Strawberry Labs — Research

Todd B. Adams

Table of contents

Preface

Strawberry Labs is Todd B. Adams's research programme: the systemic-risk / market-phase-transition thesis, with the SBFoundation platform as its empirical instrument.

It gathers, in one place: - the **thesis** (proposal, research questions, factor catalogue and models) - the account of the **platform as a research substrate** - the empirical **findings**, the **methodology** and **experiments** - the working **journal** and **publications** - a set of **appendices** — the platform-depth and whitepaper documents (surfaced from their canonical home in the SBFoundation repository) and a generated **Reference Library** of the academic sources the work builds on.

Part I.

Thesis

1. Doctoral Research

PhD-track research and reference material.

- proposal — PhD research proposal: dynamic heterogeneous multiplex networks for factor-based systemic risk & market phase transitions.
- factor models and benchmarks — factor loadings & returns, the CAPM decomposition `r =  + ·market + `, systematic vs. idiosyncratic return, factor→signal wiring, benchmark selection, and a canonical factor catalogue — grounded in the SBFoundation codebase.
- reference library — the unified library of paper summaries, including the networks/GNN literature (proposal §2) and the platform's factor/cost/survivorship references, each with a link to its source.
- research platform — the supporting evidence package showing the platform is an end-to-end (data → live trading) empirical instrument that hosts and validates this proposal's research.

2. PhD Research Proposal: Dynamic Heterogeneous Multiplex Networks for Factor-Based Systemic Risk & Market Phase Transitions

- **Candidate:** Todd B. Adams
- **Target Department:** Department of Mathematics (Disordered Systems Group) / Department of Informatics, King's College London
- **Proposed Supervisors:** Prof. Tiziana Di Matteo (Mathematics), Co-Supervisor (Informatics)

2.1. 1. Executive Summary & Research Proposal

2.1.1. 1.1 The Problem

Traditional factor investing frameworks (e.g., Fama-French, Barra) assume linear, stationary relationships and treat asset co-movements independently. They fail to capture structural dependencies, liquidity crowding, and cascading supply-chain risks. Consequently, standard risk models

fail catastrophically during market anomalies because they treat systemic risk as a collection of isolated asset volatilities rather than an emergent property of a complex, interconnected system.

2.1.2. 1.2 Objective

This research project proposes a novel framework utilizing **Heterogeneous Multiplex Networks** coupled with **Spatial-Temporal Graph Neural Networks (ST-GNNs)** to model the global equity market. By treating assets, systematic factors, and institutional funds as unique nodes across specialized topological layers, we aim to:

1. Formulate the mathematical constraints governing shock diffusion across multi-layered financial topologies.
2. Leverage the eigenvalue spectra of the network's Supra-Laplacian matrix to detect geometric signatures of **critical slowing down** (early warning signals of market phase transitions/crashes).
3. Build a scalable, non-linear message-passing architecture that outperforms traditional linear vector autoregression (VAR) in predicting systemic risk propagation.

2.2. 2. Theoretical Foundations & Academic Literature

To establish rigorous groundings in both the statistical mechanics of complex networks and state-of-the-art graph architectures, the following core literature will form the basis of the literature review:

2.2.1. Econophysics & Financial Network Topology

- **Caldarelli, G. (2007).** *Scale-Free Networks: Complex Webs in Nature and Technology.* Oxford University Press. (Foundational network physics).
- **Musmeci, N., Aste, T., & Di Matteo, T. (2015).** *Relation between financial market structure and the real economy via information filtering networks.* Journal of Network Theory in Finance. (Key KCL methodology on filtered correlation topologies).
- **Bardoscia, M., Battiston, S., Caccioli, F., & Caldarelli, G. (2017).** *Pathways towards instability in financial networks.* Nature Communications 8:14416. (Mechanics of cascading failures in economic systems).

2.2.2. Multiplex & Multilayer Network Theory

- **Boccaletti, S., et al. (2014).** *The structure and dynamics of multilayer networks.* Physics Reports. (The definitive mathematical formulation of multilayer systems).
- **Kivelä, M., et al. (2014).** *Multilayer networks.* Journal of Complex Networks. (Terminology definitions for heterogeneous vs multiplex systems).

2.2.3. Graph Neural Networks & Graph ML

- **Kipf, T. N., & Welling, M. (2016).** *Semi-Supervised Classification with Graph Convolutional Networks.* arXiv. (Inception of standard GCN message-passing mechanics).
- **Yu, B., Yin, H., & Zhu, Z. (2017).** *Spatio-Temporal Graph Convolutional Networks: A Deep Learning Framework for Traffic Forecasting.* IJCAI. (Adaptable directly to financial time-series forecasting across graphs).

2.3. 3. High-Level Architecture

The system pipeline ingests empirical or simulated market vectors, structures them into a tensor-based multiplex graph, passes them through a geometric deep learning engine, and evaluates structural stability metrics.

Pipeline Stage	Subsystems & Architectural Components	Key Processes & Mathematical Tasks
1. Data Ingestion Engine	• Factor Loadings • Supply Chain Dependencies • Institutional Fund Portfolios	Ingests continuous empirical or simulated multi-source market vectors.
2. Supra-Adjacency Tensor Builder	• Layer 1: Bipartite Stock-Factor • Layer 2: Directed Unipartite Stock-Stock • Layer 3: Bipartite Institution-Stock	Maps isolated topology layers with variable edge weights and directed linkages into a global coordinate matrix ($\mathcal{A}$).
3. Geometric Deep Learning Framework	• Spatial-Temporal GNN (ST-GNN) • Heterogeneous Graph Attention Network (HAN) • Dynamic Feature Tracking	Executes non-linear message-passing architectures across node features like rolling price, volatility, and factor edge arrays.

Pipeline Stage	Subsystems & Architectural Components	Key Processes & Mathematical Tasks
4. Physics Evaluation Engine	• Supra-Laplacian Eigenvalue Spectral Analysis • Phase Transition Identification	Extracts the algebraic connectivity matrix to identify early warning metrics like the Critical Slowing Down Index before market anomalies.

2.3.1. Architectural Subsystems

1. **The Representation Layer:** Built on top of `PyTorch Geometric (PyG)`, transforming heterogeneous nodes into independent vector spaces, mapping inter-layer couplings via a Supra-Adjacency matrix.
2. **The Message-Passing Engine:** Utilizes a Modified Heterogeneous Graph Transformer (HGT) to account for time-varying edge weights (dynamic factor loadings) and node attributes.
3. **The Spectral Analyzer:** Computes the algebraic connectivity (second smallest eigenvalue of the Supra-Laplacian) to flag structural vulnerability thresholds.

2.4. 4. Software Dependencies (`requirements.txt`)

As an expert-level Python implementation, the environment relies heavily on GPU-accelerated tensor math and specialized geometric deep learning distributions.

```
torch>=2.4.0
torch-geometric>=2.5.0
networkx>=3.3
numpy>=1.26.0
pandas>=2.2.0
scikit-learn>=1.4.0
scipy>=1.12.0
tensorly>=0.8.1
matplotlib>=3.8.0
```

2.5. 5. Data Requirements & Sourcing Options

To validate the model empirically, the architecture requires distinct financial and structural datasets:

Data Category	Target Variables	Academic / Free Sourcing	Commercial Sourcing
Asset Pricing & Factors	Daily/Intraday OHLCV, Fama-French 3/5 factor portfolios, Momentum, Volatility vectors.	French Data Library, Yahoo Finance API, OpenBB SDK.	CRSP, Axioma, Barra (MSCI), Bloomberg API.

Data Category	Target Variables	Academic / Free Sourcing	Commercial Sourcing
Supply Chain Links	Customer-Supplier transaction directionality, percentage revenues.	Academic papers, SEC Edgar Parsing (Form 10-K extraction via NLP).	FactSet Revere, Bloomberg SPLC.
Institutional Ownership	Fund holdings, quarterly share counts, capital under management.	SEC Form 13F filings via EDGAR system.	Whalewisdom, Thomson Reuters (Refinitiv).

2.6. 6. Synthetic Data Simulation Strategy (Model Proofing)

To verify the GNN framework and ensure the physics engines can reliably detect phase transitions *before* deploying real-world market data, we must build a synthetic data simulator. This guarantees a controlled environment where we can inject artificial systemic shocks.

2.6.1. 6.1 Simulation Design

We model a synthetic environment with N=100 Stocks, F=5 Factors, and I=10 Institutional Funds.

1. **Layer 1 (Factor Loadings):** Generated via structural stochastic differential equations (SDEs) where asset returns are driven by latent brownian paths of factors.
2. **Layer 2 (Supply Chain):** Structured as a Scale-Free Directed Network using a Barabási–Albert model variant.
3. **Layer 3 (Ownership Crowding):** Structured as random bipartite linkages with a tunable concentration parameter to simulate "crowded trades".

2.6.2. 6.2 Python Verification Script

The following production-ready script generates the synthetic multiplex data tensors and computes the structural stability profile using the network's Laplacian.

```
import numpy as np
import pandas as pd
import networkx as nx
import scipy.sparse as sp

class MarketMultiplexSimulator:
    def __init__(self, n_stocks=100, n_factors=5, n_funds=10):
        self.n_stocks = n_stocks
        self.n_factors = n_factors
        self.n_funds = n_funds
        self.total_nodes = n_stocks + n_factors + n_funds

        # Node index mapping
        self.stock_idx = np.arange(0, n_stocks)
        self.factor_idx = np.arange(n_stocks, n_stocks + n_factors)
        self.fund_idx = np.arange(n_stocks + n_factors, self.total_nodes)

    def generate_layer_1_factors(self):
```

```
        """Generates continuous bipartite factor exposures (Stock x Factor)"""
        # Dense random matrix representing factor loadings (Beta weights)
        loadings = np.random.normal(loc=0.5, scale=0.2, size=(self.n_stocks, self.n_factors)
        # Enforce sparsity or structural constraints if needed
        loadings[loadings < 0.2] = 0.0
        return loadings

    def generate_layer_2_supply_chain(self):
        """Generates a scale-free directed adjacency matrix for stocks"""
        g = nx.scale_free_graph(n=self.n_stocks, alpha=0.41, beta=0.54, gamma=0.05, seed=42)
        adj = nx.to_numpy_array(g)
        return adj

    def generate_layer_3_funds(self, crowding_factor=0.1):
        """Generates institutional ownership matrix (Fund x Stock)"""
        # Crowding factor increases probability of overlapping portfolios
        base_prob = 0.15 + crowding_factor
        holdings = np.random.binomial(n=1, p=base_prob, size=(self.n_funds, self.n_stocks)).
        # Weight holdings randomly to represent capital allocation percentage
        holdings *= np.random.uniform(0.01, 0.25, size=holdings.shape)
        return holdings

    def build_supra_laplacian(self, loadings, supply_chain, holdings, interlayer_coupling=1.
        """
        Assembles the comprehensive Supra-Adjacency and Supra-Laplacian matrix.
        Handles embedding the isolated layer topologies into a uniform global coordinate sys
        """
        # 1. Initialize empty global Adjacency Matrix
        W = np.zeros((self.total_nodes, self.total_nodes))

        # Embed Layer 1 (Bipartite Stock-Factor mappings)
        for s in range(self.n_stocks):
            for f in range(self.n_factors):
```

```
                weight = loadings[s, f]
                if weight > 0:
                    W[self.stock_idx[s], self.factor_idx[f]] = weight
                    W[self.factor_idx[f], self.stock_idx[s]] = weight # Symme

        # Embed Layer 2 (Unipartite Stock-Stock mappings)
        W[np.ix_(self.stock_idx, self.stock_idx)] += supply_chain

        # Embed Layer 3 (Bipartite Fund-Stock mappings)
        for fn in range(self.n_funds):
            for s in range(self.n_stocks):
                weight = holdings[fn, s]
                if weight > 0:
                    W[self.fund_idx[fn], self.stock_idx[s]] = weight
                    W[self.stock_idx[s], self.fund_idx[fn]] = weight

        # Add uniform identity-based inter-layer coupling to model identity r
        np.fill_diagonal(W, interlayer_coupling)

        # 2. Compute Degree Matrix
        row_sums = np.sum(W, axis=1)
        D = np.diag(row_sums)

        # 3. Formulate Laplacian
        L = D - W
        return L

    def compute_stability_metrics(self, L):
        """
        Uses spectral analysis of the Laplacian to identify structural vulner
        The second smallest eigenvalue (Algebraic Connectivity) measures netw
        """
        eigenvalues = np.linalg.eigvalsh(L)
```

```
        # Sorted naturally ascending by eigvalsh
        unique_vals = np.sort(eigenvalues)

        # The first eigenvalue of a valid Laplacian is always 0 (up to numerical precision)
        algebraic_connectivity = unique_vals[1]
        spectral_radius = unique_vals[-1]

        return {
            "algebraic_connectivity": algebraic_connectivity,
            "spectral_radius": spectral_radius,
            "system_brittleness_index": 1.0 / (algebraic_connectivity + 1e-6)
        }

if __name__ == "__main__":
    # Execute structural simulation
    sim = MarketMultiplexSimulator(n_stocks=100, n_factors=5, n_funds=12)

    print("[1] Simulating Normal Market Regime...")
    l1 = sim.generate_layer_1_factors()
    l2 = sim.generate_layer_2_supply_chain()
    l3 = sim.generate_layer_3_funds(crowding_factor=0.0) # Base system diversification

    L_normal = sim.build_supra_laplacian(l1, l2, l3)
    metrics_normal = sim.compute_stability_metrics(L_normal)

    print(f" -> Normal Market Connectivity: {metrics_normal['algebraic_connectivity']:.5f}")
    print(f" -> Normal System Brittleness:  {metrics_normal['system_brittleness_index']:.5f}

    print("[2] Simulating Highly Crowded/Stressed Market Regime...")
    # Injecting systemic risk via high factor concentration and institutional crowding
    l1_stressed = l1 * 2.5
    l3_stressed = sim.generate_layer_3_funds(crowding_factor=0.65) # Massive portfolio overl
```

```
    L_stressed = sim.build_supra_laplacian(l1_stressed, l2, l3_stressed)
    metrics_stressed = sim.compute_stability_metrics(L_stressed)

    print(f" -> Stressed Market Connectivity: {metrics_stressed['algebraic_co
    print(f" -> Stressed System Brittleness:  {metrics_stressed['system_britt

    print("\n[Result] Verification Successful: Brittleness increase confirms
```

3. Research Questions

Extracted from the PhD proposal §1 so the current state of the research question doesn't require reading the full proposal. When the proposal's hypothesis or questions change, update both.

3.1. Central hypothesis

Changes in the spectral properties and meso-scale organisation of dynamic multiplex financial networks — constructed from factor exposures, supply-chain dependencies, and institutional ownership — provide statistically significant early-warning signals of systemic market instability, with measurable lead time relative to standard volatility- and correlation-based indicators.

3.2. Overarching question

Can dynamic heterogeneous multiplex representations of financial markets provide robust early-warning signals of systemic instability that materially precede conventional volatility-based and return-correlation risk measures?

3.3. Current hypotheses / sub-questions

1. Which multiplex network observables (algebraic connectivity, eigenvalue gap, spectral density shifts, network entropy, community fragmentation, centrality concentration) reliably signal impending systemic instability?
2. How much lead time do these indicators provide compared with standard risk metrics (implied/realised volatility, DCC-GARCH correlation spikes, realised drawdowns)?
3. Under what network generative mechanisms (ER, BA, stochastic block models, empirically-derived topologies) do the indicators and detection performance generalise?
4. Do graph neural network models improve predictive skill over spectral/hand-crafted indicators, and if so, by how much and at what cost to interpretability?

3.4. Open questions (not yet in the proposal's formal scope)

- Does a multi-layer (factor + supply-chain + ownership) network outperform the single-layer factor/comomentum view the platform already measures, or does it just add noise? The platform's own single-layer experiment found connectedness moved *with* stress, not ahead of it — see research-platform/findings/asset-dependency-networks-econophysics-review.md. This is the open question the multi-layer extension is meant to answer, and the design is meant to fail cheaply if it doesn't.
- How should a validated early-warning signal enter the trading system without becoming an optimizer that sizes the book — i.e. staying risk observability, not a second lifecycle gate? (Boundary asserted in whitepaper §7, not yet resolved in the research.)

3.5. Where the answers will come from

- Methodology — how each question is tested.
- Experiments — the running record of what's been tried against each question.
- Journal — the raw, dated account of what changed a question, closed one, or opened a new one.

4. Factor Catalog

A comprehensive, self-contained reference covering every factor in the canonical research set — including the economic rationale, stylistic guidance, platform implementation details, and empirical performance results for each factor, organized by factor family.

Scope note — factors, not strategies. A *factor* is a measured, predictive characteristic of an instrument (this chapter); a *strategy* is a downstream `(factor_id, mechanic, params)` triple that trades one — see the Strategy Catalog. Several factors below are traded by more than one strategy, and a couple of entries exist *only* because a downstream strategy needs a distinct evaluation path (funnel-exempt entries, flagged where they occur). This chapter deliberately stops at the factor boundary: mechanic, universe, sizing, and backtest/promotion detail all live in the Strategy Catalog, not here.

4.1. Provenance

- **Theory fields** (economic justification, style, anchor citation, hypothesis class, expected sign, source table/column, lifecycle status) are pulled directly from `config/factors/*.yaml` in SBFoundation as of 2026-08-03.
- **Empirical fields** (IC-IR, MCPT) come from the diagnostic run `260526_6ddcd9` (2026-05-26), synthesized in `platform/factors.md`. That run **predates three fixes** that materially change these numbers — F-136 (MCPT nightly permutations 100 → 1000, corrected

null construction), F-137 (Alphalens forward-return winsorization), F-138 (composite contribution rescaling). Treat every MCPT figure below as provisional pending a rerun.

- **Two columns — naive t-stat and HAC (Newey–West) t-stat — are specified but not yet computed.** No sidecar currently produces them; see Method below for exactly how to fill them in. They are marked `PENDING` throughout rather than guessed.
- **Re-grounding against live config surfaced a correction to the run narrative**: `bb_lower_20`, `ma_50d`, `ma_200d`, and `vwap_20d` were described in the `260526_6ddcd9` synthesis as the four factors with "real" pre-fix MCPT significance — but all four were subsequently deprecated *for cause* on 2026-07-20 (F-289/LH-25, run `260718_31a51e`): they're raw price-level source columns with a CF-1 look-ahead issue, not a validated edge. `bb_lower_20`'s entry below reflects this; `bb_pct` (§13) is its scale-free replacement.
- **Known duplicates — resolved 2026-08-05.** `beta` `market_beta_252d` and `momentum_12_1` `momentum_12m_1m` are numerically identical to 14 digits per the run synthesis; neither pair has ever had two independent `config/factors/*.yaml` files (no `beta.yaml` or `momentum_12_1.yaml` exists — both are stale legacy ids with no config of their own). `beta` and `momentum_12_1` are retired; `market_beta_252d` and `momentum_12m_1m` are the canonical survivors and are the only ones counted in the platform's family-wise trial count (F-321).

4.2. Method: the pending t-stat columns

Every factor entry below has two `PENDING` fields. Both are specified the same way, stated once here rather than repeated 24 times:

- **Naive t-stat** — the classic Fama–MacBeth / Fama–French second-pass report. Take the daily factor-return series `F_t` for the `factor_id` from `ops.risk_factor_return` (factor-models-and-benchmarks.md §2.2), and compute `t_naive = mean(F) / (std(F) / sqrt(T))`.
- **HAC (Newey–West) t-stat** — regress the same series on a constant only (`X = [1]`) using the platform's existing HAC estimator, `sbattribution.ols.ols_hac` (`src/sbattribution/ols.py:79-125`) — the identical kernel already used for CAPM/attribution alpha (Bartlett weights, lag `L = floor(4·(n/100)^(2/9))`, Newey–West 1994). The intercept *is* the mean factor return; its HAC standard error replaces the naive `std(F)/sqrt(T)` denominator.
- **Why both**: daily factor returns built on slow-moving or overlapping-window characteristics (12-1 momentum, vol-scaling, TSMOM) are serially correlated by construction, so the naive t-stat overstates significance. The gap between the two is itself diagnostic.
- **Neither is a promotion gate.** Per research-platform/08-validation-and-anti-overfitting.md, a t-stat — naive or HAC — is exactly the single-test statistic the Harvey–Liu–Zhu multiple-testing critique targets. The platform's actual promotion gate is IC-IR + MCPT permutation p-value against an honest trial count; these two columns are diagnostic context, not a second gate.
- **Not yet wired**: `ols_hac` today only runs against strategy NAV streams (`beta_attribution_service.py`), not the per-factor return panel — computing these columns requires a small new script, not a config change.

4.3. 1. Market

Compensation for bearing undiversifiable market risk — the one factor everyone must hold; the CAPM premium. *Sharpe 1964 [#116], Lintner 1965 [#117].*

4.3.1. `market_beta_252d`

- **Style**: `beta`
- **Source**: `fact_eod.market_beta_252d_f` — trailing 252-day rolling OLS slope vs. SPY
- **Equation**: $\beta_{i,t} = \dfrac{\mathrm{Cov}_{252d}(r_i,\, r_{SPY})}{\mathrm{Var}_{252d}(r_{SPY})}$
- **Hypothesis class / expected sign**: not set in config
- **Lifecycle**: `experimental` — canonical survivor of the `beta`/`market_beta_252d` duplicate pair (resolved 2026-08-05); `beta` is retired
- **IC-IR @21d**: not itemized in the run synthesis
- **MCPT (pre-fix)**: p=1.000 (ceiling), `unbiased_ic_ir` −6.8
- **t-stat (naive)**: `PENDING`
- **t-stat (HAC, Newey–West)**: `PENDING`
- **n_obs / cadence**: daily; diagnostics "healthy" quadrant (stationary, entropy 3)
- **Notes**: `beta` and `market_beta_252d` are numerically identical to 14 digits; `beta` was a legacy id with no independent `config/factors/beta.yaml` and is now retired.

4.4. 2. Size

Small caps earn a premium for illiquidity, distress, and limited analyst coverage; partly compensation, partly a limits-to-arbitrage effect. *Banz*

1981 [#125], Fama–French 1992 [#55].

4.4.1. `market_cap`

- **Style**: `size`
- **Source**: `fact_valuation_annual.market_cap_f` — `year_end_close × shares_outstanding`
- **Equation**: $\text{MktCap}_{i,t} = P^{YE}_{i,t} \times \text{SharesOut}_{i,t}$
- **Hypothesis class / expected sign**: not set in config
- **Lifecycle**: `experimental`
- **IC-IR @21d**: −0.5 (inverted — part of the low-vol/size cluster; trade short)
- **MCPT (pre-fix)**: not itemized
- **t-stat (naive)**: `PENDING`
- **t-stat (HAC, Newey–West)**: `PENDING`
- **n_obs / cadence**: annual — short-T caveat applies (ADF/entropy diagnostics have little power at n 12)
- **Notes**: —

4.5. 3. Value

Cheap (high E/P, B/M, FCF-yield) stocks out-earn expensive ones — risk premium for distressed/low-growth firms *and* mispricing from extrapolation. *Fama–French 1992/1993 [#55][#44].*

4.5.1. `value_composite`

- **Style**: `value`

- **Source**: `fact_fundamental_annual.value_composite_f` — multi-ratio value z-score blend (earnings, book, sales, cash-flow yields)
- **Equation**: $\text{ValueComposite}_{i,t} = \frac{1}{4} \sum_{k \in \{EY,\, B/M,\, S/P,\, FCF/P\}} z(x_{k,i,t})$ — config specifies the four component yields and calls it a "blend"; the equal weighting shown here is the simplest reading and is not independently confirmed by a documented weight vector.
- **Hypothesis class / expected sign**: `risk_premium`, positive
- **Lifecycle**: `experimental`, promotion candidate
- **IC-IR @21d**: **+0.75** — highest in the run
- **MCPT (pre-fix)**: not itemized
- **t-stat (naive)**: `PENDING`
- **t-stat (HAC, Newey–West)**: `PENDING`
- **n_obs / cadence**: annual
- **Notes**: sign-flips at h=1 (don't trade intraday) — classic value-investor pattern, right direction only at a month+ horizon

4.5.2. `earnings_yield`

- **Style**: `value`
- **Source**: `fact_valuation_annual.earnings_yield_f` — `eps_diluted / year_end_close`
- **Equation**: $EY_{i,t} = \frac{EPS_{i,t}^{diluted}}{P_{i,t}^{YE}}$ (inverse of `pe_ratio`)
- **Hypothesis class / expected sign**: `risk_premium`, positive
- **Lifecycle**: `experimental`
- **IC-IR @21d**: sign-flip — negative at h=1, positive at h 21
- **MCPT (pre-fix)**: not itemized
- **t-stat (naive)**: `PENDING`
- **t-stat (HAC, Newey–West)**: `PENDING`
- **n_obs / cadence**: annual
- **Notes**: —

4.5.3. `pe_ratio`

- **Style**: `value`
- **Source**: `fact_valuation_annual.pe_ratio_f` — `year_end_close / eps_diluted`
- **Equation**: $PE_{i,t} = \dfrac{P^{YE}_{i,t}}{EPS^{diluted}_{i,t}}$
- **Hypothesis class / expected sign**: `risk_premium`, negative
- **Lifecycle**: `experimental`
- **IC-IR @21d**: -0.5 to -0.7 (inverted group, with `volatility_*`/`market_cap`/`amihud`)
- **MCPT (pre-fix)**: not itemized
- **t-stat (naive)**: `PENDING`
- **t-stat (HAC, Newey–West)**: `PENDING`
- **n_obs / cadence**: annual
- **Notes**: —

4.5.4. `pb_ratio`

- **Style**: `value`
- **Source**: `fact_valuation_annual.pb_ratio_f` — `market_cap / total_stockholders_equity`
- **Equation**: $PB_{i,t} = \dfrac{P^{YE}_{i,t} \times \overline{Shs}^{diluted}_{i,t}}{\text{StockholdersEquity}_{i,t}}$
- **Hypothesis class / expected sign**: `risk_premium`, negative
- **Lifecycle**: `experimental`
- **IC-IR @21d**: not itemized
- **MCPT (pre-fix)**: raw Alphalens top-minus-bottom spread +37.40 at h=21 — a **pre-F-137 artifact** (un-winsorized cross-section outlier dominance), not a real spread; bounded post-fix
- **t-stat (naive)**: `PENDING`
- **t-stat (HAC, Newey–West)**: `PENDING`
- **n_obs / cadence**: annual

- **Notes**: —

4.5.5. `ps_ratio`

- **Style**: `value`
- **Source**: `fact_valuation_annual.ps_ratio_f` — `market_cap / revenue`
- **Equation**: $PS_{i,t} = \dfrac{P_{i,t}^{YE} \times \overline{Shs}_{i,t}^{diluted}}{\text{Revenue}_{i,t}}$
- **Hypothesis class / expected sign**: `risk_premium`, negative
- **Lifecycle**: `experimental`
- **IC-IR @21d**: not itemized
- **MCPT (pre-fix)**: raw Alphalens spread +44.78 at h=21 — pre-F-137 artifact, bounded post-fix
- **t-stat (naive)**: `PENDING`
- **t-stat (HAC, Newey–West)**: `PENDING`
- **n_obs / cadence**: annual
- **Notes**: —

4.5.6. `pfcf_ratio`

- **Style**: `value`
- **Source**: `fact_valuation_annual.pfcf_ratio_f` — `market_cap / free_cash_flow`
- **Equation**: $PFCF_{i,t} = \dfrac{P_{i,t}^{YE} \times \overline{Shs}_{i,t}^{diluted}}{\text{FCF}_{i,t}}$ (inverse of `fcf_yield`)
- **Hypothesis class / expected sign**: `risk_premium`, negative
- **Lifecycle**: `experimental`
- **IC-IR @21d**: not itemized
- **MCPT (pre-fix)**: raw Alphalens spread −36.78 at h=21 — pre-F-137 artifact, bounded post-fix
- **t-stat (naive)**: `PENDING`

- **t-stat (HAC, Newey–West)**: PENDING
- **n_obs / cadence**: annual
- **Notes**: —

4.6. 4. Momentum

Past 12-1 winners keep winning — under-reaction to news and delayed information diffusion (behavioral). *Jegadeesh–Titman 1993 [#49], Carhart 1997 [#57].*

4.6.1. momentum_12m_1m

- **Style**: momentum
- **Source**: fact_eod.momentum_12_1_f
- **Equation**: $\text{Mom}_{i,t} = \frac{P_{i,t-21}}{P_{i,t-252}} - 1$ (12-month cumulative return, skipping the most recent month)
- **Hypothesis class / expected sign**: behavioral, positive
- **Lifecycle**: experimental — canonical survivor of the momentum_12_1/momentum_12m_1m duplicate pair (resolved 2026-08-05); momentum_12_1 is retired
- **IC-IR @21d**: **+0.586**
- **MCPT (pre-fix)**: not itemized
- **t-stat (naive)**: PENDING
- **t-stat (HAC, Newey–West)**: PENDING
- **n_obs / cadence**: daily
- **Notes**: momentum_12_1 and momentum_12m_1m are numerically identical to 14 digits; momentum_12_1 was a legacy id (still used as a raw

bundle/feature column name in `sbbacktest`, but with no independent `config/factors/momentum_12_1.yaml`) and is now retired as a factor id.

4.6.2. `momentum_12_1_vol_scaled`

- **Style**: `momentum`
- **Source**: `fact_eod.momentum_12_1_vol_scaled_f` — the 12-1 trend divided by trailing realized volatility (worked example: factor-models-and-benchmarks.md §6)
- **Equation**: $\text{Mom}_{i,t}^{vol} = \frac{\text{Mom}_{i,t}}{\sigma_{i,t}^{126d}\sqrt{252}}$, where $\text{Mom}_{i,t}$ is the §4 12-1 return above and $\sigma_{i,t}^{126d}$ is the trailing 126-day daily-return standard deviation
- **Hypothesis class / expected sign**: `behavioral`, positive
- **Lifecycle**: `experimental`, promotion candidate
- **IC-IR @21d**: **+0.629**
- **MCPT (pre-fix)**: ceiling-pinned pre-fix; vol-scaling is expected to help it survive overlap correction post-F-136
- **t-stat (naive)**: `PENDING`
- **t-stat (HAC, Newey–West)**: `PENDING`
- **n_obs / cadence**: daily
- **Notes**: same source column also backs the distinct factor entity `tsmom_12_1_vol_scaled` (§12) — identical *value*, different evaluation path (this factor is scored cross-sectionally; §12 is funnel-exempt). The strategies that trade each — `classic-momentum` / `classic-momentum-long-only` / `classic-momentum-dollar-neutral` for this one, `etf-trend` for §12 — are documented in the Strategy Catalog.

4.6.3. `residual_momentum_252d_eod` *(not in the original 13-row citation table — added here because it has the strongest result in the run)*

- **Style**: `momentum`
- **Source**: `fact_eod.residual_momentum_252d_f` — market-neutral momentum: residual of a 252-day rolling OLS of return vs. an equal-weighted benchmark, summed over `[-252:-21]` and z-scaled
- **Equation**: $\varepsilon_{i,\tau} = r_{i,\tau} - (\hat{\alpha}_i + \hat{\beta}_i\, r_{EW,\tau})$ from a 252-day rolling OLS, then $\text{ResMom}_{i,t} = z\left(\sum_{\tau=t-252}^{t-21} \varepsilon_{i,\tau}\right)$
- **Hypothesis class / expected sign**: not set in config
- **Lifecycle**: `experimental` — atomic sibling of the *deprecated* composite `residual_momentum_252d` (`status: deprecated`, B-F-108.1/TASK-907; no `FactorCompositionService` exists to compute the composite recipe, so it carries zero rows). Consumers should use this atomic id, not the composite.
- **IC-IR @21d**: **+0.724** — highest in the entire run
- **MCPT (pre-fix)**: not itemized
- **t-stat (naive)**: `PENDING`
- **t-stat (HAC, Newey–West)**: `PENDING`
- **n_obs / cadence**: daily; diagnostics "healthy" quadrant
- **Notes**: candidate for a 14th row in the citation table given this result — anchor citation would be Blitz–Huij–Martens (residual momentum) **[#50]**. Traded by the `residual-momentum` strategy — see Strategy Catalog §4.

4.7. 5. Profitability / Quality

Profitable, safe, well-managed firms (high gross-profitability, ROIC, QMJ) out-earn junk — a premium for quality that markets under-price. *Novy-Marx 2013 [#92], Fama–French 2015 [#56], Asness–Frazzini–Pedersen 2019 [#113].*

4.7.1. `qmj_composite`

- **Style**: `quality`
- **Source**: `fact_fundamental_annual.qmj_composite_f` — Asness–Frazzini–Pedersen Quality-Minus-Junk blend across profitability, growth, safety, and payout
- **Equation**: $QMJ_{i,t} = z(\text{Profitability}_{i,t}) + z(\text{Growth}_{i,t}) + z(\text{Safety}_{i,t}) + z(\text{Payout}_{i,t})$
- **Hypothesis class / expected sign**: `behavioral`, positive
- **Lifecycle**: `experimental`. `kind: atomic by design` — despite the name, it's a single pre-blended z-score column computed upstream, not a true composite; will always land as `single_input` in the factor-contribution sidecar (F-138).
- **IC-IR @21d**: not itemized
- **MCPT (pre-fix)**: floor bucket (`null_mean_ic_ir` 0, degenerate permutation null — typical of slow-moving annual fundamentals)
- **t-stat (naive)**: `PENDING`
- **t-stat (HAC, Newey–West)**: `PENDING`
- **n_obs / cadence**: annual — short-T caveat applies
- **Notes**: —

4.7.2. `roic_spread`

- **Style**: `quality`

- **Source**: `fact_moat_annual.roic_spread_f` — current-year ROIC minus WACC, winsorized and industry-z-scored to [0,1]
- **Equation**: $\text{ROICSpread}_{i,t} = \text{clip}_{[0,1]}\Big(z_{industry}(\text{ROIC}_{i,t} - \text{WACC}_{i,t})\Big)$, where ROIC = NOPAT/InvestedCapital
- **Hypothesis class / expected sign**: `risk_premium`, positive
- **Lifecycle**: `experimental`
- **IC-IR @21d**: not itemized
- **MCPT (pre-fix)**: not itemized
- **t-stat (naive)**: `PENDING`
- **t-stat (HAC, Newey–West)**: `PENDING`
- **n_obs / cadence**: annual — short-T caveat applies
- **Notes**: —

4.7.3. `gross_profitability`

- **Style**: `quality`
- **Source**: `fact_moat_annual.gross_profitability_f` — `gross_profit / total_assets` (Novy-Marx 2013)
- **Equation**: $GP_{i,t} = \dfrac{\text{GrossProfit}_{i,t}}{\text{TotalAssets}_{i,t}}$
- **Hypothesis class / expected sign**: `risk_premium`, positive
- **Lifecycle**: `experimental`
- **IC-IR @21d**: not itemized
- **MCPT (pre-fix)**: not itemized
- **t-stat (naive)**: `PENDING`
- **t-stat (HAC, Newey–West)**: `PENDING`
- **n_obs / cadence**: annual — short-T caveat applies
- **Notes**: —

4.7.4. piotroski

- **Style**: quality
- **Source**: fact_fundamental_annual.piotroski_f — 9-criterion financial-strength score
- **Equation**: $F_{i,t} = \sum_{k=1}^{9} \mathbb{1}[\text{criterion}_k \text{ met}] \in [0, 9]$
- **Hypothesis class / expected sign**: behavioral, positive
- **Lifecycle**: experimental
- **IC-IR @21d**: sign-flip — negative at h=1, positive at h 21
- **MCPT (pre-fix)**: not itemized
- **t-stat (naive)**: PENDING
- **t-stat (HAC, Newey–West)**: PENDING
- **n_obs / cadence**: annual
- **Notes**: —

4.8. 6. Investment / Accruals

Firms that invest conservatively and have low accruals out-earn aggressive investors — over-investment and earnings-quality mispricing. *Sloan 1996 [#83], Cooper–Gulen–Schill 2008 [#103], Fama–French 2015 [#56].*

4.8.1. accruals

- **Style**: quality
- **Source**: fact_moat_annual.accruals_f — Sloan (1996) Total-Accruals-to-Total-Assets
- **Equation**: $\text{Accruals}_{i,t} = \dfrac{\text{NetIncome}_{i,t} - \text{CFO}_{i,t}}{\text{TotalAssets}_{i,t}}$

- **Hypothesis class / expected sign**: `behavioral`, negative
- **Lifecycle**: `experimental`
- **IC-IR @21d**: not itemized
- **MCPT (pre-fix)**: not itemized
- **t-stat (naive)**: `PENDING`
- **t-stat (HAC, Newey–West)**: `PENDING`
- **n_obs / cadence**: annual — short-T caveat applies
- **Notes**: §10's original mapping also names `earnings_growth_qoq` for this family; no config file for that id was found under this exact name — needs reconciling against whatever earnings-growth id is actually registered.

4.9. 7. Low volatility / Defensive

Low-risk stocks earn higher risk-adjusted returns — leverage constraints and lottery-preference bid up high-vol names (the low-vol anomaly / BAB). *Ang et al. 2006 [#108], Frazzini–Pedersen 2014 [#93], Novy-Marx 2014 [#10].*

4.9.1. `volatility_30d`

- **Style**: `volatility`
- **Source**: `fact_eod.volatility_30d_f` — STDDEV of daily log returns over 30 sessions $\times \sqrt{252}$
- **Equation**: $\sigma_{i,t}^{30d} = \text{STDDEV}_{30d}(\ln(P_{i,\tau}/P_{i,\tau-1})) \times \sqrt{252}$
- **Hypothesis class / expected sign**: `behavioral`, negative
- **Lifecycle**: `experimental`; strong signal — invert sign on promotion
- **IC-IR @21d**: -0.5 to -0.7, alongside `volatility_60/126/252d` — part of the low-vol/size inverted cluster

- **MCPT (pre-fix)**: not itemized
- **t-stat (naive)**: `PENDING`
- **t-stat (HAC, Newey–West)**: `PENDING`
- **n_obs / cadence**: daily
- **Notes**: —

4.10. 8. Liquidity / Illiquidity

Illiquid stocks (high Amihud price-impact, low $ADV) require a return premium for the cost and risk of trading them. *Amihud 2002 [#48].*

4.10.1. `amihud`

- **Style**: `liquidity`
- **Source**: `fact_eod.amihud_f` — mean of `|daily_return| / dollar_volume` over a trailing window
- **Equation**: $\text{Amihud}_{i,t} = \frac{1}{N} \sum_{\tau=t-N+1}^{t} \frac{|r_{i,\tau}|}{\$\text{Vol}_{i,\tau}}$
- **Hypothesis class / expected sign**: not set in config
- **Lifecycle**: `experimental`
- **IC-IR @21d**: inverted group (with `pe_ratio`, `volatility_*`, `market_cap`)
- **MCPT (pre-fix)**: not itemized
- **t-stat (naive)**: `PENDING`
- **t-stat (HAC, Newey–West)**: `PENDING`
- **n_obs / cadence**: daily; diagnostics “healthy” quadrant
- **Notes**: —

4.10.2. `adv_dollar_20d`

- **Style**: `liquidity`
- **Source**: `fact_eod.adv_dollar_20d_f` — mean of `adj_close × volume` over the trailing 20 sessions
- **Equation**: $\text{ADV\$}_{i,t}^{20d} = \frac{1}{20} \sum_{\tau=t-19}^{t} P_{i,\tau} \times V_{i,\tau}$
- **Hypothesis class / expected sign**: not set in config
- **Lifecycle**: `experimental`
- **IC-IR @21d**: not itemized
- **MCPT (pre-fix)**: not itemized
- **t-stat (naive)**: `PENDING`
- **t-stat (HAC, Newey–West)**: `PENDING`
- **n_obs / cadence**: daily
- **Notes**: don't confuse with `adv_20d` (undiscounted volume, no dollar scaling) — that id sits in the "drift" diagnostics quadrant (non-stationary, high entropy), a distinct factor.

4.11. 9. Short interest / Positioning

Heavily-shorted / high-days-to-cover names underperform — informed short sellers *and* short-sale-constraint overvaluation (divergence of opinion). *Boehmer–Jones–Zhang 2008 [#101], Rapach–Ringgenberg–Zhou 2016 [#100], Miller 1977 [#111].*

4.11.1. `days_to_cover`

- **Style**: `positioning`

- **Source**: `fact_short_interest.days_to_cover_f` — FINRA's `daysToCoverQuantity`, trusted verbatim from the vendor (no platform recompute from ADV)
- **Equation**: $\mathrm{DTC}_{i,t} = \dfrac{\mathrm{ShortInterest}_{i,t}}{\mathrm{ADV}_{i,t}}$
- **Hypothesis class / expected sign**: `behavioral`, negative
- **Lifecycle**: `experimental`
- **IC-IR @21d**: not itemized
- **MCPT (pre-fix)**: not itemized
- **t-stat (naive)**: `PENDING`
- **t-stat (HAC, Newey–West)**: `PENDING`
- **n_obs / cadence**: FINRA disclosure-lag snap applies (18 calendar days)
- **Notes**: —

4.11.2. `short_pct_float`

- **Style**: `positioning`
- **Source**: `fact_short_interest.short_pct_float_f` — short interest / diluted shares outstanding (a float proxy)
- **Equation**: $\mathrm{ShortPctFloat}_{i,t} = \dfrac{\mathrm{ShortInterest}_{i,t}}{\mathrm{DilutedSharesOut}_{i,t}}$
- **Hypothesis class / expected sign**: `behavioral`, negative
- **Lifecycle**: `experimental`
- **IC-IR @21d**: not itemized
- **MCPT (pre-fix)**: not itemized
- **t-stat (naive)**: `PENDING`
- **t-stat (HAC, Newey–West)**: `PENDING`
- **n_obs / cadence**: FINRA disclosure-lag snap applies (18 calendar days)
- **Notes**: —

4.12. 10. Seasonality

Same-calendar-month historical returns recur — persistent cross-sectional seasonalities from mood, liquidity, and information cycles. *Heston–Sadka 2008 [#37], Hirshleifer et al. 2020 [#40].*

4.12.1. `seasonality_same_month`

- **Style**: `momentum`
- **Source**: `fact_eod.seasonality_same_month_f` — mean, over the prior 20 years (5 required), of the instrument's monthly return in the same calendar month (Heston–Sadka

  2008)

- **Equation**: $\text{Seas}_{i,t} = \frac{1}{K}\sum_{y=1}^{K} r^{month}_{i,\,\text{same-month}(t),\,y}$, $5 \le K \le 20$
- **Hypothesis class / expected sign**: `behavioral`, positive
- **Lifecycle**: `experimental`
- **IC-IR @21d**: not itemized
- **MCPT (pre-fix)**: not itemized
- **t-stat (naive)**: `PENDING`
- **t-stat (HAC, Newey–West)**: `PENDING`
- **n_obs / cadence**: daily
- **Notes**: —

4.13. 11. Earnings momentum / PEAD

Prices under-react to earnings surprises (SUE), drifting for weeks after the announcement — the post-earnings-announcement drift. *Bernard–*

Thomas 1989 [#75].

4.13.1. `sue`

- **Style**: `growth`
- **Source**: `fact_earnings_momentum.sue_f` — `eps_surprise_pct / STDDEV(eps_surprise_pct)` over the strictly-prior 8 announcements
- **Equation**: $\text{SUE}_{i,q} = \dfrac{\Delta EPS\%_{i,q}}{\text{STDDEV}_8(\Delta EPS\%_{i,\,q-1:q-8})}$
- **Hypothesis class / expected sign**: `behavioral`, positive
- **Lifecycle**: `experimental`
- **IC-IR @21d**: not itemized
- **MCPT (pre-fix)**: not itemized
- **t-stat (naive)**: `PENDING`
- **t-stat (HAC, Newey–West)**: `PENDING`
- **n_obs / cadence**: not itemized
- **Notes**: traded by the `sue-earnings-momentum` strategy — see Strategy Catalog §5.

