → 21.86 | 25 min / 1.4 h | 352 |
| 20 | 80,001–160,000 | 53 → 5 | 663 → 93.82 | 12.51 → 18.76 | 27 min / 1.5 h | 317 |
| 21 | 160,001–320,000 | 80 → 8 | 861 → 129.15 | 10.76 → 16.14 | 30 min / 1.7 h | 286 |
| 22 | 320,001–640,000 | 120 → 12 | 1,119 → 167.85 | 9.32 → 13.99 | 33 min / 1.8 h | 257 |
| 23 | 640,001–1,280,000 | 180 → 18 | 1,454 → 218.10 | 8.08 → 12.12 | 36 min / 2.0 h | 231 |

* with the current pool remainder of ~100.6M AMA2.

# 7. Primary Safe Market

At any given moment, the system may have one opened safe available for purchase. It has a limited lifetime and is created according to the rules of the current epoch.

The user can:

- buy a safe manually;
- use automatic purchase rules.

After purchase:

- the safe is assigned to the user in the database;
- an on-chain mint queue is created;
- the NFT is minted to the user’s internal IN wallet.

## Primary Opening Formulas

If the full lifetime of an opened safe is **T**, and **t** time has passed since opening, then the price of the safe is defined by a linear function:

$$P(t) = P0 - (P0 - P1) * t / T$$

And its weight:

$$A(t) = A0 - (A0 - A1) * t / T$$

Here, not only the absolute price matters, but also the unit price per weight:

$$C(t) = P(t) / A(t)$$

where **C(t)** shows how much the user pays for 1 AM at a specific point in time.

## Balance of Early and Late Entry

The price and the weight of an open safe fall linearly, but the weight falls faster. The epoch finish values are chosen so that

$$C(T) = P1 / A1 = 1.5 * P0 / A0 = 1.5 * C(0)$$

The younger the safe, the better the deal per AM: at the very start one AM is the cheapest, by the middle of the safe's life it costs about 5% more, at the end 1.5 times more. The safe itself gets cheaper, though, so a small budget can afford it. The player chooses: catch the safe at once at the best price per weight, or wait for a cheaper but lighter and less profitable safe, with the risk that someone buys it first.

The showcase shows the live benefit of buying next to the price and the weight: how much cheaper one AM is now than at the end of the safe's life:

$$B(t) = 1 - C(t) / C(T)$$

Right after the safe opens **B ≈ 33%**, by the end 0%.

The price of a bought safe does not vanish: it is shared among all safe holders by weight (minus the epoch distribution fee).

## Why Entry Timing Matters

Buying at opening is important because:

- the parameters of the safe are set by the current epoch;
- an early purchase may provide strong weight at an early stage;
- strong early weight provides more time to accumulate AMA2;

- as the system grows, such early entries become more limited.

The long-term effect of early entry is estimated by:

$$V_{long} \approx A_{entry} * H$$

where:

- $A_{entry}$ is the weight with which the user entered the system;
- **H** is the length of the investment horizon.

The earlier and stronger the entry, the greater the long-term effect.

### Sales

On two random days a month (not in a row) the showcase holds sales. On such a day every next safe becomes a sale safe with a 30% chance and gets a weight multiplier of x2 or x3.

- Whether the next safe is a sale is decided when the previous one leaves the showcase, so players learn about a sale in advance: a mark on the showcase, an e-mail (for those with sale e-mails on; by default they are on for active players — signed in within the last 30 days or owning bought safes) and a push in the app.
- The multiplier applies if the safe is bought within the first 10% of its showcase life: $t_{purchase} - t_{appearance} \le 0.1 * T_{life}$. Then $w = w_{showcase}(t) * m$. The price does not change.
- Any way of buying works: the site, the mobile app, an auto-buy rule.
- It is purchase weight, not a modifier key: a sale safe can still take a key, and weight-growth challenges count this weight.
- The days, the chance, the multipliers and the window are set by the administration; the current rules are in the "Sales" guide topic.

## 8. Auto-Buy Rules

The project implements three basic purchase strategies:

1 Stepwise price increase purchase

- the user defines the increase step, price limit, and number of attempts in advance.

2 Purchase at the opening price

- the system attempts to buy the safe immediately when it opens.

3 Purchase after price decrease

- the system waits until the price falls to the selected share of the starting price.

This turns safe purchasing from a manual process into a strategic instrument. The user can build an individual entry tactic rather than simply wait for a convenient moment.

### Auto-Buy Formulas

If the user chooses stepwise price increase with step **s**, number of steps **n** and a cap **L** (a share above the starting price, $L \geq s$), the strategy's bids are $\mathbf{P0 * (1 + s)^k}$ for **k = 1..n**, and the maximum bid is the largest of them that does not exceed the cap:

$$P_{bid} = \max \{ P0 * (1 + s)^k : k \leq n, P0 * (1 + s)^k \leq P0 * (1 + L) \}$$

The highest bid wins and the winner pays its first step above the best bid of the remaining bidders (a second-price auction). At equal bids the higher account level wins, at equal levels the order is random (the same for the starting-price and price-drop rules); a bidder whose balance is short drops out without raising the price for the next one. The number of rules is limited by the account level: **2 + floor(n / 2)**.

If the user wants to enter only when the price falls to the share **q** of the starting price, the target entry price is:

$$P_{target} = P0 * q$$

In a linear price decrease model, the strategy trigger time is:

$$t_{target} = T * (P0 - P_{target}) / (P0 - P1)$$

If $P_{target} \leq P1$, the price never gets that low and the rule buys at the minimum price in the last minute of the safe's life.

For comparing strategies, a useful metric is:

$$R = A^* / P^*$$

where:

- A* is the expected entry weight;
- P* is the expected entry price.

The higher **R**, the more weight the user receives per unit of invested capital.

## 9. Secondary Market

NFTAMA has a secondary market between users.

The user can:

- sell a safe;
- send a purchase request;
- agree on deal terms;
- complete the transfer of the safe to a new owner.

This means that safes receive not only initial value but also accumulated value tied to:

- weight;
- class;
- age;
- level;
- history;
- potential future yield.

The market value of a safe on the secondary market can be represented as:

$$V_{safe} = f(A, q, age, lvl, y)$$

where value depends on:

- weight;
- quality;
- age;
- level;
- expected future yield.

# 10. AMA2 Tokenomics

## What AMA2 Is

AMA2 (Amatu) is the main token for development and internal settlement within NFTAMA.

It is used for:

- safe upgrades;
- account upgrades;
- minting new safes;
- internal purchases;
- withdrawals;
- internal exchange with BNB (at the PancakeSwap rate; the 2% exchange fee stays with the project and covers the pool fee and slippage);
- reward and distribution economics.

## Total Supply

AMA2 token parameters:

- total token amount: `141,000,000`;
- decimals: 18;
- total amount in smallest units: **141,000,000 * $10^{18}$**.

This means that in user-facing representation, the total project supply is:

```
141 000 000 AMA2
```

Additional token issuance is not provided for.

This means that the AMA2 model is limited in supply and has no infinite emission.

