Auto Mining Token (AMT)

Whitepaper

Abstract: Various techniques exist to tokenize Bitcoin (BTC) mining. Our proposal is to create a token that offers a unique and simple mechanism for users, allowing them to earn mining returns simply by holding the tokens in their wallet and freely transacting them. Additionally, we aim to provide a suite of tools necessary to improve transparency and overall operations for the token holders.

Introduction:

AutoMiningToken is an ecosystem that, through its token, AMT, allows investment in Bitcoin mining projects. Users holding AMT will earn returns based on the profitability generated by the project, relative to their share of AMT in circulation. For example, if a wallet holds 50% of the circulating tokens, it will receive 50% of the respective payments.

The project divides payments in two ways:

Direct Profit:

Direct profit is the portion of BTC generated on which AMT holders can directly earn by keeping the token in their wallet.

Growing Liquidity Reserve (GLR):

The GLR is the portion of BTC generated that is placed in a smart contract, which allows AMT holders to exchange their tokens for BTC within that contract. The exchange is done proportionally to the amount of AMT exchanged against the total AMT in circulation. The AMT exchanged in this vault are burned, improving the overall profitability of the ecosystem. The GLR functions as a guaranteed minimum exchange rate against BTC for AMT holders (beyond market price) that grows over time. The project guarantees a base amount of BTC (30 BTC) and will make deposits over time and daily, based on the project's evolution.

The distribution ratio between both profits is variable but aims to stabilize over time as the token develops in the market.

Profit distribution occurs daily, reading the various mining pools where the equipment is connected and deducting production costs.

Transparency and Hashing Power:

There will be a maximum of 100,000,000 AMT, issued over time as hashing power is acquired. The project began with 70,000 TH/s and a token issuance establishing a ratio of 1 TH/s = 625 AMT. Currently, the project has issued 55,035,719 tokens with a TH/s ratio of 1 TH/s = 246.3 AMT. This is due to the policy of purchasing equipment based on token sales and the increase in mining power (currently reaching 223,446 TH/s).

All our equipment will connect to mining pools with open observers available 24 hours for consultation. Thus, users can verify daily the benefit they are entitled to. Additionally, the various equipment connections and possible token issuances will be announced through the project's communication channels.

The entire project scheme contemplates the costs of administration, energy, and amortization of the equipment with the tokens that the project itself will retain, following this scheme:

Initial AMT Issuance:

75% Mining contributor of hashing power
20% For public sale
5% Administration and marketing

External audits will also be conducted to verify the correct functioning and security of the smart contract ecosystem.

Guaranteed Exchange Liquidity:

A liquidity pool was created on PancakeSwap, locking liquidity tokens for two years to allow free user exchange.

The initial liquidity provision amounts were:

49.5 BTCB (Equivalent to approximately $1,500,000 USD on that date)
1,000,000 AMT

The locking transaction and amounts can be confirmed at the following link:

https://bscscan.com/tx/0x82a33707058607a4da956951a25561e1281e5fc7836a2f2ef69bc9100ef2eeba

"Great Reset":
During the course of the project, a general assessment was conducted, and it was decided to increase the profitability of the token to make it more attractive to new investors. This was achieved by reducing the price through a sale by the project administrators in the liquidity pool, while guaranteeing all holders at the time an airdrop of the corresponding amount of AMT to prevent loss of value. The tokens were distributed from the project's reserves without requiring any new issuance. The result was a more accessible token and an increase in profitability for both new participants, by accessing a more economical token, and for existing holders, who ended up with a greater number of tokens, thereby increasing their Bitcoin returns.

Bitcoin Mining Investment Opportunity

The project develops its mining operations in Latin American territory, always seeking the lowest production costs (mainly competitive electricity prices) and represents the union of various mining groups with previous experience of working together.

Our aim is to democratize mining and allow users worldwide to invest and access its benefits, regardless of their geographical location.

Through our Growing Liquidity Reserve system, we guarantee users always have a potential exit for exchange against Bitcoin, aiming to match the market price as the project develops, grows, and expands.

We work daily to improve the overall profitability of the project by incorporating new equipment, enhancing our facilities, seeking more competitive prices, and optimizing our operations in general. Our goal is to systematically improve the TH/s to AMT ratio to enhance the profitability for token holders.

Blockchain - BNB Smart Chain

The BNB Smart Chain (formerly Binance Smart Chain) has proven its capacity, security, and public trust to host large-scale projects. In this context, we commit to using it to provide the best cost-benefit in terms of user transaction costs, wallet compatibility, integration with other dAPPs, and the ability to use BBTC as a representation of Bitcoin within this network.