4.14. 12. Time-series momentum / Trend

An asset's *own* past 12-1 return predicts its next-month return across asset classes — trend-following / slow-diffusion premium. *Moskowitz–Ooi–Pedersen 2012 [#115], Hurst–Ooi–Pedersen 2017 [#95].*

4.14.1. `tsmom_12_1_vol_scaled`

- **Style**: `momentum`
- **Source**: `fact_eod.momentum_12_1_vol_scaled_f` — same source column as `momentum_12_1_vol_scaled` (§4), but registered as its own factor entity because it has its own evaluation path: `funnel_exempt: true` in config, so it is never scored by the cross-sectional IC/MCPT/Alphalens funnel. Its only evidence comes from the strategy that trades it — see Strategy Catalog §3 for the `long_flat_trend` mechanic (own-trend-sign selection) and its walk-forward backtest.
- **Equation**: $\text{TSMom}_{i,t} = \text{Mom}^{vol}_{i,t}$ — identical formula to §4's `momentum_12_1_vol_scaled` (same source column); the `long_flat_trend` mechanic consumes $\text{sign}(\text{TSMom}_{i,t})$, not its cross-sectional rank
- **Hypothesis class / expected sign**: `risk_premium`, positive
- **Lifecycle**: `experimental`
- **IC-IR @21d**: not itemized (funnel-exempt — not scored by cross-sectional IC)
- **MCPT (pre-fix)**: not itemized
- **t-stat (naive)**: `PENDING`
- **t-stat (HAC, Newey–West)**: `PENDING`
- **n_obs / cadence**: daily
- **Notes**: —

4.15. 13. Short-horizon reversal

Very-short-horizon losers bounce (and winners fade) — liquidity provision / over-reaction correction (contrarian). *Lehmann 1990 [#96].*

4.15.1. `bb_pct`

- **Style**: `volatility`
- **Source**: `fact_eod.bb_pct_f` — `(adj_close - bb_lower_20_f) / (bb_upper_20_f - bb_lower_20_f)`
- **Equation**: $\%B_{i,t} = \dfrac{P_{i,t} - L^{20}_{i,t}}{U^{20}_{i,t} - L^{20}_{i,t}}$, where $U^{20} = \mathrm{SMA}^{20d} + 2\sigma^{20d}$ and $L^{20} = \mathrm{SMA}^{20d} - 2\sigma^{20d}$ (0 at the lower band, 1 at the upper band) — a ratio of price differences, so a future split scales numerator and denominator identically and cancels (scale-free, PIT-clean w.r.t. corporate actions; F-289/LH-25)
- **Hypothesis class / expected sign**: `behavioral`, negative
- **Lifecycle**: `experimental`
- **IC-IR @21d**: not itemized
- **MCPT (pre-fix)**: not itemized
- **t-stat (naive)**: `PENDING`
- **t-stat (HAC, Newey–West)**: `PENDING`
- **n_obs / cadence**: daily; diagnostics "healthy" quadrant
- **Notes**: —

4.15.2. `bb_lower_20` *(superseded predecessor — not in the original 13-row table, included for the correction it forces)*

- **Style**: `volatility`
- **Source**: `fact_eod.bb_lower_20_f` — 20-day SMA of `adj_close` − 2×20-day stddev
- **Equation**: $L^{20}_{i,t} = \mathrm{SMA}^{20d}_{i,t} - 2\,\sigma^{20d}_{i,t}$ — a raw price level (not scale-free), which is precisely the CF-1 look-ahead defect that got it deprecated (see Lifecycle below)
- **Hypothesis class / expected sign**: `behavioral`, positive — short-horizon mean-reversion (Lehmann 1990 contrarian reversal)

- **Lifecycle**: `deprecated for cause` (F-289/LH-25, 2026-07-20, run `260718_31a51e`) — raw price-level source column with a CF-1 look-ahead defect. Replaced by `bb_pct` (scale-free).
- **IC-IR @21d**: not itemized
- **MCPT (pre-fix, run `260526_6ddcd9`)**: one of only four factors with a non-collapsed permutation null and positive `unbiased_ic_ir` (0.21–1.35) at the time — **this result is now known to be contaminated by the CF-1 look-ahead**, not evidence of a real edge. The other three factors in that same "real significance" group at the time — `ma_50d`, `ma_200d`, `vwap_20d` — were deprecated for the identical reason on the same date and are not part of this catalog's 13 families.
- **t-stat (naive)**: `PENDING` (moot — factor is deprecated)
- **t-stat (HAC, Newey–West)**: `PENDING` (moot — factor is deprecated)
- **n_obs / cadence**: daily
- **Notes**: kept in the catalog specifically as a cautionary entry — the `260526_6ddcd9` run synthesis called this factor's MCPT result the cleanest in the library; two months later it was found to be a look-ahead artifact. A clean permutation p-value is necessary evidence, not sufficient.

4.16. Next steps

1. Write the query/script computing `t_naive` and `t_HAC` per `factor_id` against `ops.risk_factor_return`, and fill in every `PENDING` field above.
2. ~~Resolve the two duplicate pairs~~ — done 2026-08-05: `market_beta_252d` and `momentum_12m_1m` are the canonical survivors; `beta` and `momentum_12_1` are retired. Remaining follow-up: if either legacy

id has rows in `ops.factor`, flip their status there too (a DB change, out of scope for this doc pass).
3. Backfill `hypothesis_class`/`expected_sign` for the factors missing it in config (`market_beta_252d`, `market_cap`, `amihud`, `adv_dollar_20d`).
4. Reconcile the §6 (Investment/Accruals) mapping — `earnings_growth_qoq` has no matching config file under that id.
5. Regenerate the empirical fields against the first post-F-136/137/138 `NIGHTLY_FULL`/`RESEARCH_DAY` run.
6. Consider formally adding `residual_momentum_252d_eod` (§4) as a 14th citation row, given its result is the strongest in the run.

4.17. Cross-links

- strategy-catalog.md — the downstream companion to this chapter: every registered `(factor_id, mechanic, params)` strategy that trades a factor listed above, with its universe, sizing, promotion gates, and backtest status.
- factor-models-and-benchmarks.md — the theory this catalog assumes: factor loadings, factor returns, CAPM decomposition, systematic/idiosyncratic return, benchmark selection.
- experiments/2026-08-03-factor-catalogue-empirical-results — the dated provenance record for the empirical fields above (run id, method, journal link).
- research-platform/04-factor-layer-as-multiplex-layer-1.md — why this catalog matters to the PhD proposal: these factors are its Layer 1.
- research-platform/08-validation-and-anti-overfitting.md — why no single column above (including the t-stats, once filled in) should be read as a standalone verdict.

5. Strategy Catalog

A comprehensive, self-contained reference covering every registered strategy in the platform — the downstream companion to the Factor Catalog. Where that chapter documents *factors* (measured, predictive characteristics of an instrument), this chapter documents *strategies*: the portfolio-construction mechanics that trade them, organized by the factor family they draw on.

Scope note — strategies, not factors. Per the platform's canonical model (see Factor & Strategy Lifecycle), a strategy is a `(factor_id, mechanic, params)` triple: it always names exactly one underlying factor, plus a *mechanic* — the reusable selection/weighting algorithm that turns the factor's value into positions (e.g. rank the cross-section and go long/short the extremes, or just follow each instrument's own sign) — plus the sizing/universe/rebalance parameters around it. A strategy cannot exist without a factor beneath it; the dependency runs factor → strategy, never the reverse. Strategy promotion (`incubating → paper → live`, F-054) additionally requires the underlying factor's `ops.factor` status to already be `validated`.

5.1. Provenance

- **Config fields** (`factor_id`, `mechanic`, `params`, universe, benchmark, promotion gates, `enabled_in_nightly` / `research_day_backtest`) are pulled directly from `config/strategies/*.yaml` in SBFoundation as of 2026-08-05 — eight strategies, all `kind: factor` (every

registered strategy today is factor-driven; no other strategy `kind` exists yet).

- **Empirical fields** (backtest CAGR/Sharpe/drawdown, PBO, DSR, strategy-level MCPT) have no equivalent to the Factor Catalog's single dated run synthesis — there is no one diagnostic run that scored all eight strategies at once. Each strategy's numbers live in its own backtest sidecar (surfaced on the Strategies hub, `#/run/<run_id>/pipeline/strategies`) and, where a strategy has a dedicated findings write-up, that document is the authoritative source; both are cited per entry below. Everywhere else, empirical fields are marked `PENDING` rather than guessed — see Method.
- **None of the eight are currently live-submitting paper orders.** `config/execution.yaml`'s `experiment_portfolio_ids` — the list that binds a strategy's construction output to the paper order path — is empty. All eight are evaluated purely by nightly/research-day backtest + promotion-gate comparison against their `sharpe_min`/`max_drawdown_max` thresholds. This is a distinct, earlier-stage track from the ML signal book's live paper execution (F-282, first live fills 2026-07-21) — that book trades a separate predictor ensemble, not one of these factor-strategy triples.

5.2. Method: where the backtest evidence lives

Each strategy entry below has empirical fields that are frequently `PENDING`. All are sourced the same way, stated once here rather than repeated eight times:

- **Backtest run** — `sbbacktest` drives a Zipline subprocess per strategy (nightly for `enabled_in_nightly: true` strategies;

research-day-only for `research_day_backtest: true` strategies), producing a NAV series, tearsheet, and cost-drag sidecar (`sbbacktest.sidecar_builder`). CAGR/Sharpe/MaxDD/cost-drag come from there.

- **Overfitting controls** — `sbpbo` (Probability of Backtest Overfitting / CSCV) and strategy-level MCPT (`compute_strategy_mcpt`, research-day-only, `pbo_splits` / `mcpt_permutations` from the strategy's own `research:` block) are the anti-overfitting layer referenced by research-platform/08-validation-and-anti-overfitting.md — the same epistemic caveat as the Factor Catalog's t-stat columns applies here: a clean Sharpe is necessary evidence, not sufficient.
- **The actual promotion bar** is the `promotion_gates` block (`min_elapsed_days`, `min_green_nightlies`, `sharpe_min`, `max_drawdown_max`), evaluated against `ops.strategy` by the F-054 lifecycle machinery — not any single backtest metric in isolation.
- **Where to look**: the Strategies hub (`#/run/<run_id>/pipeline/strategies`, CLAUDE.md §18) surfaces the current run's tearsheet, health timeline, and (for the six cross-asset/momentum strategies below) benchmark comparison for each strategy.

5.3. 1. Cross-sectional equity momentum

The platform's primary momentum family — four strategies sharing one factor (`momentum_12_1_vol_scaled`, Factor Catalog §4) and one universe (`us_large_cap`), each isolating a different construction choice: quantile long/short, long-only, beta-hedged dollar-neutral, or a fixed external yardstick. *Jegadeesh–Titman 1993 [#49].*

5.3.1. `classic-momentum`

- **Underlying factor**: `momentum_12_1_vol_scaled`
- **Mechanic**: `long_short_quantiles` — long the top decile, short the bottom decile of the cross-section, ranked monthly
- **Universe**: `us_large_cap` — 500 names, NYSE/NASDAQ, $1B market cap, $5 price, ranked by 20-day dollar ADV
- **Params**: 50 long / 50 short, monthly rebalance, 21-day hold, $500k capital, `max_leverage: 2.2`, vol-managed to a 10% portfolio-vol target (RP-1)
- **Benchmark**: SPY (+ 17 more — QQQ/IWM/DIA/MTUM, VIXY/TLT/HYG, 10 sector SPDRs)
- **Lifecycle**: `enabled_in_nightly: true`. Promotion gate: `sharpe_min 1.1`, `max_drawdown_max -0.20`, 30 elapsed days, 5 green nightlies.
- **Notes**: `factor_id` was corrected (TASK-1811, 2026-06-26) from `momentum_12m_1m` to `momentum_12_1_vol_scaled` so the declared factor honestly names the column the backtest actually trades — the traded signal was unchanged, only the label. One consequence: the underlying factor's `ops.factor` status is still `experimental` (not `validated`), so this strategy is knowingly trading a pre-validation factor for research purposes, and the Strategies SPA card renders that status honestly rather than a validated factor it doesn't trade.

5.3.2. `classic-momentum-long-only`

- **Underlying factor**: `momentum_12_1_vol_scaled` (same correction/caveat as above)
- **Mechanic**: `long_only_top_n` — long the top decile, no short leg
- **Universe**: `us_large_cap` (identical to `classic-momentum`)
- **Params**: 50 long / 0 short, monthly rebalance, 21-day hold, $500k capital, `max_leverage: 2.2`, `max_position_size_pct: 0.20`, vol-managed to 10%

- **Benchmark**: SPY, QQQ, IWM
- **Lifecycle**: `enabled_in_nightly: true`; `seed: true` — F-245/LH-21 seeds this YAML as the auto-spawned incubating stub once `momentum_12_1_vol_scaled` validates (the production `STRATEGY_CASCADE_ENFORCE` switch itself stays OFF). Same promotion gate as `classic-momentum`.
- **Notes**: the net-long baseline (F-207/TASK-1656, DRAFT-017) — deliberately the directional sibling of `classic-momentum-dollar-neutral` on the identical factor, so comparing the two isolates whether the edge is cross-sectional (survives beta-hedging) or merely directional (mostly market beta).

5.3.3. `classic-momentum-dollar-neutral`

- **Underlying factor**: `momentum_12_1_vol_scaled` (same correction/caveat as above)
- **Mechanic**: `dollar_neutral_rank` — long the top decile / short the bottom decile, rank-weighted to net 0 exposure
- **Universe**: `us_large_cap` (identical to `classic-momentum`)
- **Params**: 50 long / 50 short, monthly rebalance, 21-day hold, $500k capital, `max_leverage: 2.2`, vol-managed to 10%
- **Benchmark**: SPY, QQQ, IWM
- **Lifecycle**: `enabled_in_nightly: true`. Same promotion gate as `classic-momentum`.
- **Notes**: the market-beta-hedged sibling in the DRAFT-017 three-way comparison — run to isolate cross-sectional alpha from directional market exposure, independent of `classic-momentum-long-only`'s net-long read on the same question.

5.3.4. `clinic06-benchmark`

- **Underlying factor**: `momentum_12_1_vol_scaled`

- **Mechanic**: `long_only_top_n` — long the top 20, equal-weight, no shorts
- **Universe**: `us_large_cap` (500 names, same filters as `classic-momentum`)
- **Params**: 20 long / 0 short, monthly rebalance, 21-day hold, $1M capital, `max_leverage: 1.1`, `max_position_size_pct: 0.60`
- **Benchmark**: SPY, QQQ, IWM
- **Lifecycle**: `enabled_in_nightly: true`, but **not a promotion candidate** — a fixed external yardstick, not traded.
- **Empirical (F-329/IR-14, TASK-2853, populated 2026-08-02)** — dual-cost comparison over 2016-01-04→2025-12-30 (2,513 daily NAV rows), from the internal `docs/research/clinic06_benchmark_findings.m` findings note:

Regime	CAGR	Sharpe	MaxDD	Cost drag (bps)
TC-2 (platform-default, per-name Corwin–Schultz spread)	18.04%	0.70	−43.77%	503.57
Fixed (Clinic-faithful flat 1% spread)	16.42%	0.65	−44.07%	613.98

- **Notes**: a platform-native replica of the QSResearch Clinic-06 long-only top-20 vol-scaled-momentum book — not a new alpha source. Six deliberate structural divergences from the original

harness (survivorship-free universe, dollar- vs share-volume ranking, dropped magnitude threshold, TC-2 vs flat-1% cost, etc.) are catalogued in the findings doc; each is the *measurement of a platform effect*, not noise. Its purpose is separating "platform effect" from "strategy edge" for every other strategy in this catalog, not generating a tradeable signal of its own.

5.4. 2. Sector rotation

Same factor and mechanic as the `us_large_cap` family above, applied to a much thinner, single-asset-class cross-section — the 11 SPDR sector ETFs rather than 500 individual names.

5.4.1. `etf-sector-momentum`

- **Underlying factor**: `momentum_12_1_vol_scaled`
- **Mechanic**: `long_only_top_n` — long every ETF in the panel ranked into the top tier (`long_q: 11` = the full 11-name panel), equal-weight
- **Universe**: `us_etf_sector_rotation` — the 11 SPDR sector ETFs (XLK/XLF/XLV/XLE/XLI/ XLC/XLY/XLP/XLB/XLRE/XLU)
- **Params**: 11 long / 0 short, monthly rebalance, 21-day hold, $1M capital, `max_leverage: 1.1`, `max_position_size_pct: 0.30`
- **Benchmark**: SPY + the 11 sector ETFs
- **Lifecycle**: `enabled_in_nightly: false` — backtest-only, paper-off. Requires `ALLOW_ETFS=ON` for the upstream allowlist to admit the sector panel. Promotion gate is looser than the equity family: `sharpe_min 0.5`, `max_drawdown_max -0.30`.

- **Notes**: the F-247 ETF Universe Extension MVP demo — proves the same cross-sectional-momentum mechanic works on a thin, single-asset-class cross-section once ETFs are admitted at all, ahead of any nightly enablement decision.

5.5. 3. Time-series momentum / Trend

The platform's own-trend-sign complement to cross-sectional momentum: instead of ranking names against each other, each instrument is judged only against its own history — long if trending up, flat otherwise. Works on cross-asset panels too thin for a meaningful cross-sectional rank. *Moskowitz–Ooi–Pedersen 2012 [#115], Hurst–Ooi–Pedersen 2017 [#95].*

5.5.1. `etf-trend`

- **Underlying factor**: `tsmom_12_1_vol_scaled` — funnel-exempt; this strategy's own backtest *is* the factor's only evidence (see Factor Catalog §12)
- **Mechanic**: `long_flat_trend` — long every ETF whose own vol-scaled 12-1 trend is positive, flat otherwise (no shorts, no cross-sectional comparison)
- **Universe**: `us_multi_asset_etf_trend` — a ~14-name cross-asset panel, one liquid long-history instrument per asset class (SPY/QQQ/IWM equity, EFA/EEM international, TLT/IEF/LQD/HYG rates & credit, GLD gold, DBC commodities, UUP dollar, VNQ REIT, TIP inflation-linked), pinned explicitly so a venue filter can't drift the panel

- **Params**: long_q 14 (the full panel cap), monthly rebalance, 21-day hold, $1M capital, `max_leverage: 2.2`, vol-managed to a 10% portfolio-vol target
- **Benchmark**: SPY, AGG
- **Lifecycle**: `enabled_in_nightly: false`, `research_day_backtest: true` (opts into the research-day-only backtest fan-out so its tearsheet surfaces on the Strategies hub without going nightly). Backtest-only, paper-off. Requires `ALLOW_ETFS=ON`. Promotion gate: `sharpe_min 0.5`, `max_drawdown_max -0.25`.
- **Notes**: the platform's first per-asset trend sleeve (F-283/R6-ETF slice) — the highest-value diversifier available to a US-equity book per Hurst–Ooi–Pedersen 2017. Per-asset inverse-vol weighting (rather than today's equal-weight trend-positive names) is a deferred follow-on (PM-2).

5.6. 4. Residual (market-neutral) momentum

The beta-hedged sibling of ordinary cross-sectional momentum: strips out market beta *before* ranking, via a rolling regression against an equal-weighted benchmark. *Blitz–Huij–Martens 2011 [#50].*

5.6.1. `residual-momentum`

- **Underlying factor**: `residual_momentum_252d_eod` — the strongest IC-IR result in the diagnostic run synthesis (+0.724, Factor Catalog §4); note this is the atomic id, not the *deprecated* composite `residual_momentum_252d` (no `FactorCompositionService` exists to compute that recipe)

- **Mechanic**: `long_short_quantiles` — long top decile / short bottom decile of the 252-day residual-return z-score
- **Universe**: `us_large_cap`
- **Params**: 50 long / 50 short, monthly rebalance, 21-day hold, $1M capital
- **Benchmark**: SPY
- **Lifecycle**: `enabled_in_nightly: false`, `research_day_backtest: true`. Promotion gate: `sharpe_min 1.1`, `max_drawdown_max -0.20` (same bar as the equity momentum family).
- **Notes**: despite trading the run's single strongest factor result, the strategy itself is still research-day-only, not nightly-tracked — factor-level IC-IR and strategy-level backtest promotion are evaluated on separate cadences and neither implies the other has cleared its own gate.

5.7. 5. Earnings momentum / PEAD

Trades the drift that follows a standardized earnings surprise — prices under-react to the news and continue drifting toward it for weeks. *Bernard–Thomas 1989 [#75], Chan–Jegadeesh–Lakonishok 1996 [#79].*

5.7.1. `sue-earnings-momentum`

- **Underlying factor**: `sue` (Standardized Unexpected Earnings, Factor Catalog §11)
- **Mechanic**: `long_short_quantiles` — long top decile / short bottom decile of SUE, the standard PEAD convention
- **Universe**: `us_large_cap`

- **Params**: 50 long / 50 short, monthly rebalance, 21-day hold, $1M capital
- **Benchmark**: SPY
- **Lifecycle**: `enabled_in_nightly: false` by explicit design (config comment: "start dormant; flip once validated in a Zipline backtest"), `research_day_backtest: true`. Same promotion gate as `residual-momentum` (`sharpe_min 1.1`, `max_drawdown_max -0.20`).
- **Notes**: the backtest wiring for this strategy was broken (a Friday `NIGHTLY_FULL` exit-2 crash) and repaired 2026-08-01 (B-279.1) — the fix spanned four drift-prone layers, so a re-break here is a plausible failure mode to check first if this strategy's research-day backtest goes red again.

5.8. Next steps

1. Compile a per-strategy empirical table analogous to the Factor Catalog's diagnostic-run synthesis — today only `clinic06-benchmark` has a dedicated, dated findings write-up; the other seven have `PENDING` CAGR/Sharpe/MaxDD/PBO/DSR fields above.
2. Once `residual_momentum_252d_eod` and `sue` progress toward `validated` in `ops.factor`, revisit whether `residual-momentum` and `sue-earnings-momentum` should move from `research_day_backtest` to `enabled_in_nightly`.
3. Reconcile each strategy's `ops.strategy` lifecycle state (`incubating`/`paper`/`live`, F-054) against this chapter — the config-level `enabled_in_nightly`/`research_day_backtest` flags shown here are readiness signals, not the lifecycle state of record.

4. Add strategy-level MCPT/PBO figures once `compute_strategy_mcpt` has run against the full eight-strategy set in one research-day pass, so results are directly comparable.

5.9. Cross-links

- factor-catalog.md — the upstream companion to this chapter: every factor traded by a strategy above, with its economic rationale, style, and (where available) cross-sectional IC-IR/MCPT evidence.
- factor-models-and-benchmarks.md — the factor-return theory (loadings, CAPM decomposition, benchmark selection) that strategy-level beta attribution and benchmark comparison assume.
- Factor & Strategy Lifecycle — the canonical platform doc for the `incubating → paper → live` promotion state machine these strategies move through.
- Portfolio Construction — how a promoted strategy's target weights become orders (construction algorithms, compliance, cost model).
- research-platform/08-validation-and-anti-overfitting.md — why no single backtest metric above (Sharpe included) should be read as a standalone verdict.

6. Factor Models, the CAPM Decomposition, and Market Benchmarks

A PhD-level reference note, grounded in the SBFoundation platform.

- **Version:** 1.0 · **Created:** 2026-08-02
- **Scope:** factor loadings; factor returns; the CAPM regression `r = + ·market +` ; systematic vs. idiosyncratic return; how factors become buy/sell signals; how a market benchmark is chosen; academic literature on benchmark construction; data sources.
- **Grounding:** every concept is tied to the module, table, or config that implements it in this repository. Code references use `path:line`. Academic citations resolve to `docs/reference-papers/README.md` (row numbers in brackets, e.g. [**#116**]).

  Reading guide. §1–§4 are the theory (loadings, returns, CAPM, systematic/idiosyncratic). §5–§6 connect factors to tradable signals and give a worked example. §7–§9 cover benchmark choice, the literature, and data provenance. §10 is the canonical factor catalogue. A recurring **"In this platform"** callout maps each idea to real code.

6.1. 0. Two regression geometries (read this first)

Almost every confusion about factor models dissolves once you separate the **two orthogonal regression geometries** the field uses. SBFoundation implements both, in different packages, and keeps them deliberately distinct.

	Time-series regression	**Cross-sectional regression**
Runs *over*	dates, for one asset (or one portfolio)	assets, for one date
Estimates	**loadings/betas** (asset's sensitivity to a factor's *return*) and **alpha**	**factor returns** `F` (the period's payoff to a unit of exposure)
Factor is	an observable **return series** (e.g. the market excess return, HML)	a **characteristic** (e.g. book-to-market, momentum score)
Canonical estimator	OLS with HAC errors — CAPM, Fama-French time-series tests	Fama–MacBeth two-pass [**#122**] / Barra WLS
In this platform	`sbattribution` (return-based to SPY/MTUM/FF5)	`sbriskmodel` (daily WLS characteristic regression)

The words *loading*, *exposure*, and *beta* all name the coefficient that multiplies a factor. Which geometry produced it determines what it means: a **time-series** is a slope on a factor *return*; a **cross-sectional exposure** is a standardized *characteristic* that the cross-sectional regression will later price.

6.2. 1. Factor loadings

6.2.1. 1.1 Definition

A **factor loading** (equivalently **exposure**, **beta**) is the sensitivity of an asset's return to a common factor. In the linear factor model

$$r_{i,t} \;=\; \alpha_i \;+\; \sum_{k=1}^{K} \beta_{i,k}\, f_{k,t} \;+\; \varepsilon_{i,t},$$

`_{i,k}` is asset i's loading on factor k. `f_{k,t}` is the factor's realization at t, and `_{i,t}` is the asset-specific residual. Loadings are the bridge between an asset and the systematic forces that move it.

There are two operational definitions, matching the two geometries in §0:

(a) Time-series loading (a slope on a factor *return*). Regress the asset's realized returns on the factor's realized return series over a rolling window. The market beta is the archetype:

$$\beta_i \;=\; \frac{\mathrm{Cov}(r_i,\, r_m)}{\mathrm{Var}(r_m)}.$$

(b) Cross-sectional exposure (a standardized *characteristic*). Take a raw characteristic — book-to-market, 12-1 momentum, log market cap — and standardize it across the cross-section on each date so exposures are comparable across factors and time. This is the Barra convention.

6.2.2. 1.2 How loadings are calculated

Time-series market beta is computed directly from the covariance/variance identity over a trailing window. SBFoundation computes a **252-trading-day rolling market beta** against SPY, persisted as `beta_f` in `gold.fact_eod`. Because DuckDB has no windowed `REGR_SLOPE`, it evaluates the identity with windowed aggregates (`src/sbfactors/eod_feature_service.py:364-431`):

```
COVAR_SAMP(stock_return, spy_return) OVER (
    PARTITION BY instrument_sk ORDER BY date_sk
    ROWS BETWEEN 251 PRECEDING AND CURRENT ROW
) / NULLIF(
    VAR_SAMP(spy_return) OVER (
        PARTITION BY instrument_sk ORDER BY date_sk
        ROWS BETWEEN 251 PRECEDING AND CURRENT ROW), 0
) AS beta_f
```

Returns are log returns `ln(adj_close / lag(adj_close))`; SPY is joined cross-instrument by `date_sk`. This is exactly estimator (a): a rolling `Cov(r_i, r_SPY)/Var(r_SPY)`.

Cross-sectional exposures are standardized characteristics. The production risk model (`sbriskmodel.FactorExposureService`, `src/sbriskmodel/factor_exposure_service.py:347-396`) builds the exposure matrix $\mathbf{X}$ each day by:

1. selecting the investable cross-section (require sector + all style sources non-null);
2. applying a per-factor transform (`identity | log | negate`);
3. **robust z-scoring** across the daily cross-section — `z = (x − median) / (1.4826 · MAD)` (`_robust_zscore`, `:502`), which is outlier-resistant relative to a mean/ z-score;
4. **winsorizing** to ±cap (`:515`);

5. appending **sector one-hot dummies** (with `Industrials` dropped as the baseline), holding the column count stable at $\mathbf{K = 20}$ (10 style + 10 non-baseline sector columns) so the downstream factor covariance sees a constant dimension day-to-day.

The result is written to `ops.risk_factor_exposure` (raw / z-scored / winsorized), keyed `(date, instrument_sk, factor_id)`. Sign conventions are folded into the transform so that a *positive* exposure always means *more of the priced attribute*: low-vol = −volatility, size = −log(market_cap), liquidity = −log(ADV$), value = −fcf_yield (`sbfactorrisk/factor_risk.py:703-714`).

> **In this platform.** Two exposure surfaces coexist. The **point-in-time B-matrix** in `sbriskmodel.factor_exposure_service` → `ops.risk_factor_exposure` is the load-bearing one (it feeds the daily factor-return regression). A **lighter cross-sectional z-score cube** in `sbfactorrisk.factor_risk._build_exposure_cube` (`factor_risk.py:725-750`) is used only for the exposure-spread / stress report. The time-series `beta_f` is a *third* construct — a market loading, not a Barra style exposure — which the risk model then re-imports as the `market_beta` style (`factor_risk.py:706`).

6.2.3. 1.3 Point-in-time discipline

Loadings must be knowable at the moment they are used. The risk model regresses **yesterday's** exposures `X_{t-1}` on **today's** return `r_t` (§2.2), annual fundamentals are joined with a conservative `calendar_year = year(as_of) - 1` as-of rule, and FINRA short-interest exposures are snapped forward 18 calendar days for the disclosure lag (`FactorValueLiftService`, `src/sbfactors/factor_value_lift_service.py`). A loading computed with information the market did not yet have is look-ahead — the single most common way a backtest lies.

6.3. 2. Factor returns

6.3.1. 2.1 Definition

A **factor return** `f_{k,t}` is the payoff, in period t, to holding one unit of exposure to factor k while being neutral to everything else. Where a *loading* is a property of an *asset*, a *factor return* is a property of a *factor* — a time series you could, in principle, earn by holding the factor-mimicking portfolio. There are two standard constructions, and SBFoundation builds both.

6.3.2. 2.2 Construction A — cross-sectional regression (Fama–MacBeth / Barra)

Estimate the factor returns as the **slopes of a cross-sectional regression of realized returns on lagged exposures**, one regression per date. This is the first pass of Fama–MacBeth [**#122**]; SBFoundation uses the Barra USE3 weighting (`sbriskmodel.CrossSectionalRegressionService`, `src/sbriskmodel/cross_sectional_regression_service.py:135-170`):

$$r_{i,t} \;=\; \sum_{k} X_{i,k,t-1}\, F_{k,t} \;+\; \varepsilon_{i,t}, \qquad w_i = \sqrt{\text{market cap}_i}.$$

Solved as ridge-regularized weighted least squares:

```
w    = np.sqrt(mc)                          # sqrt-cap weights (Barra USE3)
XtWX = X.T @ (w[:, None] * X)
XtWr = X.T @ (w * r)
F    = np.linalg.solve(XtWX + ridge_lambda * np.eye(k), XtWr)   # the K facto
residuals = r - X @ F                       # unweighted  → idiosyncratic
```

`F` is the vector of **K factor returns for date t**, written to `ops.risk_factor_return (date, factor_id, factor_return, regression_r_squared, regression_n_obs)`. Cap-weighting makes the estimate approximate the return a real, cap-aware portfolio would have earned; the ridge term `·I` stabilizes the solve when exposures are collinear. Because `X` is `X_{t-1}`, the regression is point-in-time by construction.

The factor **covariance** `Σ_F` is then a rolling 252-day Ledoit–Wolf shrinkage of the factor-return panel [#23][#24] (`factor_covariance_service.py`), and the **idiosyncratic variance** `D` is a rolling 63-day residual variance with a 5th-percentile floor (`idio_variance_service.py`). Together `(X, F, Σ_F, D)` form the risk-model snapshot `ops.risk_factor_*` that portfolio construction and attribution read.

6.3.3. 2.3 Construction B — long/short quantile spread

The alternative, common in academic factor libraries (Fama–French [#44][#112], AQR), is a **portfolio spread return**: sort the cross-section by the characteristic, go long the top quantile and short the bottom, and record the spread's return each period. SBFoundation builds this as a **top-decile-minus-bottom-decile, equal-weighted** series (`scripts/research/validate_diy_factors.py:117-188`):

```
WITH factor_deciles AS (      -- cross-sectional NTILE(10) each formation date
  SELECT date_sk, instrument_sk,
         NTILE(10) OVER (PARTITION BY date_sk ORDER BY value) AS decile
  FROM gold.fact_factor_value WHERE factor_id = ? AND value IS NOT NULL),
daily_returns AS (            -- close-to-close per instrument
  SELECT e.instrument_sk, e.date_sk, d.full_date,
         e.adj_close / LAG(e.adj_close) OVER (...) - 1.0 AS ret
```

```
  FROM gold.fact_eod e JOIN gold.dim_date d ON ... AND d.is_us_market_day),
held AS (                      -- ASOF: assign each day the most recent decile
  SELECT dr.full_date, dr.ret, fd.decile FROM daily_returns dr
  ASOF JOIN factor_deciles fd
    ON dr.instrument_sk = fd.instrument_sk AND dr.date_sk > fd.date_sk),
legs AS (
  SELECT full_date,
         AVG(CASE WHEN decile=10 THEN ret END) AS long_ret,
         AVG(CASE WHEN decile= 1 THEN ret END) AS short_ret
  FROM held GROUP BY full_date)
SELECT full_date, long_ret - short_ret AS ls_ret FROM legs ORDER BY full_date
```

The `ASOF JOIN … date_sk >` enforces strict no-look-ahead (each trading day is scored by the *most recent decile assignment formed before it*). This series is what the platform cross-correlates against the Kenneth French and AQR published factor series to validate that its constructed factors behave like the academic ones (`sbdiag.factor_series_correlation`, tripwire `SB_DIY_FACTOR_CORR_TRIPWIRE`; F-316/F-318).

> **The two are not interchangeable.** Construction A is the *risk-model* factor return (used for risk decomposition and attribution); Construction B is a *validation/benchmark* factor return (used to sanity-check construction against the literature). Keeping them separate is deliberate — see the summary table.

Notion	Where	Method	Storage
Risk-model factor return `F_t`	`sbriskmodel.cross_sectional_regression`	**WLS cross-sectional regression** (ridge), `X_{t-1}` → `r_t`	`risk_factor_return`

Notion	Where	Method	Storage
Long/short quantile factor return	`scripts/research/build_diy_factors.py` (read-only report)	**NEW (10)** **top-minus-bottom decile**, equal-weighted, ASOF-held	`factors.pq`

6.4. 3. The CAPM decomposition: `r =  + ·market +`

6.4.1. 3.1 The model

The Capital Asset Pricing Model (Sharpe 1964 **[#116]**, Lintner 1965 **[#117]**, Mossin 1966 **[#118]**, Treynor **[#119]**) states that in equilibrium the *expected excess return* of any asset is proportional to its market beta:

$$\mathbb{E}[r_i] - r_f \;=\; \beta_i \,(\mathbb{E}[r_m] - r_f).$$

Its **empirical, testable form** is the time-series regression of *realized excess returns*:

$$\underbrace{r_{i,t} - r_{f,t}}_{\text{asset excess return}} \;=\; \alpha_i \;+\; \beta_i \;\underbrace{(r_{m,t} - r_{f,t})}_{\text{market excess return}} \;+\; \varepsilon_{i,t}.$$

Writing `market  (r_m - r_f)` and folding `r_f` into the left side, this is the requested form `r =  + ·market + `. Each term below is defined,

then its estimator and its data source (FMP first, then free sources) are given.

6.4.2. 3.2 Each term, and how to calculate it

`r` — the dependent variable (asset or portfolio excess return). The realized return of the thing being evaluated, in excess of the risk-free rate. For a single stock, `r_{i,t} = adj_close_t / adj_close_{t-1} - 1`. For a strategy, it is the daily NAV return of the book. In SBFoundation, the evaluated streams are NAV return series: `ops.strategy_nav.daily_return` (backtest books) and `ops.signal_backtest_nav.ret_net` (the walk-forward ML book) — see `sbattribution/beta_attribution_service.py:340-378`.

`market` — the market excess return (the single systematic factor). `market_t = r_{m,t} - r_{f,t}`, where `r_m` is the return of a broad market proxy. The choice of proxy is the subject of §7. In SBFoundation the proxy is **SPY** (the S&P 500 ETF), stored as a normalized NAV level in `ops.benchmark_nav` and differenced onto each stream's own dates (`attribution_kernel.py:136-149`). The academic market factor `mkt_rf` (value-weighted CRSP-universe excess return) is additionally available from Kenneth French (§3.4, F-316).

— the market loading (systematic sensitivity). The slope of `r` on `market`; `= Cov(r, market)/Var(market)`. `= 1` moves one-for-one with the market; `> 1` is amplified; `< 1` is defensive. Estimated by OLS (below). At the single-stock level the platform persists a 252-day rolling `beta_f` (§1.2); at the strategy level `sbattribution` estimates per NAV stream.

— the intercept (Jensen's alpha, risk-adjusted excess return). `= [r] - · [market]` — the average return *not explained* by market exposure (Jensen 1968 [**#120**]). A positive, statistically significant is

the empirical signature of skill (or of a missing risk factor). The platform's central institutional question — *does the book beat cheap beta?* — is exactly *"is significantly positive after subtracting SPY (and MTUM, and FF5+UMD) exposure?"* (`sbattribution`, F-294/IR-3). Alpha is reported both per-period (`alpha_daily`) and annualized *by the stream's own frequency* — a monthly book is not inflated ×21 (`settings.py:85-89`).

— **the residual (idiosyncratic / asset-specific return).** `_t = r_t -  - ·market_t` — the part of the return orthogonal to the market. Its variance is the idiosyncratic (diversifiable) risk. See §4.

6.4.3. 3.3 The estimator: OLS with Newey–West HAC errors

SBFoundation estimates and with ordinary least squares, but computes the standard errors with the **Newey–West heteroskedasticity-and-autocorrelation-consistent (HAC)** estimator **[#123]**, because daily NAV returns are serially correlated and heteroskedastic — assuming i.i.d. Gaussian errors would understate the standard errors and overstate significance. The kernel is `sbattribution.ols.ols_hac` (`src/sbattribution/ols.py:79-125`):

- by `np.linalg.lstsq`; residuals `e = y - X .`
- **HAC covariance** with Bartlett weights:

$$V = (X^\top X)^{-1}\Big(S_0+\sum_{l=1}^{L} w_l\,(S_l+S_l^\top)\Big)(X^\top X)^{-1},\quad w_l = 1-\tfrac{l}{L+1},\quad S_l = \sum_t e_t e_{t-l}\,x_t x_{t-l}^\top.$$

- **Lag truncation** `L =  4·(n/100)^{2/9}` (Newey–West 1994 rule of thumb).
- **SE** `= sqrt(diag(V))`; **t-stat** `=  /SE`; **p-value** `= erfc(|t|/√2)` (a two-sided normal approximation, to avoid a hard scipy dependency); **R²** `= 1 - SS_res/SS_tot`.

The design matrix is `[1, market]` for the single-factor CAPM, extended to `[1, r_SPY, r_MTUM]` for the two-factor spanning regression the platform runs by default, and to `[1, mkt_rf, smb, hml, rmw, cma, umd]` for the FF5+UMD basis (§3.4). A `|t| < 2` is greyed out on the SPA as " beta" — indistinguishable from cheap market exposure.

6.4.4. 3.4 The data — FMP first, then free sources

This is the provenance the calculations actually use, ordered as requested (paid FMP first, then free, cheapest-relevant-source-first for anything FMP does not cover).

Term	Primary data (FMP, paid)	Free / non-FMP data	Where it lands
r (stock returns)	**FMP** EOD bulk prices `eod-bulk-price` → `silver.fmp_eod_bulk_price` → `gold.fact_eod.adj_close`; split+dividend-adjusted via `historical-price-eod/dividend-adjusted` (the `adj_close` correction, B-147.6)	—	`gold.fact_eod`
r (strategy NAV)	derived from FMP prices through the backtest	—	`ops.strategy_nav`, `ops.signal_backtest_nav`

6.4. 3. The CAPM decomposition: `r =  + ·market +`

Term	Primary data (FMP, paid)	Free / non-FMP data	Where it lands
`market` (proxy return)	**FMP** SPY (and MTUM) daily closes from `silver.fmp_eod_benchmark_prices`, normalized to NAV	**Kenneth French** value-weighted market excess `mkt_rf` (free, Dartmouth) → `silver.kenneth_french_factors`	`ops.benchmark_nav`; French table
`r_f` (risk-free)	—	**FRED** short-end T-bill `fred-dgs1mo`/`fred-dgs3mo` (free) → `silver.fred_dgs1mo/3mo`; French `rf` column (free)	`silver.fred_*`, `silver.kenneth_french_factors`
`,` (estimates)	computed by `sbattribution.ols_capm` over the above	FF5+UMD loaded from French (free) for the richer spanning regression	`beta_attribution_<run_id>.json`

Two honest caveats the platform documents:

1. **The risk-free rate is currently a benchmark-relative simplification.** SBFoundation's default "excess return" is *excess of benchmark* (SPY / MTUM / French `rf`), not *excess of a resolved short-end RF*. FRED short-end tenors were ingested (F-317) precisely so the *first* excess-return-vs-RF consumer can wire the correct tenor; today no such consumer resolves an RF, and Sharpe/Sortino are computed with `RF = 0` (`signal_backtest_stats.py`). The only

place a FRED yield is used as a rate today is WACC/cost-of-equity, which correctly uses the **10-year** `fred-dgs10`.

2. **SPY vs. the academic market.** SPY (S&P 500, ~500 large caps) is the *investable* proxy; the French `mkt_rf` (the entire CRSP universe, value-weighted) is the *academic* proxy. They differ in small-cap coverage; F-316 added the French basis so attribution is not hostage to a single thin two-ETF basis.

6.5. 4. Systematic vs. idiosyncratic returns

6.5.1. 4.1 The decomposition

Given the factor model, every asset return splits into two orthogonal parts:

$$r_{i,t} = \underbrace{\sum_{k} \beta_{i,k} f_{k,t}}_{\text{systematic (common)}} + \underbrace{\varepsilon_{i,t}}_{\text{idiosyncratic (specific)}} .$$

- **Systematic return** is the part explained by common factors — market, size, value, momentum, sector. It is *undiversifiable*: every asset shares it, so holding more names does not remove it. In equilibrium theory, only systematic risk is compensated with expected return.
- **Idiosyncratic (specific) return** is the residual — the part unique to the asset (a product recall, an earnings surprise, a lawsuit). It is *diversifiable*: in a large portfolio the 's partly cancel, so specific variance falls roughly as `1/N`.

The variance decomposition is `Var(r_i) = _i^T Σ_F _i + Var( _i)`; the **systematic share** is the regression `R²`, and `1 − R²` is the idiosyncratic share.

6.5.2. 4.2 How it is computed here

Three residualizations coexist, each answering a different question:

1. **Risk-model specific return** (the canonical one). The daily cross-sectional regression (§2.2) returns `residuals = r − X·F`; these `_{i,t}` are persisted to `ops.risk_residual` and rolled into `residual_var_63d` (the specific variance `D`) by `IdiosyncraticVarianceService`. The regression's cap-weighted `R²` *is* the systematic-variance share (`cross_sectional_regression_service.py:163-170`).
2. **Market-model residual** for crowding analysis. `sbcrowding` computes `residual_return = stock_return − beta_f · spy_return` in SQL, reusing the persisted `beta_f` (`comomentum_crowding_service.py:223`); abnormal correlation *among these residuals* within a momentum decile is the Lou–Polk **[#74]** comomentum crowding metric.
3. **Residual momentum**. A factor in its own right: momentum computed on the market-model *residual* return (regressed against an equal-weighted market proxy), so the trend signal is neutral to the market beta (Blitz–Huij–Martens **[#50]**; `eod_feature_service.py` Pass 3b → `residual_momentum_252d_f`).

Why it matters for the platform's thesis. The whole point of `sbattribution` (§3) is to ask whether a strategy's return is *systematic* (cheap beta you could buy with an ETF) or genuinely *idiosyncratic alpha*. A book whose return is fully spanned by SPY+MTUM has `0`, small: it is repackaged beta. Persistent, significant with material residual variance is the only thing that justifies active fees.

6.6. 5. From factors to buy/sell signals

A **factor** is a forecast of the cross-section of returns; a **signal** is the decision that forecast implies; a **strategy** is `(factor_id, mechanic, params)` — one forecast wired to a construction rule (`config/strategies/<name>.yaml`). The path from factor value to order is:

1. Score the cross-section. Each name gets a factor value `gold.fact_factor_value.value` (e.g. its momentum score), or an ML signal score `fact_signal_score` from a model trained on many factors (`sbsignals.MLSignalService`).

2. Rank and select (the mechanic). The mechanic turns scores into weights: * `long_only_top_n` — sort by score, buy the top N equal-weighted (e.g. `clinic06-benchmark`: top-20, equal-weight, monthly). Higher score → buy; falling out of the top-N → sell. * `long_short` / dollar-neutral — long the top quantile, short the bottom, `long_ret - short_ret` (the same spread as §2.3). * `long_flat_trend` — hold each name whose *own* time-series trend is positive, else flat (the TSMOM ETF sleeve, F-283). The sign convention matters: a factor with `direction: negative` (e.g. `pe_ratio`, `volatility_30d`) predicts high returns for *low* values, so the mechanic ranks ascending.

3. Validate the forecast (does the signal predict?). The **information coefficient** measures predictive power: the per-date Spearman rank correlation between factor value and forward return, aggregated to an **IC-IR** = mean(IC)/ (IC) (`sbic`, `ic_service.py:1297-1367`). A factor only earns promotion to back a live strategy after clearing IC, permutation-significance (MCPT/HLZ), orthogonality, and overfitting gates — the life-cycle in `sbpromotion`.

4. Damp turnover (hysteresis). Re-ranking daily against a noisy signal churns the book. A hysteresis band only trades a name out once its

rank drifts past a buffer, trading turnover (and cost) against tracking of the ideal book (`sbportfolio.hysteresis`; calibrated in F-310).

5. Size and route. Construction applies executability floors (price $5, $1M ADV — F-304), leverage/position caps, and optional vol-targeting, producing the target book; the settlement watch submits orders through gate checks (decay, PBO/DSR, plausibility, margin — F-307).

So "buy/sell" is not a threshold on one number; it is: *score → rank → select-per-mechanic → (only if the factor has proven IC) → band → size → route.* The factor supplies the **ranking**; the mechanic and gates supply the **decision**.

