

Auto Mining Token

AMT

Whitepaper

16, julio 2021

Resumen: Existen diversas técnicas sobre las cuales se ha intentado tokenizar la minería de BitCoin (BTC). Nuestra apuesta es crear un mecanismo único y sencillo para el usuario permitiendo mantener en todo momento sus tokens y transaccionarlos cuando quiera. Combinando la técnica de snapshot-token con la velocidad, seguridad y bajo costo de transacción de la BSC lograremos crear un ecosistema escalable, transparente y rentable para los inversores.

Introducción

Auto Mining Token es un ecosistema, que permite, a través de su token: AMT, invertir en proyectos de minería de BTC. Los usuarios que tengan AMT podrán cobrar sobre la rentabilidad generada por el proyecto en función de la cantidad de AMT que tengan sobre la totalidad de tokens en circulación. Es decir: Si una billetera posee, por ejemplo, el 50% de la totalidad de los tokens circulantes, cobrará el 50% de la rentabilidad pagada a los tenedores.

(Produccion de BTC - costos operativos) → 50% Lucro Directo

→ 50% Baúl de Garantía

Además, la mitad de la rentabilidad recibida diariamente por el tenedor del token (luego de deducidos los costos) se colocará en un contrato inteligente que funcionará como baúl. El baúl siempre le permitirá a los tenedores de AMT liquidar sus tokens a una tasa determinada por los BTC en el baúl sobre la totalidad de los AMT. De esta forma se irá generando un respaldo progresivo sobre el AMT que solamente podrá ser extraído a partir de la liquidación del token. Todos los tokens liquidados en el baúl serán quemados y sacados de circulación para siempre.

Podrán existir como **máximo 100 millones de AMT** que serán emitidos a lo largo del tiempo en función de la adquisición de mayor poder de minería. El proyecto comenzará a funcionar respaldado por aproximadamente 70.000 TH/s con una emisión de AMT calculada para que la rentabilidad sea interesante al inversor, y para que el minero pueda tolerar el impacto de la disminución de su poder de hash (en

beneficio del inversor del token) hasta que se adquiran nuevos equipos de manera tal de aumentar la rentabilidad a los inversores iniciales, y al minero aportante de poder de hash inicial, compartiendo el poder de hash nuevo introducido al ecosistema.

Transparencia y poder de hash

Todos nuestros equipos se conectarán a una pool de minería abierta con observadores públicos que pondremos a disposición en la web. De esta forma se podrá contrastar diariamente cuál es el beneficio que el proyecto entero está generando. Asimismo se avisará cuando se conecten nuevos equipos y se avisará con antelación la emisión de nuevos tokens.

Todo el esquema del proyecto contempla los costos de administración, energía y amortización con los tokens que se quedará el proyecto en sí mismo. Estos gastos serán tratados con transparencia y se le permitirá a los tenedores de AMT participar de las decisiones a través del voto.

Emisión Inicial AMT → 75%	Minero aportante poder hash
→ 20%	Market place (1 AMT= 1USD)
→ 5%	Aministracion y Marketing

Además, se incorporarán auditores externos e independientes para garantizar el buen funcionamiento de los Smart Contracts y la seguridad de los mismos.

Exposición a la minería de BITCOIN, liquidez y oportunidad para trading

Este proyecto le sumará al ecosistema general de exposición a la minería un nuevo mecanismo de cobros y aumentará la liquidez general. A diferencia de otros proyectos, no será necesario colocar los tokens en un Smart Contract o bloquearlos por cierto tiempo para recibir los beneficios. Además se creará una pool de liquidez en Pancake Swap y se bloquean los tokens de liquidez por un plazo de 2 años. Con un volumen de 1.000.000 de AMT y el equivalente en BTC de 1.000.000 usdt al momento del lanzamiento, para asegurar un valor inicial en paridad con el dólar americano. El bloqueo se realizará a partir de un Smart Contract derivado del TimeLock tradicional, agregando la funcionalidad poder cobrar rentabilidad por sobre los tokens bloqueados. De esta forma, garantizamos el desarrollo continuo del proyecto y la apuesta al mismo. Se le suma a la exposición a la minería en general, exposición al desarrollo mismo del proyecto en 2 sentidos:

1. Continuamente se aumentará el respaldo natural del AMT contra BTC a través del baúl que recibe parte de las rentabilidades. Garantizando siempre una salida a todos los holders que podrá ser, dependiendo del momento y del precio al que se compró el token, a pérdida o ganancia. Pero siempre con posibilidad de retiro con una parte de los beneficios generados a lo largo de toda la historia del proyecto.