In short form:

$$S_{total} = 141\ 000\ 000$$

## Why AMA2 Is Needed

AMA2 performs several roles simultaneously in the ecosystem:

- reward token;
- reinvestment token;
- development token;
- token for internal settlements.

It is not an auxiliary token, but the central element of the account growth mechanic.

## What AMA2 Is Spent On

## 1. Safe Upgrades

The level price is the table **100 * 1.5^(n-1)** AMA2 times `√(A / A0)` — the square root of the safe's weight against the epoch start weight, at least 30 AMA2 (see section 4, "Level"). AMA2 spent on levels leave circulation.

## 2. Account Level

The account level is reached with experience (section "Account Experience and Level"), and the missing experience of the next level can be bought for AMA2:

$$\text{Price} = (E_L - e) \cdot (1 + 0.03 \cdot L) \cdot k$$

where $E_L = 30 \cdot L^2$ is the experience of level **L**, **e** the experience already earned, and **k** how many times the showcase price of 1 AM is higher now than when experience started. A whole level 10 costs **3,000 · 1.3 = 3,900 AMA2**, level 20 — 19,200 AMA2, level 50 — 187,500 AMA2; the way from 0 to 100 is about 33 million AMA2. Usually a player buys only the rest, and the price falls with every point earned.

## 3. Minting a New Safe

The mint price is tied to the start price of a new safe in the current epoch $P_e$, so it rebalances itself when the epoch changes:

$M(k) = s \cdot P_e \cdot k / 2 \cdot \max(1, (V / k / W_e)^{0.75})$, where **s = 0.3**, **k ≥ 2**, **V** is the weight the parents pass on (see section 12)

A mint from two safes costs 30% of the epoch start price, and every safe in the forge adds half of that. When the parents are heavier than the epoch start weight $W_e$ on average, the price grows with the weight they pass on.

Example for an epoch with $P_e$ = **510 AMA2** and parents no heavier than the start weight:

- 2 safes -> 153 AMA2
- 3 safes -> 229.5 AMA2
- 4 safes -> 306 AMA2
- 6 safes -> 459 AMA2

## 4. Buying Safes from Users

Internal P2P deals use AMA2 as the settlement price.

## 5. AMA2 Withdrawal

Withdrawal requires not only an AMA2 balance, but also BNB for gas.

## How AMA2 Are Distributed

The main logic of AMA2 accrual is based on the user’s share of weight.

First, the user’s share in total weight is determined:

$$D_i = A_i / A_{sum}$$

## Total Pool Formula per Cycle

$$Q = S * r$$

where:

- **Q** is the amount of AMA2 directed to distribution per cycle;
- **S** is the amount of AMA2 available for distribution;
- **r** is the share directed to distribution in one cycle.

## Formula After Fee

$$Q_{net} = Q * (1 - f)$$

Base fee rate (the epoch's $\mathbf{comission_{ama}}$):

$$f = 0.10$$

## User Formula

$$R_i = Q_{net} * D_i$$

or in full form:

$$R_i = S * r * (1 - f) * A_i / A_{sum}$$

If the participant was invited, the referral share is taken from their accrual, and the referrer's level bonus is paid by the project on top:

$$R_{childnet} = (1 - \alpha) * R_{child}$$

$$R_{ref} = (\alpha + \beta) * R_{child}$$

where **α** is the referral rate (9%) and β is the referrer's account level bonus (0.2 pp per level, up to 3 pp).

## A showcase safe price and the fresh pool

The emission (**Q** above) goes to all holders by weight. The price **P** of every safe bought on the showcase is split differently:

- **f = 10%** to the project (the epoch fee);
- 50% to all holders by weight share $\mathbf{D_i}$;
- 40% to the **fresh pool**: the weight added in the last 30 days.

A player's fresh weight $\mathbf{F_i}$ is the sum over the last 30 days of:

- the start weight of safes bought on the showcase (any way: site, app, auto-buy rule);

- the weight of safes from the forge;
- 50% of a safe level's gain;

while the safe stays with that player. The fresh pool's part:

$$P_{fresh} = \min(0.4 * P, c * T * F_{sum} / (A_{sum} + c * F_{sum}))`, where `T = P * (1 - f)`, `c = 5$$

— a fresh 1 AM gets at most **c = 5** times a usual one; the rest goes to all holders. A player's credit from one purchase:

$$R_i = (T - P_{fresh}) * A_i / A_{sum} + P_{fresh} * F_i / F_{sum}$$

So every purchase pays back at once: a newcomer, whose weight is almost all fresh, gets a share of every safe bought after them in the first month, and to stay in the pool one buys again. The parameters (40%, 30 days, **c**, 50% of a level) are project settings.

## The first safe and coupons

A player's first safe bought on the showcase weighs ×1.5 — that is its start weight — and costs 20% less. Coupons from the safe of luck (section 11): −20% off a showcase safe for an hour and +15% weight to the next showcase safe within 7 days; they work with any way of buying. Discounts never add up: when a player has both the first-safe discount and a coupon, the bigger one is taken, and a smaller coupon is kept for the next purchase.

## The showcase for newcomers

Every 4th showcase safe can, for its first 10 minutes, be bought only by players with fewer than 5 paid safes; auto-buy rules do not buy it during that time. After that it is sold as usual. This way new players get safes at their highest weight even when big players buy out the showcase with rules. The numbers (4, 10 minutes, 5 safes) and the quota itself are switched on and set by the administration.

## Simple Distribution Example

Assume:

- total system weight = 10 000 AM;
- participant weight = 1 000 AM;
- participant share = 10%;
- 500 AMA2 goes to the cycle after fee.

Then:

$$UserAMA2 = 500 * 10\% = 50\ AMA2$$

If the participant’s weight increases to `1 500 AM` with the same total system weight:

$$\mathbf{UserAMA2 = 500 * 15\% = 75\ AMA2}$$

Conclusion:

- growth of weight directly increases the size of the main reward.

$$\mathbf{R_i \propto A_i}$$

under otherwise equal conditions.

### Other Sources of AMA2

In addition to the main distribution, the project already contains additional sources:

- the safe of luck (a little AMA2, keys, demo safes, coupons);
- bonus codes;
- referral accruals;
- separate internal accruals;
- prospective LP incentives.

### Buying AMA2 with Telegram Stars

In the @nftama_bot AMA2 can be bought with Telegram Stars; the account is connected to the bot: the “Connect” button in the bot opens the site, which sends you back to the bot after sign-in (or use the link in the profile). Step-by-step instructions are in the guide, “Telegram bot and Stars”. The rate is:

$$\mathbf{AMA2 = Stars \times P_{star} / P_{AMA2} \times (1 - f_{stars})}$$

where $\mathbf{P_{star}}$ is the dollar value of a Star (a project setting, 0.013 $ by default — what the project receives per Star on withdrawal), $\mathbf{P_{AMA2}}$ is the AMA2 price on PancakeSwap and $\mathbf{f_{stars}}$ the project fee (0 by default). The rate is fixed in the invoice for 15 minutes, AMA2 go to the internal balance right after payment, a payment is credited once; if crediting fails the Stars are refunded. Packages are from 50 to 2500 Stars, with a daily limit per player. Only an account with a confirmed e-mail can buy AMA2.

