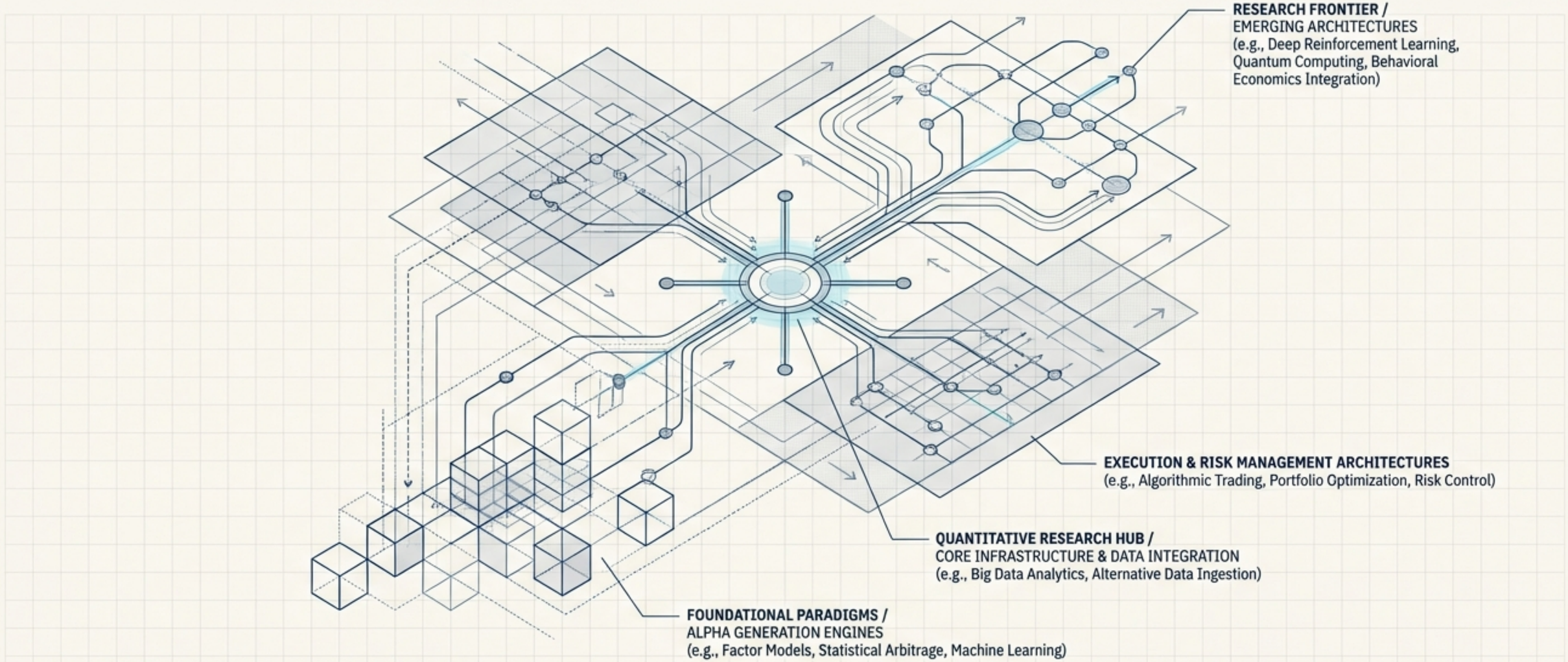
ARCHITECTURES OF QUANTITATIVE STRATEGY

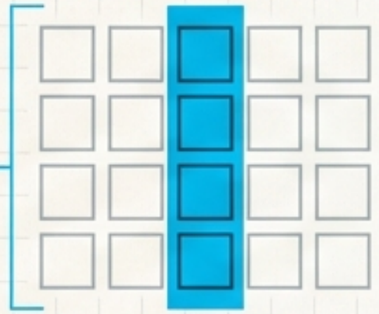
A Technical Blueprint from Foundational Paradigms to the Research Frontier



THE FOUNDATIONAL CHOICE: HOW WE SIMULATE TIME

Backtesting applies a trading strategy's rules to historical data to estimate performance before deploying capital. The architecture of the backtester fundamentally constrains the types of strategies that can be realistically evaluated. The core choice is between two paradigms:

VECTOR-BASED BACKTESTING

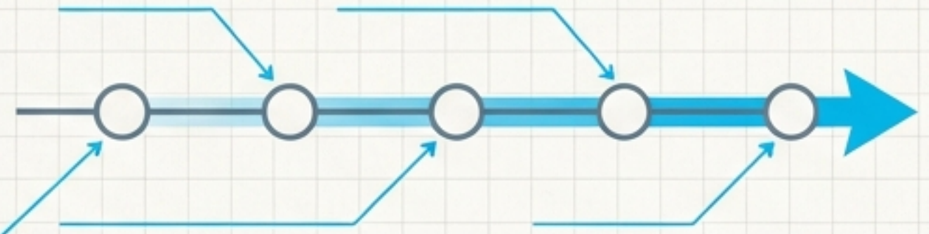


Concept: Processes entire data series (e.g., all daily closes for 500 stocks) in a single, batch operation.

Mechanism: Computes indicators, signals, and rebalances in vectorized passes, typically at bar close or the next bar open. Utilizes optimized libraries like NumPy and pandas.

