# Auto Mining Token

# AMT

# Whitepaper

July 16, 2021

**Abstract:** There are several techniques based on which efforts were made in order to tokenize BitCoin (BTC) mining. Our goal is to create a unique and simple mechanism for the user allowing them to keep their tokens at all times and make transactions with them whenever they want. Combining the snapshot-token technique with the speed, security, and low transaction cost of the BSC, we will be able to create a scalable, transparent and profitable ecosystem for investors.

## Introduction

Auto Mining Token is an ecosystem that allows, through its token - AMT, the investment in BTC mining projects. Users who own AMT will be able to profit from the returns generated by the project based on the amount of AMT they own on all tokens circulating. In other words: If a wallet holds, for example, 50% of all the circulating tokens, it will collect 50% of the return paid to the holders.

(BTC Production – operational costs) → 50% Direct Profit

→ 50% Warranty Vault

In addition, half of the profits received daily by the token holder (after costs have been deducted) will be placed in a smart contract that will function as a vault. The vault will always allow AMT holders to liquidate their tokens at a rate determined by the BTC stored in the vault according to the total of AMT. In this way, a progressive backup will be generated on the AMT, which can only be extracted as per the liquidation of the token. All tokens liquidated in the vault will be duly burned and taken out of circulation for good.

There may be a maximum of **100 million AMT** which will be issued over time based on the acquisition of greater mining power. The project will start operating backed by approximately 70,000 TH/s with an AMT issuance calculated in a way that the profitability is beneficial to the investor, and in order for the miner to tolerate the impact of the decrease in their hash power (for the benefit of the token investor) until new equipment is acquired in such a way as to increase the profitability of the initial

investors and of the miner who contributes to the initial hash power, sharing the new hash power introduced into the ecosystem.

**Transparency and hash power**

All of our equipment will be connected to an open mining pool with public observers that will be made available on the web. In this way, it will be possible to verify daily what is the profit that the entire project is generating. Likewise, there will be notifications whenever a new equipment is connected, as well as prior notice whenever new tokens are issued.

The whole project framework provides for the costs of administration, energy and amortization with the tokens that the project itself will keep. These expenses will be discussed openly, and AMT holders will be allowed to take part in the decisions through voting.

| | | | |
|---|---|---|---|
| Initial Issue | AMT | → 75% | Contributing miner - hash power |
| | | → 20% | Market place (1 AMT= 1USD) |
| | | → 5% | Administration and Marketing |

In addition, external and independent auditors will be incorporated in order to ensure the smooth running of the Smart Contracts and their security.

The exposure to BITCOIN mining, liquidity and trading opportunity

This project will add to the overall ecosystem of mining exposure a brand-new mechanism of collection and it will also increase overall liquidity. Unlike other projects, it will not be necessary to deposit tokens in a Smart Contract or lock them for a certain period to earn the referred profits. In addition, a liquidity pool will be created on Pancake Swap and the liquidity tokens will be blocked for a period of 2 years. With a volume of 1,000,000 AMT and the equivalent in BTCB of 1,000,000 usdt at the time of launch, in order to ensure an initial value in parity with the US dollar. The referred blocking will be achieved through a Smart Contract derived from the traditional TimeLock, adding the functionality of being able to earn profits over the blocked tokens. This will ensure the continuous development of the project and the commitment to it. In addition to the overall exposure to mining, exposure to the development of the project itself will occur based on 2 aspects:

1. The natural backup of the AMT against the BTC will be continually increased through the vault that receives part of the returns. That will always give an option to all holders which could be either, depending on the moment and the price at which the token was bought, at a loss or a profit. But they will always have the possibility of withdrawing part of the profits generated throughout the project's history.

2. By improving the AMT/HashPower ratio with the new infrastructure investments. The aim will always be to ensure that AMT holders have a better hash power ratio compared to new holders who acquire recently issued ones. This system will continue until it reaches the point of stability of 100 million circulating tokens, in which the definitions regarding the new investments to be made will begin to have much more weight.

In every sense, due to the operation we develop in Latin America, we managed to achieve extremely competitive costs for energy, labor and equipment acquisition. Thanks to the union of several mining ventures in the region, our token will be able to provide better results than those currently found on the market.

**Blockchain - BNB Smart Chain**

The BNB Smart Chain (former Binance Smart Chain) has proven to have the capacity, security and trust of the general public to host high-scale projects. In this sense, we are committed to using it in order to provide the best cost-benefit when it comes to user transaction costs, compatibility with wallets, integration with other dAPPs and the possibility of using the BBTC as a representation of Bitcoin within this network.