Wrapped Bitcoin - BTCB

BTCB is a BEP2/BEP20 wrapped asset on Binance Smart Chain with a 1:1 peg to BTC locked on the Bitcoin blockchain. BTCB operates with a centralized and trust-based model. The 1:1 ratio means that the amount of wrapped BTCB equals the amount of BTC locked at a public address. The centralized and trust-based model implies that Binance is the issuer of the wrapped BTCB tokens.

Transparency is ensured through the Proof of Assets webpage (https://www.binance.org/en/assets-proof), where you can verify the current supply of all locked and issued assets on the public blockchain. The amount of locked BTC may not be exactly the same as the wrapped BTC because these audit data are not updated in real-time but are processed weekly.

Once BTC is mined, we will use a Binance bridge to generate the corresponding BBTC and use it to pay the yields. In addition, we guarantee the availability of wallets containing backup BBTC so that, in the event of any bridge failure, the corresponding payment can be made within the determined timeframe.

Project Changes and Expansion Possibilities

Beyond the initial launch decisions, our vision includes being prepared for the constant evolution of the technology on which the project is developed. To the best of our ability, we will work to increase the project's decentralization and aim to be ever closer to Bitcoin's purest forms.

In this regard, we are constantly researching the various alternatives and technological possibilities that are being created and tested to provide the safest and most reliable environment for users.

Technical Aspects of the Token

AMT will be developed in accordance with the ERC-20 standard (BEP-20 within the BSC). This ensures its compatibility with any wallet and external applications that recognize it.

Snapshot functionality will be added, utilizing the standard's derivative developed by Open Zeppelin (https://github.com/OpenZeppelin/openzeppelin-contracts/blob/master/contracts/token/ERC20/extensions/ERC20Snapshot.sol).

This derivative allows for the definition of specific moments (snapshots) at which a wallet's balance can be retroactively consulted. These moments will be determined by privileged operations carried out by the administrator, in the

same transaction in which the return payment is made. Thus, it is ensured that collection can be exclusively and only once made by the holders of AMT at the moment the payment was executed. In addition, the total circulation of AMT will be recorded in the snapshot, allowing holders to collect based on the specific proportion of their AMT to the total available at the time of payment, ensuring that their tokens do not lose value due to future issuances.

Staking Vault and Buyback

The AMT project implements two additional smart contracts to facilitate operations for users, especially those with a smaller number of tokens.

The AMT token implementation allows holders to collect their profitability each time a payment is made by executing a transaction on the contract that distributes the returns. This involves paying the gas fees required by the network, which could eventually make the operation unprofitable for those with small amounts of AMT. Therefore, the AMT project develops a smart contract that allows users to deposit their tokens, and the project assumes the gas fees for collecting from that contract. Once the user decides to withdraw their tokens, they will receive the initially deposited tokens plus the BTCB corresponding to all the collections that occurred during the deposit period for the AMT they deposited. While this vault is geared towards users with a small amount of AMT, it is available to all who wish to take advantage of this system.

Inspired by the same concept, we developed another smart contract called the buyback vault, which emulates the operation of the previous vault but uses all the collected BTCB to purchase AMT at market price. This allows users to automatically accumulate AMT and increase their profitability with each collection. In turn, it contributes to the overall price growth of AMT by executing purchase operations periodically.

Profitability Generation for Liquidity Providers

Considering that the ability to provide liquidity in decentralized markets should be a public right for all holders, and given that the liquidity pool contract retains the tokens in this case, we have implemented a system whereby even those providing liquidity on Pancake Swap can collect returns generated by AMT.

A contract will be implemented that masks the liquidity provisioning process, allowing those who provide through our interface to receive, instead of the original LP liquidity token, an auxiliary token called liqAMT, paired at a 1:1 ratio with the respective LP token. The contract will safeguard the LP token and utilize it whenever the user decides to withdraw liquidity, burning the issued liqAMT and returning the pool tokens represented by the LP.

