

2026-2027

Best practices for architecting quantitative trading systems

Integrating real-time data streaming and agentic
AI for modern financial services applications



What's in this handbook

Introduction	3
Trading infrastructure in the era of AI	4
The challenges holding firms back	4
Financial services architectural use cases	5
Agentic AI use cases	6
Real-world stories	7
New York Stock Exchange: real-time market data in the cloud	7
Jump Trading: a next-generation messaging platform for global markets	8
Alpaca: 100x faster order processing for high-volume algorithmic trading	9
StoneX: lower latency, lower cost, and new client products	10
Zafin: a reliable system of reference for the world's leading banks	11
Why trading firms choose Redpanda	12
Redpanda Data Platform	12
Redpanda Agentic Data Plane	12
Reference architecture	14
Get started with Redpanda	15
Chat with an expert	15
Further reading	15

Trading firms, exchanges, and brokerage platforms use Redpanda Data Platform and Redpanda Agentic Data Plane to move market data, orders, and AI agents at the speed of the market. This guide shares the best practices from these real-world use cases.

Introduction

In quantitative trading, the edge belongs to the firm whose data moves fastest and most reliably. Market data feeds, order management systems, risk engines, and research pipelines all depend on a continuous flow of events, and every millisecond of delay or every dropped message shows up as slippage, missed fills, or decaying alpha.

Two shifts are now raising the stakes. Strategies increasingly rely on machine learning models that retrain on live market data, and AI agents are moving from demos into research, surveillance, risk, and operations workflows. Both depend on the same foundation: real-time data infrastructure that stays predictable under the most volatile market conditions, and a governed way to give AI/ML applications access to that data.

This guide covers:

- Trends and challenges facing quantitative trading infrastructure in the era of AI
- Where real-time data streaming and agentic AI create value across the trading lifecycle
- How NYSE, Jump Trading, Alpaca, StoneX, and Zafin run mission-critical financial workloads on Redpanda
- How Redpanda Data Platform and Redpanda Agentic Data Plane fit together, and how to get started

Trading infrastructure in the era of AI

The trends reshaping quantitative trading

Market data volumes keep climbing, and they don't climb steadily. Firms ingest feeds from dozens or hundreds of exchanges, electronic communication networks (ECNs), and vendors, each in its own format. Traffic is stochastic and leap orders of magnitude in seconds: a market open, a central bank announcement, even a single social media post. NYSE alone handles nearly 1.1 trillion records a day across its markets.

That volume is also changing what "using" the data means. Quant teams increasingly train and retrain models on live market data rather than end-of-day snapshots, which means feature pipelines, prediction models, and hedging systems all need fresh data at consistent latency, plus historical data close at hand for backtesting and replay.

AI agents are showing up in the middle of that shift. Research assistants query market and trade data, surveillance agents triage alerts, operations agents investigate incidents, and all three need to move quickly from experiments into production. Each one needs access to sensitive data and systems, which raises the same questions every time: whose identity, what permissions, what it costs, and whether it can be audited.

None of this runs in one place. Latency-sensitive components stay close to the exchange venues, while research, analytics, and client-facing data products move to the cloud, so infrastructure now spans colocation, private data centers, and multiple clouds at once. Teams need one platform that behaves the same way in all of them.

And, at the end of it all, sits a regulator who expects a straight answer. Firms must be able to

reconstruct every order, risk check, and model-driven decision, and as AI takes on more of that work, "what happened" isn't enough anymore. Firms need to show which agent acted, on whose behalf, and with what data.

The challenges holding firms back

Tail latency at exactly the wrong moment. Legacy messaging infrastructure runs on the Java Virtual Machine (JVM). Garbage-collection pauses and JVM behavior under bursty load create latency spikes during high-volatility periods, the moments when predictable performance matters most.

Using Apache Kafka®, written in Java, platform engineers spend their time keeping the cluster healthy, tuning, and rebalancing instead of building what trading and research teams actually need. Teams end up with **operational overhead that grows with scale**.

Every additional broker, storage tier, and support contract cuts into the economics of the next strategy or data product. **Infrastructure costs rise with data volume**, often faster than the business case that justified them.

