WHITEPAPER · VERSION 1.0 · BY JAGUARBYTE LABS

THE DECENTRALIZED AUTONOMOUS WALLET

macroswap

Rules you write on-chain. A wallet that follows them, even while you are offline. Custody that never leaves your hands.

I

Non-custodial

Funds stay in your wallet until each purchase.

II

Trustless

No proxy, no upgrade, fees fixed in code.

III

Runs offline

Sign once. Execution needs no one online.

IV

100% on-chain

Every decision that moves value is code.

ABSTRACT

MacroSwap is a set of immutable smart contracts that lets a wallet follow its owner's investment rules (buy a Bitcoin-led basket only when Bitcoin dips, accumulate from a personal deposit address, return to stablecoins at a target) without handing custody to anyone.

The rules are written to the blockchain the moment the owner signs. From then on, execution needs neither the owner, nor the MacroSwap website, nor any server operated by the project: any address may trigger an order once the contract itself has confirmed, from Chainlink prices, that the conditions are met. Funds remain in the owner's wallet until the instant of each purchase, and the purchased tokens return to that same wallet within the same transaction. We call this design a **Decentralized Autonomous Wallet (DAW)**. This paper describes its principles, mechanism, architecture, fee model and risks.

00 CONTENTS

Contents

01	The problem	Presence or trust
02	The DAW	A new primitive
03	Four principles	The design contract
04	Non-custody	Your keys, your funds
05	Trustless by construction	What no one can change
06	Runs while you are offline	Autonomy & resilience
07	100% on-chain architecture	Two layers
08	The dip order	Mechanism
09	The basket	Allocation
10	Instant, sell & take-profit	Every direction
11	Fees	Fixed in code
12	The Tracker	Soulbound record
13	Security	Engineering practice
14	Deployments	Official addresses
15	Roadmap: Solana	Next network
16	Risks & disclosures	Read before use

01 THE PROBLEM

Discipline requires presence. Automation has required trust.

Buying weakness is simple in principle and hard in practice. Drawdowns arrive at three in the morning, markets move faster than people do, and discipline fades exactly when it matters.

Until now, automating that discipline has meant giving something up:

- **Exchange bots** require a deposit. The funds become a claim on a company, exposed to its solvency, its freezes and its terms of service.
- **Third-party trading services** require API keys or signing permissions. Whoever holds them can act on the account.
- **Self-hosted scripts** require a machine that never sleeps, with a private key living on a server.

Each option pays for convenience with custody, with trust or with uptime. MacroSwap is built so that none of the three is required.

> The strategy should live where the money lives: in the owner's wallet, enforced by code that no one can change.

02 THE DAW

The Decentralized Autonomous Wallet

MacroSwap introduces a simple primitive: a wallet that keeps its own funds and follows rules written by its owner, executed by an open network instead of a company.

DEFINITION

A Decentralized Autonomous Wallet is a wallet that follows rules its owner wrote on-chain: rules that anyone may execute, no one may alter, and that never let anyone but the owner withdraw.

Three parties take part, and none of them has to trust the others:

THE OWNER

Writes the rules

Creates an order or opens a deposit address, approves a budget, and can cancel, revoke or withdraw at any moment.

THE CONTRACTS

Enforce the rules

Verify every condition against oracle prices, cap every amount, route every swap and deliver every token to the owner.

THE KEEPERS

Press the button

Anyone. They trigger execution when the contract allows it and receive the tip the owner chose. They decide nothing.

The result covers the strategies long-term holders actually use (buying the dip, dollar-cost averaging, buy and hold, and hedging back to stablecoins at a target) inside a wallet that never stops being the owner's.

03 PRINCIPLES

Four principles, enforced in code

These are not aspirations or policies. Each one is a property of the deployed contracts, and anyone can verify it from the published source.

I · NON-CUSTODIAL

Your funds never wait with us

Stablecoins stay in your wallet until the instant of each purchase. Purchased tokens arrive in your wallet in the same transaction. No contract keeps a user balance between calls.

