

# Non-Atomic Blockchain Transaction Identification

## 测试报告

2025 年 12 月 21 日

### 1 概述

- 目标：验证非原子套利识别网站在**数据采集、算法计算、API 输出、前端呈现、部署可用性**上的正确性、健壮性与可维护性。
- 方法：黑盒功能测试 + 部署验证，辅以关键路径健壮性和性能抽查，覆盖正常流与异常流。

### 2 测试环境

- 运行：`docker-compose up --build` 启动 db (PostgreSQL 15)、backend (FastAPI)、frontend (React)、nginx。
- 配置：`.env` 含 `DATABASE_URL/POSTGRES_USER/POSTGRES_PASSWORD`；  
前端 `REACT_APP_USE_BACKEND=true` 读取真实数据，否则回退模拟数据。
- 数据：卷 `postgres_data` 持久化；时区统一 UTC。

### 3 测试范围

- **数据采集与入库**：Binance/Etherscan 拉取、TLS/限流重试、去重逻辑 (哈希 + `log_index`)、schema 升级幂等、UTC 时间戳校验。
- **套利计算**：启发式过滤、时间窗匹配、利润/方向计算、清表再写、手续费/滑点参数一致性、空表健壮性。
- **后端 API**：

- 健康: `/api/health`, `/api/db-check`。
- 价格: `/api/price-data` (日期窗口、OHLC 聚合、空数据返回结构)。
- 套利: `/api/arbitrage/statistics`, `/api/arbitrage/behaviors` (分页/排序/过滤)、`/api/arbitrage/opportunities`(时间过滤/利润率过滤/ISO8601)。
- CORS: 本地/代理跨源。
- 前端:
  - Info: 内容完整、外链属性。
  - Navbar 搜索: 高亮、结果数量、跳转、外部点击关闭。
  - Price Dashboard: 后端/模拟切换、K 线缩放与拖拽、对比模式、日期定位、刷新回退提示、表/图 hover 同步。
  - Arbitrage Analysis: 统计卡、分页表格、排序/过滤、时间轴缩放与拖拽、tooltip/图例、离线/空数据提示。
- 部署/代理: Nginx 路由转发、端口映射、服务依赖 (`depends_on` + 健康检查)、卷持久化验证。

4 用例摘要

领域	关键验证	要点/输入	结果
采集/入库	限流/重试、去重、schema 升级、UTC	主/备 Binance API 切换; Etherscan 分块; 哈希 +log_index 防重; 出错回滚; 缺失.env 报错提示	轻微问题
套利计算	过滤与匹配、利润率、方向	300s 窗口; 手续费/滑点参数一致; 利润率百分比转换; 空表/单侧数据健壮性	通过
后端 API	校验与分页、空数据、时间格式	start > end 拒绝; 空数据返回空结构; ISO8601/Z 解析; 分页/排序边界	通过
前端交互	路由/搜索/图表/过滤	K 线缩放拖拽、对比模式、时间轴缩放拖拽、筛选器复位、hover 同步、回退提示	通过
部署/代理	反代路由、健康依赖	/api → backend, / → frontend; 卷持久化; 健康检查后启动	通过

5 主要测试步骤

- **采集脚本**: 断开主 Binance, 验证 fallback; 模拟 429/451, 观察窗口缩减与重试; 重复拉取同一时间段, 确认去重有效。
- **套利计算**: 清空目标表, 运行 compute\_arbitrage.py 与 compute\_opportunities.py, 检查利润率、方向、时间序列是否递减返回。
- **API**: 构造非法日期 (start > end), 非法时间格式, 缺失参数; 分页边界 page=0/page\_size=200; min\_profit\_rate 过滤; 空表时结构完整。
- **Price Dashboard**: 滚轮缩放、拖拽平移、日期选择对齐; 后端断开时回退模拟数据并提示; 对比模式显示 spread 与双曲线。
- **Arbitrage Analysis**: 筛选 (利润率、时间范围)、重置; 时间轴缩放与拖拽; 表格分页与排序一致性; 空数据/加载/错误占位。

