
Tech4Biz
Excellence Guaranted!

Tech4Biz
Excellence Guaranted!

CASE STUDY 05

AUTONOMOUS FINANCIAL INTELLIGENCE

Real Time Fraud, Risk and Regulatory Operations on a Databricks Lakehouse

DELIVERED FOR A MAJOR AUSTRALIAN BANKING GROUP

We built one governed copy of every financial event the group produces, streamed in as it happens, scored by models that carry their own lineage, and served to fraud operations, risk, treasury and the regulator from the same tables. The client has asked not to be named. What follows is the architecture we delivered and the reasoning behind the decisions we took.

The Business Problem

Where the data actually sits

The client is a large Australian banking group. Like every institution of its size it runs core banking, cards, payments, trading, treasury, AML, KYC, collections and CRM on systems that were bought or built across thirty years. Each of them holds its own copy of the customer, its own definition of an exposure and its own batch window. None of them was designed to be read by anything other than itself.

The consequence is that the group has plenty of data and very little intelligence. A fraud pattern that is obvious when card, payment and login events are seen together is invisible when each channel is scored by its own engine. A concentration risk that is clear when trading and lending are viewed as one exposure is discovered at quarter end.

What the group was doing instead

- **Nightly extracts into a warehouse.** Which means every risk answer is at best a day old, and the trading day it describes is already closed.
- **A separate fraud engine per channel.** Card, net banking, UPI and wire each with their own rules, their own thresholds and no shared view of the customer.
- **Risk in spreadsheets.** Because the warehouse does not carry the attributes the risk team needs, so the last mile is done by hand and cannot be audited.
- **Regulatory returns written to order.** A team of analysts rebuilding the same numbers every quarter from slightly different sources.

When the supervisor asks how a number was produced, the honest answer takes weeks to assemble.

Where the cost sits

Cost	Driver
Fraud losses	Detection that happens after settlement rather than before authorisation.
False positive review	Analyst hours spent clearing alerts that a wider view of the customer would never have raised.
Capital held against uncertainty	Risk that cannot be measured precisely has to be covered conservatively.
Regulatory remediation	Findings that come from inconsistent numbers rather than from bad lending.
Reconciliation headcount	People employed to make two systems agree about the same event.
Time to answer	Weeks between a board question and a defensible answer.
Model sprawl	Scores in production that nobody can retrain, because the training data no longer exists.

> **THE REQUIREMENT IN ONE SENTENCE**
>
> Land every financial event the group produces into one governed lakehouse within seconds of it happening, serve fraud decisioning, risk measurement, treasury and compliance from the same curated tables, and be able to show a regulator the exact lineage of any number on any report.

Solution Overview

A single Databricks Lakehouse holds every event in three refinement stages, one catalogue governs who can see what, one feature store serves the same features to training and to the live decision path, and a set of bounded agents does the routine analytical work that currently occupies people. Everything above the raw layer is derived, versioned and reproducible.

Layered architecture

LAYER 7 DECISION

Command centre and channel APIs

Executive views, fraud operations console, risk workbench and low latency scoring endpoints.

LAYER 6 AGENTS

Bounded autonomous workflows

Fraud triage, alert narrative, treasury what if, regulatory drafting, each with an explicit authority limit.

LAYER 5 REASONING

Vector search, knowledge graph and retrieval

Policy, regulation, contracts and counterparty relationships made searchable and traversable.

LAYER 4 MODELS

Feature store, MLflow and serving

Fraud, credit, AML, liquidity and forecasting models with one registry and one promotion path.

LAYER 3 GOVERNANCE

Unity Catalog across every asset

Tables, volumes, models, features and dashboards under one permission model with lineage.

LAYER 2 CURATION

Delta Live Tables, bronze to gold

Declarative pipelines with data quality expectations enforced at write time.

LAYER 1 INGEST

Streaming, change capture and files

Kafka, Event Hubs, database change capture, Auto Loader and API pulls into one landing zone.

The authority boundary

Autonomy in a regulated institution has to be bounded by what a supervisor would accept. The platform acts on its own only where the action is reversible and the evidence is complete. Everything else is prepared for a human and waits.

