CASE STUDY • SWEDISH MARKET • FINTECH

Accounting Platform

SaaS for the Swedish market: integrations with national services (Nordigen, Skatteverket, Riksbank), AI transaction categorization, MSSQL → PostgreSQL migration

ASP.NET Core | Entity Framework | Angular | PostgreSQL | Docker | CI/CD | OpenAI API

2.5 yr
In production serving clients

40%
Faster load times with server-side caching

MSSQL ↓ PG
Database migration with zero downtime

CI/CD
Full pipeline build → test → deploy

PROBLEM

What it was like before us

The platform had accumulated significant technical debt that was slowing down development and creating business risks:

Legacy code without tests

The codebase had grown over the years with no test coverage. Any change could break another module. Developers were afraid to refactor.

MSSQL dependency

High licensing costs and limitations when deploying in containers. Some SQL used MSSQL-specific syntax (TOP, NOLOCK, GETDATE).

Manual categorization

Accountants manually classified every transaction. Hundreds of transactions per month meant hours of work and a high risk of errors.

No CI/CD

Manual deployment via RDP to the production server. No automated builds, no quality checks, no rollback strategy.

ABOUT THE PROJECT

Overview

Client	Swedish FinTech company
Market	Sweden — SMBs, accounting firms
In production	2.5+ years
National integrations	Nordigen, Skatteverket, Riksbank

Our role

Refactoring the legacy codebase, adding test coverage, implementing CI/CD, migrating from MSSQL to PostgreSQL, developing AI transaction categorization, and integrating with Swedish financial infrastructure.

SOLUTION

Key solutions

01 — AI categorization of bank transactions

A two-tier system for automatic transaction classification into accounting accounts:

1. **Rule-based engine**
   A set of pattern-based rules: counterparty, amount, description. Covers ~70% of standard operations (rent, payroll, subscriptions)
2. **OpenAI fallback**
   For non-standard transactions: GPT analyzes the description and suggests a category with a confidence score
3. **Self-learning**
   When an accountant makes a correction, the system memorizes the pattern. Over time, rules cover more and more cases

Result: Match mode (Manual / Ruler / AI with confidence %) — accountants focus only on disputed transactions. Manual work reduced by 60-80%.

02 — Automated bank data reconciliation

Connection to client banks via Nordigen API (PSD2, Open Banking):

WHAT IS RECONCILED	HOW
Statement ↔ Accounting	Matching by date, amount, and description. Fuzzy comparison to handle differences between banks
Discrepancies	Unmatched transactions are highlighted — easy to see what was missed or not posted
Multi-currency	Conversion at the Riksbank exchange rate on the transaction date. "Convert to currency" button

