

TECH & AI CASE STUDY - FINTECH - 2025

Streaming ETL Cut Data Freshness to Seconds

Based on a delivery engagement with Pinebrook Financial

**6 hours to
seconds**
DATA FRESHNESS

<200 ms
FRAUD SCORING LATENCY

-84%
PIPELINE INCIDENTS

to zero
PARTNER ESCALATIONS

A fillable practice edition: the full story, 8 actions,
6 reflections and a 90-day plan - work it with your team.

Free - no email required
emersontechnology.com

MAKE THIS PRACTICE BOOK YOURS

Work it, do not just read it

This practice book turns the Pinebrook Financial case study into a working session for your team. Read the full story first - what happened, what was built and what it changed. Then tick the 8 actions you can commit to, write short answers to the 6 reflections, and finish with a 90-day plan. Every field in this PDF is fillable: open it in Adobe Acrobat Reader, Chrome, Edge or macOS Preview, type directly into the boxes and save.

YOUR NAME	ROLE
COMPANY	TEAM / DEPARTMENT
DATE	EMAIL (OPTIONAL)

MY GOAL FOR THIS PRACTICE SESSION

- 01 The story** - What happened at Pinebrook Financial - before, during and after the project.
- 02 The results** - The numbers that changed, plus industry benchmarks from public research.
- 03 The 8-action checklist** - Tick boxes as your organization completes each step. Owner and date fields included.
- 04 The 6 reflections** - Fillable questions that pressure-test your own roadmap.
- 05 My 90-day plan** - Three fillable phase cards that turn this case study into your roadmap.

RUN IT AS A 90-MINUTE TEAM SESSION

25 min READ Read the story together. Ask everyone to note the one number that surprised them most.	40 min PRACTICE Work the 8-action checklist and the 6 reflections. Assign an owner and a date to every action you commit to.	25 min COMMIT Fill in the 90-day plan. One goal per phase, one number that tells you it worked. Book a review date before you leave the room.
--	--	---

THE STORY

What happened, and why it worked

01 The Company

Pinebrook Financial is a payments platform processing roughly 40 million transactions per month for merchants and partner institutions. The product promise is speed and reliability of settlement, and behind that promise sits a data platform that decides, in effect, which transactions get approved, which get flagged, and how quickly partners see their money move. The engineering organization is lean: a small data team operated the pipelines, the fraud infrastructure and the reporting layer, while an operations team handled settlement exceptions and partner relationships.

Pinebrook's architecture had been built for a smaller company. As transaction volume grew, the nightly batch pattern that once worked acceptably had become the operational center of gravity — and the operational bottleneck — for everything the business did with data.

02 Where Things Stood

The entire data platform ran as nightly batch jobs. Fraud models scored transactions against account balances and risk flags that were up to six hours stale, which in payments is an eternity: patterns that matter — card-testing bursts, velocity spikes, account takeovers — develop in minutes. The ops team discovered failed settlement files not from monitoring but when partners called to ask where their money was, which is the most expensive possible alarm. Each evening's run had a single point of failure: one corrupted file could stall the entire job chain, triggering a 2 a.m. pager and a full rerun that pushed data freshness even further behind. The team's on-call rotation revolved around a ritual they had come to accept as normal.

Analysts opened each day with yesterday's incomplete picture, and morning reporting meetings began with caveats about what had not loaded. Behind the incidents sat a quieter structural issue: because every consumer shared one batch chain, a failure anywhere degraded everyone at once, and there was no mechanism to replay or isolate a bad input. Recovery, when it happened, was manual reruns executed under pressure — the process equivalent of restarting a machine until it works. A compliance review added a quieter but sharper worry: the firm was criticized for its inability to reconstruct data lineage quickly — tracing a single record from partner feed to ledger to report took hours of engineer time, and in a regulated payments business, that is a finding waiting to repeat. Gartner has described continuous intelligence — real-time analytics integrated directly into operations — as a top strategic differentiator for data-driven organizations; Pinebrook was running the opposite model, and paying for it nightly.

03 The Challenge

The challenge was architectural and cultural at the same time. Architecturally, batch was load-bearing: fraud scoring, partner settlement checks, warehouse tables and analyst reporting all consumed the same overnight pipeline, so nothing could move to real time until the pipeline itself was re-platformed. The batch chain's fragility — one file stalls everything, one rerun delays everything — meant reliability and freshness were the same problem wearing two faces. Every fix to the batch jobs treated symptoms; the design itself had reached its limit.