The platform decides	A human decides
Decline or step up a transaction	Close or exit a customer relationship
Raise and rank an alert	File a suspicious activity report
Assemble the evidence pack	Approve a credit limit change
Recommend a liquidity action	Execute a treasury trade
Draft a regulatory return	Sign and submit the return
Flag a model as drifting	Promote a model to production

WHY THE BOUNDARY IS DRAWN HERE

Every action on the left is reversible within minutes and leaves a customer who can be contacted. Every action on the right creates a legal or contractual position that cannot be undone by rolling back a pipeline. Model risk governance treats the two categories very differently, and the architecture keeps them physically separate so the difference can be evidenced rather than asserted.

Ingestion and Source Integration

How each class of source arrives

Source	Mechanism and latency
Core banking	Log based change capture off the database redo stream into Kafka. Seconds, no load on the core.
Card and switch	ISO 8583 authorisation messages tapped at the switch, published to Kafka. Sub second.
Payments	SWIFT MT and ISO 20022 XML parsed at the gateway. Near real time on the outbound leg.
Trading and positions	FIX drop copy plus end of day position files. Intraday for fills, batch for positions.
Market data	Vendor feeds landed as files and streams, with a point in time snapshot kept for every valuation run.
CRM and servicing	Change capture on the operational store, hourly for reference data.
AML and KYC	Case and screening outcomes pushed on state change.
Documents and voice	Contracts, statements and call recordings into governed volumes, then parsed asynchronously.

The landing contract

Every producer publishes against a registered schema. Compatibility is checked in the registry before the message is accepted, so a change on a source system fails at the producer rather than silently corrupting a downstream table three days later.

What makes financial streaming hard

ORDERING

Events arrive out of order and that matters

An authorisation, a reversal and a settlement for the same transaction can arrive in any order. Bronze keeps arrival order, silver reconstructs business order from the event timestamps and the transaction key.

DUPLICATION

Exactly once is a property of the sink

Producers retry, so duplicates are normal. Idempotent merges keyed on the source identifier plus the event sequence make replay safe, which is what allows a pipeline to be rerun without fear.

LATE DATA

Watermarks tuned per stream

Card traffic settles in seconds, correspondent banking messages can be hours late. Each stream carries its own watermark and its own late arrival table rather than one global setting.

BACKFILL

History and stream on one code path

The same transformation runs over the historical file load and over the live stream, so a model trained on history sees exactly what the live path will produce.

REPLAY IS A DESIGN REQUIREMENT, NOT A RECOVERY PLAN

Regulators ask institutions to reproduce a decision made months ago. That is only possible if the raw layer is immutable, the transformations are versioned in source control and the pipeline can be pointed at a past window and rerun. Build for that from the first pipeline rather than retrofitting it after the first finding.

The Lakehouse Data Model

Three stages, each with a clear job

Stage	What lives there and why
Bronze	The message exactly as it arrived, with source, offset, ingest time and schema version attached. Nothing is corrected here. This is the evidence layer, and it is append only.
Silver	Parsed, typed, deduplicated and conformed. Customers resolved to one identifier, currencies normalised, reference data joined, slowly changing dimensions maintained with validity dates.
Gold	Business objects that people and models actually consume. Customer 360, exposure, transaction with full context, position, limit utilisation and the regulatory reporting marts.

Quality enforced at write time

Delta Live Tables expectations are declared alongside the transformation rather than run as a separate audit afterwards. A row that fails a hard expectation is quarantined instead of published, and the quarantine table is itself monitored, so bad data becomes a visible operational queue rather than a silent gap in a report.

Expectation class	Example
Structural	Mandatory keys present, amounts numeric, currency in the approved list.
Referential	Every transaction resolves to an account that exists on the same business date.
Business	Debits and credits balance for a posting group, settlement never precedes authorisation.
Statistical	Daily volume and value within the expected band for that source and that day type.

Physical design that keeps queries cheap

- **Partitioning by business date** on the large event tables, because almost every query and every regulatory extract is bounded by a date range.
- **Liquid clustering on the high cardinality keys** such as customer, account and counterparty, so investigation queries do not scan the whole table.
- **Z ordering or clustering on the join keys** used by the feature pipelines, since those run continuously and dominate compute cost.
- **File compaction on a schedule**, because streaming writes produce many small files and small files are the most common reason a lakehouse gets slow six months in.
- **Deletion vectors** for corrections and right to erasure, so a single record can be removed without rewriting a large partition.
- **Time travel retention set by policy**, long enough for reproducibility, short enough to control storage and to satisfy retention rules.

