Solana Cashback Token - Developer Documentation

Overview

The Solana Cashback Token is an SPL token with a unique reward mechanism where **buyers receive a portion of the sell tax, rather than holders**. The last **$5,000 of buy volume** forms a **rolling reward pool**, and 90% of the sell tax is redistributed proportionally among buyers in this pool.

This contract also includes **MEV bot protection, Sybil resistance, and anti-arbitrage mechanisms** to prevent abuse.

Tokenomics

Buy & Sell Tax Structure

- **Buy Tax:** 0% (to encourage buy pressure)
- **Sell Tax:** X% (configurable at deployment, recommended: 5-10%)
    - **90% of sell tax** is distributed to the most recent **$5,000 of buy volume**.
    - **10% of sell tax** is allocated for **buybacks and liquidity injections**.

Reward Distribution Mechanics

- The **last $5,000 of buy volume** is tracked dynamically.
- Buyers **own a percentage of this $5k pool** based on their purchase size.
- When a sell occurs, 90% of the **sell tax is distributed** proportionally based on a buyer's **% ownership of the $5k pool**.

Example of Reward Calculation:

- **Alice buys $2,500** → Owns **50% of the $5k pool**.
- **Bob buys $1,500** → Owns **30% of the $5k pool**.
- **Charlie buys $1,000** → Owns **20% of the $5k pool**.
- A **sell occurs, generating 5 SOL in sell tax rewards**.
    - Alice gets **2.5 SOL** (50%)
    - Bob gets **1.5 SOL** (30%)
    - Charlie gets **1 SOL** (20%)

If a new buy occurs and pushes out an older buy, the pool shifts, ensuring that **only the latest $5k of buy volume is eligible for rewards**.

Smart Contract Design

1. Buy Volume Tracking System

- Maintain a **dynamic rolling $5k buy pool**.
- Track each buyer's **wallet address, buy amount, and timestamp**.
- Ensure **new buys push out older buys**, keeping the **total tracked buy volume at exactly $5,000**.

Data Structure:

```
struct Buyer {
    address wallet;
    uint256 amount;
    uint256 timestamp;
}
Buyer[] public lastBuyers;
uint256 public totalBuyVolume;
```

- When a new buy occurs:
    - If `totalBuyVolume + newBuyAmount > 5000`, remove **oldest buys** until total volume is <= $5k.
    - Update the buyer list dynamically.

---

2. Sell Tax Redistribution

- **When a sell occurs:**
    - **90% of the tax** is distributed to buyers in the **last $5k buy pool**.
    - Distribution is weighted based on each buyer's **proportional share** of the $5k.
    - **10% of the tax is used for buybacks and liquidity injections**.

Logic for Sell Tax Distribution:

```
function distributeSellTax(uint256 taxAmount) internal {
    uint256 totalRewards = taxAmount * 90 / 100; // 90% for rewards
    uint256 buybackAmount = taxAmount * 10 / 100; // 10% for buybacks

    for (uint256 i = 0; i < lastBuyers.length; i++) {
        uint256 buyerShare = (lastBuyers[i].amount * totalRewards) / totalBuyVolume;
        payable(lastBuyers[i].wallet).transfer(buyerShare);
    }
    // Buyback & liquidity logic here
}
```

---

3. MEV Bot & Sybil Protection

To prevent **bots and exploiters from farming rewards**, the contract includes:

1️⃣ Front-Running & MEV Protection

- Detects if a buy transaction occurs **in the same block as another** and **flags it as ineligible**.
- Detects high **gas priority fees** as an indicator of MEV activity.

2️⃣ Sybil Wallet Farming Prevention

- Blocks wallets from **receiving sell tax rewards** if they have:
  - **Bought and sold within a short time window** (e.g., 10 minutes).
  - **Multiple small buys from the same IP address**.
  - **Spam-bought multiple times in the same block**.

3️⃣ Arbitrage & Wash Trading Protection

- Implements a **sell cooldown (e.g., 10 minutes)** to prevent rapid in/out trades.
- If a wallet **buys and sells the exact same amount** in a short window, **they are disqualified from tax rewards**.

Disqualification Logic in Solidity:

```
mapping(address => bool) public disqualified;
function checkForMEV(address wallet, uint256 amount) internal {
    if (tx.origin != msg.sender || tx.gasprice > highGasThreshold) {
        disqualified[wallet] = true;
    }
}

function checkForSybil(address wallet, uint256 amount) internal {
    if (block.timestamp - lastSellTime[wallet] < 600) {
        disqualified[wallet] = true;
    }
}
```

---

4. Additional Features

Buyback & Auto-Liquidity (10% of Sell Tax)

- Uses **a portion of sell tax** to buy back tokens, reducing sell pressure.
- Helps stabilize the token price and prevent price crashes.

Flash Cashback Events

- **Temporary 2x rewards** for a set time window.

- Encourages users to buy **during low volume periods**.

Last Buyer Bonus Jackpot

- The **last buyer before a sell event gets an extra 5% of the sell tax**.
- Encourages **buy sniping** to time the last buy before sells.

Deployment & Configuration

Contract Parameters (Editable at Deployment)

Parameter	Description
`sellTaxRate`	% tax applied to sells (recommended 5-10%)
`buybackAllocation`	% of tax used for buybacks (recommended 10%)
`minBuyAmount`	Minimum buy size to be eligible for tax rewards
`antiMEVProtection`	Enables/disables front-running protection
`sellCooldownPeriod`	Time limit between buy & sell to prevent arbitrage

Deployment Process

1. Deploy contract with **configurable tax rates & protections**.
2. Provide liquidity & launch trading.
3. Enable **MEV and Sybil filtering mechanisms**.
4. Monitor **reward distribution & buyback automation**.

Conclusion

This contract introduces an **innovative buy-to-earn model**, forcing **constant buy pressure while disincentivizing dumping**. With **MEV bot protection, Sybil resistance, and arbitrage prevention**, this token is built for **sustained growth and viral trading activity**.

This project is designed for **high-volume, high-engagement Solana traders**, ensuring **FOMO-driven perpetual buy demand** while keeping **Ponzinomics sustainable**. 🚀