Section 04 →

II · TRUSTLESS

No one can change the deal

No proxy and no upgrade path. The basket, routes, oracles and fees are fixed in the code. The few remaining parameters are bounded, public and delayed by a 24-hour timelock.

Section 05 →

III · AUTONOMOUS, EVEN OFFLINE

Sign once. Close the laptop.

Your rules live in contract storage, not in an app session. No open tab, no running bot, no API key, no server holding your key. The site, the API and our keepers can all go dark and your orders remain executable.

Section 06 →

IV · 100% ON-CHAIN

Every decision is code

Price, eligibility, allocation, routing, minimum output and settlement are all computed by the contracts from on-chain data. Off-chain software only reads state and proposes transactions.

Section 07 →

04 NON-CUSTODY

Your keys. Your funds. Every second.

MacroSwap never sees a private key, a seed phrase or a password. It never pools user money, and it never holds a balance on anyone's behalf.

4.1 Approve a budget, not an account

An order is created with `createOrder` and funded by a standard ERC-20 `approve` for the budget alone. The stablecoins do not move. They stay in the owner's wallet, where they can be spent, moved or reassigned at any time.

When the conditions are met, the contract pulls **only the amount of that single purchase**, and within the same transaction it pays the fixed fee to the treasury, pays the chosen tip to the caller, swaps the remainder through on-chain pools and sends every purchased token **directly to the owner**.

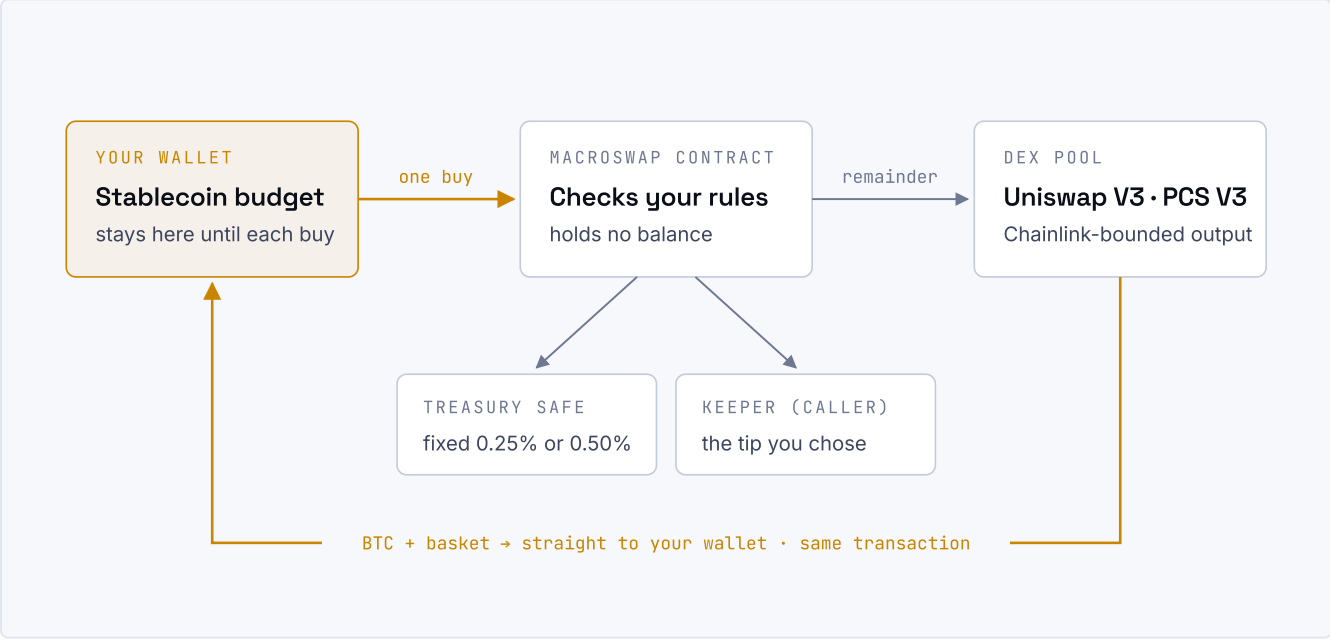


FIGURE 1 A purchase is one atomic transaction. Either every step succeeds or nothing moves.

