

AI Momentum Trading Architecture (IBKR TWS Integration) Last Updated: 2025-10-24

This document outlines the upgraded architecture and logic flow for the AI Momentum Trading Bot, designed to integrate with Interactive Brokers (IBKR) Trader Workstation (TWS) via API. It includes a modern data pipeline, predictive engine upgrades (technical + sentiment), slippage-feedback loops, and a timeframe-weighted outcome scoring system (QOS).

---

1. System Overview - Data Ingress: Real-time market, order book, and sentiment data ingestion - Feature Store: Engineered technical/microstructure/sentiment features - Models: Fast lane (instant inference) and slow lane (policy RL) models - QOS Scorer: Timeframe-weighted signal aggregator - Order Management: Risk checks, execution logic, trailing stops - Feedback Loop: Slippage analysis and model adaptation - Monitoring/UI: Visualization and audit layer integrated with the GUI

---

2. Data Pipeline Sources: IBKR, News/Sentiment, Fundamentals. Flow: IBKR Stream → Preprocessor → Feature Store → Model Inference → OMS/Execution Feedback. Storage: Parquet + DuckDB; Redis for hot features. Latency: Ingest  $\leq$ 100ms, Inference  $\leq$ 50ms, Order  $\leq$ 50ms.

---

3. Feature Engineering - Technical Indicators (MACD, RSI, VWAP, ADX, ROC, etc.) - Microstructure: Spread, Imbalance, Trade Aggression, Volume Bursts - Sentiment: News embedding, headline polarity, event tagging - Regime Context: Volatility percentile, market breadth, trend classifier

---

4. Model Architecture - Fast Lane: Gradient Boosted Trees / MLPs for instant prediction - Slow Lane: Reinforcement Learning (PPO/DDPG) for timing & scaling - Hybrid Nets: Embedding technical indicators in neural architectures

---

5. Timeframe-Weighted QOS Scorer  $QOS = \sum(w_{tf} * S_{tf})$  Dynamic weights by regime, coherence, and performance. Entry if  $QOS > 0.35$ ; exit if  $\Delta QOS < -\theta_{decay}$ .

---

6. Slippage Detection & Feedback Compute arrival\_price, fill\_price, shortfall\_bps, spread\_bps, latency. Adaptive order logic: adjust order type, offset, slicing. Feedback loop updates model features & order rules.

---

7. Order & Risk Management - PDT, margin, exposure, and loss caps - Volatility-scaled sizing - Trailing stops and cool-downs after losses

---

8. Training & Retraining - Backtest: vectorbt / Backtrader - Drift Detection: PSI / KS tests - Retraining cadence: Weekly (fast), Monthly (policy) - Metrics: hit-rate, Sharpe, expectancy, slippage cost

---

9. Monitoring & UI - Panels for Model, Execution, and Risk - Alerts for drift, slippage anomalies, regime flips

---

10. Recommended Stack - IBKR API (ib\_insync, FastAPI) - ML: LightGBM, PyTorch, ONNX Runtime - NLP: FinBERT, DistilBERT - Storage: DuckDB, Redis - Workflow: Prefect / Celery - Visualization: Grafana, GUI overlays

---

11. API Schemas /api/signal → receive model predictions /api/feedback → execution and slippage data

---

12. Expected Benefits Higher precision, lower cost, adaptive automation, risk-aware trading, explainable AI.

---