6.7. 6. Worked example — vol-scaled 12-1 momentum

The factor. `config/factors/momentum_12_1_vol_scaled.yaml` — `source_column: momentum_12_1_vol_scaled_f`, `source_table: fact_eod`, `style: momentum`, `expected_sign: positive`, `hypothesis_class: behavioral`. Economic story: investors under-react to information, so past 12-month winners keep winning over the next month (Jegadeesh–Titman 1993 [**#49**]); scaling by realized volatility targets the same edge while damping the crash-prone tail (Barroso–Santa-Clara 2015 [**#90**], Daniel–Moskowitz 2016 [**#21**]).

The market data it consumes. Only `gold.fact_eod.adj_close` (split- and dividend-adjusted daily closes, sourced from **FMP** `eod-bulk-price` / `historical-price-eod/dividend-adjusted`). From that one column:

1. **Daily log returns:** `g_t = ln(adj_close_t / adj_close_{t-1})`.

2. **The 12-1 trend (the "aspect" being measured):** cumulative return from ~12 months ago to ~1 month ago — the window `[t-252, t-21]` — *skipping the most recent ~21 days* to avoid the well-documented short-term reversal:

$$M_{i,t} \;=\; \prod_{s=t-252}^{t-21} (1 + \mathrm{ret}_{i,s}) - 1.$$

3. **Volatility scaling:** divide the trend by the name's trailing realized volatility (annualized of daily returns), so a unit of signal carries a comparable risk contribution across names:

$$\mathrm{momentum\_12\_1\_vol\_scaled}_{i,t} \;=\; \frac{M_{i,t}}{\sigma_{i,t}}.$$

All of this is evaluated in DuckDB windowed SQL inside `EodFeatureService` (the same engine that produces the `beta_f` slope shown verbatim in §1.2), never in Python — per the platform's "feature math in SQL" constraint.

From value to loading to signal. The wide `momentum_12_1_vol_scaled_f` column is lifted narrow into `gold.fact_factor_value` (`FactorValueLiftService`). From there: (a) the **risk model** standardizes it cross-sectionally into an *exposure* and prices it into a *factor return* (§1–§2); (b) `sbic` measures its **IC-IR** at horizon 21d to confirm it predicts; (c) a `long_only_top_n` mechanic turns the ranked scores into a **buy list** (top-20). The identical *value* also underlies the funnel-exempt `tsmom_12_1_vol_scaled` factor, where the *selection* differs (own-trend sign, not cross-sectional rank) — a clean illustration that a factor's *value* and its *use* are separable.

6.8. 7. Choosing a market benchmark

6.8.1. 7.1 What a benchmark is for

The benchmark plays three distinct roles, and the "right" choice depends on which one you mean:

1. **The market factor** in an asset-pricing model — the `market` term in §3. Theory (CAPM) wants the *true market portfolio*: the cap-weighted portfolio of **all** risky assets.
2. **The performance yardstick** — what a strategy's return is compared against to judge skill ( ) and to compute active return / information ratio.
3. **The risk-neutral reference** — the exposure a long-only book is implicitly benchmarked to when measuring active bets.

6.8.2. 7.2 The theoretical problem: Roll's critique

The true market portfolio is **unobservable** (it includes human capital, private assets, real estate, foreign equities). Roll (1977) [**#124**] showed that *every* empirical CAPM test is therefore a **joint test of the model and of the benchmark proxy**: a "failed" CAPM might just mean the proxy is inefficient. There is no way around this — only mitigation. It is why the platform reports attribution against **multiple** bases (SPY, then SPY+MTUM, then FF5+UMD): no single proxy is privileged, so robustness is shown across proxies.

6.8.3. 7.3 The practical criteria

A usable benchmark should be, per the CFA/index-industry consensus and Grinold–Kahn:

- **Unambiguous & investable** — known constituents you could actually hold (SPY, not "the market");
- **Cap-weighted & broad** — reflects the aggregate opportunity set and is low-turnover;
- **Measurable, with priced total returns** — dividends reinvested;
- **Specified in advance** — chosen before the evaluation period, not fitted to flatter the strategy;
- **Appropriate to the strategy's universe** — a US large-cap momentum book is measured against a US large-cap index, not a global aggregate.

6.8.4. 7.4 How SBFoundation chooses

- **Market proxy = SPY.** Broad, liquid, cap-weighted, investable, dividend-adjusted, near-zero cost to replicate — it satisfies §7.3 for a US-equity platform. It is the `market` term in attribution and the dashed line on every equity-curve overlay.
- **Factor references = MTUM (+ the FF5+UMD basis).** Because the platform *trades momentum*, a pure market proxy is not enough to prove skill — a momentum book should be measured against a *momentum* benchmark. `MTUM` (MSCI USA Momentum ETF) is the investable one; the Kenneth French `umd`/`hml`/`rmw`/… series are the academic ones. This is exactly the "does it beat cheap *factor* beta, not just market beta?" test.
- **A platform yardstick = `clinic06-benchmark`.** A long-only, top-20, equal-weight, monthly, vol-scaled-momentum replica run through the *real* pipeline (`config/strategies/clinic06-benchmark.yaml`). It is not a tradable alpha; where it diverges from the external reference, *the divergence is the measurement of a platform behaviour* (PIT-clean universe, broker-true costs) — a benchmark for the *machinery*, not for the *market*.
- **Categories are enforced.** Each configured benchmark carries a `category {core_equity, bonds_volatility, sector_etf}`

(`benchmark_meta` in the strategy YAML), so overlays can group like-for-like.

6.9. 8. Academic studies on benchmark construction, and the calculation process

6.9.1. 8.1 The literature

Theme	Study	What it establishes
The market portfolio / CAPM benchmark	**Sharpe (1964)** [**#116**], **Lintner (1965)** [**#117**], **Mossin (1966)** [**#118**]	Equilibrium prices risk relative to the cap-weighted market portfolio — the benchmark is *the market*.
Benchmark unobservability	**Roll (1977)** [**#124**]	The true market is unobservable; every benchmark is a proxy, so tests are joint tests of model + proxy.
Zero-beta / flat SML	**Black (1972)** [**#121**]	Without riskless borrowing the benchmark-relative security market line is flatter than CAPM — motivates Betting-Against-Beta [**#93**].

Theme	Study	What it establishes
Multifactor benchmarks	**Fama–French (1992/1993)** [#55][#44][#112], **Carhart (1997)** [#57], **Fama–French (2015)** [#56]	The benchmark is *multidimensional* (market + size + value + profitability + investment + momentum); defines how the SMB/HML/RMW/CMA/UMD factor-benchmark portfolios are built.
Policy benchmark dominance	**Brinson–Hood–Beebower (1986)** [#70], **Brinson–Fachler (1985)** [#27]	The benchmark (asset-allocation policy) explains the bulk of return variation — attribution is defined *relative to* it.
Active-management benchmark theory	**Grinold (1989)** [#25], Grinold–Kahn (2000)	Formalizes active return / active risk / information ratio *against a benchmark*; the Fundamental Law.
Naive vs. optimized benchmark	**DeMiguel–Garlappi–Uppal (2009)** [#60]	1/N is a hard benchmark to beat out-of-sample — a cautionary reference for "smart" benchmarks.

6.9.2. 8.2 The benchmark-calculation process (cap-weighted total-return index)

The canonical construction the literature and index providers use, and the one SPY/the S&P 500 follows:

1. **Define the universe & selection rules** — eligibility (listing, liquidity, float, domicile). For factor benchmarks (French/AQR), rank the universe on the characteristic and take quantile breakpoints.
2. **Weight the constituents — float-adjusted market-capitalization weight** `w_i = (price_i · float_shares_i) / Σ_j (price_j · float_shares_j)`. Factor benchmarks are typically equal- or value-weighted *within* the long and short legs (French uses value-weighted 2×3 sorts).
3. **Compute the index return** — `R_t = Σ_i w_{i,t-1} · r_{i,t}` with **total return** `r_{i,t}` (dividends reinvested). This is precisely the weighted-average-of-constituent-returns that `benchmark_nav.py` reproduces by normalizing an ETF's total-return NAV to 1.0.
4. **Reconstitute & rebalance** on a schedule (quarterly/annual), applying buffering to limit turnover — the analogue of the platform's hysteresis.
5. **Excess-return form** — for asset-pricing use, subtract the risk-free rate: `mkt_rf = R_m - r_f` (French subtracts the 1-month T-bill; the platform's FRED short-end tenors, F-317, are the resolved-RF equivalent).

SBFoundation does **not** re-derive an index from constituents; it consumes a provider's already-computed total-return series (SPY/MTUM ETF closes) and *normalizes* it to a NAV level — the pragmatic equivalent of steps 2–3 above (`benchmark_nav.py:1-27`).

6.10. 9. Data sources for benchmarks (FMP first, then free)

Rank	Source	Cost	Series	How it enters the platform
1	**FMP**	Paid (primary vendor)	SPY, MTUM, QQQ, IWM, sector SPDRs, bond/vol ETFs (18 total) — daily closes	`eod-bulk-price` → `silver.fmp_eod_bulk_price` → normalized → `ops.benchmark_nav`
2	**Kenneth French Data Library** (Dartmouth)	**Free**	`mkt_rf` (academic market), `smb`, `hml`, `rmw`, `cma`, `umd`, `rf` — daily & monthly	`sbops.KennethFrenchFactorS` (stdlib urllib/zipfile) → `silver.kenneth_french_fact` (F-316)
3	**FRED** (St. Louis Fed)	**Free**	Risk-free tenors `DGS1MO/DGS3MO` (short-end), `DGS10` (WACC), plus term/credit/vol spreads	keymap `fred-*` → `silver.fred_*` (F-317)

Rank	Source	Cost	Series	How it enters the platform
4	**AQR Data Library**	**Free** (manual)	Published `QMJ`, `BAB`, `Value`, `TSMOM` monthly factor returns	committed CSV `data/reference/aqr/aqr_factor_returns.csv` (F-318 reference shelf)

Guidance: for an **investable** benchmark and total-return NAVs, FMP ETF closes are primary. For an **academic** market/factor benchmark (and the risk-free rate), the free Kenneth French and FRED series are preferred over any paid equivalent — they are the field's ground truth and cost nothing. AQR provides an independent free cross-check on constructed factors. The platform deliberately holds all three free series so its attribution and factor-validation are not captive to a single paid vendor.

6.11. 10. A canonical factor catalogue (economic justification + style)

Thirteen well-established factors, each with its economic rationale, its **style** tag (the platform's own taxonomy), the anchoring citation, and the SBFoundation `factor_id` that implements it. Styles in use across `config/factors/*.yaml`: `momentum, value, quality, volatility, growth, liquidity, positioning, size, beta`.

#	Factor	Style	Economic justification (why a premium exists)	Anchor citation	Platform `factor_id`
1	**Market**	`beta`	Compensation for bearing undiversifiable market risk — the one factor everyone must hold; the CAPM premium.	Sharpe 1964 [**#116**], Lintner 1965 [**#117**]	`market_beta_252d` (`beta_f`)

#	Factor	Style	Economic justification (why a premium exists)	Anchor citation	Platform `factor_id`
2	**Size**	`size`	Small caps earn a premium for illiquidity, distress, and limited analyst coverage; partly a compensation, partly a limits-to-arbitrage effect.	Banz 1981 **[#125]**, Fama–French 1992 **[#55]**	`market_cap`

#	Factor	Style	Economic justification (why a premium exists)	Anchor citation	Platform `factor_id`
3	**Value**	`value`	Cheap (high E/P, B/M, FCF-yield) stocks out-earn expensive ones — risk premium for distressed/low-growth firms *and* mispricing from extrapolation.	Fama–French 1992/1993 **[#55][#44]**	`earnings_yield`, `pe_ratio`, `value_composite`, `pb_ratio`, `ps_ratio`, `pfcf_ratio`

6.11. 10. A canonical factor catalogue (economic justification + style)

#	Factor	Style	Economic justification (why a premium exists)	Anchor citation	Platform `factor_id`
4	**Momentum**	`momentum`	Past 12-1 winners keep winning — under-reaction to news and delayed information diffusion (behavioral).	Jegadeesh–Titman 1993 **[#49]**, Carhart 1997 **[#57]**	`momentum_12m_1m`, `momentum_12_1_vol_scaled`

#	Factor	Style	Economic justification (why a premium exists)	Anchor citation	Platform `factor_id`
5	**Profitability / Quality**	`quality`	Profitable, safe, well-managed firms (high gross-profitability, ROIC, QMJ) out-earn junk — a premium for quality that markets underprice.	Novy-Marx 2013 [**#92**], Fama–French 2015 [**#56**], Asness-Frazzini-Pedersen 2019 [**#113**]	`qmj_composite`, `roic_spread`, `gross_profitability`, `piotroski`

6.11. 10. A canonical factor catalogue (economic justification + style)

#	Factor	Style	Economic justification (why a premium exists)	Anchor citation	Platform `factor_id`
6	**Investment / Accruals**	`quality/growth`	Firms that invest conservatively and have low accruals out-earn aggressive investors — over-investment and earnings-quality mispricing.	Sloan 1996 **[#83]**, Cooper-Gulen-Schill 2008 **[#103]**, Fama–French 2015 **[#56]**	`accruals`, `earnings_growth_qoq`

#	Factor	Style	Economic justification (why a premium exists)	Anchor citation	Platform `factor_id`
7	**Low volatility / Defensive**	`volatility`	Low-risk stocks earn higher risk-adjusted returns — leverage constraints and lottery-preference bid up high-vol names (the low-vol anomaly / BAB).	Ang et al. 2006 [**#108**], Frazzini–Pedersen 2014 [**#93**], Novy-Marx 2014 [**#10**]	`volatility_30d`

#	Factor	Style	Economic justification (why a premium exists)	Anchor citation	Platform `factor_id`
8	**Liquidity / Illiquidity**	`liquidity`	Illiquid stocks (high Amihud price-impact, low $ADV) require a return premium for the cost and risk of trading them.	Amihud 2002 **[#48]**	`amihud`, `adv_dollar_20d`

#	Factor	Style	Economic justification (why a premium exists)	Anchor citation	Platform `factor_id`
9	**Short interest / Positioning**	`positioning`	Heavily-shorted / high-days-to-cover names underperform — informed short sellers *and* short-sale-constraint overvaluation (divergence of opinion).	Boehmer-Jones-Zhang 2008 [**#101**], Rapach-Ringgenberg-Zhou 2016 [**#100**], Miller 1977 [**#111**]	`days_to_cover`, `short_pct_float`

#	Factor	Style	Economic justification (why a premium exists)	Anchor citation	Platform `factor_id`
10	**Seasonality**	`momentum`	Same-calendar-month historical returns recur — persistent cross-sectional seasonalities from mood, liquidity, and information cycles.	Heston–Sadka 2008 **[#37]**, Hirshleifer et al. 2020 **[#40]**	`seasonality_same_month`

#	Factor	Style	Economic justification (why a premium exists)	Anchor citation	Platform `factor_id`
11	**Earnings momentum / PEAD**	`growth`	Prices under-react to earnings surprises (SUE), drifting for weeks after the announcement — the post-earnings-announcement drift.	Bernard–Thomas 1989 **[#75]**	`sue`

#	Factor	Style	Economic justification (why a premium exists)	Anchor citation	Platform `factor_id`
12	**Time-series momentum / Trend**	`momentum`	An asset's *own* past 12-1 return predicts its next-month return across asset classes — trend-following / slow-diffusion premium.	Moskowitz-Ooi-Pedersen 2012 **[#115]**, Hurst-Ooi-Pedersen 2017 **[#95]**	`tsmom_12_1_vol_scaled`

#	Factor	Style	Economic justification (why a premium exists)	Anchor citation	Platform `factor_id`
13	**Short-horizon reversal**	`volatility`	Very-short-horizon losers bounce (and winners fade) — liquidity provision / over-reaction correction (contrarian).	Lehmann 1990 [**#96**]	`bb_pct`

Caveats the platform enforces. A factor's premium may be *risk compensation* or *mispricing*; the `economic_rationale.hypothesis_class` field (`risk_premium | behavioral | structural | data_mining`) records which, and a `data_mining` story is refused at the gate. Premia also **decay after publication** (McLean–Pontiff 2016 [**#14**]), so IC and significance are re-measured every run, not assumed. And the *factor zoo* is largely redundant (Feng–Giglio–Xiu 2020 [**#13**], Jensen–Kelly–Pedersen 2023 [**#98**]) — SBFoundation collapses near-duplicate factors to a single effective trial before counting significance (LH-6y /

F-295).

6.12. 11. References

All citations resolve to `docs/reference-papers/README.md` by the bracketed row number. The CAPM/estimation foundations (Sharpe, Lintner, Mossin, Treynor, Jensen, Black, Fama–MacBeth, Newey–West, Roll, Banz) are catalogued there under **§16**, added alongside this note. Foundational, benchmark, factor-zoo, and per-factor sources are catalogued under §§1–15.

Primary code touchpoints (for the reader who wants to trace any claim to source):

- Loadings / exposures — `src/sbriskmodel/factor_exposure_service.py`; `src/sbfactors/eod_feature_service.py:364-431` (`beta_f`).
- Factor returns — `src/sbriskmodel/cross_sectional_regression_service.py`; `scripts/research/validate_diy_factors.py` (L/S spread).
- CAPM / attribution — `src/sbattribution/ols.py`, `beta_attribution_service.py`; `src/sborchestration/tasks/benchmark_nav.py`.
- Systematic/idiosyncratic — `ops.risk_residual`, `IdiosyncraticVarianceService`; `src/sbcrowding/services/comomentum_crowding_service.py`.
- Factors & signals — `config/factors/*.yaml`, `src/sbcontracts/settings.py` (`ACTIVE_FACTOR_IDS`); `src/sbic/services/ic_service.py`; `src/sbsignals/`, `src/sbportfolio/`.
- Free reference data — `src/sbops/kenneth_french_service.py`, `silver.kenneth_french_factors`, `silver.fred_*`, `data/reference/aqr/`.

Part II.

The Platform as Research Substrate

7. SBFoundation as a Research Platform

Author: Todd B. Adams **Companion to:** PhD Research Proposal — Dynamic Heterogeneous Multiplex Networks for Factor-Based Systemic Risk & Market Phase Transitions **Status:** draft evidence package (supervisor-facing)

7.1. Thesis

The proposal argues that systemic risk is an emergent property of a multi-layered financial network, and that a spatial-temporal graph model can detect its phase transitions. Testing that claim credibly needs more than a notebook and a static academic panel — it needs a **live, point-in-time-correct, survivorship-adjusted, cost-aware, executable** empirical instrument. SBFoundation *is* that instrument. It runs the full chain from raw vendor data to real paper-broker fills every night, it already implements the proposal's **stock–factor layer** in production, it already carries **correlation-network and crowding** machinery, and it has **already stress-tested the supervisor's own published connectedness methodology** on its universe. This folder is the evidence for those claims, mapped gate-by-gate to the proposal.

7.2. How to read this folder

The docs are numbered as a reading order. Each is skimmable (~1–2 pages) and cross-links into the deeper platform documentation rather than restating it.

#	Doc	Answers	Reinforces proposal §
00	**README** (this file)	What is this and why does it exist?	§1
01	Platform as research substrate	Why is a live E2E trading platform the right instrument for this research?	§1, §3
02	The end-to-end pipeline	Does it really run data → live trading, reproducibly?	§3
03	The multiplex data substrate	Can it source the proposal's three network layers?	§5
04	The factor layer as multiplex Layer 1	Is the bipartite stock–factor layer hypothetical, or built?	§3, §6

#	Doc	Answers	Reinforces proposal §
05	Systemic-risk & crowding groundwork	Is there prior, supervisor-aligned network work here?	§2, §3
06	The integrated target architecture	Where does the Supra-Laplacian / ST-GNN engine mount?	§3
07	Gap analysis & research roadmap	What is built vs. proposed, honestly?	§3, §5
08	Validation & anti-overfitting	How are network claims protected from self-deception?	§6
09	Reproducibility & experiment infrastructure	Is each experiment auditable and repeatable?	§3
10	Literature bridge	Where do the proposal's papers live in the codebase?	§2

7.3. What this package is engineered to demonstrate

1. **Reinforce the proposal** — the research question already has platform-grounded precedent (docs 01, 04, 05, 10).

2. **The platform supports the research** — substrate, mount points, honest gaps, and validation rigor (docs 03, 06, 07, 08, 09).
3. **End-to-end, data to live trading** — a production chain terminating in real broker fills, not a simulator (docs 02, 09).

7.4. Status legend

Every forward-looking claim in this folder carries one of three tags, so nothing is over-claimed:

- `BUILT` — running in production today; traceable to committed packages and nightly runs.
- `PARTIAL` — substrate exists and is adjacent, but the specific capability is incomplete.
- `PROPOSED` — the research extension; not yet implemented. This is the PhD.

7.5. Relationship to `docs/phd/`

The proposal and its bundled source papers stay in `docs/phd/`. This folder is the *supporting evidence*: read the proposal for the research argument, read here for the platform that makes it testable.

8. The Platform as a Research Substrate

Author: Todd B. Adams **Reinforces:** Proposal §1 — Executive Summary & §3 · **Reading order:** doc 01 of the research-platform package

> **The claim of this doc.** Most factor-network research is validated on static, survivor-biased, cost-free academic panels — precisely the conditions the proposal (§1.1) criticizes traditional risk models for assuming. SBFoundation removes those conveniences. It is a live empirical instrument built to be *hard on itself*, which is exactly what makes it a credible testbed for a claim as strong as "we can detect market phase transitions."

8.1. Why a production trading platform, not a notebook

The proposal's model must survive four failure modes that kill most quant-network research. The platform addresses each as a standing architectural property, not a one-off cleanup.

Failure mode	How amateur research fails	How SBFoundation is built	Status
Look-ahead leakage	Features accidentally use future data; the "signal" is time travel	Gold is point-in-time by construction — a historical date sees only what was knowable then	`BUILT`
Survivorship illusion	Dead companies dropped; predictability manufactured, not returns	4,858 delisted names spliced back with realistic terminal losses; the evidence gauntlet subtracts the measured haircut	`BUILT`
Cost fantasy	Gross edges that die after the spread	Per-name spread + capacity model; every result reported net-of-cost	`BUILT`
Overfitting	Best-of-many-tries kept and reported as if tested once	Deflated-Sharpe, PBO, purged-CV, honest family-wise trial counts	`BUILT`

A network model that clears *these* conditions is making a far stronger claim than one fit on a clean CRSP panel. That is the substrate's entire value to the research.

8.2. What the substrate offers the network model

1. **Point-in-time tensors for free.** The proposal's Supra-Adjacency tensors (factor loadings, dynamic edge weights) would be built from Gold — inheriting its no-look-ahead guarantee without new engineering. See doc 02.
2. **A real cross-section, not N=100 synthetic nodes.** The proposal's simulator (§6) uses 100 synthetic stocks to *proof* the physics; the platform offers the empirical graduation target — a full, liquid, survivorship-corrected equity universe with ~2 decades of history.
3. **An executability verdict.** A phase-transition signal is only useful if it can inform a tradeable book. The platform terminates in cost-aware, risk-capped, broker-submitted orders (doc 02), so a network signal is judged on executed P&L, not in-sample fit.
4. **A validation gauntlet already calibrated to the literature.** The anti-overfitting machinery (doc 08) applies to *any* new producer — including a network one — so GNN claims face the same bar as every factor.
5. **Reproducible experiments.** Every run is immutable and audited (doc 09).

8.3. The honest boundary

SBFoundation does **not** implement multiplex networks, GNNs, supply-chain ingestion, or 13F ownership today. Those are the research (`PROPOSED`). What it provides is the *substrate and the discipline* — the reason the research can be done credibly here rather than started from

zero. Where each proposed piece attaches is doc 06; what is missing and in what order it lands is doc 07.

Cross-links: platform whitepaper · business capability map · proposal §1

9. The End-to-End Pipeline: Data to Live Trading

Author: Todd B. Adams **Reinforces:** Proposal §3 — High-Level Architecture · **Reading order:** doc 02 of the research-platform package

> **The claim of this doc.** SBFoundation is not a research notebook that stops at a backtest. It is a production assembly line that runs *every night* from raw vendor data to a reconciled paper-broker order, under **one run identity, one decision window, and no look-ahead**. That production discipline — not a bespoke model — is what qualifies the platform to host and honestly evaluate the proposal's network research.

9.1. The nightly assembly line — `BUILT`

```
flowchart LR
    A["Acquire<br/>raw end-of-day data"] --> B["Validate<br/>clean & conform"]
    B --> C["Model<br/>point-in-time"]
    C --> D["Factors<br/>rank the names"]
    D --> E["Strategy<br/>apply a mechanic"]
    E --> F["Portfolio<br/>weights + risk caps"]
    F --> G["Paper order<br/>submit & reconcile"]
```

One run identity end-to-end; one nightly decision window; no future data is ever visible on a historical date. Adapted from the platform whitepaper §4.

Each stage is a real, tested subsystem with its own package, its own immutable output, and its own dashboard view. The chain is not aspirational — it is what a `NIGHTLY` run does.

Stage	What it does	Package(s)	Evidence / detail	Status
Acquire	Ingest exact vendor payloads, preserve untouched (append-only Bronze); every fetch and every *failure* recorded	`sbbronze`	Bronze contracts, dataset reference	`BUILT`
Validate	Clean, type, dedupe, conform; restrict to an investable universe	`sbsilver`, `sbuniverse`	pipeline steps, universe API	`BUILT`

Stage	What it does	Package(s)	Evidence / detail	Status
Model	Star-schema analytics layer with strict **point-in-time integrity** — on any date you see only what was knowable then	`sbgold`	DuckDB storage contract	`BUILT`
Factors	Rank every name point-in-time; one row per date × instrument × factor	`sbfactors`	factors, factor lifecycle	`BUILT`
Strategy	Apply a mechanic (e.g. "hold top-ranked for a month") to a *validated* factor	`sbcontracts`, `sbportfolio`	portfolio construction	`BUILT`

Stage	What it does	Package(s)	Evidence / detail	Status
Portfolio	Turn rankings into position sizes, **compliant by construction** (position/sector/leverage caps baked in before any order)	`sbportfolio`, `sbriskmodel`, `sballocation`	portfolio construction	`BUILT`
Paper order	Whole-share orders to an Interactive Brokers **paper account**; real fills at the real open; next-day reconciliation	`sbexecution`	execution reference	`BUILT`

Orchestration across all seven stages is a single Prefect flow with one `run_id` — see the run-type task reference and prefect task reference.

9.2. The "live" end is real, and honestly gated — BUILT / PARTIAL

The final stage is the one most quant research never reaches. SBFoundation reaches it:

- **Real broker fills, not a simulator.** A pre-open submit-and-settle watch submits the target book to IB at ~09:10 ET, then polls until every order settles, capturing real fills and marking positions same-day (execution reference). The first live paper fills are recorded — a real run booked 7 fills / 951 shares.
- **A fail-closed safety rail.** Between "decide" and "submit," a guard re-checks the intended book against its risk caps; any breach rejects the *entire* submission — no partial, no override.
- **Paper-only, by dated decision — not "never."** The loop runs paper-only through the current phase; the lifecycle gate only permits live capital after a **63 market-day** attributed paper soak. Live money is *gated behind survived evidence*, not out of scope. PARTIAL because the crossover to real capital is scheduled, not yet taken.

9.3. Why this matters for the research

Three properties of the pipeline are exactly what the proposal's network model needs from a host — and what a standalone GNN codebase would have to build from scratch:

1. **Point-in-time integrity end-to-end.** A network-based early-warning signal is only meaningful if it never peeks at the future. The proposal's Supra-Adjacency tensors (factor loadings, dynamic edge weights) would be built from Gold, which is already engineered so a historical date sees only what was knowable then. PROPOSED network layers inherit this guarantee for free.

2. **A terminal executability test.** The proposal's phase-transition signal is only useful if it can inform a *tradeable* book. The platform already turns a signal into cost-aware, capacity-aware, risk-capped, broker-submitted orders — so a network signal can be evaluated on the metric that matters (net, executed P&L), not just in-sample fit.
3. **Every run is an immutable experiment.** Each night emits a `run_id`, per-stage JSON sidecars, and a dashboard bundle (doc 09). A network producer added to the flow becomes a first-class, reproducible experiment with the same audit trail as every existing stage.

The proposal's data-ingestion engine and the platform's Acquire→Model stages are the *same idea at different maturities*. Where the proposal's three network layers map onto this substrate — one built, two gaps with adjacent machinery — is the subject of doc 03; where the network engine mounts onto the flow is doc 06.

Cross-links: platform whitepaper · business capability map · architecture packages

10. The Multiplex Data Substrate

Author: Todd B. Adams **Reinforces:** Proposal §5 — Data Requirements & Sourcing · **Reading order:** doc 03 of the research-platform package

> **The claim of this doc.** The proposal names three data layers (§5). One of them runs in production here today; the other two are gaps — but gaps with real adjacent machinery already in the codebase, not blank slates. This is the honest map.

10.1. The three proposal layers, mapped to platform reality

Proposal layer (§5)	Target variables	Platform reality today	Status
Asset pricing & factors (Layer 1)	OHLCV, Fama-French 3/5, momentum, volatility	Fully sourced: point-in-time EOD (`gold.fact_eod`), a governed factor library (`gold.fact_factor_value`), FF5+UMD return series (`silver.kenneth_french_factors`), and an AQR factor-return reference shelf	`BUILT`
Supply-chain links (Layer 2)	Customer–supplier directionality, revenue %	Not ingested. But the platform already runs a stdlib **SEC EDGAR** fetch/parse harness (F-319 PIT-integrity audit) — the 10-K extraction path the proposal names	`PROPOSED` (adjacent: EDGAR machinery `BUILT`)

Proposal layer (§5)	Target variables	Platform reality today	Status
Institutional ownership (Layer 3)	13F holdings, share counts, AUM	Not ingested. But **FINRA short-interest** is sourced for positioning factors (F-299), and `sbcrowding` computes a return-correlation crowding proxy — both approximations of the holdings-based ground truth	`PROPOSED` (adjacent: FINRA + crowding `BUILT`)

10.2. Layer 1 — the built layer

This is the load-bearing point of the whole package: the proposal's *hardest* data layer to get right — clean, point-in-time, survivorship-corrected factor loadings — is the one the platform already delivers. Details in doc 04. The proposal generates this layer *synthetically* (§6.1, structural SDEs); here it is empirical and audited.

Beyond raw factors, the platform carries the academic reference series the proposal's benchmarking needs: - **Fama-French 5 + Carhart UMD**, daily and monthly, in `silver.kenneth_french_factors` (F-316) — the exact "French Data Library" source the proposal cites, already ingested.

- **AQR factor returns** (QMJ, BAB, TSMOM) on a committed reference shelf (F-318) for cross-checking constructed factors against published ones. - **FRED macro series** (term/credit/vol spreads, risk-free curve) in `silver.fred_*` (F-317) — regime covariates for a phase-transition model.

10.3. Layer 2 — supply chain (PROPOSED)

The proposal's free-sourcing path is "SEC Edgar Parsing (Form 10-K extraction via NLP)." The platform **already has an EDGAR client** — a stdlib `urllib` fetcher with CIK resolution, filings-index parsing, and XBRL companyfacts extraction — built for the F-319 point-in-time integrity audit (EDGAR PIT findings). Extending it from *fundamentals verification* to *customer–supplier edge extraction* is a scoped ingestion task, not new infrastructure. This is the least-mature of the three layers and is honestly the largest single data-engineering item in the roadmap (doc 07).

10.4. Layer 3 — institutional ownership (PROPOSED)

The proposal wants 13F holdings to model ownership crowding. The platform approaches the same construct from two directions today: - **FINRA short interest** (`source: finra`) is ingested and drives positioning-style factors (`days_to_cover`, `short_pct_float`, F-299) — the short side of ownership positioning, point-in-time-snapped to settlement + dissemination lag. - **`sbcrowding` comomentum** infers crowding from *return correlations* rather than holdings — a proxy that the 13F layer would replace with ground truth (doc 05).

13F ingestion reuses the same EDGAR client as Layer 2, so the two gaps share a sourcing path.

10.5. Why the gaps are credible, not hand-waving

Every `PROPOSED` layer above attaches to a `BUILT` mechanism: EDGAR ingestion, FINRA sourcing, crowding computation, and the keymap-driven dataset framework (dataset reference) that makes adding a new dataset a declarative config change, not a rewrite. The research adds *datasets and a model*; it does not add a data platform.

Cross-links: dataset reference · EDGAR PIT findings · proposal §5 · factor layer

11. The Factor Layer as Multiplex Layer 1

Author: Todd B. Adams **Reinforces:** Proposal §3 (Layer 1) & §6.1 ·
Reading order: doc 04 of the research-platform package

> **The claim of this doc.** The proposal's first network layer — the bipartite **stock factor** graph with weighted edges (factor loadings) — is not something the research must build. It exists in production as `gold.fact_factor_value`, point-in-time and governed by a full lifecycle. The proposal simulates this layer with structural SDEs (§6.1); here it is empirical.

11.1. The bipartite layer is already a table

The proposal's Layer 1 (§3, Supra-Adjacency Tensor Builder) is a bipartite graph: stock nodes on one side, factor nodes on the other, edge weight = the factor loading. The platform's factor store *is* that edge list:

- **One row per (date, instrument, factor_id)** in `gold.fact_factor_value` — literally a timestamped weighted edge between a stock node and a factor node.

- **Factor loadings proper** — market beta (`beta_f`) and the broader factor cross-section — are computed point-in-time, so the edge weights are the real, knowable-on-the-date values, not a synthetic draw.
- **The active node set** is governed by `sbcontracts.settings.ACTIVE_FACTOR_IDS`; factors carry economic style tags, so the factor-side nodes are typed (value / momentum / quality / positioning / …).

Details of the factor mathematics and the `r =  + ·market +` decomposition are in the codebase-grounded primer factor-models-and-benchmarks.md; the factor engineering pipeline is in factors.md.

11.2. The edges are trustworthy, not just present — `BUILT`

A network is only as good as its edge weights. The platform subjects every factor edge to the validation funnel before it is trusted (the "mathematics of trust", doc 08): information-coefficient hurdle, permutation significance, survivorship haircut, net-of-cost check, overfitting/lockbox, and orthogonality. A factor that fails stays `experimental`; only a `validated` factor's edges back a real strategy. So the Layer-1 graph the research would consume is *pre-cleaned* — its edges have already survived a literature-grade gauntlet.

11.3. What the proposal adds on top of Layer 1

Proposal element	Relationship to the built factor layer	Status
Bipartite stock–factor layer (§3, Layer 1)	Is `gold.fact_factor_value` today	`BUILT`
Dynamic (time-varying) edge weights	Factor values are already recomputed every run, point-in-time	`BUILT`
Node features (rolling price, volatility, factor arrays)	Already columns in `gold.fact_eod` / `gold.fact_factor_value`	`BUILT`
Embedding Layer 1 into a Supra-Adjacency tensor with inter-layer coupling	The tensor assembly across Layers 1–3	`PROPOSED`
Heterogeneous message-passing (HGT / ST-GNN) over the tensor	The geometric deep-learning engine	`PROPOSED`

11.4. Why this matters for the research

The proposal's §6 simulator exists precisely because a synthetic Layer 1 is needed to *proof* the physics before touching real data. On this platform the empirical Layer 1 is already available, point-in-time and validated — so the research can move from synthetic proofing to real-data evaluation without first building a factor-loading pipeline. The stock–factor layer is the bridgehead; Layers 2 and 3 (doc 03) and the tensor/engine (doc 06) build outward from it.

Cross-links: factor-models primer · factors pipeline · factor lifecycle · proposal §3/§6

12. Systemic-Risk & Crowding: Existing Groundwork

Author: Todd B. Adams **Reinforces:** Proposal §2 — Theoretical Foundations & §3 (network layers) · **Reading order:** doc 05 of the research-platform package

> **The claim of this doc.** The proposal's network idea is not foreign to this platform — a version of it is already *running*. SBFoundation carries correlation-network, factor-network, and crowding machinery in production, and it has **already run an empirical spike against the supervisor's own published connectedness methodology**. That prior work both establishes precedent and sharpens the proposal's distinct contribution.

12.1. Network structure already lives in the platform — `BUILT`

The proposal treats assets, factors, and funds as nodes across topological layers. The platform already computes network statistics over two of those object types today:

Existing capability	What it is, in network terms	Package / file	Grounded in
Comomentum crowding (FC-1)	Abnormal within-decile return *correlation* among a factor's constituent names — a correlation-network statistic over the traded cross-section	`sbcrowding` → `comomentum_crowding_service.py`, `_comomentum_math.py`	Lou & Polk (2012), *Comomentum*
Factor -network (LH-6y)	Pairwise |Spearman | among all active factors — a weighted factor–factor graph, collapsed to effective trial count	`sbic` → `pairwise_correlation_service.py`	Benjamini–Yekutieli FDR over a dependency graph
Orthogonality / confound (LH-6, LH-6z)	Residual-IC of a candidate factor against the validated book — an edge-removal / independence test on the factor graph	`sbic` → `orthogonality_service.py`, `factor_confound_service.py`	rank-space residualization (F-305)

Existing capability	What it is, in network terms	Package / file	Grounded in
Factor risk model	Factor covariance + idiosyncratic variance — the dense dependency matrix the proposal denoises	`sbriskmodel`	Barra-style decomposition (F-073)

The network concept is therefore *already present at the factor level and in crowding*. What the proposal adds is an **asset-level, multi-layer** read and a **spectral phase-transition** signal — genuinely new, but built on a substrate that already speaks this language.

12.2. The Di Matteo connection — prior empirical work, already done — `BUILT`

The proposal's supervisor, Prof. Tiziana Di Matteo, co-authored the survey the platform *already stress-tested*: **Raddant & Di Matteo (2023), *A look at financial dependencies by means of econophysics and financial economics.*** In July 2026 a read-only research spike ran a **Diebold–Yilmaz total connectedness index** over the platform's own 2004–2026 universe (5,698 trading days × 11 GICS sectors, rolling VAR → generalized FEVD) to test whether network connectedness *leads* market stress.

The honest finding — and why it strengthens, not weakens, the proposal:

- Connectedness proved **coincident, not leading** (`corr(TCI, forward-vol)` fell monotonically with horizon: +0.42 at h=0 → +0.24 at h=42d), with a saturated dynamic range and sensitivity to universe composition. Full write-up: asset-dependency networks findings.
- This *falsified the simplest network-as-early-warning hypothesis* before any production code shipped — exactly the empirical discipline a doctoral programme rewards.
- **It also localizes the proposal's contribution precisely.** The proposal does not claim DY connectedness predicts crashes. It claims the **eigenvalue spectrum of the Supra-Laplacian** (algebraic connectivity / critical slowing down across a *multi-layer* graph) carries an early-warning signature. That is a *different and stronger* mathematical object than a single-layer VAR connectedness index — and the platform's spike is the evidence that the weaker version was tested and found wanting. The PhD picks up exactly where the spike left off.

12.3. How this maps onto the proposal's layers

- **Layer 3 (institutional ownership / crowding).** The platform's comomentum producer is a *return-correlation* proxy for crowding today; the proposal's 13F ownership layer is the *holdings-based* ground truth. One is an approximation of the other — see the gap analysis in doc 03 and doc 07.
- **Physics evaluation engine (§3).** The platform already has the report-only, evidence-first grammar (`ops.research_*` table → sidecar → dashboard card, no gate at first) that a Supra-Laplacian spectral producer would inherit — the `sbcrowding` producer is the literal template. See doc 06.

12.4. Convention note

The Raddant & Di Matteo (2023) paper is **not yet captured** in `docs/reference-papers/` (it is open-access, CC-BY). Doc 10 flags it for bundling; Lou & Polk (2012) *Comomentum* is already bundled there (row 74).

Cross-links: proposal §2 literature · bundled network/GNN papers · DY-connectedness spike

13. The Integrated Target Architecture

Author: Todd B. Adams **Reinforces:** Proposal §3 — High-Level Architecture · **Reading order:** doc 06 of the research-platform package

> **The claim of this doc.** This is the vision-forward view: the fully integrated future state in which the proposal's network engine runs *inside* SBFoundation. Everything network-specific here is `PROPOSED` — it is the PhD. What is deliberate is that each proposed component attaches to an existing, `BUILT` platform seam rather than a green field, and follows a grammar the platform has used a dozen times.

13.1. The target architecture

```
flowchart LR
    subgraph BUILT["Built today"]
      A["Gold<br/>point-in-time"] --> B["Factor layer<br/>fact_factor_value"]
    end
    subgraph PROPOSED["Proposed research engine"]
      C["Supra-Adjacency<br/>Tensor Builder"] --> D["ST-GNN / HGT<br/>message passing"]
```

```
    D --> E["Spectral Analyzer<br/>Supra-Laplacian eigenspectrum"]
    E --> F["Phase-transition<br/>early-warning signal"]
  end
  B --> C
  G["Supply-chain layer (L2)"] -.-> C
  H["13F ownership layer (L3)"] -.-> C
  F --> I["Report-only sidecar<br/>+ dashboard card"]
```

Solid = built and feeding the engine; dashed = proposed data layers (doc 03); the whole `PROPOSED` block is the research.

13.2. Mapping the proposal's subsystems onto platform seams

Proposal subsystem (§3)	Proposed home	Attaches to (built seam)	Status
Data Ingestion Engine (L1/L2/L3)	Keymap datasets + EDGAR/FINRA ingestion	`sbbronze`/`sbsilver`/`sbgold` (L1 EDGAR client (F-319)	PARTIAL (L1 built)
Supra-Adjacency Tensor Builder	New leaf pkg (e.g. `sbnetwork`) reading Gold	`gold.fact_factor_returns`, `gold.fact_eod`	`PROPOSED`
Geometric Deep-Learning Framework (ST-GNN / HGT)	New engine pkg on PyTorch Geometric	consumes the tensor; runs as a research-day producer	`PROPOSED`

Proposal subsystem (§3)	Proposed home	Attaches to (built seam)	Status
Physics Evaluation Engine (Supra-Laplacian spectrum)	Spectral producer → `ops.research_*` + sidecar	the `sbcrowding` report-only producer pattern	`PROPOSED`

13.3. Why the mount points are credible

The platform has a **repeated, boring grammar** for adding an analytical producer, and the network engine would follow it exactly:

1. **One method per package.** `sbpbo`, `sbic`, `sbpermtest`, `sbcrowding` each isolate a single statistical method. A `sbnetwork` / `sbspectral` leaf fits this precedent.
2. **Report-then-enforce.** New signals ship **report-only** first — computed, persisted, surfaced, but gating nothing — behind a default-OFF switch. The Supra-Laplacian early-warning signal would start as pure observability, never touching portfolio construction until evidence earns it.
3. **`ops.research_*` → sidecar → dashboard card.** Every producer writes an ops table, emits a per-run JSON sidecar, and renders one dashboard view. `sbcrowding` (doc 05) is the literal template.
4. **The X7 boundary is respected.** Portfolio and allocation are compliant-by-construction, *not* optimizers (the `universe-and-allowlist` decision-record family). A network signal enters as **risk observability**, not as a covariance-optimizer input — keeping the research clear of a settled architectural stance (see the DY-connectedness findings §2.2).

13.4. The one genuinely new dependency

The engine introduces `torch` / `torch-geometric` (proposal §4) — a real, GPU-accelerated dependency the platform does not carry today. It would live *only* in the research engine package, isolated from the nightly production path, and run on research-day cadence rather than every night. That isolation is itself a design decision the roadmap (doc 07) makes explicit.

Cross-links: proposal §3 · crowding groundwork · gap & roadmap · architecture packages

14. Gap Analysis & Research Roadmap

Author: Todd B. Adams **Reinforces:** Proposal §3 & §5 · **Reading order:** doc 07 of the research-platform package

> **The claim of this doc.** Here is the honest ledger of what is built versus what the PhD proposes, expressed as an increment plan in the platform's own idiom. Nothing below hides a gap; the point is that each increment is scoped against an existing seam, and the earliest, cheapest increment is the one that most de-risks the research.

14.1. Built vs. proposed — the ledger

Capability	Proposal role	Status	Where
Point-in-time data + Gold model	Feeds all tensors	`BUILT`	doc 02
Stock–factor bipartite layer (L1)	Multiplex Layer 1	`BUILT`	doc 04

Capability	Proposal role	Status	Where
Correlation / crowding / factor-networks	Layer-3 proxy, risk observability	BUILT	doc 05
FF5+UMD, AQR, FRED reference series	Benchmarks / regime covariates	BUILT	doc 03
Validation gauntlet (MCPT/PBO/DSR/purged-CV)	Anti-overfitting for network claims	BUILT	doc 08
EDGAR + FINRA ingestion machinery	Sourcing path for L2/L3	PARTIAL	doc 03
Supply-chain layer (L2)	Directed unipartite stock–stock	PROPOSED	—
13F ownership layer (L3)	Bipartite fund–stock	PROPOSED	—
Supra-Adjacency tensor builder	§3 subsystem 2	PROPOSED	doc 06
ST-GNN / HGT engine	§3 subsystem 3	PROPOSED	doc 06
Supra-Laplacian spectral analyzer	§3 subsystem 4 (early warning)	PROPOSED	doc 06

14.2. The increment plan

Ordered so the earliest step is the cheapest and most decisive — the platform's standard "spike before you build" discipline (the DY-connectedness spike is the precedent: it killed the weakest hypothesis for the price of a script).