4.2 A personal deposit address that only you control

For owners who prefer to send stablecoins from any wallet or exchange, each wallet has its own deposit address: an EIP-1167 minimal clone, derived deterministically with CREATE2 from the owner's address. It can receive funds before it is even deployed.

That address has exactly **two exits**, and both lead to the owner:

- a purchase of the basket, delivered to the **owner** (never to the sender, which may be an exchange hot wallet);

– `withdrawDeposit`, which only the owner can call, for any token held there.

Neither MacroSwap, nor the keeper, nor whoever sent the deposit can take funds out by any other path.

4.3 Exit at any moment

The owner can leave at any time, without permission and without a fee: `cancelOrder` ends an order, an `approve` of zero revokes the allowance, `withdrawDeposit` empties the deposit address, and the sell contract only ever touches the tokens the owner explicitly approved to it.

What MacroSwap can never do. Move your stablecoins outside the amounts and conditions you signed · send purchased tokens anywhere but your wallet · withdraw from your deposit address · read or use your keys, seed phrase or signatures beyond the transactions you approve.

05 TRUSTLESS

Trustless by construction

Trust is a liability. MacroSwap is designed so that using it requires no trust in the team, the keepers or the website: only in code that anyone can read and that no one can replace.

5.1 No master key

The contracts have **no proxy and no upgrade mechanism**. The bytecode you approve today is the bytecode that runs forever. Source code is verified on Etherscan and BscScan. A future version would be a new contract, and every owner would decide, freely, whether to migrate.

5.2 A basket and fees that cannot be edited

There is no function to change the basket's tokens, swap routes or price feeds. An internal review established that any key able to change a route could, in principle, divert swaps into a pool it controls, so those functions were removed before deployment. Fees are constants (0.25% and 0.50%) with no setter at all: not even the project's own Safe can raise, lower or waive them.

5.3 Narrow, public, delayed governance

A handful of operational parameters remain, each with a hard ceiling written in code. None can touch user funds. Every change follows the same path: `propose` emits a public `Proposed` event with the full calldata, a **24-hour timelock** runs, and only then can `executeProposal` apply it, within a 7-day window. Calling a setter directly reverts.

PARAMETER	HARD BOUNDS IN CODE	EFFECT
Default allocation	BTC 75–100% · A ≤ 25% · B ≤ 5%	Only for wallets that did not set their own mix
Deposit tip	≤ 1%	Covers keeper gas on Instant purchases
Minimum per purchase	$5 – $1,000	Keeps each swap economically sensible
Dust threshold	$0.01 – $5	Slices below it are routed to BTC
BTC-only ceiling	$0 – $1,000	Small purchases buy BTC only

The owner role is a 2-of-2 Safe, set at deployment, with no `transferOwnership`. It can call `renounceOwnership`, freezing every parameter permanently. The treasury, a separate 2-of-2 Safe and immutable, only receives fees and can recover tokens sent to the contract by mistake.

5.4 A price nobody makes up

Every price comes from **Chainlink**, never from a pool's spot price, which can be manipulated within a block. The contract builds its own price history from rounds it reads directly on each feed: `snapshot` is open to anyone, and `backfill` accepts only round identifiers, re-reading price and timestamp on the feed itself, so no data can be invented. Each feed has its own staleness limit, measured from its heartbeat. Every swap has a minimum output derived from the oracle minus a per-token tolerance.

5.5 A keeper without power

Execution functions (`execute`, `executeMany`, `sweep`, `takeProfit`) are open to any address. The caller chooses nothing: amounts, routes, recipients and limits come from the contract and the owner's rules. A keeper cannot force swaps to fail by starving them of gas, because a reserved gas floor makes the whole transaction revert instead. Project keepers submit through private transaction channels to avoid sandwich attacks.

