

AI Crypto Wallet Whitepaper

AI Crypto Wallet

A Self-Custodial Wallet with an On-Device AI Trading Agent

Version 1.0 | August 2026

Red Devil Team (红魔团队)

Abstract

AI Crypto Wallet is the first self-custodial cryptocurrency wallet that embeds an autonomous AI trading agent directly on the user's device. It separates the two concerns that have historically forced users to choose between convenience and control: the agent that reasons about markets, and the keys that authorize transactions. Private keys never leave the device and are never exposed to the AI. The agent reads market data, watches positions, proposes and executes trades within a hard-wired safety gate, and logs every decision to an auditable trail. The wallet supports ten EVM and non-EVM chains, a built-in Web3 DApp browser, and a platform token, R-MAB, that grants ecosystem participant privileges.

This document describes the system architecture, the agent runtime, the safety and risk framework, the multi-chain data layer, and the governance model. It is a technical specification, not an investment solicitation.

1. Introduction

Trading cryptocurrency well requires two abilities that are difficult to combine safely: continuous attention to volatile markets, and secure custody of the assets being traded.

Human traders face a structural problem. Markets move around the clock, technical indicators update continuously, and positions need monitoring between sleep cycles, work, and other obligations. Software can watch markets without fatigue, but giving third-party bots custody of funds introduces a trust asymmetry — the bot, or the platform that runs it, is a single point of failure and a tempting target.

Existing products resolve this tension by choosing a side. Signal services and trading bots take custody of the user's keys to execute on their behalf, sacrificing self-sovereignty for automation. Self-custodial wallets keep keys on-device but offer no trading intelligence, leaving the user to manually monitor indicators and act on fleeting signals.

AI Crypto Wallet is designed to reject that trade-off. It keeps the keys on the device and under the user's sole control, while a self-contained AI agent handles the continuous reasoning. The agent is not granted custody; it is given tools. Every write operation passes through a safety gate that the agent cannot override.

2. Problem Statement

2.1 The custody dilemma

In the current landscape, the trust model of most trading tools is inverted. A user who wants automated trading must surrender their private key or a signing capability to a remote service. This creates three failure modes:

- **Custody risk.** A centralized operator, or a compromised server, can move or lose user funds.
- **Counterparty risk.** The operator's interests may not align with the user's, and opaque execution can be gamed.
- **Single point of compromise.** A single key repository is a high-value target for attackers.

2.2 The attention problem

Even when a user retains custody, effective trading demands near-constant attention: reading RSI, MACD, moving averages, and Bollinger Bands across multiple timeframes; monitoring open positions; and reacting to stop-loss or take-profit levels. Manual execution is slow, error-prone, and emotionally driven.

2.3 The compliance and safety gap

Autonomous trading amplifies risk. An agent with full autonomy and no guardrails can over-trade, chase losses, or transact with genuinely malicious tokens. Any system that automates the movement of money must therefore be bounded by explicit, verifiable limits that the automation itself cannot disable.

The design of AI Crypto Wallet addresses all three problems at once: local custody, continuous analysis, and hard safety limits enforced at the execution layer.

3. Design Principles

Four principles govern the entire architecture.

Self-custody by construction. Private keys are generated and stored on the device. They are never serialized to disk in a location the agent can read, never transmitted, and never provided to the LLM. All signing happens inside the wallet’s signing module. The AI only ever expresses intent; it never holds the means to authorize.