> **THE GOLD LAYER IS A CONTRACT, NOT A CONVENIENCE**
>
> Once a gold table is consumed by a regulatory report, its schema and its definitions are under change control. New columns can be added, existing meanings cannot quietly change. Publish the definitions with the table so that the business meaning of a column and the code that produces it are versioned together.

Governance and Control

One catalogue for everything

Tables, files, machine learning models, features, dashboards and functions all sit under the same permission model. That single fact removes most of the audit burden, because access to a model that was trained on customer data is governed the same way as access to the customer data itself.

Control	Implementation
Namespace	Catalogue by environment and legal entity, schema by domain, so cross border access is a permission rather than a copy.
Row filters	Applied on the table, so an entity or a branch sees only its own rows regardless of which tool asks.
Column masks	Account numbers, identifiers and contact details masked by default and unmasked by role and purpose.
Tag based policy	Sensitivity tags drive masking automatically, so a new column inherits protection instead of waiting for someone to notice it.
Lineage	Captured automatically from query execution, column level, across notebooks, pipelines, dashboards and models.
Audit	Every read, write and permission change logged to an immutable store with its own retention.
Sharing	Open protocol sharing to auditors and group entities without copying data out of the platform.

Why lineage is the whole argument

When a supervisor questions a figure in a return, the institution has to show where the number came from, which transformation produced it, which version of the code was running and who could have changed it. Column level lineage collected from actual execution answers that in minutes. A lineage document maintained by hand answers it in weeks and is usually wrong.

Data quality as an operational signal

MONITORS

Profile and drift tracked per table

Distribution, null rate, volume and freshness compared against a baseline window, with alerts routed to the owning team.

FRESHNESS

Every gold table publishes its own age

Consumers can see how current a table is before they use it, and a stale table is visible on the dashboard rather than assumed to be fine.

OWNERSHIP

A named owner per domain

Quality failures go to a team that can fix the source, not to a central platform group that can only report them.

DECIDE THE ENTITY MODEL BEFORE THE FIRST PIPELINE

Whether a customer identifier is shared across legal entities, and whether data can move across borders, determines the catalogue layout, the masking rules and in some jurisdictions the physical region. Changing it later means rebuilding the permission model on live data. Settle it with legal and compliance in week one.

The Model Estate

What runs, and on what

Model	Job and shape
Fraud ring detection	Graph neural network over the customer, device, account and counterparty graph. Catches collusion that per transaction scoring cannot see, because the signal is in the structure rather than in any single payment.
Transaction scoring	Gradient boosted trees on behavioural and velocity features, tuned for a fixed latency budget and calibrated so the score maps to an expected loss.
Credit risk	Probability of default, loss given default and exposure at default, with monotonic constraints where the regulator expects a direction of effect.
AML typologies	Sequence models over transaction chains, layered on top of the mandated rules rather than replacing them.
Document understanding	Transformer extraction over contracts, trade confirmations and statements, producing structured fields with a confidence score.
Liquidity and cash	Time series forecasting of inflow and outflow by currency and account group for intraday liquidity management.
Portfolio optimisation	Constrained optimisation over positions, limits and funding cost, run as a scenario rather than as an instruction.

The feature store earns its place

The most common cause of a fraud model that tests well and performs badly is that the features used in training were computed differently from the features available at decision time. One definition, published once and read by both paths, removes that class of failure.

- **Point in time correctness.** Training joins are as of the event timestamp, so a model never learns from a fact that was not yet known.
- **Online and offline from one definition.** The same aggregation feeds the batch table and the low latency store, with freshness measured on both.
- **Lineage from feature to model to decision.** Which makes a validation review a query rather than an archaeology project.

Lifecycle and model risk

REGISTER

Every candidate logged with data, code and metrics

Nothing reaches review without a reproducible training run.

VALIDATE

Independent review before promotion

Performance, stability, explainability and bias testing recorded against the version.

PROMOTE

Champion and challenger in parallel

The challenger scores live traffic in shadow before it ever influences an outcome.

MONITOR

Drift on inputs and on outcomes

Population shift detected on features, and performance measured once labels mature.

The Real Time Decision Path

The latency budget

A card authorisation gives the institution a fixed window before the network times out and the transaction is approved by default. The budget is set first and the design is fitted to it, rather than the other way round.

