VOLT ACTIVE DATA

INDUSTRY BRIEF

Real-Time Fraud Prevention

Volt Active Data | BFSI & FinTech Solution Brief

Executive Summary

This brief examines real-time fraud decisioning architecture for banks, FinTechs, and payment providers operating at transaction scale. It covers the structural limitations of current fraud platforms, the specific data-plane requirements imposed by sub-50ms decisioning, and how Volt's in-database execution model addresses them.

Who should read this:

Systems engineers, solutions architects, and fraud technology leaders responsible for fraud detection and prevention infrastructure. Engineering and product leads evaluating the feasibility of autonomous fraud controls are also addressed in the Agentic AI section.

Related topics addressed in context:

Real-time payment rail compliance (AML, sanctions screening, APP fraud reimbursement mandates) and payment gateway infrastructure, as they intersect with fraud decisioning. Dedicated briefs on both topics are referenced at the end of this document.

The Fraud Decision Happens Inside a Window that Most Architectures Cannot Reach

FinTech fraud looks different from bank fraud. Velocity is higher, synthetic-identity attacks are more common, authorized push-payment scams are rising sharply, and customers expect the same experience whether a transaction is approved or declined. The institution's own scoring contribution typically needs to complete in under 50ms, and that budget covers everything: feature lookup, rule execution, model scoring, and the decision record.

Getting it wrong costs in both directions. A missed fraud event on a real-time rail cannot be recalled once funds clear. A false decline is a churn event, and in FinTech the correlation between false-decline rate and customer lifetime value is direct and measurable. Institutions that instrument this relationship typically find that false declines cost more in lifetime revenue than the fraud they were designed to prevent.

Non-card flows compound the problem. A fraud platform that works for card authorization also has to handle instant P2P transfers, account opening, withdrawal requests, and cross-product event correlation, all from a single consistent state. Most platforms were not built with that scope in mind, and the operational cost of maintaining separate fraud stacks for each flow type is substantial.

Where the Current Architecture Falls Short

Most FinTechs run fraud on a combination of vendor rules engines (Sift, Forter, Sardine, Unit21) and their own ML models. The orchestration layer connecting those components requires a state plane that holds velocity counters, device fingerprints, behavioral history, and a live transaction graph, served at the latency that the authorization window demands.

Streaming platforms built on Flink or Kafka Streams can compute event-time features well. They are not well-suited to serving sub-50ms point queries with strict consistency, and they were not designed to be the system of record for fraud decisions. Using them that way creates a category of bug that is hard to reproduce in testing and expensive in production: the fraud feature is technically up to date, but the consistency guarantee is not strong enough to prevent a race condition on a high-velocity attack.

Redis is a common alternative for the feature-serving path. It is fast, but it carries no real ACID (Atomicity, Consistency, Isolation, and Durability) guarantees, no SQL, and insufficient durability for a fraud system of record. The architecture being replaced in most Volt fraud conversations is Redis plus a rules engine plus something else holding the authoritative state. Each additional component adds operational overhead, a failure domain, and a consistency boundary that fraud teams then have to reason about.

Per-event vendor pricing creates a ceiling problem at scale. A FinTech that routes 10 million transactions per day through Sift or Forter is paying a cost structure that does not compress as volume grows. The per-transaction economics that make sense at 500k events per day become the largest line item in the fraud budget by the time the platform reaches maturity. Several of the larger FinTechs have reached inflection points where vendor fraud-tool spend exceeds the cost of in-house infrastructure by a factor of three or more.

Rules versioning is operationally painful across all of these approaches. Fraud patterns change faster than weekly deployment cadence, but most teams cannot safely change a rule under live load. The staging window required to test and deploy a new rule is the window in which an active attack can continue uncontested. During a coordinated synthetic-identity campaign, a week-long deployment cycle is not an operational inconvenience. It is a material loss event.

How Volt Addresses It

Volt holds the entire fraud state plane in a single ACID store: velocity counters per device, per user, per merchant, and per BIN; device fingerprint history; behavioral baselines; blocklists and allowlists; the decision audit trail. Feature lookup, rule execution, and decision recording all happen inside the same system, in a single round trip, without network hops between components.