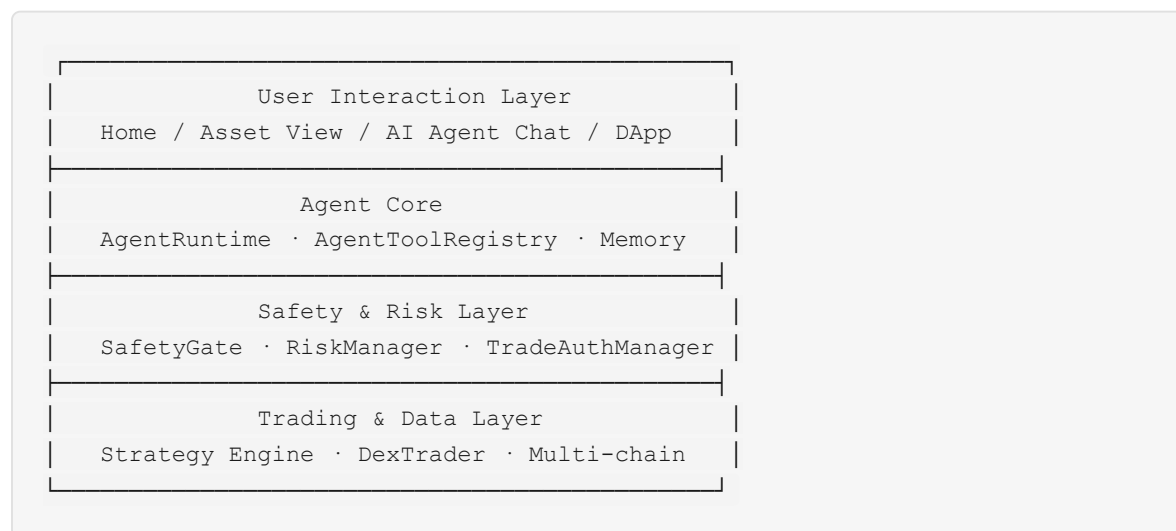
Safety before autonomy. Every write operation — a token swap, an approval, a transfer — passes through the SafetyGate, a mandatory gate that the agent cannot bypass. Circuit breakers, daily and per-trade limits, and a token whitelist are enforced here.

Evidence-driven decisions. The agent is prompt-bound to base conclusions on on-chain data and technical indicators, to state its reasoning, and to admit when data is insufficient rather than fabricate a signal.

Auditability. Every tool call, every safety check, and every decision is written to a structured log. The user can inspect the full reasoning trail for any executed action.

4. System Architecture

The system is organized into four layers. A user interaction layer (the wallet UI and the agent chat interface) sits on top of an agent core, which is wrapped by a safety tier, which in turn drives a trading and data layer.



4.1 Layer 1: User interaction

The wallet exposes a home screen with asset, market, trade, discovery, and profile tabs; a dedicated AI agent screen with chat and detail views; and a Web3 DApp browser. The AI is activated only when the user presses the agent button — it never starts autonomously.

4.2 Layer 2: Agent core

The agent core is the reasoning engine. It binds an LLM to a fixed set of tools through a bounded thinking loop, and grounds the model with a persistent memory file.

4.3 Layer 3: Safety and risk

The safety tier is the enforcement boundary between the agent's intent and the movement of funds. It is described in detail in Section 6.

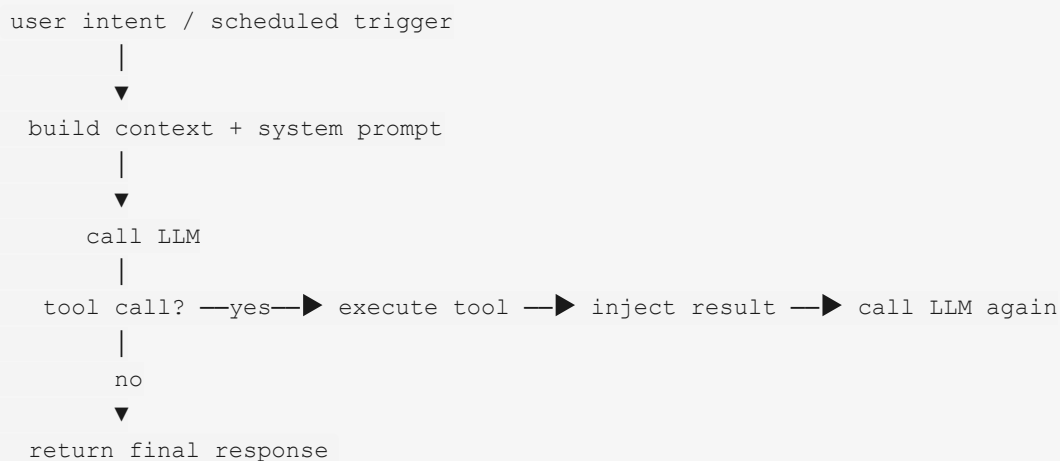
4.4 Layer 4: Trading and data

This layer provides decentralized exchange execution, position management, technical-indicator computation, and multi-chain RPC access with automatic node failover.