2. Mejorando la relación AMT/HashPower con las nuevas inversiones de infraestructura. Siempre se apuntará a garantizar a los tenedores de AMT una mejor relación de poder de hash respecto a los nuevos que adquieran los recientemente emitidos. Este sistema continuará hasta alcanzar el punto de estabilidad de los 100 millones de tokens circulantes, donde comenzará a tener mucho más peso las definiciones respecto a las nuevas inversiones a realizar.

En todo sentido, debido a la operación que desarrollamos en territorio latinoamericano, logramos conseguir costos extremadamente competitivos de energía, mano de obra y adquisición de equipos. Con la unión de muchos emprendimiento de minería de la región nuestro token ofrecerá resultados superiores a los que se encuentran actualmente en el mercado.

Blockchain - BNB Smart Chain

La BNB Smart Chain (ex Binance Smart Chain) ha demostrado tener la capacidad, la seguridad y la confianza del público en general para albergar proyectos de alto porte. En ese sentido apostamos a utilizarla para tener la mejor relación entre costos de transacción de los usuarios, compatibilidad con billeteras, integración con otras dApps y la posibilidad de utilizar el BBTC como representación del BitCoin dentro de esta red.

Wrapped BitCoin - BTCB

BTCB es un activo BEP2/BEP20 envuelto (fijado) en Binance Chain/Binance Smart Chain con una vinculación 1:1 a BTC bloqueada en la cadena de bloques de Bitcoin. BTCB funciona con un modelo centralizado y basado en la confianza. La vinculación 1:1 significa que la cantidad de BTCB envuelto es igual a la cantidad de BTC bloqueada en una dirección pública. El modelo centralizado y basado en la confianza implica que el emisor de los tokens BTCB envueltos es Binance.

La transparencia se garantiza a través de la página web Prueba de activos (<https://www.binance.org/en/assets-proof>), donde se puede verificar el suministro actual de todos los activos bloqueados y emitidos en la cadena de bloques pública. La cantidad de BTC bloqueados puede no ser exactamente la misma que la de los BTC envueltos porque estos datos de auditoría no se actualizan en tiempo real, sino que se procesan semanalmente.

Una vez los BTC sean minados, utilizaremos un bridge de Binance para generar los BBTC respectivos y utilizarlos para pagar las rentabilidades. Además garantizamos la existencia de billeteras que tendrán BBTC de respaldo para que, ante eventuales fallas del bridge, se pueda realizar el pago correspondiente en el tiempo determinado.

Aspectos técnicos del token AMT

AMT será desarrollado cumpliendo el estándar ERC-20 (BEP-20 dentro de la BSC). De esta forma garantizamos su portabilidad con cualquier billetera y aplicaciones externas que los reconozcan.

Se le añadirá la funcionalidad de snapshot utilizando la derivación del estándar desarrollada por Open Zeppelin (<https://github.com/OpenZeppelin/openzeppelin-contracts/blob/master/contracts/token/ERC20/extensions/ERC20Snapshot.sol>)

Esta variante permite definir momentos específicos (snapshots) donde el balance de una billetera puede ser consultada de forma retroactiva. Estos momentos serán determinados por operaciones privilegiadas de administrador en la misma operación que realiza el pago de rentabilidad. De esta forma se garantiza que el cobro pueda realizarlo exclusivamente y por única vez los tenedores de AMT en el momento que se realizó el pago. Inclusive, la cantidad total de AMT en circulación quedará registrada en el snapshot, de forma tal que los tenedores podrán cobrar en función a la relación específica de sus AMT en función al total del momento del pago, por lo cual, sus tokens no perderán valor con futuras emisiones.

Generación de rentabilidad para proveedores de liquidez

Debido a que la posibilidad de proveer liquidez en mercados descentralizados será un derecho público para todos los tenedores. Y visto que el contrato de la pool de liquidez es quien tiene los tokens en ese caso. Implementamos un sistema por el cual inclusive aquellos que estén proveyendo liquidez en Pancake Swap puedan cobrar las rentabilidades generadas por el AMT.

Se implementará un contrato que enmascara el proceso de provisión de liquidez permitiendo que quienes provean a través de nuestra interfaz obtengan en vez del token de liquidez LP original obtengan un token auxiliar llamado liqAMT emparejado en relación 1:1 con el LP token respectivo. El contrato resguardará el token LP y lo utilizará cuando el usuario decida retirar la liquidez, quemando los liqAMT emitidos y devolviendo los tokens de la pool representados por el LP.