Wrapped BitCoin - BTCB

BTCB is a BEP2/BEP20 asset wrapped (pegged) on Binance Chain/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 peg means that the amount of BTCB wrapped is equal to the amount of BTC locked in a public address. The centralized and trust-based model implies that the issuer of the wrapped BTCB tokens is Binance.

The transparency is ensured through the Proof of Assets webpage (https://www.binance.org/en/assets-proof), where you can verify the current supply for 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 this audit data is not updated in real-time, but instead, it's processed weekly.

Once the BTCs are mined, we will use a Binance bridge to generate the respective BBTC and will use them to pay the returns. In addition, we ensure the availability of wallets that will hold backup BBTC so that, in the event of any bridge failure, the corresponding payment can be made within the determined time.

Technical aspects of the AMT token

AMT will be developed in compliance with the ERC-20 standard (BEP-20 within the BSC). In this way, we can guarantee their portability with any wallet and external applications that can recognize them.

Snapshot functionality will be added to it, by using the deviation of the standard developed by Open Zeppelin (https://github.com/OpenZeppelin/openzeppelin-contracts/blob/master/contracts/token/ERC20/extensions/ERC20Snapshot.sol)

This variation allows the definition of specific moments (snapshots) in which the balance of a wallet can be consulted retroactively. These moments will be determined by privileged operations made by the administrator, in the same operation on which the return payment is made. In this way, it is guaranteed that the collection can be made exclusively and only once by the AMT holders at the time the payment was made. Also, the total amount of AMT in circulation will be recorded in the snapshot, so that the holders will be able to collect them based on the specific ratio of their AMT and the total available at the time of payment, hence their tokens will not lose value upon future issues.

**Profitability generation for liquidity providers**

Considering that the possibility of providing liquidity in decentralized markets will be a public right for all holders and since the liquidity pool contract holds the tokens in this case, we have implemented a system by which even those who are providing liquidity on Pancake Swap will be able to collect the returns generated by the AMT.

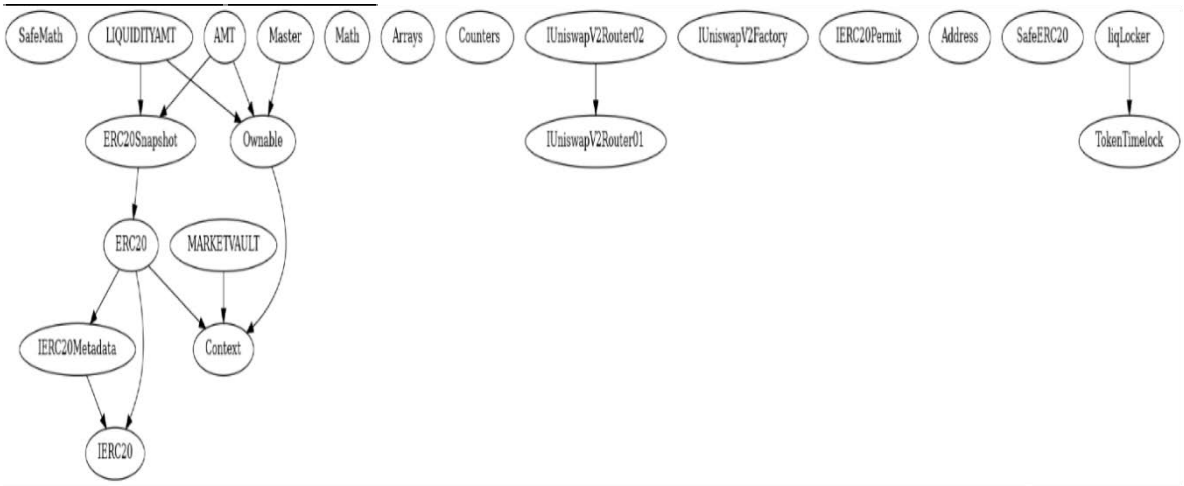
A contract that masks the process of liquidity provision will be implemented allowing those who provide through our interface to obtain instead of the original LP liquidity token an auxiliary token called liqAMT paired in a 1:1 ratio with the respective LP token. The contract will safeguard the LP token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

Market and vault

The initial sales of the token will be carried out by a Smart Contract created specifically for this purpose, which will also act as a backup vault for AMT, earning its respective percentage for each return payment made. In this sense, the sales process will be 100% transparent.