5. The AI Trading Agent

5.1 AgentRuntime: a bounded thinking loop

The agent operates as a loop: build context, call the LLM, observe whether the model requested tool use, execute the requested tools, feed results back, and repeat until the model produces a final response. The loop is capped at eight rounds (`maxRounds = 8`) to prevent runaway tool-call chains — enough for a full analysis plus at most one or two trades.



The runtime supports both OpenAI function-calling and Anthropic Claude tool-use protocols, and differs in how the system prompt is placed for each (a top-level `system` field for Claude, a `role=system` message at the head of the array for OpenAI). Transient network failures (timeout, connection, DNS, 5xx, 429) trigger a retry after a short backoff; authentication failures (401/403) do not.

5.2 AgentToolRegistry: twelve tools

The agent is granted twelve tools, divided into read-only and write categories.

Read-only tools (no gas, no on-chain state change, no gate check):

Tool	Purpose
<code>get_wallet_address</code>	Resolve the wallet's address on a given chain
<code>get_native_balance</code>	Native-coin balance and USD value
<code>get_token_balance</code>	ERC-20 token balance by contract address
<code>get_token_price</code>	Current price of a major token in USD
<code>get_position</code>	All open positions with profit/loss
<code>get_market_data</code>	K-line technical indicators (RSI/MACD/MA/Bollinger)
<code>get_safety_status</code>	Safety-gate state: circuit breaker, limits, error rate
<code>call_contract_read</code>	Arbitrary read-only contract call (<code>eth_call</code>)

Write tools (consume gas, change on-chain state, mandatory SafetyGate check):

Tool	Purpose
<code>swap_tokens</code>	DEX token swap via router, auto-handles approval
<code>approve_token</code>	ERC-20 spend approval
<code>send_native</code>	Native-coin transfer to an address
<code>call_contract_write</code>	Arbitrary state-changing contract call

Every write tool requires an `operation_desc` string that states the intent; that string is recorded in the audit log.

5.3 AgentMemory: a self-editing memory file

Each wallet has an independent memory file (`agent_memory.json`) that encodes the agent's identity, personality, owner naming, check interval, daily-loss limit, and trading chain. The agent can adjust a bounded set of its own configuration through an `@SET field=value` instruction, with range validation on each field. Wallet addresses are never persisted here; they are resolved live from the signing module.

5.4 System prompt and safety red lines

The assembled prompt injects the agent's identity, the current market state, the safety state, the tool definitions, and the memory. It also embeds an immutable set of safety red lines:

- The agent does not need the private key; all signing happens locally.
 - All write operations must pass the safety gate.
 - Non-whitelisted token buys require explicit user confirmation.
 - No trades during a circuit break (except protective position closes).
 - Never expose the private key, seed phrase, or API keys.
 - Never promise a guaranteed gain; only provide analysis.
 - Every trade decision must cite its analytical basis.
-

6. Safety and Risk Framework

The safety tier is the difference between an assistant and an unaccountable bot. It is organized as three sequential defense lines.

6.1 Defense line one: TradeAuthManager

The AI trading feature cannot even be enabled until the account meets an eligibility threshold: at least USD 200 in major-coin assets, or at least 20,000 R-MAB tokens held. The check runs on a background thread, and a cached result can be read synchronously on the main thread without blocking the UI.

6.2 Defense line two: SafetyGate

SafetyGate is a mandatory five-step validation for every write request:

1. **Circuit breaker.** Three consecutive losses invoke a 30-minute break; a daily error rate above 50% invokes a 60-minute break. During a break, all write operations are rejected except protective position closes.

2. **Daily limit.** Transactions are rejected once the day's total exceeds the configured daily-loss limit.
3. **Per-trade limit.** A single transaction larger than the per-trade cap is rejected.
4. **Whitelist.** By default every token is blacklisted. The agent must present the real target contract address; native currency and the DEX router are exempt. A non-whitelisted token triggers a confirmation dialog with three choices — whitelist and continue, allow this once, or reject. In the absence of any UI context, the request is automatically rejected.
5. **Audit.** Every approved operation is stamped with its `operation_desc` and recorded.

The R-MAB platform token is permanently whitelisted, so its routine operations do not require repeated user confirmation.

6.3 Defense line three: RiskManager

RiskManager governs execution parameters: a maximum position size of 10% of total assets per token, a default stop-loss at -5% and take-profit at +15% of the buy price, a 3% slippage tolerance, and a daily cap of 20 trades. A position monitor continuously compares live prices against stop-loss and take-profit levels and triggers protective sells.

