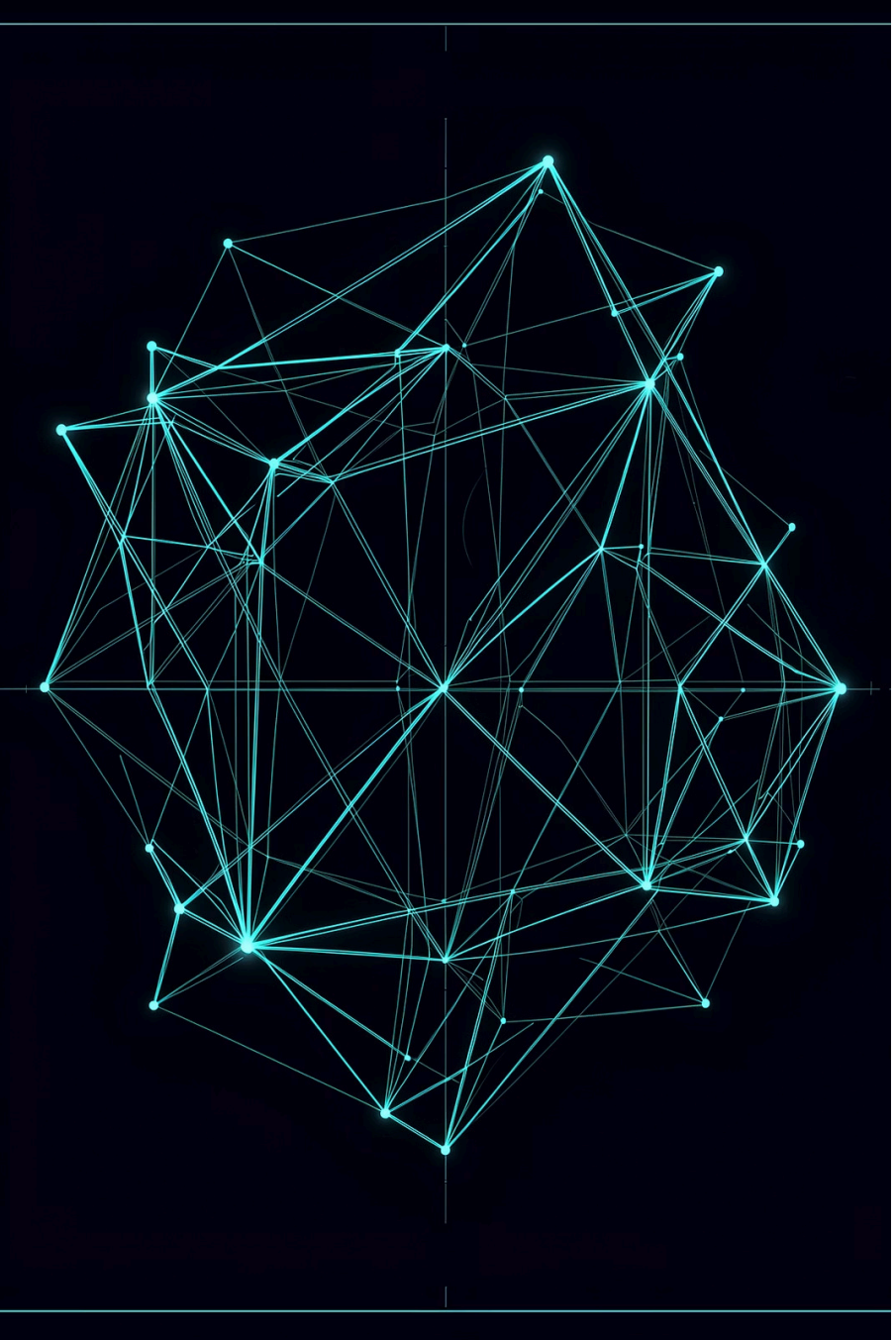


Jagged Boundaries in Programmable Treasury Architecture

A methodology note for treasury-systems and quant-risk professionals. Companion to *The Missing Third Layer of Treasury AI* (David Martin, April 2026).

The Problem the Clean Model Hides

The three-domain model of treasury analytics is clean. Domain 1: deterministic rules. Domain 2: probabilistic descriptive. Domain 3: probabilistic prescriptive. It is a useful map. But the failures that bite are not inside the domains — they live in the seams between them, invisible in the map precisely because the map shows well-bounded boxes.