THE PROJECT'S SAFE…	CAN
Move, freeze or withdraw user funds	NEVER
Change fees	NEVER
Change tokens, routes or oracles	NEVER
Upgrade or replace contract logic	NEVER
Block anyone from executing, cancelling or withdrawing	NEVER
Adjust bounded parameters, after a public 24 h delay	YES
Renounce control permanently	YES

Trustless means the rules are in the code and can be checked by anyone, not that the code cannot have bugs. See Risks & disclosures.

06 OFFLINE

It runs while you are offline

A MacroSwap order is not a session, a subscription or a bot. It is state on the blockchain. Once signed, it waits for its conditions for hours or months, with no one watching.

6.1 You can be offline

Create the order, close the browser, switch off the phone. There is no tab to keep open, no software to keep running and no API key to keep alive, and your private key never has to sit on a server. When Bitcoin reaches your conditions, any keeper executes the order and the tokens are already in your wallet the next time you look. Nothing you do, or forget to do, while away can stall it.

> Sign once. The rules keep working while you sleep, travel or forget the password to the website.

6.2 We can be offline

The contracts depend on nothing outside the blockchain. Every off-chain component is a convenience, built so that its failure stops no order, no deposit and no withdrawal.

IF THIS GOES DARK	WHAT KEEPS WORKING	HOW
The website	Orders, deposits, sales, withdrawals	The “Use without the site” page lists official addresses and step-by-step explorer flows; the site builds as a content-addressed IPFS package
The API and edge cache	Everything that moves value; the app reads the chain directly	The Tracker can render its metadata and image entirely on-chain
The project's keepers	Execution	Execution is open to anyone, including you, through any block explorer; the tip goes to whoever executes
A public RPC	App, keeper and API	Multiple RPC endpoints with automatic failover
The project itself	Every deployed contract	Immutable code; open `snapshot` and `backfill` keep price history alive

6.3 Stepping in yourself

If no keeper runs, the owner, or anyone else, can execute directly from Etherscan or BscScan:

01 **Check**
Read `canExecute(yourAddress)`. It returns `true` or the exact reason it is waiting.

02 **Execute**
Call `execute(yourAddress)` from any wallet. The contract re-checks every rule.

03 **Receive**
The basket arrives in your wallet, and the tip you set comes back to you as the caller.

07 ON-CHAIN

A 100% on-chain core

MacroSwap separates what decides and settles from what merely displays and relays. The first is entirely on-chain. The second holds no funds, no user keys and no authority.