Analogy: A spreadsheet calculation over an entire column at once.

EVENT-BASED BACKTESTING



Concept: Simulates a live trading environment by processing discrete market events one by one, in chronological order.

Mechanism: An event loop sequentially handles events (ticks, bar closes), with strategy, portfolio, and execution modules reacting to each one.

Analogy: A real-time simulation that processes information exactly as it would arrive.

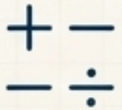
PROTOTYPE 1: THE VECTORIZED BACKTESTER FOR RAPID PROTOTYPING

A high-speed engine designed for strategies that rebalance on fixed time intervals (e.g., daily, weekly) using bar-level data (OHLCV). It leverages parallel processing (SIMD/BLAS) for maximum computational efficiency.

HOW IT WORKS: A 5-STEP WORKFLOW

1. Data Preprocessing: Load adjusted OHLCV data into a pandas DataFrame.

2. Indicator Calculation: Use vectorized functions (`.rolling()`, `.ewm()`) to compute indicators like moving averages or RSI across all assets in one pass.



3. Signal Construction: Compare indicator columns to generate a signal matrix (+1 for long, -1 for short, 0 for flat) for every asset at every timestamp.

4. Portfolio Rebalancing: Multiply signals by a position-sizing rule to get target positions. Fills are simulated at the next bar's open or close price.

5. P&L Calculation: Compute returns and aggregate P&L. Apply a uniform transaction cost assumption.

KEY PARAMETERS FOR TUNING & VALIDATION

- Indicator look-back windows (e.g., 20 vs. 50-day moving average).
- Signal generation thresholds.
- Rebalancing frequency (daily, weekly, monthly).
- Position sizing rules (e.g., fixed-dollar, volatility-scaling).

IDEAL USE CASES

- High-level factor research.
- Cross-sectional momentum studies.
- Rapid iteration over parameter grids.

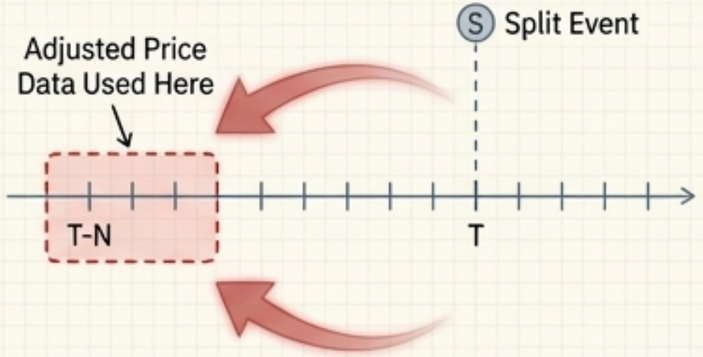
THE CRITICAL FLAW: LOOK-AHEAD BIAS

Look-ahead bias occurs when a backtest inadvertently uses information that would not have been available at the time of the trade. This structural flaw leads to unrealistically positive results, as the simulation has "seen the future."

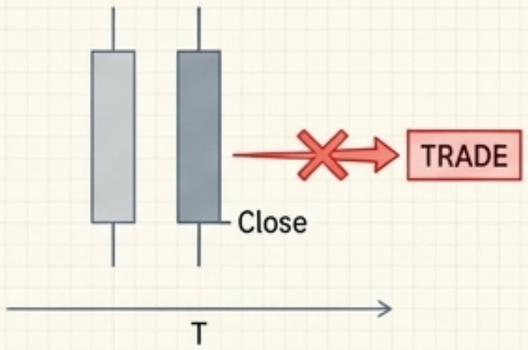
> *"Developers have reported wasting months of work on promising strategies, only to discover a subtle lookahead bias introduced by a single line of vectorized code."*

COMMON WAYS LOOK-AHEAD BIAS IS INTRODUCED

Using Adjusted Prices



Incorrect Bar Offsets



In-Sample Parameter Optimization

Using Adjusted Prices

Price data adjusted for future splits or dividends implicitly contains future information. The backtest "knows" about a stock split before it happens.

Incorrect Bar Offsets

Generating a signal on bar T's closing price and assuming execution at that same close price. In reality, the closing price is only known after the bar is complete.

In-Sample Parameter Optimization

Using the entire dataset to find the best parameters (e.g., the optimal moving average window) and then applying them to the start of the same dataset.

