Cockroach Labs

CASE STUDY

How Form3's real-time payment platform survives cloud outages

FORM3



FORM3

How Form3's real-time payment platform survives cloud outages

INDUSTRY:
Fintech

CHALLENGES:
Building a payments platform that can survive an entire cloud failure.

SOLUTION:
A highly scalable, adaptable, resilient, multi-cloud platform built on CockroachDB.

About Form3

When you send money to another person who banks with a different financial institution, the banks talk to one another to confirm each account holder is legit. Then they physically move the money. Sounds simple, right?

It can be pretty simple, until you serve a global customer base and need to navigate different payment schemes (the rules for performing payment transactions) in different countries. Form3 was founded to simplify global transactions through a managed payments infrastructure with a single API that helps banks adhere to payment schemes wherever they do business. Customers don't have to worry about integrations, security, or reliability – it's all handled by Form3's real-time payments platform.

Building a cloud-agnostic, enterprise-grade payment technology using microservices and Kubernetes is no easy undertaking. Especially when that platform also has to be highly scalable, adaptable, and resilient to cloud failures — which is why Form3 turned to CockroachDB.

Watch: Rogger Fabri, Lead Engineer at Form3

3

PUBLIC CLOUD PROVIDERS

700

TRANSACTIONS PER SECOND

24/7

AVAILABILITY

Why a single cloud provider was never an option

Founded in the UK, Form3 started its business to cater to banks and fintechs that do business in the UK and EU (and recently, the US market). Within these regions there are five different payment schemes that must be integrated. While they all have the same function — transferring money — the implementation, requirements, and payment schemes are all different for each one. What's more, these schemes are constantly changing.

For example, when connecting to the UK's real-time Faster Payment System (FPS), companies are required to have leased lines, physical HSMs, and other hardware, which means they cannot operate exclusively in the cloud. This is a challenge especially for digital-first mobile banks: to integrate to FPS, they would be required to have their own equipment, connect to physical data centers, and continually manage the integration's frequent changes. Form3's platform handles all of this complexity for them, satisfying all related regulatory requirements.

Beyond reconciling different regional payment schemes, Form3 must also adhere to a variety of regulations and legislations. In the UK, the PRA (Prudential Regulation Authority) mandated that every bank and financial institution must have a formal cloud exit strategy because one public cloud provider's unfortunate event — a major outage, sudden steep price hikes, etc. — could have dire consequences for the entire UK economy. And because Form3 provides technology to banks, they also needed to find a cloud exit strategy.

Form3 originally selected AWS as their cloud provider and started off building on Amazon RDS for PostgreSQL, which initially worked well for their needs. They liked that PostgreSQL was reliable and could deliver strong data consistency for payments. However, they didn't like that they had to take their entire platform offline for upgrades. Any amount of downtime for payments was unacceptable.



Given all the emerging regulations and their need to provide an always-on payment platform, they thought there was a better solution: Don't depend on any single cloud provider. Instead, run across multiple clouds at the same time. This way the platform would be able to survive even a full cloud provider outage. Going multi-cloud, however, meant changing to a database that could run across all three major cloud providers, as well as hybrid/on-prem.

In summary, the requirements for this new database were:

- Cloud-agnostic supports multi-cloud
- SQL compatibility
- Easy to run and maintain
- Strongly consistent
- Scalable
- Easy to run with Kubernetes

Distributed databases like Aurora and Spanner met a lot of these requirements, but they are both tied to a particular platform vendor and are not cloud agnostic. Other databases that provided the necessary scale and performance (like Cassandra and Redis) didn't offer data consistency. Ultimately, satisfying their full set of requirements led Form3 to CockroachDB.

CockroachDB is cloud-agnostic, check. Comes with built-in horizontal scalability and low latency, check. Fulfills the consistent storage requirements, check. Finally, CockroachDB can natively run across Kubernetes clusters. Very few relational databases have the same distributed nature as Kubernetes, and running a non-distributed SQL database on Kubernetes can make deployment and management difficult.