The Failure Mode

A hedge-ratio optimizer goes to a corner solution the moment a correlation regime breaks. The policy bound, meant to constrain the optimizer, becomes the decision-maker. The system ships a hedge ratio chosen by a rule, not by the variance minimizer — and nobody downstream sees it. The audit trail records the clipped output, not the fact that the clip bound.

The Governance Consequence

That is not a Domain 3 failure. It is a failure of the typed contract between Domain 1 and Domain 3. When a regulator asks "what did the system decide and why," the answer is: "an interface assumption we cannot now reconstruct." That is a provenance failure with governance consequences — in front of model validators, risk managers, and compliance.

⚠ This note makes the interface — the typed, directed contract between domains — the primary engineering object.

Three Candidate Models for the Structure

Each structural model captures something real — but only one preserves both direction and type on the cross-domain boundary.

1	2	3
Three-Partite Graph with Cross-Partition Edges Correct about the scalar — how many cross-domain connections exist — but silent about direction and what flows across the boundary. A rule that *bounds* an optimizer is treated identically to an output that *informs* a threshold. The distinction that matters is erased.	**Hypergraph** Handles native multi-domain operators cleanly as a single hyperedge — but direction is lost. The covenant bounds the forecast; the forecast informs the remediation; collapsing them into co-membership papers over the structure that matters most.	**Typed-Edge DAG (Adopted)** Direction is preserved in the graph; type is preserved on the edge. The contract vocabulary becomes the analysis. This is the model adopted throughout this note — the only candidate that makes the seam visible as an engineering object.

Formal Substrate: The Typed-Edge DAG

Let **D = (V, E, τ)** be a directed acyclic graph. V is a finite vertex set partitioned across V_1, V_2, V_3 — domain-membership partitions representing which stratum of the treasury architecture a vertex belongs to, *not* the ASC 820 / IFRS 13 observability hierarchy. τ maps each edge to a contract type from a finite vocabulary.

> ⓘ The type of an edge — not just its existence — carries the engineering content of the seam.

Five Contract Types

bound

Domain 1 outputs a hard constraint that gates a downstream result.

score-to-threshold

Domain 2 outputs a probabilistic score; a Domain 1 threshold collapses it to pass/fail.

forecast

Domain 2 outputs a probabilistic description of state.

recommendation

Domain 3 outputs a priced decision.

clip

A Domain 1 bound applied to a Domain 3 output.

The Interface Contract as Engineering Object

Every cross-domain typed edge carries an interface contract with four components — the thing a model validator should be auditing, and the thing a descriptive-only architecture does not build.

What Data Crosses

Schema, units, frequency, horizon, point vs. distribution. Drift modes: schema break, unit mismatch, frequency mismatch.

What Assumption Is Encoded

Stationarity, independence, distributional shape, regulatory equivalence — usually implicit. Drift modes: regime change, correlation break, distributional shift.

What Audit Trail Survives

Provenance: what arrived, which contract version was in force, what the output was, whether an override occurred. Drift modes: silent override, lost provenance, irreproducibility.

What Breaks When It Drifts

Blast radius: which downstream nodes silently consume bad output, which fire alerts, which fail closed.

A jagged-frontier failure is a failure of one of these four components on a typed cross-domain edge.

Worked Example 1 — Hedge-Ratio Recommendation

The Circuit

v_policy (Domain 1 bound, interval [0.5, 1.0]) → correlation estimator **v_corr** (Domain 2) → minimum-variance optimizer **v_opt** (Domain 3) → shipped ratio **v_ship** = v_opt clipped by v_policy.

The Failure

The load-bearing edge is the **clip** edge v_policy → v_ship. Its contract silently assumes variance minimization is still right in the regime where the bound binds. When correlation breaks, the optimizer goes to a corner at 1.0, the bound binds, and the system ships a ratio the *bound* chose — not the optimizer.

⚠ The audit trail does not record that the clip was binding. The audit-trail component of the clip contract is the engineering gap.

Worked Example 2 — Sanctions Screen

The Circuit

Probabilistic name-matcher **v_match** (Domain 2) → deterministic gate **v_gate** (Domain 1), with threshold **v_thresh**.

The Failure Mode

The load-bearing **score-to-threshold** edge encodes a calibration assumption: a 0.85 means the same thing across the sanctions list as across a fresh prospect file. Recalibrate the matcher without updating the threshold and the assumption is silently breached — same binary output, moved false-positive rate, compliance sees nothing.

⊗ Architecturally identical to the hedge-ratio case: a deterministic envelope around a probabilistic interior. Same pattern, different contract type, different drift mode.