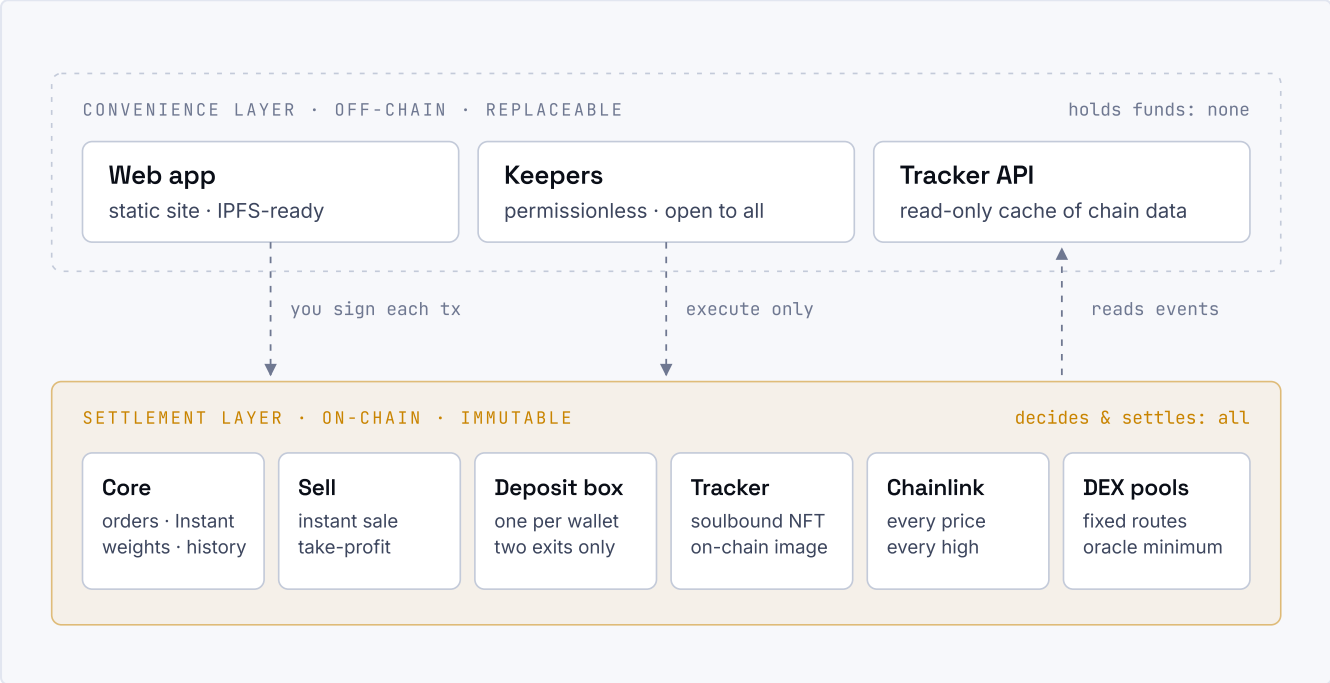


FIGURE 2 Off-chain software reads and proposes. It never decides and never holds.

CONCERN	WHERE IT IS DECIDED
Is Bitcoin in a dip?	Contract, from Chainlink rounds and on-chain highs
Is this purchase allowed now?	Contract: dip, reference, interval, budget, balance, allowance
How is it split across the basket?	Contract: owner's mix, drawdown weights, exclusions
Which pool, what minimum?	Contract: fixed routes, oracle price minus tolerance
Where do the tokens go?	Contract: always to the owner, in the same transaction

CONCERN	WHERE IT IS DECIDED
When is it triggered?	Anyone, but only when the contract says yes

08 MECHANISM

The dip order

A dip order is a limit order on Bitcoin's weakness. It buys only when Bitcoin is meaningfully below a recent high, and, by default, only when the purchase lowers the order's average price after costs.

PARAMETER	MEANING
Budget & amount per buy	Total to deploy and the size of each purchase
Dip window	30-day high, ~1-year high (53 weekly buckets) or all-time high
Dip threshold	Minimum drop below that high, from 1%
Reference rule	Automatic margin (default), a fixed margin up to 50%, or off
Maximum wait	After this period without a buy, the next dip buys even above the reference (default 180 days)
Minimum interval	Spacing between purchases, from one hour
Keeper tip & expiry	Tip per purchase (up to 5%) and the order's end date

An order executes only when every condition holds at once:

```
// 1 · the dip signal (mandatory)
drawdown = 1 - P_BTC / High_window  ≥ dip threshold
// 2 · the reference rule
P_BTC ≤ Ref × (1 - margin)    or  elapsed ≥ maxWait
Ref    = BTC price at creation, then the order's average BTC price
margin = fee + tip + 0.30%            // automatic
// 3 · housekeeping
interval elapsed · budget left · balance · allowance
```

The automatic margin means each purchase is meant to lower the order's average *after* the MacroSwap fee, the keeper tip and an estimate of pool costs. The maximum wait answers the opposite risk: a market that only rises would otherwise never let the order buy again. After that period, the next qualifying dip buys and the clock restarts.

Price history lives in the contract: 30 daily buckets per feed, 53 weekly buckets for Bitcoin, and an all-time high that only rises, set from rounds verified on the feed itself. In long downtrends, the yearly and all-time windows can stay in a dip for months, so the reference rule becomes the effective filter. Larger thresholds (30–40%) suit those windows.

`executeMany` processes several ready orders in one transaction, skipping any that cannot execute, so keepers share fixed costs across owners while every owner still receives their own tokens.