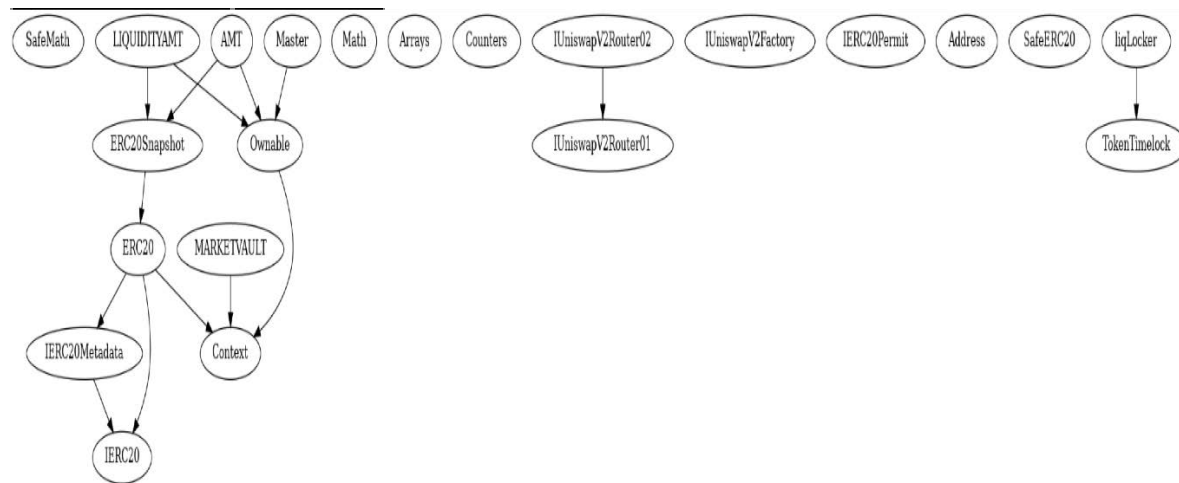
Mercado y baúl

Las ventas iniciales del token serán realizadas por un Smart Contract creado específicamente para este fin que además obrará como baúl de respaldo de AMT recibiendo su respectivo porcentaje por cada pago de rentabilidad realizado. En este sentido el proceso de ventas será 100% transparente.

Además, el baúl recibirá BBTC sistemáticamente con cada pago, entregandolos únicamente a los tenedores de AMT en caso que quieran liquidar su posición en una relación establecida por el *balance de BBTC del baúl : AMT liquidado*. Quemando los AMT en el proceso. De esta forma la relación establecida sólo puede ir en aumento,

garantizando siempre un precio mínimo de liquidación a los usuarios que es creciente en el tiempo. Implementamos este sistema porque creemos firmemente en el BTC y promovemos su desarrollo y entendemos que es fundamental para los inversores contar con un mecanismo transparente de respaldo. No existirá otra forma de retirar los BBTC del baúl más allá que la establecida mediante liquidación de AMT.

Estructura



Integración de mineros externos

Crearemos un sistema por el cual mineros que no se encuentren en nuestros establecimientos puedan participar del proyecto. En virtud del poder de hash aportado se le garantizara un aumento en el poder de hash de un 5%, debido a la capitalización obtenido por la venta de los tokens emitidos con el aval de este aporte externo, podremos usarlo para:

1. Amortizar el poder de hash aportado en caso de que ante una eventual desconexión los AMT generados sigan respaldados.
2. Adquisición de nuevos equipos que permitan mejorar la relación general entre AMT:HashPower

De esta forma, en tanto nuestra capacidad energética nos lo permita, lograremos trasladar la inversión de minería de BTC de zonas menos rentables a más rentables. Permitiendo al minero externo una transición suave y más rentable que la que obtendría minando en un lugar menos rentable.

Conclusiones finales y objetivos

Si bien es imposible al día de hoy generar un sistema 100% descentralizado de tokenización de minería de BitCoin, nuestro ecosistema intenta alcanzar la máxima transparencia que el estado del arte actual de la tecnología de blockchain lo permite.

Nos encontramos aquí ante una integración de varias metodologías aplicadas por otros proyectos de manera separada para construir un ecosistema pujante en un entorno competitivo y azotado por volatilidades de mercado.

Nuestro objetivo al final del día es construir sistemáticamente la confianza de los usuarios en el AMT para lograr, no únicamente su valorización, sino que un crecimiento sostenido en el tiempo de la minería en territorios rentables. Aportando con la inversión obtenida a través de la tokenización una escala comercial que permite competir de manera firme contra el resto de las ofertas del mercado.

Anexo

1- Estructura de Smart Contracts

- Smart Contracts

- ○ Propios (Serán verificados en bscscan al momento del deploy)
 - AMT.sol
 - LIQUIDITYAMT.sol ■ liqLocker.sol
 - MARKETVAULT.sol ■ Master.sol
- ○ Importados (open zeppelin github) ■ ERC20
 - ERC20-snapshot ■ TokenTimelock ■ Ownable
- ○ Importados (Uniswap liquidity conexions) ■ IUniswapV2Router02

2- Vista general de funciones

```
( $\$$ ) = 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] backingWithdrawal #
- [Pub] buy #
```