For every AMA2 sold for Stars the project buys back the same amount on PancakeSwap with the hot wallet's BNB: internal balances are backed by tokens and the purchases support the price. The buyback goes in parts every 10 minutes, each part moving the price by at most 2% (below the price stabiliser threshold); small amounts are collected up to the minimum buyback or a day.

## 11. Safe of Luck and Account Development

Once an hour a player can open the **safe of luck** (a lootbox): prizes flick through the safe's window, the player presses “Stop” and the safe opens on the prize the server chose by the odds:

| Prize | Chance |
|---|---|
| AMA2 to the balance: ×1 / ×2 / ×5 of the account base reward | 12% / 6% / 2% |
| Modifier key ×1.01 / ×1.02 / ×1.03 / ×1.05 / ×1.10 | 12% / 11% / 6% / 3% / 1% |
| Level key: +1 level to any safe for free (valid 30 days) | 4% |
| Growth key: +0.01% of the base weight a day to any safe forever | 5% |
| Coupon −20% off a showcase safe (60 minutes) | 6.5% |
| Coupon +15% weight to the next showcase safe (7 days) | 6.5% |
| Demo safe with ×2 weight | 15% |
| Nothing | 10% |

In total: 20% AMA2, 55% keys and coupons, 15% a demo safe, 10% nothing. Keys and coupons are useful only to a safe owner (one key per safe) — the safe of luck pushes towards a purchase instead of replacing it, and farming AMA2 with many accounts makes no sense.

Conditions:

- a confirmed email;
- at most once an hour;
- a daily attempts limit.

The daily limit:

$$N_{day} = \min(N_0 + \min(p_7, 3), N_{max}) + \min(\text{floor}(L / 5), 2)$$

where:

- $N_0 = 3$ — attempts every day;
- $p_7$ — the player's showcase and forged safes of the last 7 days (each +1, up to +3);
- $N_{max} = 6$;
- $L$ — the account level (+1 for every 5 levels, up to +2).

At most 8 attempts a day. The account weight no longer gives attempts — purchases do.

The AMA2 prize of each next attempt of the day is smaller: the $n$-th attempt pays the share $k_n = \max(d^{(n-1)}, 0.1)$, where $d = 0.75$ (1, 0.75, 0.56, 0.42…). The expected AMA2 per day for the base reward $F$ and $N_{day}$ attempts:

$$F_{day} = 0.34 * F * \Sigma_{n=1..N_day}\, k_n$$

where 0.34 is the average AMA2 multiplier of one attempt (0.12·1 + 0.06·2 + 0.02·5). With **F = 0.6** and 8 attempts that is about 0.73 AMA2 a day — 3.5 times less than the former wheel gave with 16 attempts.

## Account Experience and Level

The account level (0 to 100) is reached with **experience**. Level **L** needs

$\mathbf{E_L = 30 \cdot L^2}$ experience

(level 1 — 30, 10 — 3,000, 20 — 12,000, 100 — 300,000). A ring around the level badge fills up with experience.

### What gives experience:

| Action | Experience |
|---|---|
| AMA2 spent in the game (a showcase safe, a mint, a safe level, a capacity expansion, tempering a key) | 30% of the sum |
| Buying a safe from a player | 10% of the price, at most 100 a day |
| AMA2 top-up (Stars, USDT, BNB→AMA2, transfer to the wallet) | 10% — only above the highest "brought in minus taken out" so far |
| AMA2 lock | 1% of the amount for every 30 days, when it ends |
| Mint / challenge reward / a friend bought the first safe | +20 / +25 / +25 |
| Daily login / 7 days in a row / a key / safe of luck | +3 / +15 / +5 / +1 — only with a paid safe, at most 20 a day |
| Once: first safe, e-mail, Telegram, app sign-in | +20 / +10 / +10 / +10 |

Experience mostly comes from AMA2 really spent, so it cannot be farmed with many empty accounts. **The missing experience can be bought** (section "Account Level" above): a point costs (1 + 0.03 · L) AMA2 × **k**, growing with the level and the showcase price, and only the rest is paid. Without buying: an active player (~575 experience a month) reaches level 9 in half a year and 14 in two years, a big one (~2,300) — 18 in a year.

The level raises:

- cashback (starting at 3): $\mathbf{B_{n+1} = \min(10,\ B_n \cdot 1.05)}$ — at most 10%;
- the base free AMA2 reward (starting at 0.6): $\mathbf{F_{n+1} = F_n \cdot (1 + h_n)}$, where $\mathbf{h_n}$ = +20% (levels 1–5), +8% (6–10), +5% (11–15), +1.5% (16–20), +0.5% (21+).

Besides that, every account level gives perks (each has a ceiling):

- AMA2↔BNB exchange fee: −0.1 pp per level (never below half of the base);
- mint discount: 1.5% per level, up to 30%;
- modifier keys: +1 day of life (up to +30) and +2% to the bonus cap (up to +40%) per level;
- auto-buy rules: **2 + floor(n / 2)**, at most 20; at equal price or moment the higher account level wins;
- referrer bonus: +0.2 pp per level on top of the referral share, up to +3 pp; the project pays it;

- safe of luck: +1 daily attempt for every 5 levels, up to +2;
- rare safe chance: up to +13 pp to Silver/Gold classes (**13 · (1 − e^(−n/12))**);
- safes: +1% capacity per level (up to +20%), +0.002% growth a day per 5 levels (up to +0.02), +0.1 pp gene in the mint (up to +2).

AMA2 spent on buying levels leave circulation.

## AMA2 Lock

A player can lock AMA2 for 30 or 90 days. The tokens leave circulation and come back in full when the lock ends, together with a modifier key; a lock cannot be opened early (up to 5 locks at a time).

- 30 days: from 300 AMA2 an eternal x1.10 key, from 1,000 x1.20, from 3,000 x1.25;
- 90 days: from 300 AMA2 x1.30, from 1,000 x1.60, from 3,000 x2.00, from 10,000 x2.50 (hot keys with a lifetime).

The lock reduces the AMA2 supply in circulation and the sell pressure, and the player gets a key without losing tokens.

## Modifier Keys and Challenges

A modifier key multiplies the AM weight of one safe. A key is used once and a safe takes one key. Keys are given for challenges (16 of them, listed below), for confirming the e-mail, for the AMA2 lock and as project prizes.

Since AMA2 rewards are shared by weight, a key creates no new tokens: it moves part of the daily rewards to active players. So keys are limited by three rules.

## 1. Bonus cap

A key with multiplier **m** adds to a safe of weight **w** at most two fresh safes of the current epoch:

$$\Delta w = \min(w * (m - 1), (m - 1) * 2 * W_e)$$

where $W_e$ is the start weight of a safe in the current epoch. A key is worth the same on a fresh safe and on a very heavy one: an x7.7 key on a 2,500 AM safe adds 469 AM with $W_e = 35$, not 16,750 AM. On top of that, a key never lifts the weight above the safe capacity.