1. **Increment A — Spectral observability producer (report-only).** A `sbnetwork/sbspectral` leaf that assembles the *built* Layer-1 factor graph (plus the crowding correlation network) into a single-layer Laplacian and publishes its algebraic connectivity / spectral radius as a report-only sidecar. Reuses the `sbcrowding` template end-to-end. *De-risks the core physics claim on real data before any GNN or new dataset.*
2. **Increment B — Supply-chain layer (L2).** Extend the EDGAR client from fundamentals verification to 10-K customer–supplier edge extraction → a directed stock–stock dataset. Largest data-engineering item.
3. **Increment C — 13F ownership layer (L3).** Reuse the EDGAR client for 13F holdings → a bipartite fund–stock dataset, replacing the comomentum crowding *proxy* with holdings ground truth.
4. **Increment D — Multiplex tensor + ST-GNN engine.** Assemble Layers 1–3 into the Supra-Adjacency tensor; introduce the isolated `torch-geometric` engine on research-day cadence.
5. **Increment E — Phase-transition evaluation & (eventually) a gate.** Validate the multi-layer spectral early-warning signal through the full gauntlet (doc 08); only after survived evidence does it inform anything downstream (report-then-enforce).

14.3. Honest risks, stated up front

- **The core hypothesis may not hold.** The platform's own spike found single-layer connectedness *coincident, not leading* (findings §6). Increment A is designed to test whether the *multi-layer spectral* version does better — and to fail cheaply if it does not.
- **L2 supply-chain data is noisy.** 10-K relationship extraction is an NLP problem with real precision limits; the roadmap treats it as a research task with its own evaluation, not a solved ingestion.
- **The GNN dependency is heavy.** `torch-geometric` is quarantined to one research-day package (doc 06) so it never touches the nightly production path.

Cross-links: proposal §3/§5 · target architecture · DY-connectedness spike · the internal `epics` roadmap directory

15. Validation & Anti-Overfitting

Author: Todd B. Adams **Reinforces:** Proposal §6 — Model Proofing · **Reading order:** doc 08 of the research-platform package

> **The claim of this doc.** GNN-in-finance research has a reputational problem: it overfits, and it is rarely held to the multiple-testing and out-of-sample standards the asset-pricing literature demands. SBFoundation brings a production-grade, literature-calibrated validation gauntlet that any new producer — including a network one — must pass. This is the platform's single biggest contribution to the *credibility* of the proposal's results.

15.1. The gauntlet already in production — `BUILT`

The platform's "mathematics of trust" is not aspirational; it gates real promotions today. Each check is adopted from the literature, not invented. Full narrative: whitepaper §5.

Check	Question it answers	Grounded in
Information Coefficient hurdle	Does the signal actually predict?	Grinold & Kahn, *Fundamental Law*

Check	Question it answers	Grounded in
Multiple-testing / false-discovery	Did I get lucky across many tries?	Harvey, Liu & Zhu (2016); Benjamini–Yekutieli (2001)
Permutation / Monte-Carlo (MCPT)	Could randomness produce this?	White (2000) Reality Check; the "Masters" suite
Survivorship haircut	Are dead companies flattering it?	Shumway (1997); Brown et al. (1992)
Net-of-cost + capacity	Does the edge survive trading?	Corwin–Schultz (2012); Novy-Marx & Velikov (2016)
Deflated Sharpe + PBO + lockbox	Am I overfitting by tuning?	Bailey & López de Prado (2014); López de Prado (2018)
Purged + embargoed CV	Is my out-of-sample actually clean?	López de Prado (2018)
Honest family-wise trial count	Deflated against how many trials, really?	platform's cross-axis trial ledger (F-321)

15.2. Why this is decisive for the proposal

1. **The synthetic simulator graduates into the gauntlet.** The proposal's §6 simulator proofs the physics on 100 synthetic stocks by *injecting* a known shock and confirming the brittleness index moves. That is a necessary first step — but a synthetic positive is not evidence of a real edge. On this platform, the network signal's empirical results would be run through MCPT, PBO, deflated-Sharpe and a lockbox, so a "phase-transition detector" claim is tested for luck, tuning, and leakage before it is believed.

2. **Multiple-testing rigor a GNN paper rarely applies.** A spatial-temporal GNN has enormous architectural degrees of freedom (layers, heads, windows, couplings). The platform's honest trial-count deflation (F-321) and PBO are exactly the instruments for "best-of-many-architectures" — the dominant failure mode of deep-learning finance.
3. **Report-then-enforce keeps unproven signals harmless.** A network early-warning signal ships report-only first (doc 06); it cannot move a book until it has survived attributed evidence — the same discipline that keeps every experimental factor out of real capital.
4. **The platform is willing to publish a negative.** The DY-connectedness spike (findings) is a documented, honest falsification. That culture — disclosure over avoidance — is what makes a *positive* result from the same platform worth trusting.

15.3. The standard the network claim must clear

For a Supra-Laplacian early-warning signal to be reported as real, it would need to show — on the platform's empirical universe, not a synthetic panel — a signal that survives permutation, deflates under honest trial counts, holds out-of-sample through a lockbox, and does so *net of the composition-drift and coincidence artifacts the spike already surfaced*. That is a high bar by design, and clearing it is precisely what would make the proposal's contribution defensible.

Cross-links: whitepaper §5 — mathematics of trust · reference papers · proposal §6

16. Reproducibility & Experiment Infrastructure

Author: Todd B. Adams **Reinforces:** Proposal §3 — pipeline & evaluation · **Reading order:** doc 09 of the research-platform package

> **The claim of this doc.** Doctoral research is judged partly on reproducibility. On SBFoundation every run is an immutable, identity-stamped, fully audited experiment — not a notebook re-run whose state is lost. A network producer added to the platform inherits this audit trail automatically, so its results are reproducible by construction.

16.1. Every run is an experiment record — `BUILT`

Property	What it gives the research	Where
One `run_id` per run (`YYMMDD_XXXXXX`)	A single identity stamped end-to-end; every artifact traces to one run	run analysis

Property	What it gives the research	Where
Immutable inputs/outputs	A run's data and results do not change after the fact	`architecture`
Per-stage JSON sidecars	Each producer emits a structured, versioned result blob (schema-versioned)	run-type task reference
Harvested run bundle	All sidecars rolled into one `run_<id>.json` the dashboard reads	run analysis
Reharvest	Re-derive a past run's bundle against current Gold — reproduce a result on demand	run analysis
Operator dashboard (SPA)	Every run, stage, factor, and signal browsable	SPA components
Run modes	`NIGHTLY` (monitoring) vs `NIGHTLY_FULL` / `RESEARCH_DAY` (deep evidence)	run-type task reference

16.2. Provenance, not just results

The platform records *what it received and when*, not only what it concluded:

- **Audit-first ingestion.** Every vendor fetch — including failures — is recorded in Bronze, so "what did we actually have on date X?" is always answerable (Bronze contract). For a phase-transition study,

being able to reconstruct the exact information set available on a historical date is essential and non-negotiable.

- **Point-in-time knowledge dates.** Fundamentals and slow-cadence factors are snapped to the market day their information became knowable, with the leak audited against SEC EDGAR ground truth (EDGAR PIT findings).
- **Schema-versioned sidecars.** Producers version their output so a downstream reader degrades gracefully across runs of different vintage — the network producer would do the same.

16.3. A disciplined change process behind every producer

The platform's own SDLC (project-sdlc) is itself part of the reproducibility story: features are specified as design briefs, decomposed into tracked backlog tasks, built test-first, and validated through tiered gates before merge. A research increment (doc 07) enters through the same process, so the code that produces a published result is reviewed, tested, and traceable to a specification — not a one-off script.

16.4. What this means for the proposal

The proposal's evaluation engine (§3, subsystem 4) needs to compare a network signal against realized stress across many historical dates and re-run cleanly as the model changes. On this platform that comparison is a producer whose every run is identity-stamped, sidecar-recorded, and reharvestable — so a reviewer (or a future you) can reproduce any reported number, and the model's evolution is a legible history rather than a lost sequence of notebook states.

---

16. Reproducibility & Experiment Infrastructure

Cross-links: run analysis · project SDLC · SPA components · proposal §3

17. Literature Bridge

Author: Todd B. Adams **Reinforces:** Proposal §2 — Theoretical Foundations · **Reading order:** doc 10 of the research-platform package

> **The claim of this doc.** The proposal's bibliography and the platform's own reference library are one continuous body of work, not two disconnected reading lists. This doc reconciles them and states the single convention that keeps every cited paper captured.

17.1. The proposal's §2 papers — bundled in `docs/phd/articles/`

The network / multilayer / GNN literature that grounds the proposal is downloaded (open-access) with per-paper summaries in `docs/phd/articles/`:

Paper	Role in the proposal
Musmeci, Aste & **Di Matteo** (2015)	KCL information-filtering-network methodology (supervisor lineage)
Bardoscia, Battiston, Caccioli & Caldarelli (2017)	Mechanics of cascading failure / systemic risk

Paper	Role in the proposal
Boccaletti et al. (2014)	Definitive multilayer-network formulation
Kivelä et al. (2014)	Heterogeneous vs. multiplex terminology
Kipf & Welling (2016)	GCN message-passing foundation
Yu, Yin & Zhu (2017)	Spatio-temporal GCN, adaptable to financial series
Caldarelli (2007)	Scale-free network physics (monograph, citation-only)

17.2. The platform's library — `docs/reference-papers/`

The asset-pricing, cost, survivorship, overfitting and crowding literature that grounds the *platform's* validation gauntlet lives in `docs/reference-papers/README.md`. The overlap with the proposal is the bridge:

- **Lou & Polk (2012), *Comomentum*** (row 74) — already bundled; grounds the platform's crowding producer *and* the proposal's Layer-3 ownership-crowding motivation (doc 05).
- **Harvey, Liu & Zhu (2016); Bailey & López de Prado (2014); Benjamini–Yekutieli (2001)** — the anti-overfitting core (doc 08) that would discipline the network claims.

17.3. One gap to close, and the convention that governs it

- **Raddant & Di Matteo (2023), *A look at financial dependencies by means of econophysics and financial economics*** — the supervisor's survey the platform already stress-tested (findings) — is **not yet captured** in `docs/reference-papers/`. It is open-access (CC-BY) and should be bundled as `Raddant2023-financial-dependencies-econophysics.pdf` when the first network `/design-brief` proceeds. Flagged here so it is not missed.

The convention (platform rule, [CLAUDE.md §7]): every academic paper cited *anywhere* in the repo must be captured in `docs/reference-papers/` — an index row plus a bundled open-access PDF (or a "paywalled — citation only" note). The proposal's `docs/phd/articles/` bundle is the network-literature satellite of that same shelf. As the research proceeds, new network/GNN citations land in one of the two homes, never uncaptured.

17.4. How the two reading lists meet

The proposal contributes the **network-physics + geometric-deep-learning** half; the platform contributes the **asset-pricing rigor + crowding** half. The PhD sits exactly at their intersection — multiplex network structure, evaluated with asset-pricing-grade anti-overfitting discipline, on a live executable platform. That intersection is the whole argument of this folder.

Cross-links: proposal §2 · phd/articles · reference papers · crowding groundwork

Part III.

Findings

18. Asset-Dependency Networks (Econophysics) — Paper Review & Platform Fit Findings

Date: 2026-07-30 **Reviewer:** initial-thinking pass (operator-requested) **Source paper:** Raddant, M. & Di Matteo, T. (2023). *A look at financial dependencies by means of econophysics and financial economics.* Journal of Economic Interaction and Coordination 18:701–734. DOI 10.1007/s11403-023-00389-6. Open access (CC-BY 4.0). **Question posed:** Does the core idea of introducing a *connected network of assets* improve the platform's model, and how much change would it require?

Update 2026-07-30: an empirical research spike was run against `gold.fact_eod` to pressure-test the one recommended path (Tier A — Diebold–Yilmaz connectedness as a *leading* regime signal). **Result: the leading-signal hypothesis failed** — see §6. This changes the recommendation: do **not** build the Tier-A time-series connectedness index expecting to anticipate stress. Details below.

> This is a **reasoned-analysis + research-spike findings doc**, not a producer or a validated edge. No production code was changed (read-only spike). It exists to inform a go/no-go on a future `/design-brief`.

18.1. 1. What the paper actually is

A **2023 survey/review** — not a single method with a backtested edge. It tours the econophysics + financial-economics toolkit for modeling the **N×N dependency structure among assets** as a network. Load-bearing techniques it catalogs:

Technique	What it produces	Maturity in the paper
Correlation matrix (Pearson, exp-weighted, partial, rank/tail)	dense N×N dependency	foundational
Random Matrix Theory / PCA	denoised covariance, "market mode" + group modes	foundational
Information filtering networks — MST, PMFG, TMFG + DBHT clustering	sparse graph (N−1 or 3(N−2) edges) + endogenous clusters	the paper's centerpiece
Multivariate GARCH (DCC/BEKK/DECO)	*dynamic* conditional covariance	mature but hard to scale
Granger-causality / pairwise de-GARCHed regression	directed lead-lag network	**explicitly weak** — Billio et al.: significant links barely exceed the 5% false-positive rate except during crises
Diebold–Yilmaz variance decomposition / TVP-VAR	directed **spillover / connectedness** network → systemic-risk proxy	the most cited *applied* one

Crucial framing: none of these are presented as alpha. The paper's own

applications are risk topology, systemic risk, comovement/segmentation, and covariance denoising — descriptive structure, not return prediction.

18.2. 2. Does the "connected network of assets" improve the platform's model?

It depends entirely on **which of three distinct value propositions** you mean. Conflating them is where this kind of idea usually goes wrong. Two of the three are a poor fit for *this* platform specifically.

18.2.1. 2.1 As an alpha source (network structure → return prediction) — Recommend against

The only genuinely return-predictive angle is economic-links / lead-lag momentum (Cohen–Frazzini, Granger networks). The paper itself flags it as regime-dependent and barely-above-noise outside crises. The platform already has a rigorous factor→signal→ranked-cross-section pipeline with an honest verdict that purged-CV IC is 0 on the current predictor set; a weak, fragile network-alpha signal buys little and adds a lot of surface.

18.2.2. 2.2 As a portfolio-optimizer input (denoised/filtered covariance → weights) — Recommend against, on principle

This is the classic RMT + TMFG-LoGo payoff: a better-conditioned Σ for mean-variance optimization. But the architecture has a **repeatedly reaffirmed settled decision (X7)**: `sbportfolio` and `sballocation` are *compliant-by-construction*, **not optimizers** — no covariance inversion,

no objective function. Adopting the covariance-optimization payoff would mean silently reversing a deliberate architectural stance. Do not let a survey paper relitigate X7 as a side effect.

18.2.3. 2.3 As a risk / crowding / systemic-observability layer — This is the real fit, and it's a good one

Where the concept genuinely earns its place — *and where the platform is already leaning*:

- **`sbcrowding` (FC-1 comomentum) is already a correlation-network statistic** — abnormal within-decile return correlation among the momentum family (`comomentum_crowding_service.py` + `_comomentum_math.py`, ~600 LOC). The paper is the general theory of exactly that move.
- **`sbriskmodel`** already carries a factor covariance + idiosyncratic variance model (`factor_covariance_service.py`, `idio_variance_service.py`).
- **`sbic`** already builds a **factor-level** pairwise-| | network (`pairwise_correlation_servi`
  LH-6y / O-089).

So the network concept is **not foreign** — it exists at the *factor* level and in *crowding*. What's genuinely missing is an **asset-level dependency/connectedness read**: a market-wide "how correlated/fragile is the whole book right now" regime signal, plus network-based clustering to (a) sanity-check GICS-sector neutralization, and (b) detect concentration/crowding in the traded cross-section that sector labels miss.

The single highest-value, lowest-regret extraction from this paper is Diebold–Yilmaz total connectedness as a report-only systemic-risk / regime diagnostic, optionally paired with **TMFG + DBHT** clustering of the universe. That fits the platform's established grammar exactly: one-method-per-package, report-then-enforce, `ops.research_*` + sidecar → SPA, evidence-first, no gate at first (`sbcrowding` is the literal precedent).

18.3. 3. Effort to bring it in

Scoped in platform units, anchored on the `sbcrowding` precedent (~600 lines of math+service + a close-out producer + `ops.research_*` table + sidecar + SPA card). Three tiers:

Tier	Scope	Rough effort	Risk
A — Diebold–Yilmaz connectedness diagnostic (recommended MVP)	One new producer (extend `sbcrowding` or new `sbconnectedness` leaf): rolling VAR + generalized FEVD over a sector/index return panel → total & directional connectedness index → `ops.research_*` + sidecar → one SPA regime card. Report-only, no gate.	**~1 small epic / 4–6 tasks**, comparable to FC-1. Math is standard statsmodels VAR.	Low — no portfolio/gate coupling.

Tier	Scope	Rough effort	Risk
B — TMFG + DBHT universe clustering	Add filtered-graph + clustering producer over the correlation matrix; surface network-clusters vs GICS sectors as an integrity/observability panel; feed a crowding/concentration read on the traded book.	**+1 epic / 5–8 tasks.** The TMFG + DBHT graph algorithms are the only genuinely new machinery — no currently-vendored library does them, so hand-rolled (Massara et al. 2017 is the scalable reference).	Medium — algorithmic surface; needs its own tests.
C — Dynamic conditional covariance / network-aware sizing	DCC-GARCH covariance feeding diversification-aware sizing in `sballocation`.	**Large / multi-epic**, and **collides with X7.**	High — don't, unless the no-optimizer stance is being revisited deliberately.

18.4. 4. Recommendation

Original recommendation (pre-spike): do **Tier A only, as a report-only observability producer**, and gate any move toward B/C

on Tier A actually showing a connectedness signal that *leads* regime shifts.

Revised recommendation (post-spike, §6): the Tier-A leading-signal premise **did not hold** on this platform's universe — DY total connectedness is **coincident, not leading**, has a **saturated/narrow dynamic range** at sector granularity, **misses equity-specific selloffs**, and is contaminated by **composition drift**. So:

- **Do not build the Tier-A time-series connectedness index as a regime *predictor*.** Its marginal information over just watching realized vol is low, and it does not anticipate stress.
- If a network producer is still wanted, the surviving rationale is the **cross-sectional crowding/clustering** angle (Tier B — TMFG/DBHT on the *traded book*, "which of my current holdings are one correlated cluster"), **not** the market-wide TCI. That is a different, heavier build and should get its own spike before commitment.
- Everything worth having remains reachable **without touching `sbportfolio`, `sballocation`, or X7**; everything that would touch them is weak (alpha, §2.1) or against a settled decision (optimizer, §2.2). Unchanged.

Honest caveat (reaffirmed): this is a survey — **no published edge to import**. The spike confirms the concept's value here is at best coincident risk *legibility*, not P&L or early warning.

18.4.1. Suggested next step (pick one)

1. **Shelve the network work** — the cheapest spike killed the strongest hypothesis; a coincident-with-vol index isn't worth a producer. (Recommended default.)

2. **Tier-B spike** — if the crowding/clustering use still appeals, run a *cross-sectional* spike: TMFG/DBHT-cluster the current traded book and measure whether within-cluster concentration would have flagged real drawdown episodes. Only then consider a brief.
3. ~~Tier-A `/design-brief`~~ — **withdrawn**; §6 removes its justification.

18.5. 5. Reference-paper capture (convention reminder)

Per CLAUDE.md §7, any cited paper must be captured in `docs/reference-papers/` (surfaced in-book as the Reference Library). This paper is open-access (CC-BY); if a `/design-brief` proceeds, add a row to `docs/reference-papers/README.md` and bundle the PDF as `Raddant2023-financial-depe` Not done here (no code/brief committed yet) — flagged so it isn't missed at brief time.

18.6. 6. Empirical research spike (2026-07-30) — does DY connectedness *lead* stress?

Read-only spike against `gold.fact_eod`. Goal: de-risk Tier A before any brief by testing its load-bearing premise — that a Diebold–Yilmaz total connectedness index (TCI) would give an *early-warning* regime signal in this platform's own universe.

18.6.1. 6.1 Method

- **Panel:** sector-level daily equal-weight mean log return, built in DuckDB from `gold.fact_eod.adj_close` joined to `dim_instrument.sector_sk`. Cross-section restricted to liquid, non-ETF names (`adj_close` `$5`, `adv_dollar_20d_f` `$1M`, per-name return clipped to ±0.5 to guard split/data artifacts). Result: **5,698 trading days (2004-01-01 → 2026-07-29) × 11 GICS sectors.**
- **Estimator:** rolling 200-day **VAR(2)** → MA representation → **generalized (order-invariant) FEVD** (Diebold–Yilmaz 2012) at horizon H=10 → row-normalized → **TCI = 100 × off-diagonal mass / N**, stepped every 5 days (1,100 points).
- **Stress proxy:** 21-day rolling annualized realized vol of the equal-weight market (mean sector) return.
- **Tests:** (a) `corr(TCI_t, forward vol_{t+h})` for h = 0/10/21/42d — *rising* with h leads; *falling* coincident/lagging. (b) lead/lag cross-correlation of `ΔTCI` vs `Δvol`. (c) crisis-window peak TCI vs full-sample median.
- Reproduce: `scratchpad/dy_extract.py` (lock-polite extract) + `scratchpad/analyze_dy.py` (offline). Series: `dy_tci_series.csv`. Chart below.

18.6.2. 6.2 Results

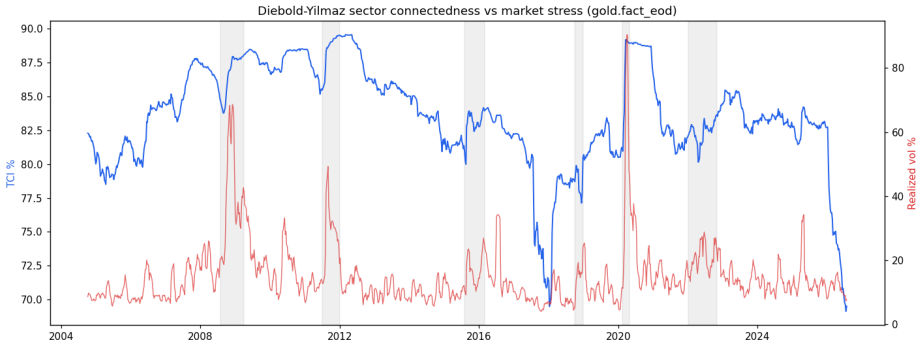


Figure 18.1.: DY sector connectedness vs market stress

- **TCI is coincident, not leading.** `corr(TCI_t, forward vol_{t+h})` = **+0.42 (h=0) → +0.38 → +0.31 → +0.24 (h=42d)** — monotonically *declining* with horizon, the opposite of a leading indicator. Cross-correlation of changes peaks at **lag 0 (+0.35)** and goes flat/negative at positive (TCI-leads) lags (−0.17 at +5d). The `ΔTCI` change-form has a weak forward blip at h=+10d (+0.29) but nothing that would time a regime.
- **Saturated, narrow dynamic range.** TCI mean **83.6%**, band **69–90%**. At sector granularity sectors are ~84% connected *always*; crises add only a few points.
- **Flags the big systemic crises, misses equity-specific ones.** Peak-vs-median: GFC **+4.5**, Euro-2011 **+6.0**, COVID **+5.7** — but Q4-2018 **−3.0** (below median), 2015-16 **+0.7**, 2022 rate-shock **+0.1**. And the systemic peaks land at the crisis *bottom* (GFC peak 2009-03-26; COVID peak 2020-03-25) — coincident/lagging, not early.
- **Composition drift contaminates the level.** Large multi-year TCI swings unrelated to stress (secular decline 2012→2018, a structural step ~2017-18, a sharp drop at the very end of sample) track changing cross-section membership/coverage more than

market regime — a robustness problem for any thresholded use.

18.6.3. 6.3 Verdict

The spike **kills the Tier-A leading-signal hypothesis.** DY connectedness here is a coincident echo of realized vol with low marginal information, blind to equity-specific selloffs, and sensitive to universe composition. It is not worth a producer as an early-warning/regime signal. This is the intended outcome of a cheap spike: the strongest recommended path was falsified before any code shipped.

Scope honesty: this tests the *time-series/systemic* use (Tier A) only. It does **not** test the *cross-sectional crowding/clustering* use (Tier B — network centrality of individual holdings), which remains unexamined and is the only surviving rationale for network work here — pending its own spike (§4, next-step 2). Also unswept: VAR lag order, window length, FEVD horizon, and alternative stress definitions (drawdown/VIX) — but the coincident signature is strong and consistent enough that tuning is unlikely to reverse it.

19. SEC EDGAR Point-in-Time Filing-Date Integrity Audit of FMP Fundamentals — Findings (F-319)

Status: empirical verdict **recorded 2026-07-30** from a live SEC EDGAR fetch (sample AAPL / MSFT / JPM / XOM / JNJ). The first live run (2026-07-27) reported a spurious "185/185 @ ~−210 days" that was an **instrument bug**, not a platform leak — the loader reconstructed quarterly period-ends from a calendar-quarter assumption while `gold.fact_fundamental_quarter` stores *fiscal* `calendar_year`/`period`, so non-December-FYE names (MSFT, June FYE) mis-paired to the wrong EDGAR filing. Fixed under **B-319.1 / TASK-2791** (derive period-end from the row's own `period_date_sk`); the numbers below are the **corrected** re-run. The harness remains read-only and re-runnable; re-run after each data refresh via §5.

19.1. 1. Verdict

LOOK-AHEAD DETECTED — 184 of 184 matched fundamentals carry a knowledge date earlier than their EDGAR 10-K/10-Q filing date. Panel rows evaluated: 1181 (matched 184, unmatched 997, violations 184). Median violation gap **−49 days**; range roughly **−45**

to −53 days. The leak is **systematic and appears in both cadence legs**, exactly as the a-priori prediction below anticipated:

- **Annual** (`earnings_yield` / `pe_ratio` / `roic`, from `fact_valuation_annual` / `fact_moat_annual`): stamped at the F-292 **Jan-1 snap** (e.g. `20190102`) but the 10-K was filed ~late Feb (e.g. 2019-02-20) → ~−49 days. **Prediction confirmed.**
- **Quarterly** (`earnings_growth_qoq`, from `fact_fundamental_quarter`): stamped at **period-end** (e.g. 2020-12-31) with **no dissemination lag**, filed ~2021-02-22 → −53 days. This **refutes** the a-priori guess that the quarterly leg was safer: `TABLE_DATE_SK_EXPR["fact_fundamental_quarter"] = COALESCE(period_date_sk …)` stamps knowledge at the fiscal period-end, and the keymap `min_age_days` lag does **not** propagate into that knowledge `date_sk`.

Two caveats bound how alarming this is (read before acting):

1. **Anchor = the 10-K/10-Q `filed` date, but preliminary figures usually reach the market ~2–3 weeks earlier via an 8-K earnings release.** So **−49/−53 is an *upper bound*** on the economically-relevant look-ahead. Even measured against the earlier 8-K, though, a *period-end* stamp (Dec-31) and a *Jan-1* snap still precede the announcement — a non-trivial leak remains, it is just smaller than the headline. **F-324.1 (2026-08-01) wired this in code:** the audit now fetches 8-K Item 2.02 earnings releases (`EARNINGS_RELEASE_FORM` / `parse_earnings_ releases` / `pair_earnings_release`), pairs the earlier of {8-K release, definitive filing} as the true first-availability date, and the report now carries BOTH `gap_days` (vs definitive, the upper bound) and `gap_days_vs_ release` (the tighter estimate). The **tighter empirical numbers land on the next live `--live` run** (a Tier-4 operator step — the code + offline fixtures are delivered and unit-tested; the live 8-K fetch has not yet been run).
2. **Small / partial coverage.** 184 matched of 1181 (997 unmatched — only 5 tickers, only 10-K/10-Q forms, a 20-day pairing tolerance).

The verdict is "100% of what could be verified," not "100% of the universe." The as-reported value-agreement leg was **0 comparable of 0** because the panel concepts audited (`earnings_yield` / `pe_ratio` / `roic` / `earnings_growth_qoq`) are DERIVED ratios with no single us-gaap XBRL tag. **F-324.1 addressed both halves:** (a) the sample is widened to 15 tickers mixing fiscal-year-ends (`SAMPLE_TICKERS` now includes MSFT-Jun / NKE-May / WMT-Jan / … alongside the Dec-FYE names) so the non-Dec-FYE pairing path is exercised; (b) a FMP us-gaap concept map (`FMP_CONCEPT_TO_US_GAAP` / `resolve_us_gaap_concepts`) is wired, two RAW as-reported sources with real twins (`fact_annual.revenue → Revenues`, `fact_annual.net_income → NetIncomeLoss`) are added to the panel loader so the value-agreement leg can produce non-zero comparable evidence, and the four derived-ratio concepts are now reported explicitly as **"no us-gaap twin"** rather than silently as 0-of-0. Non-zero comparable counts land on the next live run.

Reasoned a-priori prediction (retained for the record; now confirmed for the annual leg, refuted for the quarterly leg). The platform's F-292 cadence-aware knowledge-date snap stamps annual (10-K-sourced) fundamentals at *the first US market day on/after Jan-1 of the following calendar year* (`sbfactors.factor_value_lift_service` / `resolve_date_sk_expr`). A 10-K is actually **filed ~60–90 days after fiscal year-end** — late Feb–Mar for the December-FYE population — so the annual leg was predicted to leak by ~40–90 days (confirmed at ~49). The quarterly leg was predicted *safer* on the assumption that the keymap `min_age_days` dissemination lag pushed the knowledge date past the period; the live run shows that lag never reaches the quarterly knowledge `date_sk`, so the quarterly leg leaks too.

19.2. 2. Method

Read-only, stdlib-only (`urllib` / `json` / `dataclasses` / `datetime`); no ingestion, no Gold write, no new runtime dependency.

- **CIK resolution** — EDGAR `company_tickers.json` parsed into a `{TICKER: CIK}` map (`parse_ticker_cik_map` / `resolve_cik`), then zero-padded to the 10-digit path token (`zero_pad_cik`).
- **Filing dates** — the submissions API (`data.sec.gov/submissions/CIK…json`) `filings.recent` parallel arrays zipped into `EdgarFilings` (`parse_recent_filings`), carrying `form` / `filed` / `acceptanceDateTime` / `accessionNumber` / `reportDate`.
- **As-reported values** — the XBRL companyfacts API (`data.sec.gov/api/xbrl/comp` `facts → us-gaap → <Concept> → units → USD`, parsed into `AsReportedFacts` (`parse_company_facts`).
- **Pairing** — a panel row pairs to a filing when `abs(period_of_report - period_end) 20 days`; the **earliest-filed** match wins (first public disclosure, the conservative choice). `period_of_report` / `period_end` are used **only to pair** — never for the leak test itself.
- **The leak test** — knowledge `date_sk` (decoded to a date) vs the paired filing's EDGAR `filed` date. `gap_days = knowledge_date - edgar_filed`; a **negative** gap is a look-ahead violation. The leak is always anchored on the **filing date, never period-end** — the F-288 grain lesson.
- **As-reported agreement** — the platform's stored value vs EDGAR's as-reported value, material when the relative difference exceeds `MATERIALITY_REL_THRESHOLD = 0.01` (1%).

Sample (`SAMPLE_TICKERS` / `AUDITED_CONCEPTS` / `AUDITED_FORMS`): the 2026-07-30 verdict above used a 5-name cross-section (AAPL, MSFT, JPM, XOM, JNJ); **F-324.1 widened `SAMPLE_TICKERS` to 15 mixed-FYE names** (adding NKE-May, WMT/HD/COST/TGT-Jan/Feb, PG, KO, CSCO, ORCL, PEP) now that OQ-3's "widen only if leakage is found" is satisfied. Filings audited: recent 10-K + 10-Q for the

leak test, plus 8-K Item 2.02 earnings releases for the first-availability anchor (F-324.1); us-gaap concepts `Revenues`, `NetIncomeLoss`, `Assets`, `StockholdersEquity`.

19.3. 3. Worst-offender table

Corrected live run, 2026-07-30 (top 20 of 184 violations, most-negative `gap_days` first). Every worst-offender is **JNJ** — a December-FYE filer that files relatively late (~Feb 20), giving the widest gap; the other sampled names also violate, with smaller gaps. `knowledge_date_sk` at the fiscal period-end (quarterly) or the Jan-2 snap (annual) sits ~7 weeks before `edgar_filed`.

> **F-324.1 (2026-08-01):** this table pre-dates the 8-K anchor. The report now emits two additional columns — `edgar_first_available` (the earlier of the 8-K Item 2.02 earnings release and the definitive filing) and `gap_days_vs_release` (the tighter, less-negative gap) — so a JNJ row filing its 8-K ~2 weeks before its 10-K would show a `gap_days_vs_release` roughly ~14 days less negative than the `gap_days` shown here. The widened columns + the tighter numbers repopulate on the next live `--live` run (Tier-4). The rows below are the vs-definitive upper bound and remain valid as such.

symbol	concept	source_table	period_end	knowledge_date_sk	edgar_filed	gap_days
JNJ	earnings_growth_qoq	fact_fundamental_quarter	2020-12-31	20201231	2021-02-22	-53
JNJ	earnings_growth_qoq	fact_fundamental_quarter	2018-12-31	20181231	2019-02-20	-51
JNJ	earnings_growth_qoq	fact_fundamental_quarter	2019-12-31	20191231	2020-02-18	-49

symbol	concept	source_table	period_end	knowledge_edgar	filed_date	gap_days
JNJ	earnings_yield	fact_valuation_annual	2018-12-31	20190102	2019-02-20	-49
JNJ	earnings_yield	fact_valuation_annual	2020-12-31	20210104	2021-02-22	-49
JNJ	pe_ratio	fact_valuation_annual	2018-12-31	20190102	2019-02-20	-49
JNJ	pe_ratio	fact_valuation_annual	2020-12-31	20210104	2021-02-22	-49
JNJ	roic	fact_moat_annual	2020-12-31	20210104	2021-02-22	-49
JNJ	roic	fact_moat_annual	2018-12-31	20190102	2019-02-20	-49
JNJ	earnings_growth	fact_fundamental_quarterly	2021-12-31	20211231	2022-02-17	-48
JNJ	earnings_growth	fact_fundamental_quarterly	2022-12-31	20221231	2023-02-16	-47
JNJ	earnings_growth	fact_fundamental_quarterly	2023-12-31	20231231	2024-02-16	-47
JNJ	earnings_yield	fact_valuation_annual	2019-12-31	20200102	2020-02-18	-47
JNJ	pe_ratio	fact_valuation_annual	2019-12-31	20200102	2020-02-18	-47
JNJ	roic	fact_moat_annual	2019-12-31	20200102	2020-02-18	-47
JNJ	earnings_growth	fact_fundamental_quarterly	2024-12-29	20241229	2025-02-13	-46
JNJ	earnings_growth	fact_fundamental_quarterly	2025-12-28	20251228	2026-02-11	-45
JNJ	earnings_yield	fact_valuation_annual	2021-12-31	20220103	2022-02-17	-45
JNJ	earnings_yield	fact_valuation_annual	2023-12-31	20240102	2024-02-16	-45

symbol	concept	source_table	period_end	knowledge_date	edgar_filed_date	gap_days
JNJ	pe_ratio	fact_valuation_annual	2021-12-31	20220103	2022-02-17	-45

19.4. 4. Coordination boundary (F-313 / SBM-7)

F-319 owns **only** the point-in-time filing-date + as-reported-value integrity leg — *is a fundamental usable before it was filed, and does its stored value match EDGAR's?* The EDGAR **delisting-confirmation** leg (positive confirmation of a `performance` vs `benign` delist via EDGAR) is a **separate concern owned by `survivorship-bias-measurement` F-313 / SBM-7**. F-313 was unbuilt at F-319's authoring, so this script carries its **own** stdlib EDGAR fetcher + CIK-resolution surface; any shared EDGAR fetch / CIK-resolution surface must be **reused, not forked** — if/when F-313 lands, the two fetchers should be consolidated into one shared EDGAR access module rather than maintained in parallel.

19.5. 5. How to run

Read-only, re-runnable, no gate:

```
# Live fetch (Tier-4 operator step; caches snapshots for reproducibility):
.venv\Scripts\python.exe scripts\research\edgar_pit_audit.py --live --out docs\research\edga

# Default (re-runs against the already-cached local snapshot):
.venv\Scripts\python.exe scripts\research\edgar_pit_audit.py
```

`--sample AAPL,MSFT,...` overrides the audited tickers; `--snapshot-dir` overrides the local cache (default `data/reference/edgar_snapshots`);

`--out` writes the report markdown (default stdout). SEC requires a declared User-Agent on every request — already set in `EDGAR_USER_AGENT`. The DuckDB connection is opened `read_only=True`; if the DB or the live fetch is unavailable the script prints a clear INDETERMINATE report and exits 0 (it is a diagnostic, not a gate). Transcribe the verdict + worst-offender rows from the live run into §1 / §3 above.

19.6. 6. Follow-ups (raised by the 2026-07-30 result)

The corrected run turns the leak from a hypothesis into a measured fact, but two things must happen before it can be acted on with confidence — tracked under the follow-up milestone **F-324** (`m-357`):

1. **Widen the anchor to 8-K earnings-release dates — DELIVERED in code (F-324.1 / TASK-2802, 2026-08-01).** The audit now fetches 8-K Item 2.02 earnings releases, pairs the earlier of {8-K release, definitive filing} as the true economic first-availability date, and reports both the conservative `gap_days` (vs definitive) and the tighter `gap_days_vs_release`. The sample was widened to 15 mixed-FYE tickers and the FMP us-gaap concept map wired (two raw as-reported sources added) so the value-agreement leg produces evidence. **Remaining: the live `--live` re-run** (Tier-4) to populate §1/§3 with the tighter empirical numbers + non-zero comparable counts. The B-319.1 fiscal `period_date_sk` period-end derivation is preserved.

2. **Scope the knowledge-date dissemination-lag remediation — F-324.2 / TASK-2803 (design brief pending).** The honest fix is to push the knowledge `date_sk` to on-or-after the actual filing/announcement date — a dissemination lag on the quarterly period-end stamp and a move of the annual Jan-1 snap onto the real filing date, mirroring F-299's FINRA `min_age_days` snap. **This is not a quick patch:** changing the knowledge date rewrites every

downstream PIT join and **forks factor history**, so it needs its own design brief + report-then-enforce switch + Tier-4 bring-up, not an inline change. The lag size should be chosen from F-324.1's 8-K-anchored numbers once the live re-run lands.

Remediation delivered as F-442 / LH-29 (default-OFF switch `SB_FUNDAMENTALS_DISSEMINATION_LAG_ENABLED`, report-then-enforce): a fixed proxy lag = period-end + N calendar days snapped forward to the first US market day — **quarterly 60d / annual 90d** — applied to `fact_fundamental_quarter`, `fact_fundamental_annual`, `fact_valuation_annual`, `fact_moat_annual` (superseding the F-292 Dec-31/Jan-1 annual snap when ON), with the signal panel + coverage rerouted through the switched expression (flipping ON forks `model_id`). Tier-4 re-lift CLI `python -m sbfactors.cli relift-fundamentals-factors [--all]`. OFF is byte-identical. Brief: `docs/backlog/feature-fundamentals-dissemination-lag-design-brief.md`.

Part IV.

Methodology & Experiments

20. Methodology

This page indexes where each part of the methodology already lives rather than restating it — the detail is maintained in `research-platform/` and `phd/`, generated from or grounded in the platform itself.

20.1. Data sources

- Data Catalog — provenance, licensing, refresh schedules for every dataset the research draws on.
- The multiplex data substrate — whether the platform can source the proposal's three network layers (factor exposure, supply-chain, institutional ownership).
- PhD proposal §"Data & reproducibility" — public and commercial sources named for the research specifically (French data library, CRSP-like sources, SEC 13F, SEC 10-K parsing).

20.2. Statistical methods

- Factor models & benchmarks — factor loadings & returns, the CAPM decomposition, systematic vs. idiosyncratic return, benchmark selection.
- The factor layer as multiplex Layer 1 — the bipartite stock–factor layer as built, not hypothetical.

- Systemic-risk & crowding groundwork — prior network work (comomentum, crowding) the proposal builds on.
- Whitepaper §5 — the mathematics of trust — the statistical core (Information Coefficient, permutation testing, false-discovery correction, overfitting probability) that any candidate signal — including a network-based early-warning indicator — must clear.

20.3. Validation approach

- Validation & anti-overfitting — how network claims specifically are protected from self-deception, beyond the platform's existing factor-validation gates.
- Reproducibility & experiment infrastructure — what makes an experiment auditable and repeatable here.
- Gap analysis & research roadmap — honest `BUILT` / `PARTIAL` / `PROPOSED` status per capability, so methodology claims don't outrun what's actually implemented.

See also: Research Questions for what each of these methods is being used to answer, and Experiments for the running record of applying them.

21. Experiments

The running record of research results and how to reproduce them. This is distinct from research-platform/findings/, which holds polished, publication-style write-ups of results that have already stabilized — experiments here are the raw material those get built from, including ones that didn't pan out (see journal/dead-ends.md).

Status: one experiment logged under this convention so far (below). Older results still live in research-platform/findings/ until they're migrated or superseded.

21.1. Logged experiments

- 2026-08-03-factor-catalogue-empirical-results — provenance record (run id, config lookups, method) for the empirical fields in phd/factor-catalog.md, the per-factor, per-family catalog. Naive and HAC (Newey–West) t-stat fields are specified but pending computation.

21.2. Convention for a new experiment entry

Each experiment gets a dated folder, `experiments/YYYY-MM-DD-short-name/`, containing:

- `README.md` — what question (research-questions.md) this tests, what was run, and the result.

- the config / parameters used (or a link to the commit/run-id in SBFoundation if the run itself lives there — see Reproducibility & experiment infrastructure for what a run identity captures).
- a link to the journal entry that motivated it, and the entry that recorded the outcome.

21.3. Existing results (published-quality)

- asset-dependency-networks-econophysics-review.md
- edgar-pit-integrity.md

21.4. Reproducibility

Every experiment inherits the platform's point-in-time and run-identity guarantees — see Reproducibility & experiment infrastructure for what "auditable and repeatable" means concretely here.

22. Experiment: Empirical Results for the Canonical Factor Catalogue

Date: 2026-08-03 **Status:** partially run — theory/config fields populated; two statistical fields (naive and HAC t-stat) specified but not yet computed **First entry under the `experiments/` convention** (see experiments/README.md)

> **2026-08-03 update:** the per-factor results this experiment produced now live in `phd/factor-catalog.md` as a single, self-contained, per-family catalog (justification + style + platform wiring + empirical results in one place per factor), at the operator's request — rather than split between the §10 table in `phd/factor-models-and-benchmarks.md` and this file. This entry now holds the **provenance record**: what was run, what was looked up, and the exact method for the two fields still pending. Read the catalog for current numbers; read this for how they were (or will be) produced.

22.1. Question this tests

`phd/factor-catalog.md` (formerly §10 of `phd/factor-models-and-benchmarks.md`) lists the platform's canonical factors with their economic justification and style, but that alone is no evidence the platform's own implementation

of each one actually carries a signal. This experiment supplies that evidence.

It also bears directly on research-platform/04-factor-layer-as-multiplex-layer-1.md: these factors *are* the proposal's Layer 1 (the factor-exposure bipartite layer of the multiplex network). A network built on factors with no real predictive content is a network built on noise — so this is a prerequisite check for the PhD proposal's L1, not just a platform-hygiene exercise.

22.2. What was run

The empirical fields in the catalog are drawn from the existing five-sidecar diagnostic run `260526_6ddcd9` (2026-05-26), synthesized in `platform/factors.md` — factor diagnostics (ADF/entropy), per-factor IC, factor contribution, Alphalens, and MCPT. No new run was executed for this entry; this re-projects that run's results onto the catalog, plus fresh config lookups against `config/factors/*.yaml` in SBFoundation for `style`, `hypothesis_class`, `expected_sign`, `source_table`/`source_column`, and current `status`.

That config lookup surfaced a correction, folded into the catalog: `bb_lower_20`, `ma_50d`, `ma_200d`, and `vwap_20d` were the four factors the `260526_6ddcd9` synthesis called out as having "real" pre-fix MCPT significance — but all four were deprecated for cause on 2026-07-20 (F-289/LH-25, run `260718_31a51e`) for a CF-1 look-ahead defect in their raw price-level source columns. The "cleanest MCPT result in the library" turned out to be a look-ahead artifact two months later — see the `bb_lower_20` entry in the catalog for the detail. That is exactly the kind of thing re-grounding against live config catches and a stale narrative wouldn't.

Caveat carried forward: `260526_6ddcd9` predates three fixes that materially affect these numbers — F-136 (MCPT nightly permutations 100 → 1000, corrected null construction), F-137 (Alphalens forward-return

winsorization), and F-138 (composite contribution rescaling). The catalog's MCPT fields are provisional until regenerated against a post-fix `NIGHTLY_FULL/RESEARCH_DAY` run.