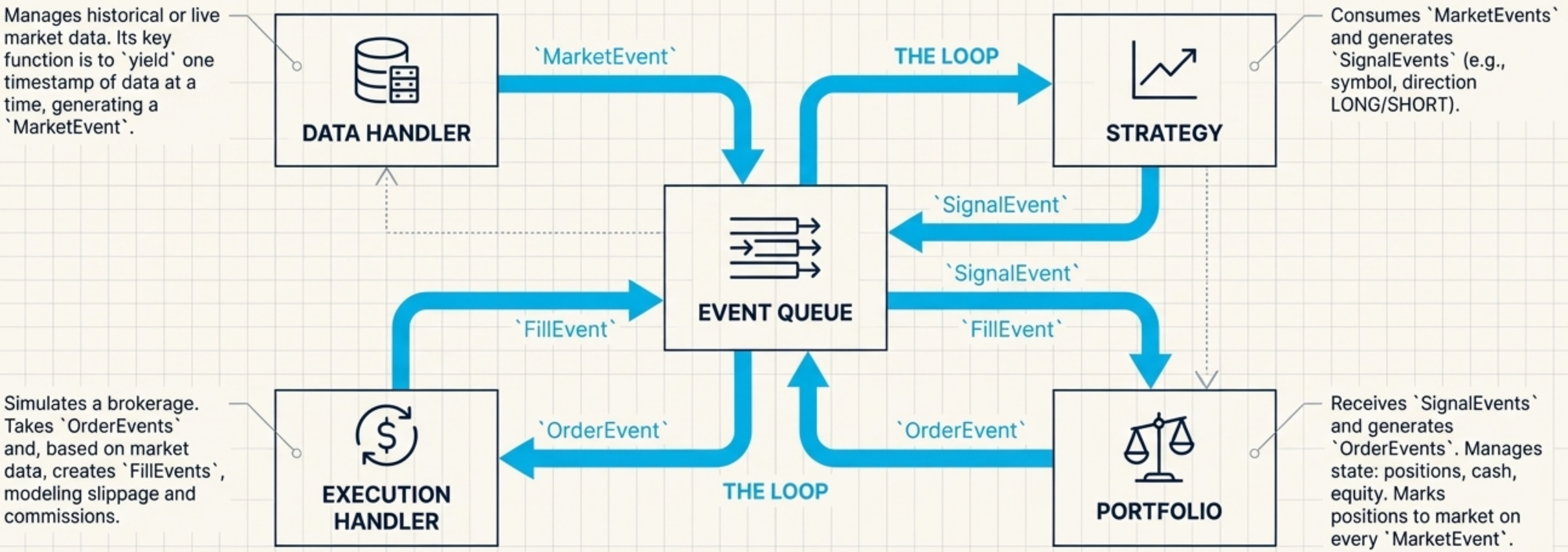
THE ARCHITECTURAL SOLUTION

The definitive fix is not just programmer discipline, but an architecture where look-ahead bias is structurally impossible. This requires a sequential simulation that mimics the chronological flow of information, leading directly to the event-driven paradigm.

PROTOTYPE 2: THE CLASSIC EVENT-DRIVEN ENGINE

An object-oriented system that processes events chronologically via an event queue. This design enforces temporal discipline, making look-ahead bias impossible. It provides higher fidelity at the cost of increased complexity and slower execution speed.

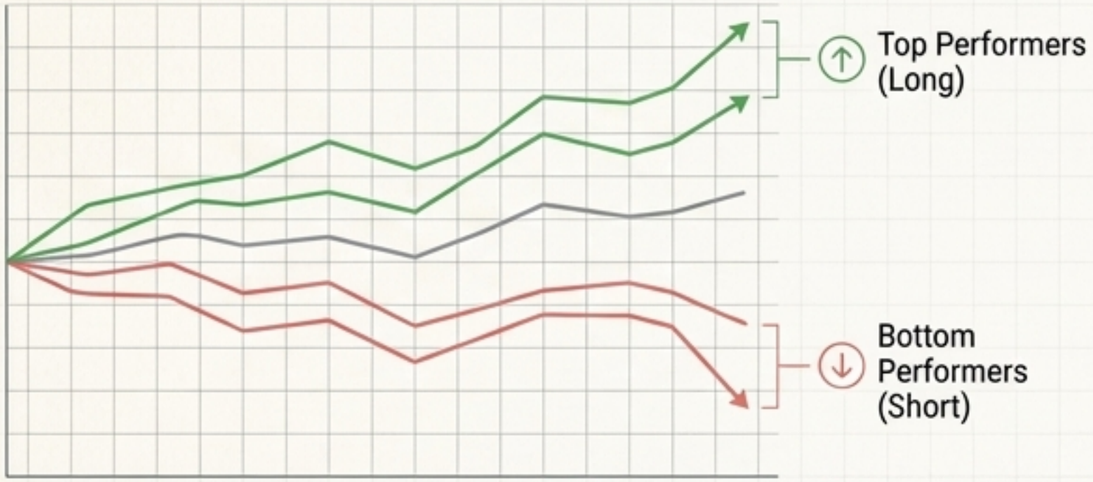
SYSTEM ARCHITECTURE: THE CORE COMPONENTS



EVENT-DRIVEN IN ACTION: CLASSIC BAR-LEVEL STRATEGIES

The modular event-driven engine can implement any strategy that operates on bar data. Below are two canonical examples.

STRATEGY ARCHETYPE 1: CROSS-SECTIONAL MOMENTUM



Theoretical Foundation
Assets that have performed well in the recent past tend to continue performing well. This strategy ranks a universe of assets and goes long the top performers and short the bottom performers.

$$\text{Momentum}(t) = \frac{P(t)}{P(t-N)} - 1$$

Key Parameters
Lookback period (N), rebalance frequency, number of assets in long/short legs.

STRATEGY ARCHETYPE 2: Z-SCORE MEAN REVERSION

Theoretical Foundation
Asset prices have a statistical tendency to revert to their long-term average. A Z-score quantifies how far a price has deviated from its rolling mean in terms of standard deviations.

$$\text{Z-score}(t) = \frac{\text{Price}(t) - \text{SMA}(t, N)}{\text{StdDev}(t, N)}$$

Key Parameters
Lookback period (N) for SMA and StdDev, Z-score thresholds for entry and exit.

