Consequitur Technical Document

Edition 1.0 — June 2026

Quantcha

June 2026

Contents

Abstract . . . 2
1. Introduction . . . 2
2. Lineage and related work . . . 4
3. System model . . . 5
4. Estimate semantics . . . 7
5. Consensus . . . 10
6. Propagation and scenarios . . . 11
7. Worked example: the December 2026 Fed anchor graph . . . 14
8. Limitations . . . 16
9. Roadmap . . . 18
10. References . . . 21

This document describes the Consequitur system as shipped in the June 2026 public beta. It is a systems document in the tradition of the *RiskMetrics Technical Document* [9], not a research paper: the component mathematics is established literature, cited throughout, and the contribution we claim is the set of design decisions that assemble those components into a running product. No empirical results are presented because none exist yet; a future revision will incorporate crowd-estimate data accumulated during the public beta. Questions and corrections: ed@quantcha.com.

Cite as: Quantcha. *Consequitur Technical Document*, Edition 1.0. June 2026. quantcha.com

Abstract

Consequitur is a cross-event sensitivity platform: it maintains a directed graph of real-world events and the conditional probability relationships between them. Event contract markets price the *levels* of individual events continuously and well; they price the *conditionals* between events almost nowhere. Consequitur fills that gap with a proprietary, exchange-decoupled event registry; a sensitivity graph whose edges carry conditional log-odds shifts seeded by an AI synthesis and corrected by crowd estimates; and a propagation engine that lets a user move any event's probability and watch the implied repricing ripple through the graph. This document specifies the machinery precisely—the estimate representation, the consensus aggregation, the propagation semantics as locked in code—together with the assumptions each component makes and the conditions under which each breaks. Every mathematical component is standard; the system is the product.

1. Introduction

1.1 The asymmetry that motivates the system

A binary event contract is a clean instrument: it trades between 0 and 1, and its price is a direct, continuously updated estimate of the probability that a well-specified event occurs. Exchanges and venues—Kalshi, Polymarket, PredictIt, Manifold, among others—now price thousands of such events across macroeconomics, politics, energy, and finance. The *marginal* probability of almost any newsworthy event is observable in real time.

What is not observable is the structure between events. If August payrolls print below +100k, what happens to the probability that the Fed's target range is at or below 3.50% after the September meeting? Every macro desk holds a view on that conditional; no liquid market quotes it. Combinatorial markets that would price joint and conditional outcomes directly have been proposed, prototyped, and studied for over two decades, and the literature is clear about why they have not scaled: pricing general combinations is computationally intractable, and liquidity fragments across an exponential outcome space (§2). The result is a persistent asymmetry: levels are priced; conditionals are not.

Consequitur addresses the asymmetry from the analytics side rather than the market-design side. Instead of building a venue that prices conditionals through trade, it maintains the conditional structure as a curated, AI-seeded, crowd-corrected data layer *on top of* the existing markets—and makes that structure explorable, shareable by link, and (on the roadmap) consumable as a sensitivity feed.

1.2 What Consequitur is

Consequitur maintains three layered objects, specified in §3:

1. an **event registry**—a proprietary ontology of real-world binary events, each with an objective resolution criterion and a deadline, keyed independently of any exchange;
2. a **sensitivity graph**—directed edges between registry events, each edge carrying the conditional probability of the target given the catalyst's resolution, stored as a conditional log-odds shift with full provenance;
3. a **mapping layer**—associations from registry events to external probability sources (exchange contracts, and in time polls and expert forecasts), from which each event's live baseline probability is synthesized.

On top of these sit the user-facing mechanics: an **estimate flow** through which any visitor can record their own conditional view on an edge (§4); a **consensus computation** that aggregates estimates into a crowd value per edge (§5); and a **Scenarios engine** that propagates hypothetical probability shocks through the graph in real time (§6).

The public beta is live as of June 9, 2026 at `consequitur.com`. The beta ships a single-anchor graph—21 events and 22 dependencies organized around the December 2026 FOMC rate question (§7)—with anonymous estimating, the Scenarios view, and a published methodology page. Estimates are captured and consensus is computed from day one; the crowd consensus surface itself activates when estimate volume justifies it. Throughout this document, present tense is reserved for what is live in the beta; everything else is explicitly marked as roadmap.

1.3 Who this document is for

The intended reader is quant-literate: an event contract trader who came from options and wants to know exactly what the numbers mean before trusting them; an institutional evaluator deciding whether the eventual sensitivity feed belongs in a modeling pipeline; an academic-adjacent reader checking whether the system respects the literature it draws on. We assume comfort with logits, conditional probability, and market microstructure vocabulary. The companion methodology page on the product is the gentle version of this material; this document is the precise one.

1.4 Reading guide

§2 places the system in its intellectual lineage and states plainly what is and is not novel. §3 specifies the three layered objects. §4 defines the estimate representation—the single most important set of semantics in the system—including the invariant choices the model makes and what they do *not* mean. §5 specifies consensus aggregation. §6 specifies propagation as locked in the shipped code. §7 walks the launch graph end to end with a concrete trace. §8 is an unhedged account of the system's limitations. §9 describes the roadmap, including hazard-implied drift and synthetic market discovery.

2. Lineage and related work

The mathematical components of Consequitur are established, and the system is designed so that this is a feature: each component is well understood, its failure modes are documented in prior literature, and our design decisions are auditable against that literature.

Combinatorial information markets. Hanson introduced combinatorial information markets and the market scoring rule, showing how a single subsidized scoring rule can support trade over combinations of outcomes [1], and developed the logarithmic market scoring rule (LMSR) whose modularity properties made combinatorial aggregation practical in principle [2]. Hanson's earlier work on decision markets [3] framed the use of conditional prices as decision inputs—the same use case Consequitur's conditional structure ultimately serves, with an analytics layer in place of a market mechanism.

The tractability wall. Chen, Fortnow, Lambert, Pennock, and Wortman established that maintaining correct LMSR prices over combinatorial outcome spaces is #P-hard in natural settings [4]. This result is, in a precise sense, the reason Consequitur is shaped the way it is: a fully general joint distribution over the event registry is unattainable, both computationally and as a crowd-elicitation problem (a node with k inbound dependencies would need 2^k conditional table entries). Consequitur adopts the standard escape—a log-linear model, linear in the number of edges—and accepts its independence assumptions explicitly (§6, §8).