22.3. Method for the two pending fields

Neither t-stat exists in any current sidecar — `compute_ic`, `compute_factor_mcpt`, and `compute_factor_diagnostics` don't compute a mean-return significance test, and `sbattribution.ols_hac` is currently wired only to strategy NAV streams (`beta_attribution_service.py`), not to the per-factor return panel. The exact method (so it's reproducible once run) lives in the catalog's Method section:

- **Naive t-stat**: `mean(F) / (std(F) / sqrt(T))` on each factor's `ops.risk_factor_return` series — the classic Fama–MacBeth / Fama–French second-pass report.
- **HAC (Newey–West) t-stat**: same series, intercept-only OLS via the platform's existing `sbattribution.ols.ols_hac` (`src/sbattribution/ols.py:79-125`) — the same kernel already used for CAPM/attribution alpha.
- **Why both**: comparing naive vs. HAC on the same series is itself diagnostic — how much of the naive "significance" is just serial correlation from overlapping-window/slow-moving characteristics.
- **Neither is a promotion gate** — per research-platform/08-validation-and-anti-overfitting.md, the platform's actual gate is IC-IR + MCPT against an honest trial count.

22.4. Next steps

1. Write the query/script computing `t_naive` and `t_HAC` per `factor_id` and fill in the `PENDING` fields in `phd/factor-catalog.md`.

2. Resolve the two duplicate pairs (`beta`/`market_beta_252d`, `momentum_12_1`/`momentum_12m_1m`) before republishing.
3. Regenerate against the first post-F-136/137/138 `NIGHTLY_FULL`/`RESEARCH_DAY` run.
4. Once stable, promote the catalog's empirical layer from provisional to publication-quality (`research-platform/findings/`-grade confidence), noting that in the catalog's provenance section rather than moving the content again.

22.5. Journal

Motivated by and recorded in journal/2026/2026-08-03.md.

Part V.

Publications & Journal

23. Publications

Nothing published yet. Placeholder structure so entries have an obvious home as they exist, per the PhD proposal's timeline (Paper 1 — Year 1, "Spectral Indicators of Instability in Heterogeneous Financial Networks"; Paper 2 — Year 2; Paper 3 — Year 3).

23.1. Papers

(none yet)

23.2. Conference submissions

(none yet)

23.3. Technical reports

(none yet — research-platform/findings/ holds the closest current equivalent: internal-facing, evidence-grade write-ups that aren't yet aimed at a venue.)

24. Research Journal

The raw, dated account of the research process — what was tried, in what order, and why — as opposed to experiments/ (structured results) or research-platform/findings/ (polished write-ups). Entries are allowed to be messy and wrong in the moment; that's the point of keeping a journal instead of only publishing conclusions.

24.1. How to write an entry

Add a file under the current year, `journal/YYYY/YYYY-MM-DD.md`. No fixed template, but a useful entry usually says:

- what you worked on and why (link the research question it bears on, if any),
- what happened — including partial/negative results,
- what you'd do differently next time, or what's now an open question.

If something clearly failed or was abandoned, add a line to dead-ends.md with a pointer back to the entry — dead ends are easy to forget and expensive to re-discover.

If something generalizes beyond the specific entry — a modeling mistake worth not repeating, a data quirk that will bite again — add it to lessons-learned.md.

24.2. Entries

24.2.1. 2026

- 2026-08-03 — repository restructured around a research-notebook shape (this journal included).

24.3. Running logs

- Dead ends
- Lessons learned

25. Lessons Learned

Generalizable takeaways from the journal — things worth not re-learning the hard way a second time.

(none logged yet)

26. Dead Ends

Things that were tried and abandoned, with a pointer back to the journal entry — kept so they don't get quietly re-attempted later.

(none logged yet)

27. 2026-08-03

Restructured `strawberry-labs` from a single business-whitepaper README into a research-notebook shape: Vision → Literature Review → Research Questions → Methodology → Experiments → Publications → Research Journal → Code → Data Catalog → Academic References.

What moved and why:

- The original whitepaper content moved to whitepaper/README.md unchanged — it's still the right document for a potential *user* of the platform, but it isn't a research vision statement, so it no longer owns the top-level README.
- `phd/`, `research-platform/`, `platform/`, and `library/` were left in place — they're already solid and heavily cross-linked, so the new top-level folders (`methodology/`, `experiments/`, `data-catalog/`, `code/`, `references/`, `journal/`) are thin indexes into them rather than copies.
- Genuinely new: research-questions.md (extracted from `phd/proposal.md` so the current hypothesis doesn't require reading the full proposal), this journal, and references/bibliography.md (DOI-normalized, distinct from the library's prose summaries).

Open follow-ups, tracked so they don't get lost:

- `library/research-map.md` is a stub — the actual paper-relationship graph hasn't been generated yet (candidate: the `graphify` skill).

- `data-catalog/licensing.md` and `data-catalog/refresh-schedules.md` are scaffolds — licensing terms per data vendor (FMP, FRED, FINRA) and refresh cadence aren't documented centrally yet, even though the platform enforces them operationally.
- `experiments/` has no entries yet under the new per-experiment convention; existing results still live in `research-platform/findings/`.

27.1. Later the same day: first experiments/ entry — factor catalogue results

Added experiments/2026-08-03-factor-catalogue-empirical-results, the first entry under this morning's new convention. Motivation: the canonical 13-factor catalogue in phd/factor-models-and-benchmarks.md §10 states economic justification and style but no empirical evidence — and these factors are literally the proposal's multiplex Layer 1, so "does the platform's own implementation of each one actually show a signal" isn't optional context, it's a prerequisite.

What happened:

- Re-projected the existing five-sidecar run `260526_6ddcd9` (already synthesized in `platform/factors.md`) onto the 13 catalogue rows, and pulled `style/hypothesis_class/expected_sign` from `config/factors/*.yaml` in SBFoundation directly rather than guessing.
- Added two new columns per the operator's request: a naive mean-return t-stat and a HAC (Newey–West) t-stat, computed the same way §3.3 of the factor-models note already computes CAPM/attribution alpha (`sbattribution.ols.ols_hac`) — reusing the existing estimator on the factor-return panel (`ops.risk_factor_return`) instead of the strategy NAV streams it's currently wired to.

- Both t-stat columns are **specified but not computed** — no sidecar currently produces them, and I didn't want to hand-wave numbers into a repo that otherwise polices this hard. Marked `PENDING` with the exact method so a future pass (or a script) can fill them in mechanically.
- Surfaced two things worth acting on before this table is trusted:

  (1) `beta/market_beta_252d` and `momentum_12_1/momentum_12m_1m` are known-identical duplicates per the run synthesis — counting both sides would inflate the honest trial count (F-321) the platform otherwise tracks; (2) `bb_lower_20` and the atomic `residual_momentum_252d_eod` have real evidence (the cleanest MCPT pass, and the highest IC-IR in the run respectively) but aren't in the §10 catalogue at all, while the *deprecated* composite `residual_momentum_252d` occupies conceptual space that could be confused with its atomic sibling.

Open follow-up: write the actual `t_naive/t_HAC` computation and rerun this once F-136/F-137/F-138 have all landed in a `NIGHTLY_FULL` run, so the MCPT column stops being pre-fix evidence.

27.2. Later still: merged the catalog into one self-contained, per-family document

Operator asked to restructure again: list each factor in its own section, grouped by family, list-style rather than wide tables, each factor self-contained (justification, source table/column, empirical results together) — and to stop splitting the catalog across `phd/factor-models-and-benchmarks.md` §10 and the `experiments/` entry.

- Created phd/factor-catalog.md: 13 family sections, each factor as its own subsection with style, source `table.column`, hypothesis

class/expected sign, lifecycle status, IC-IR, MCPT, and the two pending t-stats — all pulled from the same config lookups and run synthesis as the earlier pass, just reshaped and consolidated.

- `phd/factor-models-and-benchmarks.md` §10 is now a short pointer to the new file instead of the table.
- Slimmed `experiments/2026-08-03-factor-catalogue-empirical-results/README` down to a provenance record (what was run, the config-lookup correction, the pending-field method) rather than a duplicate of the data — the numbers live in one place now.
- Added two factors the original 13-row table didn't cover but the run data called for: `residual_momentum_252d_eod` (highest IC-IR in the run) under Momentum, and `bb_lower_20` under Short-horizon reversal — kept specifically as a cautionary entry, since re-checking current config revealed it (and `ma_50d`/`ma_200d`/`vwap_20d`) were deprecated for cause in July for a look-ahead defect, *after* being the run's cleanest MCPT result. Good reminder that a clean permutation p-value is necessary, not sufficient.

Part VI.

Code & Data

28. Code

This repository is documentation-only — the implementation lives in the private SBFoundation repository. This page is a curated index into the modules most relevant to the research questions here, not a mirror of the code itself.

Module	Relevance to the research
`src/sbfactors`	The factor library — Layer 1 (factor-exposure) of the proposal's multiplex network; see factor-layer-as-multiplex-layer-1.
`src/sbcrowding`	Comomentum / crowding measurement — the existing single-layer precedent for the proposal's systemic-risk signal; see systemic-risk-and-crowding-groundwork.
`src/sbriskmodel` / `src/sbfactorrisk`	The factor risk model — correlation structure the network layers build on.
`src/sbic`	Information Coefficient computation — the core "does it predict" test, whitepaper §5.1.
`src/sbpermtest`	Permutation / null testing — whitepaper §5.3.

Module	Relevance to the research
`src/sbpbo`	Probability of Backtest Overfitting — whitepaper §5.6; the discipline any new network indicator has to survive too.
`src/sbsurvivorship`	Survivorship-bias correction — whitepaper §5.4.
`src/sbcost`	Transaction-cost modeling — whitepaper §5.5.
`src/sbsweep`	Parameter/universe sweep engine — the search infrastructure experiments/ will drive.
`src/sbuniverse`	Point-in-time investable universe construction — see Universe API.

For narrative (not link-index) descriptions of how these fit together, see Platform depth and the end-to-end pipeline.

29. Data Catalog

29.1. Data provenance

The full domain/dataset reference already lives in platform/domain-datasets-reference.md — 34 datasets across 4 domains (`eod`, `quarter`, `annual`, `eow`), sourced from FMP, FRED, and FINRA, with Bronze/Silver/Gold table mappings and per-dataset quality flags. That document is generated from the platform's own config (`config/dataset_keymap.yaml`) and should stay the single source of truth for provenance rather than being duplicated here.

For the research specifically (beyond what the trading platform ingests), the PhD proposal's data section also names sources not yet in the platform: SEC 13F institutional holdings and SEC 10-K supply-chain parsing, needed for the ownership and supply-chain network layers. See gap-analysis-and-research-roadmap for what's `BUILT` vs. `PROPOSED` there.

29.2. Licensing

- licensing.md — per-vendor terms (FMP, FRED, FINRA, and any research-only sources added for the network layers).

29.3. Refresh schedules

- refresh-schedules.md — cadence per domain, drawn from the dataset reference's per-dataset `Refresh` column.

30. Data Licensing

Not yet documented centrally. The platform ingests from three vendors — each has its own terms that govern what can be redistributed, republished, or used in a public-facing paper/dataset release:

- **FMP** (Financial Modeling Prep) — requires `FMP_API_KEY`; commercial terms apply. Redistribution limits need confirming before any derived dataset from FMP-sourced fields is published alongside a paper.
- **FRED** (Federal Reserve Economic Data) — requires `FRED_API_KEY`; FRED data is generally public domain / open (US government data), but confirm per-series terms before redistribution.
- **FINRA** — no API key required; confirm redistribution terms for the short-interest dataset specifically.

Research-only sources named in the PhD proposal (SEC 13F filings, SEC 10-K text, French data library) are separately licensed — SEC filings are public record, but derived/processed extracts (e.g. parsed supply-chain links) may carry their own terms depending on the parsing service used.

Action needed: confirm and record actual license terms per source before any dataset built from them is published as part of a paper's supplementary material.

31. Refresh Schedules

Refresh cadence is already tracked per-dataset in platform/domain-datasets-reference.md (the `Refresh` column, e.g. "Daily (1)", "Quarterly (90)"). This page exists to summarize that at the domain level for research planning — e.g. "how stale can my network snapshot be" — without re-deriving it from the full per-dataset table each time.

Domain	Cadence	Notes
`eod`	Daily	Primary daily-refresh domain; feeds the nightly factor layer.
`quarter`	Daily ingestion, quarterly content	New rows land daily but reflect quarterly filings — season-gated (`QuarterService` skips outside earnings windows unless explicitly requested).
`annual`	Annual content	See domain-datasets-reference.md for exact per-dataset cadence.

Domain	Cadence	Notes
`eow`	Weekly	Index constituents, macro indicators, splits, analyst estimates, earnings surprises, short interest.

Not yet covered here: cadence for the research-only sources (SEC 13F — quarterly by regulation; SEC 10-K — annual) once they're actually ingested for the ownership/supply-chain network layers. Update this table when that ingestion exists.

Part VII.

Appendices

Reference Library

Plain-language, one-page summaries of every academic paper that grounds this research programme and the SBFoundation platform — factor research, transaction-cost, survivorship, risk-model, and lifecycle-gate design, plus the networks / graph-neural-network literature behind the systemic-risk thesis. Each entry links the paper's canonical source; the PDFs themselves are not redistributed.

Networks & Graph Neural Networks (PhD proposal)

- **Pathways towards instability in financial networks** — source — Bardoscia, Battiston, Caccioli & Caldarelli (2017)
- **The structure and dynamics of multilayer networks** — source — Boccaletti et al. (2014)
- **Semi-Supervised Classification with Graph Convolutional Networks** — source — Kipf & Welling (2016/2017)
- **Multilayer Networks** — source — Kivelä et al. (2014)
- **Relation between Financial Market Structure and the Real Economy: Comparison between Clustering Methods** — source — Musmeci, Aste & Di Matteo (2015)
- **Spatio-Temporal Graph Convolutional Networks: A Deep Learning Framework for Traffic Forecasting** — source — Yu, Yin & Zhu (2017/2018)

Platform Reference Library

Multiple testing, overfitting & significance

- **... and the Cross-Section of Expected Returns** — source — Harvey, Liu & Zhu (2016)
- **The Deflated Sharpe Ratio: Correcting for Selection Bias, Backtest Overfitting and Non-Normality** — source — Bailey & López de Prado (2014)
- **The Probability of Backtest Overfitting** — source — Bailey, Borwein, López de Prado & Zhu (2014)
- **The Control of the False Discovery Rate in Multiple Testing under Dependency** — source — Benjamini & Yekutieli (2001)
- **False-Positive Psychology: Undisclosed Flexibility in Data Collection and Analysis Allows Presenting Anything as Significant** — source — Simmons, Nelson & Simonsohn (2011)
- **The Garden of Forking Paths: Why Multiple Comparisons Can Be a Problem, Even When There Is No "Fishing Expedition" or "p-hacking" and the Research Hypothesis Was Posited Ahead of Time** — source — Gelman & Loken (2013)
- **Leakage and the Reproducibility Crisis in ML-based Science** — source — Kapoor & Narayanan (2023)
- **The Sharpe Ratio Efficient Frontier** — source — Bailey & López de Prado (2012)

Anomaly replication, the factor zoo, universe & decay

- **Replicating Anomalies** — source — Hou, Xue & Zhang (2020)
- **A Taxonomy of Anomalies and Their Trading Costs** — source — Novy-Marx & Velikov (2016)
- **Understanding Defensive Equity** — source — Novy-Marx (2014)

- **Zeroing In on the Expected Returns of Anomalies** — source — Chen & Velikov (2023)
- **Navigating the Factor Zoo around the World: An Institutional Investor Perspective** — source — Bartram et al. (2021)
- **Taming the Factor Zoo: A Test of New Factors** — source — Feng, Giglio & Xiu (2020)
- **Does Academic Research Destroy Stock Return Predictability?** — source — McLean & Pontiff (2016)
- **Monotonicity in Asset Returns: New Tests with Applications to the Term Structure, the CAPM, and Portfolio Sorts** — source — Patton & Timmermann (2010)
- **Anomalies across the Globe: Once Public, No Longer Existent?** — source — Jacobs & Müller (2020)

Transaction-cost estimation & implementation

- **A Simple Way to Estimate Bid-Ask Spreads from Daily High and Low Prices** — source — Corwin & Schultz (2012)
- **A Simple Estimation of Bid-Ask Spreads from Daily Close, High, and Low Prices** — source — Abdi & Ranaldo (2017)
- **Trading Costs of Asset Pricing Anomalies** — source — Frazzini, Israel & Moskowitz (2015)
- **Optimal Execution of Portfolio Transactions** — source — Almgren & Chriss (2000)
- **Direct Estimation of Equity Market Impact** — source — Almgren et al. (2005)
- **Dynamic Trading with Predictable Returns and Transaction Costs** — source — Gârleanu & Pedersen (2013)
- **Anomalies and Their Short Sale Costs** — source — Muravyev, Pearson & Pollet (2025)

Crowding, unwinds & factor crashes

- **What Happened to the Quants in August 2007? Evidence from Factors and Transactions Data** — source — Khandani & Lo (2007)
- **Momentum Crashes** — source — Daniel & Moskowitz (2016)
- **How Can "Smart Beta" Go Horribly Wrong?** — source — Arnott, Beck, Kalesnik & West (2016)
- **Comomentum: Inferring Arbitrage Activity from Return Correlations** — source — Lou & Polk (2012)
- **Contrarian Factor Timing is Deceptively Difficult** — source — Asness et al. (2017)
- **The Siren Song of Factor Timing (aka "Smart Beta Timing" aka "Style Timing")** — source — Asness (2016)

Portfolio construction, risk model & attribution

- **Improved Estimation of the Covariance Matrix of Stock Returns with an Application to Portfolio Selection** — [source](https://doi.org/10.1016/S0927-5398(03) — Ledoit & Wolf (2003)
- **Honey, I Shrunk the Sample Covariance Matrix** — source — Ledoit & Wolf (2004)
- **The Intuition Behind Black-Litterman Model Portfolios** — source — He & Litterman (1999)

Survivorship & delisting bias

- **The Delisting Bias in CRSP Data** — source — Shumway (1997)
- **The Delisting Bias in CRSP's Nasdaq Data and Its Implications for the Size Effect** — source — Shumway & Warther (1999)

- **Survivorship Bias in Performance Studies** — source — Brown et al. (1992)
- **Mutual Fund Survivorship** — source — Carhart, Carpenter, Lynch & Musto (2002)

Scheduling & control theory (lifecycle-state design)

- **Scheduling Algorithms for Multiprogramming in a Hard-Real-Time Environment** — source — Liu & Layland (1973)
- **A Proof for the Queuing Formula: L = W** — source — Little (1961)
- **Control Chart Tests Based on Geometric Moving Averages** — source — Roberts (1959)

Seasonality & calendar anomalies

- **Seasonality in the Cross-Section of Stock Returns** — source — Heston & Sadka (2008)
- **Mood Betas and Seasonalities in Stock Returns** — source — Hirshleifer, Jiang & Meng (2020)

Momentum, value & factor-model foundations (wiki strategy/foundation pages)

- **Returns to Buying Winners and Selling Losers: Implications for Stock Market Efficiency** — source — Jegadeesh & Titman (1993)
- **Residual Momentum** — source — Blitz, Huij & Martens (2011)
- **Volatility-Managed Portfolios** — source — Moreira & Muir (2017)
- **Momentum Has Its Moments** — source — Barroso & Santa-Clara (2015)

- **Factor Momentum and the Momentum Factor** — source — Ehsani & Linnainmaa (2022)
- **The Other Side of Value: The Gross Profitability Premium** — source — Novy-Marx (2013)
- **Betting Against Beta** — source — Frazzini & Pedersen (2014)
- **Fads, Martingales, and Market Efficiency** — source — Lehmann (1990)
- **Simple Technical Trading Rules and the Stochastic Properties of Stock Returns** — source — Brock, Lakonishok & LeBaron (1992)
- **Common Risk Factors in the Returns on Stocks and Bonds** — [source](https://doi.org/10.1016/0304-405X(93) — Fama & French (1993)
- **Time Series Momentum** — source — Moskowitz, Ooi & Pedersen (2012)

Portfolio construction & diversification (wiki strategy pages)

- **Optimal Versus Naive Diversification: How Inefficient is the 1/N Portfolio Strategy?** — source — DeMiguel, Garlappi & Uppal (2009)
- **Building Diversified Portfolios that Outperform Out of Sample (Hierarchical Risk Parity)** — source — López de Prado (2016)

Strategy-specific, thematic & AI (wiki strategy/architecture pages)

- **Beyond the Status Quo: A Critical Assessment of Lifecycle Investment Advice** — source — Anarkulova et al. (2025)

- **The Alchemy of Multibagger Stocks: An Empirical Investigation of Factors that Drive Outperformance in the Stock Market** — source — Yartseva (2025)
- **ChatGPT-based Investment Portfolio Selection** — source — Romanko et al. (2023)
- **The Mosaic Permutation Test: An Exact and Nonparametric Goodness-of-Fit Test for Factor Models** — source — Spector et al. (2024)
- **Empirical Asset Pricing via Machine Learning** — source — Gu, Kelly & Xiu (2020)

Diagnostics, attribution & multi-asset mechanics (wiki gap-fill pages, 2026-07-10)

- **Carry** — source — Koijen, Moskowitz, Pedersen & Vrugt (2018)
- **A Century of Evidence on Trend-Following Investing** — source — Hurst, Ooi & Pedersen (2017)

Event-driven capture — PEAD, index reconstitution, analyst revisions (wiki doc-063, 2026-07-10)

- **Do Demand Curves for Stocks Slope Down?** — source — Shleifer (1986)
- **Momentum Strategies** — source — Chan, Jegadeesh & Lakonishok (1996)

32. Academic References — Bibliography

A DOI-normalized index of every source behind the reference library. This is a different artifact from the library: the library holds *prose summaries* ("why this matters to the research"); this page holds *citation data* (identifier, venue) for anyone who needs to cite the work formally.

Zotero status: not yet set up. The entries below were compiled by hand from the library's source links. To move to Zotero-managed citations: import each DOI/arXiv ID below into a shared Zotero library, then regenerate this page as a Zotero-exported bibliography (e.g. via Better BibTeX) instead of maintaining it by hand. Until that happens, treat this as the interim source of truth.

Coverage note: many working papers (NBER, SSRN) never receive a DOI even after wide circulation — those rows are marked *no DOI* with their canonical working-paper link instead of a fabricated identifier.

32.1. Multiple testing, overfitting & significance

Citation	Identifier
Harvey, C., Liu, Y. and Zhu, H. (2016). …and the Cross-Section of Expected Returns. *Review of Financial Studies*.	no DOI — NBER w20592
Bailey, D. and López de Prado, M. (2014). The Deflated Sharpe Ratio. *Journal of Portfolio Management*.	no DOI — SSRN 2460551
Bailey, D., Borwein, J., López de Prado, M. and Zhu, Q. (2014). The Probability of Backtest Overfitting.	no DOI — SSRN 2326253
Benjamini, Y. and Yekutieli, D. (2001). The Control of the False Discovery Rate under Dependency. *Annals of Statistics*.	10.1214/aos/1013699998
Simmons, J., Nelson, L. and Simonsohn, U. (2011). False-Positive Psychology. *Psychological Science*.	10.1177/0956797611417632
Gelman, A. and Loken, E. (2013). The Garden of Forking Paths.	no DOI — author copy
Kapoor, S. and Narayanan, A. (2023). Leakage and the Reproducibility Crisis in ML-based Science.	arXiv:2207.07048
Bailey, D. and López de Prado, M. (2012). The Sharpe Ratio Efficient Frontier.	no DOI — SSRN 1821643

32.2. Anomaly replication, the factor zoo, universe & decay

Citation	Identifier
Hou, K., Xue, C. and Zhang, L. (2020). Replicating Anomalies.	no DOI — NBER w23394
Novy-Marx, R. and Velikov, M. (2016). A Taxonomy of Anomalies and Their Trading Costs.	no DOI — NBER w20721
Novy-Marx, R. (2014). Understanding Defensive Equity.	no DOI — NBER w20591
Chen, A. and Velikov, M. (2023). Zeroing In on the Expected Returns of Anomalies. *JFQA*.	10.17016/FEDS.2020.039
Navigating the Factor Zoo around the World (2021). *Journal of Business Economics*.	10.1007/s11573-021-01035-y
Feng, G., Giglio, S. and Xiu, D. (2020). Taming the Factor Zoo. *Journal of Finance*.	no DOI — NBER w25481
McLean, R. and Pontiff, J. (2016). Does Academic Research Destroy Stock Return Predictability? *Journal of Finance*.	10.1111/jofi.12365
Patton, A. and Timmermann, A. (2010). Monotonicity in Asset Returns. *JFE*.	10.1016/j.jfineco.2010.06.006
Jacobs, H. and Müller, S. (2020). Anomalies across the Globe. *JFE*.	10.1016/j.jfineco.2019.06.004

32.3. Transaction-cost estimation & implementation

Citation	Identifier
Corwin, S. and Schultz, P. (2012). A Simple Way to Estimate Bid-Ask Spreads from Daily High and Low Prices. *Journal of Finance*.	10.1111/j.1540-6261.2012.01729.x
Abdi, F. and Ranaldo, A. (2017). A Simple Estimation of Bid-Ask Spreads from Daily Close, High, and Low Prices. *RFS*.	10.1093/rfs/hhx084
Frazzini, A., Israel, R. and Moskowitz, T. (2015). Trading Costs of Asset Pricing Anomalies.	no DOI — SSRN 2294498
Almgren, R. and Chriss, N. (2000). Optimal Execution of Portfolio Transactions. *Journal of Risk*.	10.21314/JOR.2000.041
Almgren, R. et al. (2005). Direct Estimation of Equity Market Impact. *Risk*.	no DOI — risk.net
Garleanu, N. and Pedersen, L. (2013). Dynamic Trading with Predictable Returns and Transaction Costs. *Journal of Finance*.	10.1111/jofi.12080
Muravyev, D., Pearson, N. and Pollet, J. (2025). Anomalies and Their Short Sale Costs. *Journal of Finance*.	10.1111/jofi.13501

32.4. Crowding, unwinds & factor crashes

Citation	Identifier
Khandani, A. and Lo, A. (2007). What Happened to the Quants in August 2007?	no DOI — SSRN 1015987
Daniel, K. and Moskowitz, T. (2016). Momentum Crashes. *JFE*.	no DOI — NBER w20439
Arnott, R., Beck, N., Kalesnik, V. and West, J. (2016). How Can "Smart Beta" Go Horribly Wrong?	no DOI — Research Affiliates
Lou, D. and Polk, C. Comomentum: Inferring Arbitrage Activity from Return Correlations.	no DOI — SSRN 2029199
Asness, C. et al. (2017). Contrarian Factor Timing is Deceptively Difficult.	no DOI — SSRN 2928945
Asness, C. (2016). The Siren Song of Factor Timing.	no DOI — SSRN 2763956

32.5. Portfolio construction, risk model & attribution

Citation	Identifier
Ledoit, O. and Wolf, M. (2003). Improved Estimation of the Covariance Matrix of Stock Returns.	10.1016/S0927-5398(03)
Ledoit, O. and Wolf, M. (2004). Honey, I Shrunk the Sample Covariance Matrix. *JPM*.	10.3905/jpm.2004.110

Citation	Identifier
He, G. and Litterman, R. (1999). The Intuition Behind Black-Litterman Model Portfolios.	no DOI — SSRN 334304

32.6. Survivorship & delisting bias

Citation	Identifier
Shumway, T. (1997). The Delisting Bias in CRSP Data. *Journal of Finance*.	no DOI — JSTOR 2329566
Shumway, T. and Warther, V. (1999). The Delisting Bias in CRSP's Nasdaq Data. *Journal of Finance*.	no DOI — JSTOR 797998
Brown, S., Goetzmann, W., Ibbotson, R. and Ross, S. (1992). Survivorship Bias in Performance Studies. *RFS*.	10.1093/rfs/5.4.553
Carhart, M., Carpenter, J., Lynch, A. and Musto, D. (2002). Mutual Fund Survivorship. *RFS*.	10.1093/rfs/15.5.1439

32.7. Scheduling & control theory

Citation	Identifier
Liu, C. and Layland, J. (1973). Scheduling Algorithms for Multiprogramming in a Hard-Real-Time Environment.	10.1145/321738.321743
Little, J. (1961). A Proof for the Queuing Formula: L = W. *Operations Research*.	10.1287/opre.9.3.383
Roberts, S. (1959). Control Chart Tests Based on Geometric Moving Averages. *Technometrics*.	10.1080/00401706.1959.10489860

32.8. Seasonality & calendar anomalies

Citation	Identifier
Heston, S. and Sadka, R. (2008). Seasonality in the Cross-Section of Stock Returns. *JFE*.	10.1016/j.jfineco.2007.02.003
Hirshleifer, D., Jiang, D. and Meng, Y. (2020). Mood Betas and Seasonalities in Stock Returns.	no DOI — NBER w24676

32.9. Momentum, value & factor-model foundations

Citation	Identifier
Jegadeesh, N. and Titman, S. (1993). Returns to Buying Winners and Selling Losers. *Journal of Finance*.	no DOI — JSTOR 2328882
Blitz, D., Huij, J. and Martens, M. (2011). Residual Momentum. *Journal of Empirical Finance*.	10.1016/j.jempfin.2011.01.003
Moreira, A. and Muir, T. (2017). Volatility-Managed Portfolios.	no DOI — NBER w22208
Barroso, P. and Santa-Clara, P. (2015). Momentum Has Its Moments. *JFE*.	10.1016/j.jfineco.2014.11.010
Ehsani, S. and Linnainmaa, J. (2022). Factor Momentum and the Momentum Factor.	no DOI — NBER w25551
Novy-Marx, R. (2013). The Other Side of Value: The Gross Profitability Premium. *JFE*.	10.1016/j.jfineco.2013.01.003
Frazzini, A. and Pedersen, L. (2014). Betting Against Beta. *JFE*.	10.1016/j.jfineco.2013.10.005
Lehmann, B. (1990). Fads, Martingales, and Market Efficiency. *QJE*.	10.2307/2937816
Brock, W., Lakonishok, J. and LeBaron, B. (1992). Simple Technical Trading Rules and the Stochastic Properties of Stock Returns. *Journal of Finance*.	10.1111/j.1540-6261.1992.tb04681.x
Fama, E. and French, K. (1993). Common Risk Factors in the Returns on Stocks and Bonds. *JFE*.	10.1016/0304-405X(93)

Citation	Identifier
Moskowitz, T., Ooi, Y. and Pedersen, L. (2012). Time Series Momentum. *JFE*.	no DOI — SSRN 2089463

32.10. Portfolio construction & diversification

Citation	Identifier
DeMiguel, V., Garlappi, L. and Uppal, R. (2009). Optimal Versus Naive Diversification. *RFS*.	10.1093/rfs/hhm075
López de Prado, M. (2016). Building Diversified Portfolios that Outperform Out of Sample.	no DOI — SSRN 2708678

32.11. Strategy-specific, thematic & AI

Citation	Identifier
Anarkulova, A. et al. (2025). Beyond the Status Quo: A Critical Assessment of Lifecycle Investment Advice.	no DOI — SSRN 4590406
Yartseva (2025). The Alchemy of Multibagger Stocks.	no DOI — BCU open access
Romanko, O. et al. (2023). ChatGPT-based Investment Portfolio Selection.	arXiv:2308.06260

Citation	Identifier
Spector, A. and Candès, E. (2024). The Mosaic Permutation Test.	arXiv:2404.15017
Gu, S., Kelly, B. and Xiu, D. (2020). Empirical Asset Pricing via Machine Learning. *RFS*.	10.1093/rfs/hhaa009

32.12. Diagnostics, attribution & multi-asset mechanics

Citation	Identifier
Koijen, R. et al. (2018). Carry.	no DOI — NBER w19325
Hurst, B., Ooi, Y. and Pedersen, L. (2017). A Century of Evidence on Trend-Following Investing. *JPM*.	10.3905/jpm.2017.44.1.015

32.13. Event-driven capture

Citation	Identifier
Shleifer, A. (1986). Do Demand Curves for Stocks Slope Down? *Journal of Finance*.	10.1111/j.1540-6261.1986.tb04518.x
Chan, L., Jegadeesh, N. and Lakonishok, J. (1996). Momentum Strategies.	no DOI — NBER w5375

32.14. Networks, multiplex & GNNs (PhD-proposal literature)

Citation	Identifier
Bardoscia, M. et al. (2017). Pathways towards instability in financial networks.	arXiv:1602.05883
Boccaletti, S. et al. (2014). The structure and dynamics of multilayer networks.	arXiv:1407.0742
Kipf, T. and Welling, M. (2016). Semi-Supervised Classification with Graph Convolutional Networks.	arXiv:1609.02907
Kivelä, M. et al. (2014). Multilayer Networks.	arXiv:1309.7233
Musmeci, N., Aste, T. and Di Matteo, T. (2015). Relation between Financial Market Structure and the Real Economy.	arXiv:1406.0496
Yu, B., Yin, H. and Zhu, Z. (2017). Spatio-Temporal Graph Convolutional Networks.	arXiv:1709.04875

SBFoundation Whitepaper

This chapter is generated from the canonical platform doc, which lives in `docs/` *(single source of truth).*

Disciplined Risk-Premium Harvesting for the Sole Trader

Telling Edge from Luck

A whitepaper on the business problem Strawberry Labs solves, and the mathematics it uses to solve it.

About this paper. It is written for a *potential user* — someone who understands markets and risk but is not steeped in quantitative statistics. It is deliberately technology-light: you will meet the business problem and the reasoning behind every decision the platform makes, but no code, no vendor names, and no jargon left unexplained. Where the platform uses a number — a hurdle, a threshold, a measured discount — this paper shows you the actual number and tells you which piece of published research it came from.

The single most important thing to understand before reading further: **this platform's value is not speed, secrecy, or cleverness. Its value is discipline** — a refusal to let a trading signal reach real money until it has survived a gauntlet of statistical tests that the academic literature designed specifically to catch signals that only *look* good.

---

1. Executive summary

Strawberry Labs is a **sole-trader, nightly, multi-asset risk-premium harvester**. In plain terms: it is a system for one person to systematically capture the persistent returns that markets pay for bearing certain risks — run once each evening on end-of-day data, starting with stocks and expanding deliberately outward.

What makes it different is not what it trades but what it *refuses* to trade. Most people operating at this scale lose money not because they lack ideas, but because they cannot tell a real edge from a lucky accident — and they bet real capital on the accident. Strawberry Labs is built, end to end, as an **evidence funnel** whose entire job is to make that mistake structurally difficult.

The platform is honest about where it stands, and you should hold it to that:

- **Mature on equities.** The data pipeline, factor research, validation gates, risk model, portfolio construction, and operator dashboard all work and are exercised in production for stocks.
- **Phase-0 MVP available July 1, 2026** — one strategy walked the entire path on corrected evidence, trading in a **paper account only** (real broker fills, no real money).
- **Phase 1 completes the funnel and broadens the surface (paper).** By the end of Phase 1 the validation gauntlet is *complete* — a causal-rationale front gate, decile monotonicity, factor independence, and a crowding check join the statistical core (§5.7) — at least **three** strategies have walked it, and ETF rotation, event capture, and a long/short experiment all trade in paper.
- **First strategy goes live October 1, 2026** — the proven momentum strategy crosses from paper into real capital after it earns the right (see §7).

- **Expanding deliberately thereafter** into ETFs, events, futures, commodities, and crypto — with several strategy classes excluded *forever, by design.*

Those three dates — **July 1 paper, October 1 live, deliberate expansion after** — are the headline to remember.

This paper is for one kind of reader: a single operator who wants to harvest durable risk premia at a daily-or-slower cadence, and who values a system that will tell them *no*. It is explicitly **not** for high-frequency traders, statistical-arbitrage shops, or anyone whose edge depends on speed or information no individual can obtain.

---

2. The business problem

Why most people at this scale lose

A single operator competing in public markets faces four structural disadvantages that no amount of effort or code can erase:

1. **Latency** — institutions act in microseconds; you act the next morning.
2. **Capital scale** — meaningful diversification across some strategies requires far more capital than one person has.
3. **Market access** — prime brokerage, securities lending, expert networks, and the ability to act as an authorized participant are institutional plumbing you cannot replicate.
4. **Attention** — you are one person running everything; strategies that need constant monitoring will eventually be monitored badly.

The rule that follows is simple: any strategy whose edge depends on out-running these disadvantages is a losing fight. That single principle eliminates most of what people *think* of as "quant trading" — and eliminating it is the first act of discipline.

The one edge that survives

There are only two sources of trading edge in existence:

Edge	What it is	Fit for one operator
Risk premium	Persistent compensation for bearing a systematic risk or exploiting a durable behavioral pattern	**Strong** — survives at one-person attention and capital scale
Market inefficiency	Fleeting mispricing from events, microstructure, sentiment, or positioning	**Poor** — fast-decaying, and fights every disadvantage above

Strawberry Labs is, by deliberate design, a **risk-premium harvester**. It pursues returns that exist because they compensate someone for a real risk — and which therefore persist long enough for a person checking the market once a day to capture them. The inefficiency track is not deferred; it is *excluded* (see §3).

That risk premia exist, and persist, is not folklore. It is the central finding of decades of financial economics — from the Capital Asset Pricing Model of the 1960s, through Fama and French's three-factor model (1992), to the momentum effect documented by Jegadeesh and Titman (1993). The

engine that turns a small edge into real performance is captured in one relationship, the **Fundamental Law of Active Management** (Grinold; Grinold and Kahn):

> **Performance** $\approx$ **skill** $\times \sqrt{\textbf{breadth}}$ — a tiny predictive edge, applied across many independent bets, compounds into a meaningful result.

This is precisely why a *cross-sectional, many-names, nightly* approach is the right shape for a sole trader: you do not need to be very right about any one stock; you need to be slightly right across hundreds of them, repeatedly.

The silent killer: overfitting

Here is the danger that actually destroys retail capital. Test enough ideas against historical data and — by pure chance — one of them will look brilliant. Deploy it, and it reverts to noise, taking your money with it. The more ideas you try, the *more certain* it becomes that your best-looking result is a fluke.

A sole trader's capital cannot survive this. So the platform's central, non-negotiable job is to **distinguish a genuine edge from a lucky one** — and to do so with enough statistical force that the answer can be trusted. Sections 5 is devoted entirely to how it does this.

There is even an unflattering twist for *real* edges: McLean and Pontiff (2016) showed that once a predictive pattern is published, its returns fall roughly 26% out of sample and 58% after publication — worst for the very strategies that looked best in the backtest. So the platform treats *every* backtest as optimistic and discounts it before trusting it. Honesty is not a posture here; it is a risk control.

The counterintuitive advantage: capacity inversion

One disadvantage cuts the *other* way. The smallest, least-liquid stocks — microcaps — are a place institutions **cannot** go: a multi-billion-dollar fund cannot deploy meaningful size into a tiny company without moving the price against itself. That habitat is therefore the **structural advantage of being small**. A sole trader can harvest risk premia in names that are simply off-limits to large players. This does not reverse the rule above — the excluded strategy classes stay excluded — but it reframes microcap from "too risky to touch" into "your home turf."

---

3. The operating model and its deliberate boundaries

Every design decision in Strawberry Labs flows from one frame: **a single human operator, making one decision per day.** Concretely:

- A **nightly cycle** that produces a target portfolio from end-of-day data.
- **One price source per instrument** — no cross-venue plumbing.
- **Daily-or-slower** decisions — signals that decay over weeks-to-months, not minutes.
- **Manual or simple-bridge order placement** the next morning — no autonomous 24/7 execution stack.
- **No external clients, no regulatory reporting** — you are the only investor.

These are not gaps. They are **seven deliberate design choices**, and presenting them as features is the point:

#	Design choice	Why it fits one operator
1	Nightly end-of-day cadence	One human, one predictable decision window
2	Single price source per instrument	No multi-venue complexity; matches a single account
3	Slow, statistically rigorous validation	Sole-trader capital cannot survive overfitting
4	Long-horizon factor & strategy lifecycle	No incentive to chase fast-decay edges that need babysitting
5	Cross-sectional ranking as the test of merit	The right framework for the risk-premium strategies in scope
6	Manual / simple-bridge order placement	Avoids the cost and risk of a 24/7 order-management system
7	No external client or regulatory surface	Removes ~30% of the institutional surface that simply does not apply

The negative space — what is deliberately never built

A platform's discipline is most visible in what it refuses to attempt. The following are **out of scope permanently**, because each fights a sole-trader disadvantage that no code can close:

- **Intraday, tick, and microstructure trading** — conflicts with the nightly cadence; institutional latency wins every contest.
- **Cross-venue and dual-listing arbitrage** — requires a multi-venue execution stack.
- **Statistical arbitrage** (pairs, cointegrated baskets) — a decayed strategy class whose realistic net return after costs and borrow is approximately zero at retail scale.
- **Merger and event arbitrage** — needs a legal / expert-network informational edge an individual does not have.
- **High-frequency and latency-sensitive execution** — uneconomical for one person.
- **Multi-strategy formal capital allocators, compliance, and client reporting** — institutional overhead irrelevant to a single operator.

The message is not "we haven't gotten to these yet." It is "**chasing these would be a mistake, and refusing them is part of the value.**" A sharp tool that does one thing well beats a generic one that does many things badly.

---

4. The pipeline as an operational flow

What actually happens each night, between raw market data and a target portfolio? The platform is best understood as a **repeatable assembly line** with a single identity stamped on each night's run, so everything that happens is traceable and reproducible.

```
flowchart LR
    A["Acquire<br/>raw end-of-day data"] --> B["Validate<br/>clean & conform"
    B --> C["Model<br/>point-in-time"]
```

```
    C --> D["Factors<br/>rank the names"]
    D --> E["Strategy<br/>apply a mechanic"]
    E --> F["Portfolio<br/>weights + risk caps"]
    F --> G["Paper order<br/>submit & reconcile"]
```

Diagram A — the Nightly Assembly Line. One run identity end-to-end; one nightly decision window; no look-ahead.

The three data stages, in business terms:

- **Acquire.** Ingest exact vendor data and preserve it untouched. Every fetch — including every *failure* — is recorded, so the platform can always answer "what did we actually receive, and when?"
- **Validate & conform.** Clean and type the data, remove duplicates, and restrict it to an investable universe (excluding, for example, instruments too illiquid or structurally unsuitable to trade).
- **Model.** Assemble an analytics layer designed so that **on any historical date, you see only what you could have known on that date** — never the future. This *point-in-time integrity* is the bedrock of trust: a backtest that accidentally peeks at future data is worthless, and most of the ways amateurs fool themselves trace back to exactly this leak.

From there, the research-to-order spine runs: **Factor → Strategy → Portfolio → Order.** A *factor* is a measurable property used to rank stocks; a *strategy* applies a mechanic (e.g. "buy the top-ranked, hold a month") to a validated factor; *portfolio construction* turns rankings into position sizes; and the final stage turns those into orders. (The factor layer here — the point-in-time map of which names load on which factors — is also the first layer of the market-wide network studied by the platform's research program; see §7.)

Running example — the 12-1 momentum factor. Throughout this paper we follow one real factor: **12-1**

momentum — *rank every stock by its return over the past twelve months, skipping the most recent month* (the skip avoids short-term reversal noise). Intuitively: "buy what has been winning over the past year, ignoring last month's wobble."

On the assembly line, each night the platform computes this factor point-in-time for every name in the investable universe and stores the ranking. No claim has been made yet that it *works* — that is precisely the question Section 5 exists to answer. At this stage, 12-1 momentum is just a candidate.

---

5. The mathematics of trust

This is the heart of the platform and the heart of this paper. Each subsection below answers one business question, explains the concept in plain language, shows the platform's actual number, and names the published research that the hurdle comes from. The recurring message: **Strawberry Labs did not invent its standards — it adopted the literature's, and in one important case adopted a standard the literature explicitly recommended that most practitioners ignore.**[1]

Picture the whole section as a **funnel**. Many candidate factors enter at the top. Each gate rejects some. Only a few survive to become eligible for real (paper) capital.

[1] Figures in the running example (information ratio 0.49, nightly p 0.0099, deep-run p 0.0020, cost drag 650 bps, survivorship gap −0.035, 4,858 delisted names) are drawn from actual platform runs and are illustrative; exact values vary by run and data window. Citation page numbers and volumes should be confirmed against source PDFs before formal publication.

```
flowchart LR
    Z["Many candidate factors"] --> Q0{"Should it exist?<br/>economic rationale"}
    Q0 -->|pass| Q1{"Does it predict?<br/>IC &gt; 0.2, monotone"}
    Q1 -->|pass| Q2{"Not just luck?<br/>false-discovery panel"}
    Q2 -->|pass| Q3{"Not random?<br/>permutation, t &gt; 3"}
    Q3 -->|pass| Q4{"Survives dead companies?<br/>survivorship haircut"}
    Q4 -->|pass| Q5{"Survives trading costs?<br/>net-of-cost + capacity"}
    Q5 -->|pass| Q6{"Not overfit?<br/>overfitting / lockbox"}
    Q6 -->|pass| Q7{"Independent &amp; uncrowded?<br/>orthogonality + crowding"}
    Q7 -->|pass| V["VALIDATED<br/>eligible for paper capital"]
    Q0 & Q1 & Q2 & Q3 & Q4 & Q5 & Q6 & Q7 -->|fail| R["rejected"]
```

Diagram B — the Evidence Funnel. Wide at the top, narrow at the bottom; most ideas are rejected, by design. The two outer gates — "should it exist?" at the front and "independent & uncrowded?" at the back — are the validation-completeness gates Phase 1 adds (§5.7); the six in the middle are the statistical core already in production.