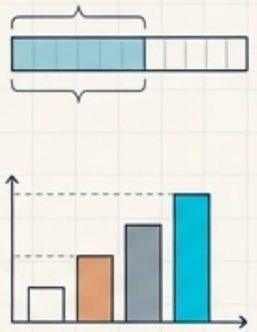
TUNING AND VALIDATING BAR-LEVEL PROTOTYPES

THE GOAL OF VALIDATION: To assess a strategy's performance and robustness across different market conditions and parameter settings. This involves systematic testing of key inputs.

KEY PARAMETERS TO TWEAK AND VALIDATE

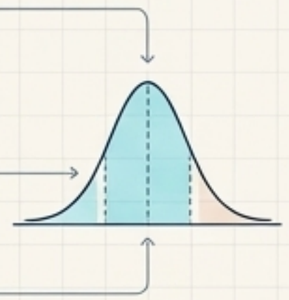
For Momentum:

- **Lookback Window**: How far back to measure performance (e.g., 3, 6, 12 months)?
- **Holding Period**: How long to hold the portfolio before re-ranking?
- **Quantile Size**: Long top 10% and short bottom 10%, or another split?

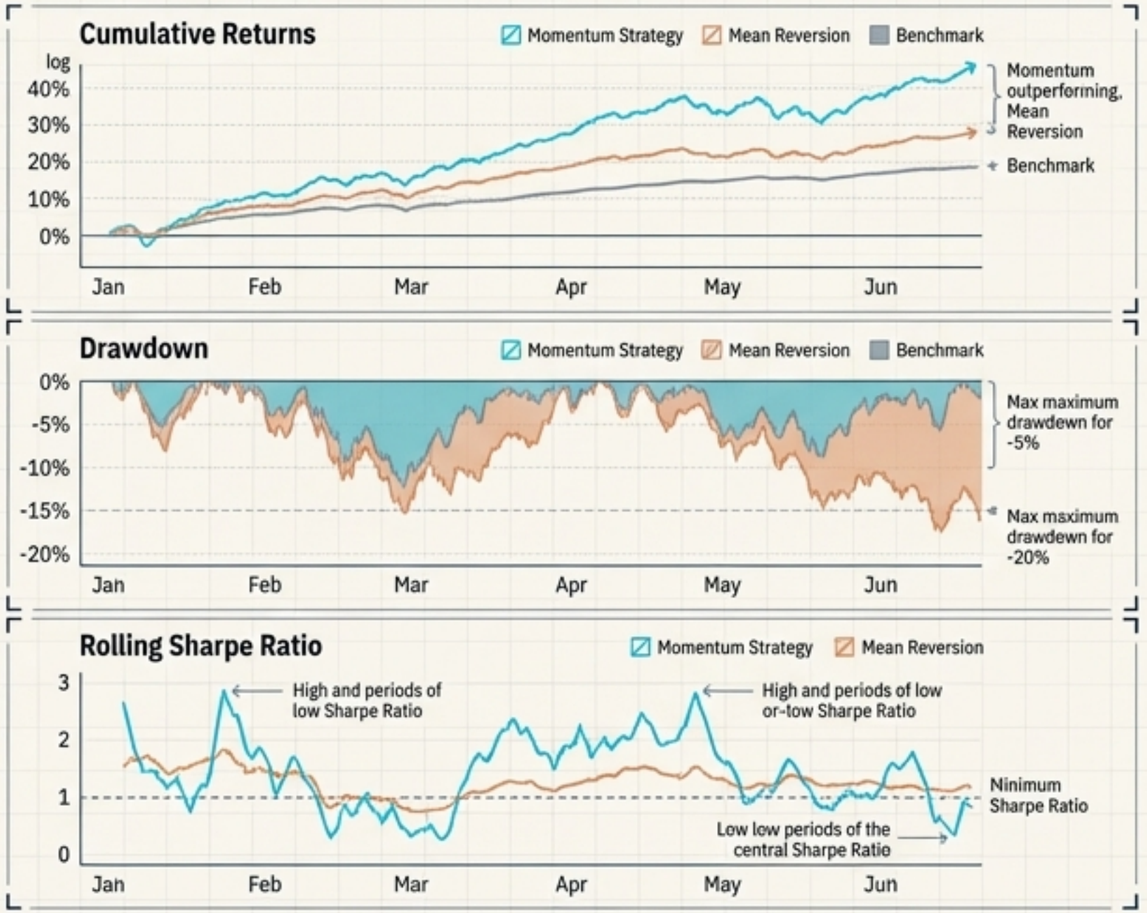


For Mean Reversion:

- **Lookback Window**: The period for calculating the rolling mean and standard deviation.
- **Z-Score Entry Threshold**: At what level of deviation to initiate a trade (e.g., Z < -2.0)?
- **Z-Score Exit Threshold**: When to exit the trade (e.g., when Z crosses 0)?