Lastly, live streams often sit in one system, historical tick data in another, and reference data somewhere else, so every new use case means another custom integration, another ETL pipeline, and another point of failure. Fragmentation and a missing governance layer compound each other: without a common control layer, teams lack an inventory of existing agents, agents borrow human credentials, spend is hard to attribute or cap, and when something goes wrong, there is no single audit trail.

Financial services architectural use cases

High-frequency trading (HFT) and ultra-low-latency (ULL) systems operate at single-digit microsecond latencies, or with tick-to-trade hardware running at submicrosecond speeds. Traditionally, batch data pipelines can take minutes or hours to catch up. This creates data gaps and risks as the market operates increasingly far ahead of analytical systems.

Real-time data streaming technologies minimize the freshness delays of traditional batch processing, allowing the analytical domain to get data in milliseconds after trades go through. The use cases below show where real-time data streaming shines.

Market data ingestion and normalization (fan-in)

Ingest feeds from exchanges, ECNs, and data vendors in different formats, then normalize them into a consistent model downstream systems can consume without custom adapters for each source.

Low-latency market data distribution (fan-out)

Deliver the same normalized stream to trading, pricing, risk, dashboards, and quant research systems simultaneously. Exchanges and data providers can also deliver feeds directly to customers' cloud applications, as NYSE does with NYSE Cloud Streaming.

Order and execution data flows

Order management and execution management systems depend on a continuous, ordered stream of events: new orders, amendments, cancels, partial fills, and rejects. A durable, low-latency streaming backbone keeps that flow sequential and recoverable, and supports AI-assisted smart order routing and algorithmic execution logic that decides where and how to slice orders.

Positions, P&L, and intraday risk

Stream fills and market data into positions and P&L engines for an up-to-date view of exposure throughout the day, rather than waiting for end-of-day reconciliation. Risk checks and limit monitoring run on the same live data.

Real-time analytics, anomaly detection, and market sentiment

Detect asset and market shifts as they happen, flag anomalies in prices or flows, and feed sentiment signals into strategy and risk systems.

Machine learning features and model training

Feed live data into feature pipelines and prediction models, including predictive hedging systems, and move model training from batch to streaming so models reflect current market conditions.

Telemetry, logs, and observability

Trading infrastructure generates terabytes of logs and metrics a day. Streaming that telemetry in real time lets platform teams spot latency regressions, failing components, and slow consumers before they affect a trading session.

Historical tick capture and replay

Retain market and order data cost-effectively in object storage so teams can replay a session to reconstruct an incident, investigate a disputed fill, or backtest a strategy against the exact sequence of events.

Streaming into the lakehouse

Land streams directly as Apache Iceberg™ tables so research and analytics teams can query current and historical data with standard SQL, without a separate ETL pipeline to maintain.

Agentic AI use cases

AI agents are most useful in quantitative trading, where people spend time gathering context from many systems before making a judgment. In each use case below, the agent needs fresh data, scoped access, and a complete record of what it did.

Quant research assistants

An agent answers a researcher's question by querying live topics and historical Iceberg tables together, testing several hypotheses in parallel, and summarizing the results with the underlying queries attached. It runs under the researcher's own permissions and within a set token budget.

Trade surveillance and compliance triage

An agent reviews surveillance alerts, pulls the relevant orders, executions, and market data around each event, and drafts a case summary for a compliance analyst. The system logs every data access and tool call so regulators can reconstruct the review.

Risk and exposure monitoring

An agent watches intraday positions and limits, explains why an exposure moved, and escalates to a risk manager with the supporting data. Guardrails restrict it to read-only access and prevent it from acting on orders.

Incident investigation and operations

When a feed stalls or latency spikes, an agent correlates telemetry, logs, and recent changes to propose a likely cause, shortening mean time to resolution for platform and SRE teams.

Post-trade reconciliation and exception handling

An agent matches trades across internal records, counterparties, and clearing data as events arrive, resolves routine breaks, and routes the rest to operations staff with context, instead of waiting for end-of-day batch processes.

Regulatory and client reporting

An agent assembles data for recurring reports, drafts narrative sections, and flags gaps for human review, with a transcript of every step available for audit.

Whether an agent program reaches production usually depends less on the model than on whether agents can safely access real enterprise data. That means per-user identity, policy enforcement, cost controls, and a replayable record of their actions.

Later in this guide, you'll learn how the Redpanda Agentic Data Plane provides that layer.