Stage	Budget
Message receipt and parse	~3 ms
Online feature lookup	~8 ms
Graph neighbourhood fetch	~10 ms
Model inference	~12 ms
Rules and policy overlay	~5 ms
Decision, log and respond	~7 ms
Headroom for the tail	remainder of the window

Everything that cannot be done inside the budget is done outside it. Heavy graph computation runs continuously and writes a small precomputed neighbourhood signal that the live path only has to read.

Decision tiers

APPROVE

Score below the low threshold

No friction, logged with the score and the top contributing features.

STEP UP

Score in the uncertain band

Additional authentication requested rather than an outright decline, because a wrong decline costs a customer.

DECLINE

Score above the high threshold

Reversible, with the evidence pack already assembled for the review queue.

HOLD AND REVIEW

High value or unusual counterparty

Payment queued for a named approver with the reason stated in plain language.

THRESHOLDS ARE A BUSINESS SETTING, NOT A MODEL SETTING

The model produces a calibrated probability. Where the thresholds sit is a decision about the cost of a missed fraud against the cost of a declined genuine customer, and it belongs to the business with the numbers in front of them. Keeping the two separate means thresholds can be tuned weekly without retraining anything.

DESIGN THE FAILURE MODE EXPLICITLY

If the scoring service is unavailable, the switch still has to answer. Decide in advance whether the fallback is a conservative static rule set or a straight approve, measure how often it is used, and alert on it. An undefined fallback becomes an unmeasured one.

Agents and the Reasoning Layer

What the agents are given

The reasoning layer is not a general assistant. Each agent has a defined question set, a defined set of tools, and a defined authority limit. It reads from the gold tables, searches the policy and regulation corpus, walks the counterparty graph, and produces a written answer with references.

Agent	Scope and limit
Fraud triage	Assembles the case, ranks alerts by expected loss, drafts the narrative. It cannot close an alert.
AML investigation	Traces fund flows, pulls prior cases with similar typologies, prepares the pack. It cannot file a report.
Credit review	Summarises exposure, covenants and recent behaviour for a named borrower. It cannot change a limit.
Treasury analyst	Runs liquidity scenarios against live positions and states the funding implication. It cannot execute.
Regulatory drafting	Populates a return from the reporting marts and lists every figure it could not source. It cannot submit.
Control tester	Samples transactions against a stated control and reports exceptions with evidence.

The knowledge corpus

Regulation and circulars · **Internal policy** · **Product terms** · **Credit agreements**

Prior case files · **Model documentation** · **Data dictionary** · **Audit findings**

Grounding contract

Condition	Required behaviour
Figure available in a gold table	State it, name the table and the as of timestamp.
Figure only in a source system	Say so, and flag it as a gap in the reporting mart rather than estimating it.
Policy question	Quote the clause reference. Never paraphrase a rule without pointing at it.
Conflicting policy versions	Prefer the most recent effective date and state that the guidance changed.
Several plausible explanations	Rank them and state the test that would separate them.
Insufficient evidence	Refuse, and record the refusal so the gap can be closed.

REFERENCES ARE MANDATORY

Every substantive claim carries a table, a timestamp and where relevant a clause reference that the analyst can open. That is what makes the output reviewable before anyone acts on it, and it is the single feature that decides whether experienced risk staff use the system or work around it.

REFUSALS ARE A PRODUCT FEATURE

The list of questions the agent could not answer is a precise map of where the gold layer is thin. Over a year that list is worth more than the agent, because it tells the platform team exactly what to build next instead of leaving it to opinion.

Worked Example, End to End

TRIGGER

A card present authorisation for 84,000 rupees at a merchant the customer has never used, at 02:14, in a city the customer has not transacted in for two years.

STEP 1 READ

Message parsed and enriched

Customer, account, device and merchant resolved against silver in under three milliseconds.

STEP 2 FEATURES

Behavioural window retrieved

Spend velocity, merchant category history, geographic pattern and time of day profile read from the online store.

STEP 3 GRAPH

Precomputed neighbourhood read

The merchant terminal has been touched by nine cards in six hours, four of which were later disputed.

STEP 4 SCORE

Two models, one decision

The transaction model returns a moderate score. The ring signal from the graph raises it into the decline band.

STEP 5 ACT

Declined and packaged

Response returned inside the window, evidence pack written, case opened and ranked for the fraud desk.

ANALYST

"Why did this one decline when the amount is not unusual for this customer?"