Here is the translated section focusing on profitability generation for liquidity providers, market dynamics, and the Growing Liquidity Reserve:

Market and Growing Liquidity Reserve

Fixed-price token sales will be conducted through a Smart Contract created specifically for this purpose, which we will call the MarketPlace (MP). In this regard, the sales process will be 100% transparent as long as the administrator has tokens to be sold. The Contracts will have a limited amount of AMT to be sold at a dollar-denominated price in line with the market. Initially, it will be slightly above the price of the Liquidity Pool (PancakeSwap) to prevent instant arbitrage and to take advantage of a potential rise in BTC which would cause AMT to increase and surpass the value of our MarketPlace. At that moment, the user will likely opt, if they wish to acquire AMT, to buy them in our MP instead of PancakeSwap, potentially even selling these discounted tokens on Pancake at a higher price to rebuy on our MP, in a process known as "arbitrage."

The Growing Liquidity Reserve (GLR) started operations with 30 bitcoins contributed by the miners and administrators of the project. Since its creation, it has been recharged daily with a percentage of the production. It is designed to deliver the BTCB it holds exclusively to AMT holders in case they want to liquidate their position.
It will have a rate established by: BTCB in the vault / AMT in circulation
It will deliver the BTCB and proceed to burn the AMT in the same transaction. Thus, the established proportion can only increase, always guaranteeing a minimum liquidation price for users, which grows over time.
For example:
Let's assume we have 30 BTC deposited in the GLR and the circulating AMT at this moment are 30,000,000, so the exchange rate is:

30/30,000,000 = 0.000001 BTC

Let's say user Bob owns 1,000,000 AMT and wishes to liquidate them in the GLR. He deposits his AMT and immediately receives 1 BTC. In the same transaction, 1,000,000 AMT are burned, so the final exchange rate is:

29/29,000,000 = 0.000001 BTC

The relationship remains the same; subsequent users can access the same amount of BTC for their AMT. As mentioned, the GLR is recharged daily with a production percentage, so let's assume that in a month, the amount of BTC in the GLR is again 30 BTC, then at that moment we will have:

30/29,000,000 = 0.00000103 BTC

This clearly illustrates why we say the GLR can only increase, because:

1. AMT is finite, there is an insurmountable maximum of 100,000,000
2. AMT can be burned and removed from circulation.
3. The BTC in the GLR increases daily.

We have implemented this system because we strongly believe in BTC, we promote its development, and understand that it is essential for investors to have a transparent, constantly growing, and fully guaranteed liquidity backup mechanism, as there will be no other alternative to withdraw BTCB from the vault than the one established through the liquidation of AMT. This smart contract will not allow changes in its structure, not even by the administrative group. Any AMT holder has the right from the first moment to exchange them for BTCB in this Vault, and it is only a matter of time before the value is equal to or higher than the open market quotation. It is also worth noting that the open market value cannot be lower than the value of the AMT quoted in the GLR, as otherwise, any user could liquidate their tokens in the GLR and repurchase them on Pancake, causing their value to rise there until it is no longer viable to perform this process, which as mentioned before is known as arbitrage. This phenomenon leads to the quoted price in the GLR acting as a

minimum price for the AMT, and there will always be liquidity to back the AMT beyond the circumstances of the open market.

Technical Annex
Technical Definitions of the Main Contracts Forming Part of the AMT Ecosystem

Code Repository: https://github.com/AutoMiningToken

AMT

Address: 0x6Ae0A238a6f51Df8eEe084B1756A54dD8a8E85d3

Deploy transaction hash:
0xe8a9859d21c71617831df8c5b40f3845cc549bec5ae0548b871b1de604ee88a1

This contract implements the ERC20 token named "AutoMiningToken" (AMT) using the OpenZeppelin library. The token is owner-controllable and enables the taking of snapshots of its state at any given moment. The maximum amount of tokens that can be created is 100,000,000 AMT.

<u>Functions</u>

Constructor: This function is called once upon deploying the contract. It initializes the token's name and symbol.
mint: This function is to mint or create new AMT tokens. Only the contract owner can call this function. It ensures that the total amount of AMT tokens does not exceed the set maximum, which is 100,000,000 AMT.
snapshot: This function creates a snapshot of the current state of the token. A snapshot is a capture of the token distribution at a specific moment. Only the contract owner can call this function.
getCurrentSnapshotId: This function returns the identifier of the most recent snapshot.
burn: This function is to burn or destroy a certain number of tokens from the caller.

burnFrom: This function is to burn a certain number of tokens from a specific account. The caller must have approval to burn the tokens from the specified account.

The functions mint, snapshot, burn, and burnFrom are functions that the contract owner can call, while getCurrentSnapshotId can be called by any user who wants to know the identifier of the most recent snapshot.

