$FAIR · BNB CHAIN · AGENT ID FAIR MINT

FAIR Whitepaper

Built by someone who got cut.
A project no one can change.

Rules written in stone. No one gets a say.

Status notice - read first

The $FAIR contracts are **deployed on mainnet (11 November 2026)** and **verified exact_match on Sourcify** (creation and runtime).

FairToken: `0x008455724ed189d21b869842b2ff1c1a570951d1` (block 132976338)
FairMinter: `0xe99b9b3a34bf276d817d5a92973cc4e097e30c89` (block 132976340)

Any token using the $FAIR name at any other address is a copycat and is not this project. Verify the address character by character before minting.
Mint opens: **15 November 2026, 12:00 CST (Sunday)**
Fallback liquidity date: **15 December 2026, 12:00 CST (Tuesday)**

Version 1.0 (English) · November 2026
This document describes rules that are fixed in contract code. Where anything in this paper and the deployed, verified contract disagree, the contract wins - and you should not mint.

Abstract

$FAIR is a fair mint for BNB AI Agent ID holders on BNB Chain. It replaces trust with code: a fixed price, instant delivery, a zero team allocation, pro-rata liquidity, and a transaction tax that can only ever be used to buy back and burn.

Most fair launches ask you to trust the team. $FAIR is designed so you do not have to. Every promise in this paper maps to a line of immutable contract code that anyone can read before minting opens, and verify on-chain after deployment.

The model in one paragraph

One share costs **0.001 BNB** and delivers **5,000,000 FAIR** to your wallet in the same transaction - no waiting period, no second claim step. Supply is capped at **100,000 shares**. Each Agent ID may mint at most **5 shares**. The token supply of **1 trillion FAIR** is split evenly: 500 billion for the mint pool, 500 billion for the liquidity pool, and **zero for the team**. When the mint sells out - or 30 days after it opens, whichever comes first - all BNB raised is paired with a pro-rata amount of tokens on PancakeSwap V2, the LP tokens are sent to the burn address, and every unsold token is burned. A 2% buy tax and 4% sell tax, fixed forever, are collected in tokens, swapped to BNB, and used only to buy back FAIR and burn it.

Contents

1. Background: why this project exists
2. Design principles: the seven iron rules
3. Tokenomics
4. Mint mechanics and the Agent ID gate
5. Liquidity: finalize() and the pro-rata formula
6. Tax and buyback-and-burn
7. Contract architecture and immutable parameters
8. How to verify everything yourself
9. Testing and self-audit status
10. Timeline and roadmap
11. Risks
12. FAQ, disclaimer and appendix

1. Background: why this project exists

I am not a project team. I am someone who mints projects.

Over the past month I minted more than a hundred BNB Agent ID projects. Again and again, the same pattern:

- **WHALE** took 20% of every share straight into the project wallet, and refunded only 80% when a round did not fill.
- **BAGENT** promised free daily mining - then, on the second morning, quietly required buying 5 paid shares first to unlock it.
- **Broccoli** offered 500 free shares that bots took in two minutes. My wallet was eligible and had never claimed. I simply lost a typing race.

I can accept losing money. What I cannot accept is that the rules themselves are rigged: promises can be edited, payout wallets can be swapped, start times can slip, and "free" can turn into "paid" overnight. As long as the rules live in someone's hands, "fair" is just a slogan.

So $FAIR inverts the logic: **do not trust me - you do not need to.**

- Team allocation: 0 shares, 0 tokens.
- Mint proceeds go only into the contract, never into a personal wallet.
- Sell out, and liquidity is added immediately. Miss the sell-out, and liquidity is still added after 30 days, pro-rata to what was sold. Unsold tokens are burned.
- Buy tax 2%, sell tax 4%. Tax is collected in tokens, auto-swapped to BNB, and used to buy back and burn - all fixed in code.
- Source code is verified on Sourcify before launch. After deployment there is no pause, no blacklist, no tax change, no address change - because those functions do not exist.

> **Verify first, mint second.** When the contract address is published, check it on-chain yourself using Section 8 of this paper. If anything fails verification, do not mint.