Culturally, the team had adapted to the constraints in ways that made the status quo feel safer than it was. The 2 a.m. pager was staffed and tolerated. Stale-data caveats were ritualized into meeting scripts. Partner escalations were handled heroically rather than prevented. DORA's State of DevOps research reports that elite-performing teams typically deploy on demand and recover from incidents dramatically faster than low-performing peers — and by that standard, Pinebrook's data platform was operating in the lowest tier, not because its engineers lacked skill but because its architecture made recovery slow and failure loud. The mandate for the program was therefore

explicit: change the recovery model first, the freshness second, and let neither come at the cost of the other.

04 Discovery and Diagnosis

Emerson Technology began with a three-week audit of the batch estate. Every nightly job was documented with its inputs, dependencies, failure history and consumers, producing a dependency graph that made the single point of failure visible: a settlement-file parser sitting upstream of nearly everything. Freshness requirements were mapped per dataset — which tables could tolerate being hours stale, and which could not — and the audit found that only a minority of use cases genuinely needed seconds of freshness, but they were the ones that made money and managed risk: fraud scoring, settlement monitoring, partner status.

The diagnosis of failure modes was equally concrete. Six months of incident history showed the same pattern repeating: a malformed or corrupted file entered the chain, the chain stalled, an engineer was paged, a full rerun was launched, and freshness degraded for everyone downstream. Schema drift from partner feeds was the leading cause of corrupted input, and nothing validated messages at the boundary — bad data propagated until something downstream broke. The compliance lineage finding traced to the same architecture: batch pipelines left little reconstructable history, so tracing a record meant reading code and logs. Forrester's Total Economic Impact studies of enterprise streaming platforms, which typically report triple-digit returns driven by engineering efficiency and faster incident recovery, framed the economics: most of the return would come not from dashboards but from the elimination of the nightly failure ritual.

05 The Solution

Emerson Technology re-platformed ingestion around Kafka, keeping batch only where batch genuinely belongs. The design has three layers.

Streaming the core

Settlement feeds, transaction streams and partner webhooks now flow into Kafka topics processed by stream consumers written in Python and Kafka Streams. Change-data-capture through Debezium continuously syncs the core Postgres ledger into Snowflake, so warehouse tables lag production by seconds instead of hours — achieved without touching application code, which is why CDC was sequenced as the first win rather than a rip-and-replace of the platform. The heaviest historical aggregates still run in orchestrated batch, because reconciliation and regulatory reporting belong there; streaming everything would be architecture fashion rather than architecture judgment.

Fraud scoring moves online

Fraud scoring moved to an online path: features computed in-stream feed a lightweight model that returns a decision in under 200 milliseconds, while the full historical model continues to retrain nightly on warehouse data. The lightweight model handles the latency-critical decision; the nightly model keeps it calibrated. This split mattered for accuracy and for trust — the online scorer was run in shadow mode against the batch model for weeks, with every divergence reviewed, before it was trusted to decide in production. Freshness stopped being a fraud vulnerability: velocity and balance signals are now seconds old instead of six hours stale.

Operability by design

Every pipeline emits structured events to a monitoring layer with schema-registry validation at the boundary, so malformed messages are quarantined rather than propagated — most streaming incidents turn out to be schema drift in disguise, and Pinebrook now catches drift before it corrupts anything. Replayable topics and dead-letter queues, built from day one, turned most 2 a.m. incidents into morning fixes that nobody is woken for: a bad file quarantines cleanly, an engineer reviews it with full context, and replay tooling reprocesses it without a full rerun.

Airflow now orchestrates only what genuinely needs batching — reconciliation and regulatory reporting — and freshness SLAs are published next to every business KPI, so consumers always know how current their numbers are.

06 The First 90 Days

Weeks one and two produced the dependency graph and the freshness classification, and set the alerting philosophy for the program: alert on symptoms users actually feel — stalled pipelines, lag spikes, quarantines — not on every informational warning, because escalation fatigue is the reason real incidents get ignored. Weeks three through six delivered the Kafka foundation and the Debezium CDC sync of the Postgres ledger, the fastest path to near-real-time warehouse tables. The moment analysts saw dashboard data lagging by seconds, skepticism turned into pull — teams that had been indifferent to the project began requesting streaming feeds for their own workflows.