FRAUD TRIAGE AGENT

"The amount was not the driver. The terminal at merchant M-4471 has been used by nine distinct cards since 20:30, and four of those authorisations have already been disputed. This customer has no prior activity at that merchant category in this city. The transaction score alone was 0.41, which would have been a step up. The terminal ring signal took it to 0.87. Source tables are gold.txn_enriched as of 02:14:07 and gold.merchant_graph_signal as of 02:10:00."

ANALYST

"Are the other five cards still live?"

FRAUD TRIAGE AGENT

"Five are live and three have transacted in the last twenty minutes. I have ranked them by exposure and prepared a block recommendation for each. Blocking is your decision, not mine. If you want the merchant itself reviewed, the counterparty pack is ready and the acquirer relationship sits with the merchant services team."

WHAT MADE THIS WORK

Nothing in this sequence required a new model. It required the card stream, the dispute stream and the merchant reference data to be in the same governed place, joined on a key both sides agree on, with the heavy graph work done in advance so the live path only had to read a number.

Security and Resilience

Security posture

Control	Implementation
Network	Private connectivity end to end, no public endpoints on the workspace or the storage account, egress restricted to an approved list.
Identity	Single corporate directory, short lived tokens, no shared service accounts, machine identities scoped per pipeline.
Encryption	Customer managed keys for storage and for managed services, with rotation and revocation held by the institution rather than the platform.
Secrets	Held in the enterprise vault and referenced, never present in notebooks, jobs or repositories.
Least privilege	Access granted to a group by role and purpose, reviewed on a recertification cycle, with standing access to production data treated as an exception.
Tokenisation	Card numbers and national identifiers replaced at ingest, with detokenisation available only to a small number of controlled paths.
Audit immutability	Access logs written once to a separate account under different administration.

Continuity

Capability	Target and method
Storage durability	Cross region replication of the Delta storage, continuous.
Metadata recovery	Catalogue and permission model reproduced from code, not restored from a backup by hand.
Streaming recovery	Checkpoints replicated, streams restart from the last committed offset with idempotent writes.
Decision path	Scoring endpoints active in two regions, traffic shifted by health check, because this is the only component with a customer waiting.
Analytical path	Recovered after the decision path, with a longer recovery objective agreed with the business.

SEPARATE THE TWO RECOVERY CONVERSATIONS

The real time decision path and the analytical estate have very different tolerances. Treating them as one system produces an expensive design that satisfies neither. Agree a recovery objective for each, write it down, and test both by actually failing over rather than by reviewing a document.

RESIDENCY IS AN ARCHITECTURE INPUT

Where data may be stored and processed, and whether a group function outside the country may query it, changes the region layout and the sharing model. It is far cheaper to design for than to remediate, and it is knowable on day one.

How We Delivered It

The phases we ran

Phase	Scope	What it produced
Phase 0 Foundation	Entity and residency model, catalogue layout, network and identity, source inventory with owners.	Signed data governance design and a working landing zone.
Phase 1 First domain	Cards and payments end to end, bronze to gold, with quality expectations and lineage.	Gold tables reconciled to the source of record for a full month.
Phase 2 Decisioning	Feature store, transaction scoring in shadow mode against live traffic.	Latency budget met at peak and score agreement measured against the incumbent engine.
Phase 3 Risk and reporting	Credit and exposure marts, first regulatory return produced from the platform in parallel with the existing process.	Two consecutive returns matched, with differences explained.
Phase 4 Agents	Fraud triage and regulatory drafting released to a named user group.	Accuracy and reference quality accepted by the risk function.
Phase 5 Decommission	Legacy extracts and shadow spreadsheets retired domain by domain.	Measured reduction in reconciliation effort.

How it runs now

PIPELINES AS CODE

Everything in source control

Notebooks, pipeline definitions, expectations, permissions and dashboards deployed through the same review process as application code.

ENVIRONMENTS

Development, test and production isolated

Separate catalogues with production data never present below production, and synthetic or masked data used for development.

MODEL PROMOTION

Automated gates, human approval

Tests, bias checks and validation artefacts produced by the pipeline, with the promotion itself signed by a person.

COST CONTROL

Chargeback per domain

Compute tagged to the owning team so cost is visible where the decision to run a query is made.

