

## **1. Introduction**

The purpose of this simulator is to replicate the **core mechanics of a financial exchange** in a simplified, deterministic environment. While it does not incorporate all real-world complexities such as high-frequency latency, market depth beyond a few levels, or hidden orders, it captures the **essential elements of order matching, market dynamics, and agent interactions**. This simulator is designed both for educational purposes and as a foundation for future extensions, such as reinforcement learning agents interacting with the market.

---

## **2. Core Market Mechanics**

### **Order Types and Submission**

The simulator allows agents to submit **buy and sell limit orders** with a defined quantity and price. Each order is assigned a **timestamp** upon submission, ensuring **time priority** is tracked, as is standard in most modern exchanges.

### **Order Matching and FIFO**

Orders are matched based on **price–time priority**:

- **Price priority:** Buy orders with higher prices are matched before lower prices; sell orders with lower prices are matched before higher prices.
- **Time priority (FIFO):** Within the same price level, earlier orders are executed before later ones.

This faithfully mimics real exchange behaviour, where **first-in orders at the best price are filled first**, ensuring fairness among market participants.

### **Trade Execution Logic**

A trade is executed only when the **best bid price is greater than or equal to the best ask price**. Partial fills are handled when order quantities do not exactly match, and fully executed orders are removed from the order book. All executed trades are **logged with price, quantity, aggressor, and timestamp**, creating a **trade tape** similar to those used by real exchanges.

---

## **3. Market Environment and Agents**

### **Agent-Environment Interaction**

The simulator models a **discrete-event market environment**:

## Title: How My Simulator Mimics a Real Exchange

- Agents observe the **current market state** (best bid, best ask) and submit orders.
- The environment handles all **order routing, book updates, and trade execution**.
- Agents never directly access the order book, ensuring **modular separation** similar to real-world API constraints.

### Agent Types

1. **Random Trader:** Submits orders randomly, simulating noise in the market.
2. **Market Taker:** Places aggressive orders that cross the spread.
3. **Market Maker:** Maintains both buy and sell quotes to provide liquidity.
4. **Noise Trader:** Generates Poisson-distributed arrivals, simulating stochastic market activity.

This variety captures different market behaviours and creates a **dynamic order book**, reflecting liquidity formation and spread fluctuations observed in real markets.

---

### 4. Deterministic Simulation

The simulator uses a **fixed random seed** and a **discrete event clock**, allowing **deterministic replay**. This means that:

- Submitting the same sequence of orders results in identical trades.
  - This property is crucial for debugging and for future **reinforcement learning experiments**, where reproducible state-action sequences are required.
- 

### 5. Market Calibration Parameters

Although minimal for Week 1, several **calibration knobs** have been included:

- **Tick size:** The smallest price increment.
- **Lot size:** Minimum tradable quantity.
- **Spread floor:** Ensures the order book maintains a minimum spread.
- **Order arrival rate:** Governs stochastic order submission.

These parameters are exposed to allow **future tuning** and experimentation, even though they are not actively adjusted in this phase.

---

### 6. Trade Logging and Visualization

## Title: How My Simulator Mimics a Real Exchange

Executed trades are recorded in a **trade logger**, capturing the price, quantity, order IDs, and timestamps. While visualization is minimal in this version, the logger supports **depth charts and market snapshots**, which can be used for:

- **Top-of-book visualization (L1)**
- **Depth-of-market curves (L2)**
- Analysis of **spread convergence** and liquidity impact

This mirrors real exchange infrastructure, where trade tapes and order book snapshots are fundamental to monitoring and research.

---

## 7. Limitations and Simplifications

While the simulator reproduces many core exchange functions, it does **not fully replicate real-world markets**:

1. No latency, queuing delays, or network effects.
2. Limited order types (no stop orders, iceberg orders, or hidden liquidity).
3. Simplified agent behaviours; no strategic learning is yet implemented.
4. The book only tracks discrete prices (tick size) and small depth levels.

Despite these simplifications, the simulator provides a **faithful educational model of market mechanics**, suitable for algorithmic strategy testing and conceptual understanding.

---

## 8. Conclusion

This Week-1 simulator captures the **essential structure and dynamics of an electronic exchange**, including:

- Agent interactions with a market environment
- Order matching using price-time priority
- Trade logging and replayability
- Exposure of key calibration parameters

By modularly separating **agents, environment, order book, and logger**, the simulator is **extensible, deterministic**, and ready for further development, including the integration of **reinforcement learning agents** or more sophisticated trading strategies.