

Dickie & Sloane

White Paper

August 4, 2025

The Project

Dickie & Sloane is more than just a pair of tokens, it's a self-contained market system built for traders, holders, and anyone who understands the power of volatility.

When you buy DICKIE, you pay a 5% tax: 4% is instantly used to buy SLOANE and send it to your wallet, and 1% is added to the liquidity pool. When you buy SLOANE, the same thing happens in reverse. 4% buys DICKIE for you, and 1% goes to liquidity. The result? Every buy strengthens liquidity and gives you a piece of the other token automatically.

But Dickie & Sloane can do more than sit still. Through an attachable and replaceable contract called the Gap Engine, taxes on one or both tokens can be changed temporarily to create price imbalances. This deliberate disruption fuels arbitrage opportunities for active traders. When the run is over, taxes return to their standard 4% + 1% split, restoring stability for holders. Multiple Gap Engines can be created for different strategies and swapped in, each with its own schedule and system for stirring the market.

All trades pass through another swappable contract called the Helper. The Helper can be replaced to adapt to new DEXs, trading pairs, or DeFi innovations without changing the tokens themselves. This makes Dickie & Sloane future-proof. These tokens are designed to evolve alongside the market, not be left behind by it.

Although ownership can be renounced, it's not always the best move. Dickie & Sloane have extensive settings, from router management to liquidity thresholds to contract upgrades that can be managed by a single trusted wallet, a multi-sig, or a governance contract controlled by the community. And maintenance costs aren't a problem: on BNB Smart Chain, gas is cheap, and these tokens can even reimburse owner-called gas fees when funds are available from the liquidity tax.

Dickie & Sloane isn't just a token pair, it's a micro-market ecosystem where cross-rewards, liquidity growth, and strategic disruption can work together to create profit potential on both sides of the trade.

Token Specs

Name: Dickie
Ticker: DICKIE
Cashtag: \$DICKIE
Supply: 69,304,420
Decimals: 18
Blockchain: BNB Smart Chain
Contract Address: 0x00

Name: Sloane
Ticker: SLOANE
Cashtag: \$SLOANE
Supply: 69,304,420
Decimals: 18
Blockchain: BNB Smart Chain
Contract Address: 0x00

Deployment

1. The Helper contract was deployed first from BIG_TRADER_HELPER_v1.sol with 200 runs.
2. The Hub contract was then deployed from BIG_TRADER_v1.sol with 200 runs along with the Helper contract address and PancakeSwap's router address as constructor arguments. The DICKIE and SLOANE contracts were deployed automatically as part of this transaction.

- Deployer Address: 0x00
- Hub Address: 0x00

The first trading pairs were created on PancakeSwap for both DICKIE and SLOANE.

- 60,000,000 DICKIE / 0.006 BNB
- 60,000,000 SLOANE / 0.006 BNB

The liquidity pool (LP) tokens were sent to the null address, making them irretrievable and rendering the initial liquidity provided to the trading pairs for each token rug-proof.

- DICKIE LP tokens burned at:
0x00
- SLOANE LP tokens burned at:
0x00

The Hub Contract

- Central Control Point
 - Deploys both Dickie & Sloane at the same time.
 - Stores both token addresses permanently. No external setter exists, ensuring the sister relationship can never be changed.
- Router Management
 - Maintains a whitelist of approved DEX routers for swaps and liquidity.
 - Supports adding and removing routers in bulk.
 - Enforces that at least one router must always be active.
- Helper & Gap Engine Management
 - Stores the addresses of the Helper and optional Gap Engine contracts.
 - Can update these addresses after verifying that they're valid contracts with required functions.
 - Includes time locks and permanent locks to prevent changes after a certain point if desired.
- Trade Notifications
 - Receives trade events from both tokens (`notifyTrade`) and forwards them to the Gap Engine if set.
 - Allows the Gap Engine to adjust taxes or track activity in real time.
- Locking Mechanisms
 - Time-Based Lock: Freeze changes for a set number of days.
 - Permanent Lock: Freeze forever.
 - Separate locks for the Helper and Gap Engine.
- Ownership & Security
 - Uses two-step ownership transfer for safety.
 - Supports renouncing ownership entirely if desired.
 - Includes reentrancy protection across critical functions.
- Gas Refund Support
 - Owner-called functions can automatically refund gas (if funds available), funded from liquidity tax revenue.

- Recovery Functions
 - Can recover stuck tokens or BNB, excluding:
 - BNB.
 - The token itself (Dickie/Sloane).
 - The sister token (Dickie/Sloane).
 - LP tokens.
 - Includes events for all recovery actions for transparency.
 - Surplus BNB or surplus restricted tokens can be claimed since they are not being used in any operations or are not owed to a holder.