Bayesian network market makers. Sun, Hanson, Laskey, and Twardy showed how a Bayesian network's factored representation makes combinatorial market computation tractable when the dependency structure is itself sparse and graph-shaped [5]. Consequitur's sensitivity graph is the same structural insight applied to an analytics product: sparsity is imposed by curation (only analytically significant dependencies are seeded), and the graph, not a joint table, is the object that scales.

Stochastic structure of traded probabilities. Recent work has begun supplying prediction markets with the stochastic-process vocabulary options markets have had for fifty years; Dalen's logit jump-diffusion kernel with multi-event correlation [6] is a representative example, treating traded probabilities as martingales with quotable volatility, jump, and dependence factors. Consequitur operates one level up from this literature—its object is the conditional structure between events rather than the dynamics of any single price—but shares its working space: log-odds, where probability arithmetic is additive and bounded-domain pathologies disappear.

Info finance. Buterin's framing of "info finance" [7]—starting from a fact you want to know and designing a market to elicit it—describes the demand side Consequitur serves. The conditional probability of a December rate outcome given an October oil level is a fact someone wants to know; no market elicits it; Consequitur elicits it with an AI prior and a crowd correction instead.

The per-edge quantity is a textbook object. The coefficient Consequitur stores per edge—the conditional log-odds shift, $\beta = \text{logit}(P(B \mid A)) - \text{logit}(P(B))$—is the coefficient on a binary predictor in a logistic regression, and a close relative of the log odds ratio of epidemiology [8]. We did not coin it and do not claim to have. The same is true of every other component: the law of total probability, Bayes-coherent reverse conditionals, exponential recency decay, weighted means, quantile dispersion statistics.

What we claim. The contribution is the assembled, productized system: a cross-platform, exchange-decoupled event ontology; an AI-seeded, crowd-corrected conditional layer with full provenance and replayable consensus; propagation semantics precise enough to lock in code and document here; and, on the roadmap, synthetic market discovery as a packaged user-facing capability (§9.2). Where this document uses the word "novel," it means the productization—the thesis that this assembly is buildable and useful as a live product—not the mathematics.

3. System model

3.1 The event registry

The registry is the source of truth for what an event *is*, keyed independently of any exchange. Each event carries:

- a **slug**—a URL-safe, human-readable, immutable identifier (published links never break);
- a **description**—the canonical statement of the event;
- a **canonical resolution criterion**—the registry's own written rule for evaluating the outcome, objective enough to be evaluated at the deadline. The criterion is deliberately settlement-payout-free: unlike an exchange contract, it does not need legally adjudicable settlement language, only objective evaluability. This loosens what the registry can include relative to listed markets;
- a **deadline**—the moment at which the resolution criterion can be evaluated. Every event has one; and
- a **temporal flavor**—*hard-deadline* (the event resolves at the deadline regardless, e.g. a rate level after a scheduled meeting) or *soft-deadline* (the event formally resolves by a boundary but may effectively resolve earlier, e.g. a ceasefire holding through a date).

Two registry disciplines matter analytically. First, **every event is dated and objective**. Continuous concepts ("dollar strength," "oil stays elevated") do not enter the registry as such; they are operationalized into specific dated observations ("Brent front-month close $\geq$ \$100 on 2026-12-04") or excluded. An event the system cannot date and evaluate is an event the graph cannot propagate through. Second, **each event has one canonical framing**, chosen at curation time. The event's question is asked once and everywhere reads the same way; dependencies express directionality through the sign of their conditional shift, never by re-asking the question in the opposite polarity.

Internally, curated events follow a (*quantity, term, strike*) shape familiar from listed derivatives: the quantity is the observable (fed funds upper bound, core CPI month-over-month, Brent close), the term is the observation moment, and the strike is the threshold. Events sharing a quantity and strike across different terms form a series, and a curation rule—**one instance per series per target**—keeps the graph clean: for any given target, at most one member of a series is seeded as a catalyst, the instance with the latest deadline on or before the target's own deadline. §7 shows why this rule is also the propagation architecture.

3.2 The sensitivity graph

A **dependency** (an *edge*, in data-layer vocabulary) is a directed relationship from a source event A (the *catalyst*, in that relationship) to a target event B. The edge carries the conditional structure: $P(B \mid A{=}\text{YES})$ and $P(B \mid A{=}\text{NO})$, with the YES-direction conditional stored as a log-odds shift β (§4) and the NO-direction conditional derived from it by the law of total probability rather than estimated separately. The two conditionals are not free of each other—they are tied through the catalyst's marginal—so eliciting one and deriving the other enforces within-edge coherence by construction.

Edges carry full provenance. Every edge is seeded with an initial conditional, and every seed records its inputs: the contributing market quotes consulted, news context, and the synthesis reasoning. Seeds live in **epochs**: when the system's own re-evaluation of an edge moves the seeded β by at least 0.1 in log-odds (roughly one point of displayed conditional), a new epoch is created rather than the old seed being silently overwritten. Epochs exist because they carry consensus-bucketing semantics: a crowd estimate is an argument against a specific prior, and estimates made against a superseded prior are preserved but not pooled with estimates made against the current one (§5).

Curation imposes structure the schema alone would not. Edges are seeded in the **forward causal direction** (catalyst earlier, target later). Where propagation requires a reverse-direction conditional, it is derived on the fly via Bayes from the forward β and current marginals rather than stored (§6). The launch graph is acyclic by curation; the propagation engine carries cycle guards regardless, and richer cyclic structure—including informational relationships where a market is *symptomatic* of a driver rather than causal, and its movement is evidence about the driver—is an anticipated future direction rather than a supported present one (§8.7, §9).

Operationally, each edge in the launch graph is classified by where the disagreement lives: **confir-**

mation edges (direction agreed, magnitude debated—the backbone), **contested-magnitude** edges (sign agreed, size disputed—most data-print edges), and **contested-sign** edges (genuine directional disagreement—the analytically interesting ones). The classification is editorial rather than schematic, but it shapes both curation and content: a healthy graph routes its disagreement onto edges built to carry it.

3.3 The mapping layer

Registry events map to **zero, one, or many** external probability sources: exchange contracts today (Kalshi, Polymarket, PredictIt, Manifold), with polls and expert forecasts as natural future source types. Each mapping records the external market's own resolution criterion alongside the registry's canonical one, so divergence between the two is auditable—a mapping can be *exact* or an explicit *proxy* with documented divergence (the launch graph contains both; §7).