03 — Validation against Swedish tax rules

VAT validation

VAT 25%, 12%, 6%, 0% by operation type

VAT Account

Linked to the Swedish BAS chart of accounts

Validations

Errors caught before period closing

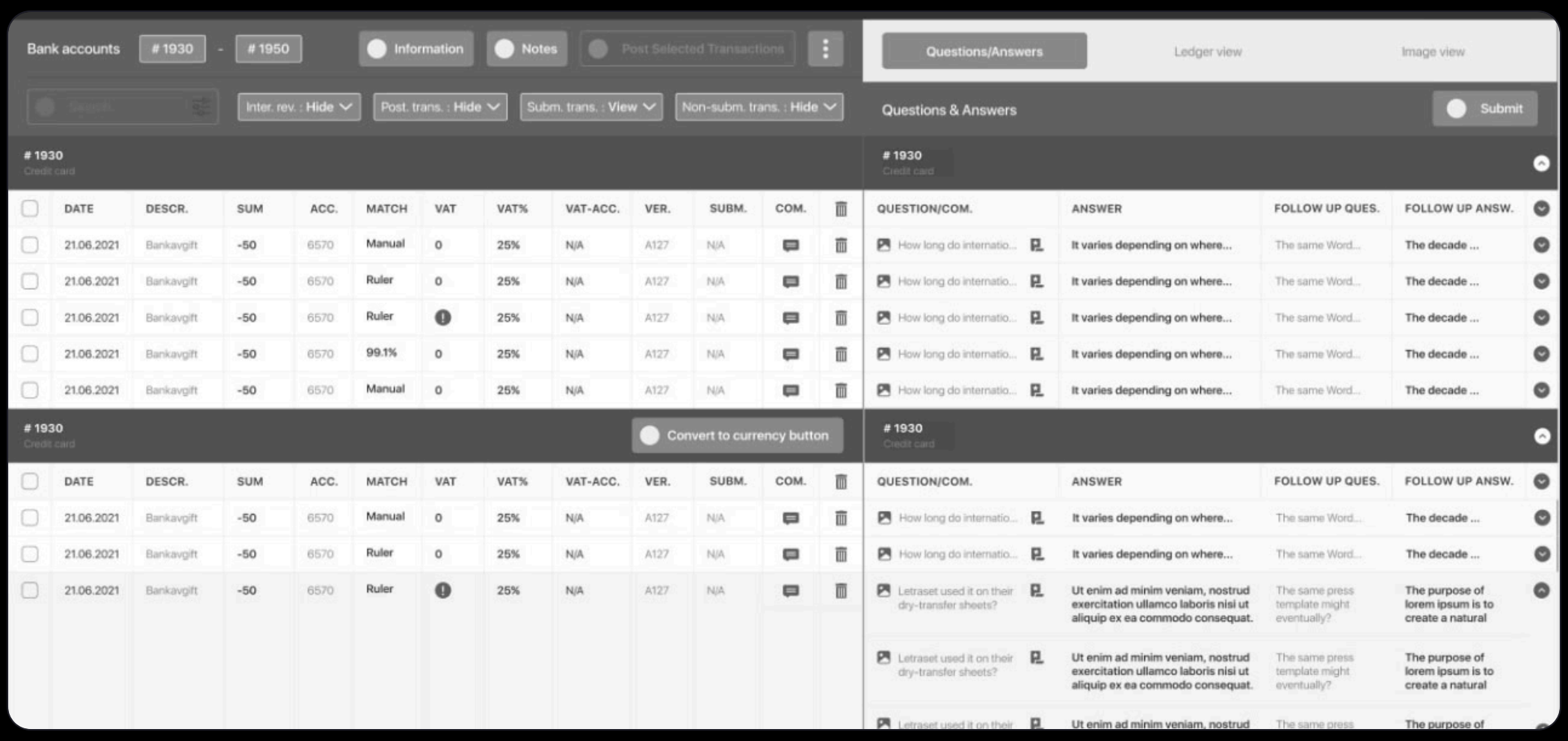
VER

Verification numbering (A127)

INTERFACE

Dual-panel transaction reconciliation

Left — journal entries (date, amount, account, Match, VAT, verification). Right — Q&A panel for auditor comments.



ELEMENT	PURPOSE
Bank accounts (#1930-#1950)	Filter by bank account range in the Swedish BAS chart
Match (Manual / Ruler)	Categorization method: manual or rule-based. Confidence: 99.1% for AI-match
VAT / VAT%	VAT rate (25%) and VAT account. Errors marked with a warning icon
VER (Verification)	Verification number (A127) — audit trail for each journal entry
Questions & Answers	Questions from the accountant to the client, responses and follow-ups for disputed transactions

TECHNICAL TRANSFORMATION

Migration MSSQL → PostgreSQL

1 **SQL audit**

Review of raw SQL and stored procedures for MSSQL-specific syntax (TOP, NOLOCK, GETDATE)

2 **Entity Framework**

Switching EF Core to Npgsql. Adapting migrations and seed data

3 **Data migration**

Migration script with validation: FK, unique indexes, data types

4 **Testing**

Parallel runs on both DBMSes, result comparison, load testing

CI/CD Pipeline

01 **Build**

ASP.NET + Angular on every push

02 **Test**

Unit + integration, coverage threshold

03 **Docker**

Multi-stage build, image size -60%

04 **Deploy**

Auto staging, manual prod, rollback

INTEGRATIONS

Swedish financial infrastructure

SERVICE	INTEGRATION	PROTOCOL
Nordigen (GoCardless)	Open Banking (PSD2) — import of transactions, balances, account details	REST API, OAuth 2.0
Skatteverket	Tax authority — VAT rates, counterparty tax IDs, reporting rules	REST API
Riksbank	Central bank — current and historical exchange rates	REST API, daily sync

STACK

Technology stack

LAYER	TECHNOLOGIES
Frontend	Angular, TypeScript, RxJS, Angular Material
Backend	ASP.NET Core, C#, REST API, JWT authorization
ORM	Entity Framework Core (Code First, Migrations)
Database	PostgreSQL (migrated from MSSQL)
AI	OpenAI API — categorization of non-standard transactions
Infrastructure	Docker, CI/CD, Serilog (structured logging)

RESULTS

Business outcomes

2.5 yr
In production
serving clients

40%
Faster load times
with server-side caching

MSSQL → PG
Migration with
zero downtime

- Eliminated MSSQL licenses → PostgreSQL in Docker
- Daily deployments instead of weekly
- AI categorization automates routine work

- Tests for critical calculations (VAT, reconciliation)
- Compliance: validation against Skatteverket
- 40% faster load times with server-side caching