CORE PERFORMANCE METRICS (EXAMPLE DASHBOARD)

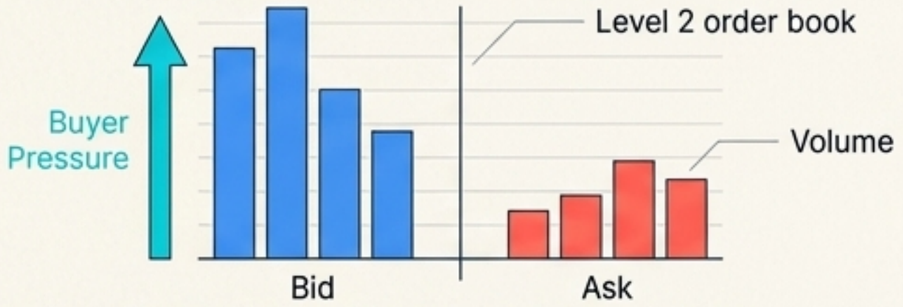


BEYOND THE BAR: SIMULATING MARKET MICROSTRUCTURE

Bar data (OHLCV) hides intra-bar dynamics. For strategies sensitive to execution, liquidity, or the bid-ask spread, we need to model the order book at the tick level.

CORE MICROSTRUCTURE CONCEPTS

ORDER BOOK IMBALANCE (OBI)



Quantifies the relative pressure between the buy and sell sides at the top of the order book. An OBI > 0 indicates excess buyer pressure.

$$\text{OBI} = \frac{\text{Aggregated Bid Volume} - \text{Aggregated Ask Volume}}{\text{Aggregated Bid Volume} + \text{Aggregated Ask Volume}}$$

Predictive Power: Empirically, a positive OBI is correlated with short-term upward price movements.

STOIKOV'S MICRO-PRICE

A volume-weighted adjustment to the mid-price that incorporates the information from the OBI. It is a better predictor of the next tick's price than the standard mid-price.

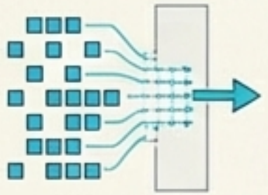
$$\text{Micro-Price} = \frac{\text{BidSize} \times \text{AskPrice} + \text{AskSize} \times \text{BidPrice}}{\text{AskSize} + \text{BidSize}}$$

PROTOTYPE 3: THE MICROSTRUCTURE ‘SNIPER’ STRATEGY

A high-frequency strategy designed to capture the bid-ask spread by predicting short-term price movements using Order Book Imbalance Tick (OBI) and the Micro-Price.

HOW IT WORKS: AN EVENT-DRIVEN, TICK-LEVEL LOGIC

1. DATA INGESTION



2. REAL-TIME CALCULATION

C++

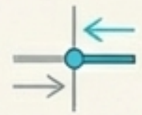
$$\text{OBI} = \frac{V_{\text{bid}} - V_{\text{ask}}}{V_{\text{bid}} + V_{\text{ask}}}$$

$$\text{Micro-Price} = \frac{P_{\text{bid}} * V_{\text{ask}} + P_{\text{ask}} * V_{\text{bid}}}{V_{\text{bid}} + V_{\text{ask}}}$$

3. SIGNAL GENERATION

If OBI > threshold$_A$ and Micro-Price > Mid-Price + threshold$_B$, anticipate an upward move.

PASSIVE ‘FADING’



Place a limit order against the imbalance (e.g., place a sell limit order if strong buying pressure is detected, anticipating a short-term mean reversion).

AGGRESSIVE ‘TAKING’

Place a market order to trade in the direction of the imbalance if momentum is confirmed.

KEY PARAMETERS FOR TUNING & VALIDATION

OBI Threshold	The level of imbalance required to generate a signal.
Micro-Price Deviation Threshold	The required divergence between the micro-price and mid-price.
Max Order Lifetime	The time (in ms) an un-filled passive order remains active before cancellation.
Fill Probability Model	Parameters (alpha, beta, gamma) for the stochastic model that estimates the likelihood of a passive limit order getting filled based on queue position and volatility.

TOWARDS PRODUCTION: THE ASYNCHRONOUS REALITY

THE PROBLEM

Real-world trading systems don't just react to market data. They ingest multiple, asynchronous data streams: news feeds, social media sentiment, economic releases, etc. Each stream has its own latency and processing time.

THE CRITICAL DISTINCTION: EVENT TIME VS. KNOWLEDGE TIME

Core Principle: Confusing these two is the most subtle and dangerous form of look-ahead bias. A backtest must process events in the order they become *known* to the system.

FORENSIC EXAMPLE: PROCESSING AN NLP SENTIMENT SIGNAL

1. **NewsEvent:** `event_time: 10:30:45.123Z` (Published), `knowledge_time: 10:30:46.234Z` (Crawled).
2. **SentimentSignal:** `event_time: 10:30:46.234Z` (Now we know), `knowledge_time: 10:30:47.456Z` (Sentiment model finished).
3. **TradeSignal:** `event_time: 10:30:47.456Z` (Now we have sentiment), `knowledge_time: 10:30:47.890Z` (Strategy decision made).

Implication: The system's central event queue must be a priority queue ordered by `knowledge_time` to ensure causal integrity.

PROTOTYPE 4: THE HYBRID ‘ANALYST’ SYSTEM

Core Concept: A production-ready, event-driven architecture that processes multiple asynchronous data sources with strict temporal causality. It employs a hybrid decision model, separating probabilistic signal generation from deterministic risk enforcement.