Real-world stories

Leading exchanges, proprietary trading firms, brokerage platforms, and financial technology providers have re-architected their mission-critical systems with real-time data delivery and agentic AI. Here are a few that rely on Redpanda to succeed.

New York Stock Exchange: real-time market data in the cloud

~1.1 trillion

records handled
daily by NYSE

Sub-100 ms

market data delivery
to cloud customers

Hours or days

fractional orders in
under a minute

The New York Stock Exchange is the world's largest stock exchange by total market capitalization of its listed securities. A significant part of its business is selling real-time and historical market data to hedge funds, high-frequency traders, and retail and institutional investors.

Challenge

NYSE wanted to extend its multi-year cloud journey by offering real-time market data on AWS. Consuming real-time data directly from NYSE had required customers to build and maintain on-premises environments or work through third-party providers. The Kafka variants NYSE evaluated didn't handle bursty traffic well because their Java foundations affected latency profiles.

Solution

NYSE selected Redpanda Self-Managed (Enterprise) on AWS to power NYSE Cloud Streaming, a data distribution platform that lets customers access real-time market data with sub-hundred-millisecond latency in the cloud. Redpanda's C++ implementation of the Kafka protocol suits unpredictable workloads and scales to meet changing demand.

Results

Redpanda provides lower latency than the alternatives NYSE evaluated for consuming data feeds in AWS, with better control over memory and CPU usage and no JVM or garbage collection. Redpanda Console saved NYSE from building its own monitoring front end, and tiered storage keeps data in object storage. Most importantly for the business, NYSE can now onboard a new data customer within hours or days.

`"Redpanda is the only Kafka-compatible data streaming platform that can cost-effectively and reliably manage the massive scale of the NYSE. When milliseconds matter, Redpanda delivers real-time market data so that our customers can make important investment decisions and manage risk."`



Vinil Bhandari Head of Cloud & Full Stack Engineering, NYSE

Jump Trading: a next-generation messaging platform for global markets

Billions
of messages
processed daily

50 ms P95 / 150 ms P99
latency on the
telemetry pipeline

Zero
data loss

Jump Trading is a proprietary trading firm specializing in algorithmic and high-frequency trading. Its success depends on real-time insights from data-intensive, mission-critical workflows.

Challenge

Over 20 years of evaluating messaging systems for log and telemetry use cases, Jump Trading found each platform lacking in some combination of consistent tail latency, tolerance of slow consumers, reliable delivery, cost-effectiveness, and access to low-level metrics. As the firm grew to millions of messages per second across thousands of nodes, it needed a platform that could scale with its real-time trading infrastructure, and it was hesitant to put a Java application in such a mission-critical role.

Solution

Jump Trading chose Redpanda for its Kafka-compatible protocol without a JVM dependency, simple installation and upgrades, Prometheus support, stable low-latency messaging, and cost-effective retention with S3-compatible tiered storage. The firm started with self-managed Redpanda on bare metal, added containerized deployments on Podman and Kubernetes, and has since expanded into fully managed, dedicated cloud instances.

Results

Redpanda reliably processes billions of messages a day for Jump Trading. Its telemetry pipeline, one of its largest and most complex workloads, publishes market data, system measurements, and other information for real-time consumption at 50 ms P95 and 150 ms P99 latency. Jump Trading's CTO described the migration as a non-event.

`"We needed assurance we'd have zero data loss. We also needed something that would meet our high-throughput requirements. That put Redpanda in a class of its own... We used to have problems with latencies, servers, scale, etc. We don't deal with any of that anymore. Redpanda just works."`



Alex Davies, CTO, Jump Trading

Alpaca: 100x faster order processing for high-volume algorithmic trading

100x

faster order
processing

\$1B+

in trades per day

1 million

fractional orders in under a
minute

Alpaca offers financial services APIs for stock and crypto investing, real-time market data, and end-to-end brokerage infrastructure. Its custom Order Management System (OMS) sits between every client order and the market, serving millions of investors worldwide.

Challenge

After years of high-volume algorithmic trading, Alpaca set out to make major performance improvements to its OMS. It needed a high-performance streaming platform that could keep up with rapid growth without adding complexity. Apache Kafka was familiar, but its administrative overhead and configuration complexity in production gave the team pause.

