Architecture diagram: (OMS → exchange → tracker pipeline)

1. Strategy logic creates a signal → constructs an Order.
2. Order is passed to OMS, which validates and acknowledges.
3. OMS sends the order to the Limit Order Book, which tries to match it.
4. On any fill, the Order Book emits execution reports.
5. Each report is passed to the Position Tracker, which updates state and logs.

Key metrics and insights:

Strategy	Total Return	Max Drawdown	Sharpe Ratio	# of Trades
Trend Following	+4.78%	-22.18%	-131.22	161
Mean Reversion	-828.39%	-17.18%	-11.10	333
Arbitrage	+213.14%	-22.98%	-21.40	682

With this table (as seen on simulation.ipynb), we can see that arbitrage had the highest returns despite a sizable drawdown. Clearly, the mean reversion strategy was the poorest performer. An explanation for this was the final P&L being extremely high due to large short exposure with AAPL. Trend following had a mild performance but a poor Sharpe ratio, which indicates a high volatility for little return. Thus, it likely suffered from noisy signals or lack of strong trends during the test period.

Challenges & next-step ideas (e.g., iceberg orders, live WebSocket feed, RL order routing):

The current trading simulator provides a solid foundation but reveals key challenges such as inadequate risk management, unrealistic execution modeling, and limited market microstructure representation. Strategies like Mean Reversion suffer from unchecked position sizing and lack of

stop-loss logic, while the order execution engine lacks features like slippage, latency, and iceberg orders. To enhance realism and research capability, next steps could include simulating iceberg and hidden orders, integrating live data feeds via WebSockets, and developing a reinforcement learning agent for intelligent order routing. Other promising directions include upgrading to intraday or tick-level simulations, building a meta-strategy allocator based on market regimes, and creating an interactive analytics dashboard to visualize performance and risk metrics in real time.