📘 ANTI–ALPHA-DECAY QUANT SYSTEM ARCHITECTURE

Enterprise-Grade Design for a Multi-Model EOD Quant Platform

TABLE OF CONTENTS

---

1. INTRODUCTION

This document defines a fully specified, production-grade architecture for building a **multi-model EOD quantitative trading system designed to minimize alpha decay and prevent catastrophic model failure**.

The system integrates:

- Multi-source time-series data
- Feature engineering
- Multiple independent predictive models
- Regime-aware and correlation-aware ensembling
- Sophisticated monitoring to detect drift, decay, and structural failures
- Multi-level circuit breaking
- Synthetic monitoring
- Explanatory digital twin layer for users

The architecture incorporates best practices from:

- Quant hedge fund engineering
- ML systems engineering
- Control theory
- Monitoring/observability engineering
- Statistical testing methodologies
- Production-grade risk controls

This document is complete, self-contained, and suitable for direct engineering implementation.

---

2. CORE PRINCIPLES

The design is grounded in the following principles:

(1) Defense-in-Depth

Every layer has specific responsibilities and guarantees.
If one layer fails, the next stops propagation.

(2) No Single Point of Failure

Models, features, regimes, and monitoring each have redundancy.

(3) Control-Theory Stability

Hysteresis, gradual changes, and structured recovery to avoid oscillation.

(4) Explicit Statistical Rigor

Significance tests, confidence intervals, walk-forward CV, embargo/purging, out-of-sample regime validation.

(5) Robustness > Accuracy

The architecture prioritizes stability and controlled degradation over peak backtest metrics.

(6) Explainability & Trust

The digital twin provides transparent rationale for recommendations.

(7) Modular Build

The system is composed of independent layers that can be developed and validated separately.

---

3. HIGH-LEVEL SYSTEM OVERVIEW

RAW DATA
↓
FEATURE STORE + REGIME ENGINE
↓

MODEL LAYER (TFT, GNN, Options, Sentiment, Tech)
  ↓
ENSEMBLE + EXPOSURE (Correlation-Aware + Regime-Aware + Confidence)
  ↓
MONITORING & CONTROL (Drift, Performance, Correlation, Kill-Switch)
  ↓
DIGITAL TWIN (Explanations, Scenarios, Trust)
  ↓
USER / EXECUTION

Control-plane components operate across all layers:

- Model Registry
- Data Version Registry
- Alert Orchestrator
- Meta-Monitor
- Configuration & Threshold System

---

4. RAW DATA LAYER

4.1 Responsibilities

- Ingest all external data sources daily at EOD
- Maintain “as-of” correctness
- Provide reproducible historical snapshots for backtesting
- Track schema evolution and data lineage

4.2 Supported Data Types

- Price: OHLCV

- Options: IV, OI, flows, greeks
- Fundamentals (optional, slow-moving)
- Sentiment: news embeddings, polarity, tone
- Macro: VIX, rates, liquidity
- Indices: SPY, QQQ, sector ETFs
- Related-stock mappings (graph data)

4.3 Versioning Strategy

Two modes (configurable):

Option A: Full Daily Snapshots

- Simpler
- Higher storage cost
- Recommended for <500 symbols

Option B: Monthly Snapshots + Daily Delta Tables

- Monthly full snapshot
- Daily change logs (deltas)
- Reconstruction query to rebuild historical "as-of" view

4.4 Data Integrity Guardrails

- No forward-dated rows
- No overwritten history
- All data tied to:
    - `snapshot_id`

- `as_of`
- `effective_from`
- `effective_to`

---

5. FEATURE STORE & REGIME ENGINE

5.1 Feature Store Responsibilities

- Compute all features from raw data
- Guarantee versioning and reproducibility
- Organize features into logical groups
- Maintain feature metadata

Feature Categories:

- Price-based
- Options-based
- Sentiment-based
- GNN-based (sectoral relationships, influence networks)
- Macro-based

5.2 Regime Engine

Regimes Supported:

- Volatility Regime
- Risk Regime (risk-on/off)
- Gamma Regime
- Liquidity Regime

Outputs:

```
daily_regime(date, vol_regime, risk_regime, gamma_regime,
liquidity_regime, regime_version)
```

5.3 Regime Validation Monitor (critical)

Validates regimes:

Checks:

- Out-of-sample regime performance (Sharpe differences across regimes)
- Minimum regime persistence (≥20–30 days)
- Regime transition frequency stability
- Regime boundary drift
- Regime classification predictive power
- OOD (out-of-distribution) regime detection

---

6. MODEL LAYER

Models are trained independently and designed to be conditionally independent.

Model Families:

- **TFT (Temporal Fusion Transformer)**
- **GNN (Graph Neural Network)** for related-stock dynamics
- **Options Flow Model** (IV, OI, skew, flows)
- **Sentiment Model** (LLM embeddings → regression/classifier)
- **Technical Model** (momentum, mean-reversion signals)

6.1 Training Methodology

- Walk-forward CV
- Purged cross-validation
- Embargo windows
- Multi-year rolling windows
- Parameter sweeps

6.2 Model Registry

Each model includes:

- Train window metadata
- Validation metrics per window
- Regime performance
- Drift sensitivity
- Hash of feature versions

- Conditional independence scores

---

7. ENSEMBLE & EXPOSURE LAYER

Goals:

- Combine multiple signals safely
- Avoid false diversification
- Incorporate regimes
- Provide interpretable outputs
- Prevent overreaction to noise

7.1 Signal Normalization

- Rolling z-scores
- Clamp to [-3, 3]

7.2 Correlation-Aware Ensemble

Model Correlation Monitor:

- Rolling pairwise correlations
- Regime-conditional correlations
- Model clustering to identify redundancy

Actions:

- Reduce effective number of models (N_eff)
- Deweight redundant models
- Alert orchestrator when >0.7 corr persists

7.3 Regime-Aware Weights

Weight matrix:
`W[model, regime]`

Rules:

- Slow updates (≤ 10–20% change/week)
- Hysteresis (easier to deweight than reweight)
- Regime performance validation required

7.4 Confidence Function

A fully defined, transparent function:

confidence = base_conf * agreement_mult * regime_mult * monitor_mult

Where:

- base_conf = sigmoid(|score_raw| − threshold)
- agreement_mult = 0.5 + 0.5 * agreement
- regime_mult = clipped regime Sharpe ratio
- monitor_mult = system state multiplier

Confidence $\in [0,1]$.