The Token Contracts (DICKIE & SLOANE)

- Standards & Metadata
 - BEP-20 compliant (`balanceOf`, `transfer`, `approve`, `transferFrom`, etc.).
 - `decimals = 18`.
 - Fixed supply per token: 69,304,420 minted once to the initial owner.
- Twin/Sister Relationship
 - Each token's sister is derived from the Hub (no external setter).
 - The link is immutable after deployment.
- Buy Tax Logic (Default)
 - Total buy tax: 5% (in basis points).
 - Sister rewards: 4% of the buy amount is taken in the bought token, accrued, then swapped via the Helper into the sister token for the buyer.
 - Liquidity: 1% of the buy amount is accrued for automated liquidity adding.
 - Sells: no sister/liquidity accrual by default (still classified for rules/notifications).
- Sister Reward Delivery
 - On buy, `sisterTokensOwed[buyer]` is credited.
 - The Helper handles the swap/delivery; if a swap can't execute (slippage/liquidity), the owed balance remains and can be claimed anytime by the holder (`claimSloaneTokens` on DICKIE; `claimDickieTokens` on SLOANE).
- Auto-Liquidity
 - Liquidity amounts accrue on buys.
 - When `liquidityThreshold` is reached (owner-configurable), the token approves the Helper and calls `addLiquidity()`.
 - (LP token handling/burn is performed on the Helper side according to your deployment configuration.)
- Dynamic Taxes (Gap Engine)
 - Before applying defaults, the token queries the Hub's Gap Engine.
 - If active, the engines (`totalTaxInBasisPoints`, `sisterTaxInBasisPoints`, `liquidityTaxInBasisPoints`) override the defaults for that trade's flags.

- Trade Classification & Notifications
 - The Helper classifies each transfer as buy / sell / liquidity via `classifyTransfer(...)`.
 - After a buy/sell, the token notifies the Hub (`notifyTrade`) with amount and side.
- Launch Restrictions (Anti-Bot / Anti-Whale)
 - Max Buy = 1% of supply (693,044.2 tokens).
 - Max Sell = 1% of supply (693,044.2 tokens).
 - Max Wallet = 2% of supply (1,386,088.4 tokens).
 - Cooldown = 1 tx per block per wallet.
 - The owner can lift restrictions forever when ready.
 - The deployer, helper contract, token contracts, routers and trading pairs are exempt from these initial restrictions.
- Ownership & Safety
 - Two-step ownership transfer (`transferOwnership` → `acceptOwnership`).
 - Renounce is available but not automatic at launch.
 - Critical paths use a reentrancy lock (`lockGuard` / `_locked`).
- Gas Reimbursement
 - Owner-called functions marked with `refundGas` attempt to auto-refund gas (best-effort, only if contract BNB balance is sufficient).
 - The refund pool is effectively funded by the liquidity tax economics.
- Recovery & “Stuck” Funds
 - Separate `claimStuckBNB` / `claimStuckTokens` exist in the hub and token contracts with additional safeguards.
- User-Facing Convenience
 - Each token provides buy helpers with BNB:
 - DICKIE: `buyDickieTokens(minOut)` and `buySloaneTokens(minOut)`
 - SLOANE: `buySloaneTokens(minOut)` and `buyDickieTokens(minOut)`
 - Each token exposes a claim function for owed sister token rewards.

- Events & Transparency
 - Emits standard `Transfer/Approval` plus `SisterTokensOwed`, `SisterTokensSent`, `PairSet`, `LiquidityThresholdSet`, `RestrictionsLifted`, and ownership transfer events for on-chain auditability.

The Helper Contract (the doer)

What it is:

The Helper is the execution engine that touches the DEX. It classifies trades, performs swaps, and adds liquidity. The tokens *ask*; the Helper *does*.

What it does:

- Classifies transfers as buy / sell / liquidity so the token knows which rules apply.
- Delivers sister rewards: swaps the accrued cut into the sister token and sends it to the buyer (or leaves it claimable if a swap can't execute).
- Adds liquidity automatically when the token's liquidity bucket hits its threshold.
- Handles BNB→token buys for convenience functions (site buttons, simple flows).
- Reads config for a token via getter functions (no heavy storage inside Helper).

How it's controlled & secured:

- Whitelisted routers only (set by the Hub).
- Only the Dickie/Sloane tokens can call the Helper's sensitive functions.
- Upgradeable via the Hub with timelocks/permanent locks available.
- The Hub sanity-checks that state-changing Helper functions are not callable by the Hub itself (extra safety).

Why it exists:

It keeps the token contracts lean and future-proof: if DEX behavior or routing needs to change, you update the Helper (subject to locks) because the tokens cannot be altered.

The Gap Engine Contracts (the tuner)

What it is:

An optional controller that can override tax settings in real time. It tunes how much goes to sister rewards vs. liquidity and the total tax, per trade type.

What it does:

- Sets effective taxes trade-by-trade (total %, sister %, liquidity %), overriding defaults if active.
- Responds to market conditions: can calm things down (more liquidity, less rewards) or stir action (more rewards) to create tradable price gaps.
- Receives trade pings from the Hub ([notifyTrade](#)) for analytics/reactivity.