2. Design principles: the seven iron rules

Each rule below is enforced by contract code, not by a promise. The "how to check" column tells you where the enforcement lives.

Rule 1 - Team gets zero. Of the 1 trillion supply, 500 billion go to the mint pool and 500 billion to the LP pool. Team, marketing, treasury, advisor and airdrop allocations are all 0. *Check: token distribution in the verified source; deployer balance must be 0 after setup.*

Rule 2 - One fixed price. 0.001 BNB per share, 5,000,000 FAIR per share, 2 x 10^{-10} BNB per token. The price is an immutable constant; there is no function to change it. *Check:* `pricePerShare` *is* `immutable` *in FairMinter.*

Rule 3 - Five shares per ID, maximum. The cap is keyed to the hash of your Agent ID handle, not just your wallet address, so switching wallets does not reset it. *Check: the per-handle counter in FairMinter.*

Rule 4 - All proceeds go to liquidity. 100% of BNB raised is paired in finalize(). The minter contract has no withdrawal function at all. *Check: search the source for withdraw / rescue - there are none.*

Rule 5 - LP tokens are burned. LP tokens are sent directly to 0x000...dEaD in the same finalize() transaction. No team wallet, no timelock that later expires into someone's hands. *Check: the addLiquidityETH recipient in FairMinter.*

Rule 6 - Tax only buys back and burns. Buy 2%, sell 4%, fixed constants. Tax never reaches a personal wallet; it accumulates in the token contract and can only leave as a buyback sent to the burn address. *Check: BUY_TAX_BPS / SELL_TAX_BPS constants; buybackAndBurn() destination.*

Rule 7 - No admin powers. No owner to renounce because the minter has no owner functions. No pause, no blacklist, no minting after deployment, no tax or router changes. The deployer's only privilege is a one-time setMinter during setup. *Check: full function list in the verified source.*

3. Tokenomics

Item	Value
Token	FAIR ($FAIR) on BNB Chain (BSC)
Total supply	1,000,000,000,000 FAIR (1 trillion), fixed at deployment, no further minting
Mint pool	500,000,000,000 FAIR (50%) - 100,000 shares x 5,000,000
LP pool	500,000,000,000 FAIR (50%) - used pro-rata at finalize(); the unused part is burned
Team / marketing / treasury / advisors / airdrops	0 FAIR (0%)
Price per share	0.001 BNB
Tokens per share	5,000,000 FAIR, delivered instantly
Price per token	$0.001 / 5{,}000{,}000 = 2 \times 10^{-10}$ BNB
Maximum raise	100 BNB (100,000 shares sold out)
Per-ID cap	5 shares (0.005 BNB) per Agent ID
Minimum IDs to sell out	20,000 distinct Agent IDs
Buy / sell tax	2% / 4%, fixed, all to buyback-and-burn

The cap of 100,000 shares is deliberate. A 10,000-share mint can be cornered by a small group; 100,000 shares need at least 20,000 different Agent IDs to sell out, which spreads the supply far wider and makes a bot sweep of the whole mint impractical.

What is NOT in the tokenomics

- No presale, no whitelist round, no private allocation at any price.
- No staking yield, no reflections, no promised APY. The only deflation mechanism is the tax-funded buyback-and-burn described in Section 6, plus the one-time burn of unsold tokens at finalize().
- No reserved supply for future "ecosystem" use. If it is not the mint pool or the LP pool, it does not exist.

4. Mint mechanics and the Agent ID gate

4.1 Direct mint, instant delivery

Call `mint(shares)` with exactly `shares x 0.001 BNB`. Your FAIR arrives in the same transaction. There is no subscription period, no allocation lottery and no second claim transaction. The mint has no early closing switch: it ends only when it sells out or when finalize() runs.

