Market Maker Bot Design Summary

1. Architecture & Data Flow

System Components

1. **Execution Engine**
    - Generates trader order flow (random BUY/SELL orders).
    - Routes orders to each market's Market Maker.
    - Runs a 200-tick simulation loop.
2. **Market Maker Bot**
    - Computes mid, bid, ask prices.
    - Adjusts spreads dynamically based on risk.
    - Handles inventory, imbalance, and adaptive behaviour.
3. **Market State Store**
    - Tracks inventory, PnL, fills, exposure, notional traded.
    - Maintains risk limits.
4. **Matching Logic**
    - Checks if the trader's price crosses bot's bid/ask.
    - Executes fills and updates state.
5. **Logger**
    - Records time-series trace (trace.json).
    - Outputs final metrics (simulation_report.csv).

Data Flow

1. Execution Engine → generates trader orders
2. Orders → Market Maker (to produce quotes)
3. Market Maker → returns bid/ask quotes
4. Matching Logic → checks for fills
5. Fills → update Market State
6. State changes → logged to JSON + CSV

2. How Mid, Spread, and Size Are Determined for a Binary Event Contract

A binary event contract pays **1 if an event happens, 0 if not**, so its fair value is a **probability between 0 and 1**.

Mid-Price

Base mid represents the bot's estimate of the probability:

mid_shaded = mid - skew

- **Skewing** pushes mid **against inventory** to flatten exposure:
    - Long inventory → mid moves **down**
    - Short inventory → mid moves **up**

Spread

Spread widens based on:

- **Order Flow Imbalance** (buying vs selling pressure)
- **Inventory Risk** (size of current position)

Calculation:

spread_factor = 1 + abs(imbalance)/10 + abs(inventory)*inventory_skew
spread = clip(base_spread * spread_factor, min_spread, max_spread)

Quote Size

Size shrinks as inventory grows:

size = max(1, 10 / (1 + 0.01 * abs(inventory)))

This prevents runaway risk in one direction.

3. Handling Imbalanced Order Flow

The bot tracks a rolling imbalance window of recent trades:

imbalance = sum(last_20_flows)

Reaction to One-Sided Flow

- **Spreads widen sharply** to avoid adverse selection.
- **Mid adjusts slowly** in the direction of persistent buying/selling.
- **Quote skewing** discourages adding more inventory in the same direction.

Example:
Heavy buying →
✓ mid drifts up
✓ ask widens
✓ bid drops slightly
✓ size reduces

This stabilizes the bot under pressure.

4. Risk Management Logic

The bot uses layered risk controls:

1. Inventory-Based Mid Skewing

Pushes quotes toward neutralizing risk.

2. Spread Widening

More risk → wider spreads.

3. Size Reduction

Inventory ↑ → quote size ↓.

4. Defensive Repricing (80% of limit)

```
if abs(inventory) > 0.8 * limit:
   mid -= 0.001 * sign(inventory)
```

Acts as an emergency "unwind mode."

5. Hard Limits

- **Inventory limit:** 200
- **Exposure limit:** $10,000
  Orders are rejected if limits would be breached.

5. Stability After Disconnects or Data Gaps

The design is resilient to interruptions:

- **State stored centrally** (inventory, mid, spread, PnL).
- **Imbalance window bounded** (max 20 entries) → prevents stale signals.
- **Quotes are always recomputed from state**, so the bot can resume at any tick.
- **Spread floor and ceiling** prevents extreme reactions on restart.
- **Mean reversion** nudges mid slowly back to neutral if no new data arrives.

Overall:
✓ No dependence on unbounded history
✓ Controlled adaptation
✓ Self-stabilizing quoting logic

High-Level System Architecture
Components:

- Execution Engine – Generates order flow & routes orders.
- Market Maker Bot – Computes mid, quotes bid/ask, manages risk.
- Market State Store – Tracks inventory, pnl, fills, exposure.
- Matching Logic – Matches trader orders vs bot quotes.
- Logger – Records JSON trace + CSV metrics.