It's the best of NoSQL in terms of horizontal scale, and the best of relational in terms of ACID compliance and write consistency so we don't lose any payments. It's the only offering we saw on the market that really solves that problem. And coupled with the fact that it's super easy to run in Kubernetes, CockroachDB became an easy choice.



Kevin Holditch

VP of Engineering: Platform, Form3

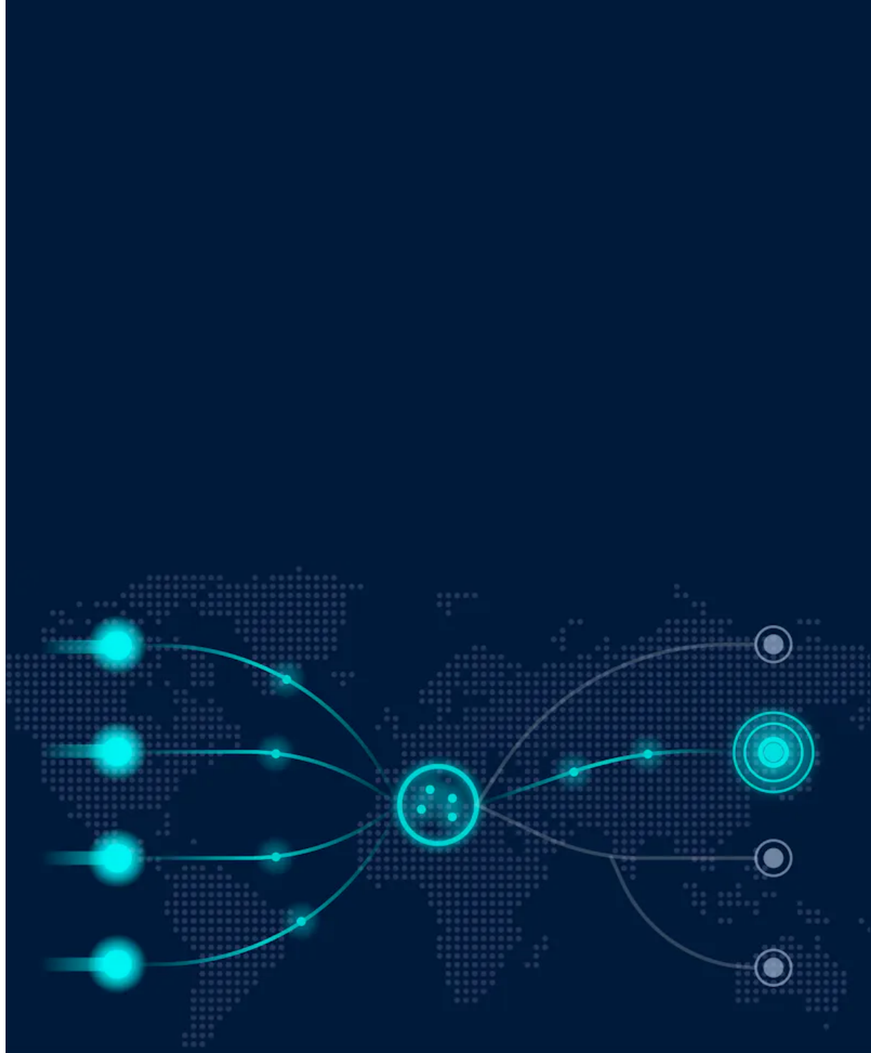
One platform, three clouds, zero compromises

While Form3 aimed for a cloud-first strategy, connecting to certain payment systems required a blend of on-premises hardware and public cloud infrastructure. By adopting CockroachDB, they designed a deployment architecture centered on operational resilience. This model treats individual cloud providers like standard availability zones in a traditional setup: active, active, active, with zero single points of failure.