4.2 Who can mint: verified Agent ID holders

Minting is gated by the Binance Agent ID (.ai), the SPACE ID Agent ID NFT on BSC mainnet:

Item	Value
Registry contract	`0x60B9262456dE011F7d15a54CdD8c48bAaa8B022F`
Contract name / symbol (on-chain)	"SPACE ID Agent ID" / "AI" (ERC-721)
Check 1	`balanceOf(your wallet) > 0` - you hold at least one Agent ID NFT
Check 2	`handleOf(your wallet)` is non-empty - a .ai name is actually registered
Cap key	`keccak256(handle)` - the 5-share cap follows your ID, not your wallet

Both checks must pass. This double check (balance plus non-empty handle) is the same method used by multiple live Agent ID projects. Note: the ERC-8004 registry (0x8004A169...) is a different identity system and is **not** accepted here.

> **Positive sample verified on mainnet (11 Nov 2026):** a real holder wallet returned balanceOf = 1 and handleOf = "scat" - both checks pass for a genuine Agent ID holder.

4.3 Before the start time

The start time - 15 November 2026, 12:00 CST - is an immutable value written into the contract at deployment. Calling mint() earlier reverts with "not started". Nobody, including the deployer, can open the mint early or delay it.

5. Liquidity: finalize() and the pro-rata formula

5.1 Two triggers, first one wins

1. **Sold out:** when all 100,000 shares are sold, finalize() can be called immediately.
2. **30-day fallback:** 30 days after the start (15 December 2026, 12:00 CST), finalize() can be called even if the mint has not sold out.

finalize() is **permissionless**: anyone may call it, the deployer has no priority and no ability to block it. Calling it before either trigger reverts with "too early". One honest caveat: finalize() is not automatic - someone has to send the transaction and pay a small gas fee. Any holder has the incentive to do it, because trading cannot start until it happens.

5.2 The formula

Let **S** = shares sold and **D** = BNB raised = S x 0.001.

- LP tokens = 500,000,000,000 x S / 100,000 = S x 5,000,000 FAIR - exactly equal to the amount minted.
- The pool is created with those tokens plus **all** of D.
- Opening price = D / LP tokens = 2×10^{-10} BNB - **exactly the mint price, at any sell-through level**.
- Unsold mint-pool tokens, (100,000 - S) x 5,000,000, and the unused LP-pool tokens are burned in the same transaction.
- LP tokens go directly to 0x000...dEaD. Liquidity can never be withdrawn.

Shares sold	Minted	LP tokens + BNB	Burned	Opening price
5,000 (5%)	25 billion	25 billion + 5 BNB	950 billion	2×10^{-10} BNB
50,000 (50%)	250 billion	250 billion + 50 BNB	500 billion	2×10^{-10} BNB
100,000 (100%)	500 billion	500 billion + 100 BNB	0	2×10^{-10} BNB

5.3 Why pro-rata matters

If the LP always received the full 500 billion tokens regardless of sales, then at 5,000 shares sold the opening price would be 1/20 of the mint price - minters would open 95% down through no fault of their own. That design quietly transfers value from minters to whoever buys the open. Pro-rata LP keeps the opening price pinned to the mint price at every outcome. This is the single most important fairness mechanism in $FAIR.

6. Tax and buyback-and-burn

> **Read this before anything else in this section:** buying costs you 2% and selling costs you 4%. A round trip costs roughly 6%, plus swap fees and gas. There is no short-term trading edge in $FAIR. This is stated up front, not buried.

6.1 The tax

- Buy tax **2%**, sell tax **4%** - compile-time constants (200 / 400 basis points). There is no setTax function.
- Tax applies only to buys and sells against the PancakeSwap V2 pair. Minting is untaxed, and ordinary wallet-to-wallet transfers are untaxed.
- No personal wallet is exempt. The only exemptions are technical necessities: the token contract itself, the minter, and the router. The pair is deliberately not exempt - that is how the tax is collected.