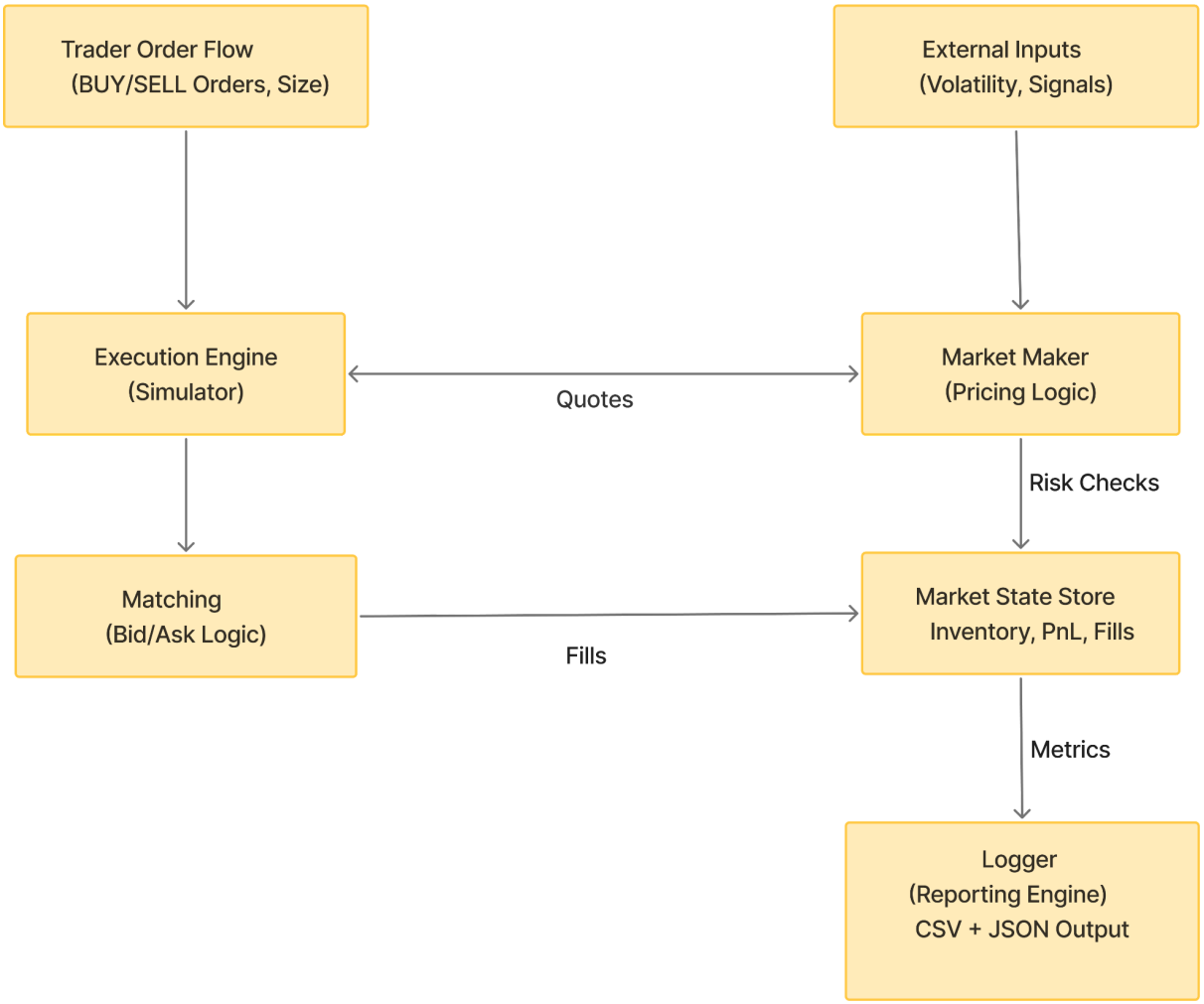
Oluwatunmise Olaoluwa

Flow:

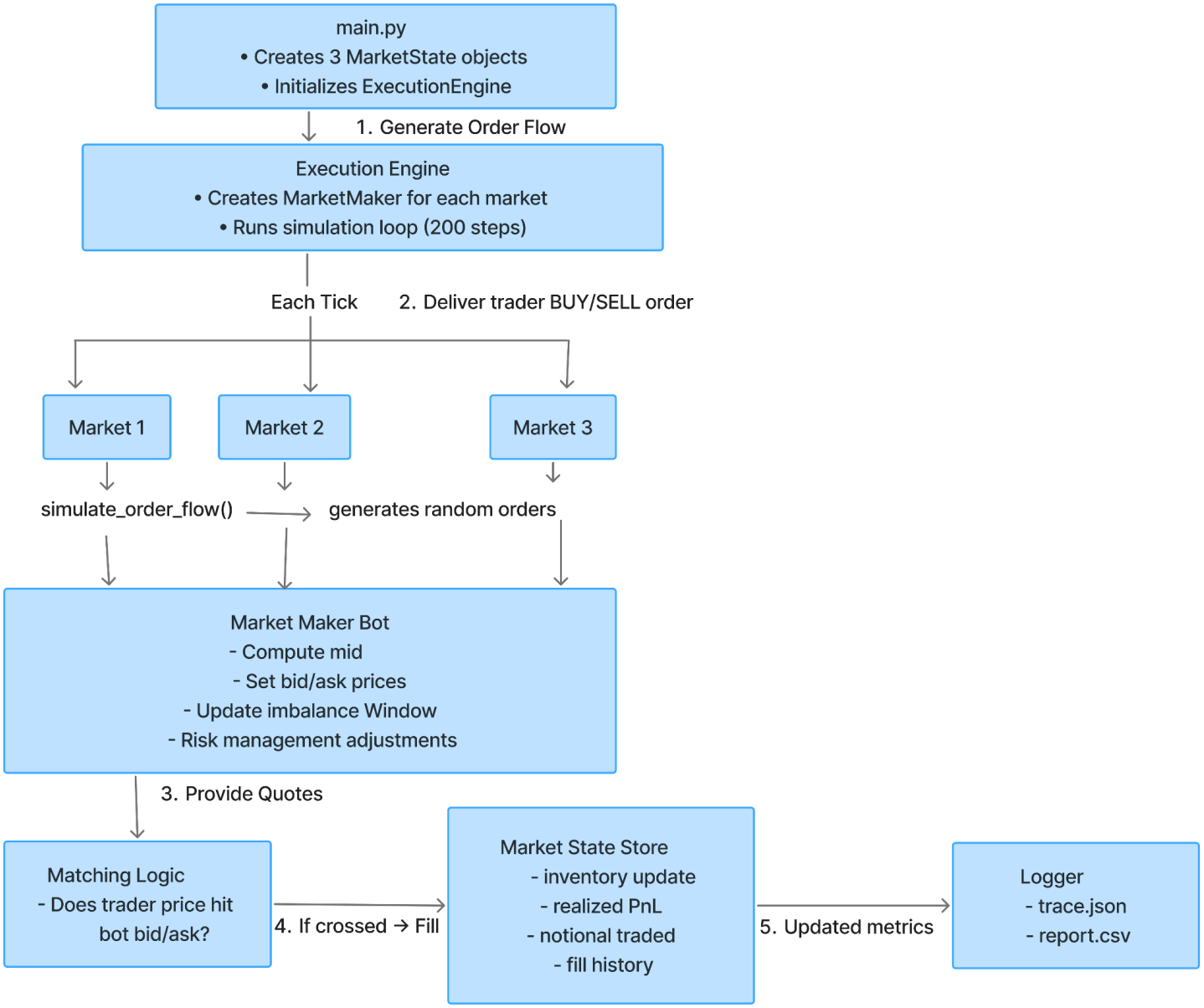
1. Trader Flow → Execution Engine
2. Execution Engine → Market Maker (order)
3. Market Maker → Execution Engine (bid/ask quote)
4. Execution Engine → Matching Logic
5. Matching Logic → Market State (fills)
6. Market State → Logger (pnl, inventory, fills)

Oluwatunmise Olaoluwa

System Architecture Diagram



Data Flow Diagram



Final Summary

This Market Maker bot is a **probability-based quoting engine** designed for **binary prediction markets**. It adapts to order flow, protects itself from inventory risk, and ensures stability across market regimes. The system architecture, data flow, and risk-aware quote generation create a robust simulation of real-world market making behaviour.