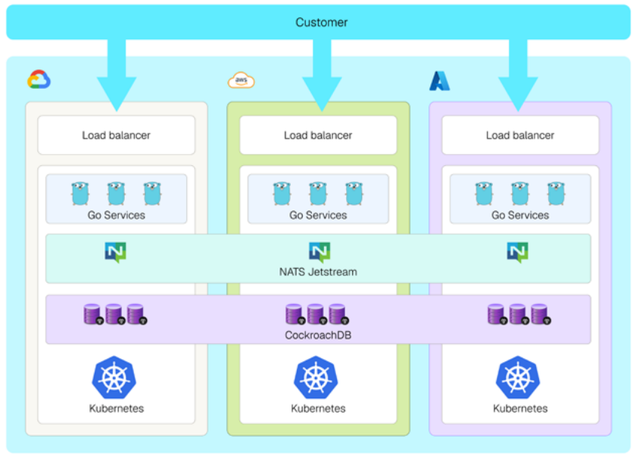
Instead of maintaining separate platform replicas on each cloud, which would have created the headache of managing diverging codebases, Form3 operates a single logical platform across AWS, GCP, and Azure simultaneously. Each provider hosts its own managed Kubernetes cluster, kept entirely independent to ensure that a localized failure never triggers a cross-cloud domino effect.

The key to this multi-cloud coordination is private cross-cloud networking. By giving Kubernetes pods unique IP addresses on the VPC host network, Form3 allows pods in different clusters to communicate directly across cloud boundaries. This foundation supports two logical clusters that span all three environments:

- **CockroachDB** — the write-consistent data layer that guarantees every payment is securely stored with strong consistency, regardless of which cloud it lands in.
- **NATS JetStream** — a high-performance messaging system that routes asynchronous payment tasks across providers. A transaction can start in GCP and finish in AWS; to the platform, it's all the same.



For the database layer, Form3 deploys nine CockroachDB nodes—one per availability zone within each cloud provider. Using the Raft consensus algorithm, the cluster requires a five-node quorum to remain operational. This allows Form3 to withstand the total failure of a cloud provider (leaving six active nodes) or even lose an additional AZ while maintaining full database availability. You can see a high-level overview of this setup below:



To safeguard this resilience during routine updates, Form3 addressed a limitation in standard Kubernetes. Native PodDisruptionBudgets (PDB) only see the local cluster, meaning multiple clouds could theoretically take a node offline simultaneously. In that state, if a cloud provider also went down, the cloister would drop to just four active nodes below the five-node quorum required to stay online.

Form3 engineered a custom Cross-Cluster PodDisruptionBudget (X-PDB). This Kubernetes extension coordinates disruptions across all three clusters; if GCP is undergoing a node rollout, AWS is prevented from taking any pods down until GCP is healthy. This ensures at least eight of nine nodes are always active, even during maintenance.

Performance is just as critical as uptime. Form3 manages concurrent traffic of up to 700 TPS while adhering to strict real-time SLAs. CockroachDB's topology-aware features allow them to route requests to the node nearest the customer such as a London node for a UK-based bank while keeping a fully synced replica in a different cloud as a safeguard.

Managing backups across multiple clouds adds another layer of complexity. Form3's RPO requirements demand incremental backups every five minutes. Working closely with the Cockroach Labs engineering team, they developed locality-restricted backups — a feature that enables staggered backups across two clouds simultaneously, so a cloud outage during a backup window never leaves Form3 without a viable restore point.

CockroachDB is not just a component of this platform, it's the whole backbone of our payment processing engine. Luckily it is very reliable and very scalable so I can go to sleep knowing we won't be paged overnight about an issue with a cluster.