From the mapped sources, the system synthesizes each event's **AI Estimate**: the live baseline (marginal) probability used everywhere downstream. The synthesis is an AI step, not a fixed formula: it takes the available market quotes, news context, and analytical reasoning as inputs and produces a probability with recorded provenance, including the specific quotes consulted and their weights. Where an event has multiple mapped sources, the synthesis weighs them; where it has none, the AI Estimate is a reasoned baseline and is labeled identically (the graph does not require an event to be listed anywhere—this is the property synthetic market discovery later builds on, §9.2). The synthesis implementation is expected to evolve—how many sources contribute, how they are weighted, how reasoning and quotes blend—and the provenance record, not a fixed aggregation formula, is the stable contract.

Every active event's baseline is refreshed on a regular cadence, and each refresh writes a timestamp and provenance snapshot whether or not the value moved—freshness in the product means "this was checked," with the value's history a separate concern. The system is deliberately decoupled at runtime from its data sources: market data enters through a curated ingest process, and the deployed platform serves only what has been written to it.

4. Estimate semantics

This section defines what an estimate *is*, mathematically and epistemically. The definitions here are the system's most load-bearing semantics, and also the easiest to misread; the second half of the section is about what the stored quantity does **not** mean.

4.1 Working space and conventions

All arithmetic on probabilities happens in log-odds space:

$$\operatorname{logit}(p) = \ln \frac{p}{1-p}, \qquad \sigma(x) = \frac{1}{1+e^{-x}}$$

with σ (the logistic sigmoid) inverting the logit. Log-odds space is unbounded, symmetric about zero, and—decisively—the space in which independent evidence is additive: updating odds by a likelihood ratio is adding a constant in log-odds. Every computation in the system follows the same shape: convert to log-odds, do additive arithmetic, convert back with σ for display.

Two numerical guards apply throughout, in the backend and the in-browser propagation engine identically: probabilities are clamped to $[\varepsilon, 1-\varepsilon]$ with $\varepsilon = 0.001$ before any logit, and every β is clamped to $[-6, +6]$. Both are starting defaults—calibration placeholders, re-fittable later. Because the system stores raw inputs and derives everything else at read time (§4.3), re-fitting them never requires a data migration.

4.2 The β representation

An edge's strength is a single number: the **conditional log-odds shift**

$$\beta \;=\; \operatorname{logit}\big(P(B \mid A{=}\text{YES})\big) \;-\; \operatorname{logit}\big(P(B)\big).$$

In words: β is how much observing the catalyst resolve YES shifts the log-odds of the target—the amount of evidence, on the additive scale, that A's resolution provides about B. A user does not enter β directly. The estimate flow asks for the absolute conditional probability $P(B \mid A{=}\text{YES})$ on a picker, and the interpreter derives

$$\beta_{\text{estimate}} \;=\; \operatorname{logit}(p_{\text{entered}}) \;-\; \operatorname{logit}\big(P(B)_{\text{at estimate time}}\big),$$

where $P(B)_{\text{at estimate time}}$ is the target's AI Estimate at the moment of submission, snapshotted onto the estimate record. The displayed current conditional for any edge is always derived live:

$$P(B \mid A{=}\text{YES})_{\text{displayed}} \;=\; \sigma\big(\operatorname{logit}(P(B)_{\text{current}}) + \beta\big).$$

β is stored (conceptually; see §4.3) because it describes a *relationship* rather than a price level. A stored conditional of 0.55 goes stale the moment the target's market moves; a stored shift of -0.55 keeps producing a sensible conditional as the baseline moves underneath it. The estimate's semantics are **total effect**: a β is the net conditional shift over the implied time horizon, integrating every channel through which the catalyst bears on the target into one number. The beta does not decompose direct from mediated effects (a deliberate simplification; §8).

4.3 Storage discipline: raw in, derived out

The system never persists an interpreted β, a consensus value, or any other derived quantity. An estimate record stores the raw entered probability together with everything needed to make it self-contained: the seed β in effect at estimate time, the source and target marginals at estimate time, and what the estimator actually saw on screen. Interpretation happens at read time through a pure function. Estimates are append-only: a user re-estimating writes a new record, never an update, and the aggregation layer takes the latest per participant (§5). The clamps, the precision constant, the recency half-life, the aggregation itself can all be changed retroactively and replayed against the raw record. Nothing baked, nothing migrated.

4.4 What an estimate is, and what the display is not

An estimate is an **immutable historical data point**: the fact that at a given time, under a given baseline, a given participant judged the conditional to be a given value. That fact never changes.

The displayed current conditional is a different object. Suppose a participant estimated $P(B \mid A{=}\text{YES}) = 0.45$ when the target's baseline was 0.68, expressing $\beta \approx -0.95$. The baseline later drifts to 0.55 on its own. The system now displays $\sigma(\text{logit}(0.55) - 0.95) \approx 0.32$ for that participant's conditional. It is tempting to read this as "the participant's view updated from 45% to 32%." It did not. The participant expressed exactly one data point at one moment; they never said 32%. The 32% is the **system's fixed-β extrapolation**, produced under a specific modeling assumption—holding β constant as the baseline moves—and that assumption is a choice, not a truth. The same single estimate yields different current conditionals under different extrapolation rules: holding β fixed gives one number, holding the conditional probability fixed gives another, holding the probability-point shift fixed gives a third. One estimate does not determine which curve its author was on.

The copy discipline across the product follows from this: estimates **stay meaningful** as baselines move—they are never claimed to stay *correct*. A baseline that has drifted far from where an estimate was anchored is itself a staleness signal, and the honest responses are the ones the system actually implements: recency down-weighting in aggregation (§5) and solicitation of a fresh estimate, not harder extrapolation of the old one.

4.5 Operation-dependent invariants

The fixed-β extrapolation of §4.4 is one of two invariant choices in the system, and they are *different choices for different operations*:

Operation	What is held fixed	What is derived
Estimate auto-update (displaying a recorded estimate against a moved baseline)	the estimate's β	the displayed conditional
Marginal-drift propagation (a catalyst's probability moves without resolving; §6)	the edge's two conditionals $P(B \mid A{=}\text{YES})$, $P(B \mid A{=}\text{NO})$	the target's new marginal

For display, holding β fixed keeps a recorded relationship meaningful as the world moves. For propagation, holding the conditionals fixed makes the target's marginal an exact linear function of the catalyst's (the law of total probability; §6.1), with the clean properties that a zero move transmits zero and a full resolution transmits exactly the conditional. Neither choice is the full odds-ratio treatment of the underlying joint distribution; both are deliberate, documented simplifications in the same family. They coexist because they answer different questions—"what does this old estimate imply now?" versus "what does this hypothetical move do to the rest of the graph?"—and the system is explicit, per operation, about which question it is answering.