HOW IT WORKS: A HYBRID ARCHITECTURE

THE ‘ALPHA’ LAYER

Combines signals from various sources: technical indicators, NLP sentiment scores, TDA regime detection models, etc. Each signal includes a `confidence" score.

Outputs: Flexible, probabilistic trading suggestions (e.g., "High confidence BUY signal for ACME")

THE ‘GUARDRAIL’ LAYER

Outputs: Deterministic execution decisions (BUY/SELL/HOLD) that have passed all safety checks

Deterministic Execution & Risk

Receives probabilistic signals and translates them into concrete orders. Before execution, every proposed order is checked against a set of non-negotiable risk guardrails.

NON-NEGOTIABLE RISK GUARDRAILS (EXAMPLES)

Position Limits:
abs(proposed_order.quantity) <= max_position_size

Leverage:
total_notional / equity <= max_leverage

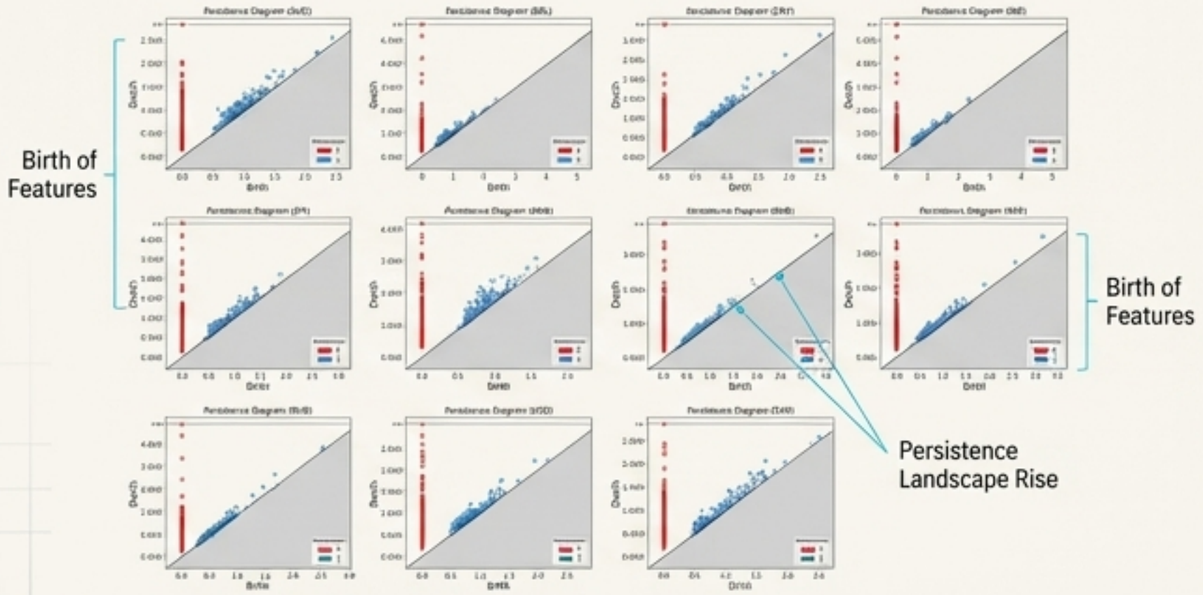
Portfolio Drawdown:
realized_pnl / peak_equity >= -max_dd

THE RESEARCH FRONTIER: ADVANCED SIGNAL & RISK MODELS

Challenge: Traditional signals (e.g., moving averages) can fail during market phase transitions, and traditional portfolio optimizers (Mean-Variance) are notoriously unstable. Advanced methods aim to address these failures.

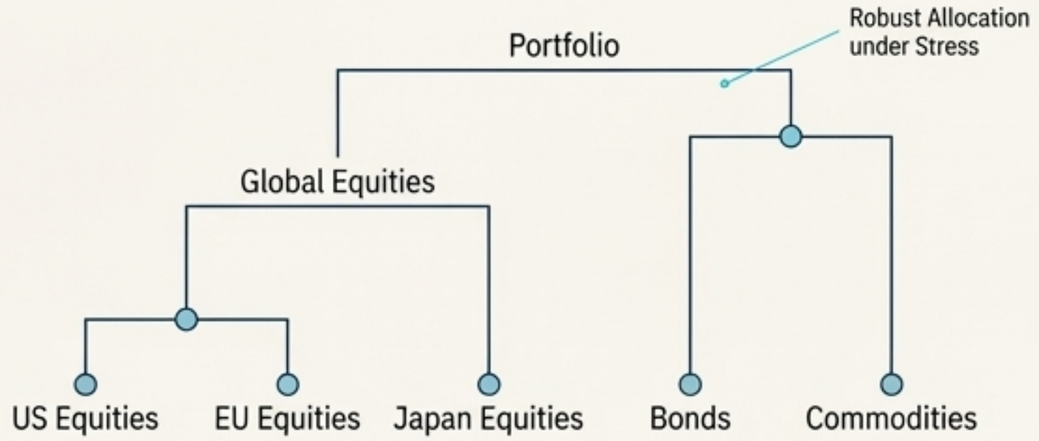
ADVANCED SIGNAL: TOPOLOGICAL DATA ANALYSIS (TDA) FOR REGIME DETECTION

- **Core Idea:** TDA analyzes the "shape" or 'topology' of market data (e.g., asset correlations) to detect structural breaks and phase transitions that precede crises.
- **Mechanism:** Uses **Persistent Homology** to track the birth and death of topological features (like clusters and loops) in a point cloud of returns.
- **Output:** A time-series indicator, such as the **L^1-norm of the Persistence Landscape**, which tends to rise in advance of major market crashes.