6.2 Why the tax is collected in tokens first

On PancakeSwap V2 a token contract cannot receive BNB directly from a swap, so the tax cannot be charged in BNB at the moment of trade. Instead, tax is deducted in FAIR, accumulates in the token contract's tax pool, and is converted in two permissionless steps:

1. `swapTaxToBNB()` - once the tax pool is worth at least 0.05 BNB, anyone can call this to swap the pooled FAIR into BNB, which stays inside the token contract as the buyback pool.
2. `buybackAndBurn()` - once the buyback pool holds at least 0.05 BNB, anyone can call this to buy FAIR on PancakeSwap and send it straight to 0x000...dEaD. No intermediate wallet, ever.

Both steps have a 1% slippage cap and reentrancy protection. The deployer cannot pause them, front-run them by privilege, or redirect them.

6.3 The honest cost of this design

Converting tax to BNB and buying back takes two swaps. Fees, slippage and sandwich bots will eat a small part of the tax pool, so the amount actually burned will be slightly less than the tax collected. That is the price of a buyback design (which supports the price) over simply burning the tax tokens (which does not). We state it rather than hide it.

7. Contract architecture and immutable parameters

7.1 Two contracts

Contract	Job
FairToken	The FAIR ERC-20: fixed 1 trillion supply minted once in the constructor, pair tax logic, tax pool, buyback pool.
FairMinter	The mint: Agent ID gate, per-ID cap, BNB custody, finalize() liquidity and burn.

Neither contract is a proxy: no delegatecall, no selfdestruct, no upgrade path.

7.2 Immutable parameters

Parameter	Value	Locked by
Price per share	0.001 BNB	immutable
Max shares	100,000	immutable
Per-ID cap	5 shares	immutable
Start time	15 Nov 2026, 12:00 CST (unix 1794715200)	immutable
Fallback finalize	Start + 30 days (15 Dec 2026, 12:00 CST)	immutable
Buy / sell tax	2% / 4%	constants
Tax / buyback threshold	0.05 BNB equivalent	constant
Buyback slippage cap	1%	constant
LP recipient	0x000...dEaD (burn)	hardcoded
Agent ID registry	0x60B9262456dE011F7d15a54CdD8c48bAaa8B022F	constructor

7.3 Powers that do not exist

- No owner on FairMinter at all; nothing to renounce because nothing was granted.
- No pause for minting or trading. Minting stops only by selling out, by finalize(), or by not having started.
- No blacklist, no whitelist, no freeze.
- No additional minting after the constructor. No change to tax, price, cap, router or registry.
- No withdrawal or rescue function in either contract. BNB leaves FairMinter only into the pool, and FairToken's BNB only into a buyback that burns.
- The deployer's only act is one-time `setMinter` during setup - set once, only while unset; if set wrong, the only fix is redeploying. Hence the Section 8 checks.

8. How to verify everything yourself

You do not need to trust this paper. The mainnet addresses are on the cover and in Appendix A. Run these checks before minting:

1. **Source verification:** on Sourcify, the token and minter contracts must show **exact_match**. Both already do (verified 11 Nov 2026; links in Appendix A). Partial match or no match = do not mint.
2. **Supply location:** the minter contract must hold the full 1 trillion FAIR before the start, and the deployer wallet must hold **0**. If the deployer holds any FAIR, the "team zero" claim is false. Verified on-chain on 11 Nov 2026: the minter held the full 1 trillion and the deployer held 0.
3. **setMinter:** FairToken's minter must be set, and must equal the published FairMinter address.
4. **Parameters:** read price (0.001 BNB), maxShares (100,000), startTime (1794715200) and totalSold() directly from the contract on BscScan.
5. **Function audit:** in the verified source, search for withdraw, rescue, pause, blacklist, setTax, and any owner-only function. None should exist (other than the one-time setMinter, already consumed).
6. **Registry:** confirm the Agent ID registry in the contract is 0x60B9262456dE011F7d15a54CdD8c48bAaa8B022F - not the ERC-8004 registry and not a lookalike.
7. **After finalize():** confirm the LP token balance of 0x000...dEaD, the burned FAIR balance of the dead address, and that FairMinter's FAIR and BNB balances are zero.

> **Impostor warning:** the official addresses are on the cover and in Appendix A. Every other "$FAIR" token is fake. Compare the address character by character - copycats differ by one or two characters.