5.1 Does this signal actually predict? — the Information Coefficient

Business question: *Does ranking stocks by this factor tell me anything about which ones will go up?*

The **Information Coefficient (IC)** measures exactly this: it is the correlation between how a factor ranks stocks *today* and how those stocks actually perform *later*. A perfect crystal ball would score 1.0; pure noise scores 0.0. Real, useful financial factors live in a humble range — an IC of even 0.05 sustained across hundreds of names is valuable, because of the breadth multiplier from the Fundamental Law (§2). The platform measures IC as a rank correlation (so a few extreme stocks cannot dominate) and combines it across multiple holding horizons.

The platform's promotion hurdle is an **information ratio** (the IC's consistency-adjusted strength) with absolute value **above 0.2**.

Grounded in: Grinold and Kahn's **Fundamental Law of Active Management** — the formal reason a small but consistent IC, applied across breadth, is a genuine edge rather than a rounding error.

5.2 Did I just get lucky across many guesses? — multiple-testing correction

Business question: *I tested dozens of factors. Doesn't testing many guarantee that one looks good by accident?*

Yes — and this is where most retail quant quietly dies. If you test 50 unrelated factors, you should *expect* a couple to clear a normal "95% confidence" bar (a t-statistic above 2.0) purely by chance. The conventional threshold is far too lenient when many ideas are in flight.

The platform raises the bar to a t-statistic **above 3.0** (roughly a 0.3% chance of being a fluke, versus 5% at the conventional bar), and additionally applies a **false-discovery correction** across the whole panel of factors tested together.

This has a stark, honest consequence worth showing in full. The platform tests "not random" by shuffling the data many times (see §5.3). A cheap *nightly* check uses 100 shuffles — but with only 100 shuffles, the best possible result is a p-value of about **0.0099**, which **can never** clear the strict 0.0027 bar. So a serious promotion decision requires a *deep* run of around **5,000 shuffles** (best possible p-value 0.0002). The platform is candid about this: **nightly runs are monitoring; promotion is a deliberate deep-research decision.**

Grounded in: **Harvey, Liu and Zhu (2016)** — who, surveying hundreds of published factors, concluded the conventional $t > 2.0$ hurdle is far too lenient and recommended $\mathbf{t > 3.0}$. The platform adopts that

recommendation verbatim. Reinforced by **Harvey and Liu (2020)** on false-and-missed discoveries; **Benjamini and Yekutieli (2001)** for the dependence-robust false-discovery correction (factor results are heavily correlated, so a naïve correction under-controls); and **Feng, Giglio and Xiu (2020), "Taming the Factor Zoo"** — most candidate factors add nothing beyond the ones you already have.

5.3 Could pure randomness have produced this? — permutation testing

Business question: *How do I know this result isn't just a pattern I'd see in random data?*

The cleanest way to answer is to *manufacture* randomness and compare. The platform repeatedly **shuffles** the factor's values — thousands of times — destroying any genuine relationship while preserving the data's overall shape, and records how good a result *random* data produces. It then asks: where does the *real* result fall in that distribution of luck? If the real result is far out in the tail, randomness is an implausible explanation. If it sits in the crowd, the factor is noise.

Grounded in: a coherent suite of **permutation and randomization tests** (the "Masters methods" — Monte-Carlo permutation testing, combinatorial cross-validation, selection-bias correction, return and drawdown confidence bounds, family-wise error control, and confirming-superiority tests), adopted as a methodology with reference implementations ported for numerical fidelity. The conceptual lineage runs back to White's (2000) Reality Check for data-snooping.

5.4 Are dead companies flattering my backtest? — survivorship bias

Business question: *My historical data is full of companies that still exist. What about the ones that went bankrupt and vanished?*

This is a subtle, expensive trap. If your historical universe only contains companies that *survived* to today, your backtest is quietly cheating: it never had to live through the bankruptcies, delistings, and wipeouts that real money would have suffered. Worse, the academic literature shows the danger is not mainly in inflated *returns* — it is in inflated **predictability**: survival-filtering manufactures the *appearance* of a working signal where none exists.

Strawberry Labs measures this directly. It splices **4,858 delisted companies** back into history — assigning realistic terminal losses to names that went to zero — and re-runs the evidence. The measured damage to the platform's central momentum factor was an information-ratio gap of about **−0.035**. The validation gate now **subtracts** this haircut rather than pretending dead companies never existed.

Grounded in: **Brown, Goetzmann, Ibbotson and Ross (1992)** — survival-truncation manufactures false persistence; **Shumway (1997)** and **Shumway and Warther (1999)** — the canonical delisting-loss constants (roughly −30% for NYSE/AMEX, −55% for Nasdaq) and the finding that correcting for this can make an apparent "size premium" vanish entirely; **Carhart, Carpenter, Lynch and Musto (2002)** — the bias grows with sample length; **Beaver, McNichols and Price (2007)** — the bias is anomaly-specific in sign.

5.5 Does the edge survive trading costs? — net-of-cost reality

Business question: *Fine, the signal predicts. But after I pay to trade it, is there anything left?*

A great many "edges" are real on paper and dead after costs. The platform estimates the **bid-ask spread** for every name directly from its daily price range, applies a broker-realistic cost model, and reports performance **net of costs**. In one real paper run, the gap between the gross backtest and the net reality — the **cost drag** — was about **650 basis points** (6.5%). It then asks a second question institutions cannot ignore but a sole trader sometimes can exploit: **capacity** — how much capital can actually be deployed into a name before the trade itself moves the price.

The blunt lesson from the literature is worth stating plainly: **statistical significance is not a tradeable edge.** A factor can pass every test above and still be economically dead once you pay the spread.

Grounded in: **Corwin and Schultz (2012)** — the daily high-low spread estimator the platform implements; **Novy-Marx and Velikov (2016)** — net returns collapse for higher-turnover anomalies, motivating a turnover ceiling; **Chen and Velikov (2023)** — the average anomaly earns only 4 basis points per month *net*; **Perold (1988)** — the implementation-shortfall framework; with **Abdi and Ranaldo (2017)** and **Frazzini, Israel and Moskowitz (2015)** as cross-checks.

5.6 Am I fooling myself by tuning? — overfitting probability

Business question: *I tried many variations and kept the best one. Isn't "the best of many tries" itself a form of cheating?*

It is — and it is the most seductive form. The platform quantifies it. It computes a **Probability of Backtest Overfitting**: across many splits of the data, how often does the configuration that looked best in one half fail to be best in the other half? A high probability means you were tuning to noise. It also **deflates** reported performance by the *number of trials* you ran — a strategy chosen from 1,000 attempts must clear a far higher

bar than one tested once. Finally, it reserves a **lockbox**: a slice of out-of-sample data the search process structurally *cannot* see, so its result is an honest confirmation rather than a contaminated one.

Grounded in: **Bailey and López de Prado (2014)** — the Deflated Sharpe Ratio, which deflates by the number of trials; **Bailey, Borwein, López de Prado and Zhu** — Probability of Backtest Overfitting via combinatorially-symmetric cross-validation; **López de Prado (2018), *Advances in Financial Machine Learning*** — purged cross-validation with an embargo, so the "out-of-sample" test isn't quietly leaking through overlapping data.

5.7 Completing the funnel — the four gates Phase 1 adds

The six gates above are the platform's statistical core, in production today. They answer, with literature-grade force, *"is this pattern real?"* But a pattern can be real and still not deserve real money — and the institutional factor-validation playbook closes that gap with four more checks. **Phase 1 adds all four**, two wrapping the front of the funnel and two the back, so that by Phase-1-complete the funnel asks not only *"is it real?"* but *"should it exist, is it smooth, is it new, and is it crowded?"*

Should this signal even exist? — the economic-rationale gate. This is the front gate, and conceptually the most important. Every statistical test answers *"is the pattern real?"*; none answers *"should it be there at all?"* If you test enough patterns, some will clear even the strict bars above by sheer luck — so before any statistics, the platform now asks for a **causal story**: is the edge a payment for bearing a risk, a durable behavioral mistake, or a structural feature of the market? A factor with no plausible mechanism — the textbook example is *"stocks whose ticker begins with A outperform"* — is refused regardless of how good its backtest looks. The platform already imposes *more* multiple-testing rigor than

most institutions; this gate adds the one thing it lacked — a **causal-prior filter** — so a story-less factor faces a higher bar rather than the same one.

Is the relationship smooth? — decile monotonicity. Sort the universe into ten buckets by the factor. A genuine risk premium should climb roughly **monotonically** from the bottom bucket to the top — not earn all its apparent edge from a single freak decile. A jagged, one-bucket pattern is usually a mechanical artifact, and this gate catches it even when the headline statistics look significant.

Is it just a known factor in disguise? — neutralization / independence. Many "new" factors are simply value, size, or momentum wearing a different name. The platform regresses a candidate against the factors it has *already* validated; if the edge evaporates once those are controlled for, the factor adds nothing and is rejected. Only a factor that earns its slot — that carries information beyond the existing book — passes.

Is the trade already crowded? — crowding. This is the back gate and a primary *ruin* risk for a risk-premium harvester. When too much capital chases the same factor, forward returns compress and the position becomes vulnerable to a coordinated, violent unwind — the 2007 "quant quake" and the 2009 momentum crash are the canonical examples. The platform measures crowding today from **comomentum** — abnormal co-movement among the factor's favored names, which rises as arbitrage capital concentrates in the same trade (Lou and Polk). Two richer signals the literature recommends — **ownership concentration** (how tightly a small set of funds holds those names, from institutional filings) and the **valuation spread** between the long and short legs — are a deliberate research extension, and that same institutional-ownership data is what feeds the systemic-risk research program described in §7. A heavily crowded factor is flagged for watch — surfaced honestly, never silently traded.

Grounded in: **Harvey, Liu and Zhu (2016)** again — their framework treats the *prior* probability a factor is real as a first-class input, which is exactly the economic-rationale gate; **Patton and Timmermann (2010),**

"Monotonicity in Asset Returns" — the formal test that returns climb across sorted buckets rather than concentrating in the extreme; **Feng, Giglio and Xiu (2020), "Taming the Factor Zoo"** — the canonical test of whether a new factor adds anything beyond the existing set; and on crowding, **Lou and Polk's "Comomentum"**, **Khandani and Lo (2007), "What Happened to the Quants in August 2007?"**, and **Arnott, Beck, Kalesnik and West (2016), "How Can Smart Beta Go Horribly Wrong?"** — valuation spreads as a leading indicator of crowded, return-poor factors.

5.8 The verdict, made honest

These gates do not operate in isolation; they **stack into a single corrected verdict**, and that verdict governs two parallel lifecycles:

- A **factor lifecycle**: *experimental* → *validated* → *deprecated*.
- A **strategy lifecycle**: *incubating* → *paper* → *live*.

The two are joined by one hard rule: **a strategy cannot advance unless the factor beneath it is `validated`.** You cannot trade on a signal that hasn't earned trust.

And here is the platform's discipline made visible in its own operations: under this corrected, literature-grade gate, **no factor can be promoted on a cheap nightly check.** Promotion is deliberately a deep-research-run decision; nightly evidence is for monitoring only. A lesser system would quietly lower the bar to make promotions happen on schedule. This one tells you the bar is high and that clearing it takes real work.

> **Running example — 12-1 momentum down the funnel (real numbers).** Watch the candidate from §4 pass through each gate:

Gate	Question	12-1 momentum result	Verdict
5.1	Does it predict?	Corrected information ratio **0.49** (hurdle 0.2)	pass
5.2	Survives the false-discovery panel?	Dependence-robust adjusted p **0.008**	pass
5.3	Not random, under the *strict* hurdle?	*Nightly* (100 shuffles) p **0.0099** — cannot clear 0.0027	fail on nightly
5.3	Re-run *deep*	Research run (500+ shuffles) p **0.0020**	clears t > 3.0
5.4–5.6	Survives dead companies, costs, overfitting?	Survivorship haircut applied; cost drag measured (§6)	

Result: validated on deep evidence, not on a nightly shortcut. This is the honest "monitoring vs. decision" split, demonstrated live on the platform's own flagship factor. The same factor now carries forward into Section 6 — where it becomes a strategy and meets real trading costs.

6. From a validated signal to a real (paper) order

A validated factor is still just knowledge. Turning it into orders — safely — is the job of the final stages.

- **Portfolio construction.** The platform turns rankings into position sizes and builds the book **compliant by construction**: limits on individual position size, sector concentration, and overall leverage are baked into the weights *before* any order exists. The portfolio is born within its risk limits rather than being clipped into them afterward.
- **The execution loop (paper).** Weights become whole-share orders, which are submitted to an interactive-broker **paper account** — real fills at the real market open, not a simulator. Positions are marked daily, and a **next-day reconciliation** checks the platform's internal record against the broker's, catching any discrepancy immediately.
- **A final safety rail.** Between "decide" and "submit," a **fail-closed guard** re-checks the intended book against its risk caps. If anything breaches a limit, the *entire* submission is rejected — no partial, no override. Safety defaults to *stop*.

The honest boundary, phased — not "never"

The loop runs **paper-only through Phases 0 and 1** (July 1 – September 2026): real broker fills, no real money — *by deliberate decision, until the evidence earns the next step.* The lifecycle gate only enforces after a **63 market-day** attributed paper soak. When that soak completes — around **October 1, 2026** — the proven MVP strategy becomes the **first to trade live.** Live execution is therefore not "out of scope forever"; it

is **gated behind survived paper evidence.** That is the entire honesty argument of this paper, restated as a date.

Grounded in: **Perold (1988)** — the gap between a paper portfolio (filled at the decision price) and a real one is a measurable quantity (implementation shortfall), which the platform's next-open-fill-versus-backtest-close comparison directly captures; and **McLean and Pontiff (2016)** — the reason the paper-tracking gate compares realized results to a *decay-haircut* of the backtest, not the raw backtest. A real edge is expected to fade, so the honest bar is beating a *discounted* expectation.

> **Running example — 12-1 momentum reaches real capital.** The validated factor becomes a registered strategy, is promoted to **paper**, and is bound to the live paper book — the platform's actual MVP path. Its honest cost footnote: a real paper run measured a cost drag of about **650 basis points**, the visible difference between the gross backtest and the net-of-cost reality. The factor that *looked* strong in §5 is still strong after costs — but now you have *seen* the discount applied, not assumed it away.

The last beat is the deliberate, *dated* stop sign: 12-1 momentum trades in **paper only** through the summer, accruing attributed evidence toward the 63-market-day soak — and on **October 1, 2026** becomes the platform's **first live (real-money) strategy.** The running example ends exactly where real capital begins.

---

7. The capability landscape and roadmap

What works today

For equities, the full chain is mature and in production: data acquisition, curation, and modeling; factor engineering and the validation gates of Section 5; a factor-risk model; portfolio construction; the paper-execution loop; and an operator dashboard that surfaces every run. The platform is honest about its **partial** areas too — event capture, data lineage, capacity consumption, and off-site backup are works in progress, not finished.

The capability map

The platform is organized into eight **capability bands**. The map below is the business-level view — what each band provides and where it stands when **Phase 1 is complete**. (A live, interactive version is built into the operator dashboard.)

Band	What it provides	State at Phase-1 complete
Knowledge & Data	Acquire, clean, and model end-of-day data point-in-time	Mature for equities; **ETF universe** and **event capture** (earnings, index reconstitution, corporate actions) added

Band	What it provides	State at Phase-1 complete
Research & Discovery	Factor engineering and the validation gauntlet of §5	Mature; the funnel is **completed** — economic-rationale, monotonicity, independence, and crowding gates join the statistical core
Portfolio & Risk	Position sizing, the equity risk model, pre-trade caps, the cost model	Mature; **backtest-path risk caps** and **capacity-aware sizing** added
Backtest & Attribution	Historical simulation, tearsheets, attribution	Mature backtest; **performance attribution** (what drove the P&L) added
Lifecycle & Universe	The factor and strategy lifecycles, universe management	Mature; the research universe is **re-baselined** and **3 strategies** have walked the full funnel
Execution	The paper-trading loop, broker bridge, daily reconciliation	In service (paper); the first strategy crosses to **live** around the end of Phase 1
Platform Plumbing	Orchestration, telemetry, backup, remote operation	Mature core; **off-site backup** and **remote operator access** added

Band	What it provides	State at Phase-1 complete
Human Surfaces	The operator dashboard, domain governance, run analysis	Complete

The discipline of the map is its **negative space**: three bands an institution would fill — a formal multi-strategy capital allocator, compliance / regulatory reporting, and client / investor reporting — are **deliberately empty**, because a sole trader does not need them (§3).

The phase timeline

The three milestones below are the concrete plan:

Phase	When	What	Trading mode
Phase 0 — MVP	**available July 1, 2026**	One strategy (12-1 momentum) walked end-to-end on corrected, cost-aware, survivorship-adjusted evidence; the equity factor set	**Paper only**

Phase	When	What	Trading mode
Phase 1 — Stabilize	**July – September 2026**	Make the funnel repeatable (2 more strategies), ETF rotation, event capture, time-series momentum, a long/short experiment	**Paper only**
First live strategy	**October 1, 2026**	The proven Phase-0 momentum strategy crosses from paper to **live (real-money)** trading — the first time the loop touches real capital	**Live**

Why October 1 is not arbitrary. The Phase-0 gate enforces only after a **63 market-day** attributed paper soak. Starting July 1, roughly 63 trading days elapse around **October 1** — so the go-live date is *the soak completing*, not a calendar pick. The strategy earns live capital by surviving three months of real paper fills.

Phase 1 — what gets built (directional)

Phase 1 begins the moment Phase 0 is feature-complete and runs *in parallel* with the paper soak. Its theme is **stabilize the lifecycle in paper**

— turn the one-strategy MVP into a repeatable funnel and broaden the surface, while only the single proven strategy ever touches real money. The items below are **directional, not date-committed**:

- **Complete the validation funnel.** Add the four gates of §5.7 — economic rationale, monotonicity, independence, and crowding — plus enforcement of the out-of-sample *lockbox* and the *capacity floor* the platform already measures. *Why: these close the gap between what the platform measures and what it actually decides on.*
- **Make the funnel repeatable.** Promote **2 more strategies** through the same corrected gate → paper → bind path, so the MVP becomes a process, not a one-off.
- **A long/short experiment.** Trade a dollar-neutral long/short book in paper with market-neutral risk caps — the first portfolio shape beyond long-only.
- **ETF universe.** The cheapest high-leverage unlock: lift one exclusion and re-validate factors on a wider universe, opening sector and country rotation.
- **Event capture.** Index reconstitution, an M&A calendar, and macro releases — a multiplier that sharpens equity research now and seeds later asset classes.
- **The sweep engine.** A disciplined parameter- and universe-search with a built-in overfitting budget, so exploration cannot quietly become curve-fitting.
- **Performance attribution and capacity.** Decompose realized paper P&L into factor, sector, and selection contributions, and turn the measured per-name capacity into a deployment ceiling.
- **Operational resilience.** Off-site, restore-tested backup of the whole program state, and remote operation of the box — the cheap ruin-risk hedges a sole trader should never skip.

Throughout Phase 1, **all new work stays in paper**; the only thing that goes live is the single Phase-0 MVP strategy that earned it.

The expansion waves (beyond Phase 1)

Three principles drive the order: **cheapest unlocks first, multipliers before the things they multiply, and the multi-asset risk model last** (a cross-asset risk model with no cross-asset data is shelfware).

- **Quick wins** — the ETF universe (the cheapest, highest-leverage unlock: lift one exclusion and re-validate on a wider universe) and event capture (earnings, index reconstitution, corporate actions).
- **Foundational expansions** — asset-agnostic mechanics (time-series momentum, carry) and futures with continuous-contract rolls (the first genuinely new asset class).
- **Breadth and integration** — commodities, crypto spot and perpetuals, and finally a multi-asset risk model that ties them together.

Strategy viability — what to pursue, and what to avoid

This table is the single most important strategic picture: it shows both what is reachable and what is deliberately off the board.

Strategy family	Status	Phase / availability	Asset class
Cross-sectional momentum (12-1, residual)	MVP — goes live	**Phase 0** — paper Jul 1; **first live Oct 1, 2026**	Equities
Quality / value (fundamental)	Shipping	**Phase 0** — paper Jul 1, 2026	Equities
Low-volatility / defensive	Shipping	**Phase 0** — paper Jul 1, 2026	Equities

Strategy family	Status	Phase / availability	Asset class
Cross-sectional reversal (5-day)	Ship now	**Phase 1** — new factor (Jul – Sep)	Equities
Sector / country rotation	Near	**Phase 1** — Jul – Sep (lift ETF exclusion)	ETFs
Earnings reversal · index-reconstitution drift	Near	**Phase 1** — Jul – Sep (event capture)	Equities
Time-series momentum (trend)	One sprint	**Phase 1** — Jul – Sep (new mechanic)	Equities, ETFs
Managed-futures trend · carry	One quarter	**Wave 2 – 3** (aspirational, post-Phase-1)	Futures, commodities, crypto
Stat-arb · merger-arb · HFT · microstructure	Never	Out of scope **by design**	—

Every *in-scope* row clears through one of three paths — ship now, one sprint, or one quarter. The 12-1 momentum row is the spine of this paper and the first strategy to go live. The row is the discipline made concrete: excluded by design, not by backlog.

From harvesting to systemic-risk research

The crowding gate (§5.7) is the smallest instance of a larger question the platform is built to host: **can systemic risk be seen in the *structure* of the market before it shows up in prices?** A separate research program — written up for an academic audience in the platform's research package — generalizes crowding from one factor's co-movement into a *multi-layer network*: the stocks and the factors they load on (the factor layer above), their supply-chain links, and the institutions that co-own them, read together for the early-warning signature of a market phase transition.

Two honesty notes keep this in its lane. First, it is **research, not a shipped capability** — the platform's own single-layer experiment found connectedness moved *with* stress, not ahead of it; whether a multi-layer view does better is the open question, and the design is meant to fail cheaply if it does not. Second, it respects the §3 boundary: such a signal would enter as **risk observability, never as an optimizer that sizes the book.** The data it needs — supply-chain links and institutional ownership — is a research track separate from the trading roadmap above; the trading product does not become a machine-learning system.

---

8. Why this matters to you

If you are evaluating whether this approach is worth adopting, here is the business case in four points:

1. **Capital preservation through discipline.** The gate is the product. Its entire purpose is to stop *you* from betting real money on your own best-looking mistake — the single most common way a sole trader's account dies.

2. **An evidence funnel, not a hunch machine.** Every idea is forced to earn its way toward real (paper) capital by surviving corrected, costed, survivorship-aware, overfitting-checked evidence. You are not trusting your intuition; you are trusting a process that was *designed to mistrust intuition.*
3. **Radical honesty as a trust signal.** A platform that openly tells you what it has *not* yet proven — that nightly runs only monitor, that promotion needs deep evidence, that live trading is still ahead — is one you can believe when it says something *is* proven.
4. **Purpose-built for one operator.** No institutional bloat, no fights you cannot win. The deliberate exclusions are not limitations; they are the reason the tool is sharp.

Strawberry Labs is a **sole-trader, nightly, multi-asset risk-premium harvester whose differentiator is statistical discipline borrowed from the people who measured the problem.** It is early and deliberately phased: paper-proven from July 1, 2026, with its first live strategy on October 1, 2026. What you are being shown is not a finished product oversold — it is a discipline crossing into real capital on a date it had to *earn.*

---

9. Glossary

- **Risk premium** — persistent return paid for bearing a systematic risk or exploiting a durable behavioral pattern. The one edge in scope here.
- **Market inefficiency** — fleeting mispricing; fast-decaying and out of scope for a sole trader.
- **Factor** — a measurable property of a stock (e.g. its past-year return) used to rank the universe.

- **Signal / strategy** — a strategy applies a trading mechanic to a validated factor to produce target positions.
- **Information Coefficient (IC)** — the correlation between a factor's ranking today and actual returns later; the core test of whether a signal predicts.
- **Information ratio** — a consistency-adjusted measure of a signal's strength.
- **Permutation / null test** — repeatedly shuffling the data to manufacture randomness, then checking whether the real result stands out from it.
- **Multiple-testing correction** — raising the evidence bar to account for the fact that testing many ideas guarantees lucky-looking accidents.
- **Economic rationale** — the causal story for *why* a factor should pay (a risk borne, a behavioral mistake, a structural feature); the front-gate filter that refuses story-less, data-mined factors.
- **Decile monotonicity** — the requirement that returns climb smoothly across factor-sorted buckets rather than coming from a single freak group.
- **Neutralization / independence** — testing whether a factor adds anything beyond the already-validated factors, or is merely one of them in disguise.
- **Factor crowding** — the condition where too much capital chases the same factor, compressing forward returns and risking a coordinated, violent unwind; measured from ownership concentration and valuation spreads.
- **Survivorship bias** — the distortion from a historical universe that excludes companies that failed and disappeared.
- **Net-of-cost** — performance after subtracting realistic trading costs (spread, commission, impact).
- **Capacity** — how much capital can be deployed into a name before the trade itself moves the price.
- **Overfitting / Probability of Backtest Overfitting** — tuning to historical noise; and the measured likelihood that you have done

so.

- **Point-in-time** — the principle that, for any historical date, you use only information available on that date — never the future.
- **Paper trading** — placing real orders in a simulated-money broker account; real fills, no real capital at risk.
- **Factor lifecycle / strategy lifecycle** — the two staged paths (*experimental → validated → deprecated* and *incubating → paper → live*) a signal travels before reaching real capital.

---

10. References

The platform's hurdles are the literature's hurdles. The bibliography below is grouped by the trust question each set of sources answers, so any gate in Section 5 can be traced to its origin.

Why factors are a real edge (the premium exists) - Fama, E. and French, K. (1992). *The Cross-Section of Expected Stock Returns.* Journal of Finance. - Jegadeesh, N. and Titman, S. (1993). *Returns to Buying Winners and Selling Losers.* Journal of Finance. - Grinold, R. (1989) and Grinold, R. and Kahn, R. *Active Portfolio Management* — the Fundamental Law (IR IC × √breadth). - McLean, R. and Pontiff, J. (2016). *Does Academic Research Destroy Stock Return Predictability?* Journal of Finance.

Multiple testing and the factor zoo (luck vs. edge) - Harvey, C., Liu, Y. and Zhu, H. (2016). *…and the Cross-Section of Expected Returns.* Review of Financial Studies — the $t > 3.0$ recommendation. - Harvey, C. and Liu, Y. (2020). *False (and Missed) Discoveries in Financial Economics.* Journal of Finance. - Benjamini, Y. and Hochberg, Y. (1995). *Controlling the False Discovery Rate.* JRSS-B. - Benjamini, Y. and Yekutieli, D.

(2001). *The Control of the False Discovery Rate under Dependency.* Annals of Statistics. - Feng, G., Giglio, S. and Xiu, D. (2020). *Taming the Factor Zoo.* Journal of Finance.

Economic rationale, monotonicity, and independence (the validation-completeness gates, §5.7) - Harvey, C., Liu, Y. and Zhu, H. (2016). *…and the Cross-Section of Expected Returns.* Review of Financial Studies — the prior-probability framing behind the economic-rationale gate. - Patton, A. and Timmermann, A. (2010). *Monotonicity in Asset Returns: New Tests.* Journal of Financial Economics — the decile-monotonicity test. - Feng, G., Giglio, S. and Xiu, D. (2020). *Taming the Factor Zoo.* Journal of Finance — the test of whether a new factor adds anything beyond the existing set (the independence gate).

Factor crowding and unwind risk (§5.7) - Lou, D. and Polk, C. *Comomentum: Inferring Arbitrage Activity from Return Correlations* — a crowding measure that predicts momentum crashes and lower forward returns. - Khandani, A. and Lo, A. (2007). *What Happened to the Quants in August 2007?* — the deleveraging-cascade mechanism in crowded factor portfolios. - Daniel, K. and Moskowitz, T. (2016). *Momentum Crashes.* Journal of Financial Economics. - Arnott, R., Beck, N., Kalesnik, V. and West, J. (2016). *How Can "Smart Beta" Go Horribly Wrong?* — valuation spreads as a crowding indicator.

Permutation testing and data-snooping - Masters, T. *Permutation and randomization tests for trading-system development* — the seven-method suite (MCPT, CSCV/PBO, selection bias, return/drawdown bounds, family-wise error, confirming superiority). - White, H. (2000). *A Reality Check for Data Snooping.* Econometrica (conceptual lineage).

Overfitting and backtest validity - Bailey, D. and López de Prado, M. (2014). *The Deflated Sharpe Ratio.* Journal of Portfolio Management. - Bailey, D., Borwein, J., López de Prado, M. and Zhu, Q. (2014/2017). *Probability of Backtest Overfitting via CSCV.* - López de Prado, M. (2018). *Advances in Financial Machine Learning* — purged k-fold cross-validation + embargo.

Survivorship bias - Brown, S., Goetzmann, W., Ibbotson, R. and Ross, S. (1992). *Survivorship Bias in Performance Studies.* Review of Financial Studies. - Shumway, T. (1997). *The Delisting Bias in CRSP Data.* Journal of Finance. - Shumway, T. and Warther, V. (1999). *The Delisting Bias in CRSP's Nasdaq Data and the Size Effect.* Journal of Finance. - Carhart, M., Carpenter, J., Lynch, A. and Musto, D. (2002). *Mutual Fund Survivorship.* Review of Financial Studies. - Beaver, W., McNichols, M. and Price, R. (2007). *Delisting Returns and Their Effect on Accounting-Based Anomalies.* Journal of Accounting and Economics.

Transaction costs, capacity, and implementation - Corwin, S. and Schultz, P. (2012). *A Simple Way to Estimate Bid-Ask Spreads from Daily High and Low Prices.* Journal of Finance. - Abdi, F. and Ranaldo, A. (2017). *A Simple Estimation of Bid-Ask Spreads from Daily Close, High, and Low Prices.* Review of Financial Studies. - Novy-Marx, R. and Velikov, M. (2016). *A Taxonomy of Anomalies and Their Trading Costs.* Review of Financial Studies. - Chen, A. and Velikov, M. (2023). *Zeroing In on the Expected Returns of Anomalies.* Journal of Financial and Quantitative Analysis. - Frazzini, A., Israel, R. and Moskowitz, T. (2015). *Trading Costs of Asset Pricing Anomalies.* - Perold, A. (1988). *The Implementation Shortfall: Paper versus Reality.* Journal of Portfolio Management.

Sole-trader risk posture (supporting) - Ilmanen, A. *Expected Returns* — hedge the ruin risks that cost nothing first; the philosophy behind the deliberate exclusions.

Factor Library

This chapter is generated from the canonical platform doc, which lives in `docs/` (single source of truth).

Read all five sidecars for run `260526_6ddcd9`. Below is what each page shows, what your data actually says, and how that maps to the `experimental → validated → deprecated` lifecycle in `ops.factor`.

---

1. Factor Diagnostics — `compute_factor_diagnostics_260526_6ddcd9.json`

What it shows (per ResearchFactorDiagnosticsView.js): - Quadrant scatter: x = `-log10(ADF p-value)`, y = Shannon entropy (bits). Thresholds at p<0.05 and H 3. - Three sortable tables split by cadence (daily / quarterly / annual).

Your 79 factors (all currently `factor_status: experimental`):

Quadrant	Count	Read
Healthy (stationary + H 3)	**7**	`beta`, `market_beta_252d`, `ma_50d`, `residual_momentum_252d_eod`, `rsi_14d`, `amihud`, `bb_pct`
Thin (stationary, low entropy)	32	Mostly the quarterly fundamentals (`net_margin`, `gross_margin_q`, `momentum_12_1` variants, etc.) — they pass ADF but the values are nearly degenerate over the cross-section
Drift (non-stationary, high entropy)	4	`pct_from_52w_high`, `vwap_20d`, `adv_20d`, `momentum_6m` — trending series, not statistically stationary
Excluded (non-stationary + low entropy)	36	Most annual factors — 10–12 observations is too few for ADF to be reliable at all

Annual factors (n_obs=10–12) and quarterly factors (n_obs=36) are too short for ADF to be informative. The ADF column for those rows should be treated as decorative, not decisive.

Yes — and the right answer is mostly **not** "add data." Here's why, and what actually moves the needle.

Where the constraint actually is

Looked at factor_diagnostics_service.py: the diagnostic computes a daily **cross-sectional mean** of the factor across the allowlist, then runs ADF + entropy on *that single time series*. Per-cadence lookback already adapts:

Cadence	Lookback	min n_obs
daily	252 days	60
quarterly	3650 days (~10 yr)	8
annual	4380 days (~12 yr)	5

So the test is ingesting all the history that exists. The 10–12 obs you see for annual factors *is* the full Strawberry history — extending the lookback alone won't help until more data is in Bronze.

Options, ranked by effort × payoff

1. (Best) Stop using ADF on cross-sectional means for slow-cadence factors

ADF on a 12-point series has essentially no power — the test was designed for T 50. For stock-picking factors what actually matters is **cross-sectional rank stability and dispersion**, not whether the *grand mean* of ROIC across the universe is mean-reverting (it never is — fundamentals trend with the economy). Replace or supplement with:

- **Cross-sectional dispersion stability** — for each period, compute IQR (or std) of the factor across instruments; then test whether *that dispersion series* is stationary. This measures "is the factor still discriminating?" which is the real question.
- **Rank persistence** — Spearman between instrument ranks at t and t+1. A factor that ranks the same names quarter-after-quarter is informative regardless of level non-stationarity.
- **Half-life from AR(1)** — fit on the cross-sectional mean and report `-log(2)/log(| |)`. Works with N=10 (won't be precise, but won't be meaningless either).

This is a code change in `_diagnostics_math.py` and the sidecar schema — no new ingestion.

2. Test the change, not the level

Half your fundamental factors are non-stationary in levels by construction but stationary in *changes*:

- `roic` ADF p=0.97; ΔROIC almost certainly passes.
- `roa`, `roe`, `gross_margin_moat`, `incremental_roic` — same pattern.

If the downstream signal uses YoY change anyway (`revenue_growth_yoy`, `earnings_growth_yoy` already do — and they pass ADF cleanly), the diagnostic should match. For each fundamental factor declare in `config/factors/<id>.yaml` whether the diagnostic should run on `level` or `first_difference`; default to `first_difference` for ratios and margins. Cheap, mechanical fix.

3. Switch to panel unit-root for cadences with low T

Since the service already pools across the universe, you already have what panel tests need: short-T, large-N. **Im-Pesaran-Shin (IPS)** or **Pe-**

saran's CIPS are the standard tools for `T=10, N=3000` panels. Statsmodels doesn't ship them but `arch` or `linearmodels` does (or ~80 lines hand-rolled). Far more power than per-series ADF, no new data needed.

Implemented in F-134 (`docs/backlog/feature-panel-unit-root-diagnostic-design-brief.md`) — Pesaran (2007) CIPS, hand-rolled in `src/sbdiag/services/_diagnostics_math.py`, case II (intercept-only), `p_lags=0`. Runs on quarterly + annual factors with `N  100` instruments and `T  8/5` observations; daily factors are routed to `panel_test_method='skipped_daily'` by design. The v3 sidecar surfaces `panel_test_method`, `panel_test_statistic`, `panel_test_pvalue`, `panel_test_n_instruments`, `panel_test_n_dates` per item plus an `n_panel_tested` envelope tally; no third-party dependency added (neither `linearmodels` nor `arch` ships CIPS).

4. Emit `insufficient_history` instead of a low-power p-value

Today, an annual factor with n_obs=12 returns `is_stationary: false` (because p=0.4 is above 0.05) — but `false` here is meaningless; ADF just couldn't reject. The honest output is a fourth state alongside ok/error/insufficient: `low_power`, gated on per-cadence n_obs thresholds (e.g. ADF needs 50 obs, KPSS 30, AR(1) half-life 10). The current `status: ok` quadrant scatter is overconfident — those 36 "excluded" annual factors aren't necessarily bad, they're just untestable.

5. (Only if you really want more data) Backfill fundamentals to 25–30 years

FMP's annual/quarterly endpoints typically go back to ~1995 for liquid US large/mid caps, ~2000 for the broader Russell 3000. You'd gain:

- Annual: 12 → ~30 obs (enough for ADF to have power)
- Quarterly: 36 → ~120 obs (already healthy, would become very healthy)

- Cost: ~3× the fundamental Bronze footprint, one-shot historical ingest via the CLI subcommands `python -m sborchestration.cli backfill-annual <start> <end>` / `backfill-quarter <start> <end>` (Bronze-only). There is **no** `AnnualBronzeIngester.__main__` block — that mechanism was corrected by F-140; see CLAUDE.md §13.

Worth doing eventually, but it doesn't *fix* the diagnostic — it just gets the bad diagnostic to a workable n. Doing 1–4 first is higher leverage and cheaper. Note the backfill alone is inert unless `sbdiag.settings.ANNUAL_LOOKBACK_DAYS` / `QUARTERLY_LOOKBACK_DAYS` are also widened (both `11000` 30 yr after F-140), or the diagnostic trailing window caps n_obs regardless of Gold depth. This is exactly what F-140 (milestone m-113) scopes.

6. Things I'd *not* recommend

- **Forward-fill quarterly → daily** to inflate n. Several factors already do this implicitly (`revenue_growth_yoy` shows n_obs=1219) and the result is a sky-high ADF rejection that is spurious — forward-filled series have artificial autocorrelation that biases ADF *toward* stationarity. Worse than the original problem.
- **Block-bootstrap synthetic histories**. Doesn't add information, makes the result look more confident than it is. Mostly stats theater.

What I'd actually do

Two-PR sequence, both small:

1. **PR-1 (no data)**: add `diagnostic_transform: level | first_difference` to `config/factors/*.yaml` (default `first_difference` for ratio/margin factors); add cross-sectional dispersion +

AR(1)-half-life + rank-persistence metrics to the sidecar; gate `status='low_power'` per cadence. This alone will reclassify ~30 of your 36 "excluded" annual factors into a usable diagnostic state.

2. **PR-2 (only if PR-1 leaves gaps)**: implement IPS panel unit-root in `_diagnostics_math.py`, surface alongside the per-series ADF for quarterly/annual factors. Borrows strength from the cross-section instead of pretending T=12 is enough. **Landed as F-134** with Pesaran CIPS rather than IPS — CIPS handles cross-sectional dependence (driven by common market shocks) which IPS does not, see F-134 brief §1 for the discussion.

Backfilling fundamentals is a separate, larger task — only worth it once you've exhausted the analytical fixes above.

---

2. Per-Factor IC — `compute_ic_260526_6ddcd9.json`

What it shows (per ResearchIcView.js): matrix of 79 rows × 5 horizon columns (1/5/10/21/63 day forward returns). Each cell colour-coded by |IC_IR|: green 0.5, amber 0.2–0.5, red <0.2. Right-edge bar = Fisher-z composite (single value per factor — same across the row by design).

Promotion horizon is 21d. Standouts at h=21d:

Group	Factor	mean_IC@21d	IC_IR@21d	Sign
Momentum (passes)	`residual_momentum_252d_skip21_zscore`	0.0268	**+0.724**	+

Group	Factor	mean_IC@21d	IC_IR@21d	Sign
Momentum (passes)	`momentum_12_1` / `momentum_12m_1m` (duplicates)	+0.063	**+0.586**	+
Momentum (passes)	`momentum_12_1_vol_scaled`	+0.061	**+0.629**	+
Value (passes)	`value_composite`	+0.090	**+0.750**	+
Quality / earnings (weak-moderate)	`internal_forecast_eps`, `operating_margin_q`, `pct_from_52w_high`	0.07–0.13	0.30–0.52	+
Low-vol / size (inverted)	`volatility_30d`/`60d`/`126d`/`252d`, `market_cap`, `amihud`, `pe_ratio`	-0.07 to -0.13	-0.5 to -0.7	−
Sign-flip with horizon	`piotroski`, `earnings_yield`, `roa`, `fcf_yield`	neg at h=1, pos at h 21	n/a	depends on horizon

The sign-flip group is the classic value-investor pattern (wrong direction intraday, right direction at month+). The high-magnitude negative IRs (vol, size) are *real* signal — the raw factors are anti-predictive, so a strategy would trade them inverted.

Two things to flag: - **`beta` and `market_beta_252d` are mathematically identical** (ADF stat, IC, entropy all match to 14 digits). One is a stale alias — delete it. - **`momentum_12_1` and `momentum_12m_1m` are identical** for the same reason. - `market_beta_252d` only has horizons `(21, 63, 126)` while everything else has `(1, 5, 10, 21, 63)`. This is a config divergence in `config/factors/market_beta_252d.yaml` worth fixing.

---

3. Factor Contribution — `factor_contribution_260526_6ddcd9.json`

What it shows (per ResearchFactorContributionView.js): XGBoost gain / weight / cover importances of the *source factors* that feed each composite. Only meaningful for composites.

Your data on this run: 79 of 80 items are `status: single_input` (atomic — nothing to attribute). Exactly one composite produced contribution rows: `residual_momentum_252d` (2 inputs):

Input	Gain	Weight	Cover
`momentum_12m_1m`	4.99e+29	1516	2.1M
`market_beta_252d`	2.10e+29	1146	4.4M

Reframed in F-138 (fixed in `feature-composite-factor-library-audit-design-brief.md`, milestone m-111):

- **The composite library is structurally empty by design — not by a classification bug.** The four named "composites" (`value_composite`, `quality_composite`, `qmj_composite`, `momentum_12_1_vol_scaled`) are *correctly* declared `kind: atomic by design` in `config/factors/*.yaml`. They are computed upstream in `src/sbfactors/fundamental/value.py` and siblings as single z-score blend columns and lifted via `FactorValueLiftService` as one `factor_id` each. They will *always* land as `single_input` in this sidecar until a future feature builds `FactorCompositionService` (promised in CLAUDE.md §1).

- **`residual_momentum_252d` is the only true composite — and it is `deprecated`** per `config/factors/residual_momentum_252d.yaml:23` (B-F-108.1 / TASK-907). F-138 routes deprecated factors to a new `status='deprecated_skipped'` sidecar item with zero rows in `ops.research_factor_contribution`; the SPA renders a third collapsible "Deprecated (skipped)" section. **Post-F-138, this composite no longer pollutes the matrix.**
- **The 4.99e+29 gain magnitudes are fixed at the booster, not at the data.** F-138 z-scores `fwd_return` to unit variance, passes `base_score=0.0`, and pins `reg_alpha=0.1` + `reg_lambda=1.0` on the XGBRegressor. Gain magnitudes are now in standardised-return units (typically O(0.01)–O(1)); the relative ordering of source factors within a composite is preserved. Sidecar `schema_version` bumped 1 → 2 with a new top-level envelope key `y_standardized: true` so consumers can interpret the rescaled units.

---

4. Alphalens — `write_alphalens_tearsheets_260526_6ddcd9.json`

What it shows (per ResearchAlphalensView.js): mean return by quantile (typically 5 quintiles), top-minus-bottom spread, top/bottom quantile IC, and turnover per factor × horizon. Tearsheet HTMLs sit next to the sidecar.

Status: 79 × 5 = 1965 quantile rows, `n_skipped=0`.

Fixed in F-137 (sidecar schema_version 2). Forward returns are now winsorized at `(0.01, 0.99)` per-date cross-section before quantile aggregation, and the sidecar emits a `non_monotonic_quintiles` tri-state flag per (factor, horizon). The pre-fix sample from this run is retained on every `summary[]` row as `raw_top_minus_bottom_spread` for forensic

comparison — e.g. `adv_20d` h=21 carried `raw_top_minus_bottom_spread = -0.302` against the +44 / +37 / -36 absurdities listed below.

factor	h=21 spread (pre-fix)
`ps_ratio`	+44.78
`pb_ratio`	+37.40
`market_cap`	+37.06
`pfcf_ratio`	-36.78
`fcf_yield`	-3.31
`ma_50d`	-0.99

Post-fix `top_minus_bottom_spread` magnitudes are bounded — Tier 4 acceptance in F-137 brief §15 requires `|top_minus_bottom_spread| < 1.0` at every horizon and `< 0.30` for `pb_ratio` / `pfcf_ratio` at h=21. Verification fills in here from the first post-deploy nightly. The cumulative-vs-period suspect raised alongside this bug was audited and closed in docs/architecture/return-convention-audit.md; the root cause is exclusively un-winsorized cross-section outlier dominance.

Non-monotonic quintiles is now a first-class flag — the SPA renders a `non-monotonic` chip whenever top- and bottom-quintile IC carry the same sign. Pre-fix examples included `ps_ratio` and `bb_lower_20` at h=21d.

Turnover values (0.06–0.7) look sensible — momentum/MACD around 0.5–0.7 (high churn), fundamentals around 0.05–0.15 (slow).

5. Factor MCPT — `compute_factor_mcpt_260526_6ddcd9.json`

What it shows (per ResearchFactorMcptView.js): Masters MCPT — empirical p-value of the factor's IC_IR against `B=100` bar-permuted null realizations, plus the bias-corrected `unbiased_ic_ir`.

