Token Real-time Profiling and On-chain Behavior Analysis API Technical Document

Version: v1.0

Last Updated: 2025-10-15

Author: Internal Risk Control Analysis and On-chain Data Team

# I. System Architecture Overview

## 1.1 Frontend Architecture

Framework: Vue.js + JavaScript

Functions:

• Data visualization display (ECharts / D3.js)

• Real-time monitoring panel and chart rendering

• Dynamic interaction and Token profiling visualization

## 1.2 Backend Architecture

Language & Framework: PHP 7.x

Web Service: Nginx

Database: MySQL + Redis

Cache Mechanism: Redis for high-frequency data caching and request throttling

Storage Structure:

• MySQL stores Token metadata, user configurations, and indexing information

• Redis stores Token real-time snapshots and transaction cache

# II. Security Design

|  |  |  |
| --- | --- | --- |
| Security Module | Technology Implementation | Description |
| Authentication | JWT (JSON Web Token) | Each request requires a valid token for identity validation and access control |
| Transmission Security | SSL / HTTPS | Full-link encryption to prevent man-in-the-middle attacks |
| Data Encryption | AES Symmetric Encryption | Sensitive information (e.g., addresses, API keys) is encrypted for storage |
| Application Protection | WAF (Web Application Firewall) | Prevents common attacks like SQL injection, XSS, CSRF |
| Access Control | Role-based and Token-based access rights | Distinguishes between public API and restricted API access levels |

# III. Core Functionality Description

|  |  |  |
| --- | --- | --- |
| Module | Function | Output Content |
| Token Basic Information | Query contract, symbol, price, etc. | Basic metadata, price, FDV, liquidity |
| Holder Structure Analysis | Analyze the distribution and concentration of token holders | TopN holder ratio, dispersion labels |
| Transaction Behavior Analysis | Analyze on-chain activity and interaction relationships | Transfer density, associated wallet ratio, risk labels |
| Comprehensive Health Assessment | Calculate overall health score | Comprehensive health score (0~100) |
| Trend Profiling | Track holder changes and activity variations | holderChange, uniqueWallets, buy/sell ratio |

# IV. Analysis Process

┌─────────────────────────────┐

│ Token Interaction Analysis │

└─────────────────────────────┘

│

▼

Get Token basic information and holder data

│

▼

Get recent transaction records

│

▼

Calculate holder dispersion index

│

▼

Analyze transaction relationships between major addresses

│

▼

Calculate interaction and concentration score

│

▼

Calculate comprehensive score and output results

5. Core Algorithm Logic (Demonstrative Description)

① Holding Dispersion Score

Objective: Measure whether a token is highly concentrated among a few addresses.

Core Concept:

Compute the concentration parameter by summing the squares of the top 100 holders’ proportions.

Formula:

\text{Dispersion} = 1 - \sum\_{i=1}^{N} (p\_i)^2

Where pᵢ = proportion of holdings for the i-th address.

Interpretation:

• Higher dispersion → lower concentration risk

• Visualization: “Top Holder Distribution Curve”

• Output Tags: High Dispersion / Medium Dispersion / Low Dispersion

② Transfer Correlation Score

Objective: Analyze whether major holders exhibit excessively frequent interactions.

Formula (illustrative):

R = 1 - (0.4×P₁ + 0.3×P₂ + 0.3×Rₜ)

Explanation:

• Lower correlation → more natural distribution → higher decentralization.

③ Comprehensive Token Score

Objective: Combine dispersion and correlation scores to evaluate token decentralization and health.

Formula:

\text{TokenScore} = 100 × (0.5×Dispersion + 0.5×R)

Interpretation:

• Higher score → healthier and more decentralized token.

6. Visualization & Display Guidelines

• Use ECharts or D3.js for dynamic rendering

• Support real-time updates and animated transitions

• Display score labels with gradient indicators

• Provide export options (PNG / PDF / JSON)

7. Data Sources & Synchronization Strategy

• On-chain data aggregation via blockchain APIs

• Market data integration via exchange feeds

• Scheduled synchronization and Redis caching

• Fallback mechanisms for data consistency assurance

8. Performance & Stability

Metric Target

Average Response Time < 300 ms

Concurrency Capacity 10,000+ QPS (scalable horizontally)

Cache Hit Rate > 95%

Data Latency ≤ 1 min (market data) / ≤ 10 min (distribution data)

9. System Diagram (Illustration)

┌────────────────────────┐

│ Token Real-Time Profiling System │

└────────────────────────┘

│

┌───────────────┼────────────────┐

▼ ▼

[On-Chain Data Aggregation] [Market Data Aggregator]

│ │

▼ ▼

MySQL + Redis ←→ Analysis Engine ←→ Visualization Frontend (Vue)

10. Disclaimer

The scores and risk labels described in this document are algorithmic demonstration results for informational purposes only.

They do not constitute investment advice or guarantees.

Actual risk evaluation should incorporate multi-dimensional data and human analysis.