5. Consensus

Consensus is the crowd's value for an edge: a weighted aggregate of the interpreted β values of its estimates. Three properties define the implementation.

5.1 A pure function of the record

Consensus is computed on read—nothing is materialized—and is a pure function of the estimates and resolutions up to a timestamp. The consensus "as of" any past moment is reconstructable by replay. This is the same store-raw-derive-everything discipline as §4.3, applied at the aggregate level: when the weighting scheme changes (and it will; the current factors are starting defaults), history re-derives under the new scheme with no migration and no loss.

5.2 Counting rules and weights

Aggregation counts the **latest estimate per participant per seed epoch**. A participant can re-estimate freely (each submission is a new append-only record), but only their most recent estimate against the current epoch counts; and estimates made against a superseded epoch are preserved but not pooled—they were arguments against a different prior, and pooling them with arguments against the current prior would average across incommensurable baselines. Re-seeding is deliberately thresholded (≥ 0.1 in log-odds; §3.2) precisely because each new epoch resets this bucket.

Each counted estimate's weight is the product of three factors:

$$w = \underbrace{2^{-\Delta t/14\,\text{days}}}_{\text{recency}} \times \underbrace{\pi_m}_{\text{modality precision}} \times \underbrace{w_p}_{\text{participant weight}}$$

- **Recency** decays exponentially with a 14-day half-life, implementing the §4.4 principle that a moved world makes old estimates less current—the system down-weights rather than extrapolates harder.
- **Modality precision** π_m is the inverse-variance weight intrinsic to the input modality. The beta has a single input modality (the picker), so π_m is a single configured constant; the factor exists so future input modalities can carry their own precision without changing the aggregation's shape.
- **Participant weight** w_p is 1.0 for everyone in the beta. Anonymous participation means no track records exist yet (§8.5); earned weighting is a planned query-time change to this factor,

enabled by—and waiting on—accumulated resolution data.

The consensus β is the weighted mean of interpreted β values, computed in log-odds space and clamped to $[-6, +6]$; the displayed crowd conditional is $\sigma(\text{logit}(P(B)_{\text{current}}) + \beta_{\text{consensus}})$.

5.3 Dispersion, gated

Dispersion statistics—median, interquartile range, and 5th/95th percentile bounds—are computed over the counted β values **in log-odds space**, and reported only when the counted sample is at least $n = 20$; below the threshold the response carries the sample size and an explicit `sample_too_small` flag rather than statistics that would imply more stability than they have. Probability-space bounds, where surfaced, are a read-time transform of the β quantiles through the current baseline (apply $\sigma(\text{logit}(P(B)) + \cdot)$ to each quantile), not separately estimated objects. The threshold, like every other parameter, is a starting default.

5.4 Provenance: where edge values come from

An edge's value at any moment reflects a hybrid of origins, and the provenance model is designed to keep them distinguishable rather than blended into an unattributable number:

- **AI priors.** Every edge is seeded by an AI synthesis whose provenance records its inputs—contributing market quotes, news context—and its reasoning. In the beta, all seeds originate this way.
- **Crowd corrections.** Estimates accumulate against the seed from day one and aggregate per §5.2. The seed is the prior; the crowd is the correction; the epoch mechanism keeps each correction attached to the prior it argued against.
- **Market-implied conditionals.** Where joint or conditional pricing can be obtained from markets directly (for example, quoted combination pricing on catalyst–target pairs), it can enter as an alternative seed provenance with its own recorded basis. This channel is designed into the provenance model; it is not active in the beta.
- **Constraint-derived values.** Where event pairs are definitionally related (containment, mutual exclusivity, partition structure), the conditional is a matter of probability theory rather than judgment, and can be derived exactly. This channel likewise is designed-for rather than live.

How these origins combine is deliberately an implementation surface, not a fixed formula: the stable contracts are that every value carries its provenance, that crowd corrections are bucketed by the prior they corrected, and that the raw record supports re-aggregation under any future combination rule. The product surfaces the AI projection everywhere today; the crowd consensus surface activates as estimate volume justifies it, with the aggregation above already running underneath.

6. Propagation and scenarios

The Scenarios engine answers what-if questions: hold some events' probabilities at hypothetical values and compute the implied repricing of everything downstream. Its semantics are locked in shipped, unit-tested code; this section documents them exactly.

6.1 Single-shock transmission

A scenario is a set of **overrides**: events pinned at user-chosen probabilities. With no overrides, every event sits at its AI Estimate and the graph is silent—by construction, the target's projection equals its baseline. Recorded estimates do not move anything on their own; they shape *how* an override propagates (a catalyst at its baseline contributes zero shift regardless of which β the edge carries).

For a single edge $A \to B$ with effective shift β, when the catalyst is held at value v against baselines b_A, b_B, the engine holds the edge's two conditionals fixed (§4.5) and mixes them by the law of total probability:

$$P_{\text{yes}} = \sigma(\text{logit}(b_B) + \beta), \qquad P_{\text{no}} = \frac{b_B - P_{\text{yes}}\, b_A}{1 - b_A},$$

$$B\big|_v \;=\; P_{\text{yes}}\, v \;+\; P_{\text{no}}\,(1 - v).$$

P_{no} is the Bayes-coherent NO-direction conditional derived from the forward β and the baselines—nothing reverse-direction is separately estimated. The transmission has the two properties that make it trustworthy at the boundaries: at $v = b_A$ it returns b_B exactly (a non-move transmits nothing), and at the poles $v \in \{0, 1\}$ it returns exactly P_{no} or P_{yes} (a full resolution transmits exactly the conditional).

The **effective** β on an edge is the viewer's own derived β where they have estimated, else the AI seed. This is the product's amplification model: the engine is fully functional for a visitor who has never estimated, and each estimate sharpens propagation through that one edge.