This is the formal validation gate. Right now it's barely usable, for three reasons:

Bucket	Count	Why
`p = 1.000` (ceiling)	**37**	Overlap-corrected null exceeds the actual IC. Every high-IC daily/momentum/vol factor falls here. Their `unbiased_ic_ir` is sharply negative (e.g. `beta` → -6.8, `volatility_252d` → -4.3, `momentum_1m` → -2.1)

5. Factor MCPT — `compute_factor_mcpt_260526_6ddcd9.json`

Bucket	Count	Why
`p = 0.010` (floor)	**33**	`null_mean_ic_ir` `0.000` with zero variance — the permutation null collapses for slow-moving fundamentals because there isn't enough cross-sectional dispersion at annual/quarterly cadence
Intermediate	5	These are the only interpretable rows
Real significance	**4**	`bb_lower_20`, `ma_50d`, `ma_200d`, `vwap_20d` — non-collapsed null *and* positive unbiased IR (0.21 to 1.35)

`B=100` puts the p-value floor at `1/(B+1)` `0.010`. Half of your factors are pinned at that floor with a degenerate null and half are pinned at the ceiling with an overcorrected null. **Without raising B to 1000 and verifying that the bar-permutation overlap correction is sized right for each cadence, MCPT cannot decide promotion for the bulk of the list.**

F-191 — Per-factor nightly MCPT escalation — RETIRED (O-088 + O-090)

This mechanism no longer exists. It let an **individual** promotion-candidate factor run an elevated `B=500` MCPT during an ordinary `RunMode.NIGHTLY` (which otherwise ran at `B=100`, `min p  0.0099` — structurally unable to clear the Harvey-Liu-Zhu floor `HLZ_PVALUE_FLOOR = 0.0027`), bounded by a global cap.

O-088 made `compute_factor_mcpt` **deep-run-only** — it now runs at `B=500` for **every** factor on the Friday `NIGHTLY_FULL` deep run (and ad-hoc `RESEARCH_DAY` runs) and not on plain weekday nightlies at all. Since the deep-run `B` already equals the old escalated `B`, per-factor escalation only ever bought 4-day-earlier timing at identical rigor, so O-088 deleted the escalation machinery (`_escalated_factor_ids`, the `DEFAULT_MCPT_PERMUTATIONS_ESCALATED` / IC-IR-threshold settings, the sidecar `escalation` block). **O-090** then removed the vestigial operator surface — the `research.mcpt_escalate_nightly` factor-config flag, the `apply-escalation` CLI, and the `DEFAULT_/ENV_MCPT_NIGHTLY_ESCALATION_CAP` constants. A strongest candidate now simply accrues its `B=500` evidence on the next deep run (surfaced as an *eligible — awaiting the weekly deep-run MCPT* annotation by `next-actions`).

Brief: `optimization-mcpt-permute-weekly-cadence-design-brief.md` (O-088). Historical: `feature-nightly-mcpt-escalation-design-brief.md` (F-191).

What this tells you about your factor list

Cross-page synthesis (and ignoring the data-quality red flags above):

1. **Duplicates to remove now**: `beta` `market_beta_252d` (identical), `momentum_12_1` `momentum_12m_1m` (identical). That's 2 factors you can delete with zero information loss.

2. **The "best by IC" cluster** is small, coherent, and consistent with academic literature:

   - **Momentum**: `residual_momentum_252d_eod` (IR=0.72), `momentum_12_1_vol_scaled` (IR=0.63), `momentum_12_1` (IR=0.59) — same family, the residual version dominates
   - **Inverted low-vol / size**: `volatility_*` (IR -0.55), `market_cap` (IR -0.5) — strong negative predictive power, would be traded short
   - **Slow value/quality**: `value_composite` (IR=0.75), `internal_forecast_eps` (IR=0.50), `operating_margin_q` (IR=0.31)
   - **Price-level**: `pct_from_52w_high` (IR=0.52), `ma_200d` (IR weak but MCPT-clean), `ma_50d` (MCPT-clean)

3. **The "thin/excluded" quadrant on diagnostics maps perfectly to the "p=0.010 null-collapse" cluster on MCPT.** Annual fundamentals like `ohlson_o`, `roic_spread_trend`, `rule_of_40`, `qmj_composite`, etc. don't have enough observations for either test to be meaningful. Don't promote any of them on this run's evidence — extend the lookback or downsample less aggressively first.

4. **The drift quadrant** (`pct_from_52w_high`, `vwap_20d`, `adv_20d`, `momentum_6m`) is non-stationary by ADF but shows real IC — typical of trending price features. The "stationarity" check is a *guardrail* for these, not a kill switch; treat the diagnostics red as a known property, not a failure.

5. **Your composite layer is one-deep.** `residual_momentum_252d` is the only composite that produced contribution rows, and the named composites (`value_composite`, `quality_composite`, etc.) didn't. Either they're being misclassified at load time, or they're declared in YAML but `RecipeMethod` runs but doesn't emit contribution rows — worth checking `sbfactorcontrib` and the F-114 task.

Lifecycle impact per factor group

All 79 factors are currently `experimental`. Per CLAUDE.md §1 (Quick Reference), the validation gate is **MCPT p-value + IC IR**. Given the data quality issues above, here's the honest read:

Can move toward `validated` once MCPT is fixed (B 1000, overlap correction audited): - `residual_momentum_252d_eod` — strongest combined evidence (IC IR=0.72 at 21d, diagnostics healthy, related composite has clean contribution attribution) - `momentum_12_1_vol_scaled` — IR=0.63, MCPT currently ceiling-pinned but vol-scaling is what should let it survive overlap correction - `value_composite` — IR=0.75 at 21d, sign-flips at h=1 (don't trade intraday) - `ma_50d`, `ma_200d`, `vwap_20d`, `bb_lower_20` — the 4 factors that already pass MCPT with non-degenerate nulls today

Stay `experimental`, hold for fixes: - The volatility / size / amihud cluster — strong negative IR is real signal, but you need to confirm with `unbiased_ic_ir` after a proper MCPT run, then *promote them with the inverted convention* - All annual fundamentals with n_obs 12 — cannot be validated on current history. Extend lookback (or accept they won't promote until you have ~5 years of fundamentals).

Candidates for `deprecated` immediately: - `beta` (use `market_beta_252d` — or vice versa, just pick one) - `momentum_12m_1m` (duplicate

of `momentum_12_1`) - The most clearly degenerate annual factors with `null_mean_ic_ir = 0.000` AND |actual_IC| $<$ 0.01 (e.g. `gross_margin_moat` actual=0.010, `wacc` actual=-0.010, `operating_margin_moat` actual=-0.008) — they're near-constant in cross-section

Before promoting anyone, fix these three things or the lifecycle gate is unreliable: 1. ~~**Raise MCPT `n_permutations` from 100 to 1000** — the 0.010 floor is masking real differences within the "passing" group.~~ **RESOLVED in F-136** (2026-05-26): nightly default bumped to `mcpt_permutations_nightly = 1000` (p-floor $\rightarrow$ 0.001); methodology switched from per-symbol bar-permutation to per-date cross-sectional factor permutation (`permute_factor_cross_sectional` in `sbpermtest.permutation_core`). The ceiling-bug factors (`beta` null_mean=+6.86, etc.) and floor-bug factors (`gross_margin_moat` null_std=0) are both addressed — the latter is now routed to `skip_reason='insufficient_cross_section'` when fewer than `mcpt_min_valid_dates=20` dates have cross-sectional dispersion. See `docs/backlog/feature-F-136-mcpt-correctness-design-brief.md`. §5 above is the pre-F-136 evidence; post-F-136 bucket counts land after the first nightly run with the new method. 2. **Audit the Alphalens spread scaling** — 44$\times$ quintile spreads aren't physical; either winsorize, or fix the cumulative-vs-period return convention in `sbalphalens`. 3. **Fix the `market_beta_252d` horizon list** (currently 21/63/126 vs. everyone else's 1/5/10/21/63) — config divergence in `config/factors/market_beta_252d.yaml`.

Factor & Strategy Lifecycle

This chapter is generated from the canonical platform doc, which lives in `docs/` *(single source of truth).*

Factor & Strategy Lifecycle

How a factor is born, earns evidence, gets validated, becomes part of a strategy, and how a strategy is created and promoted to live trading.

Last updated: 2026-06-02

> There are **two parallel lifecycles** — the **factor** lifecycle and the **strategy** lifecycle. A strategy is a thin wrapper that points at *one validated factor* and applies a generic portfolio mechanic to it. The factor lifecycle *gates* the strategy lifecycle: a strategy may not promote to production until its underlying factor is `validated`.

---

1. A factor is born — a YAML declaration

Everything starts as a file: `config/factors/<factor_id>.yaml`. A real example (`config/factors/ma_50d.yaml`):

```yaml
factor_id: ma_50d
status: experimental        # ← lifecycle state
kind: atomic                # atomic (one _f column) | composite (recipe over factors)
nightly_enabled: false
source_column: ma_50d_f     # lifts straight from gold.fact_eod_features
source_table: fact_eod
```

```yaml
style: momentum
research:
  ic_horizons: [1, 5, 10, 21, 63]
  promotion_mcpt_threshold: 0.05   # ← the gate knobs live here
```

Two kinds of factor:

- **Atomic** — lifts a single pre-computed `_f` column. (Feature columns end in `_f`, computed in DuckDB SQL by `sbfactors`.) `ma_50d` → `ma_50d_f`.
- **Composite** — declares `source_factors` + a `recipe` (`linear_blend`, `ic_weighted_blend`, or `ols_residualize`) that combines other factors.

`FactorConfigService` loads and validates these. The `register_factors` step writes one row per factor into the **`ops.factor`** registry table with `status='experimental'`.

> **Key idea:** a factor is *first-class and separate from any strategy.* It exists, earns evidence, and gets validated entirely on its own — before any strategy references it.

---

2. The factor lifecycle: `experimental` → `validated` → `deprecated`

The status lives in `ops.factor`:

2. The factor lifecycle: `experimental → validated → deprecated`

Status	Meaning
`experimental`	Newly registered. Not yet trusted. **Every factor sits here today.**
`validated`	Passed the research gates. Usable by strategies in production.
`deprecated`	Past its usefulness; existing strategies grandfathered.

How a factor earns its evidence (nightly research flow)

Each night the research sub-flow (`research.py`) runs the *factor block* against every active factor and writes evidence to the `ops.research_*` tables:

```
compute_factor_diagnostics  → ops.research_factor_diagnostics  (is it stationary? degenerate
compute_ic                  → ops.research_ic_summary          (IC IR: does it predict retur
compute_factor_mcpt         → ops.research_factor_mcpt         (is the IC statistically real
+ alphalens, contribution, (research-day) FWER / selection-bias / superiority
```

- **Diagnostics** — stationarity, entropy, dispersion stability, panel unit-root (CIPS). Answers: *is this factor well-behaved and still discriminating?*
- **IC (Information Coefficient)** — cross-sectional Spearman rank correlation between factor values and forward returns; the `ic_ir` (IC / std of IC) is the signal-to-noise ratio. Answers: *does ranking by this factor sort future winners from losers?*
- **MCPT (Monte-Carlo Permutation Test)** — shuffles factor values across symbols and recomputes IC thousands of times to build a null distribution. The p-value answers: *is the observed IC real, or could random noise produce it?*

How the status actually advances — F-139 (not yet built)

Today **nothing reads the evidence back into the lifecycle**, so every factor is stuck at `experimental`. **F-139 — Factor Lifecycle Promotion Automation** (design brief) is the service that closes this loop. Each night its `FactorPromotionGate` will:

- `experimental → validated` when diagnostics are healthy **AND** MCPT p < 0.05 **AND** $|\text{IC IR}| > 0.2$ at the 21-day horizon.
- `→ deprecated` when a factor degenerates (or a validated one falls out of band) for 3 consecutive nightlies.

Every verdict is recorded in a new `ops.factor_lifecycle_event` audit table; every actual transition also writes to the `ops.factor_promotion` event log.

> F-139 is the automated bridge from "we have evidence" to "the status reflects it." It is the user-visible payoff of the entire research-phase stack.

---

3. How a factor becomes part of a strategy

A **strategy** is a `(factor_id, mechanic, params)` triple. It does **not** invent its own signal — it applies a *generic portfolio mechanic* to a factor's values. A real example (`config/strategies/classic-momentum.yaml`):

```yaml
factor_id: momentum_12m_1m          # ← the factor it rides
mechanic: long_short_quantiles      # ← how to turn factor values into positio
params:
  long_q: 10                        # long the top decile
  short_q: 10                       # short the bottom decile
```

```yaml
  holding_days: 21
universe_id: us_large_cap
benchmark: SPY

strategy:
  name: classic-momentum
  version: "1.0.0"

backtest:    { from_date: 2024-01-02, capital: 500000, long_only: false, ... }
universe:    { size: 500, min_market_cap_billions: 1.0, exchanges: [NYSE, NASDAQ] }
promotion_gates: { min_elapsed_days: 30, min_green_nightlies: 5, sharpe_min: 1.1, max_drawdo
```

The mechanic comes from the `StrategyMechanic` enum (`sbcontracts.strategy_config`) — v1 ships `long_short_quantiles`, `long_only_top_n`, `dollar_neutral_rank`. The factor supplies the *ranking signal*; the mechanic supplies the *recipe for turning ranks into a portfolio*. Swap `factor_id`, keep the mechanic → a different strategy with zero new signal code.

The hard rule: a strategy may only promote past `incubating` once its underlying **factor's status is `validated`**. The factor lifecycle gates the strategy lifecycle — validated factors are the only legal inputs to production strategies.

---

4. How a strategy is created and its own lifecycle

Creating a strategy = **write the YAML**. Per CLAUDE.md §2 constraint 13, every strategy *must* have `config/strategies/<name>.yaml`, or `StrategyConfigService` raises `StrategyConfigNotFoundError`. Gate thresholds, backtest parameters, universe filters, and benchmarks are declared in that YAML — never hardcoded in Python.

The strategy then enters its own state machine, enforced by `StrategyRegistry`:

```
registered → incubating → paper → live → retired

                      retired     retired       (rollback → reverts to prior
```

Stage	What it means
`registered`	Strategy exists in `ops.strategy`.
`incubating`	Accruing a track record. Every nightly `tick_promotion` adds a `tick` event.
`paper`	Promoted to simulated execution (paper-trading backend).
`live`	Real execution backend.
`retired`	Done; or rolled back to a prior version.

What drives strategy promotion

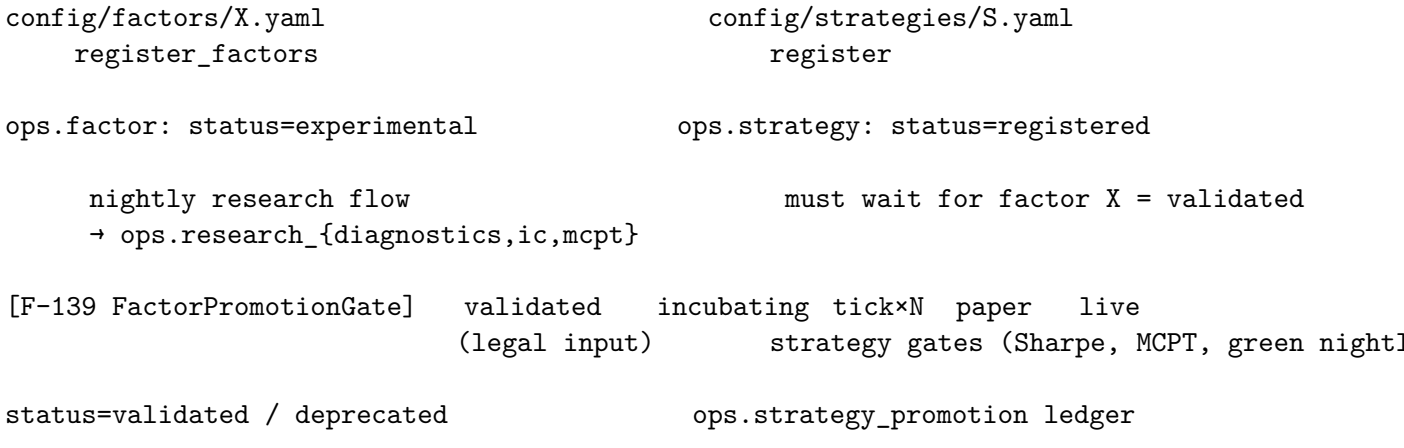
The gates in `gates.py`, AND-composed and evaluated nightly by `tick_promotion`:

- **`ElapsedMarketDaysGate`** — accrued `min_elapsed_days` tick events.
- **`ConsecutiveGreenNightliesGate`** — the last N nightlies all succeeded.
- **`BacktestMetricBandGate`** — Sharpe `sharpe_min`, max drawdown `max_drawdown_max` (read from the YAML `promotion_gates` block).
- **`MCPTSignificanceGate`** — the strategy's own MCPT p-value. This is a *single time-series test* on the backtest P&L, distinct from the *factor's* cross-sectional MCPT in §2.

Every register/promote/tick/rollback is a row in `ops.strategy_promotion`, and each transition runs in one DuckDB transaction so the headline `ops.strategy` row and its ledger event commit together.

---

5. The whole arc, end to end



One-line summary: factors are validated *cross-sectionally* (do their values rank winners vs losers, provably?) and live/die in `ops.factor`; strategies are validated *as portfolios* (does this mechanic on this validated factor make money with acceptable risk?) and live/die in `ops.strategy`. F-139 is the missing automation that moves a factor from `experimental` to `validated` so strategies have something legal to ride.

Reference map

Concern	Location
Factor declaration	`config/factors/<id>.yaml`, `FactorConfigService`
Factor registry / lifecycle	`ops.factor`
Factor evidence	`ops.research_factor_diagnostics`, `ops.research_ic_summary`, `ops.research_factor_mcpt`
Factor promotion automation	F-139 — design brief (planned)
Feature `_f` column compute	`sbfactors` (DuckDB SQL)
Strategy declaration	`config/strategies/<name>.yaml`, `StrategyConfigService`
Strategy mechanics	`StrategyMechanic` enum in `sbcontracts.strategy_config`
Strategy registry / lifecycle	`StrategyRegistry`, `ops.strategy`, `ops.strategy_promotion`
Strategy promotion gates	`gates.py`, `tick_promotion`
Research flow orchestration	`research.py`
Backtest fan-out (F-120)	`sbbacktest` (`StrategyBacktestService`)

See also: CLAUDE.md §1 (Architecture, factor-centric model), `docs/research_spec.md` (factor/strategy separation D10), and `docs/factors.md` (per-factor evidence read).

Portfolio Construction

This chapter is generated from the canonical platform doc, which lives in `docs/` *(single source of truth).*

Portfolio Construction — Momentum Backtest

How portfolio construction works in the momentum backtest (`src/sbresearch/momentum_backtest.py`), what Zipline gives you to work with, and a prioritized list of improvements.

---

1. Current State

1.1 Overview

The momentum backtest is a **Zipline-driven, equal-weighted long/short decile portfolio** wired to a declarative `StrategyConfig` loaded from `config/strategies/<name>.yaml`. There is no optimizer, no risk model in the construction step, no sector caps, no turnover control, and no signal-weighted sizing.

1.2 Pipeline & Ranking (signal → buckets)

`make_pipeline()` at `momentum_backtest.py:201-223` builds a Zipline `Pipeline`:

1. **Universe filter** — `AverageDollarVolume(window_length=20).top(universe_size)` keeps the top 500 names by 20-day ADV (`momentum_backtest.py:203-204`).
2. **Signal factor** — either `MomentumFactor` (precomputed 12-1 vol-scaled momentum at `momentum_backtest.py:176-183`) or `ResidualMomentumFactor` (252-day residual momentum at `momentum_backtest.py:186-193`). Both are 1-bar `CustomFactor`s that read columns from the `GoldEodFeatures` DataSet — features are precomputed in Gold, not recomputed in-engine.
3. **Bucket selection** — `signal.top(long_count, mask=universe)` and `signal.bottom(short_count, mask=universe)` produce boolean columns `longs`/`shorts` (`momentum_backtest.py:213-214`). Defaults are 50/50.

1.3 Per-day refresh

`before_trading_start()` (`momentum_backtest.py:255-259`) pulls `pipeline_output("momentum_pipe")` daily and stores the `longs`/`shorts` symbol lists on `context`. Shorts list is forced empty when `long_only=True`.

1.4 Rebalance & Sizing

`rebalance()` (`momentum_backtest.py:262-285`), scheduled monthly or weekly at market_open + 30min:

- **Closes** any current position not in `desired = longs  shorts` via `order_target_percent(asset, 0.0)`.
- **Long sizing** — equal-weight: `long_weight = long_budget / len(longs)` where `long_budget = 1.0` if `long_only`, else `0.5`.
- **Short sizing** — equal-weight: `short_weight = -0.5 / len(shorts)` (long-short only).
- Calls `order_target_percent(asset, weight)` for each leg, gated by `data.can_trade(asset)`.

A default long-short run targets **±1% of NAV per name** (50% / 50 names); long-only targets **+2% of NAV per long** (100% / 50 names).

1.5 Frictions

`initialize()` (`momentum_backtest.py:249-250`) sets `PerShare(cost=0.001, min_trade_cost=1.0)` commission and `VolumeShareSlippage(volume_limit=0.025, price_impact=0.1)`. No `set_long_only`, no `set_max_leverage`, no `set_max_position_size`, no `set_max_order_count`, no `set_asset_restrictions`.

1.6 Configuration

All parameters (`long_count`, `short_count`, `long_only`, `rebalance`, `universe.size`, capital, date range) come from the strategy YAML (`classic-momentum.yaml:8-15`, `residual-momentum.yaml:8-15`), with CLI overrides applied in `_backtest_from_args()` (`momentum_backtest.py:728-750`). Both YAMLs currently use `long_count=50, short_count=50, long_only=false, rebalance=monthly`.

---

2. What Zipline Provides

Zipline does not have a `set_portfolio_constructor(fn)` hook. Portfolio construction is **your function** scheduled via `schedule_function(rebalance, date_rule, time_rule)`. Zipline supplies:

2.1 Order primitives (`zipline.api`)

All accept `limit_price`, `stop_price`, and `style` (`MarketOrder` | `LimitOrder` | `StopOrder` | `StopLimitOrder`):

Function	Specifies	Notes
`order(asset, amount)`	share count (int)	hand-rolled
`order_value(asset, value)`	dollar value	
`order_percent(asset, pct)`	% of NAV, **delta**	additive
`order_target(asset, target)`	absolute share count	
`order_target_value(asset, target)`	absolute dollar value	
`order_target_percent(asset, target)`	absolute % of NAV	what we use now
`batch_market_order(share_counts)`	`pd.Series[Asset→int]`	one call for many fills

2.2 Trading controls (called once in `initialize`)

`set_long_only`, `set_max_leverage`, `set_min_leverage`, `set_max_position_size`, `set_max_order_size`, `set_max_order_count`, `set_asset_restrictions`, `set_do_not_order_list`, `set_slippage`, `set_commission`, `set_cancel_policy`.

2.3 Pipeline-side

`attach_pipeline` + `pipeline_output` for ranking/screening. The `Pipeline` API supports group-based normalisations (`.demean(groupby=...)`, `.zscore(groupby=...)`) which can deliver sector-neutral signals without a downstream optimizer.

2.4 No `zipline.optimize`

The convex optimizer (`MaximizeAlpha`, `TargetWeights`, `MaxGrossExposure`, `PositionConcentration`) shipped on Quantopian is **not** in `zipline-reloaded 3.1.x`. Optimizer-based construction has to be wired via `cvxpy`, `scipy.optimize`, or `riskfolio-lib` inside the rebalance callback.

---

3. Recommendations

Ordered by leverage (impact ÷ effort). Each item lists the *gap*, the *fix*, and the *file(s)* to touch.

3.1 — High leverage / Low effort

R1. Enforce the declared universe filter from strategy YAML

- **Gap**: `classic-momentum.yaml` declares `min_market_cap_billions: 1.0`, `min_price: 5.0`, `exchanges: [NYSE, NASDAQ]`. None of these are applied in `make_pipeline()` — only `AverageDollarVolume.top(500)`. The backtest universe silently diverges from the documented universe.
- **Fix**: Add Pipeline `Filter`s for price (`USEquityPricing.close.latest >= cfg.universe.min_price`), exchange (from `gold.dim_instrument`), and market cap (from a `MarketCap` precomputed feature). `AND` them with the ADV filter into the `universe` mask.
- **Touch**: `momentum_backtest.py::make_pipeline`, possibly a new `MarketCap` column in `GoldEodFeatures`.

Status (post-F-089): Implemented in F-089 / TASK-426. Bundle-side floor reads min(min_market_cap_billions)*1e9 / min(min_last_price) across all strategy YAMLs via `resolve_strategy_universe_floors`; Pipeline mask adds `USEquityPricing.close.latest >= cfg.min_last_price`. Per-strategy MarketCap Pipeline filter and exchange filter are deferred — see ADR §8. Brief: feature-construction-guardrails-and-universe-enforcement-design-brief.md.

R2. Add Zipline trading-control guardrails

- **Gap**: `set_long_only`, `set_max_leverage`, `set_max_position_size`, `set_max_order_count` are unused. A bug in `rebalance()` could silently lever the book or over-concentrate.
- **Fix**: In `initialize()`, derive limits from `StrategyConfig` and call:
  - `set_max_leverage(2.0)` (1.0 long + 1.0 short for long-short, 1.0 for long-only)
  - `set_max_position_size(max_notional=cfg.capital * 0.05)` (5% of NAV per name)
  - `set_max_order_count(2 * (long_count + short_count) + 50)`
  - `set_long_only()` when `cfg.long_only` is True
- **Touch**: `momentum_backtest.py::initialize`; expose limits in `BacktestConfig` and the strategy YAML.

Status (post-F-089): Implemented in F-089 / TASK-427. All four `set_*` guardrails installed inside `make_initialize` via `_derive_guardrail_defaults`. Three new optional YAML knobs under `backtest:` (`max_leverage`, `max_position_size_pct`, `max_order_count`) allow override; current book is well below the derived defaults so the guardrails are passive kill-switches. See ADR §7. Brief:

feature-construction-guardrails-and-universe-enforcement-design-brief.md.

R3. Use `batch_market_order` instead of N individual `order_target_percent` calls

- **Gap**: Each rebalance issues 100+ separate orders. Cheap but noisy in the transaction log and harder to atomically cancel.
- **Fix**: Compute target shares from target weights, build `pd.Series[Asset→int]`, call `batch_market_order` once. Keep `order_target_percent` for the close-out leg if simpler.
- **Touch**: `momentum_backtest.py::rebalance`.

R4. Integrate the FMP promotion allowlist into the backtest universe

- **Gap**: `silver.fmp_promotion_allowlist` excludes ETFs, ADRs, warrants, preferreds — but the Zipline csvdir bundle is built from Gold which may already exclude some, and the backtest does not re-apply the allowlist. Risk of holding non-equity instruments inadvertently.
- **Fix**: At `_emit_tearsheet` / `make_pipeline` time, intersect the asset finder's universe with `silver.fmp_promotion_allowlist WHERE is_allowed = TRUE` resolved at the as-of date. Cleaner option: filter at bundle export.
- **Touch**: `src/sbresearch/zipline/zipline_bundle_export_service.py`, optionally `make_pipeline`.

  Status (post-F-089): Already implemented at TASK-247 (Aug 2025) — all 5 bundle-export SQL strings are rooted in `silver.fmp_promotion_allowlist a WHERE a.is_allowed = TRUE`. F-089 / TASK-430 adds a structural lock-in test at `tests/integration/sbresearch/test_bundle_allowlist_invariant.py` to guard against future refactors that might drop the

JOIN/filter. No new production code. See ADR §8. Brief: feature-construction-guardrails-and-universe-enforcement-design-brief.md.

3.2 — High leverage / Medium effort

R5. Promote portfolio construction to `PortfolioConstructionService`

Note (post-F-088 reconciliation): The `PortfolioConstructionService` already exists at `src/sbportfolio/construction_service.py` and is wired into `_promote_signals` for the Gold phase-3 promotion path (see CLAUDE.md §1). The 5 pure-function weighting algorithms (`equal_weight`, `exponential_weight`, `inverse_volatility`, `mean_variance`, `risk_parity`) live at `src/sbportfolio/algorithms/`. The Zipline backtest did not consume them because the rebalance callback at `momentum_backtest.py:261` contained inline equal-weight arithmetic. F-088 fixes that gap by introducing a thin pure-function adapter at `src/sbportfolio/backtest_adapter.py` (`compute_backtest_weights`) that routes the Zipline rebalance through the existing `sbportfolio.algorithms` registry, and by replacing the `_active_config` module global with closure factories (`make_initialize`, `make_before_trading_start`, `make_rebalance`). No new service is created. See the F-088 design brief and the portfolio-construction ADR for the full record.

- **Gap**: Construction logic is tangled inside a Zipline callback that can't accept closures, forcing the module-level `_active_config` global (`momentum_backtest.py:130`). Logic is not unit-testable in isolation, not reusable for live trading, and not visible to the Gold

phase-3 promotion (`promote_signals`) that CLAUDE.md Section 1 lists as the home of `PortfolioConstructionService`.

- **Fix**: Introduce a pure-function service:

```
class PortfolioConstructionService:
    def target_weights(
        self,
        signal: pd.Series,           # signal value per asset
        long_mask: pd.Series,        # bool per asset
        short_mask: pd.Series,
        features: pd.DataFrame,      # vol, sector, beta, etc.
        config: PortfolioConfig,
    ) -> pd.Series:                  # signed % of NAV per asset
        ...
```

  Zipline's `rebalance()` shrinks to: fetch service output → close exits → `batch_market_order` to targets. Service is unit-testable with no Zipline dependency.

- **Touch**: New `src/sbfoundation/portfolio/portfolio_construction_service.py`; refactor `momentum_backtest.py::rebalance`; update CLAUDE.md §1 reference. Persist outputs to `gold.fact_portfolio_target` per CLAUDE.md's documented phase-3 contract.

R6. Signal-weighted (rank-weighted) sizing

Note (post-F-090 reconciliation): Implemented as YAML mode `exponential_weight` (geometric rank-decay) and the operator alias `rank_weight` → `exponential_weight(decay=0.95)`. Wiring lives in src/sbportfolio/backtest_adapter.py generic dispatch (F-090/TASK-B), routing through the `sbportfolio.algorithms.exponential_weight` registry entry. See the F-090 design brief.

- **Gap**: Equal-weighting treats the #1 signal name and the #50 signal name identically. Throws away signal dispersion.
- **Fix**: Weight by signal *rank* or *z-score* within the long/short bucket: `w_i = z_i / sum(|z|)` scaled to the side's gross budget. Add `weighting: {equal | rank | zscore | inverse_vol}` to the YAML's `backtest` block.
- **Touch**: `PortfolioConstructionService` (post-R5) or `momentum_backtest.py::reba`

R7. Sector caps / sector-neutral construction

> **Note (post-F-090 reconciliation):** Implemented via the **pipeline-side demean** option. YAML modes `sector_neutral_equal_weight` and `sector_neutral_exponential_weight` apply `signal.demean(groupby=Sector(), mask=universe)` in src/sbbacktest/momentum_backtest.py::make_pipeline before `.top()`/`.bottom()`. The `Sector` classifier is backed by a `sectors.csv` sidecar written by `ZiplineBundleExportService` (F-090/TASK-C) from `gold.dim_instrument.sector`. Post-bucket cap is not implemented; deferred to F-091 if needed.

- **Gap**: `_factor_risk_fallback` (`momentum_backtest.py:569`) reports `net_sector_exposures` — the data exists, but construction ignores it. A momentum book can quietly concentrate 60% in tech.
- **Fix**: Two options, pick one:
  - **Pipeline-side**: `signal = signal.demean(groupby=Sector, mask=universe)` makes the signal sector-neutral *before* ranking. Cheapest path, gives near sector-neutral books without an optimizer.
  - **Construction-side**: Hard cap `|sector_weight| 25%` inside `PortfolioConstructionService`; redistribute overflow within the bucket.

- **Touch**: Add a `Sector` classifier reading `gold.dim_instrument.sector`; either `make_pipeline` or the construction service.

R8. Turnover throttling

Note (post-F-090 reconciliation): Implemented as YAML field `backtest.construction.rebalance_buffer` (range `[0.0, 1.0)`) for the **backtest** path. Rank-band semantics: held names stay while their signal rank remains within `round(core_count * (1 + buffer))`. State source (backtest) is `context.portfolio.positions` (closure-local; no sidecar). `buffer=0.0` reduces to F-088/F-089 byte-identical (back-compat invariant verified by 21 unit tests, now living at tests/unit/sbportfolio/test_hysteresis.py).

Live-path wiring (2026-07-10, F-275 / TC-6): the mechanic was relocated from `sbbacktest` into src/sbportfolio/hysteresis.py (public `apply_hysteresis_band`, was private `_apply_rebalance_buffer`) — the correct dependency direction, since `sbbacktest` already imports FROM `sbportfolio`. `sbbacktest/closures.py` now imports the relocated function (behavior- identical relocation). It is also wired into the **live/paper** construction path for the first time: `sbportfolio.settings.HYSTERESIS_BUFFER` (default `0.0`, ships OFF) feeds `PortfolioConstructionService.build()` via the `"buffer"` hyperparam key on every `split_long_short`-based algorithm; `PortfolioConstructionService._resolve_current_holdings` resolves the prior rebalance's holdings by `algorithm` + trailing hyperparam-hash8 (mirroring `sbexecution.target_portfolio_reader.resolve_latest_source_id`, not exact `portfolio_id`, since the middle `CONSTRUCTION_CODE_VERSION` segment can legitimately change between rebalances). Report-then-enforce: this milestone ships the mechanism only — no live buffer value is chosen yet.

- **Gap**: Monthly rebalance refreshes the entire decile every cycle. If signal moves a name from rank 50 → rank 52, we close it and pay round-trip cost for ~zero alpha. Previously true for the backtest only; the live/paper book (the only book that actually trades) had zero throttling until F-275.
- **Fix**: Implement *buffered* decile membership: keep an existing long position while its signal stays in the top `long_count * (1 + buffer)` (e.g. top 60 instead of top 50). Add `rebalance_buffer: 0.2` to the YAML's `backtest` block for the backtest path; flip `sbportfolio.settings.HYSTERESIS_BUFFER` above `0.0` for the live path (after a report-only calibration window).
- **Touch**: `PortfolioConstructionService` (live, F-275) and `momentum_backtest.py::before_trading_start` (backtest, F-090).

3.3 — Medium leverage / Medium effort

R9. Volatility-targeted gross exposure

Status (post-F-091): Adapter-layer scaling implemented in F-091 / TASK-1048. New `target_portfolio_vol` + `max_gross_scale` YAML knobs under `backtest.construction:`; new `compute_backtest_weights_with_scaling_meta(...) -> tuple[dict, ScalingResult]` helper at src/sbportfolio/backtest_adapter.py. After the algorithm returns raw weights, the adapter computes `ex_ante_vol = sqrt(w Σw) * sqrt(252)` from the F-128/F-129 risk-model Σ and scales every weight by `min(target/ex_ante_vol, max_gross_scale)`. Works uniformly across all 7 algorithm modes — operator can vol-target an equal-weight book. `ScalingResult.applied_scale` and `ex_ante_vol` returned for the sidecar's `construction.vol_target` block (TASK-I, pending). Default-OFF: `target_portfolio_vol=None` preserves F-088 / F-089 / F-090 byte-equivalence.

- **Gap**: Gross exposure is fixed at 200% long-short / 100% long-only regardless of regime. A high-vol regime exposes 2× the target portfolio vol.
- **Fix**: Compute ex-ante portfolio vol from `volatility_30d_f` (and ideally a covariance estimate) and scale gross exposure to a target portfolio vol (e.g. 12% annualised). Add `target_portfolio_vol: 0.12` and `max_gross_exposure: 2.0` to the YAML.
- **Touch**: `PortfolioConstructionService`.

R10. Inverse-volatility weighting

Note (post-F-090 reconciliation): Implemented as YAML mode `inverse_volatility` with hyperparam `vol_lookback_days` (default 63). Wiring lives in src/sbportfolio/backtest_adapter.py dispatch routing to `sbportfolio.algorithms.inverse_volatility`. The closure fetches `history_df` via Zipline's `data.history(assets, "close", bar_count=lookback+1, "1d")` inside `make_rebalance` (F-090/TASK-F) only when the base algorithm is `inverse_volatility`. The algorithm computes internally from realised returns (not from `volatility_30d_f` Gold column, so YAML's `vol_lookback_days` knob is honoured exactly).

- **Gap**: Equal-weight gives the same risk budget to a 20%-vol name and a 60%-vol name.
- **Fix**: Within each bucket, `w_i  1 / vol_i` (using `volatility_30d_f` from Gold). Combines naturally with R6 (rank-weighting): `w_i  rank_i / vol_i`. This is the cheapest form of risk-aware construction and a precursor to R11.
- **Touch**: `PortfolioConstructionService`.

R11. Wire `factor_risk` covariance into construction (pre-trade, not post-trade)

- **Gap**: `_compute_factor_risk_block` (`momentum_backtest.py:473`) already computes factor covariance, idiosyncratic variance, marginal risk, and exposure decomposition — but only *after* the trade for reporting. The expensive math is on the floor.
- **Fix**: Hoist the factor-risk computation into a pre-trade `RiskModel` that the construction service consumes. Reuse the existing `fetch_gold_features` / `compute_factor_risk` helpers; call them at rebalance time over the past 252 days for the long+short candidate set. Two consumption modes:
  - *Conservative*: use marginal-risk numbers to scale down concentrated positions.
  - *Aggressive*: feed the covariance matrix to an MV optimizer (R12).
- **Touch**: `src/sbresearch/factor_risk.py` (split fetch from compute), new `PortfolioConstructionService`, the rebalance callback.

R12. Mean-variance / minimum-variance optimizer

Status (post-F-091): Implemented in F-091 / TASK-1046 + TASK-1047. `mean_variance` promoted from rejected to allowed in `ALLOWED_ALGORITHMS`; new `min_variance` algorithm added. Both ship with a `mv_solver: Literal["closed_form", "cvxpy"]` knob — cvxpy branch solves `argmin ·w'(Σ+ridge·I)w - w` (mean-var) or `argmin w'(Σ+ridge·I)w` (min-var) subject to the baseline trio (gross-exposure equality, per-position bounds, long-only when `cfg.long_only=True`). Covariance source preference: `risk_model_covariance` from F-128/F-129 wins; falls back to `shrink_covariance(prepare_returns_matrix(history_df))`.

Non-OPTIMAL solver status → `qp_solver_status_fallback` chain (default `inverse_volatility`). `cvxpy ^1.5` added to `pyproject.toml` — default chain ships Clarabel + SCS + OSQP + HiGHS (BSD/MIT). Algorithm-level default solver is `closed_form` for back-compat; `ConstructionConfig.mv_solver` YAML default is `cvxpy` (adapter threads it through). Sector caps + turnover constraints deferred to follow-on.

- **Gap**: Once R11 lands, the natural construction step is `argmax w'·alpha - ·w'·Σ·w` subject to budget, leverage, position, and sector caps.
- **Fix**: Add `cvxpy` (preferred — clean dependency, used widely in donor `qsresearch.strategies.factor.portfolio_construction`) inside `PortfolioConstructionService`. Make the choice configurable: `construction: {equal | rank | inverse_vol | min_var | mean_var}` in the YAML.
- **Touch**: Add `cvxpy` to `pyproject.toml`, implement in `PortfolioConstructionService`. Consult `C:\qs\QSResearch\packages\qsresearch\qsresearch\strategies\factor\portfolio_cons` (CLAUDE.md §7 Code Donor) as the reference implementation.

3.4 — Lower leverage / Low effort

R13. Stop computing `signal.bottom(short_count)` when `long_only=True`

- **Gap**: `make_pipeline()` always computes both `longs` and `shorts` columns even when `long_only=True` (`momentum_backtest.py:213-214`). The short list is then discarded in `before_trading_start` (`momentum_backtest.py:259`). Wasted Pipeline work.
- **Fix**: Branch in `make_pipeline()` on `config.long_only`; only emit the `shorts` column when long-short.
- **Touch**: `momentum_backtest.py::make_pipeline`.

Status (post-F-089): Implemented in F-089 / TASK-428. `make_pipeline` emits `shorts` only when `not cfg.long_only`; `make_before_trading_start` reads defensively via `"shorts" in pipe.columns`. Pure dead-code removal — F-089 baseline unchanged. Brief: feature-construction-guardrails-and-universe-enforcement-design-brief.md.

R14. Make rebalance day-of-month configurable

- **Gap**: `_rebalance_rule` hard-codes `month_start(days_offset=0)` and `week_start(days_offset=0)`. Some momentum studies prefer mid-month or skip-the-first-day to avoid the "month-end rebalance" crowding effect.
- **Fix**: Add `rebalance_day_offset: 0` to the YAML and thread it through.
- **Touch**: `momentum_backtest.py::_rebalance_rule`, `BacktestConfig`, strategy YAML schema.

Status (post-F-089): Implemented in F-089 / TASK-429. `_rebalance_rule(cadence, day_offset)` accepts a configurable offset with range validation (0..27 monthly, 0..6 weekly). Both YAMLs carry explicit `rebalance_day_offset: 0` for the legacy "first day of period" cadence. Brief: feature-construction-guardrails-and-universe-enforcement-design-brief.md.

R15. Per-position circuit-breaker / drawdown gate

Status (post-F-091): Implemented in F-091 / TASK-1050. New pure module src/sbbacktest/circuit_breaker.py mirroring F-090's `rebalance_buffer.py` pattern. Exposes `_apply_circuit_breakers(...) -> CircuitBreakerResult` with two independent gates: (a) per-position -stop fires

on any LONG position whose trailing N-day return (N from `position_stop_lookback`, default 5) is below `-position_stop_sigma ·` — short positions excluded because a price drop helps a short leg; (b) portfolio drawdown gate scales gross exposure by `drawdown_cut_ratio` for the next rebalance when `(current_nav - peak_nav) / peak_nav < max_drawdown_max` (the existing `promotion_gates.max_drawdown_max` YAML knob). Both gates default-OFF when `position_stop_sigma=None` and `drawdown_cut_ratio=None`. Result-stash on `context` for the sidecar `construction.circuit_breaker` block (TASK-I, pending). Wired into `make_before_trading_start` BEFORE the F-090 rebalance_buffer call (TASK-H, pending).

- **Gap**: No per-name stop-loss or portfolio-level drawdown gate. A momentum book can ride a single name to -80%.
- **Fix**: In the construction service, zero-out any position whose trailing 5-day return falls below `-3·` (configurable). At portfolio level, cut gross exposure in half if 30-day drawdown exceeds `max_drawdown_max` from the strategy YAML's `promotion_gates`.
- **Touch**: `PortfolioConstructionService`.

R16. Surface construction config in the JSON sidecar

- **Gap**: The tearsheet JSON sidecar (`_emit_tearsheet`) records metrics but not the construction choices (weighting scheme, sector caps, vol target). Two backtests with different weighting are visually indistinguishable in the run report.
- **Fix**: Add a `construction` block to the JSON written at `momentum_backtest.py:656-673` listing the resolved construction config. Render in the HTML.
- **Touch**: `momentum_backtest.py::_emit_tearsheet`, `src/sbreporting`.

Status (post-F-089): Implemented in F-089 / TASK-431. Sidecar `schema_version` bumped 4 → 5; new `construction`

block with 9 fields (algorithm, long_only, long_count, short_count, long_gross, short_gross, guardrails.{max_leverage, max_position_size_pct, max_order_count}, rebalance.{cadence, day_offset}) surfaces the resolved post-R2 / post-R14 config alongside the F-092 raw-YAML mirror. `momentum.html` renders a "Construction (resolved)" card immediately after the Strategy Configuration section. Brief: feature-construction-guardrails-and-universe-enforcement-design-brief.md.

3.5 — Documentation / Governance

R17. Write an ADR for the construction choices

- **Gap**: There is no decision record under `backlog/docs/decisions/` describing why we chose equal-weight 50/50 decile vs. rank-weighted, sector-neutral, or MV-optimized. Future contributors won't know whether the current state is a deliberate baseline or technical debt.
- **Fix**: Add `backlog/docs/decisions/portfolio-construction.md` per CLAUDE.md §7. Capture the construction taxonomy (R5–R12 above), the current baseline rationale, and the migration path.
- **Touch**: `backlog/docs/decisions/portfolio-construction.md` (new).

---

4. Proposed Milestone Ordering

A reasonable sequencing into Backlog.md milestones (Backlog.md MCP per CLAUDE.md §7):

1. **F-NNN — PortfolioConstructionService (R5)** — extract the construction step into a pure-function service, no behaviour change. Tests pin the current equal-weight outputs as the regression baseline.
2. **F-NNN — Construction guardrails & universe enforcement (R1, R2, R4, R13, R14, R16)** — small declarative fixes on top of the new service. Tightens the existing equal-weight construction without changing the weighting scheme.
3. **F-NNN — Risk-aware weighting (R6, R7, R8, R10)** — rank, sector-neutral, turnover-buffered, inverse-vol. Each switchable from the YAML so we can A/B vs. baseline.
4. **O-NNN — Pre-trade risk model (R11)** — refactor `factor_risk.py` to expose a pre-trade `RiskModel` instead of only a post-trade report block.
5. **F-NNN — Optimizer-based construction (R9, R12, R15)** — adds `cvxpy`; introduces `min_var` / `mean_var` modes with vol targeting and drawdown gating.
6. **R17 — ADR** authored alongside F1; updated as F3–F5 land.

Each milestone keeps the equal-weight baseline runnable so we can compare risk-aware variants against it under identical universe and frictions.