9. Testing and self-audit status

Done and not-done are listed separately. Nothing in the "not done" list is described anywhere in this paper as complete.

9.1 Done

- **Local test suite: 17/17 passing** (Foundry, FairTest), covering locked mainnet parameters, instant mint delivery, the 5-per-ID cap, early-mint and early-finalize reverts, pro-rata liquidity maths, unsold burn, and tax constants.
- **Live fork test:** the full flow was executed on an Anvil fork of the BSC testnet (Chapel, chain 97) against the *real* PancakeSwap V2 testnet router and factory: mint 5 shares delivered 25,000,000 FAIR instantly; a 6th share for the same ID reverted; finalize() after a simulated 30 days created a real pair, sent LP tokens to the dead address and burned 999,950,000,000 unsold FAIR.
- **Tax verified to the token on a real router:** a 0.0001 BNB buy quoted 489,234.87 FAIR and delivered 479,450.17 (exactly x 0.98), with 9,784.697 FAIR entering the tax pool. A 200,000 FAIR sell added exactly 8,000 FAIR (4%) to the tax pool.
- **Buyback chain verified:** tax pool FAIR was swapped to BNB, then buybackAndBurn() repurchased 9,745.613 FAIR and sent it directly to the burn address.
- **Line-by-line self-audit** against a hostile-project checklist (hidden withdrawals, team skims, mutable price, early open, pause, blacklist, extra minting): no such functions found. The minter's BNB can exit only into the pool.
- **Agent ID gate verified on mainnet** with a real holder wallet (balanceOf = 1, handleOf = "scat").
- **Mainnet deployment: done 11 Nov 2026.** FairToken `0x008455724ed189d21b869842b2ff1c1a570951d1` (block 132976338) and FairMinter `0xe99b9b3a34bf276d817d5a92973cc4e097e30c89` (block 132976340). The minter holds the full 1 trillion FAIR; the deployer holds 0.
- **Sourcify verification: exact_match.** Both contracts are exact_match on Sourcify, creation and runtime (verified 11 Nov 2026).

9.2 Not done - stated plainly

- **Sold-out finalize on the fork:** not run - it would need 20,000 distinct IDs. The sold-out branch uses the same finalize() code path and its arithmetic is covered by local tests.
- **Third-party audit:** not commissioned. The self-audit above is exactly that - a self-audit. Treat it accordingly.

10. Timeline and roadmap

Date (CST)	Milestone	Status
11 Nov 2026	Design locked (v2); contracts written; local tests 17/17; fork test passed; Agent ID gate verified on mainnet	Done
11 Nov 2026	Mainnet deployment; Sourcify exact_match verification; deployer-zero and minter-balance checks published; contract addresses posted on the mint page	Done
15 Nov 2026, 12:00 (Sun)	Mint opens (hardcoded in contract)	Scheduled
On sell-out	finalize() - liquidity at mint price, LP burned, trading opens on PancakeSwap V2	Automatic trigger, manual call
15 Dec 2026, 12:00 (Tue)	Fallback finalize() if not sold out - pro-rata liquidity, unsold burned	Scheduled
After opening	Ongoing: permissionless tax conversion and buyback-and-burn; listing applications (DexScreener, CoinGecko) once the pool exists; progress announcements per milestone	Planned

There is no roadmap item that requires new contract powers, because none exist. Anything promised for "later" is either an off-chain action (announcements, listing applications) or a permissionless function anyone can already call.

11. Risks

- **Thin pool risk:** if few shares sell, the pool is shallow. At 1,000 shares the pool is 5 billion FAIR + 1 BNB, and large orders will move the price violently. The mint page shows shares sold, BNB raised and the implied opening price - check them before minting.
- **No refunds:** minted FAIR arrives instantly and there is no refund path. If the mint underperforms, your exit is the pool after finalize() - at whatever price the market gives, minus 4% sell tax.
- **finalize() needs a human:** it is permissionless but not automatic. If nobody calls it at the trigger, trading waits until somebody does.