The Structural Parallel

This is the power of the typed-edge DAG: a single formal model makes the parallel visible across systems that look entirely unrelated. The sanctions screen and the hedge-ratio optimizer share the same jagged-boundary failure pattern — a Domain 1 envelope silently overriding a Domain 2 or Domain 3 interior without leaving a recoverable audit trail.

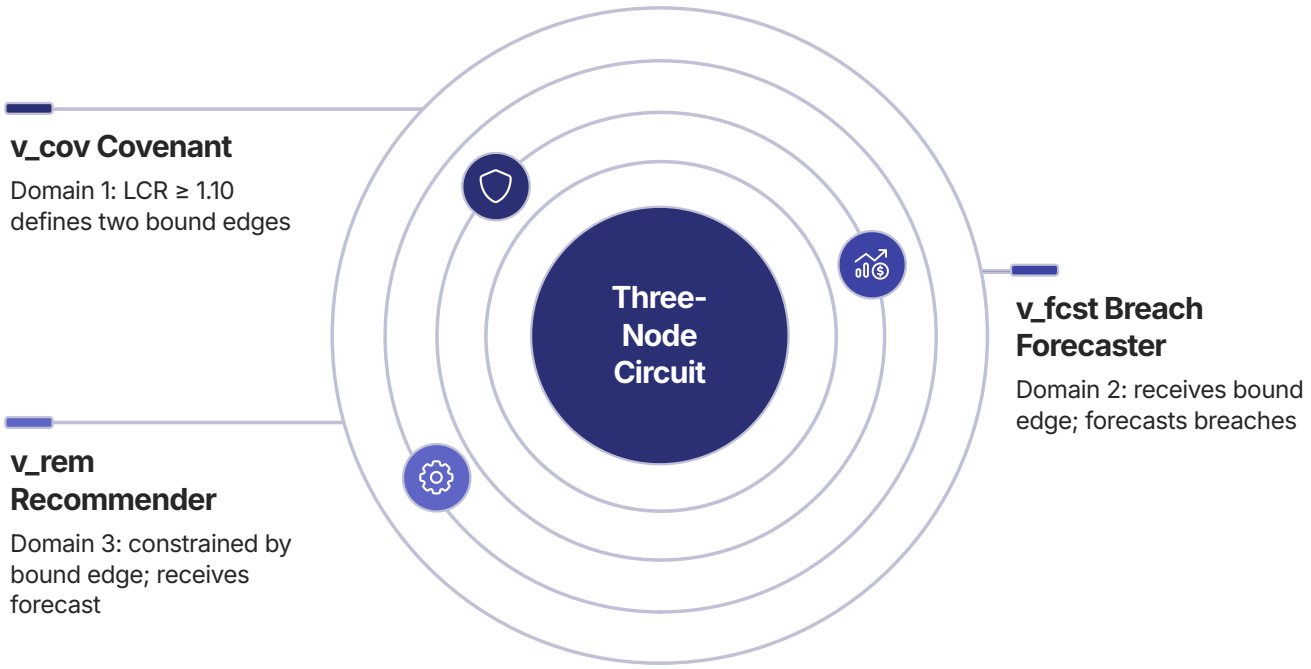
Hedge Ratio

clip edge — Domain 1 bound on Domain 3 output

Sanctions Screen

score-to-threshold edge — Domain 1 threshold on Domain 2 score

Worked Example 3 — Liquidity Covenant Tracker



The Circuit

Covenant **v_cov** (Domain 1, LCR ≥ 1.10) → breach forecaster **v_fcst** (Domain 2) → remediation recommender **v_rem** (Domain 3).

Two Bound Edges

Two **bound** edges run from the covenant: it defines the breach event the forecast forecasts, and it constrains which remediations are admissible. The **forecast** edge carries a tail-event calibration assumption: has the forecaster ever been validated on an actual LCR breach? Usually not — the output at low LCR is extrapolated, not calibrated, and the contract does not say so.

⚠ As a hybrid operator the team *names* the tracker as one hyperedge across three domains. The typed-edge DAG is how it must be *built*. The gap between naming and building is where the engineering fails.

What Changes in the Engineering Picture

The descriptive-vs-prescriptive framework is preserved in full. Domain 3 — prescriptive — is still the missing piece for digital-asset treasury. What changes is the relationship between Domain 3 and Domain 1.