Solution

Alpaca chose Redpanda Self-Managed (Enterprise) on Google Cloud for its single-binary deployment, strong performance with minimal configuration, freedom from the JVM and other dependencies, and developer-first tooling. While Alpaca built a write-ahead log for data integrity, Redpanda introduced the team to the creator of the franz-go Kafka client, who helped tailor it for better data durability and recovery.

Results

Alpaca's new OMS processes orders 100x faster, cutting order processing time from roughly 150 milliseconds to consistently under 1.5 milliseconds, with consistently low latency even during heavy trading volumes. Because order processing is sequential per account, that improvement gets each order to market dramatically faster.

`"Redpanda stood out by having a single binary we could
deploy and run, and have an entire streaming engine ready
to go. The performance, throughput and latency we got
with almost no configuration was incredible."`



Raja Bhatia, VP Engineering, Alpaca

StoneX: lower latency, lower cost, and new client products

5-6x

fewer resources
than Kafka

Single-digit ms

latency, down from
double digits

4 TB

of logs processed
daily

StoneX is a Fortune 100 financial services firm providing global market access, clearing and execution, and trading platforms. It serves more than 330,000 active retail accounts across 180 countries and generates terabytes of logs every day.

Challenge

With Kafka, StoneX struggled to meet the performance and latency requirements of its trading and risk systems. Costs were growing faster than the business, and support from its managed Kafka provider fell short. StoneX needed a more efficient platform for high-throughput observability pipelines and latency-sensitive stream processing.

Solution

StoneX replaced Kafka with Redpanda Self-Managed (Enterprise), integrated with Databricks, Spark ML, and Apache Flink®. Full Kafka protocol compatibility made the migration straightforward.

Results

Latency dropped from double-digit to single-digit milliseconds, improving workflows such as trade execution and machine learning prediction model training. StoneX handles the same workloads on 5-6x fewer resources than Kafka required, and it modernized legacy batch processing into real-time streaming. That streaming data now supports new client products, such as financial vehicles and tradable ETFs.

`"We really did just change the endpoint, and off we went...
Redpanda handled the same load on far fewer resources than Kafka
required. It was 5 to 6 times more efficient, greatly reducing our
hardware costs as well as support and licensing costs."`



Michael Pearce, Senior Architect, StoneX

Zafin: a reliable system of reference for the world's leading banks

300M+

daily events
processed

100%

Kafka compatibility

Zero

code changes
needed

Zafin is a SaaS provider that drives core modernization for the world's top banks. Its story shows how the same platform that powers trading workloads supports the wider financial institution.

Challenge

Zafin was processing hundreds of millions of events a day on a Kafka-based platform, and its engineers were burdened with heavy maintenance and frequent outages, including one that took out two regions at once. Banks will not modernize if they risk downtime, so Zafin needed streaming infrastructure that was simple to run and reliable.

Solution

During a proof of technology, Zafin swapped in Redpanda Self-Managed (Enterprise) on Azure Kubernetes Service and found everything still running, with no code changes. Redpanda now sits at the center of the Zafin Integrate & Orchestrate (Zafin IO) platform, alongside Flink, Apache NiFi, and Temporal.

Results

For one of Zafin IO's first adopters, a world-leading bank producing more than 300 million events a day, Zafin centralized real-time data into a unified, industry-aligned data model and shortened time to value. Compute costs fell, day-two operations became simpler, and Zafin planned to replace five brokers and five ZooKeeper nodes per client with three Redpanda brokers.

"Redpanda just works. We didn't need to change our code, and suddenly, our day two operations were dead simple."



Shahir Daya, CT0, Zafin

Why trading firms choose Redpanda

Redpanda gives financial firms two complementary layers.

1. **Redpanda Data Platform** streams, connects, and queries mission-critical data with low latency and full control over where it runs.
2. **Redpanda Agentic Data Plane** gives AI agents governed access to that data and to the rest of the enterprise, with identity, policy, cost controls, and a complete audit trail.

Redpanda Data Platform

Redpanda Data Platform brings three capabilities together under one platform and one support relationship.

Redpanda Streaming: stream data in real time.