6.2 Composition: multiple catalysts, multiple hops

Each override source propagates outward along the directed graph by depth-first traversal, computing at each downstream node the transmitted marginal along that path, and contributing a log-odds delta $\Delta = \text{logit}(\text{transmitted}) - \text{logit}(\text{baseline})$. A node's scenario marginal is then

$$m \;=\; \sigma\Big(\text{logit}(b) + \sum_S \Delta_S\Big),$$

the baseline shifted by the **sum of per-source deltas in log-odds space**. Additivity across sources is an explicit independence assumption—no interaction terms between catalysts—and is a deliberate simplification in the same family as the fixed-shift display assumption (§4.5). What two simultaneous shocks mean *jointly*, beyond the sum of what they mean severally, is not represented (§8.1).

Three rules complete the composition semantics:

- **Direct edges win, per source.** If an override source S has a direct edge to node N, that edge's (total-effect) β carries S's entire effect on N, and S's indirect contributions to N through intermediate nodes are dropped—S is never double-counted on N. Chain edges still drive their own intermediate projections. The deduplication is anchored at the shock source: when a shock originates *upstream* of a node that has both a direct and a mediated route to a target, both routes transmit (§8.2 discusses the overlap this admits).
- **Overrides pin.** An overridden node displays exactly the value the user set, and blocks transmission through itself: a shock cannot propagate through a node the user has pinned, and the UI grays the upstream paths this renders inert.
- **Cycle guards.** Traversal carries a per-path visited set. The launch graph is acyclic by curation, so the guard is presently a safety property rather than a semantics; it exists so that future graph shapes do not require re-architecting the engine.

6.3 Shocks are node-specific: a v1 modeling assumption

A scenario override answers a specific question: *if this event's probability were v, what does the graph imply?* The engine propagates the shock through **that node's edges only**. It does not model the override as the visible symptom of a common upstream cause that would *also* have moved other events directly.

The distinction matters. Setting Brent-above-$100 to certainty propagates through Brent's outbound dependencies (CPI, the Fed path). But most worlds in which Brent is actually at $100 are worlds where something—a supply shock, an escalation—is *also* bearing on other events through channels that do not pass through the Brent node. The engine deliberately does not invent that latent cause: the override means "Brent repriced," not "the world that reprices Brent." Users who want the common-cause scenario express it directly by overriding the cause—the ceasefire event—and letting the graph distribute it. This keeps scenario semantics precise and auditable at the cost of understating correlated moves when users shock a symptom rather than a cause; richer treatment of symptomatic relationships is future work (§9.4).

6.4 Attribution

Every node in a scenario carries a source-attribution state, read off the computed result: **user-defined** (explicitly overridden), **user-modeled** (not overridden, but its computed marginal differs from baseline because an upstream override propagated to it), or **AI fallback** (computed marginal equals the baseline—including the corner case where adjustments happen to land a node exactly on its baseline, which renders as fallback by design: the marker and the displayed delta are always mutually consistent). Scenario state is deliberately ephemeral: overrides travel by URL and are evaluated against *current* baselines at view time—a shared scenario is a fixed hypothesis against a live world. Recorded estimates persist; hypothetical marginals do not. The split is the persistence model of the product in one sentence: the conditional is your view of how two events relate (durable, recorded); the marginal is the world's live state (sandbox only).

7. Worked example: the December 2026 Fed anchor graph

The beta launches with a single-anchor graph: **21 events and 22 dependencies** organized around the anchor question *"Is the fed funds target range upper bound at or below 3.50% after the December 2026 FOMC?"*—operationally, does the Fed deliver at least one cut in 2026. The anchor sits on a live institutional disagreement: market pricing implies cuts beginning Q3/Q4, while a prominent dissenting view holds that energy-driven inflation makes the next move a hike.

> **A note on the numbers.** The probabilities and conditionals in this section are an *illustrative synthesis* prepared during launch composition. Live values are re-synthesized against fresh quotes on a regular cadence, so production values differ from the figures below. The numbers are internally coherent—every target marginal satisfies the law of total probability against its catalysts' values within rounding—which is what makes them suitable for tracing the machinery.

7.1 The level-path backbone as propagation rail

The graph's spine is a five-event series at the same (quantity, strike): the rate level $\leq$ 3.50% after each remaining 2026 FOMC meeting—June, July, September, October, December (call them F1–F5). Edges run series-consecutively: F1→F2→F3→F4→F5.

Two design decisions are visible here. First, **levels, not meeting actions**. "Fed cuts at the September meeting" is rejected as an event framing because its conditionals are sign-ambiguous—a June cut can raise or lower September-cut probability depending on the regime read. Level events are monotone everywhere: a cut having happened by June can only help the level being low in July. Second, **the series rule is the architecture** (§3.1). One instance per series per target forces the clean sequential chain, and long-range influence travels *through* the chain: June CPI affects December through June→July→...→December propagation, never via a redundant direct edge. A redundant direct edge would be worse than unnecessary—under direct-edges-win (§6.2) it would *suppress* the chain's information for shocks at that source.

The backbone's conditionals are deliberately extreme—$P(F_{k+1} \mid F_k\text{=YES})$ in the 95–97% range, a confirmation ratchet (once the first cut lands, reversal requires a hike). The backbone is the propagation rail, not the disagreement surface; disagreement is routed onto the inbound edges.

7.2 The inbound structure

Feeding the spine: a **CPI series** (core CPI m/m $\geq$ 0.3%, one instance per meeting slot; contested-magnitude edges—everyone agrees a hot print is hawkish, the dispute is size), a **payrolls series** (sub-+100k prints; confirmation edges), and a set of cross-domain catalysts into the anchor—a Q3 negative GDP advance print, a large-bank FDIC receivership (a tail catalyst *designed* to sit near 3%—an extreme value is not a stale value), an SPY drawdown threshold, and a government shutdown event whose edge into the October meeting carries a non-obvious contested sign: a shutdown delays the data releases the Fed needs to justify moving, so cut probability *falls* despite the growth hit.

The showcase chain is **geopolitics** $\to$ **oil** $\to$ **inflation** $\to$ **Fed**: a US–Iran ceasefire event (G1) feeding two Brent threshold events (B1, September; B2, December), with B1 feeding both August-core-CPI (C3) *and* the September Fed level (F3) directly, and B2 feeding the anchor (F5). The oil edges are the launch graph's contested-sign centerpiece: supply-shock inflation reads hawkish, growth shock reads dovish, and the seeded view—oil above $100 in December compresses cut probability from 75% to 55%—is exactly the kind of edge built to attract corrective estimates.

B1's double role exercises direct-edges-win live: when a user shocks B1 itself, its effect on F3 flows through the direct B1→F3 edge alone (the B1→C3→F3 route is dropped for that source—the direct edge carries B1's total effect), while C3's own projection still moves through B1→C3.