ADVANCED ALLOCATION: HIERARCHICAL RISK PARITY (HRP)

- **Core Idea:** An alternative to Mean-Variance Optimization that avoids inverting the unstable covariance matrix.
- **Mechanism:**
  1. Uses hierarchical clustering to group assets based on their correlation structure.
  2. Recursively allocates risk between and within clusters (quasi-diagonalization).
- **Advantage:** More robust to regime shifts. When correlations spike in a crisis, HRP automatically groups those assets and manages their combined risk, preventing portfolio concentration where MVO would fail.

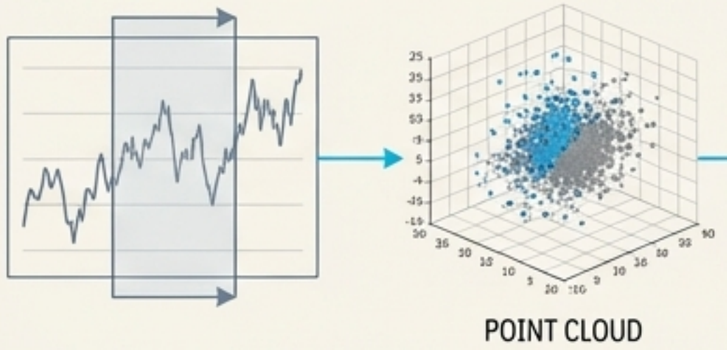


PROTOTYPE 5: THE TOPOLOGICAL ALLOCATOR STRATEGY

Core Concept: A dynamic asset allocation strategy that uses TDA as an early-warning system for market instability, switching between "Risk-On" and "Risk-Off" portfolios constructed using the robust Hierarchical Risk Parity algorithm.

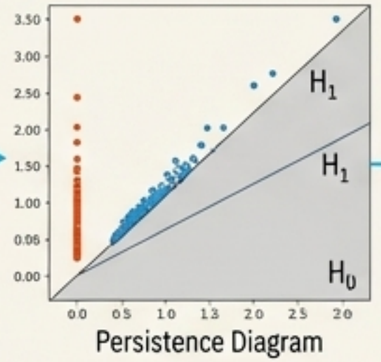
HOW IT WORKS: A TDA-HRP WORKFLOW

1 SLIDING WINDOW & POINT CLOUD



On a rolling basis (e.g., daily), take the last 60 days of asset returns. Create a **point cloud** from this data (e.g., using a correlation distance matrix: $D_{ij} = \sqrt{0.5 * (1 - \rho_{ij})}$).

2 PERSISTENT HOMOLOGY

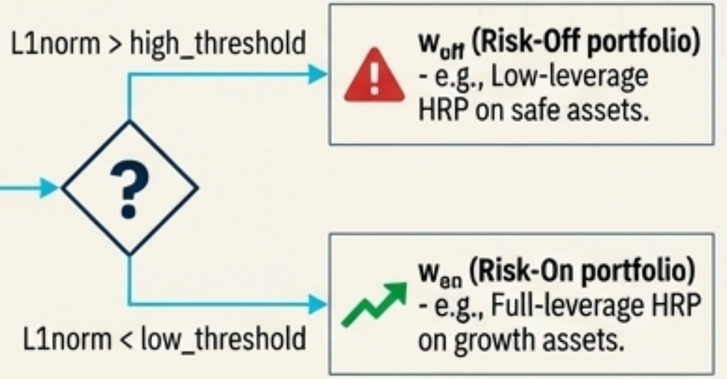


Compute **persistence diagrams** for the point cloud to quantify its topological structure.

3 TOPOLOGICAL INDICATOR

Vectorize the diagram into a feature, like the **L^1-norm of the Persistence Landscape.** This creates a time series of market "topological stability."

4 REGIME SWITCHING LOGIC



Instability detected
Instability detected (Path 1) → Switch to Risk-Off.
Stability returns (Path 2) → Switch to Risk-On.

PARAMETERS FOR TUNING & VALIDATION

- **`Sliding Window Length (W)`**: Balances signal responsiveness with statistical noise (e.g., 60-90 days).
- **`Homology Dimensions`**: Typically H_0 (clusters) and H_1 (loops).
- **`Regime Thresholds`**: Percentiles (e.g., 90th) of the topological indicator's historical distribution.
- **`Portfolio Definitions`**: Asset universe and leverage for `$w_{on}$` vs. `$w_{off}$` HRP allocations.