Universe Construction API

This chapter is generated from the canonical platform doc, which lives in `docs/` *(single source of truth).*

Universe Construction API

Full spec for `UniverseService`. Referenced from `CLAUDE.md §16`.

`UniverseService` (`src/sbuniverse/services/universe_service.py`) provides three point-in-time universe methods. All filter `gold.dim_instrument` lifecycle columns (`ipo_date`, `delisting_date`) so only instruments active on `as_of_date` are returned.

Methods

Method	Description	Reliable From
`get_index_tickers(index_code, as_of_date)`	Exact historical index membership via `gold.fact_index_constituent`	SP500: Oct 2008; NASDAQ100: Jan 1995; DJIA: Jan 1994
`get_broad_universe_tickers(as_of_date)`	All `fmp_promotion_allowlist`-eligible instruments alive on date	~Jan 1995 (~1,546 symbols); grows to ~6,885 by 2024
`get_filtered_tickers(exchanges, sectors, industries, countries, as_of_date)`	Dimension-filtered universe via three-tier fallback	Same as broad; market cap filter uses current snapshot only

---

Three-Tier Fallback (get_filtered_tickers)

1. **Tier 1** — `silver.fmp_market_screener` (preferred; not yet populated)
2. **Tier 2** — `silver.fmp_company_profile_bulk` (available; has exchange, sector, industry, country, market_cap snapshot)
3. **Tier 3** — `gold.dim_instrument JOIN silver.fmp_promotion_allowlist` (always available; no dimension filters, no market cap)

Package Boundary

- `sbuniverse.infra.universe_repo.UniverseRepo` — canonical implementation of all query methods
- `sbuniverse.infra.universe_repo.UniverseRepo` — equivalent standalone implementation for use within sbfoundation (avoids circular import via `sbfoundation/__init__.py`)
- `sbuniverse.services.universe_service.UniverseService` — adds run-context utilities (`run_id`, `now`, `today`, `from_date`, `last_quarter_end`, `next_market_day`) on top of the repo

---

Research Universe (factor-research scope)

The **factor-research** universe is separate from the strategy/ingestion universes above. It is the cross-section every factor-research SQL surface (`sbic`, `sbalphalens`, `sbpermtest`, `sbdiag`, `sbfactorcontrib`) measures factors over, injected as a shared membership CTE rather than a hard-coded allowlist join.

Production universe (F-148 / RUS-1)

`sbuniverse.research_universe_membership_cte(config)` emits a CTE `(instrument_sk, date_sk)` for the single production research universe, driven by `config/universes/research.yaml` via `sbcontracts.research_universe_config.ResearchUniverseConfigService`:

- `mode: allowlist_only` (default, current production) — allowlist lifecycle US-market-day.
- `mode: base` — additionally a point-in-time price floor + trailing dollar-ADV floor (the tradeable **Base** universe).

Variants (F-168 / RUS-2)

`sbuniverse.research_universe_variant_cte(base_config, variant)` emits the membership CTE chain for a **universe variant** — `Base` `axis-filter` — for the deferred calibration sweep (RUS-3). The 9 variants are declared in `config/universes/research-variants.yaml` and loaded by `sbcontracts.research_universe_variant.ResearchUniverseVariantService`:

Variant(s)	Axis	Filter (Base)
`base`	base	none (Base only)

Variant(s)	Axis	Filter (Base)
`cap_small` / `cap_mid` / `cap_large`	cap	market-cap tier vs per-date percentile cutoffs
`liq_top20` / `liq_top50`	liquidity	top fraction by per-date dollar-ADV percentile rank
`vol_high` / `vol_low`	volatility	above/below per-date median trailing-vol
`sector_neutral`	sector_neutral	Base membership; **scoring transform** (see below)

Notes:

- **Base inheritance** — every membership variant forces `mode: base` (`dataclasses.replace`), so the price/ADV floors always apply regardless of the production `research.yaml` mode. Base floors are inherited from `research.yaml`, not duplicated.
- **Cap breakpoints** — config-selectable `breakpoint_method` (`nyse` default | `cross_sectional`): NYSE-listed (`gold.dim_exchange.exchange_code`) per-date percentiles applied to the full cross-section, over a config-selectable `market_cap_source` (`computed_pit` default = `close × ASOF carry-forward`gold.fact_quarter.weighted_average_shs_out_diluted;profile_s `implemented;`annual_market_cap_f‘ deferred).
- **Sector-neutral** is **not** a membership filter — it z-scores the factor within `(date_sk, sector_sk)` before the cross-sectional IC, carried as the `ResearchUniverseVariant.sector_neutralize` flag and applied by `sbic.services._ic_sql.filtered_factor_cte` / threaded through `ICService.factor_ic_ir(..., variant=...)`.

RUS-2 ships **definitions + the scoring flag only** — the sweep runner, ranking, sidecar, and SPA are RUS-3. See `docs/backlog/feature-research-universe-var`

Known Gaps

- **Historical market cap** — `market_cap` in Tier 1/2 reflects today's snapshot, not the `as_of_date` value. Point-in-time market cap requires `fact_eod` backfill (see `universe_screener_features_exec_plan.md`).
- **Liquidity filters** — ADV, ADTV unavailable until `fact_eod` is backfilled.
- **SP500 pre-2008** — constituent change log is incomplete before Oct 2008.
- **NULL `ipo_date`** — ~162 allowlist symbols have no `ipo_date`; treated as always-eligible (conservative).

Dataset Reference

This chapter is generated from the canonical platform doc, which lives in `docs/` *(single source of truth).*

SBFoundation Domain & Dataset Reference

Last Updated: 2026-07-27 (F-317 — added 9 FRED macro series; 25 → 34 datasets)

This document is the comprehensive reference for all data domains and datasets in SBFoundation. For API usage and pipeline operations, see `docs/api-usage.md`. For dataset configuration details, see `config/dataset_keymap.yaml`.

Domain Overview

SBFoundation ingests data across **4 domains** containing **34 datasets** from 3 sources (FMP, FRED, FINRA).

Domain	Datasets	Scope	Description	API Entry Point
`eod`	3	Mixed	End-of-day prices and company profiles	`api.run_eod()`

Domain	Datasets	Scope	Description	API Entry Point
`quarter`	4	Global	Quarterly financial statements and key metrics	`api.run_quarter()`
`annual`	5	Global	Annual financial statements, key metrics, and ratios	`api.run_annual()`
`eow`	22	Mixed	End-of-week: index constituents, macro indicators, stock splits, analyst estimates, earnings surprises, short interest	`api.run_eow()`

Execution order: `eod` → `quarter` → `annual` → `eow`

Data sources: - **FMP** (Financial Modeling Prep) — 22 datasets; requires `FMP_API_KEY` - **FRED** (Federal Reserve Economic Data) — 11 datasets; requires `FRED_API_KEY` - **FINRA** (Financial Industry Regulatory Authority) — 1 dataset; no API key required

EOD Domain (3 datasets)

End-of-day market data and company profiles. This is the primary daily-refresh domain.

Market Data

Dataset	Scope	Silver Table	Key Columns	Refresh	API Path
`eod-bulk-price`	global	`fmp_eod_bulk_price`	`symbol, date`	Daily (1)	`eod-bulk`
`eod-price-history`	per_ticker	`fmp_eod_bulk_price`	`symbol, date`	Daily (0)	`historical-price-eod/dividend-adjusted`

- **eod-bulk-price**: Daily bulk CSV of OHLCV + adjusted close for all instruments. Primary source for `gold.fact_eod`. At Gold promotion, OHLC columns are adjusted by the `adj_close / close` ratio so all prices are split- and dividend-adjusted.
- **eod-price-history**: Per-ticker historical price backfill (deep-history `backfill-eod-chunked` + delisted survivorship backfill). Writes to the same Silver table as `eod-bulk-price`. **B-147.6 (F-147):** endpoint corrected from `historical-price-eod/full` (which returns no `adjClose` → NULL `adj_close` for ~28% of `gold.fact_eod`) to `historical-price-eod/dividend-adjusted` (returns `adjOpen/adjHigh/adjLow/adjClose`), mapped via the dedicated `EodDividendAdjustedPriceDTO` — `adjClose` populates both `close` and `adj_close`, so the rows are already-adjusted and identical to the bulk path through Gold's back-adjustment math. Not in the nightly EOD path (excluded by `EodBronzeIngester`); driven only by the backfill/delisted commands.
- **Fields**: `symbol`, `date`, `open`, `high`, `low`, `close`, `adj_close`, `volume`, `data_quality_flag` (F-086 — DTO-time severity-wins

flag: `non_positive_close` (error) / `non_positive_volume` (warn) / `ohlcv_corrected` (info))

- **Symbol filter**: `symbol_filter_col: symbol` — only allowlisted instruments promoted to Silver
- **Quality thresholds (F-086 / TASK-313)**: `eod-bulk-price` carries an optional `quality_thresholds` block consumed by `PriceDataQualityService` during the factor phase. Defaults: `flag_high_price_z=5.0`, `flag_high_volume_z=5.0`, `max_gap_business_days=5`. Override per-key in `config/dataset_keymap.yaml`; missing keys fall back to the service defaults.
- **Documentation**: FMP EOD Bulk | FMP Historical Price

Company Metadata

Dataset	Scope	Silver Table	Key Columns	Refresh	API Path
`company-profile-bulk`	global	`fmp_company_profile_bulk`	`symbol`	Daily (1)	`profile-bulk`

- **company-profile-bulk**: Snapshot of all company profiles (name, exchange, sector, industry, country, market cap, IPO date, etc.). Paginated via `part` parameter. Feeds `gold.dim_instrument`, `gold.dim_company`, and the promotion allowlist.
- **Open operator follow-up (TASK-2677)**: on at least one observed run, FMP returned `"Invalid or missing query parameter"` for this recipe at `part>=4` (an otherwise well-formed, unchanged-shape paginated request), and pages 1-3 also showed minor row-count shortfalls vs. the expected ~22,616 rows/page. `BronzeService._process_paginated_recipe` now retries this specific error (bounded, same backoff as the existing 429/502/503/504 classes) and, if the retry budget is exhausted, records it as a genuine Bronze failure (logged at ERROR with the raw FMP body, and surfaced on the `ingest_bronze` sidecar's per-domain `failed_pages`

alert) instead of the pre-fix behavior of silently treating it as end-of-pagination — but the root cause on FMP's side (transient bulk-endpoint glitch vs. a durable `part`-parameter validation bug beyond some threshold) has not been confirmed with FMP support. An automated agent cannot open a support ticket; a human operator should verify the `part` parameter's documented behavior for `profile-bulk` against FMP's current API docs/support (see `help_url` above) and confirm whether the page>=4 failure recurs across the next several nightly runs (also tracked as Tier-4-pending on TASK-2677).

- **`is_etf` allowlist semantics (F-247 / EU-1)**: the bulk feed carries `is_etf` per row; `InvestableUniverseService.rebuild()` writes it onto every `silver.fmp_promotion_allowlist` row (allowed *and* rejected). With `sbuniverse.settings.ALLOW_ETFS=False` (the default, env `SB_ALLOW_ETFS`) ETFs reject with `rejection_reason='ETF_FLAG'`, byte-identical to today; with `ALLOW_ETFS=True` the ETF arm is skipped so ETFs land `is_allowed=TRUE` and flow Silver→Gold (the funds / mutual-fund 5X / warrant / preferred / ADR / rights arms are untouched). Downstream surfaces partition by `is_etf` — the equity research `_SPINE` + `StrategyUniverseService` + `UniverseRepo.get_filtered_tickers` re-exclude ETFs so the equity factor cross-section stays equity-only (TASK-1903), and the F-247 / TASK-1904 `EtfFactorIcService` scores the 7 price factors against `instrument_class="etf"` only.
- **Documentation**: FMP Profile Bulk

Quarter Domain (4 datasets)

Quarterly financial statements and key metrics ingested as global bulk CSVs. One file per quarter covers all allowlisted instruments.

Dataset	Silver Table	Key Columns	Refresh	API Path
`income-statement-bulk`	`fmp_bulk_income_statement`	`symbol, period, calendar_year`	Daily (1)	`income-statement-bulk?peri`
`balance-sheet-statement-bulk`	`fmp_bulk_balance_sheet`	`symbol, period, calendar_year`	Daily (1)	`balance-sheet-statement-bu`
`cashflow-bulk`	`fmp_bulk_cash_flow`	`symbol, period, calendar_year`	Daily (1)	`cash-flow-statement-bulk?p`
`key-metrics-bulk`	`fmp_bulk_key_metrics`	`symbol, period, calendar_year`	Quarterly (90)	`key-metrics-bulk?period=qu`

- **Gold table**: `gold.fact_quarter` — one row per (instrument_sk, period_date_sk, period). Built from income + balance sheet + cash-flow via optional LEFT JOINs.
- **Key fields — income**: `revenue`, `gross_profit`, `operating_income`, `net_income`, `ebitda`, `eps`, `eps_diluted`, `interest_expense`, `depreciation_and_amortization`, `weighted_average_shs_out`, `weighted_average_shs_out_diluted`
- **Key fields — balance sheet**: `total_assets`, `total_liabilities`, `total_stockholders_equity`, `cash_and_cash_equivalents`, `long_term_debt`, `net_debt`, `goodwill_and_intangible_assets`, `retained_earnings`
- **Key fields — cashflow**: `operating_cash_flow`, `capital_expenditure`, `free_cash_flow`, `dividends_paid`, `common_stock_repurchased`
- **Key fields — metrics**: `roic`, `invested_capital`, `capex_to_ocf`, `ev_to_ebitda`, `days_sales_outstanding`, `days_payables_outstanding`, `days_inventory`
- **Symbol filter**: All use `symbol_filter_col: symbol`
- **Row-level quality flag (F-086 / TASK-307+TASK-310)**:

`fmp_balance_sheet_bulk_quarter` and `fmp_income_statement_bulk_quarter` each gain a nullable `data_quality_flag VARCHAR` column. DTOs stamp `balance_sheet_identity_violation` (severity warn) when `|total_assets - (total_liabilities + total_stockholders_equity)| / |total_assets| > 0.05`, and `net_income_exceeds_revenue` (severity warn) when `revenue > 0 AND net_income > revenue`. Each fired flag also lands a row in `ops.silver_anomaly`.

- **Season gate**: `QuarterService` skips runs outside earnings windows unless `year`+`period` are provided explicitly.
- **Documentation**: FMP Financial Statements Bulk | FMP Key Metrics Bulk

---

Annual Domain (5 datasets)

Annual (FY) financial statements, key metrics, and ratios. Same bulk CSV pattern as quarterly.

Dataset	Silver Table	Key Columns	Refresh	API Path
`income-statement-bulk`	`fmp_income_statement_bulk_annual`	`symbol, calendar_year`	Monday (1)	`income-statement-bulk?period=FY`
`balance-sheet-statement-bulk`	`fmp_balance_sheet_bulk_annual`	`symbol, calendar_year`	Tuesday (1)	`balance-sheet-statement-bulk?period=FY`
`cashflow-bulk`	`fmp_cash_flow_bulk_annual`	`symbol, calendar_year`	Wednesday (1)	`cash-flow-statement-bulk?period=FY`
`key-metrics-bulk`	`fmp_key_metrics_bulk_annual`	`symbol, calendar_year`	Thursday (365)	`key-metrics-bulk?period=FY`
`ratios-bulk`	`fmp_ratios_bulk_annual`	`symbol, calendar_year`	Friday (365)	`ratios-bulk?period=FY`

- **Gold table**: `gold.fact_annual` — one row per (instrument_sk, period_date_sk). Merged from income + balance sheet + cashflow + key metrics + ratios via optional LEFT JOINs.
- **Key fields — income**: `revenue`, `gross_profit`, `operating_income`, `net_income`, `ebitda`, `eps`, `eps_diluted`, `interest_expense`, `depreciation_and_amortization`, `weighted_average_shs_out`, `weighted_average_shs_out_diluted`
- **Key fields — balance sheet**: `total_assets`, `total_liabilities`, `total_stockholders_equity`, `cash_and_cash_equivalents`, `long_term_debt`, `net_debt`, `goodwill_and_intangible_assets`, `retained_earnings`
- **Key fields — cashflow**: `operating_cash_flow`, `capital_expenditure`, `free_cash_flow`, `dividends_paid`, `common_stock_repurchased`
- **Key fields — ratios**: `gross_profit_margin`, `operating_profit_margin`, `net_profit_margin`, `effective_tax_rate`, `debt_ratio`, `interest_coverage`
- **Symbol filter**: All use `symbol_filter_col: symbol`
- **Row-level quality flag (F-086 / TASK-307+TASK-310)**: `fmp_balance_sheet_bulk_annual` and `fmp_income_statement_bulk_annual` each gain a nullable `data_quality_flag VARCHAR` column. Same detectors as the quarter variants — `balance_sheet_identity_violation` (warn) and `net_income_exceeds_revenue` (warn) — with each fired flag also landing a row in `ops.silver_anomaly`.
- **Season gate**: `AnnualService` skips runs outside Jan–Mar unless `year` is provided explicitly.
- **Documentation**: FMP Financial Statements Bulk | FMP Key Metrics Bulk | FMP Ratios Bulk

---

EOW Domain (22 datasets)

End-of-week data: index constituent history, macro indicators, company delisted list, mergers & acquisitions, stock splits, analyst estimates, earnings surprises, and FINRA equity short interest. Ingested weekly (Saturday) or per-ticker. Bronze-only task: `ingest_bronze_eow_task`; Silver promotion: `promote_silver_eow_task`.

Index Constituent History

Dataset	Silver Table	Key Columns	Refresh	API Path
`sp500-constituent`	`fmp_sp500_constituent_history`	`symbol`, `date_of_change`	Weekly (7, Sat)	`historical-sp500-constituent`
`nasdaq100-constituent`	`fmp_nasdaq100_constituent_history`	`symbol`, `date_of_change`	Weekly (7, Sat)	`historical-nasdaq-constituent`
`djia-constituent`	`fmp_djia_constituent_history`	`symbol`, `date_of_change`	Weekly (7, Sat)	`historical-dowjones-constituent`

- **Gold table**: `gold.fact_index_constituent` — one row per (index_sk, instrument_sk, effective_from). `effective_to = NULL` means still a member. Built by `GoldIndexService` by replaying Silver change-log tables.
- **Fields**: `symbol`, `date_of_change`, `added_security`, `removed_ticker`, `removed_security`, `reason`
- **No symbol filter** — index constituent data includes removed tickers by design.
- **Reliable from**: SP500 Oct 2008; NASDAQ100 Jan 1995; DJIA Jan 1994.
- **Documentation**: FMP SP500 | FMP NASDAQ | FMP DJIA

Company Metadata (EOW)

Dataset	Scope	Silver Table	Key Columns	Refresh	API Path
`company-delisted`	global	`fmp_company_delisted`	`symbol`	Weekly (7, Sat)	`delisted-companies`
`mergers-acquisitions`	global	`fmp_mergers_acquisitions`	`symbol`, `targeted_symbol`, `transaction_date`	Weekly (7, Sat)	`v4/mergers-acquisitions-`

- **company-delisted**: List of delisted companies with IPO and delisting dates. Feeds `gold.dim_instrument` lifecycle columns (`delisting_date`). No symbol filter — this dataset *defines* the allowlist boundary.
- **mergers-acquisitions**: FMP M&A RSS feed capturing acquirer (`symbol`, `company_name`, `cik`) and acquired target (`targeted_symbol`, `targeted_company_name`, `targeted_cik`) plus `transaction_date`, `acceptance_time`, `url`. The delisting-reason source (F-312): `targeted_symbol` is the join key linking a delisted instrument to its M&A event. Silver-only; no Gold promotion yet.
- **Documentation**: FMP Delisted Companies | FMP Mergers & Acquisitions

Stock Splits

Dataset	Scope	Silver Table	Key Columns	Refresh	API Path
`stock-split`	per_ticker	`fmp_stock_splits`	`symbol`, `date`	Monthly (30, Sat)	`splits`

- **Purpose**: Historical stock split events (numerator, denominator, split type). Feeds `gold.fact_stock_split`.
- **Fields**: `symbol`, `date`, `numerator`, `denominator`, `split_type`
- **Documentation**: FMP Stock Splits

Macro / Economics Indicators

Dataset	Source	Silver Table	Key Columns	Refresh	API Path
`fred-dgs10`	fred	`fred_dgs10`	`date`	Saturday (1)	`series/observations`
`fred-usrecm`	fred	`fred_usrecm`	`date`	Saturday (28)	`series/observations`
`fred-dgs1mo`	fred	`fred_dgs1mo`	`date`	Saturday (1)	`series/observations`
`fred-dgs3mo`	fred	`fred_dgs3mo`	`date`	Saturday (1)	`series/observations`
`fred-dgs1`	fred	`fred_dgs1`	`date`	Saturday (1)	`series/observations`
`fred-dgs2`	fred	`fred_dgs2`	`date`	Saturday (1)	`series/observations`
`fred-dgs5`	fred	`fred_dgs5`	`date`	Saturday (1)	`series/observations`
`fred-dgs30`	fred	`fred_dgs30`	`date`	Saturday (1)	`series/observations`
`fred-t10y2y`	fred	`fred_t10y2y`	`date`	Saturday (1)	`series/observations`
`fred-baa10y`	fred	`fred_baa10y`	`date`	Saturday (1)	`series/observations`
`fred-vixcls`	fred	`fred_vixcls`	`date`	Saturday (1)	`series/observations`

Dataset	Source	Silver Table	Key Columns	Refresh	API Path
`market-risk-premium`	`fmp`	`fmp_market_risk_premium`	`country`	Saturday (30)	`market-risk-premium`

- **fred-dgs10**: 10-Year Treasury Constant Maturity Rate (risk-free rate proxy for CAPM/DCF). Series starts 1962-01-01. Used by `EowHistoryService` as the representative EOW freshness cursor.
- **fred-usrecm**: NBER U.S. Recession Indicator (1 = recession, 0 = expansion). Series starts 1967-01-01.
- **F-317 FRED macro-series expansion (9 datasets)**: Silver-only macro time-series (each `series/observations`, Saturday cadence, `min_age_days: 1`), sharing one generic `FredObservationDTO` over `{date, value}`. **Data-only** — no macro-style factor is derived from them yet (deferred to Wave 2).
  - **fred-dgs1mo** / **fred-dgs3mo**: 1-Month / 3-Month Treasury Constant Maturity Rate — the short-end of the curve, the correct excess-return risk-free tenor (the 10Y `fred-dgs10` stays the cost-of-equity / WACC proxy).
  - **fred-dgs1** / **fred-dgs2** / **fred-dgs5** / **fred-dgs30**: 1/2/5/30-Year Treasury Constant Maturity Rate — the Treasury constant-maturity yield curve.
  - **fred-t10y2y**: 10-Year minus 2-Year Treasury term spread (yield-curve / regime signal).
  - **fred-baa10y**: Moody's Baa corporate yield minus 10-Year Treasury credit spread (credit-regime signal).
  - **fred-vixcls**: CBOE Volatility Index (VIX) close (volatility-regime signal).
- **market-risk-premium**: Equity risk premium by country (total equity risk premium, country risk premium).
- **Gold**: All FRED + market-risk-premium datasets are Silver-only — no `instrument_sk` FK. Consumed directly by the feature engine.

- **Requires**: `FRED_API_KEY` for FRED datasets.
- **Documentation**: FRED DGS10 | FRED USRECM | FRED DGS1MO | FRED DGS3MO | FRED DGS1 | FRED DGS2 | FRED DGS5 | FRED DGS30 | FRED T10Y2Y | FRED BAA10Y | FRED VIXCLS | FMP Market Risk Premium

Analyst Estimates & Earnings Surprises

Dataset	Discriminator	Scope	Silver Table	Key Columns	Refresh	API Path
`analyst-estimates`	`annual`	per_ticker	`fmp_analyst_estimates`	`ticker`, `fiscal_year`, `fiscal_quarter`, `as_of_date`	Weekly (7, Sat)	`analyst-estimates?symbol=__ticker__&pe`
`analyst-estimates`	`quarter`	per_ticker	`fmp_analyst_estimates`	`ticker`, `fiscal_year`, `fiscal_quarter`, `as_of_date`	Weekly (7, Sat)	`analyst-estimates?symbol=__ticker__&pe`
`earnings-surprises`	—	per_ticker	`fmp_earnings_surprises`	`ticker`, `date`	Quarterly (90, Sat)	`earnings?symbol=__ticker__`

- **analyst-estimates**: Forward-looking consensus estimates (EPS, revenue, EBITDA, net income) with high/low ranges and analyst count. Two recipes share one Silver table: `discriminator=annual` writes fiscal-year rows (`fiscal_quarter=0`); `discriminator=quarter` writes quarterly rows (`fiscal_quarter=1..4`). `fiscal_year`/`fiscal_quarter` are derived in `AnalystEstimatesDTO.transform_df_content` from the period-end `date` using a row-density heuristic (annual if rows/unique_years < 2). **2026-04-14**: migrated from the retired v3 endpoint to `/stable/analyst-estimates`; upstream field names flattened (`epsAvg`, `revenueAvg`, `ebitdaAvg`,

`numAnalystsEps`) and mapped back to unchanged Silver column names via `api:` keys in `dto_schema`. **TASK-3001**: `as_of_date` (synthesized by `SilverService` from the Bronze manifest coverage date — `row_date_col` stays unset) was appended to the natural key of BOTH recipe blocks, so the weekly Saturday re-pull persists a **weekly consensus time-series** (one row per snapshot date) instead of UPSERT-overwriting a single row per `(ticker, fiscal_year, fiscal_quarter)`. `AnalystEstimatesDTO.KEY_COLS` is kept in lockstep.

- **earnings-surprises**: Historical actual-vs-estimated EPS per reporting date. Feeds `gold.fact_earnings_surprises` and is aggregated into beat-rate statistics for `gold.fact_eps_forecast`. **2026-04-14**: migrated from retired v3 `/api/v3/earnings-surprises` to `/stable/earnings`; fields `epsActual` / `epsEstimated` map to Silver `actual_earning_result` / `estimated_earning`.
- **Key fields — analyst-estimates**: `estimated_eps_avg`, `estimated_eps_high`, `estimated_eps_low`, `estimated_revenue_avg`, `estimated_revenue_high`, `estimated_revenue_low`, `estimated_ebitda_avg`, `estimated_net_income_avg`, `number_analyst_estimated`
- **Key fields — earnings-surprises**: `actual_earning_result`, `estimated_earning`
- **Symbol filter**: Both use `symbol_filter_col: symbol`
- **Gold**: Drives three Gold fact tables — `fact_analyst_estimates`, `fact_earnings_surprises`, `fact_eps_forecast` (the last built by `EpsForecastService`).
- **Documentation**: FMP /stable/ Analyst Estimates | FMP /stable/ Earnings

Equity Short Interest

Dataset	Source	Scope	Silver Table	Key Columns	Refresh
`equity-short-interest`	`finra`	global	`finra_equity_short_interest`	`symbol`, `settlement_date`	Weekly (7 Sat)

- **equity-short-interest**: FINRA bi-monthly equity short interest regulatory files. Free, auth-free CSV files fetched by `FinraFileAdapter` from settlement-date-templated URLs. Feeds short interest analytics.
- **Fields**: `symbol`, `settlement_date`, plus FINRA short interest columns.
- **Symbol filter**: `symbol_filter_col: symbol`
- **No API key required** — FINRA publishes auth-free regulatory data.
- **Gold**: `gold.fact_short_interest` (F-070 / TASK-061) — built by `GoldFactService._build_fact_short_interest()` after `_build_fact_annual()` in the Gold core phase. Three `_f` features: `short_interest_delta_f`, `short_interest_change_pct_f`, and `short_pct_float_f` (the last uses `fact_annual.weighted_average_shs_out_diluted` as a proxy for true float via a point-in-time LEFT JOIN; resolves to NULL when no annual report is yet available).
- **Factor layer (F-299)**: `gold.fact_short_interest` now also feeds the factor pool — the platform's first `positioning`-style factors, `days_to_cover` (a new `days_to_cover_f` column) and `short_pct_float` (reuses the existing `short_pct_float_f` column above), both `direction: negative`, `experimental`. `FactorValueLiftService` snaps each factor row's knowledge `date_sk` forward to the first US market day on/after `settlement_date + 18 calendar days` (FINRA's ~16-18 day dissemination lag) via a new `TABLE_DATE_SK_EXPR["fact_short_interest"]` entry — no dataset/schema/cadence change here, consumer-side only.
- **Ingestion cadence / n_obs accrual-rate context (TASK-**

2678): the keymap recipe's `data_source_path` (`equity/otcmarket/biweekly/shrt{s` and `run_days: [sat]` confirm FINRA settles this file **bi-weekly** — twice a month, one file per settlement date — not daily. `days_to_cover` / `short_pct_float` are nonetheless evaluated by `sbdiag.FactorDiagnosticsService` against the **daily**-cadence `min_n_obs=60` bar (`fact_short_interest` is absent from `sbdiag.settings.SOURCE_TABLE_CADENCE`, so `_resolve_cadence` falls back to `"daily"`). At ~2 new settlement dates per month, organic forward accrual alone would take roughly **2.5 years** (60 obs ÷ ~2 obs/month) to clear that bar — the 17 observed dates behind the original TASK-2533/2534/2535 finding came from a historical backfill, not from nightly accrual, and further backfill (not the passage of nightly runs) is what will actually move these two factors past `min_n_obs`. This is a genuine cadence mismatch in the same family as the F-292/LH-26 annual/quarterly-factor defect, but is **deliberately not fixed by TASK-2678** — the fix here (`insufficient_data` status + auto-promote) works correctly regardless of which cadence bucket a factor is evaluated against; reclassifying `fact_short_interest`'s cadence bucket (e.g. a new `"biweekly" CADENCE_PARAMS` entry) is a separate, out-of-scope follow-on if the ~2.5-year accrual horizon proves unacceptable in practice.

- **Documentation**: FINRA Equity Short Interest

ETF Reference Data (F-343 / EU-2)

Dataset	Source	Scope	Silver Table	Key Columns	Refresh	API Path
`etf-info`	fmp	per_ticker	`fmp_etf_info`	`ticker`	Weekly (Sat)	`etf/info?symbol=__tick`
`etf-holdings`	fmp	per_ticker	`fmp_etf_holdings`	`ticker, asset`	Weekly (Sat)	`etf/holdings?symbol=__`

- **Domain**: `etf` (new recipe domain registered by F-343). Both datasets are **Silver-only** (no Gold promotion) — reference data for sector-concentration / correlation modeling.
- **etf-info**: One row per ETF profile (expense ratio, AUM, asset class, domicile, inception, sector/country exposure summary). DTO `EtfInfoDTO`.
- **etf-holdings**: N rows per ETF, one per underlying holding (asset symbol, name, weight, shares, market value). DTO `EtfHoldingsDTO`.
- **Symbol scope**: both carry the declarative `symbol_scope: etf` keymap field, which scopes ingestion via `IngestPlannerService` to `is_etf=TRUE` allowlist symbols only (gated on `SB_ALLOW_ETFS=ON`, on since F-247).
- **Gold**: none. Gold facts + holding→`instrument_sk` resolution + holdings-weighted proxies are deferred to EU-4.
- **Tier-4**: the live Bronze pull + constraint-7 field-casing verification + Silver promote is an operator step (TASK-2948).
- **Documentation**: FMP /stable/ ETF Info | FMP /stable/ ETF Holdings

---

Gold Layer Summary

Gold tables are built from Silver data by `GoldDimService`, `GoldFactService`, and `GoldIndexService`. No Silver table is modified during Gold promotion.

Dimension Tables

Table	Type	Source
`dim_date`	Static	SQL bootstrap
`dim_instrument_type`	Static	SQL bootstrap
`dim_country`	Static	SQL bootstrap
`dim_exchange`	Static	SQL bootstrap
`dim_sector`	Static	SQL bootstrap
`dim_industry`	Static	SQL bootstrap
`dim_index`	Static	SQL bootstrap (SP500, NASDAQ100, DJIA)
`dim_instrument`	Data-derived	`company-profile-bulk` + `company-delisted`
`dim_company`	Data-derived	`company-profile-bulk`

Fact Tables

Table	Source Datasets	Grain
`fact_eod`	`eod-bulk-price` (+ `_f` feature columns computed by `EodFeatureService` / `TechnicalFeatureService`, incl. F-266 `seasonality_same_month_f` — same-calendar-month return seasonality, Heston-Sadka 2008)	(instrument_sk, date_sk)
`fact_quarter`	quarterly income + balance sheet + cashflow + key metrics	(instrument_sk, period_date_sk, period)

Table	Source Datasets	Grain
`fact_annual`	annual income + balance sheet + cashflow + key metrics + ratios	(instrument_sk, period_date_sk)
`fact_stock_split`	`stock-split`	(instrument_sk, split_date)
`fact_index_constituents`	sp500/nasdaq100/djia constituent history	(index_sk, instrument_sk, effective_from)
`fact_analyst_estimates`	`analyst-estimates` (annual + quarter)	(instrument_sk, fiscal_year, fiscal_quarter)
`fact_earnings_surprises`	`earnings-surprises`	(instrument_sk, date_sk)
`fact_eps_forecast`	`analyst-estimates` + `earnings-surprises`	(instrument_sk, fiscal_year, fiscal_quarter)
`fact_fundamental_annual`	annual fundamentals + market cap (built by `FundamentalFactorService`)	(instrument_sk, fiscal_year)
`fact_signal_score`	`_f` predictors from `fact_eod` + `fact_fundamental_annual` (built by `MLSignalService`); one row per `(instrument_sk, score_date_sk, model_id)` so multiple algorithm/hyperparam variants coexist	(instrument_sk, score_date_sk, model_id)

Table	Source Datasets	Grain
`fact_portfolio_target`	per-instrument target weights from a portfolio construction algorithm (built by `PortfolioConstructionService`); one row per `(portfolio_id, instrument_sk, rebalance_date_sk)` so multiple algorithm/hyperparam variants coexist	(portfolio_id, instrument_sk, rebalance_date_sk)
`fact_screener`	month-end snapshot joining `fact_eod` price/technical features + fundamentals + `composite_signal_s` from `fact_signal_score` + `target_weight_w` from `fact_portfolio_target`	(instrument_sk, rebalance_date_sk)

Fundamentals knowledge-date dissemination lag (F-442 / LH-29, default OFF): the factor knowledge `date_sk` derived from the four reported-fundamentals tables — `fact_fundamental_quarter`, `fact_fundamental_annual`, `fact_valuation_annual`, `fact_moat_annual` — carries an optional dissemination lag (switch `SB_FUNDAMENTALS_DISSEMINATION_LAG_ENABLED`, `sbfactors.settings`; **default OFF → byte-identical**). When ON, the knowledge date is pushed to period-end +

N calendar days (quarterly +60d / annual +90d) snapped forward to the first US market day, so factor values are stamped on-or-after the provable SEC filing date (closing the F-319 EDGAR PIT look-ahead). Announcement-grain / estimate / short-interest tables (`fact_earnings_momentum`, `fact_eps_forecast`, `fact_short_interest`) are excluded — already PIT-safe.

Silver-Only Datasets (No Gold Promotion)

Silver Table	Reason
`fred_dgs10`	No `instrument_sk` FK — macro time-series
`fred_usrecm`	No `instrument_sk` FK — macro indicator
`fred_dgs1mo`	No `instrument_sk` FK — macro time-series
`fred_dgs3mo`	No `instrument_sk` FK — macro time-series
`fred_dgs1`	No `instrument_sk` FK — macro time-series
`fred_dgs2`	No `instrument_sk` FK — macro time-series
`fred_dgs5`	No `instrument_sk` FK — macro time-series
`fred_dgs30`	No `instrument_sk` FK — macro time-series
`fred_t10y2y`	No `instrument_sk` FK — macro time-series
`fred_baa10y`	No `instrument_sk` FK — macro time-series

Silver Table	Reason
`fred_vixcls`	No `instrument_sk` FK — macro time-series
`fmp_market_risk_premium`	No `instrument_sk` FK — country-level data
`kenneth_french_factors`	No `instrument_sk` FK — market-wide factor return series
`fmp_etf_info`	Reference data (F-343 / EU-2); Gold facts + holding→`instrument_sk` resolution deferred to EU-4
`fmp_etf_holdings`	Reference data (F-343 / EU-2); Gold facts + holding→`instrument_sk` resolution deferred to EU-4

Bespoke reference tables (not in `config/dataset_keymap.yaml`). The following Silver table is populated by a dedicated fetch→Silver service that bypasses the Bronze recipe engine entirely, so it has **no keymap entry and no Bronze parquet** behind it. Its provenance is a lightweight `fetched_at` / `source_url` pair rather than the standard §10.4 3-column Bronze lineage:

- **`kenneth_french_factors`** (F-316): the Kenneth R. French Data Library factor-return series — Fama-French 5-factor (`mkt_rf`/`smb`/`hml`/`rmw`/`cma`), Carhart momentum (`umd`), and the risk-free rate (`rf`), keyed on `(date, frequency)` with `frequency` `{daily, monthly}`. Fetched via `sbops.kenneth_french_service.KennethFrenchFac` (stdlib `urllib`/`zipfile`, no new runtime dependency); values are stored as decimals (French serves percent). The live download is a Tier-4 operator step. A keymap identity entry was **deferred** (it would require editing the F-102 exact-count inventory guard for no runtime benefit — the table is populated out of band).

Designed Earliest Date

The `designed_earliest_date` field on each dataset entry in `config/dataset_keymap.yaml` is the per-dataset commitment of how far back the platform aspires to ingest. Used by the Combined Coverage Heatmap (F-083) to draw the green per-row marker and to contextualize gaps in `MIN(coverage_from)` against design intent. `null` indicates a forward-looking or snapshot dataset where historical depth does not apply.

Dataset	Discriminator	Designed Earliest	Notes
`eod-bulk-price`	—	1980-01-01	Matches platform `FROM_DATE`.
`eod-price-history`	—	1980-01-01	Per-ticker delisted backfill.
`company-profile-bulk`		—	Snapshot.
`company-delisted`	—	—	Snapshot.
`income-statement-bulk-quarter`		1985-01-01	Conservative per-symbol floor.
`balance-sheet-bulk-quarter`		1985-01-01	
`cashflow-bulk-quarter`		1985-01-01	
`key-metrics-bulk-quarter`		1985-01-01	
`income-statement-bulk-annual`		1985-01-01	
`balance-sheet-bulk-annual`		1985-01-01	
`cashflow-bulk-annual`		1985-01-01	
`key-metrics-bulk-annual`		1985-01-01	
`ratios-bulk-annual`		1985-01-01	

Dataset	Discriminator	Designed Earliest	Notes
`sp500-constituent-history`		—	Change-log of current-as-of constituents.
`nasdaq100-constituent-history`		—	Change-log.
`djia-constituent-history`		—	Change-log.
`stock-split`	—	1980-01-01	
`analyst-estimates`	annual	—	Forward-looking.
`analyst-estimates`	quarter	—	Forward-looking.
`earnings-surprises`		1980-01-01	Pull-back to platform floor.
`market-risk-premium`		—	Snapshot.
`fred-dgs10`	—	1962-01-02	FRED series start.
`fred-usrecm`	—	1900-01-01	Clamped to align with the pre-1960 bucket.
`fred-dgs1mo`	—	2001-07-31	FRED series start.
`fred-dgs3mo`	—	1982-01-04	FRED series start.
`fred-dgs1`	—	1962-01-02	FRED series start.
`fred-dgs2`	—	1976-06-01	FRED series start.
`fred-dgs5`	—	1962-01-02	FRED series start.
`fred-dgs30`	—	1977-02-15	FRED series start.

Dataset	Discriminator	Designed Earliest	Notes
`fred-t10y2y`	—	1976-06-01	FRED series start.
`fred-baa10y`	—	1986-01-02	FRED series start.
`fred-vixcls`	—	1990-01-02	FRED series start.
`equity-short-interest`		2007-01-01	FINRA reporting begins 2007.

Vendor Earliest Date

The `vendor_earliest_date` field is the **hard runtime floor** on the dataset — the earliest period the upstream vendor actually serves data for. Distinct from `designed_earliest_date` (aspirational depth). Consumed by the F-082 `quarter_ingest_schedule` and `annual_ingest_schedule` prechecks (TASK-325) to clamp the history-extension window to `max(FROM_DATE, vendor_earliest_date)`, so the schedule never asks FMP for a period it returns empty for.

Seeded from observed `MIN(date)` of each Silver table after multiple nightly runs. Update this field when the vendor extends coverage (lowering the floor) or when we learn of a stricter floor than what's currently set. `null` (or omitted) means no known floor; the precheck falls back to `FROM_DATE`.

Dataset	Vendor Earliest	Source of observation
`income-statement-bulk-quarter`	1984-12-31	silver_min on run 260513_7e0c61

Dataset	Vendor Earliest	Source of observation
`balance-sheet-bulk-quarter`	1983-06-30	silver_min on run 260513_7e0c61
`cashflow-bulk-quarter`	1985-06-30	silver_min on run 260513_7e0c61; FMP returned empty for all 4 quarters in 1984.
`key-metrics-bulk-quarter`	1983-06-30	silver_min on run 260513_7e0c61
`income-statement-bulk-annual`	1983-12-31	FMP returned empty for year=1982 on run 260513_7e0c61.
`balance-sheet-bulk-annual`	1983-12-31	same
`cashflow-bulk-annual`	1983-12-31	same
`key-metrics-bulk-annual`	1983-12-31	same
`ratios-bulk-annual`	1983-12-31	same

Per-Dataset Ingest Caps (F-102)

The `max_date_quantity` and `max_symbol_quantity` keymap fields (F-102) replace the four global `INGEST_*_COUNT` constants formerly in `sbshared.settings`. The planner (`sbbronze.ingest_planner.IngestPlannerService`) reads these caps to decide:

- **DateList variant** — (`ticker_scope=global, date_key set, max_date_quantity > 1`). Planner returns up to `max_date_quantity` ISO dates to ingest.

- **SnapshotGate variant** — `(date_key=null OR max_date_quantity=null OR max_date_quantity=1)`. Planner returns a bool gate (ingest yes/no).
- **SymbolList variant** — `(ticker_scope=per_ticker)`. Planner returns up to `max_symbol_quantity` symbols (or all allowlisted when null).

Dataset	Discriminator	Variant	`max_date_quantity`	`max_symbol_quantity`
`eod-bulk-price`	—	DateList	20	—
`company-profile-bulk`	—	Snapshot	null	—
`income-statement-bulk-quarter`	—	DateList	4	—
`balance-sheet-bulk-quarter`	—	DateList	4	—
`cashflow-bulk-quarter`	—	DateList	4	—
`income-statement-bulk-annual`	—	DateList	4	—
`balance-sheet-bulk-annual`	—	DateList	4	—
`cashflow-bulk-annual`	—	DateList	4	—
`key-metrics-bulk-quarter`	—	DateList	4	—
`key-metrics-bulk-annual`	—	DateList	4	—
`ratios-bulk-annual`	—	DateList	4	—
`fred-dgs10`	—	DateList	8	—
`fred-usrecm`	—	DateList	8	—
`fred-dgs1mo`	—	DateList	8	—
`fred-dgs3mo`	—	DateList	8	—
`fred-dgs1`	—	DateList	8	—
`fred-dgs2`	—	DateList	8	—
`fred-dgs5`	—	DateList	8	—
`fred-dgs30`	—	DateList	8	—
`fred-t10y2y`	—	DateList	8	—
`fred-baa10y`	—	DateList	8	—
`fred-vixcls`	—	DateList	8	—
`market-risk-premium`	—	Snapshot	null	—
`company-delisted`	—	Snapshot	null	—
`mergers-acquisitions`	—	Snapshot	null	—

Dataset	Discriminator	Variant	max_date_quantity	max_symbol_quantity
sp500-constituent-history		Snapshot	null	—
nasdaq100-constituent-history		Snapshot	null	—
djia-constituent-history		Snapshot	null	—
stock-split	—	SymbolList	null	null
eod-price-history		SymbolList	null	null
analyst-estimates	annual	SymbolList	null	null
analyst-estimates	quarter	SymbolList	null	null
earnings-surprises		SymbolList	null	null
equity-short-interest		DateList	8	—

When introducing a new dataset, set `max_date_quantity` per the variant intent. Defaults: 20 for daily EOD-style data, 8 for weekly cadences, 4 for quarterly/annual fundamentals, `null` for snapshots.

> **Self-healing gap-fill (O-093 / TASK-2812)**: the nightly bulk EOD ingest plan (`eod-bulk-price`) forward-fills up to `max_date_quantity` (20) missed market days per night, so a single run can absorb roughly a four-week outage. In the no-backlog steady state the plan is still just `[today]` (or `[]` on a non-market day); the cap only affects the saturated/outage path.

Quick Reference

Metric	Count
Total datasets (keymap entries)	34
DateList variant (F-102)	22

Metric	Count
Snapshot variant (F-102)	7
SymbolList variant (F-102)	5
Global-scope datasets	29
Per-ticker datasets	5 (`stock-split`, `eod-price-history`, `analyst-estimates` ×2, `earnings-surprises`)
FMP datasets	22
FRED datasets	11
FINRA datasets	1
Daily refresh (eod)	3
Weekly refresh (Sat, eow)	19
Monthly refresh	2
Quarterly refresh	2
Yearly / cadence refresh	2
Gold fact tables	12
Gold dimension tables	9 (7 static + 2 data-derived)