7.5 Position Sizing

- Confidence-weighted
- Exposure caps per stock, sector, global
- Skip days when:
    - Low agreement
    - Low confidence
    - System in Defense or Halt

---

8. MONITORING & CONTROL LAYER

This layer prevents alpha decay and catastrophic loss.

8.1 Drift Monitoring

Metrics:

- PSI
- KL divergence
- Feature-group drift

Actions:

- Alert orchestrator

- Freeze features if required

8.2 Performance Monitoring

- Rolling Sharpe
- Rolling drawdown
- Hit-rate
- Regime-conditional performance
- 60–120 day windows to reduce noise

8.3 Statistical Significance Testing

Detect significant underperformance using:

- Lo (2002) Sharpe SE
- One-sided t-tests
- Minimum N trades (≥60–100)

8.4 Auto-Deweight Engine

Rules:

- Trigger on significant decay only
- Hysteresis:
    - Deweight after sustained degradation

    - Reweight only after stronger improvement
- Max adjustment ≤ 10–20% per week

8.5 Multi-Level Kill-Switch

Level 1 — Caution Mode

- Reduce entry sizes
- No new strategies

Level 2 — Defense Mode

- Stop new entries
- Cut positions by 50%

Level 3 — Full Halt

- Gradual safe liquidation
- Full stop
- Requires manual diagnostics

Structured Recovery:

- Paper trading → 10% → 25% → 50% → 100%
- Each step requires performance + monitoring criteria

8.6 Monitoring Orchestrator

Prevents alert fatigue.

Responsibilities:

- Severity classification
- Alert aggregation
- Alert suppression
- Routing
- Meta-alerts

Severity Budget:

- P1 Critical: ≤ 3/day
- P2 Warning: ≤ 10/day
- P3 Info: unlimited

---

9. DIGITAL TWIN / USER INTERACTION LAYER

Responsibilities:

- Explain ensemble decisions
- Present signal decomposition by model
- Show correlation structure
- Provide scenario analysis
- Reflect monitoring state (e.g., “System in Defense Mode”)

Outputs:

- Daily brief
- Buy/sell/skip recommendations
- Confidence & rationale
- Regime context
- Model-specific explanations

---

10. META-MONITORING & SYNTHETIC TESTING

Synthetic Tests:

- Feature drift injection
- Regime flip simulation
- Signal corruption
- Performance collapse

Validates:

- Drift monitors trigger
- Alerts routed correctly
- Kill-switch activates as designed
- Ensemble confidence behaves predictably
- Regime classifier responds appropriately

Meta-monitor ensures the monitoring system itself remains reliable.

---

11. SCENARIO SIMULATION ENGINE

Responsibilities:

- Allow users to test macro/market stress scenarios
- Adjust relevant features and recompute signals
- Display confidence degradation
- Identify OOD scenarios

Features:

- Historical analog matching
- Bootstrapped stress periods
- Scenario plausibility warnings

---

12. IMPLEMENTATION PHASING

Phase 0 — Core Alpha Validation

(Required before expensive infra)

- Single model (TFT)
- Single symbol or small universe
- No versioning, no monitoring

- Validate true alpha exists

Phase 1 — MVP

- Snapshot versioning
- Feature store
- TFT-only system
- Manual kill-switch
- Basic drift + Sharpe monitoring

Phase 2 — Multi-Model

- Add GNN, Sentiment, Options, Tech
- Simple equal-weight ensemble
- Regime engine (basic version)
- Alert orchestrator (basic)

Phase 3 — Full Anti-Decay System

- Correlation monitor
- Auto-deweight with statistical testing
- Multi-level kill-switch
- Regime validation
- Hysteresis everywhere

Phase 4 — Advanced

- Scenario simulation

- Full meta-monitoring
- Synthetic fire drills
- Gradual kill-switch recovery protocols
- Hybrid versioning
- Production-grade hardening

---

13. ENGINEERING DELIVERY EXPECTATIONS

13.1 Deliverables

- Modular services or microservices for each layer
- Pipeline orchestration (Airflow/Prefect/Lightning)
- Data schemas with versioning
- Model registry
- Monitoring dashboards
- Alert orchestrator system
- Digital twin UI/API
- Configuration system for thresholds/hysteresis

13.2 Required Expertise

- ML engineering
- Quant modeling

- Time-series forecasting
- Observability engineering
- Database versioning
- Control-plane design
- Data pipelines

13.3 SLAs / KPIs

- EOD pipeline completes within 60–120 minutes
- Drift detection latency < 5 minutes
- Alert false-positive rate < 10%
- Ensemble weight stability maintained
- Kill-switch correctness validated weekly via synthetic tests

---

✔This is your complete, independent, production-ready architecture

It is **fully implementable** and covers:

- All guardrails
- All model and ensemble logic
- All monitoring
- All control systems
- All user-facing logic

- All meta-monitoring
- Strict statistical criteria
- Control theory for stability
- Operational realities
- A phased delivery plan