6.4 Cross-chain authorization

Cross-chain transactions carry their own conservative defaults — a USD 100 per-transaction cap and a USD 500 daily cap. A one-time authorization for a given chain-plus-asset-plus-address combination enables subsequent same-chain, same-asset signals to execute silently when the safety-gate conditions are satisfied.

7. Multi-Chain Data and Trading Layer

7.1 Chain support

AI Crypto Wallet supports ten chains: BNB, ETH, MATIC, AVAX, FTM, ARB, OP, BASE, SOL, and TRX. Each EVM chain is configured with its correct DEX router and wrapped native token address, avoiding the misconfiguration that commonly breaks swaps when a single router address is reused across networks.

7.2 Node resilience

RPC access is provided through a primary node with automatic failover to backup nodes. Connection timeouts are kept short so requests fail fast and fall back quickly. Batch token-balance queries reduce request volume, and in the event of an empty or malformed batch

response the system throws to trigger a fallback to individual queries with node switching, rather than silently discarding real tokens.

7.3 Market data and indicators

K-line data is retrieved at 15-minute, 1-hour, 4-hour, and daily periods, and technical indicators (RSI, MACD, moving averages, Bollinger Bands) are computed locally. For scheduled analysis, the agent distinguishes the primary cycle from confirmation cycles to avoid whipsaw entries.

7.4 DApp browser

A built-in Web3 browser lets users interact with decentralized applications and sign transactions with the wallet. The browser maintains a stable domain cache so whitelist verification remains consistent across tab lifecycles, and it rejects DApps that are incompatible with the self-custody model.

8. Platform Token: R-MAB

R-MAB is the platform token of the AI Crypto Wallet ecosystem. Holding it grants ecosystem participant privileges, including the eligibility threshold for AI trading via the TradeAuthManager. It benefits from a permanent whitelist exemption in the safety gate, so authorized operations do not require repeated manual confirmation.

R-MAB is treated as a platform-level token with special privileges within the wallet. Its role is to represent participation in the ecosystem rather than to serve as a revenue-sharing instrument.

9. Privacy and Security Model

- **Key isolation.** Private keys are generated locally and held only in the signing module. The seed phrase is used transiently during creation or import and is never stored persistently where the agent can reach it.
- **Sensitive data hygiene.** API keys are stored masked (first and last four characters). Wallet addresses are resolved live rather than persisted in the memory file.
- **Total asset visibility.** The wallet displays the current wallet's native coin, tokens, and total value immediately, while background computation of aggregate cross-wallet totals runs on a separate executor so it never blocks the UI.
- **Localization.** The interface is fully localized into Chinese, English, Japanese, and German.

10. License and Governance

AI Crypto Wallet is source-available and publicly auditable, licensed under the **Business Source License 1.1 (BSL-1.1)**. It is not an OSI-certified open-source project.

Under BSL-1.1:

- The source code may be viewed and audited by anyone.
- Individuals may download, install, and use the application for personal, non-commercial purposes.
- Commercial use, redistribution, hosting, resale, or provision as a service — in original or modified form — requires a separate written license from the Licensor (Red Devil Team).
- After the Change Date of 2031-08-06, the work converts to the Apache License 2.0.

The source is mirrored at **github.com/openxcn/AI-Crypto-Wallet**. Commercial licensing inquiries should be directed to **aibgsps@gmail.com**.

11. Roadmap

The following capabilities are planned for upcoming releases:

- News and catalyst reporting, with the agent summarizing market-moving headlines.
 - Multi-chain simultaneous monitoring beyond the primary chain.
 - Trade-history export in CSV and JSON.
 - Multiple selectable LLM backends.
 - Strategy backtesting against historical data.
 - Whale-wallet movement monitoring.
 - Cross-chain arbitrage detection.
-

12. Disclaimer

This whitepaper is a technical specification. It is not financial advice, and nothing in it constitutes an offer to sell or a solicitation to buy any financial instrument or token. Cryptocurrency trading is extremely risky; you may lose all of the funds you deploy. Never trade

with money you cannot afford to lose. The AI provides analysis and executes only within the limits the user configures and the safety framework enforces. Use at your own risk.

Copyright (C) 2026 红魔团队 (Red Devil Team). All rights reserved. Unauthorized copying, modification, or distribution is prohibited.