Domain 3 Does Not Exist Separately from Domain 1

The prescriptive capability of a treasury system exists only as the union of its interface contracts with the rules domain beneath it. An optimizer without the clip contract is not a treasury tool — it is a portfolio-optimization exercise.

The Vendor Gap Is Reframed

Descriptive-only vendors have analytical sophistication in Domain 2 and Domain 3. What they lack is the typed cross-domain interface that makes the modeling operationally deployable under audit — because the contracts presuppose a thick rules domain they did not build.

The Seam Is the Engineering Object

The seam — not the analytical domain on either side of it — is the engineering object that distinguishes deployable treasury AI from demonstration-grade treasury AI.

Open Question — The Reflexive Loop

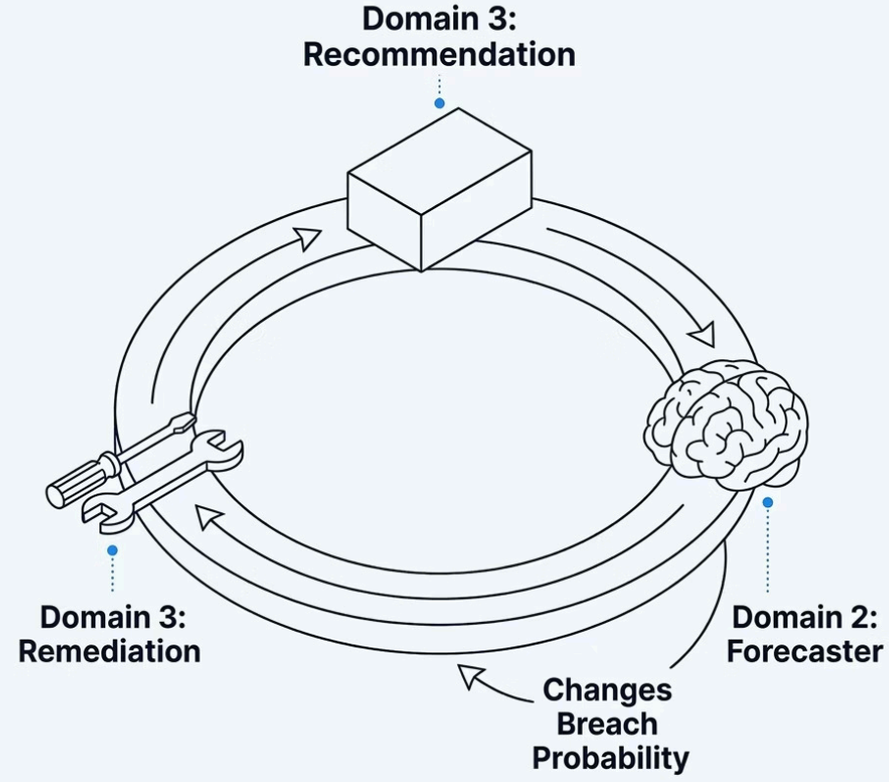
The Gap in the Contract Vocabulary

The five contract types do not cover a reflexive feedback loop: a Domain 3 output that feeds back into a Domain 2 forecaster — a remediation that changes the portfolio being forecasted, changing the breach probability, changing the remediation. This requires at least one additional contract type, not yet defined.

What the Existing Protocol Covers

The Aug 15, 2026 reflexivity calibration protocol (SSRN v3 §8.1.4) addresses a *related but distinct* problem — rho estimation, CUSUM-R drift detection, re-calibration cadence. It is a protocol for an already-identified reflexivity risk; it does not specify the missing contract type.

> Whether that work surfaces the contract type as a derived construct, or it must be specified independently, remains open.



Citation

> Dell'Acqua, F., McFowland, E. III, Mollick, E. R., et al. (2023). "Navigating the Jagged Technological Frontier." *Harvard Business School Working Paper 24-013.*

The original frame concerns AI capability across knowledge-work tasks; the transfer to treasury analytics is an analogy on the conceptual utility of "jagged," not an application of their empirical findings.

About This Note

This methodology note is a companion to *The Missing Third Layer of Treasury AI* by David Martin (April 2026). It develops the formal substrate — the typed-edge DAG and interface contract vocabulary — that underlies the three-domain treasury analytics framework described in that work.

Intended Audience

Treasury-systems engineers, quantitative risk professionals, model validators, and compliance architects working on the deployment of AI-assisted treasury tools under regulatory audit requirements.

Key Contribution

The seam — the typed, directed contract between analytical domains — is the primary engineering object. Jagged-frontier failures are failures of interface contracts, not failures of the analytical domains on either side.