PARALLEL RUN IS THE ONLY CREDIBLE CUTOVER

No risk committee will accept a regulatory number from a new platform on its first month. Run both, explain every difference until the differences are all explained by the new platform being more correct, and only then switch. The parallel period is not waste. It is the evidence that makes the switch approvable.

Benefits and Measurement

Benefits realised

One version of the number

Board pack, risk report and regulatory return sourced from the same gold tables.

Fraud caught before settlement

Decisioning inside the authorisation window rather than detection after the money has moved.

Fewer false positives

A wider view of the customer removes alerts that a single channel would have raised.

Faster regulatory response

Lineage answers a supervisory question in hours instead of weeks.

Models that can be defended

Every score reproducible from registered data, code and parameters.

Less reconciliation

Systems agree because they read the same curated table rather than two extracts.

Analyst time recovered

Evidence assembly done before the analyst opens the case.

Documented data gaps

Every refusal maps exactly where the platform still cannot answer.

KPI framework

Measure	What it tells you
Fraud loss rate	The headline number, measured per channel and per value band.
False positive ratio	Customer friction and analyst load created by the decisioning policy.
Decision latency at the tail	The ninety ninth percentile, because the average never times out and the tail does.
Data freshness by domain	How old the newest gold row is when a decision is taken.
Reconciliation breaks	Differences between the platform and the source of record, trending to zero.
Time to answer a supervisory query	The clearest external proof that governance is working.
Model coverage and drift	Share of decisions made by a monitored model still inside its validated range.
Refusal and gap rate	Where to invest the next domain build.

BASELINE BEFORE YOU BUILD

Capture the current fraud loss rate, false positive ratio, reconciliation break count and time to answer before the first pipeline is written. These are usually recoverable from existing operational reports, and without them the business case after go live is an argument rather than a measurement.

Scale and Performance

Scale the platform was built to carry

Metric	Enterprise scale
Customer base	10 to 50 million
Daily financial transactions	80 to 300 million
Streaming events	20,000 to 100,000 per second
Data ingestion	20 to 80 TB per day
Historical data	2 to 10 PB
Models in production	80 to 250
Concurrent business users	5,000 to 20,000

Engineering targets

Measure	Target
Streaming latency	Under 2 seconds
Fraud decision response	Under 500 ms
Model retraining	Daily to weekly by model class
Data pipeline service level	99.9 percent
Platform availability	99.95 percent
Query performance improvement	35 to 55 percent
Automated data quality validation	Above 95 percent

Business improvement ranges

Measure	Expected improvement
Fraud loss	22 to 35 percent reduction
False positives	30 to 45 percent reduction
AML investigation time	40 to 55 percent faster
Credit risk prediction accuracy	12 to 18 percent improvement
Treasury forecast accuracy	15 to 25 percent improvement
Regulatory report preparation	50 to 70 percent faster
Executive decision time	35 to 50 percent faster
Operational cost	15 to 22 percent reduction

HOW TO READ THESE FIGURES

The scale column describes the volumes the platform was engineered to carry. The target column is the service level it is built and operated to, and it is measured continuously. The improvement column is the range this class of platform delivers, and it is confirmed against a client's own baseline during assessment rather than claimed in advance. We capture that baseline before the first pipeline is written, because a benefit without a baseline is an argument rather than a result.

THE NUMBER THAT GOVERNS THE DESIGN

Fraud decision response under 500 milliseconds is the constraint that shapes almost everything else. It is why the graph work runs continuously and writes a precomputed signal, why features are served from an online store rather than computed on the request, and why the scoring path is deployed active in more than one region. Every other target can be met by scaling. That one has to be designed for.

Platform Capability Across the Estate

The three platforms are built on the same Databricks foundation and are operated to the same envelope. The figures below describe the capability of that foundation across the estate rather than the load of any single engagement, and they are the numbers we size and design against when a new platform is scoped.

Platform metric	Typical target
Data ingestion	20 to 150 TB per day
Structured streaming throughput	20,000 to 500,000 events per second
Historical lakehouse	2 to 30 PB
Delta tables	5,000 to 25,000
Models under management	100 to 500
Feature store features	10,000 to 100,000
Vector embeddings	100 million to 2 billion

WHY WE PUBLISH THE ENVELOPE RATHER THAN ONE NUMBER

Sizing is driven by event rate and retention, not by headcount or revenue. Quoting a single figure invites a comparison that does not hold. The range is what we design and cost against, and the point inside it is settled during assessment from the client's own event volumes.