Weeks seven through ten migrated the settlement feeds and partner webhooks onto topics with schema-registry validation and dead-letter queues. The cutover ran in parallel with the legacy batch chain for two full weeks: every file processed by both paths, outputs compared, discrepancies resolved. This double-run period was deliberately conservative — in a payments business, a silent mismatch between old and new pipelines would be a trust event worse than any outage. Weeks eleven and twelve took the online fraud path through shadow mode to live traffic, with the divergence log shrinking weekly, and closed with the first quarterly rehearsal: replaying a deliberately corrupted hour of data end to end, timed, with the full team watching. The rehearsal surfaced two gaps in the replay tooling and fixed them before a real incident could.

By the end of the quarter, the on-call phone had gone quiet, the batch chain had been reduced to reconciliation and regulatory reporting, and the compliance team could trace any record's lineage through replayable streams in minutes. The engineering team kept a written record of every incident during the transition, old platform and new, so the 84% incident reduction would stand on evidence rather than anecdote — a habit that has since outlived the project.

07 The Results

Warehouse tables now lag the ledger by seconds instead of six hours, and fraud decisions execute in under 200 milliseconds. Pipeline incidents fell 84% in the first quarter, and no settlement failure has gone undetected since cutover — partner escalations over stale files dropped to zero last quarter, compared with a pattern of embarrassed phone calls before. Analysts start from a complete, current picture instead of yesterday's partials, and the compliance team can reconstruct lineage for any record in minutes, converting a critical review finding into a capability.

METRIC	BEFORE	AFTER
Data freshness	6 hours	Seconds
Fraud scoring latency	-	Under 200 ms
Pipeline incidents	-	-84%
Partner escalations	Recurring	Zero last quarter

The operational culture changed with the architecture. The 2 a.m. pager is now a rare event rather than a rotation staple, because failures quarantine and replay instead of cascading. Incident recovery is measured in minutes, which is exactly the capability DORA's research associates with elite performance. And the fraud team's shadow-mode discipline became the template for every subsequent model change — new versions prove themselves against production traffic before they decide anything.

08 What Your Business Should Do

The path Pinebrook took is deliberately incremental, and the sequencing is the lesson. Start by mapping freshness requirements per dataset: classify your tables by how stale they can be before the business bleeds money, and stream only what justifies the added complexity — not every table needs streaming, and pretending otherwise doubles your operating burden for no return. Introduce CDC before any rip-and-replace: change-data-capture from your operational database is the fastest route to near-real-time warehouse tables and delivers freshness wins in weeks without touching application code. Keep batch for what batch does well — reconciliation, regulatory reporting, heavy historical aggregates — because streaming everything is architecture fashion, not architecture judgment.

Then engineer for failure before it happens. Validate schemas at the boundary with registry checks so malformed messages are quarantined instead of silently corrupting downstream tables; most streaming incidents are schema drift wearing a disguise. Build dead-letter queues from day one, with context and replay tooling, because the difference between a page at 2 a.m. and a calm morning fix is replayability. Put freshness SLAs on dashboards, publishing lag next to every KPI so consumers know how current their numbers are — trust survives when freshness is visible and honest. Set alert budgets to cut noise, escalating only on symptoms users actually feel. And rehearse replay and recovery quarterly, practicing a bad hour of data end to end; recovery muscle built in drills is what keeps a streaming outage from becoming a trust event.

Ask the operational questions Pinebrook had to answer: how old is the newest data in your most important dashboard; who gets paged when a nightly pipeline fails, and how long does a full rerun take; if a corrupted file hit your pipeline tonight, would it quarantine cleanly or cascade. Gartner's framing of continuous intelligence as a strategic differentiator is, in practice, a test of whether your operations run on data that is seconds old or hours stale. For a payments, marketplaces or logistics business, that gap is not cosmetic — it is the difference between preventing fraud and reporting it. Work through the eight actions above with your platform and risk teams together, and return to the reflection questions in this practice book each quarter to keep the recovery muscle trained.

REAL-WORLD PROOF & BENCHMARKS

Gartner has described continuous intelligence, the integration of real-time analytics into business operations, as a top strategic differentiator for data-driven organizations. - Gartner

Forrester's Total Economic Impact studies of enterprise streaming-data platforms have typically reported triple-digit returns, driven by engineering efficiency and faster incident recovery. - Forrester TEI

DORA's State of DevOps research reports that elite-performing teams typically deploy on demand and recover from incidents dramatically faster than low-performing peers. - DORA

THE PRACTICE PAGES

The 8-action checklist

Work through the list with your team. Tick each box as your organization completes the step, and fill in an owner and a target date so every action has a name and a deadline.

☐

1. Map freshness requirements per dataset

Not every table needs streaming. Classify datasets by how stale they can be before the business bleeds money, then stream only what justifies the added complexity.