## 2. Lifetime

The stronger the key, the sooner it cools down and burns if it is not applied:

$T(m) = \text{clamp}(\text{round}(30 * (m - 1)^{-0.6}), 7, 60)$ days

- x1.3: 60 days;
- x2: 30 days;
- x3: 20 days;
- x4.5: 14 days;
- x7.7: 10 days.

Keys up to x1.25 are eternal: they never burn. A key is valid until the end of its last day; the player is reminded 3 days before it burns.

## 3. Tempering ("Make eternal")

Any unused hot key can be tempered into an eternal one (the "Make eternal" button on the key card). Tempering keeps a fifth of the bonus:

$$m_{eternal} = clamp(1 + (m - 1) * 0.2, 1.05, 1.25)$$

For example, x2 becomes an eternal x1.2, x3 and stronger an eternal x1.25. Tempering cannot be undone. The account level strengthens keys: +1 day of life and +2% to the bonus cap per level (cap bonus up to +40%). This is the player's choice: use a strong key now, wait for a heavier safe and risk the key burning, or temper it into a weak but eternal key.

Tempering is paid: it costs a quarter of the showcase price of the weight the eternal key can add:

**price = max(10, 0.25 * ($m_{eternal}$ - 1) * 2 * P0)** AMA2

where P0 is the start showcase price of the current epoch (the 25% share is a project setting). At **P0 = 510**: x1.3 → eternal x1.06 costs 15 AMA2; x2 → x1.2 costs 51 AMA2; x3 and stronger → x1.25 costs 64 AMA2; the minimum is 10 AMA2. The AMA2 are taken from the main balance and stay with the project; without the amount the key is not tempered.

## 4. Level key

A special kind of key: it does not multiply weight, it gives +1 level to any of the player's safes for free, like an upgrade for AMA2 without paying (the weight gain follows the upgrade rules, half of it is fresh weight). The level key lives in Modifiers with the other keys, is valid for 30 days, cannot be made eternal and does not take the weight key slot: it can be applied to a safe that already has a key. The safe must be below the top level of its class. Level keys drop from the Safe of luck (4%) and from challenge rewards: 10% of rewards (a project setting) come as a level key instead of a weight key.

## 5. Growth key

Adds +0.01% of the base weight a day to the growth of any safe with traits, forever (see "Safe Traits"). It does not multiply the weight at once, takes no weight key slot and never burns; a safe growth goes no higher than its class maximum +0.05%. Growth keys drop

from the Safe of luck (5%) and from challenge rewards: another 10% of rewards (a project setting) come as a growth key.

Challenge rewards are calibrated through the cap: a key's value is expressed in fresh safes ($\Delta w_{max} / W_e$).

- daily logins, day 7: x1.3–1.5, 45–60 days, cap 0.6–1.0 fresh safe;
- daily logins, day 15: x1.6–2.0, 30–41 days, cap 1.2–2.0 fresh safes;
- daily logins, day 30: x2.2–3.0, 20–27 days, cap 2.4–4.0 fresh safes;
- AM weight growth: x3.0–4.0, 16–20 days, cap 4.0–6.0 fresh safes;
- e-mail confirmation: x2.0–2.5, 24–30 days, cap 2.0–3.0 fresh safes.

Daily logins: come every day without gaps; a missed day starts the streak again from day 1, keys already earned stay with the player. The streak can be repeated while the challenge runs.

## All challenges

Challenges have their own page, "Challenges". The thresholds and rewards below are the starting ones and may be changed by the administration; the current values are always shown on the page.

| Challenge | Condition | Key |
|---|---|---|
| Daily logins | a login streak without gaps, keys on days 7, 15 and 30 | x1.3–3.0 |
| AM weight growth | grow the weight by 20% of your weight at the start of the attempt (at least 35 AM) in 7 days — the same share for everyone instead of one target for big and small players | x3.0–4.0 |
| Newcomer | within 7 days confirm the e-mail, set a withdrawal wallet and buy the first safe (only before the first purchase) | x1.4–1.6 |
| Early bird | 3 times a week buy a safe yourself (not by a rule) within 5 minutes after it appears | x1.2–1.3, once a week |
| Buyer of the week | buy 2 safes from the showcase in a week | x1.2–1.4, once a week |
| Common goal | all players together buy 40 safes in a week, a key for everyone who bought at least one | x1.1–1.2, eternal, once a week |
| Smith | 3 mints or one mint of a Silver/Gold safe | x1.3–1.6 |
| Collector | get safes of every paid class while the challenge runs | x1.8–2.2 |
| Friend in the game | an invited friend buys the first safe: a key for both, up to 3 keys for the inviter | x1.3–1.5 |
| Trader | a deal on the secondary market for at least 50 AMA2 | x1.15–1.2, eternal |

| Challenge | Condition | Key |
|---|---|---|
| Holder | 14 days in a row with at least 300 AMA2 and no AMA2 withdrawal | x1.15–1.2, eternal |
| Veteran | a safe reaches 180 days while the challenge runs | x1.2–1.3 |
| Autopilot | a purchase by an auto-buy rule | x1.15–1.2, eternal |
| Luck | the safe of luck 7 days in a row, up to 2 times | x1.15–1.2, eternal |
| Mobile | the mobile app 7 days in a row | x1.3–1.5 |
| Buyer's ladder | the 1st, 3rd, 5th, 10th and 20th showcase safe; steps passed before the challenge started are not rewarded | 1st x1.25–1.35, 3rd x1.45–1.55, 5th x1.7–1.8, 10th x2.0–2.2 and a level key, 20th — a growth key |
| Comeback | a player who bought safes but none for 60 days buys a showcase safe within 14 days (from the invitation letter or from coming back); once | x1.5–1.8 |

The rewards are set from the game data so that all challenges together add about 3–4% of the network weight a month: a key redistributes the daily rewards and creates no tokens, and the big keys go to rare achievements.

A week is Monday to Sunday; weekly challenges give a key once a week and reset on Monday night.

Until 11 October 2026 there was the "Race of the week" — keys for the top 10 players by AM weight growth in a week. It was closed after the results of its last week: the reward for the biggest absolute growth went to the biggest players and made their share bigger still. Instead, the rewards now go to the growth of new paying players: the "Buyer's ladder", the "Comeback", the first-safe discount, the showcase for newcomers and "AM weight growth" in percent of the player's weight.

Challenges do not influence each other:

- every challenge has its own progress; one action can count in several of them, but the reward of one never moves or resets another;
- weight added by keys is not growth in "AM weight growth", so a key from one challenge never helps to complete another;
- a failure while counting one challenge does not affect the others or the action itself (a purchase, a mint, a deal).

While a challenge runs, a broken attempt starts again; finished and broken attempts are kept in the archive.

## 12. Minting New Safes

Minting a new safe for AMA2 is one of the key reinvestment mechanisms of NFTAMA.

## Requirements

To mint, the user needs:

- at least 2 own safes;
- sufficient AMA2 balance;
- sufficient BNB balance for gas.