Rogger Fabri

Lead Engineer, Form3

Easy to migrate, hard to break

When it came to migrating from PostgreSQL to CockroachDB, it was a “fairly easy process” for the Form3 team. Since CockroachDB is PostgreSQL wire-compatible, Form3’s data structures and queries just worked and only a few needed tweaking. Another factor that helped streamline the migration: Most of the Form3 platform is written in Go. A majority of existing PostgreSQL tools also work with CockroachDB, so the team could leverage the Go drivers they were already familiar with. Finally, CockroachDB also has a Kubernetes Helm chart available, packaged up nicely for an easy install.

The Form3 team recommends setting up a good test suite and running end-to-end tests that include the database. They even tried dropping the database connection to ensure that the software handled the failure correctly. Finally, before completing the migration, they tested latency to make sure that their payment journeys were rock solid and met all SLAs.

The Form3 team also highly recommends leveraging managed Kubernetes. Otherwise, you might end up needing an army of experts to run and maintain the platform and there’s no competitive advantage to managing it yourself. If Kubernetes goes down, Form3 has enterprise support with the cloud vendor so they can get someone to help. And, since they are using CockroachDB, they are not locked into a single provider, which does give them a competitive edge.

Now that the migration is complete, Form3 continues to improve its platform and engages in a practice called chaos engineering. They use a tool called Chaos Monkey to simulate the failures. They have scenarios that create cluster failures, cloud outages, and disaster recovery failures to see how the platform responds.

> Through chaos engineering, we are pushing CockroachDB to its limits and it recovers by itself. That’s one of the best things an engineer can see.



Rogger Fabri

Lead Engineer, Form3

The cloud outages that didn’t affect customers

Form3 has also survived real-world failures in production. In June of 2025, Google Cloud Platform suffered a massive global disruption triggered by an internal IAM service failure, causing worldwide service interruptions. This became the ultimate test for the architecture and fortunately payment processing remained completely undisturbed. The platform had successfully neutralized a major cloud provider failure in a live environment, performing precisely as the team had envisioned.

In October 2026, Google Cloud suffered another disruption, and once again Form3’s systems stayed online, meaning that their customers remained unaffected. The banks, fintechs, and other companies that run their payments on Form3 were able to keep serving their own customers without interruption.

That continuity is what Form3’s customers are paying for. They chose a managed payments platform so they wouldn’t have to worry about reliability, and so that Form3 would be on the hook to handle any cloud outages. Form3 keeps that promise with their three-cloud platform, built on CockroachDB. “Seeing that work translate into continuity for the businesses that rely on us made me very proud of the team,” says Kevin Holditch, VP of Platform Engineering at Form3.

> We designed our platform to continue processing payments even when a cloud provider experiences an outage. During the recent GCP outage, that design worked as intended, allowing our customers to continue processing payments. We’ve seen this resilience demonstrated through multiple cloud outages, with CockroachDB playing a key role in maintaining service.



Kevin Holditch
VP of Engineering: Platform, Form3

Built for what comes next

Form3 is on a mission to become the world's most trusted provider of payment technology. They remove the burden of managing costly, and ever-evolving, critical payment infrastructure from their customers, which allows customer to seamlessly add payment schemes as they expand into new regions. With CockroachDB as a crucial part of their platform's foundation, Form3 can navigate emerging legislations and continue to grow their business.

Today, Form3's API-based cloud technology solutions are entrusted by financial institutions, such as Lloyds Bank, and Nationwide Building Society, for delivering mission-critical payments. For now the majority of Form3's customers are still based in the UK and the EU. However, they've already expanded operations to the US and are looking to enter other international markets — a quickly achievable goal when you're using a single logical database across all three cloud providers.

Ready to get started?

Go hands-on with 100% free CockroachDB Serverless. Spin up your first cluster in just a few clicks.

Get Started

Compare Products

> CockroachDB is doing some amazing gymnastics under the bonnet. You might think it's just a simple query, but it's actually going off to different nodes where the data is physically stored to retrieve it. The performance is really good. And then you have this tremendous scaling capability. Using CockroachDB almost feels a bit magic.



Kevin Holditch
VP of Engineering: Platform, Form3