

Case Study



Current

Modern Banking Products Fueled by a User-Centric Approach

INDUSTRY

Financial Services

USE CASE

Retail Banking

GOAL

Create a new banking paradigm to better serve customers.

CHALLENGE

Traditional banking uses an account-based structure that focuses on growing the account.

SOLUTION

Create a fresh relationship-based approach to banking to empower individuals and families.

RESULTS

- Rapid innovation of flexible banking products
- Fast lookups fuel user experience

Current wanted to create a modern banking experience that broke with traditional account-centric banking. They use the Neo4j Graph Database as a flexible framework for building innovative banking products that leverage relationships.

The Company

[Current](#), a leading US challenger bank, serves people traditionally overlooked by the financial system. The bank has more than 2 million members. Current is committed to providing its members with faster and better access to their money with early direct deposits, free overdraft, points for cash back, and access to free ATMs nationwide. Their proprietary Current Core technology enables them to provide banking services without overdraft fees, minimum balance requirements, or hidden fees.

The Challenge

Traditional banks organize their products around accounts. This means, for a single household, bank products and divisions – such as home equity lines of credit, checking accounts, credit cards, and more – might compete against each other.

Current wanted to try a user-centric approach to fuel new kinds of banking products that meet people where they are, with flexible savings, spending management, and ultimately faster access to their money. “Traditional banks are focused on growing the account, but Current is focused on the user and improving their financial outcome,” said Trevor Marshall, CTO.

At the same time, financial services is highly regulated and requires strict guarantees around transactions. This would impact Current’s technology infrastructure and data ecosystem. “Because we have strong requirements around data consistency, the main contender was a relational database,” said Marshall.

However, queries that find all the relationships and qualified signatories for a minor account take too long in relational databases. “If you’re modeling on Postgres, you’re creating new tables in memory as you’re creating JOINS, and that can cause a ton of latency and unexpected performance issues,” said Marshall.

Current needed the robust stability of traditional databases without their limitations. The solution also had to fit their cloud architecture, Google Cloud Platform (GCP). “All of our architecture is tightly intertwined with GCP,” said Marshall. “Everything is hosted on Google Cloud in Kubernetes deployments. We’re using a lot of native Google technology like Google Pub/Sub and Google Dataflow.”

Case Study



“With Neo4j we get everything we need to operate a very high efficiency, high performance cluster.”

–Trevor Marshall
CTO, Current

The Solution

The Current team chose [Neo4j](#) early on. “There’s a lot of interdependency of where our product is today with the store that we’re using, and all the guarantees and features that Neo4j provides,” said Marshall.

“We’ve modeled our total data model for our core banking engine directly into Neo4j,” he said. “The main use case for us is being able to quickly traverse to different data elements, and we’re able to encode the [data model](#) as a primitive directly into the database.”

Current’s first product changes the way teens bank and learn about money. Traditionally, a parent opens an account for their teen. That account-centered structure is inflexible.

Contrast that with Current’s Teen Banking, where a parent adds a product and connects a teen. “We map that into Neo4j by creating user nodes, and then connecting relationships between that user and the products they can access,” said Marshall. “This led to a completely new type of account structure that is far more flexible and more adaptable to the modern family.”

Fast development cycles are critical. “Neo4j has given us a ton of developer velocity, because there’s a very clear mental model that developers have when they’re interacting with a user’s graph,” said Marshall. “That mapping of the data model to the way that the data is actually architected in the database makes it very clear for any developers who join our team to understand the significance of the node, the relationship and how that all comes together.”

The Results

Current put its deployment on Google’s powerful platform to ensure scalability. “We’re running pretty much the fastest processors you can put in a cloud environment, and with very fast memory. With Neo4j, we get everything we need to operate a very high efficiency, high performance cluster,” said Marshall.

Fast lookups drive the user experience. “It’s all about traversal and entry point into the [graph](#). If you are authorizing a card, we can enter a user’s graph through a card, which maps to a product, which gives us user information, wallet information, and balance information – that’s an extremely fast lookup, a few milliseconds at most,” said Marshall.

Current operates Neo4j inside a VPC. “All of the communication is very secure, and that gives us a lot of flexibility to be able to connect to our Neo4j cluster from many different parts of our architecture,” said Marshall.

Current continues to innovate, having introduced the first points rewards program for a debit card in the US. “The architecture that we’ve defined is very flexible,” said Marshall.

The graph data model speaks to the business. “My favorite part of Neo4j is really that visual component. That translates not just to the way that we write code, but also the way we explain our products to the company, investors, and the broader community.”

Neo4j is the leader in graph database technology. As the world’s most widely deployed graph database, we help global brands – including Comcast, NASA, UBS and Volvo Cars – to reveal and predict how people, processes and systems are interrelated. Using this relationships-first approach, applications built with Neo4j tackle connected data challenges such as analytics and artificial intelligence, fraud detection, real-time recommendations and knowledge graphs. Find out more at [neo4j.com](#).

Questions about Neo4j?

Contact us across the globe:
info@neo4j.com
neo4j.com/contact-us