OWNER

TARGET DATE

STATUS

--	--	--

☐

2. Introduce CDC before rip-and-replace

Change-data-capture from your operational database is the fastest path to near-real-time warehouse tables. It delivers freshness wins in weeks without touching any application code at all.

OWNER

TARGET DATE

STATUS

--	--	--

☐

3. Validate schemas at the boundary

Enforce schema-registry checks so malformed messages are quarantined instead of silently corrupting downstream tables. Most streaming incidents are just schema drift wearing a very different disguise.

OWNER

TARGET DATE

STATUS

--	--	--

☐

4. Build dead-letter queues from day one

Route failed messages somewhere reviewable, with context and replay tooling. The difference between a page at 2 a.m. and a calm morning fix is replayability.

OWNER

TARGET DATE

STATUS

--	--	--

☐

5. Keep batch for what batch does well

Reconciliation, regulatory reporting and heavy historical aggregates often belong in orchestrated batch jobs. Streaming everything is architecture fashion, not architecture judgment — reserve streams for latency-critical paths.

OWNER

TARGET DATE

STATUS

--	--	--



6. Put freshness SLAs on dashboards

Publish a lag metric next to every business KPI so consumers know how current the numbers are. Trust survives when freshness is visible and honest.

OWNER

TARGET DATE

STATUS

--	--	--



7. Set alert budgets to cut noise

Alert on symptoms users actually feel — stalled pipelines, lag spikes, quarantines — not on every informational warning. Escalation fatigue is the reason real incidents get ignored.

OWNER

TARGET DATE

STATUS

--	--	--



8. Rehearse replay and recovery quarterly

Practice replaying a bad hour of data end to end. Recovery muscle built in drills is what keeps a streaming outage from becoming a trust event.

OWNER

TARGET DATE

STATUS

--	--	--

AFTER THE CHECKLIST: OUR TOP 3 PRIORITIES

From the eight actions above, write down the three you will start first - and the single number you expect each one to move. Three priorities with owners beat eight without any.

PRIORITY 1 NUMBER IT SHOULD MOVE	PRIORITY 2 NUMBER IT SHOULD MOVE	PRIORITY 3 NUMBER IT SHOULD MOVE
--	--	--

THE PRACTICE PAGES

Reflections for your team

Use these questions in your next planning session and type short answers into the boxes. The gaps you find are usually the fastest wins.

Q1. How old is the newest data in your most important dashboard — minutes, hours or days?

Q2. What decisions are you making today on numbers that were already stale when they loaded?

Q3. Who gets paged when a nightly pipeline fails, and how long does a full rerun take?

Q4. Which of your use cases genuinely need seconds of freshness, and which just sound impressive?

Q5. If a corrupted file hit your pipeline tonight, would it quarantine cleanly or cascade?

Q6. Could you reconstruct the lineage of any single record if a regulator or partner asked this week?

THE PRACTICE PAGES

My 90-day plan

Finish the session by committing to a plan. Three phases, one goal each, one number that tells you it worked. Bring this page to your next leadership review - or to a free consultation with Emerson Technology, and we will pressure-test it with you.

Days 1-30 Focus: audit and alignment

PRIMARY GOAL

KEY ACTIONS

HOW WE MEASURE SUCCESS

OWNER

Days 31-60 Focus: build and prove

PRIMARY GOAL

KEY ACTIONS

HOW WE MEASURE SUCCESS

OWNER

Days 61-90 Focus: scale and measure

PRIMARY GOAL

KEY ACTIONS

HOW WE MEASURE SUCCESS

OWNER

Tip: save a copy per team so each group keeps its own answers. Revisit the reflections every quarter - the questions stay the same even as the numbers move.

BOOK THE REVIEW BEFORE YOU CLOSE THE BOOK

Plans without a review date drift. Before you close this practice book, agree on when you will come back to these pages, who will run that session, and when you will check the numbers again. If you want an outside pair of eyes, bring this page to a free consultation with Emerson Technology and we will pressure-test the plan with you.

REVIEW DATE	REVIEWED BY	NEXT CHECK-IN
------------------------------------	------------------------------------	--------------------------------------

Ready for a transformation like Pinebrook Financial's?

Emerson Technology designs and builds the data, AI, cloud and automation systems behind results like these. On a free consultation call we will map this exact playbook to your business - your data, your constraints, your numbers.

Book a free consultation

support@emersontechnology.com

or start online: emersontechnology.com/contact