

Market Mechanics and Matching Engine Design

Modern electronic markets rely on continuous double-auction mechanisms to match buyers and sellers efficiently. In this model, participants submit buy (bid) and sell (ask) orders that specify a price and quantity, or in the case of market orders, only a quantity. The exchange maintains an order book that aggregates outstanding orders and continuously matches compatible bids and asks. The primary objectives of such a system are price discovery, fairness, and low-latency execution.

Order Types and Priority Rules

The market supports two fundamental order types: limit orders and market orders. A limit order specifies a price constraint and remains in the order book until it is fully executed or canceled. A market order, in contrast, executes immediately against the best available prices and consumes existing liquidity. Market orders never rest in the book; any unfilled quantity is discarded once available liquidity is exhausted.

Matching follows a strict **price–time priority**. Orders with better prices are matched first, and among orders at the same price level, earlier submissions have precedence. This rule is central to fairness and predictability, ensuring that participants are rewarded for both competitive pricing and early placement.

Order Book Structure

The order book is organized into two sides: bids and asks. Each side is indexed by price levels, and each price level maintains a queue of orders in arrival order. This structure enables efficient matching while preserving time priority. Best bid and best ask prices are always accessible, allowing the matching engine to determine whether incoming orders cross the spread.

To maintain performance, price levels are kept in sorted order—descending for bids and ascending for asks—so that the top of each list represents the best available price. Within each price level, a first-in-first-out queue ensures correct execution sequencing.

Matching and Execution

When a new limit order is submitted, it is first placed into the order book and then checked for possible matches. If the order's price crosses the opposing best price, trades occur immediately. Execution proceeds until either the incoming order is fully filled or no further matching prices are available. Partial fills are handled naturally by decrementing remaining quantities and removing orders only when their quantity reaches zero.

Market orders bypass the book insertion step entirely. They iteratively match against the best available prices on the opposite side until filled or until the book is empty. This approach mirrors real exchange behavior and ensures deterministic outcomes.

Order Management

In addition to submission, the system supports order cancellation and modification. Cancellation removes an order from both its price queue and the global order registry, ensuring no stale references remain. Modification updates the remaining quantity of an order while preserving its original time priority, a common design choice in real-world markets. Reducing a quantity does not affect priority, while setting it to zero is treated as a cancellation.

Design Trade-offs

The design prioritizes clarity, correctness, and deterministic behavior over extreme micro-optimizations. Data structures such as queues and sorted price lists strike a balance between execution speed and maintainability. While more advanced engines may rely on specialized trees or hardware-accelerated components, this design closely reflects the logical structure used by production exchanges and is well-suited for simulation, research, and backtesting.

Conclusion

This matching engine captures the essential mechanics of an electronic market: price–time priority, continuous matching, and robust order management. By adhering to well-understood market principles and using simple, reliable data structures, the system provides a faithful and extensible representation of real exchange behavior while remaining accessible for experimentation and analysis.