

# Card Payment Cost Intelligence Platform

How a card payments fintech running entirely on AWS gained its first unified cost visibility platform — built on Microsoft Fabric introduced from scratch, processing millions of transactions and dozens of third-party fee files to produce a single source of truth for interchange, scheme and processing costs.

---

Client: A card payments fintech (anonymised) | Delivered by: Michał Kulikowski / Frogsbyte | Technology: Microsoft Fabric · Power BI · AWS S3 · PySpark

## EXECUTIVE SUMMARY

A fast-scaling card payments fintech operated entirely within an AWS-centric infrastructure. Transaction data lived in S3, cost files arrived daily from 10+ third-party scheme and processor providers in inconsistent flat-file formats, and there was no analytics layer capable of joining them together. The finance team was calculating interchange, scheme fees and processing costs manually in Excel — a process that was slow, error-prone and completely unscalable as transaction volumes grew.

Frogsbyte introduced Microsoft Fabric into the AWS environment from scratch, establishing it as the analytics layer on top of the existing S3 data estate. Using Fabric Shortcuts, data was accessed directly from S3 without duplication. A medallion lakehouse architecture was implemented with PySpark notebooks for transformation logic, and a full CI/CD pipeline was established for production-grade deployment. The result: a Power BI cost intelligence platform giving finance and commercial teams live visibility into payment costs at transaction, merchant, scheme and geography level.

## PROBLEM & PAIN POINTS

- No analytics layer — all cost analysis done manually in Excel by the finance team
- Transaction data in AWS S3 with no governed access path into a reporting layer
- 10+ third-party cost files (interchange tables, scheme fee schedules, processor invoices) arriving in incompatible flat-file formats — CSV, XLSX, pipe-delimited
- No unified cost taxonomy — interchange, scheme fees and processing costs calculated in separate spreadsheets with no reconciliation
- Month-end cost close took 4–5 days of manual effort and was still unreliable
- Zero visibility on cost-per-transaction by merchant, card type, scheme or geography

- No existing Microsoft tooling — entire stack was AWS — making Fabric adoption a greenfield introduction
- No CI/CD discipline on analytics code — notebooks run manually, no version control, no test environments

## SOLUTION ARCHITECTURE

Frogsbyte introduced Microsoft Fabric as the analytics and reporting layer on top of the existing AWS infrastructure. The key architectural principle was non-duplication — Fabric Shortcuts allowed direct access to S3 data without moving or copying it, preserving the AWS data estate while enabling Fabric-native processing. A full medallion architecture was implemented across three lakehouses.

LAYER	COMPONENT	ROLE
Access	Fabric Shortcuts → AWS S3	Direct shortcut connections to S3 buckets — transaction data accessed in-place, no duplication or migration required
Bronze layer	OneLake Lakehouse (raw)	Raw ingestion of transaction data (via S3 shortcut) and all third-party flat files (interchange tables, scheme fee schedules, processor invoices) in original format
Silver layer	OneLake Lakehouse + PySpark	PySpark notebooks clean, standardise and join all sources — unified cost taxonomy applied, flat-file schema normalisation, deduplication
Gold layer	OneLake Lakehouse (aggregated)	Pre-aggregated cost metrics by transaction, merchant, card scheme, geography and time period — optimised for reporting
Modelling	Fabric Warehouse + DAX	Semantic model with measures for interchange cost, scheme fees, processing cost, cost-per-transaction, margin and blended rate
Reporting	Power BI	Finance dashboard: cost breakdown by scheme/merchant/card type, monthly trend, third-party fee reconciliation, cost-per-transaction drillthrough
CI/CD	Azure DevOps + Fabric Git	Full CI/CD pipeline — Git-backed notebook versioning, automated deployment across dev/test/prod workspaces, pull request review gates

## KEY RESULTS & METRICS

<p><b>10+</b></p> <p>THIRD-PARTY SOURCES UNIFIED</p>	<p><b>4 days</b></p> <p>MONTH-END CLOSE TIME SAVED</p>	<p><b>3mo</b></p> <p>FABRIC INTRO TO PRODUCTION</p>	<p><b>0</b></p> <p>DATA COPIES ON S3</p>
--	--	---	--

- Microsoft Fabric introduced into an AWS-only environment and taken to production in 3 months
- 10+ third-party cost file formats normalised into a single unified cost model
- Month-end cost close reduced from 4–5 days to same-day automated processing
- First ever cost-per-transaction visibility by merchant, scheme and card type
- Zero data duplication — S3 data accessed in-place via Fabric Shortcuts
- Full CI/CD pipeline established — first analytics CI/CD discipline in the organisation
- Finance team Excel cost calculation process fully retired

## TIMELINE

PHASE	DURATION	DELIVERABLE
Discovery & architecture design	<b>Week 1–2</b>	AWS S3 audit, Fabric Shortcuts proof-of-concept, medallion architecture design, CI/CD toolchain selection
Bronze layer & flat-file ingestion	<b>Weeks 2–4</b>	S3 shortcut connections, raw lakehouse setup, PySpark ingestion notebooks for all third-party file formats
Silver transformation layer	<b>Weeks 4–6</b>	Cost taxonomy normalisation, schema unification, deduplication logic, interchange and scheme fee join logic
Gold layer & semantic model	<b>Weeks 6–7</b>	Aggregated cost model, Fabric Warehouse, DAX measures for cost-per-transaction, margin, blended rates
Dashboard & CI/CD pipeline	<b>Weeks 7–9</b>	Power BI cost intelligence dashboard, DevOps pipeline setup, Git integration, dev/test/prod deployment
Testing, UAT & go-live	<b>Weeks 9–12</b>	Finance team UAT, month-end parallel run validation, production go-live, handover and documentation

## TECH STACK

---

- Microsoft Fabric — Lakehouses (bronze/silver/gold), Shortcuts, Dataflows, Warehouse, Notebooks
- AWS S3 — primary data estate accessed via Fabric Shortcuts (zero migration)
- PySpark — transformation notebooks for flat-file normalisation, cost taxonomy, deduplication
- Power BI Premium — cost intelligence dashboard with drillthrough and RLS by business unit
- Azure DevOps + Fabric Git integration — CI/CD pipeline across dev/test/prod workspaces
- DAX — interchange cost measures, scheme fee calculations, cost-per-transaction, blended rate
- Python — flat-file schema detection and automated format normalisation utilities

Interested in a similar solution? Get in touch at [frogsbyte.com](https://frogsbyte.com)