Liquidity AMT

Address: 0x679Bd76cA0B3f037131AF9170d0462c9FfC9Bc27

Deploy transaction hash:
0xf24e919f13aa2acc4d044ab597c2056f3f98558ada11c4c3901b0a22ff2d1649

This contract implements the ERC20 token named "liqAutoMiningToken" (liqAMT) using the OpenZeppelin library. This token is designed for liquidity purposes. The token is owner-controllable and enables the taking of snapshots of its state at any given moment.

In the context of the system, it is used as an auxiliary token to allow liquidity providers to collect the returns due for the AMT they placed in liquidity.

Functions

Constructor: This function is called once upon deploying the contract. It initializes the token's name and symbol.
mint: This function is to mint or create new liqAMT tokens. Only the contract owner can call this function.
snapshot: This function creates a snapshot of the current state of the token. A snapshot is a capture of the token distribution at a specific moment. Only the contract owner can call this function.
getCurrentSnapshotId: This function returns the identifier of the most recent snapshot.
burnFrom: This function is to burn a certain number of tokens from a specific account. The caller must have approval to burn the tokens from the specified account.

The functions mint, snapshot, and burnFrom are functions that the contract owner can call, while getCurrentSnapshotId can be called by any user who wants to know the identifier of the most recent snapshot.

Master

Address: 0x13e98112e1c67DbE684adf3Aeb1C871F1fe6D1Ac

Deploy transaction hash:
0x555f41a7be82aa06cc4cd3cd90d53b926b5628ef3c4bf7968ad172c9861a2251

This smart contract is a multifunctional contract that manages various aspects of the AMT system. The main functions and their functionalities are as follows:

Charge Functions

charge and chargeFromTo: these functions allow an AMT holder to claim the charges from a given payment or a range of payments.
liqCharge and liqChargeFromTo: perform the same operation as the above but for liquidity providers. Liquidity providers can be rewarded for their contribution to the liquidity fund, and these functions allow these providers to claim their rewards.

Addition and Withdrawal of Liquidity

addLiquidityLocking: this function allows the contract owner to add liquidity to the contract and lock it for two years. This function was used to inject initial liquidity into the contract and ensure a certain level of stability in the liquidity pool.
addLiquidity: any user can call this function to add liquidity to the contract. Users can provide AMT and BTCB in exchange for liquidity tokens.
removeLiquidity: users can use this function to withdraw their liquidity from the fund. Liquidity tokens are burned, and users receive AMT and BTCB in return.
Token Issuance

mintMaster: this function allows the contract owner to issue new AMT tokens.

This contract acts as an interface between users and the liquidity pool on Pancake Swap. It manages the receipt of BTCB generated by the system and allows users to withdraw their corresponding charges. Additionally, it serves as a bridge for users who provide liquidity, allowing them to access profitability charges.

This contract is a fundamental piece for the AMT system, allowing users to interact with the system's liquidity, receive rewards, and manage their participation in the system's liquidity in a decentralized and secure way.

BurnVault (Growing Liquidity Reserve)

Address: 0xC9c60fd0a239fCA106989Da17AE90595bf5bA114

Deploy transaction hash:
0xa41fc9cc77eb5f6a3f3a5812e6efcca471e8d22e6b6db9778d7a0d0f75daa114

This contract is used to burn AMT (AutoMiningToken) tokens and withdraw BTCb tokens (representation of Bitcoin on the BNB Smart Chain blockchain) that are backed.

BTCb tokens are used as backing for AMT tokens.

Functions

Constructor: This function is called once when deploying the contract. It initializes the AMT token that will be used in the contract.
backingWithdraw: This function burns a certain amount of AMT tokens and withdraws the proportional amount of backed BTCb tokens. The BTCb tokens are sent to the account that burned the AMT tokens. The amount of BTCb tokens to be withdrawn is calculated as the proportion of the amount of AMT tokens burned relative to the total supply of AMT tokens, multiplied by the amount of BTCb tokens the contract holds. If the BTCb transfer is successful, a 'burnMade' event is emitted that records the amount of AMT tokens burned and the amount of BTCb tokens withdrawn.
In addition to the constructor and backingWithdraw function, the contract has some constants and variables defined:

addrBtcb: This is the address of the BTCb token on the Ethereum blockchain.
btcb: This is an ERC20 token interface for interacting with BTCb tokens.
amt: This is an immutable instance of the AMT token class. This allows for interaction with the AMT token contract, specifically to burn AMT tokens.