7.3 One estimate, end to end

Trace a single estimate through every layer of §§4–6, using the contested-sign edge B2→F5 (December oil into the December Fed anchor). Illustrative baselines: $P(\text{B2}) = 0.35$, $P(\text{F5}) = 0.68$; the AI seed holds $P(\text{F5} \mid \text{B2=YES}) = 0.55$.

Seed. The seeded shift is

$$\beta_{\text{seed}} = \text{logit}(0.55) - \text{logit}(0.68) = 0.201 - 0.754 = -0.553,$$

with the NO-direction conditional derived, not estimated: $P_{\text{no}} = (0.68 - 0.55 \times 0.35)/0.65 = 0.75$. Coherence check: $0.55 \times 0.35 + 0.75 \times 0.65 = 0.68$. ✓

Estimate. A participant holds a stronger hawkish view: oil still above $100 in December puts the cut at 45%, not 55%. They enter 0.45; the interpreter derives

$$\beta_{\text{user}} = \text{logit}(0.45) - \text{logit}(0.68) = -0.201 - 0.754 = -0.955,$$

and the record stores the raw 0.45 alongside the snapshot baselines (0.35, 0.68) and the seed β in effect—self-contained, append-only, interpreted on read.

Consensus. With this single estimate against the current epoch, the counted set has $n = 1$: consensus $\beta = -0.955$ (full weight—fresh, single modality, unit participant weight), and the response carries `sample_too_small` in place of dispersion statistics. As estimates accumulate, the weighted mean tightens around the crowd's view and dispersion unlocks at $n \geq 20$.

Propagation. In Scenarios, this participant pins B2 to 1.0 ("oil is above $100 on December 4"). Their own β wins the edge (effective β rule), so the anchor projects $\sigma(\text{logit}(0.68) - 0.955) = 0.45$—exactly their entered conditional, as the pole property of §6.1 guarantees—against 0.55 under the AI seed. A partial scenario, B2 pinned at 0.50 instead, mixes the conditionals: $0.45 \times 0.50 + 0.804 \times 0.50 \approx 0.63$ under the participant's β (the derived P_{no} being 0.804 for their steeper shift), versus ≈ 0.65 under the AI's. The anchor's tile attributes the result *user-modeled* with its delta against the AI baseline displayed inline.

A two-hop ripple. Shock the chain head instead: pin the ceasefire event G1 to 0 (collapse). The G1→B1 edge ($P(\text{B1} \mid \text{G1=YES}) = 0.22$, derived $P_{\text{no}} \approx 0.50$ against $P(\text{B1}) = 0.40$) transmits B1 from 0.40 to ≈ 0.50; B1's move then transmits onward through *both* its outbound edges—lifting C3 (August core CPI) by about 1.3 points and pressing F3 (the September level) down through the direct B1→F3 edge and the C3-mediated route, with the combined log-odds deltas summing per §6.2 to take F3 from 32% to roughly 30.5%—and the shock continues down the backbone rail toward the anchor. The depth-3 chain—geopolitics through oil through inflation into the rate path—is the multi-hop case the Scenarios engine is built to render.

8. Limitations

Every model is an approximation. This section states where Consequitur's approximations bind, where they degrade, and what the system does **not** know.

8.1 Independence in composition

Multi-catalyst composition is additive in log-odds with no interaction terms (§6.2). This is the standard log-linear escape from the intractable joint distribution, and it carries the standard cost: the model cannot represent a case where two catalysts together mean something different from the sum of what they mean apart. Hot CPI alone is hawkish; weak payrolls alone are dovish; hot CPI *and* weak payrolls together is the signature of stagflation, a regime in which the Fed may behave in a way neither single signal predicts. The composition rule adds the two shifts and moves on. Most catalyst pairs act close to independently, the fully general alternative is intractable (§2), and the architecture admits targeted interaction terms later for the specific pairs where the data shows they matter—but none of that changes what the beta's number means when two shocks are live at once: the sum of severally, not jointly.

8.2 Path overlap upstream of multi-path nodes

Direct-edges-win deduplicates a shock at its source (§6.2): shocking B1 never double-counts B1 on F3. The rule does not deduplicate *upstream* of the source: a shock at G1 reaches F3 both through B1's direct edge and through the B1→C3→F3 chain, and the two log-odds deltas add. To the extent the direct B1→F3 edge's total-effect β already internalizes the CPI channel, the chain route partially re-counts it for upstream shocks. At launch-graph depths and magnitudes the overlap is small, and curation mitigates it (the direct edge is seeded to carry channels *beyond* the mediated path), but it is an approximation inherent in per-source deduplication, in the same family as §8.1.

8.3 Model risk from stale edges

Any use of the graph for conditional repricing—"if this resolves YES today, my book reprices by X"—assumes the market, when the event actually resolves, moves to the edge's conditional. That is a model claim, not an arbitrage identity. An edge whose β was seeded or estimated under one regime can be stale under the next, and a repricing computed from a stale edge is precise nonsense.

The system's defenses are freshness (re-evaluation cadence with thresholded re-seeding), recency decay in consensus, and provenance that lets a user see *when* and *on what basis* a value arose—but the residual risk is irreducible: the graph encodes views about market behavior, and views can be wrong in ways no aggregation repairs.

8.4 Fixed-shift extrapolation degrades with baseline drift

The fixed-β display assumption (§4.4–4.5) is locally honest and globally fragile. The further a baseline drifts from where an estimate anchored it, the less the extrapolated conditional can claim about its author's view—the extrapolation rule, not the author, is generating the number. The failure is sharpest when the target's baseline moved *because the catalyst's own probability moved*: in that case much of the edge's shift is already absorbed into the baseline, and re-applying the full β on top of it over-claims. Coherence makes this precise—as a catalyst's probability approaches certainty, the conditional must converge to the marginal ($P(B \mid A{=}\text{YES}) \to P(B)$ as $P(A) \to 1$), so the *remaining* catalyst-attributable shift must shrink toward zero; a constant β never shrinks, and pushed far enough the implied NO-direction conditional exits $[0, 1]$ entirely. The window of maximum error is therefore the interval between a large correlated move and the next re-evaluation of the edge. The system's mitigations are the ones already described: down-weight by recency, flag rather than restate, solicit fresh estimates—and the re-evaluation cadence itself, which re-anchors edges against fresh marginals. What the system deliberately does not do is pick a "better" extrapolation curve and present it as the user's updated opinion; there isn't one—a single data point does not identify a curve (§4.4). A derivable decomposition that would attribute observed moves to their catalysts explicitly is described in §9.4; the beta's analytical focus is immediate usage against current baselines, not retrospective attribution.