## Limitation on Source Safes

The same safe can be used in minting a limited number of times — its mint resource:

$$L_m = F + \text{level slots}$$

where **F** is the mint resource of the safe (Regular 2–4, Silver 3–5, Gold 4–6), and safe levels 5 and 10 open one more slot each.

## What Happens During Minting

1. The user selects at least 2 own safes.
2. The system checks balances and limits.
3. AMA2 and BNB are deducted.
4. A new safe is created in the system.
5. The participation counter of the source safes is increased.
6. The new NFT is placed in the on-chain mint queue.

## How a New Safe Is Formed

The new safe:

- receives a new class;
- receives a new image;
- inherits part of the parents total weight: heavy fresh safes forge a heavy safe;
- gets its own traits (capacity, growth, gene, mint resource): the lighter the rolled weight, the better they are; 20% of capacity and growth comes from the parents;
- is minted within the current epoch.

This makes minting a reinvestment mechanism rather than a simple purchase of a predefined item.

## Minting Formulas

**Price.** $M(k) = s \cdot P_e \cdot k / 2 \cdot \max(1, (V / k / W_e)^{0.75})$, $s = 0.3$, $P_e$ is the start price of a new safe in the current epoch, **V** is the weight passed on (below). Plus gas **G** in BNB.

**Safe rarity.** $r = (m_{min} + m_{max}) / 2$ — the middle of its class AM weight multiplier range (Regular 1, Silver 1.2, Gold 2.5).

**Safe power.** $p = \text{clamp}(A / W_e; 0.25; 2)$ — the safe weight relative to the epoch start weight $W_e$.

**Freshness.** $\varphi = 1 - 0.15 \cdot m$, where **m** is how many mints the safe has already taken part in.

**Forge luck.** Contribution $c_j = r_j \cdot \sqrt{p_j} \cdot \varphi_j$. Contributions are sorted descending, the first 5 count in full, the rest at 0.5 ("overheat"). Then $\lambda = \Sigma\, c / 2 \cdot x$, where $x = 1.2^{(K-1)}$ for **K** different classes ("alloy") or $x = 0.85$ for 3+ safes of one class ("monolith"). Two fresh regular safes with the epoch start weight give $\lambda = 1$.

**Class chance.** $P(c) \propto d_c \cdot r_c^{(\lambda - 1)}$, where $d_c$ is the usual drop chance (with the account level bonus). At $\lambda = 1$ the odds equal a normal purchase; as luck grows, the rarest classes gain the most.

**Rarity genes.** Then $\Gamma = \min(40; \Sigma' \gamma_j \cdot \varphi_j + \min(2; 0.1 \cdot L_{acc}))$ p.p. is added to the top class chance ($\Sigma'$: the five strongest genes in full, the rest by half), the other classes shrink in proportion, and the top class never goes above 85%. Example: two average Regular safes ($\gamma = 2.25$) lift Gold from 4.3% to 8.8%.

The forge shows the weight of the new safe only as a range: the exact weight is a surprise.

**Weight.** Each parent passes on $v_j = \min(A_j; 4 \cdot W_e) \cdot \varphi_j$, in total $V = \Sigma\, v_j$. The new safe weighs $A_{new} = V \cdot \max(u_1 \ldots u_{(k-1)}) \cdot m_c$, where $u_i \in [0.08; 0.22]$ and $m_c$ is the class multiplier. The new safe gets 8–22% of the parents total weight: heavy fresh safes forge a heavy safe, and more safes mean more weight and more rolls ($E[\max] = 0.08 + 0.14 \cdot (k-1) / k$). Two fresh safes of the epoch start weight give as many AM per AMA2 as a fresh showcase safe.

Examples for $P_e = 510$, $W_e = 35$, Regular/Silver/Gold chances 1000/100/50:

| Safes in the forge | Price, AMA2 | Luck λ | Regular / Silver / Gold | Expected weight |
|---|---|---|---|---|
| 2 Regular of 35 AM | 153 | 1.00 | 87% / 9% / 4% | ≈11 AM |
| 2 Regular of 5 AM | 153 | 0.50 | 89% / 8% / 3% | ≈2 AM |
| 2 Regular of 35 AM, each used in 2 mints | 153 | 0.70 | 88% / 8% / 3% | ≈8 AM |
| 3 Regular of 35 AM — monolith | 229.5 | 1.27 | 86% / 9% / 6% | ≈20 AM |
| Regular + Silver of 35 AM — alloy | 153 | 1.32 | 85% / 9% / 6% | ≈12 AM |
| Regular + Silver + Gold of 35 AM | 229.5 | 3.38 | 63% / 10% / 28% | ≈26 AM |
| 2 Gold of 70 AM | 257.3 | 3.54 | 60% / 10% / 31% | ≈31 AM |

| Safes in the forge | Price, AMA2 | Luck λ | Regular / Silver / Gold | Expected weight |
|---|---|---|---|---|
| 3 Gold of 70 AM — monolith | 386 | 4.51 | 41% / 8% / 51% | ≈65 AM |
| 2 Gold + Silver of 70 AM — alloy | 386 | 5.26 | 27% / 6% / 67% | ≈73 AM |
| 4 Regular of 140 AM — monolith | 865.5 | 2.40 | 76% / 10% / 14% | ≈127 AM |
| 10 Regular of 5 AM — monolith + overheat | 765 | 1.59 | 84% / 9% / 7% | ≈12 AM |

The odds in the table are without rarity genes. Each safe can take part in at most $\mathbf{L_m}$ mints (its mint resource), so big batches burn the mint resource of many safes.

# 13. Demonstration Example for 3 Years

Below is a simple calculation example.

## Conditions

Assume:

- daily distribution: `250 AMA2`;
- horizon: **3 years = 1095 days**;
- total project weight grows linearly from `5 000 AM` to `45 000 AM`;
- all `10` safes in each scenario were purchased on the same day, the first day of the experiment;
- scenario A: the user has `10 safes of 1 AM` each, total `10 AM`;
- scenario B: the user has `10 safes of 10 AM` each, total `100 AM`;
- after 3 years, the AMA2 price reaches `0.3 USD`.

## Formula

Total system weight on day **d**:

$$\mathbf{W(d) = 5000 + d * (45000 - 5000) / 1094}$$

Since all safes were purchased on the first day of the experiment and grow at the average Regular growth **g = 0.04%** a day, the user weight is:

$$\mathbf{A_i(d) = A_0 * (1 + 0.0004 * d)}$$

where $\mathbf{A_0}$ is the user starting total weight (section "Safe Traits"; the capacity is not reached in this example).

User daily reward:

$$R(d) = 250 * A_i(d) / W(d)$$

Total for 3 years:

$$R_{tot} = \Sigma\ R(d),\ d = 0..1094$$

If the AMA2 price at the calculation point is denoted as $P_{ama}$, then the total value of the accumulated amount:

$$V = R_{tot} * P_{ama}$$

## Calculation Result

## Scenario A: 10 Safes of 1 AM Each

Starting total user weight:

```
10 AM
```

Taking into account the daily growth, by the end of the 3-year horizon their total weight rises to about `14.38 AM`.