ADVANCED VALIDATION: STRESS-TESTING WITH AGENT-BASED MODELS

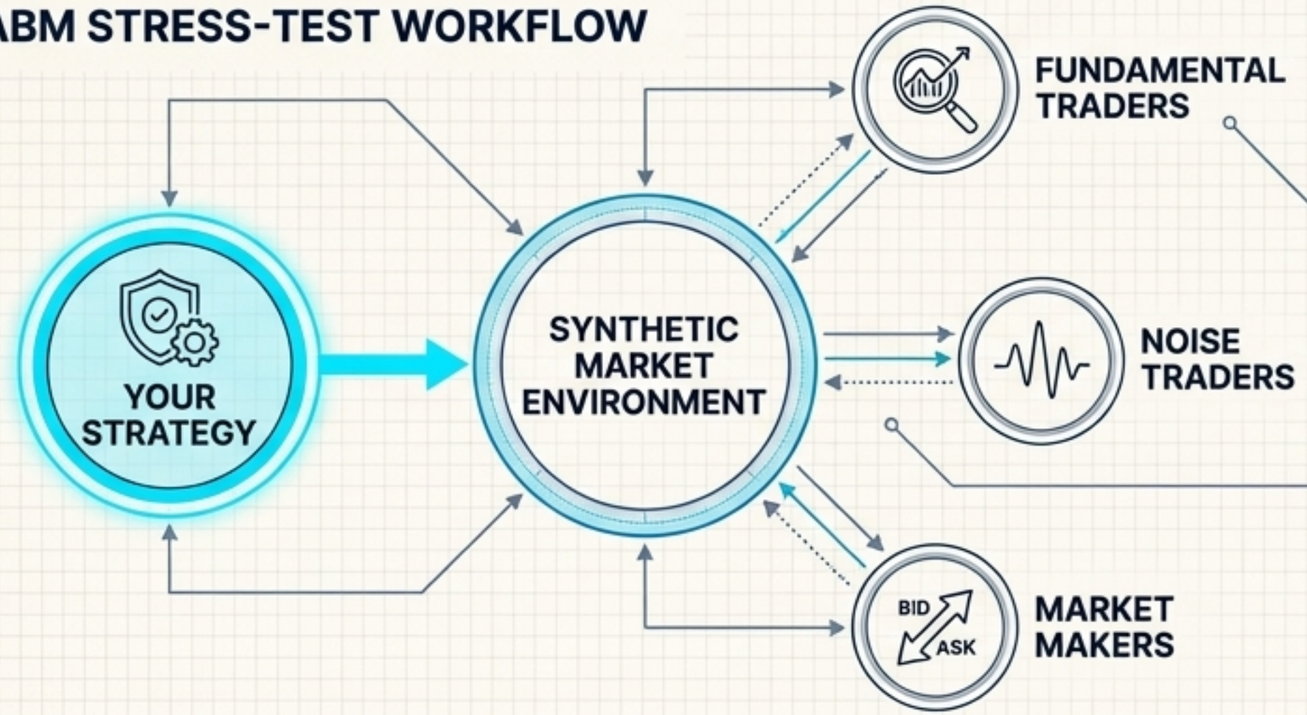
THE LIMITS OF HISTORICAL BACKTESTING

Historical data contains only one path of what actually happened. It may not contain the specific type of crisis or market dynamic that could break a strategy.

CORE CONCEPT

Agent-Based Modeling (ABM) is a computational method that simulates the actions and interactions of autonomous agents (e.g., investors, market makers) to assess their effects on the system as a whole. It allows us to create synthetic market environments and test a strategy's robustness against scenarios not present in historical data.

AN ABM STRESS-TEST WORKFLOW



1. **Define Agents:** Create a market populated by different agent types with simple behavioral rules.
2. **Embed Your Strategy:** Introduce your algorithmic strategy as one of the agents in the market.
3. **Run Scenarios:** Simulate the market under various stress conditions by changing agent parameters or introducing external shocks.

EXAMPLE SCENARIOS FOR STRESS-TESTING

- **High Volatility:** Increase the noise traders' activity or external volatility shocks.
- **Flash Crash:** Introduce a cohort of `Panic Sellers` who liquidate assets regardless of price.
- **Spoofing Manipulation:** Introduce malicious agents that place and cancel orders to manipulate the book.

VALIDATION GOAL

To analyze the strategy's P&L distribution across hundreds of simulations per scenario, identifying its breaking points and tail risks.

SYNTHESIZING THE ARCHITECTURES: A DECISION FRAMEWORK

The choice of architecture is a trade-off between speed, fidelity, and complexity. The right choice depends on the strategy's frequency, data requirements, and path dependency.

COMPARATIVE TABLE OF PROTOTYPES

FACTOR	VECTOR-BASED	CLASSIC EVENT-DRIVEN (BAR-LEVEL)	MICROSTRUCTURE ENGINE (TICK-LEVEL)	HYBRID "ANALYST" SYSTEM
Core Loop	Fixed time-step	Chronological event queue	Chronological event queue	Knowledge-time priority queue
Execution Fidelity	Bar-level (next open/close)	Bar-level (next open/close)	Tick-level (models bid-ask, volume)	Models latency & fill uncertainty
Strategy Complexity	Best for linear, path-independent logic	Handles path dependency	Handles path dependency	Handles asynchronous events & hybrid logic
Risk Management	Bar-level checks only	Bar-level checks only	Intra-bar, event-level checks	Pre-execution guardrails on probabilistic signals
Data Requirements	OHLCV bars	OHLCV bars	Tick/L2 Order Book Data	Multiple unstructured sources (news, sentiment)
Speed vs. Fidelity	Very fast, low fidelity	Slower, higher fidelity	Slowest, very high fidelity	Varies, highest fidelity
Ideal Use Cases	Rapid factor research	Intra-day strategies	HFT & market making	Production-grade multi-asset, multi-signal systems