8.5 Anonymous estimation, no track records

The beta's estimate flow is anonymous: no accounts, a stable per-browser session identity only. The consequences are stated plainly. There are no participant track records, so consensus weights every participant equally—the aggregation cannot yet distinguish a careful macro analyst from a passerby. Skill weighting is designed in as a query-time factor but is empty until resolution data accumulates, and that data accrues slowly: many catalysts resolve NO or not at all for long stretches. Abuse resistance in the beta is preventive and observational at the design level—invisible attestation on submission, single-use render-bound submission tokens, per-endpoint rate limits, and capture of integrity signals for flagging rather than blocking—and we do not detail operational specifics here. The long-run anti-manipulation mechanism is participant weighting backed by track records, not gatekeeping at the door; until then, consensus values should be read with the sample sizes the API attaches to them.

8.6 AI priors have known failure modes

Every edge in the beta is seeded by an AI synthesis (§5.4). The failure modes of LLM-derived probability estimates are documented and we do not claim immunity from any of them: miscalibration (particularly in tails), anchoring on stale or prominent context, narrative plausibility standing in for probability, and—most relevant to a graph—*correlated* errors across edges seeded by the same model with the same context, which violate exactly the independence that aggregation and composition lean on. The system's posture is that the AI value is a *prior*, labeled as such, with recorded inputs and reasoning open to inspection, and that the crowd's role is to correct it edge by edge. The beta launches with priors largely uncorrected; the correction layer is real but empty until participation fills it. Until then, the displayed conditionals are best read as "an auditable synthesis, awaiting correction"—which is what the product copy says.

8.7 Coverage, coherence, and cycles

The beta covers one anchor and 21 events in one domain. Conclusions about how the crowd behaves on the graph must wait for data, which is the stated purpose of a future revision of this document.

More broadly: an independently estimated graph is **generically incoherent**. The within-edge constraint (the law of total probability tying the two conditionals through the catalyst's marginal) is enforced by construction—the NO-direction conditional is derived, not separately estimated. Constraints beyond a single edge—agreement between a direct edge and its parallel chain, consistency around cycles, mutual consistency of crowd conditionals with market marginals, partition constraints over mutually exclusive event sets—are not enforced at estimate time and cannot be: estimators see one edge at a time. The launch posture treats detected inconsistency as *signal* (an inconsistency is variously crowd error, missing graph structure, or mutually mispriced markets—and the third is the analytically valuable case) rather than as a defect to smooth away at write time. Global reconciliation—projecting the estimated graph onto the nearest coherent joint distribution, confidence-weighted—is a designed direction, not a shipped behavior.

Finally, cycles. Propagation carries cycle guards, but the launch graph is acyclic by curation, and the deeper issue is representational: real event relationships include feedback and *symptomatic* structure—markets that do not cause a driver but reflect it, so their movement is evidence about it. The current model handles the evidential direction only through Bayes-derived reverse conditionals during propagation; it does not yet let a symptomatic market's movement update its driver as a first-class graph mechanic. We expect future revisions to need genuine support here, and the architecture reserves room for it rather than assuming acyclicity forever.

9. Roadmap

Everything in this section is future work. None of it ships in the beta, and descriptions here are design intent, not commitments of schedule.

9.1 The sensitivity surface

The largest roadmap direction is to present the graph as what it already is structurally: a sensitivity surface for a book of event-contract positions—the event-space counterpart of the Greeks book an options desk keeps. Two dimensions compose it.

The first is cross-event sensitivity, and it is the graph itself. Each edge's β is the sensitivity of a target's price to a catalyst's move; the gap it implies, $P(B \mid A{=}\text{YES}) - P(B \mid A{=}\text{NO})$, is the local slope $dP(B)/dP(A)$ (§6.1)—the role a position's first-order sensitivities play in an options book, *what moves this position and by how much*, expressed over events rather than an underlying price. The second dimension is time: **hazard-implied drift**, the per-day movement a dependent event's price is expected to absorb as a time-bounded catalyst decays toward its deadline without resolving, computed as the catalyst's expected decay rate transmitted through the edge. The standard machinery is a hazard model—an instantaneous resolution rate $\lambda(t)$ and its survival function—and the quantity answers what a time-aware desk asks reflexively: *what is the expected daily bleed from time alone?* Together, cross-event sensitivity and hazard drift fill, for a book of event-contract positions, the role a sensitivity book fills for an options position.

We are precise about what this is not. Binary event contracts do not admit the continuous-price machinery from which option sensitivities derive, and Consequitur does not compute option-style sensitivities on binaries; the parallel is to the functional role those measures play, not to the measures themselves, and we leave the quantities unnamed beyond that.

The decisive difference from an options book is the source of the numbers. Option sensitivities are read off a price process; Consequitur's edge sensitivities are the conditional views market participants explicitly express—how the crowd reads a catalyst's bearing on a target. Where an event has strong quantitative anchors, that expressed view supplements statistical ones. Where it does not—politics, policy, geopolitics, and other domains in which no price process or stable correlation exists to derive a sensitivity from—it is the only available source, and a sensitivity surface built from it has no closed-form competitor. The analog is deliberately softer than its options counterpart, but in anchor-less domains the softness is not a concession: it reflects that the underlying quantity is a matter of informed belief, not calculation.

None of this rests on a single layer. The sensitivities draw on the ranked provenance stack of §5.4—an AI prior as the always-available baseline and an analytical input in its own right; crowd corrections carrying the participant view; and, where exchanges quote joint or conditional positions, cross-exchange RFQs backing out market-implied conditionals, the most objective layer of all. The premise is that little usable structure exists here today, so Consequitur delivers these layers stacked, each tightening a sensitivity as its inputs mature. The open problems are real and are why this ships after the beta, not with it: estimating hazard shapes (constant, deadline-loaded, or lumpy around scheduled moments), separating expected drift from genuine news in observed paths, and accumulating the crowd and resolution data the upper layers depend on. Where listed term structure exists it can be read rather than assumed, and the (quantity, term, strike) family structure of the registry (§3.1) is the natural substrate.