How it's controlled & secured:

- Plug-and-play: the Hub can set/replace/remove it (subject to time/permanent locks).
- Version-aware: tokens/HUB try V2 first, fall back to V1 interface safely.
- Fails safe: if no Gap Engine is attached or if it throws an error, tokens use their built-in defaults (5% total: 4% sister, 1% liquidity).
- To remove the Gap Engine's functionality, either update the contract address in the Hub to the zero address or see if the Gap Engine contract has a toggle function to turn it on or off.
- Different Gap Engine contracts can be made to handle taxes differently.

Why it exists:

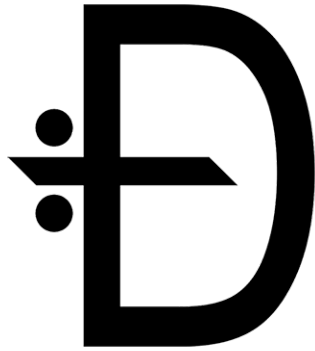
It gives you a policy layer to balance opportunity for traders (temporary gaps, volume bursts) with stability for holders (return to defaults), since tokens cannot be altered.

How They Work Together

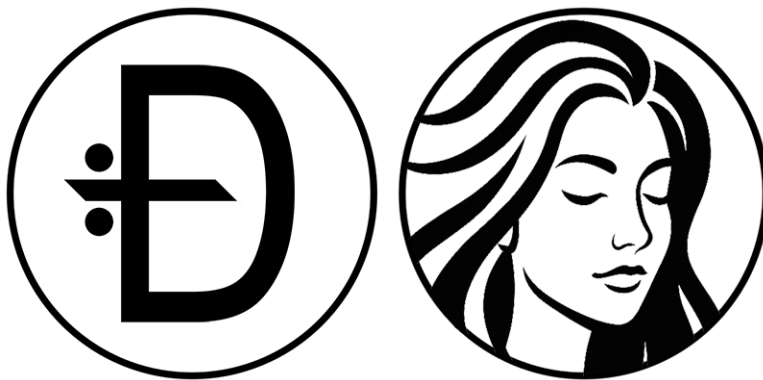
1. Buyer trades → Token asks Helper to classify the transfer.
2. Token asks the Gap Engine (if set) for the taxes to use; otherwise uses defaults.
3. Token applies taxes → accrues sister + liquidity.
4. When ready, the token authorizes the Helper to deliver the sister token and add liquidity.
5. The Hub is notified of buys/sells and can forward data to the Gap Engine.

Logos

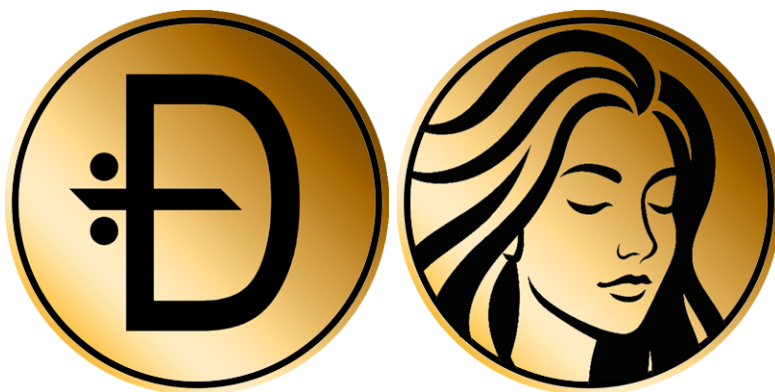
Dickie Currency Symbol:



Dickie Coin Sides 1 and 2:



Dickie Coin Sides in Color:



Sloane Currency Symbol:



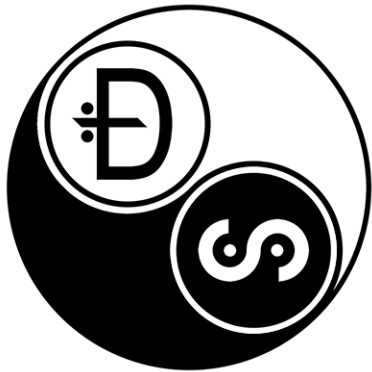
Sloane Coin Sides 1 and 2:



Sloane Coin Sides in Color:



This logo represents the project as a whole and will be used whenever a single logo is needed.



AI Models

AI generated model, Dickie:



AI generated model, Sloane:



Dickie and Sloane are more than just mascots, they are the front-facing personalities of a uniquely designed dual-token economy. They can be seen in memes and marketing.

Roadmap

- Draft a plan for *Dickie & Sloane*.
- Compose a simple white paper.
- Create a website.
- Set up social media accounts.
- Deploy the tokens to the BNB Smart Chain.
- Create trading pairs on PancakeSwap.
- Get listed on CoinMarketCap.
- Transfer ownership.
- Circulate freely in the crypto space.

The Team

There is no team, only a deployer who will oversee the launch and then pass ownership to the community at the first opportunity. No minting, burning or rug pulls are possible during this process.

Website: dickieandsloane.com

Telegram: [@dickieandsloane](https://t.me/dickieandsloane)

X: [@dickieandsloane](https://x.com/dickieandsloane)