



WHITE PAPER · JULY 2026

The Real-Time Compliance Stack

Architecting CAT, CAIS, and 15c3-5 Controls for 2026

A technical blueprint for broker-dealers rebuilding compliance as a real-time system: one order state machine, event-sourced reporting, pre-trade controls in the order path, and the reconciliation model that survives an exam.

Executive summary

Compliance at a modern broker-dealer is no longer a reporting function that runs after the market closes. SEC Rule 15c3-5 requires risk controls inside the order path. FINRA's Rule 4210 interpretations push margin toward intraday evaluation. CAT and CAIS demand a complete, linkable event record with correction windows measured in days, not quarters. Each of these obligations assumes the same underlying capability: a firm that can observe, decide, and prove — in real time, from a single record of truth.

Most firms cannot. Their order flow, risk decisions, customer data, and regulatory reporting live in four systems integrated by file transfer. This paper describes the architecture that removes those seams, and the migration sequence that gets a firm there without pausing its business.

1. The seam problem

Batch integration creates three failure modes that examiners consistently find.

Divergence: the order record in the OMS and the reported event in the CAT submission are produced by different code paths, so they drift. **Blind windows:** risk state is refreshed on a schedule, so a firm's true exposure between refreshes is unknown.

Unprovable intent: when a control fires, the firm can show the outcome but not the inputs that produced it, because the evaluated snapshot was never persisted.

Every remediation letter that begins with “the firm was unable to reconstruct” is a seam problem, not a policy problem.

2. One order state machine

The architectural foundation is a single, authoritative order state machine. Every order, modification, cancellation, route, and fill is an immutable, sequenced event appended to one log. Order management, execution, risk, and reporting all read from that log rather than maintaining private copies of order state.

This has a direct compliance consequence: the CAT event and the OMS record cannot disagree, because they are projections of the same events. Reporting stops being a translation exercise and becomes a serialization of what already happened.

3. Pre-trade controls in the order path, not beside it

15c3-5 controls must be evaluated synchronously, on the order path, within a latency budget the trading desk can live with. Single-digit microseconds of added decision time is achievable when limits are held in memory and keyed by the identifiers the order already carries. Three design rules matter:

- **Evaluate against live state.** Credit, capital, position, and margin limits should be updated by the same event stream that drives execution, so a control never reasons about stale exposure.
- **Persist the decision, not just the verdict.** Store the limit values, the exposure snapshot, and the rule version alongside the accept or reject. This is what turns an annual CEO certification into a defensible one.
- **Version the rules.** Control logic changes; examinations look backward. Rule sets should be immutable and referenced by version from every decision record.

4. Intraday margin as an engineering problem

The FINRA 4210 interpretations settle a conceptual debate and leave a systems problem. Real-time margin requires position keeping that updates on fill, a pricing feed with defined staleness tolerances, a margin methodology engine that runs on demand rather than nightly, and a path from a breach to an action — alert, restrict, or block — that executes without human latency. Firms that treat margin as a report will keep discovering breaches after they matter.

5. CAT and CAIS: event-sourced by construction

Reporting quality is a function of where the data comes from. When CAT events are generated from the order event log, and CAIS records are generated from the golden customer and account record used at onboarding, the two datasets link cleanly and correction volume collapses.

The operating model that works has four properties: same-day error visibility, lineage from every reported field back to its source event, an explicit owner for each rejection class, and correction workflows that write back to the source rather than patching the submission.

6. Reconciliation that survives an exam

A defensible stack reconciles continuously across four axes: orders to executions, executions to clearing, reported events to internal events, and customer records to account records. Each reconciliation emits an exception, and each exception has an owner, an age, and a resolution note. Examiners rarely ask whether a firm had breaks; they ask how quickly the firm knew and what it did.

7. A staged migration

No firm rebuilds this in one release. The sequence that works:

1. Establish the event log alongside existing systems and dual-write.
2. Move reporting to read from the log, reconciling against legacy output until they match.
3. Move pre-trade controls into the order path behind a kill switch.
4. Retire the batch bridges.
5. Decommission the duplicate order state.

Each stage is independently valuable and independently reversible.

8. Evaluation checklist for vendors

- Is order state authoritative in one place, and can you show the event log?
- Are 15c3-5 controls synchronous, and what is the measured added latency?
- Do you persist the evaluated inputs behind every control decision?
- Are CAT and CAIS generated from order and onboarding data, or from an extract?
- What is your correction turnaround, and who owns rejections?
- Can you reconstruct any order's full decision history for a date two years ago, in one query?

How GATO implements this

The GATO Product Suite is built on the model described above. gHub and gTrader share one order state machine. gPrecision evaluates 15c3-5 and margin controls synchronously on that state. gNexus generates CAT and CAIS submissions from the same event log. gPure and gPortal keep the customer, account, and corporate-actions record aligned with what is reported. Firms can adopt a single module or the full stack, because they all read the same events.

To request a technical walkthrough or a gap assessment against this architecture, contact the GATO engagement team at sales@gatosystems.com.