11. Risks (continued)

- **First-come competition:** a direct mint is first-come, first-served. A 100,000-share cap reduces bot advantage but cannot remove it.
- **Sell-out is not the base case:** raising the full 100 BNB needs 20,000 IDs. The 30-day pro-rata fallback is a designed outcome, not a failure mode - but it means a smaller pool than the headline numbers.
- **Tax drag:** roughly 6% round-trip tax plus swap fees and gas means mint-and-flip does not pay. If your plan needs a quick exit, this is the wrong token.
- **Buyback leakage:** two swaps stand between tax collection and burning; fees, slippage and MEV will consume part of it. Net burn will be less than gross tax.
- **Pre-open transfers:** FAIR is transferable wallet-to-wallet before the pool exists (untaxed, by design). There is no official pre-open market; anyone "selling" FAIR before finalize() is an OTC counterparty risk on you.
- **Deployment risk:** moving the full supply into the minter and setting setMinter correctly was a human-run script; it was verified on-chain on 11 Nov 2026 (minter 1 trillion, deployer 0). Section 8's checks exist so you can confirm this yourself.
- **Market risk:** nothing here promises price, liquidity, listings or returns. FAIR can go to zero. Mint only what you can afford to lose entirely.

12. FAQ

Why no free first share?

Free shares get farmed by bots and multi-wallet scripts - the opposite of fair. One uniform price for everyone is the fair rule.

Can the team change the rules later?

No. The functions to change them do not exist. That is the whole point, and it is checkable in the verified source.

What if it does not sell out?

On 15 December 2026 at 12:00 CST, anyone can call finalize(). Liquidity is added pro-rata at the same 2 x 10^{-10} BNB opening price, and all unsold tokens are burned.

What if nobody calls finalize()?

Then trading has not started yet, and the first holder who wants to trade will call it for a small gas fee. The deployer cannot block it.

Why is there no refund?

Tokens are delivered in the same transaction as payment, so there is nothing to refund against. A refund mechanism would need custody and discretion - the two things this design removes.

Is $FAIR audited?

Not by a third party. It has a public self-audit and full tests (Section 9). Do not let anyone - including this paper - upgrade that into "audited".

Disclaimer and appendix

Disclaimer

This whitepaper is a technical description of a smart contract system. It is not financial advice, not an offer of securities, and not a promise of profit, price, liquidity or listing. Cryptocurrency is high-risk; the value of FAIR may fall to zero. You are responsible for your own verification (Section 8), your own wallet security, and compliance with the laws that apply to you. Statements about testing describe tests actually run on the dates stated; mainnet behaviour was re-verified at deployment on 11 November 2026. This edition lists the deployed addresses and verification links in Appendix A.

Appendix A - Key addresses and constants

Item	Value
Agent ID registry (mainnet)	`0x60B9262456dE011F7d15a54CdD8c48bAaa8B022F`
Burn address	`0x000000000000000000000000000000000000dEaD`
PancakeSwap V2 (mainnet)	Router / factory as published in the verified contract source at deployment
FairToken address	`0x008455724ed189d21b869842b2ff1c1a570951d1` Block 132976338 · BscScan · Sourcify
FairMinter address	`0xe99b9b3a34bf276d817d5a92973cc4e097e30c89` Block 132976340 · BscScan · Sourcify
Start time	1794715200 = 15 Nov 2026, 12:00 CST
Fallback	Start + 30 days = 15 Dec 2026, 12:00 CST

Appendix B - Worked example: 5% sell-through

5,000 shares sell for 5 BNB. Minters hold 25 billion FAIR. At finalize(): 25 billion FAIR + 5 BNB enter the pool (opening price 2×10^{-10} BNB, the mint price), LP tokens go to the dead address, and 475 billion unsold mint-pool tokens plus 475 billion unused LP-pool tokens - 950 billion in total - are burned. Final circulating supply: 50 billion FAIR, half held by minters, half in the pool. Team: still zero.

Rules written in stone. No one gets a say.

$FAIR Whitepaper v1.0 (English) - November 2026 - End of document