A Kafka-compatible streaming engine delivered as a single binary with no external dependencies, auto-tuning, and auto-balancing. One engine supports four modes that can be configured per cluster or per topic, so each workload gets the right balance of latency, durability, and cost:

- **Core engine:** balanced safety and speed for mission-critical order, risk, and market data flows
- **Write caching:** extreme speed with relaxed durability for workloads that prioritize latency
- **Cloud Topics:** lower cost with relaxed latency for high-volume telemetry and logs
- **Lakehouse:** Apache Iceberg ingest for SQL analytics on streaming data
- **Deployment options.** Redpanda Data Platform is available as Self-Managed, Redpanda Cloud, and BYOC, so you can match each workload to the right environment while keeping one platform and one API.

Enterprise features include tiered storage to object storage for long-term retention and replay, Cluster Shadowing for disaster recovery, enterprise-grade security and governance, Redpanda Console for visibility, and 24/7 enterprise support.

Redpanda Connect: connect everything. More than 300 production-grade connectors, including change data capture (CDC) from databases such as PostgreSQL, MySQL, MongoDB, SQL Server, and Oracle, plus Iceberg, Salesforce, and many more. Pipelines are declarative and Kubernetes-native, and they can filter, enrich, and route data in flight, applying governance before data reaches downstream systems.

Redpanda SQL: query all your data. Redpanda SQL, powered by Oxla, is a high-performance, PostgreSQL-compatible OLAP engine purpose-built in C++. It runs ad hoc SQL against live Redpanda topics and historical Iceberg tables in a single statement, with no pipeline to build and no union logic to maintain. Analysts and researchers use the SQL and tools they already know, data stays in place, and Redpanda SQL runs inside your BYOC cluster so data never leaves your network. Because agents can fan out across tables and test hypotheses in parallel, Redpanda SQL gives them the freshness and throughput they need.

Redpanda Agentic Data Plane

The Redpanda Agentic Data Plane is one governed gateway between every agent and every system, so financial firms can put agents to work on real data without losing control. It is built around three ideas: connect, context, and control.

Connect agents to data and systems. Agents reach Redpanda topics, Redpanda SQL, Iceberg tables, internal databases, and private and public APIs through managed and external Model Context Protocol (MCP) servers. The MCP Gateway gives each agent one governed path to every system, with per-agent tool scoping and field-level policy.

Govern access to models. The AI Gateway gives every agent one endpoint for every model, using your own keys, with failover, per-agent cost attribution, and token budgets that refuse requests once a cap is reached.

Build agents or bring your own. Build agents with Redpanda's managed agents, or bring agents built on the framework of your choice. Either way, the platform handles credentials, policy, cost, and audit.

Identity and policy. Agents authenticate through your own identity provider (IdP) via OpenID Connect (OIDC) and run under on-behalf-of (OBO) authorization, so they inherit the same permissions as the people they work for rather than borrowing human credentials. Administrators set data policies and guardrails, and a kill switch stops any agent immediately.

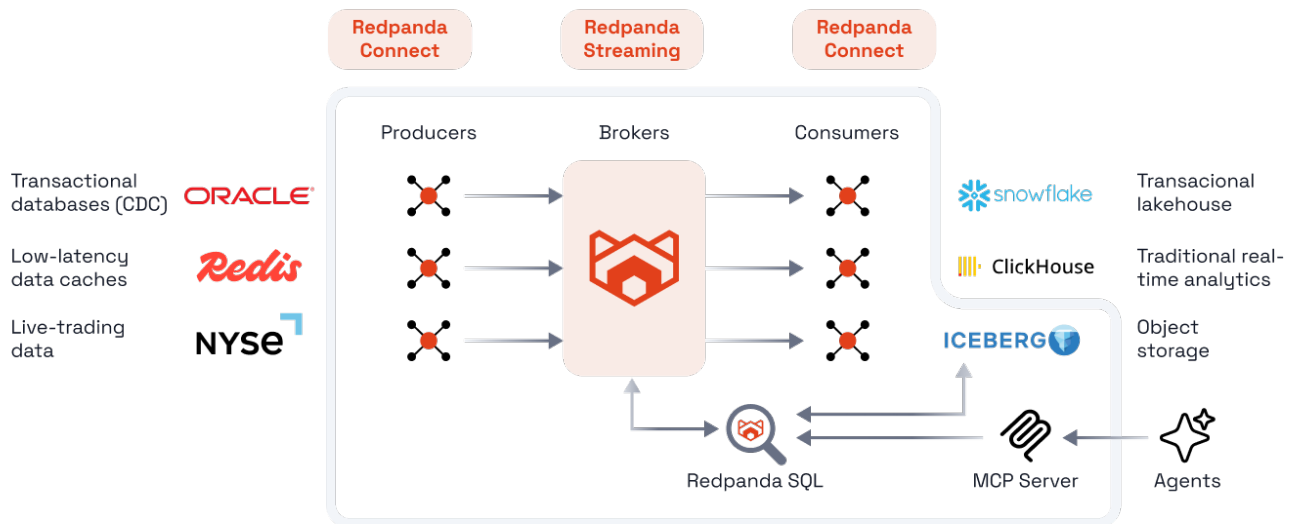
Observability and audit. Every prompt, response, tool call, and key event is captured in a replayable log, with full transcripts available in OpenTelemetry and Open Cybersecurity Schema Framework (OCSF) formats. That gives compliance teams an on-demand audit trail and gives engineers the explainability they need to trust agent output.