In addition, the vault will systematically receive BBTC with each payment, delivering them exclusively to AMT holders in case they want to liquidate their position in a ratio established by the *balance of BBTC in the vault : liquidated AMT*, burning the AMTs in the process. In this way, the established ratio can only increase, always guaranteeing a minimum price of liquidation for users, which increases over time. We have implemented this system because we firmly believe in BTC, and we promote its development and understand that it is essential for investors to have a transparent backup mechanism. There will be no other alternative to withdraw the BBTC from the vault other than the one established through AMT liquidation.

Structure




## Integration of external miners

We will create a system in which miners who are not in our facilities can take part in the project. By virtue of the hash power provided, an increase in the hash power of 5% will be guaranteed, due to the capitalization obtained by the sale of the tokens issued with the guarantee of this external contribution, we can use it to:

1. Amortize the hash power provided so that, in the event of a possible disconnection, the generated AMTs continue to be backed up.

2. Acquire new equipment that may improve the overall ratio between AMT:HashPower.

Thus, as long as we have enough energy capacity, we will be able to transfer the BTC mining investment from less profitable areas to more profitable ones. This will allow the external miner to have a smoother and more profitable transition than they would have if they were mining in a less profitable location.

## Final comments and goals

Although it is currently impossible to generate a 100% decentralized Bitcoin mining tokenization system, our ecosystem undertakes to achieve the maximum transparency allowed by the current state-of-the-art blockchain technology.

We are faced with an integration of various methodologies applied by other projects separately in order to build a thriving ecosystem in a competitive environment, also affected by market volatility.

Ultimately, our goal is to systematically build user confidence in the AMT in order to achieve, not only its valuation, but also a sustained growth of mining in profitable territories over time, thus providing, with the investment obtained through tokenization, a commercial scale that can firmly compete against any other market supply.

Attachment

*1- Smart Contract Structure*

- Smart Contracts
  - ○ Ownable (They will be duly verified on bscscan at the time of deploy)

    ■ AMT.sol
    ■ LIQUIDITYAMT.sol ■ liqLocker.sol
    ■ MARKETVAULT.sol ■ Master.sol

  - ○ Imported (open zeppelin github) ■ ERC20

    ■ ERC20-snapshot ■ TokenTimelock ■ Ownable

  - ○ Imported (Uniswap liquidity conections) ■ IUniswapV2Router02

    *2- Function overview*

```
($) = payable function
 # = non-constant function
 Int = Internal
```