9.2 Synthetic market discovery

The roadmap capability we regard as the genuinely novel productization. An event with no listed contract anywhere can still have an implied probability, provided it is connected through estimated edges to events that *do* have prices. The mechanism is the §6.1 law of total probability inverted: for an unlisted event X connected to a listed event A,

$$P(A) = P(X)\,P(A \mid X{=}\text{YES}) + (1 - P(X))\,P(A \mid X{=}\text{NO})$$

contains one unknown, and solves directly:

$$P(X) \;=\; \frac{P(A) - P(A \mid X{=}\text{NO})}{P(A \mid X{=}\text{YES}) - P(A \mid X{=}\text{NO})}.$$

Crucially, eliciting the edge never requires knowing $P(X)$—estimators answer "where does A go if X happens / doesn't," which is exactly the estimate flow that already exists. A single edge yields a fragile estimate; robustness comes from **density**. Each listed neighbor yields an independent implied value for $P(X)$, the set reconciles into a point estimate with a band—tight when the neighbors agree, wide when they do not—and the band is honest output, not apology.

The second half of the capability is constructive. The graph that implies $P(X)$ also implies the **replicating exposure**: a weighted combination of positions across X's listed neighbors whose aggregate payoff profile approximates a direct position in X. The implied value tells a user what the unlisted event is worth; the construction tells them how to *hold* it using contracts that exist, with the residual basis (the edges' uncertainty) stated alongside. Computing basket weights well—and being clear-eyed about when density is too thin to support them—is the substantive design work, and remains the unbuilt part of this capability. To be unambiguous: **no part of synthetic market discovery is live in the beta.**

9.3 The portfolio application layer

With positions linked (the registry already maps events to the venues that list them), the graph supports a portfolio surface: **conditional repricing of a book** (hold a catalyst at YES and reprice every linked position through the graph, with §8.3's model risk stated on the tin), a **sensitivity ladder** (rank the events to which a book is most exposed, including exposures that arrive only through multi-hop structure), and **cross-venue netting** (recognize when positions on different venues are, through the mapping layer, exposures to the same underlying event). This is the institutional shape of the product—sensitivity structure as modeling input, consumed the way correlation matrices are consumed—and it follows the same staging as everything else: feed-shaped, provenance-carrying, bands included.

These surfaces share a mechanic worth making concrete, because it is where the sensitivity structure becomes a trade. Hedging a single factor is one ratio: a book's exposure to a catalyst is its position size times the edge gap, and the offsetting position is that number, sign-flipped. A book long 100 of the December anchor that wants to neutralize December-oil risk reads the B2→F5 gap (-0.20 in the illustrative graph of §7), giving an exposure of -20 and a hedge of +20 oil contracts—and because the gap is constant under the model, the hedge is static, not continuously rebalanced. Hedging several factors at once is the standard linear solve a desk already runs: assemble the book's sensitivities into a matrix from the gaps and path products, then solve for the basket that zeroes the net exposure. What Consequitur supplies is the sensitivity matrix itself—and, where a factor is not directly tradable, the replicating basket of contracts that spans it (§9.2). The harder parts—interaction effects, the distinction between hedging drift and hedging a resolution jump, and confidence-weighting the sensitivities—are application-layer work, and the full treatment belongs in a practitioner note rather than here.

9.4 Nearer-term surfaces

Four items from earlier sections, gathered: the **crowd consensus surface** activates as estimate volume crosses usefulness thresholds (the aggregation already runs; §5); **additional provenance channels**—market-implied conditionals and constraint-derived values—activate as their inputs become available (§5.4); **symptomatic structure**—first-class support for evidence flowing from a reflective market back to its driver, beyond the current Bayes-derived reverse conditionals—remains the most significant representational extension on the horizon (§8.7); and **movement attribution**, the decomposition that closes §8.4's gap.

On the last: because every estimate snapshots both marginals at its anchor moment and every refresh records the quotes it consulted, an observed target move is decomposable against the graph. Propagate each direct catalyst's *realized* drift through its edge (the §6.1 transmission, applied to observed rather than hypothetical moves; direct inbound edges only, since a catalyst's own observed move already integrates its upstream causes); the sum is the move the graph *expected*, and the remainder is an unexplained residual. The residual then maintains the edge coherently—it shifts both conditionals as common evidence, while the catalyst-driven component re-weights between them—so the remaining catalyst-attributable potential shrinks as it should rather than being re-claimed in full. Two qualifications: the residual is model-relative (unexplained-by-the-graph conflates independent news with edge error, so the attribution is only as good as the edges), and the computation needs marginal *history* rather than the current snapshots the beta stores, which the storage model reserves room for. The decomposition earns its place twice over—as the principled trigger for re-seeds and fresh-estimate solicitation, and as feed-shaped output in its own right, the event-probability analog of the factor attribution institutional readers already consume. It is a designed direction, not a shipped behavior: the beta's focus is what the graph implies *now*, with recorded estimates serving as fresh inputs while recent, as the calibration archive permanently, and as flagged-when-stale views in between.

10. References

[1] R. Hanson. "Combinatorial Information Market Design." *Information Systems Frontiers* 5(1):107–

119, 2003.

[2] R. Hanson. "Logarithmic Market Scoring Rules for Modular Combinatorial Information Aggregation." *Journal of Prediction Markets* 1(1):3–15, 2007.

[3] R. Hanson. "Decision Markets." *IEEE Intelligent Systems* 14(3):16–19, 1999.

[4] Y. Chen, L. Fortnow, N. Lambert, D. M. Pennock, J. Wortman. "Complexity of Combinatorial Market Makers." *Proceedings of the 9th ACM Conference on Electronic Commerce (EC '08)*, 190–199, 2008.

[5] W. Sun, R. Hanson, K. B. Laskey, C. Twardy. "Probability and Asset Updating using Bayesian Networks for Combinatorial Prediction Markets." *Proceedings of the 28th Conference on Uncertainty in Artificial Intelligence (UAI 2012)*, 2012. arXiv:1210.4900.

[6] S. Dalen. "Toward Black–Scholes for Prediction Markets: A Unified Kernel and Market-Maker's Handbook." arXiv:2510.15205, 2025.

[7] V. Buterin. "From prediction markets to info finance." November 9, 2024.

[8] A. Agresti. *Categorical Data Analysis*, 3rd ed. Wiley, 2013.

[9] J.P. Morgan / Reuters. *RiskMetrics—Technical Document*, 4th ed. New York, 1996.