Total reward over 3 years:

$$\approx 172.23\ AMA2$$

If the AMA2 price after 3 years is `0.3 USD`, then the value of the accumulated amount:

$$172.23 * 0.3 \approx 51.67\ USD$$

## Scenario B: 10 Safes of 10 AM Each

Starting total user weight:

```
100 AM
```

Total reward over 3 years:

$$\approx 1722.28\ AMA2$$

If the AMA2 price after 3 years is `0.3 USD`, then the value of the accumulated amount:

$$1722.28 * 0.3 \approx 516.69\ USD$$

### Conclusion from the Example

With the same number of safes, the result is determined not by the number of items as such, but by the total AM weight.

If the portfolio weighs `10 times` more:

- the user’s share in the system on each day is approximately `10 times` higher;
- therefore the total AMA2 accumulation over the period is approximately `10 times` higher;
- and with the same growth, the daily gain scales across the entire portfolio simultaneously.

This confirms the key idea of the project:

- early entry into high-weight safes provides a long-term advantage;
- the earlier a user builds quality AM weight, the longer that weight works for them;
- over the same horizon, it is weight, not the “number of cards”, that determines the result.

Note:

- this is a calculated scenario under specified conditions;
- it is not a promise of profitability;
- in reality, distributions, epochs, total system weight, AMA2 exchange rate, and user activity will differ.

## 14. Why the Project Stimulates Long-Term Holding

NFTAMA is built as a system that is favorable for long-term participation.

Long-term holding is driven by:

- growth of safe age;
- growth of weight through upgrades;
- ability to receive AMA2 proportionally to weight;
- use of safes for minting new ones;
- growth of account utility;
- long-term effect of early entry.

NFTAMA is a portfolio development system, not an NFT venue for rapid resales.

## 15. Liquidity and Market

## Where AMA2 Is Traded

In the current project architecture, AMA2 is used in a pair with BNB on PancakeSwap.

Swap link used in the project:

```
https://pancakeswap.finance/swap?inputCurrency=0x126221D7635388f4aEb85c71A4701E09640EEe09&outputCu
```

## Liquidity Development Strategy

Project liquidity development strategy:

- expand liquidity in the main `AMA2/BNB` pair;
- gradually increase the volume of locked liquidity;
- stimulate organic use of AMA2 within the platform so that demand is formed not only speculatively;
- in the future, consider adding new pools and infrastructure partners.

## How the Project Uses Fees

The project keeps:

- the internal AMA2↔BNB exchange fee, 2% (less with the account level);
- the on-chain AMA2 withdrawal fee, 1.5% of the amount; the minimum withdrawal is 100 AMA2;
- 10% of every AMA2 distribution (the epoch fee).

All fees are recorded in the project revenue ledger. Monthly, half of the BNB exchange fees with the matching AMA2 are added to the PancakeSwap AMA2/BNB pool and the LP tokens are locked; 20% of the AMA2 taken out of circulation by sinks (levels, fees) are burned on-chain by sending them to `0x…dEaD`; burned tokens are shown on the token page.

## What TVL Is

TVL (`Total Value Locked`) is the total value of assets locked or used in the ecosystem or its pools.

For NFTAMA, TVL serves as a metric of:

- user trust;
- economic depth;
- liquidity maturity;
- scale of capital involvement in the ecosystem.

Project TVL is formed by:

- liquidity in AMA2 pairs;
- user balances;
- expansion of the internal economy;
- future gaming and banking mechanics.

# 16. Technical Architecture

## Why BNB Smart Chain

The project is architecturally tied to BNB Smart Chain:

- safes are minted in the BSC environment;
- AMA2 is linked to a BNB pair;
- BNB is used as gas;
- users operate through BSC-compatible wallets and transfers.

## How the Website and Blockchain Are Connected

The connection has two levels:

1 Backend and database

- store game state;
- maintain internal balances;
- calculate weight, history, rules, queues, and interface logic.

2 Smart contracts and on-chain operations

- mint NFTs;
- provide on-chain storage of the token and NFT;
- handle withdrawals and transfers of assets.

## Internal Infrastructure

The project uses:

- the user’s internal IN wallet;
- an external OUT wallet;
- queues for BNB, AMA2, and NFT withdrawals;
- NFT mint queues;
- background commands for calculations, distributions, and top-ups.

This means that part of the project logic is centralized:

- game mechanics;
- weight calculation;
- distributions;
- queues;
- part of financial scenarios.

In NFTAMA, on-chain and off-chain components work together.

## Smart Contracts

The project is based on two key smart contracts:

- AMA2 contract;
- NFT contract.

## AMA2

Current AMA2 address:

```
0x126221D7635388f4aEb85c71A4701E09640EEe09
```

Explorer link:

```
https://bscscan.com/address/0x126221D7635388f4aEb85c71A4701E09640EEe09
```

## NFT

Current NFT contract address:

```
0x6c476fB04Ee975Bf113dc02FDDc1ae11B3de3dbf
```

Explorer link:

```
https://bscscan.com/address/0x6c476fB04Ee975Bf113dc02FDDc1ae11B3de3dbf
```

## External NFT Minting Limitations

Even if an NFT technically exists on-chain, this is not sufficient for participation in the NFTAMA economy.

For a safe to:

- be counted in weight;
- participate in age mechanics;

- participate in minting;
- be visible to the internal market;
- generate AMA2,

it must be created and recorded specifically within the platform logic.

An NFT minted outside the internal project logic is not a full economic safe of NFTAMA.

# 17. Security and Trust

## Existing Protection Measures

The project already implements:

- registration and critical forms protected by captcha;
- email confirmation;
- queues and stepwise background operations instead of immediate chaotic execution;
- balance and fee checks before withdrawals;
- ownership checks before safe operations;
- restrictions on receiving free AMA2 rewards;
- restrictions on the number of mints per safe;
- separation of the user’s internal and external wallets.

## Anti-Bot Mechanics

The project already uses captcha in scenarios where automatic abuse must be limited.

- protection from mass automated registrations;
- protection from abuse in interactive rewards;
- protection from scripted farming of free AMA2 rewards;
- additional limits by time and number of attempts.

## One Player, One Account

Several accounts of one player are forbidden: they would collect free AMA2, demo safes and safe-of-luck attempts at the expense of honest players.

- temporary (disposable) e-mail addresses and blocked domains cannot register, and one IP can register at most 3 accounts a day;
- a new account is checked automatically after registration and during its first 14 days: a shared withdrawal wallet, the same mailbox, series of numbered addresses, a little-known domain with mass registrations, a shared IP;

- a clear duplicate is blocked at once, suspicious groups are reviewed by the administrator;
- a blocked account cannot sign in or act, and its weight takes no part in AMA2 payouts or statistics; safes and balances are not deleted, and a mistaken block is lifted through support.

## Protection Against Exploits

- the project reduces abuse risk through server-side checks of asset ownership, balances, limits, queue statuses, and required fees;
- critical operations are not reduced to a single button and go through validations and handlers;
- part of the logic is moved into queues and background commands, which reduces the number of race-condition scenarios at the interface and operation-processing level.