09 ALLOCATION

A Bitcoin-led basket

Bitcoin is the anchor. A measured share goes to major assets and leading DeFi tokens, weighted toward whichever has fallen the most.



75% Bitcoin

20% Majors

5% DeFi & alts

- **Your own mix.** Each wallet may set its split with `setMix`, within BTC 75–100%, group A up to 25% and group B up to 5%. It applies to orders and to the deposit address, and appears on the Tracker.
- **Weights that lean into weakness.** Inside each group, a token's weight is 5% plus its drop from its own 30-day high, with the drop counted up to 50%.
- **Automatic exclusion.** A token down more than 99%, or with a stale or broken oracle, leaves the basket on its own. Its share goes to the rest of its group.
- **Nothing wasted.** Slices below the dust threshold, and any swap that fails its minimum, go to BTC. Purchases of up to $100 buy BTC only.

	ETHEREUM	BNB CHAIN
DEX	Uniswap V3	PancakeSwap V3
Stablecoins	USDC (primary), USDT	USDT (primary), USDC
Bitcoin	WBTC	BTCB
Group A	WETH, SOL, BNB	ETH, SOL, WBNB
Group B	AAVE, LINK, UNI	AAVE, LINK, UNI, XRP, DOGE, LTC
Minimum per purchase	$100	$10

10 EVERY DIRECTION

Instant, sell & take-profit

The same guarantees hold in every direction: oracle-bounded prices, fixed fees, and tokens that only ever move to the owner.

10.1 Instant: buy and hold from a deposit address

Send USDC or USDT, from any wallet or exchange, to your personal deposit address. Once the balance reaches the network minimum, a keeper calls `sweep` and the basket is bought at once, with no dip signal, for a fixed 0.50% fee plus a gas tip of 0.30% on Ethereum and 0.15% on BNB Chain.

10.2 Two stablecoins, one rule

Each network accepts both USDC and USDT. The primary stablecoin pays when it has balance and approval; otherwise the second pays the whole purchase after conversion through a 0.01% pool, at **no less than 99.5% of 1:1**. Below that floor the purchase reverts, which protects against a loss of peg.

10.3 Sell: hedge back to stablecoins

A separate contract, `MacroSwapSell`, sells the basket back to the primary stablecoin, for a fixed 0.50%. It only touches the tokens the owner approved to it. Every swap is held to the oracle price minus tolerance: if a token would sell below it, that token simply stays in the wallet and the rest proceeds.

10.4 Take-profit: a ladder up

With `setTakeProfit`, the owner sets a gain and a share to sell. When Bitcoin reaches the reference times (1 + gain), a keeper sells that share for a fixed 0.25%. The reference is the higher of the entry price and the last automatic sale price, so each step climbs.