---

```
Ext = External
Pub = Public
+ [Lib] SafeMath
   - [Int] tryAdd
   - [Int] trySub
   - [Int] tryMul
   - [Int] tryDiv
   - [Int] tryMod
   - [Int] add
   - [Int] sub
   - [Int] mul
   - [Int] div
   - [Int] mod
   - [Int] sub
   - [Int] div
   - [Int] mod
+ [Int] IERC20
   - [Ext] totalSupply
   - [Ext] balanceOf
   - [Ext] transfer #
```

```
   - [Ext] allowance
   - [Ext] approve #
   - [Ext] transferFrom #
+ [Int] IERC20Metadata (IERC20)
   - [Ext] name
   - [Ext] symbol
   - [Ext] decimals
+  Context
   - [Int]  msgSender
   - [Int] _msgData
+  ERC20 (Context, IERC20, IERC20Metadata)
   - [Pub] #
   - [Pub] name
   - [Pub] symbol
   - [Pub] decimals
   - [Pub] totalSupply
   - [Pub] balanceOf
   - [Pub] transfer #
   - [Pub] allowance
   - [Pub] approve #
   - [Pub] transferFrom #
   - [Pub] increaseAllowance #
   - [Pub] decreaseAllowance #
   - [Int] _transfer #
   - [Int] _mint #
   - [Int] _burn #
   - [Int]  approve #
   - [Int] _spendAllowance #
   - [Int] _beforeTokenTransfer #
   - [Int] _afterTokenTransfer #
+ [Lib] Math
   - [Int] max
   - [Int] min
   - [Int] average
   - [Int] ceilDiv
   - [Int] mulDiv
   - [Int] mulDiv
   - [Int] sqrt

- [Int] sqrt

+ [Lib] Arrays
   - [Int] findUpperBound
+ [Lib] Counters
   - [Int] current
   - [Int] increment #
   - [Int] decrement #
   - [Int] reset #
+  ERC20Snapshot (ERC20)
   - [Int]  snapshot #
   - [Int] _getCurrentSnapshotId
   - [Pub] balanceOfAt
   - [Pub] totalSupplyAt
   - [Int] _beforeTokenTransfer #
   - [Prv]  valueAt
   - [Prv] _updateAccountSnapshot #
```

```
   - [Prv] _updateTotalSupplySnapshot #
   - [Prv] _updateSnapshot #
   - [Prv] _lastSnapshotId
+  Ownable (Context)
   - [Pub] #
   - [Pub] owner
   - [Int] _checkOwner
   - [Pub] renounceOwnership #
      - modifiers: onlyOwner
   - [Pub] transferOwnership #
      - modifiers: onlyOwner
   - [Int] _transferOwnership #
+  AMT (ERC20Snapshot, Ownable)
   - [Pub] #
      - modifiers: ERC20
   - [Pub] mint #
      - modifiers: onlyOwner
   - [Pub] snapshot #
      - modifiers: onlyOwner
   - [Pub] getCurrentSnapshotId
   - [Pub] burn #
   - [Pub] burnFrom #
+ [Int] IUniswapV2Router01
   - [Ext] factory
   - [Ext] WETH
   - [Ext] addLiquidity #
   - [Ext] addLiquidityETH ($)
   - [Ext] removeLiquidity #
   - [Ext] removeLiquidityETH #
   - [Ext] removeLiquidityWithPermit #
   - [Ext] removeLiquidityETHWithPermit #
   - [Ext] swapExactTokensForTokens #
   - [Ext] swapTokensForExactTokens #
   - [Ext] swapExactETHForTokens ($)
   - [Ext] swapTokensForExactETH #
   - [Ext] swapExactTokensForETH #
   - [Ext] swapETHForExactTokens ($)
   - [Ext] quote
   - [Ext] getAmountOut
   - [Ext] getAmountIn
   - [Ext] getAmountsOut
   - [Ext] getAmountsIn
+ [Int] IUniswapV2Router02 (IUniswapV2Router01)
   - [Ext] removeLiquidityETHSupportingFeeOnTransferTokens #
   - [Ext] removeLiquidityETHWithPermitSupportingFeeOnTransferTokens #
   - [Ext] swapExactTokensForTokensSupportingFeeOnTransferTokens #
   - [Ext] swapExactETHForTokensSupportingFeeOnTransferTokens ($)
   - [Ext] swapExactTokensForETHSupportingFeeOnTransferTokens #
+ [Int] IUniswapV2Factory
   - [Ext] feeTo
   - [Ext] feeToSetter
   - [Ext] getPair
   - [Ext] allPairs
   - [Ext] allPairsLength
   - [Ext] createPair #
   - [Ext] setFeeTo #
   - [Ext] setFeeToSetter #
```

```
+  LIQUIDITYAMT (ERC20Snapshot, Ownable)
   - [Pub] #
      - modifiers: ERC20
   - [Pub] mint #
      - modifiers: onlyOwner
   - [Pub] snapshot #
      - modifiers: onlyOwner
   - [Pub] getCurrentSnapshotId
   - [Pub] burn #
   - [Pub] burnFrom #
+ [Int] IERC20Permit
   - [Ext] permit #
   - [Ext] nonces
   - [Ext] DOMAIN SEPARATOR
+ [Lib] Address
   - [Int] isContract
   - [Int] sendValue #
   - [Int] functionCall #
   - [Int] functionCall #
   - [Int] functionCallWithValue #
   - [Int] functionCallWithValue #
   - [Int] functionStaticCall
   - [Int] functionStaticCall
   - [Int] functionDelegateCall #
   - [Int] functionDelegateCall #
   - [Int] verifyCallResult
+ [Lib] SafeERC20
   - [Int] safeTransfer #
   - [Int] safeTransferFrom #
   - [Int] safeApprove #
   - [Int] safeIncreaseAllowance #
   - [Int] safeDecreaseAllowance #
   - [Int] safePermit #
   - [Prv] _callOptionalReturn #
+  TokenTimelock
   - [Pub] #
   - [Pub] token
   - [Pub] beneficiary
   - [Pub] releaseTime
   - [Pub] release #
+  liqLocker (TokenTimelock)
   - [Pub] #
      - modifiers: TokenTimelock
   - [Pub] charge #
   - [Pub] release #
+  Master (Ownable)
   - [Pub] #
   - [Pub] addressLiquidityPool
   - [Pub] addressLiquidityLocker
   - [Pub] payRent #
      - modifiers: onlyOwner
   - [Pub] charge #
   - [Pub] liqCharge #
   - [Pub] addLiquidityLocking #
      - modifiers: onlyOwner
   - [Pub] addLiquidity #
   - [Pub] removeLiquidity #
```

```
   - [Pub] mintMaster #
      - modifiers: onlyOwner
+  MARKETVAULT (Context)
   - [Pub] #
   - [Pub] getBackRate
   - [Pub] backingWithdrawl #
   - [Pub] buy #
```