Hyperliquid Trade Ledger API (with optional Builder-only mode)

Objective

Build a dockerized service that provides:

- **Trade history**
- **Position history** (timeline or reconstructed state)
- **Cumulative PnL** over defined time — absolute and relative
- With an **optional builder-only filter** suitable for Insilico users or competitions

---

Mandatory Capabilities

- **All-trades mode (default):** returns complete ledger for wallet/coin/time range
- **Builder-only mode (optional flag):** returns only trades attributed to a specific builder and marks mixed activity as tainted

---

Builder-only Rules

- Define config: `TARGET_BUILDER=0x...`
- A trade is builder-attributed if the fill contains builder attribution matching `TARGET_BUILDER`
  - If public API doesn't expose builder attribution, teams can implement builder-only as "**best effort**" and must clearly document limitations
- **Taint rule:** if non-builder activity affects the same position lifecycle in the requested window, set `tainted=true` and exclude from builder-only aggregates/leaderboard outputs
- **Position lifecycle:** starts when `netSize` moves from 0 → non-zero, ends when it returns to 0

Required API Endpoints

All time parameters (`fromMs`, `toMs`) are **optional**. If omitted, returns all-time data.

1. Trades

```
GET /v1/trades?user=&coin=&fromMs=&toMs=&builderOnly=false
```

Returns normalized fills:

- `timeMs`, `coin`, `side`, `px`, `sz`, `fee`, `closedPnl`, `builder` (optional)

2. Position History

```
GET /v1/positions/history?user=&coin=&fromMs=&toMs=&builderOnly=false
```

Returns time-ordered position states sufficient to plot history:

- Minimum: `timeMs`, `netSize`, `avgEntryPx`
- Include `tainted` when `builderOnly=true`
- **Note:** Average cost method is acceptable for `avgEntryPx`

3. PnL

```
GET /v1/pnl?user=&coin=&fromMs=&toMs=&builderOnly=false
```

Returns:

- `realizedPnl` — absolute USD value (sum of `closedPnl` from fills)
- `returnPct` — relative return (see calculation below)
- `feesPaid`
- `tradeCount`
- `tainted` (when `builderOnly=true`)

Relative PnL calculation (capped normalization):

```
effectiveCapital = min(equityAtFromMs, maxStartCapital)
returnPct = realizedPnl / effectiveCapital * 100
```

- If `maxStartCapital` not specified, use actual `equityAtFromMs`
- This ensures fair comparison (large accounts don't have unfair advantage)

4. Leaderboard

```
GET /v1/leaderboard?coin=&fromMs=&toMs=&metric=volume|pnl|returnPct&builderOnly=true&maxStartCapital=1000
```

Returns ranked list:

- `rank`, `user`, `metricValue`, `tradeCount`, `tainted`

Metrics:

- `volume` — total notional traded
- `pnl` — absolute realized PnL
- `returnPct` — relative return (using capped normalization)

Optional (Bonus) Endpoints / Features

- **Deposit tracking** (for fair competition filtering):

```
GET /v1/deposits?user=&fromMs=&toMs=
Returns: totalDeposits, depositCount, deposits[]
```

Enables filtering users who reloaded capital during competition window

- **Risk fields:** `liqPx`, `marginUsed` on position endpoints
- **Partial closes and position flips** handled correctly (long → short transitions)
- **Multi-coin aggregation** (portfolio-level PnL/leaderboard)

Data Ingestion Requirements

- Must work using **public HL APIs** (Info/WS) as the source
- Must implement a **datasource abstraction** so it can be swapped to **Insilico-HL / HyperServe** later with minimal changes

---

Deliverables

- **Dockerfile** / docker-compose or similar (one-command run)
- **README** with:
  - How to run (single command)
  - Environment variables
  - What's supported in `builderOnly` mode (and how builder attribution is obtained)
  - Any limitations or assumptions

What We Provide

- List of **5 test wallet addresses**
- Expected trade counts and PnL values for validation

Judging

Criteria	**Weight**
Correctness	50%
Completeness	20%
Builder-only handling	20%
Demo clarity	10%