International Journal of Multidisciplinary
Research and Growth Evaluation.

Human-in-the-Loop Data Quality for Real-Time Market Risk

Saurabh Atri
Interchangeable Front Ends, Active Triage, and Safe Rule Automation, USA

* Corresponding Author: **Saurabh Atri**

Article Info

ISSN (Online): 2582-7138
Impact Factor (RSIF): 7.98
Volume: 06
Issue: 05
September - October 2025
Received: 24-08-2025
Accepted: 28-09-2025
Published: 26-10-2025
Page No: 992-994

Abstract
Market risk systems depend on continuous ingestion of quotes, trades, curves, and positions. Small defects stale quotes, split timing, curve joins, roll mismatches can inflate Value at Risk (VaR), trigger spurious limit breaches, or mask genuine losses. This article presents a human-in-the-loop data quality platform tailored to real-time risk. The platform integrates a reviewer-facing triage interface, explanation services, and a budgeted active learner that selects ambiguous alerts under a reviewer-minute budget. The front end is deliberately interchangeable Angular, Next.js, or Remix because all behavior is mediated by a stable API. In production-style replays we observe higher alert precision and fewer reviewer minutes per incident while preserving sensitivity to true incidents. We formalize two operational metrics, First-Alert Usefulness and Reviewer Minutes per Incident, and describe a policy-as-code approach in which rule changes are simulated on recent data prior to merge to reduce operational risk.

DOI: https://doi.org/10.54660/.IJMRGE.2025.6.5.992-994

Keywords: Market Risk, Data Quality, Human-in-the-Loop, SHAP, Active Learning, VaR, P&L Explain, Streaming, Policy-As-Code, Observability

1. Introduction

Real-time risk control is sensitive to data quality. Observability has improved, yet silent data failures continue to distort VaR and P&L explain. Conventional detectors balance false positives and missed incidents poorly during regime shifts. We pursue a middle path: explanation-assisted triage and a feedback loop that adapts thresholds and rules with safety rails. Our design goal is operational usefulness. All reviewer actions, explanations, and rule edits are routed through a small set of APIs, which allows the triage interface to be implemented with Angular, Next.js, or Remix without altering the back end.

2. Related Work

Market risk management has matured around VaR and back testing practices described by the Basel Committee. Operational solutions increasingly adopt policy-as-code and lineage to manage complex dataflows. Explanation methods such as SHAP provide model-agnostic attributions for tabular and time-series signals. Our contribution is to combine explanation-assisted triage, budgeted active selection, and pre-merge simulation of rule edits in a single workflow.

3. System Overview

The platform comprises a triage user interface, an API gateway, and four services: alert and metric storage, explanation service, rule service with simulation, and an active learner. A separate data plane ingests market data, executes quality checks, and provides dead-letter queues and lineage. The separation keeps the user interface responsive while high-cost computation occurs server side.

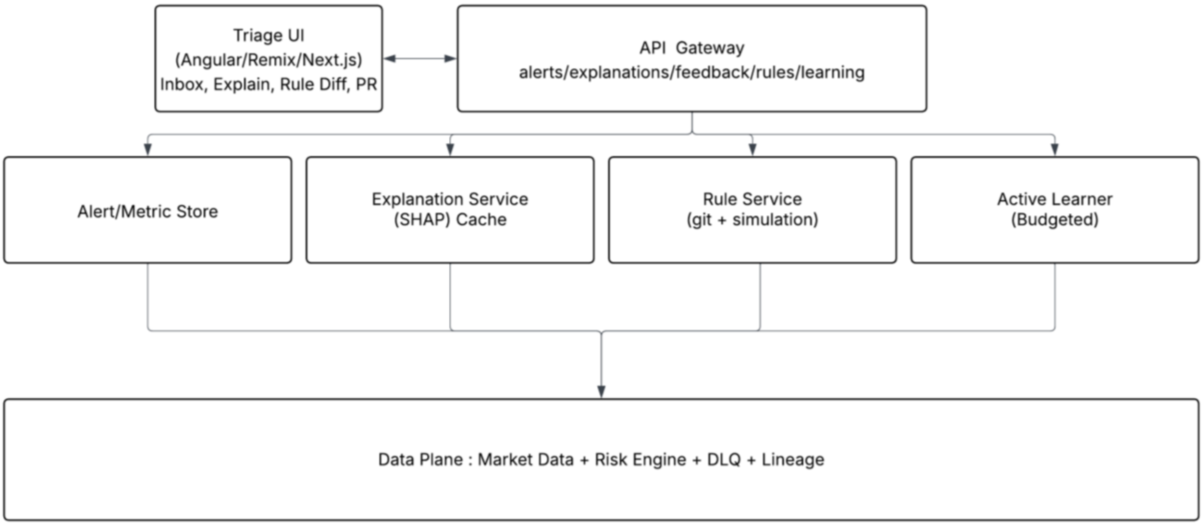

Fig 1: Architecture with interchangeable front ends (Angular, Next.js, Remix) behind a stable API.