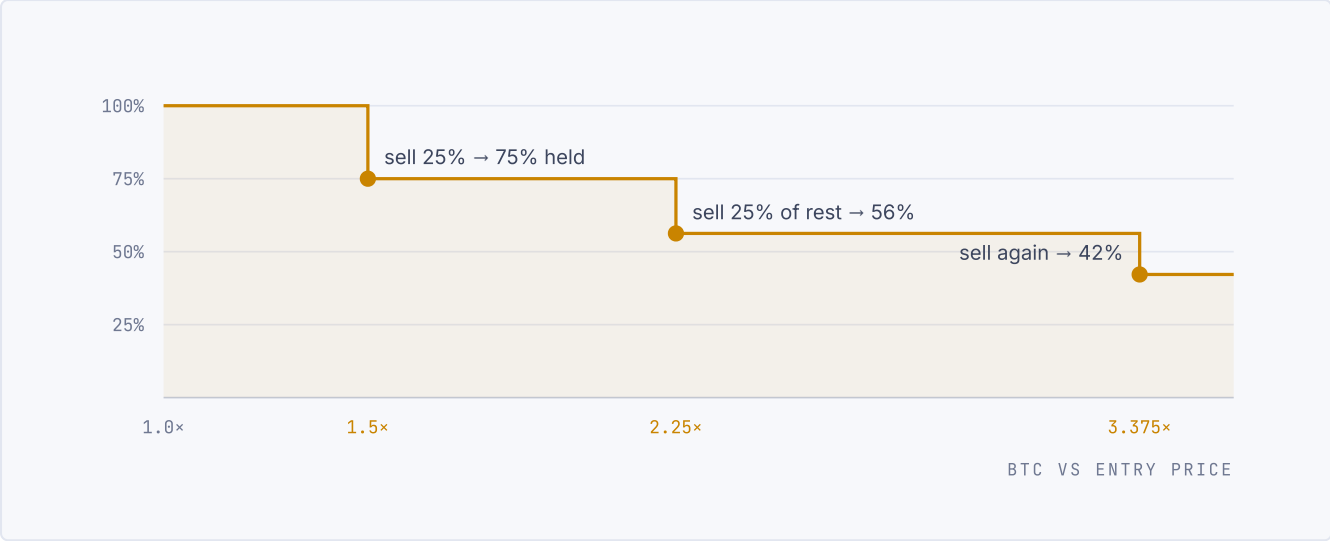


FIGURE 3 Illustrative ladder with a +50% gain and a 25% sale per step. Not a forecast.

The entry price can be set manually or left at zero to use the wallet's average buy price across all MacroSwap purchases. A sale of 100% closes the position. No sale executes below the network minimum.

11 FEES

Two fees. Fixed forever.

The model follows maker and taker logic: the patient pay less. There are exactly two fees, and both are constants in the bytecode.

PATIENT

0.25%

Dip orders & take-profit

Waits for your price: a purchase in the dip, or a sale at your target.

IMMEDIATE

0.50%

Instant purchase & sale

Executes now: a deposit-address purchase, or a market sale back to stablecoins.

- No function exists to change either fee, for anyone, including the project's Safe.
- Fees travel from the owner's wallet to the treasury Safe inside the same transaction. The contract never keeps them.
- Keeper tips are chosen by the owner on orders (up to 5% of each purchase) and are capped at 1% on deposits.
- No token, no emissions, no lock-ups, no withdrawal or cancellation fees. Network gas is paid by whoever executes.

MacroSwap has no token. Any “MacroSwap token”, presale or airdrop is not ours. The project will never ask for your seed phrase.

12 RECORD

The MacroSwap Tracker

A non-transferable record of how a wallet uses MacroSwap: purchases, average price, strategy and unrealized result.

- **Soulbound.** ERC-721 with ERC-5192 (`MSTRACK`). The token ID is the wallet address. Only the owner mints (free) and only the owner burns. Transfers and approvals revert.
- **No value, no rights.** It cannot be sold, carries no rewards, and no contract depends on it: burning it changes nothing in orders, averages or funds.
- **On-chain fallback.** Images are served from a cache in front of a read-only API. If that layer disappears, the Tracker can generate its metadata and SVG image entirely on-chain.
- **Reproducible.** Every figure it shows can be recomputed by anyone from public blockchain events.

13 SECURITY

Security as engineering practice

The safest privilege is the one that does not exist. MacroSwap's security begins by removing power, and continues with testing and operational separation.

PRACTICE	IN MACROSWAP
Privilege removal	Fixed basket, routes, oracles and fees; no upgrade path; bounded, timelocked parameters
Internal security review	Documented findings fixed before deployment: fixed basket, 24 h timelock, per-feed staleness, deposit tip capped at 1%
Fork testing	Test suites run against live mainnet state on both networks, with real pools and real oracles
Fuzzing & invariants	Property-based suites that try to break accounting and custody guarantees
Static analysis	Slither and Aderyn, with zero high or medium findings outstanding
Execution safety	Reentrancy guards, oracle-bounded minimum outputs, reserved gas floor, private transaction submission
Operational separation	The public API server holds no private keys; keepers run on a separate machine with no inbound ports; site and API live on separate domains and accounts
Front-end hygiene	No tag managers or session recording; error reports scrub addresses and signatures

14 DEPLOYMENTS

Official addresses

Live on Ethereum and BNB Chain since November 2026, at the same addresses on both networks, with verified source code.