Evaluation. The same log powers agent evaluations, so teams can validate answers, track drift, and build confidence in what agents produce. Redpanda works with customers through forward-deployed engineering to tailor evaluations to each business.

Without Agentic Data Plane	With Agentic Data Plane
Agents spread across frameworks, clouds, and identity systems	One governed control plane for every agent
No inventory and no single audit trail when something breaks	Full agent inventory and audit trail on demand
Runaway spend is a matter of when, not if	USpend attributed per agent and capped per budget
Every new agent is a bespoke integration project	Agents reach enterprise data safely through one gateway
Agents run unmonitored, using human credentials	Agents run under their own OBO authorization from your IdP, with transcripts of every action

Reference architecture

Building a real-time data architecture for quantitative trading requires understanding the east-to-west flows from transactional systems into analytical systems. The reference architecture below is simplified. It is meant to show that large organizations have multiple transactional systems of record, both SQL and NoSQL, requiring managing Change Data Capture (CDC), as well as live, large-scale data feeds from SaaS providers such as exchanges.



Redpanda Data Platform architecture for real-time data streaming, system connectivity, and SQL analytics, plus integration with agentic AI.

This data is sent to multiple downstream systems such as data lakehouses and real-time analytics engines. However, with Redpanda users can also use Redpanda SQL to introspect both the live data topics as well as historical Iceberg format tables.

It is not just human financial analysts running these queries. Increasingly analysts are employing AI and fleets of agents to do multiple simultaneous queries. These agents can conduct SQL queries through MCP. Some agents may be persistent, running constantly in the background looking for patterns in real-time data feed. Or they may be fired up as needed, running ad hoc queries to investigate a specific market condition.

Get started with Redpanda

Most trading firms start with a single high-value workload, such as a market data feed, an OMS event stream, or a telemetry pipeline that is straining an existing Kafka deployment. Because Redpanda is Kafka API-compatible, teams can run it alongside their current infrastructure, compare performance and cost on their own workloads, and migrate by pointing applications to a new endpoint. From there, firms add Redpanda SQL for live analytics, land streams in Iceberg for research, and bring agents into production with Agentic Data Plane once the governance model is in place.

Questions to discuss with your team:

- Where does tail latency or a broker failure most often affect a live trading session or client-facing feed today?
- What happens to your infrastructure and on-call teams during the most volatile moments, such as a market open or a central bank announcement?
- Which strategies, data products, or AI projects would you build if infrastructure were not the constraint?
- How would you show a regulator exactly what an AI agent accessed, and on whose behalf?

Chat with an expert

Exchanges, proprietary trading firms, and brokerage platforms all run into the same wall: infrastructure that can't keep up with the market, and AI agents that can't be trusted with real data. Redpanda solves both, so you can move market data, orders, and risk at the speed your business actually runs at.

- [Book a meeting with a Redpanda expert](#)
- [See the Redpanda Agentic Data Plane in action](#)
- [Read more customer stories](#)

Further reading

- [Streaming Wall Street: Fireside chat with NYSE](#)
- [Fireside chat with Jump Trading and Redpanda](#)
- [Redpanda vs. Confluent: A Performance and TCO Benchmark Report](#)
- [Course: Algorithmic trading with Flink](#)
- [Redpanda for Financial Services](#)