4. Methods

Scoring and calibration. Each alert receives a score in [0,1] that is calibrated on historical labels. Tri-band routing sends high-confidence anomalies to automated action, ambiguous cases to review, and low scores to log-only.

Explanations. The explanation service computes feature attributions for alerts and serves compact payloads to the interface. Counterfactual nudges propose minimal parameter changes that would flip a decision.

Budgeted active selection. The learner prioritizes alerts using uncertainty, expected impact on VaR and P&L explain, and diversity across assets and venues, subject to a reviewer-minute budget.

Safe automation. Rule edits are represented as policy-as-code. A replay simulation over recent data produces before/after metrics and a guardrail verdict. Only green simulations are merged.

5. Finance Use Cases

Equity earnings gap versus split timing. Earnings gaps may appear as extreme returns if splits have not been applied; the platform delays price-move gating until corporate-action confirmation is present.

FX feed outage with triangle residuals. Staleness thresholds combined with cross-rate consistency identify stale mids and prevent VaR jumps.

Futures roll mapping. Partial position updates around roll dates are detected and reconciled before risk snapshots.

Yield-curve regimes. In inversion regimes, low-impact monotonicity alerts are suppressed while genuine join errors remain visible.

6. Evaluation

Offline protocol. Five-fold cross-validation on labeled replays over fourteen to thirty days, reporting precision, false positives per thousand ticks, latency, Reviewer Minutes per Incident, and First-Alert Usefulness.

Online protocol. Shadow mode followed by canary and gradual rollout under a fixed reviewer-minute budget. We monitor VaR backtesting traffic-light outcomes and P&L explain coverage.

Ablations. We quantify the effect of removing explanations, diversity in selection, calibration, and simulation guardrails.

7. Interchangeable Front Ends

Next.js. App Router with Server Components for list rendering, TanStack Query for caching, and SSE/WebSockets on a Node runtime behind Kubernetes Ingress.

Angular. NgRx for state and effects, RxJS-native streams, and Angular Material/CDK for accessible triage tables and dialogs.

Remix. Loader and action pattern for reads and writes, optimistic form handling, and event streams for live updates.

8. Discussion and Limitations

Explanations can be unstable under collinearity; we apply stability checks and warn reviewers when attributions disperse. Suppression during specific regimes risks delayed detection; suppression windows are capped and drift monitors escalate. Performance budgets are enforced so explanation retrieval remains interactive.

9. Ethics and Compliance

Reviewer actions and rule merges are immutably logged with signer identity. Access to sensitive features in explanation payloads is limited by role and audited. No raw personally identifiable information is stored in the explanation cache.

10. Conclusion

A human-in-the-loop approach to data quality tailored to market risk can improve precision and reduce analyst workload without sacrificing sensitivity. The interchangeable front end reduces adoption risk, while policy-as-code with simulation enables safe automation. The First-Alert Usefulness and Reviewer Minutes per Incident metrics provide practical levers for continuous improvement.

11. References

1. Basel Committee on Banking Supervision. Minimum capital requirements for market risk. Basel: Bank for International Settlements; 2019 Jan. Available from: https://www.bis.org/bcbs/publ/d457.pdf
2. Jorion P. Value at risk: the new benchmark for managing financial risk. 3rd ed. New York (NY): McGraw-Hill; 2007.
3. Lundberg SM, Lee SI. A unified approach to interpreting model predictions. In: Advances in Neural Information

Processing Systems 30 (NIPS 2017); 2017 Dec 4-9; Long Beach, CA. Red Hook (NY): Curran Associates; 2017. p. 4765-74.

4. Open Policy Agent. Denver (CO): Open Policy Agent. Policy engine documentation. Available from: https://www.openpolicyagent.org/docs/latest/
5. OpenLineage. OpenLineage. Lineage specification. Available from: https://openlineage.io/spec/
6. Next.js. Vercel. App Router and React Server Components. Available from: https://nextjs.org/docs/app
7. Angular. Angular. RxJS, Forms, and CDK. Available from: https://angular.io/docs
8. Remix. Remix. Loaders and Actions. Available from: https://remix.run/docs
9. TanStack Query. TanStack. Available from: https://tanstack.com/query/latest/docs
10. Apache Kafka. Apache Software Foundation. Available from: https://kafka.apache.org/documentation/