Platform metric	Typical target
Daily inference requests	10 to 100 million
Enterprise users	5,000 to 50,000
Platform availability	99.9 to 99.95 percent
Automated data quality checks	Above 95 percent of published tables
Governance coverage	100 percent of production datasets
Mean time to detect a data issue	Under 15 minutes

THE TWO THAT MATTER MOST

Governance coverage and mean time to detect are the two we hold hardest. A production dataset outside the catalogue cannot be audited, and a data issue that is found by a business user rather than by a monitor has already cost the client the thing the platform was bought to protect.

What We Learned

The hard problems, and what we did about them

Problem	What we did
Source data quality	Profile every source in Phase 0 and publish the findings before committing to a delivery date. This is the dominant determinant of how long the build takes and it is knowable up front.
Customer identity resolution	Agree the matching rules and the survivorship logic with the business early, because every downstream aggregate depends on them and rework is expensive.
Latency under peak load	Load test against festival and salary day peaks, not against an average day, and measure the tail rather than the mean.
Model validation capacity	Independent validation is usually the bottleneck, not model development. Book that capacity at the start of the programme.
Cost drift	Streaming and always on serving cost money continuously. Tag, budget and review monthly from the first pipeline.
Small file accumulation	Scheduled compaction and clustering maintenance treated as a run activity with an owner.
Change on source systems	Schema registry with compatibility enforcement, so a core banking upgrade breaks loudly at the producer.
Skills concentration	Pair delivery with the client team from Phase 1, because a platform only one vendor can operate is a risk on its own.

What we settled before writing any code

- **Entity and jurisdiction scope.** Which legal entities were included and whether data could move between them, because that decides the catalogue layout and in some cases the physical region.
- **The customer master.** Whether an existing master was trusted by the risk function, or whether identity had to be resolved on the platform itself.
- **The authorisation window.** The real timeout on the card and payment rails, and the existing fallback when the scoring engine is unavailable.
- **First returns.** Which regulatory returns would move onto the platform first, and who signs them today.
- **Model risk policy.** What has to be produced before a model may influence a customer outcome.
- **Change capture on the core.** Whether it could be enabled at all, and who owned that decision.
- **The decommission list.** Which existing platforms would be retired, and on what timetable.
- **Retention and erasure.** What obligations applied to each class of data.

> **WHAT WE WOULD TELL THE NEXT CLIENT**
>
> A source readiness assessment on one domain, usually cards and payments. It costs little, it produces a measured picture of data quality, identity coverage and event volume, and it tells you honestly whether the ambition is a twelve month programme or a three year one. Finding that out in six weeks is worth far more than finding it out after the first phase has slipped.

Contact

Website

https://tech4bizsolutions.com

Contact details

If you would like to know more about Tech4Biz Solutions
and our products please contact us via email contact@tech4biz.io

Address

Bangalore
NGR Arcade,
HSR Layout Sector 1,
Bangalore, Karnataka

Surat
306-307,
Millionaire Business Park,
LP Savani Road, Adajan,
Surat, Gujarat - 395009





Tech4Biz is a provider of comprehensive engineering and IT solutions for businesses of all sizes. From cloud computing and data management to security, networking and embedded intelligence, we have the expertise and experience to help your business stay ahead of the curve.

Our mission is to provide businesses with the best possible technology solutions. We understand that a reliable and efficient technology foundation is crucial in an increasingly competitive marketplace. That is why we offer a wide range of services, from managed IT and cloud computing to data platform, applied AI and embedded engineering.

This communication contains general information only, and Tech4Biz Solutions Private Limited is not, by means of this communication, rendering professional advice or services. This document describes work delivered by Tech4Biz Solutions Private Limited. The client has requested that it not be identified, so the client name, commercial terms and any figures that could identify the organisation have been withheld. Architecture descriptions, performance characteristics, sizing figures and integration details are specific to the environment in which the work was delivered and are confirmed against the specific customer platform, workload and data volumes in any subsequent engagement. Third party product names are the property of their respective owners. Before making any decision or taking any action that may affect your finances or your business, you should consult a qualified professional adviser. Tech4Biz Solutions Private Limited shall not be responsible for any loss whatsoever sustained by any person who relies on this communication.

© 2026 Tech4Biz Solutions Private Limited