At the same time:

- as in any Web3 project, zero exploit risk cannot be guaranteed;
- security is a process, not a static state.

## Rug Pull and Trust

For long-term holding of assets in the project, standard trust indicators are important:

- transparent tokenomics;
- transparent contract addresses;
- contract verification and openness of key infrastructure;
- locked liquidity;
- timelock for critical administrative actions;
- multisig mechanism for governance.

## What Multisig Is

`Multisig` is a scheme in which a critical action cannot be performed by one person alone. Several signatures of predefined participants are required.

This is important for the project because it:

- reduces the risk of a single operator’s error;
- reduces the risk of a single key compromise;
- increases trust in contract and treasury management.

## What Timelock Is

`Timelock` is a delay before execution of an important administrative action.

For the project, timelock is useful because it:

- gives the community time to see a change;
- reduces the risk of sudden unwanted changes;
- increases transparency.

Timelock is fixed at the level of the contract, treasury, or governance procedures.

### Liquidity and Locking

In the current model:

- there is already locked liquidity of approximately `1000 USD`;
- the goal by the end of the year is to increase the lock volume to approximately `10000 USD`.

Locked liquidity parameters are published together with on-chain confirmation.

### Audits

- the project relies on transparency of the contract side;
- contract addresses, explorer links, and security materials must be publicly fixed;
- results of completed audits are issued as a separate appendix to the white paper.

## 18. Contract Transparency

Project contract transparency includes:

- AMA2 contract address;
- NFT contract address;
- network explorer links;
- confirmation of token parameters;
- information on liquidity, lock mechanisms, and governance rights.

Such transparency is especially important for a project built around long-term asset holding and reputation.

## 19. Roadmap

### What Has Already Been Implemented

At the current stage, the following have already been implemented:

- public landing page;
- help section;
- registration and authorization;
- email confirmation;
- internal cabinet;
- internal wallets;
- balance top-up and processing;
- safe purchases;
- automatic purchase rules;
- secondary market between users;
- minting of a new safe;
- safe age mechanics;
- weight change history;
- free AMA2 reward mechanics;
- withdrawal of BNB, AMA2, and NFT;
- referral program;
- blog.

## Next 3 Months

Planned functional block: Bank

Essence:

- users will be able to unite into groups or “banks”;
- banks will receive common tasks and common progress;
- bank participants will be able to receive an additional AMA2 bonus;
- group interaction mechanics will appear inside the platform.

Practical effect:

- retention growth;
- team dynamics growth;
- more reasons to hold assets in the project;
- growth of daily activity.

## 6-Month Horizon

Priorities:

- expansion of the user base;
- strengthening of AMA2 liquidity;
- referral and community growth mechanics;
- strengthening of the scarce model through new internal AMA2 spending scenarios.

Suitable strategies:

- campaigns with partners and opinion leaders in NFT/GameFi/BSC niches;
- referral challenges and seasonal activities;
- incentives for liquidity providers;
- special safe releases and game seasons;
- easier entry for new users;
- simplification of the route `buy BNB -> bring it into the platform -> buy a safe`.

### 12-Month Horizon

Promising directions:

- expansion of the banking mechanic;
- growth of locked liquidity;
- public security program;
- governance mechanisms;
- expansion of market integrations;
- additional scenarios for team play, quests, and long-term seasons.

## 20. Team and Governance

### What DAO Is

`DAO` is a decentralized governance model in which part of decisions is made through predefined rules and voting.

The DAO mechanic for NFTAMA belongs to a later stage of development after:

- stabilization of the base economy;
- accumulation of an active community;
- appearance of clear objects for voting.

### What a Governance Mechanism Is

`Governance` is a decision-making system:

- who can propose changes;

- who can vote;
- which decisions require discussion;
- which decisions require timelock or multisig.

For NFTAMA, governance mechanisms apply to:

- epoch parameters;
- banking mechanics;
- community programs;
- incentive models;
- treasury decisions.

### Recommended Governance Model by Stages

### Stage 1. Now

Team governance:

- decisions are made by the core project team;
- transparency and public logic of decisions are important.

### Stage 2. After Community Growth

Advisory governance model:

- discussion of initiatives;
- community polls;
- test voting on non-critical parameters.

### Stage 3. After Product Strengthening

Partial DAO approach:

- voting on part of community and economy parameters;
- treasury and critical functions under multisig and timelock;
- part of strategic decisions remains with the core project team.

## 21. Marketing and Growth

### How Users Learn About the Project

The NFTAMA growth model includes organic distribution:

- if it is beneficial and interesting for a user to remain in the project, that user becomes a source of recommendations;
- for a long-term project, reputation is more important than short-term informational noise;
- community growth is strengthened by engaged current participants interested in the development of the ecosystem.

NFTAMA relies on long-term trust, clear mechanics, and growth through community. Project development is not based on short-term demand formed by inflated expectations. Growth of the user base is supported by the product model and long-term utility of the ecosystem.

## Referral System

The project already has a referral program.

Current referral accrual size:

`9%`

This means that 9% of the invited user's daily weight accrual goes to the referrer, and the invited user receives the remaining 91%.

Model features:

- the referrer's account level bonus (`+0.2` pp per level, up to +3 pp) is paid by the project on top of these 9% — the invited user loses nothing for it;
- the referrer's income grows with the weight of the invited users, so helping newcomers grow pays more than collecting sign-ups.

## How AMA2 Turnover Will Grow

AMA2 turnover will increase through:

- new users;
- new safes;
- upgrades;
- minting;
- internal market;
- referral activity;
- expansion of liquidity;
- banking and team mechanics.

Turnover growth by itself does not guarantee price growth, but it is important as a sign of a living economy: the token is used, transferred, spent, and reintroduced into the cycle.

### How Scarcity Will Increase

If the project develops according to plan, AMA2 scarcity will increase through:

- limited supply;
- use of the token for upgrades;
- use of the token for minting;
- fees and internal costs;
- growth of the user base;
- growth of the number of scenarios where spending AMA2 inside the platform is more beneficial than holding it idle.

## 22. Risks and Responsibility

### 1. Market Risk

Market risk consists in the possibility that AMA2 may fail to reach the target rate.

Such factors include:

- weak market cycle;
- low demand;
- weak community growth;
- insufficient liquidity;
- general market downturn.

### 2. Liquidity Risk

Even with a good product, the token may suffer from:

- insufficient market depth;
- high volatility;
- difficult entry or exit for large volumes.

### 3. Infrastructure Risk

The project uses server-side game logic, databases, queues, and background commands.

This means that risks may include:

- backend logic errors;
- queue delays;
- technical failures;

- incomplete synchronization of off-chain and on-chain parts.

### 4. Smart Contract Risk

As in any Web3 project:

- errors in contracts are possible;
- technical and infrastructural risks are possible in the contract and operational parts of the project;
- risks related to external integrations of the BSC network are possible.

### 5. Regulatory Risk

The legal regime of NFTs, utility tokens, and gaming token economies may change across jurisdictions.