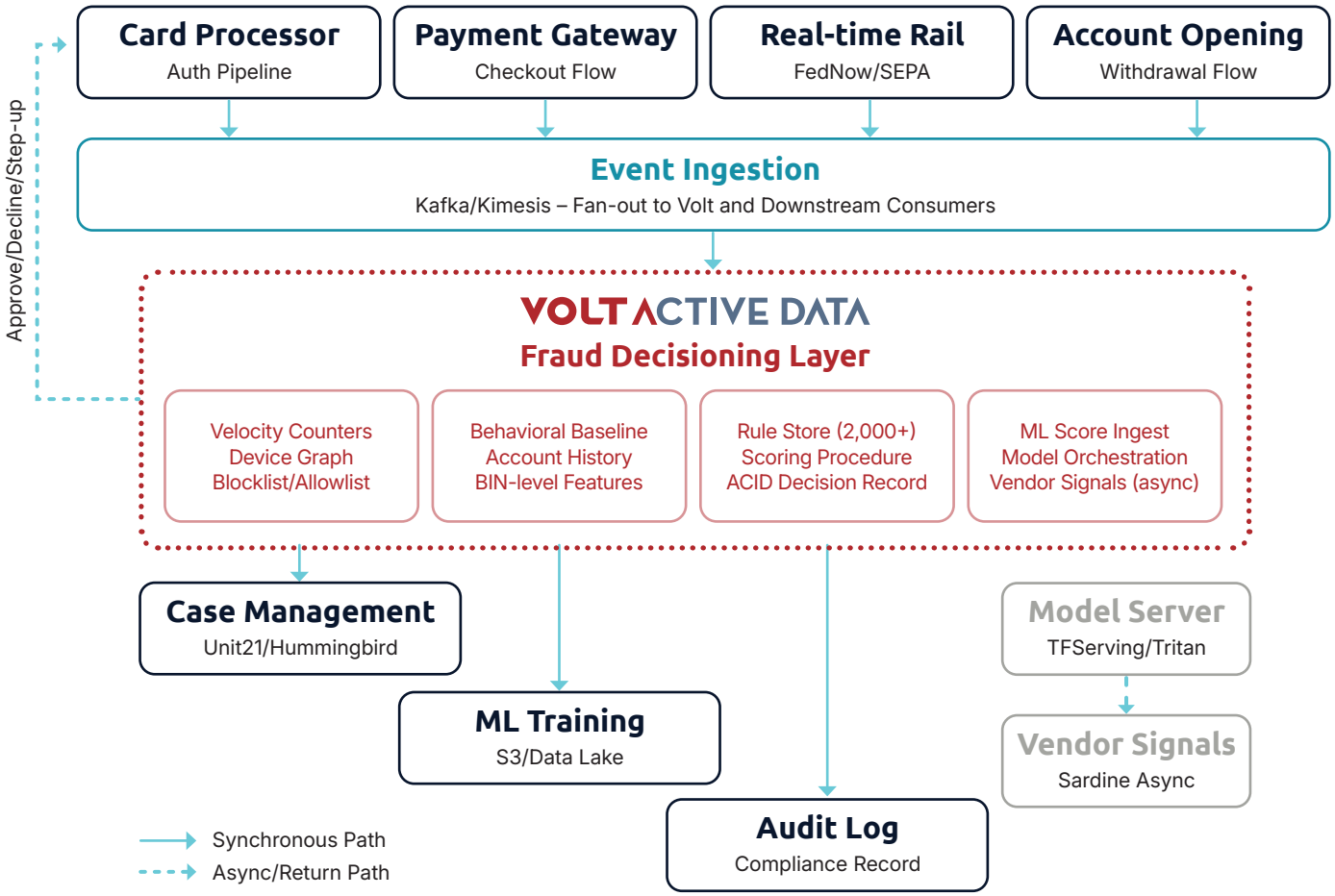
The scoring procedure runs inside the database against the in-memory feature set. At a Tier-1 bank running over 2,000 production rules, the full execution completes inside the 50ms authorization budget consistently at production throughput. The latency does not degrade as transaction volume increases, because the architecture scales linearly through shared-nothing partitioning rather than through adding coordination overhead.

Rule deployment works differently here than in any of the alternatives above. Changes to fraud logic deploy atomically, under live load, with no maintenance window and no throughput penalty. A fraud team that identifies a new attack pattern at 9am can have a rule in production by 9:15am. The deployment is transactional: either the new rule is fully live or it is not. There is no partial-deploy state in which some transactions see the old logic and others see the new.

ACID guarantees matter specifically in real-time rail flows where two near-simultaneous transactions can share a limit or exploit an account state that has not yet propagated across components. In an eventually consistent system, that window is a paid-out fraud event. In Volt, the balance decrement and the authorization decision are part of the same transaction.

Vendor signals from Sift, Sardine, Iovation, and Socure can be ingested asynchronously and stored as additional features. ML model scores from TFServing, Triton, or SageMaker endpoints feed into the decision procedure. Volt orchestrates the call, the model returns a score, and Volt finalizes the decision. The vendor and model integrations add intelligence without taking over the decision path.