- 部署：Nginx 路由转发检查；卷持久化验证（停服重启数据仍在）；CORS 在本地/代理路径下可用。

6 测试结果

- 结论：核心链路（采集 → 计算 → API → 前端展示 → 代理）可用；存在 1 条中等级问题需修复以提升稳定性。

7 项目效果展示

- 前后端打通效果：Price Dashboard（前端）读取后端真实数据，Arbitrage Opportunities 列表分页/过滤正常。
- 两幅截图并排展示：左侧 dashboard.png（仪表盘），右侧 opportunity.png（套利机会表）。

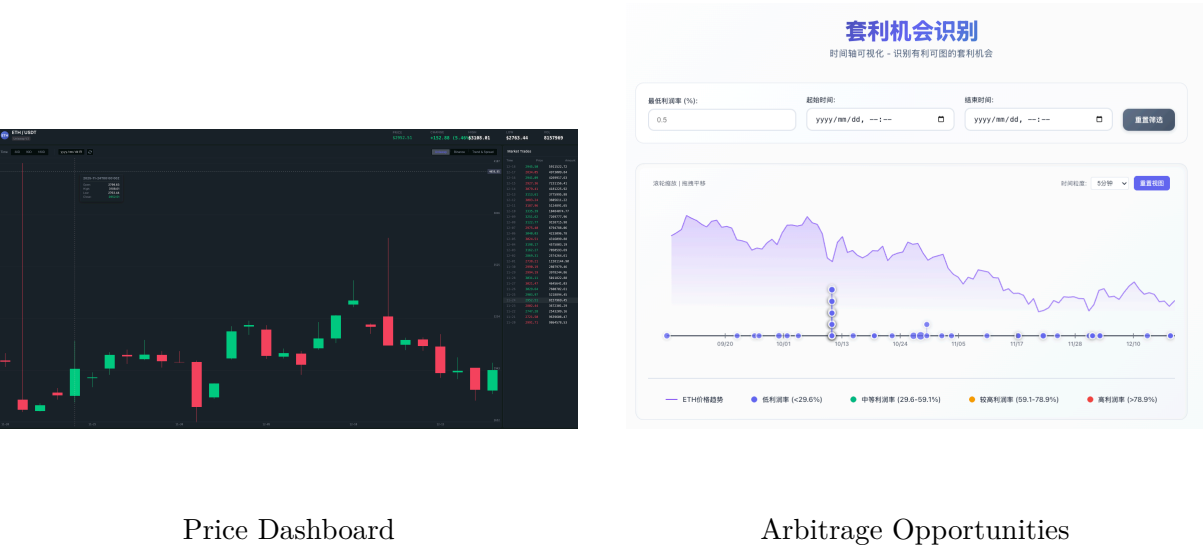


图 1: 前后端接口打通效果：仪表盘与套利机会并排展示

关键发现

- 采集限流场景缺少**最大退避/总重试上限**，长时间 429/451 会持续尝试；建议增加上限并输出报警（中）。
- 由于爬取数据 API 并非会员，由于爬取速度过快而被 API 禁止访问一段时间稳定可复现。
- 其余环节未复现阻断性缺陷。

## 8 改进与后续

- 采集：增加最大退避和失败告警；分段重试上限。
- CI：加入轻量 API 合约测试（健康、价格、套利列表）防回归。
- 前端：离线模式提示，明确数据来源（后端/模拟）。

## 9 附录：覆盖清单

- 后端接口：`/api/health`, `/api/db-check`, `/api/price-data`, `/api/arbitrage/statistics`, `/api/arbitrage/behaviors`, `/api/arbitrage/opportunities`
- 主要脚本：`fetch_data.py`, `compute_arbitrage.py`, `compute_opportunities.py`
- 前端页面：`/info`, `/price-dashboard`, `/arbitrage-analysis`