### 6. User Behavioral Risk

The user independently decides:

- when to enter;
- which safes to buy;
- whether to spend AMA2 on upgrades;
- whether to withdraw tokens;
- whether to use minting.

An incorrect strategy may worsen the result even if the project mechanics work as intended.

NFTAMA is a project with a long-term gaming and economic model, but it provides no guarantees of profitability. The price of AMA2, user activity, market depth, development stability, and practical user utility depend on ecosystem development and the quality of its implementation.

## 23. Conclusion

NFTAMA is built around the following basic principles:

- an NFT safe is not an image, but an asset with parameters and function;
- AM weight is the core of participation in the system;
- AMA2 is the main token for settlement and ecosystem development;
- the user path is built around the cycle `weight -> AMA2 -> strengthening -> new weight`.

Strengths of the model:

- clear portfolio growth logic;
- practical value of NFTs;
- economic role of age, level, and rarity;
- reinvestment cycle through AMA2;
- combination of gaming, NFT, and token economy.

Key project task:

- development of a sustainable ecosystem with user growth, liquidity, and trust infrastructure.

Further development of NFTAMA is determined by model transparency, AMA2 liquidity, infrastructure security, product development pace, and community resilience. These factors shape the project’s long-term position in the NFT, GameFi, and utility-driven token economy segment.

# 24. Variable Definitions

Below are the designations used in the formulas of the document.

- **A** — safe weight or a weight quantity in the system.
- `A0` — initial weight of the safe at the moment of opening.
- `A1` — weight of the safe by the end of the primary opening cycle.
- $\mathbf{A_i}$ — total weight of user **i**.
- $\mathbf{A_j}$ — weight of one of the safes participating in minting.
- $\mathbf{A_{in}}$ — total weight of safes participating in minting.
- $\mathbf{A_{new}}$ — resulting weight of the new safe after minting.
- $\mathbf{p_7}$ — the player's showcase and forged safes of the last 7 days (safe of luck attempts).
- $\mathbf{A_{sum}}$ — total weight of all system users.
- $\mathbf{A_{base}}$ — base weight of the safe: capacity and daily growth are counted from it.
- $\mathbf{A_{start}}$ — start weight of the safe (at purchase or mint); the +xx% growth is measured from it.
- $\mathbf{A_{entry}}$ — weight with which the user entered the system.
- $\mathbf{B_n}$ — return percentage value of the account at level **n**.
- **C(t)** — cost of one unit of weight at time **t**.
- $\mathbf{D_i}$ — share of user **i** in the total system weight.
- $\mathbf{E[A_{new}]}$ — expected weight of a new safe.
- $\mathbf{E_j}$ — epoch number **j**.
- $\mathbf{F_n}$ — base free AMA2 reward of the account at level **n**.
- $\mathbf{F_{day}}$ — expected amount of free AMA2 reward per day.

- $\mathbf{F_{month}}$ — expected amount of free AMA2 reward per month.
- **G** — gas or network operation cost in BNB.
- **H** — investment horizon.
- **C** — capacity multiplier of a safe; Cap — capacity (the maximum weight) of a safe.
- **g** — daily growth of a safe, % of its base weight.
- γ — rarity gene of a safe; Γ — what the genes add to the top class chance in a mint, p.p.
- **F** — mint resource of a safe.
- **u** — birth roll of a safe: 0 is the lightest, 1 the heaviest.
- **L** — upper price boundary in the auto-buy strategy.
- $\mathbf{L_m}$ — limit of participation of one safe in minting (mint resource plus level slots).
- **M(k)** — mint cost in AMA2 when **k** safes participate.
- $\mathbf{M_0}$ — base mint cost from two safes.
- $\mathbf{M_{total}}$ — full mint cost including gas.
- $\mathbf{N_0}$ — base number of attempts to receive free AMA2 reward per day.
- $\mathbf{N_{day}}$ — total number of attempts to receive free AMA2 reward per day.
- $\mathbf{N_{max}}$ — maximum number of attempts to receive free AMA2 reward per day.
- **P(t)** — safe price at time **t**.
- `P0` — starting safe price at opening.
- `P1` — safe price by the end of the primary opening cycle.
- `P*` — expected entry price for the selected strategy.
- $\mathbf{P_{ama}}$ — price of one AMA2 at the calculation point.
- $\mathbf{P_{bid}}$ — final bid in the price-increase purchase strategy.
- $\mathbf{P_{max}}$ — maximum price of the auto-buy strategy.
- $\mathbf{P_{target}}$ — target entry price in the wait-for-price-drop strategy (`P0 * q`).
- **Q** — amount of AMA2 directed to distribution for a cycle.
- $\mathbf{Q_{net}}$ — amount of AMA2 after fee deduction.
- **R** — ratio of expected weight to expected entry price.
- $\mathbf{R_i}$ — reward of user **i** for the distribution cycle.
- $\mathbf{R_{ref}}$ — referrer reward.
- $\mathbf{R_{child}}$ — base reward of the invited user.
- **S** — amount of AMA2 available for distribution in the cycle.
- $\mathbf{S_{total}}$ — total AMA2 supply.
- **T** — lifetime of an opened safe on the primary market.
- $\mathbf{R_{tot}}$ — total user reward for the calculation period.
- $\mathbf{U_1}$ — price of the first safe upgrade.
- $\mathbf{U_n}$ — price of the safe upgrade at level **n**.
- **V** — value of accumulated AMA2 in monetary terms.

- $\mathbf{V_{safe}}$ — market value of a safe.
- **W** — total weight of the entire system.
- $\mathbf{W_j^-}$ — lower weight boundary for epoch **j**.
- $\mathbf{W_j^+}$ — upper weight boundary for epoch **j**.
- **d** — either the number of days or a specified percentage of price decrease, depending on context.
- **f** — AMA2 distribution fee rate.
- $\mathbf{g_a}$ — base growth rate of safe weight when upgrading.
- $\mathbf{E_L}$ — experience needed for account level **L** ($30 \cdot L^2$).
- $\mathbf{g_u}$ — growth rate of safe upgrade price.
- $\mathbf{h_n}$ — growth rate of the base free AMA2 reward at account level **n**.
- **j** — epoch number.
- **k** — number of safes participating in minting.
- `lvl` — safe level.
- $\mathbf{F_i}$ — the fresh weight of player **i** (added in 30 days); $\mathbf{P_{fresh}}$ — the part of a safe price going to the fresh pool.
- **n** — level number.
- **q** — safe quality coefficient.
- **r** — share of AMA2 directed to distribution per cycle.
- **s** — price increase step in the auto-buy strategy.
- **t** — time since the safe was opened.
- $\mathbf{t_{target}}$ — time at which the price reaches the specified entry level.
- **y** — expected future yield of the safe.
- **α** — referral accrual rate.
- $\mathbf{W_j^+}$ — the upper total weight boundary of epoch **j**; each next one is twice as large.
- $\mathbf{\eta_m}$ — minting efficiency as the ratio of expected weight to cost.
- **ξ** — random component affecting the resulting weight of a new safe during minting.
- **Φ** — function determining the resulting weight of a new safe during minting.