Data Architecture



The authorization edge (card processor, gateway, account opening flow, withdrawal, real-time rail interface) sends events to Volt via Kafka or Kinesis. Volt serves as the operational fraud plane, holding the feature store, rule store, velocity counters, device graph, and decision log. The ML training pipeline reads from Volt through Kafka to S3 or a data lake, with model training running in SageMaker, Vertex, or an in-house cluster. Case management tools (Unit21, Hummingbird, or in-house) read from Volt directly. The existing event bus remains in place; Volt becomes the system of record for fraud state without displacing the streaming infrastructure already in use.

Outcomes

Fraud loss reduction of 50-85% depending on baseline. At a Tier-1 bank with 2,000+ rules, the reduction reached 83%. The range reflects different starting architectures; institutions running the weakest baselines tend to see the largest improvements.

False positive reduction of 30-50%. Richer per-transaction state means more context for each decision. The reduction in false declines is frequently the largest dollar-value lever in FinTech fraud, because of its direct relationship to customer retention.

P99 scoring latency under 50ms, consistently at over 10,000 TPS. The latency guarantee holds at higher throughput because the architecture does not introduce coordination overhead as it scales.

Rule deployment cadence from weekly to multiple times per day. The operational constraint on fraud rule changes is removed. Teams respond to emerging attack patterns on the same day they are identified.

Vendor fraud-tool spend reduced by 40-60%. Moving in-house workloads off per-event-priced vendors becomes viable once the state plane can serve them at the required latency.

Business Value

Fraud technology decisions are engineering decisions with direct P&L consequences. The business case for investing in the decisioning layer is not primarily about technology. It is about four specific financial exposures.

Direct fraud losses.

The gap between detection and decision is where losses are paid out. An 83% reduction in fraud losses at a Tier-1 bank operating at scale represents a nine-figure annual improvement. For a FinTech processing $5 billion per year in payment volume with a 0.1% fraud rate, closing that gap by 80% recovers $4 million annually in direct losses alone.

Customer lifetime value.

False declines are quiet revenue destruction. A customer who experiences a false decline at checkout has a materially higher churn probability in the 30 days that follow. FinTechs that have modeled this relationship typically find that a 10-percentage-point improvement in false decline rate increases net revenue retention by 2-4 points. At scale, that exceeds the total cost of the fraud infrastructure investment within 18 months.

Regulatory exposure.

UK PSR APP-fraud reimbursement mandates and equivalent regulations emerging across the EU and US create a new category of financial liability: failure to demonstrate that fraud controls were operating correctly at the moment of a specific transaction. This is not a technology audit requirement. It is a legal evidentiary standard. Institutions without a point-in-time ACID record of their fraud decisions carry exposure that is difficult to quantify until the first regulatory enforcement action.

Operational cost.

The hidden cost of the current multi-vendor, multi-component fraud architecture is the engineering time consumed by it. Vendor integrations, rule deployment cycles, state reconciliation, and incident investigation collectively absorb a disproportionate share of senior engineering capacity. Consolidating onto a single data plane with atomic rule deployment typically frees 30-40% of the fraud engineering team's time for work that builds competitive advantage rather than maintaining operational stability.

Agentic AI

The direction several fraud vendors are moving involves autonomous agents that deploy rule changes, adjust thresholds, and trigger step-up authentication without human review in the loop. Sardine and Resistant.ai are working in this space already. The operational appeal is a system that can react to a new attack pattern in seconds rather than hours.

The risk is equally concrete: an agent acting on a stale feature value can reject tens of thousands of legitimate transactions before anyone notices something has gone wrong. The problem is not the agent's reasoning capability. The problem is the data it is reading from. An autonomous fraud agent running against an eventually consistent state store carries systemic risk regardless of how well the model performs.

Volt's role in an agentic fraud architecture is to be the consistent feature store the agent reads from and the rule-deployment surface that accepts its writes. When the agent calls a stored procedure to deploy a new rule, either the rule is fully committed or it is not. When it reads a velocity counter, the value reflects actual current state. The agent's safety property in production is a data-plane property, not a model property.

Related Reading

Real-Time Rails: FedNow, SEPA Instant, RTP, Pix, UPI Covers the intersection of fraud, AML, and sanctions screening inside instant payment rails, including the regulatory evidentiary requirements created by APP-fraud reimbursement mandates. Available from voltactivedata.com/bfsi.

Payment Gateway Infrastructure Covers acquirer routing, idempotency, and settlement state management at gateway scale, including the fraud and compliance controls that sit within the gateway's processing budget. Available from voltactivedata.com/bfsi.

Talk To Us

If you are working through a fraud architecture decision, we are happy to walk through how Volt fits alongside your existing Kafka or Kinesis infrastructure, what a migration from a vendor rules engine looks like operationally, and where the economics cross over at your transaction volume.

voltactivedata.com/company/contact