CONTRACT	ADDRESS · ETHEREUM (1) AND BNB CHAIN (56)
MacroSwap Core	`0xB60e34a0edB0f73E83A8704E56141e12618184C1`
MacroSwap Sell	`0x5f9709738a2A436f6F736d73f7d7dFf1D72cc8C3`
MacroSwap Tracker	`0xa37b3fF7A1cABf77256a4B246eBa1652011e4569`
DepositBox (template)	`0x1bB44C20E6036a4Cd916cA718986b8E024efEd96`
Owner Safe (2 of 2)	`0xb9D326a805C2B2eC5D129AFEf1990e71B213b938`
Treasury Safe (2 of 2)	`0x1A07b1C15d9CE167b24aF15CC5A5a197Bb748599`

The only website you need is **macroswap.finance**. Data for the app and the Tracker is served from `api.macroswap.dev` and `nft.macroswap.dev`. Each deposit address is specific to its network: funds sent on the wrong network do not arrive there. Always confirm addresses on the block explorer before you sign.

15 ROADMAP

Next: Solana

A native port, not a bridge. The same guarantees, rebuilt for the Solana runtime.

- **Native program** in Rust with Anchor, covering the full scope: dip orders, Instant, per-wallet mix, the reference rule, timelock, sale, take-profit and Tracker.
- **Pyth prices and one fixed Orca Whirlpool per token**, recorded at initialization. Executors never choose routes.
- **Basket:** cbBTC, then SOL and ETH, then JUP, JTO, PYTH and AAVE. The Tracker uses Token-2022 with the `NonTransferable` extension.
- **Immutability:** the upgrade authority is to be revoked on mainnet, making the program as fixed as its EVM counterparts.
- **One disclosed exception:** because Pyth keeps signed prices for only ~197 days, the all-time high is recorded once before the program is sealed and only rises from then on. The value is public and can be checked against Chainlink.

The program runs on devnet today and is not deployed on Solana mainnet. Its mainnet launch is funded by community donations toward a 9 SOL goal; donations carry no token, airdrop or rights of any kind.

16 DISCLOSURES

Risks & disclosures

Removing custody and trust does not remove risk. These are the risks that remain, stated plainly.

Smart-contract risk

Immutable code cannot be patched. A bug would require a new contract and a voluntary migration. Trustless is not the same as bug-free.

Oracle risk

Decisions depend on Chainlink feeds. Stale or broken feeds exclude tokens automatically, but an incorrect feed would affect decisions.

Wrapped & bridged assets

WBTC, SOL and BNB on Ethereum are wrapped or bridged; BNB there is the legacy ERC-20 with thin liquidity. On BNB Chain most of the basket, and both stablecoins, are Binance-Peg tokens: custodial risk is concentrated in their issuer.

Stablecoin risk

USDC and USDT may lose their peg. The 99.5% conversion floor stops a purchase but cannot protect a balance already held.

Liquidity & slippage

Swaps may execute up to each token's tolerance versus the oracle (0.8% to 2.5%). Thin pools may cause a slice to fall back to BTC.

Execution risk

Keepers act when the tip covers gas. An order with a low tip may wait; anyone, including the owner, can still execute it directly.

Strategy risk

Buying dips during a prolonged decline keeps buying into it. The maximum wait may buy above the reference. Past behavior does not predict results.

Governance key risk

Both Safes are 2 of 2. Losing a key would freeze parameters and the treasury. It cannot affect user funds.

IMPORTANT

MacroSwap is open software, not a financial service, broker or adviser. Nothing in this document is investment, legal or tax advice, or an offer of any kind. **Not a promise of returns.** Crypto assets are volatile and you can lose money. You alone are responsible for your decisions, for complying with the laws of your jurisdiction and for verifying contract addresses before signing. This document describes the contracts as deployed; where it differs from the verified source code, the code prevails.



Buy the dip. On autopilot.

MacroSwap Whitepaper · Version 1.0 · November 2026 · macroswap.finance

MacroSwap is developed and published by **Jaguarbyte Labs**.