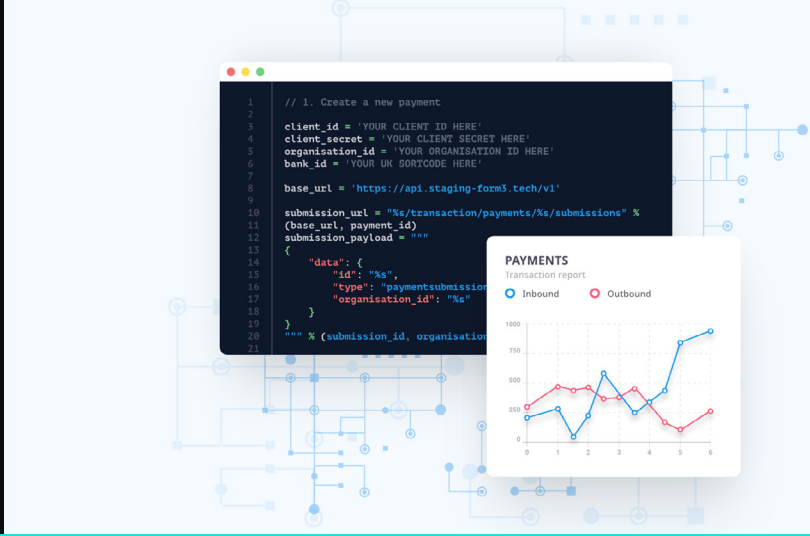
SYNADIA

How Form3 is Building a Multi-Cloud, Low-Latency Payments Platform with NATS and JetStream.

FORM3
FINANCIAL CLOUD

Case Study

FORM3
FINANCIAL CLOUD



Industry
FinTech

Location
UK / Remote workforce

Employees
350

Overview

Founded In 2016, Form3 set out to revolutionize the world of payment processing and disrupt the traditional payment infrastructure model, with an always on, cloud-native, Payments-as-a-Service platform.

Today, Form3 is trusted by some of the UK and Europe's biggest Tier 1 banks and fastest-growing FinTech's to handle their critical payments architecture. Form3's annual recurring revenue in 2021 grew by 233% and now employs over 300 people in 22 countries. Form3 has been awarded Best Cloud Payments Platform 2021 by Paytech and Fintech Finance and the most influential fintech company in 2021 by The Financial Technologist.

About Form3

Customers: Square, Mastercard, Lloyd's, Barclay's, Nationwide, Ebury, LHV

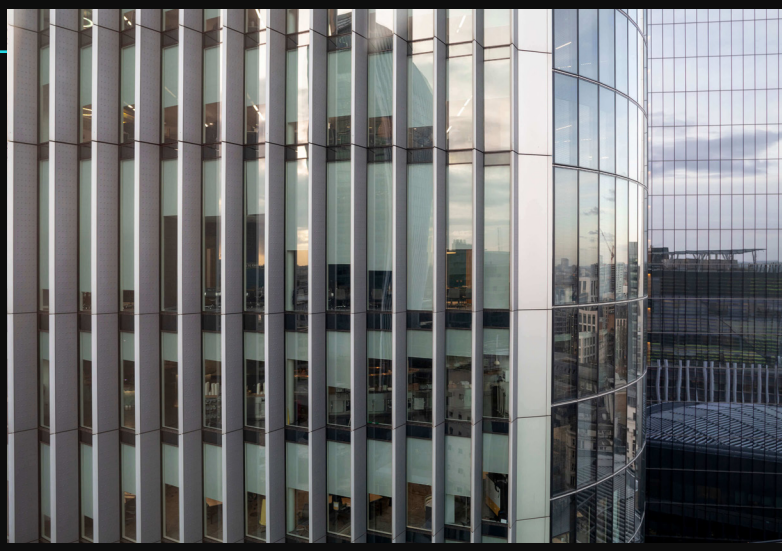
Tech Stack: Logz, Linkerd, Docker, NATS, Kubernetes, K3S, HashiCorp Vault, AWS, CockroachDB, PostGres, Linux

Application Language: Go

SYNADIA

Background

Back-end and inter-bank payment processing is a highly regulated business that is expensive, infrastructure-intensive, and suffers from a slow pace of innovation. Form3 sought to radically improve on the old payments processing technology model by moving the entire operation into the cloud and designing a cloud-native application structure with microservices. “When we are talking about banks, it’s not the most modern stack,” explains Adelina Simion, a Technology Evangelist with Form3. “To have a platform fully hosted in the cloud, that is cloud native, is a huge step forward.”



Form3’s service sits in between banks and the external payments processing systems such as Swift, BACS, Faster Payments and SEPA. Each of these systems has their own API and message definitions. The multiple standards and requirements generate complexity and require considerable maintenance and troubleshooting. Form3 provides a single payment processing API and abstract away the complexity, allowing banks to consume payment processing as a service. “We take away the burden of multiple integrations and we can do this at scale and in the cloud,” explains Simion.

> *“We take away the burden of multiple integrations and we can do this at scale and in the cloud.”*
>
> Adelina Simion, a Technology Evangelist with Form3

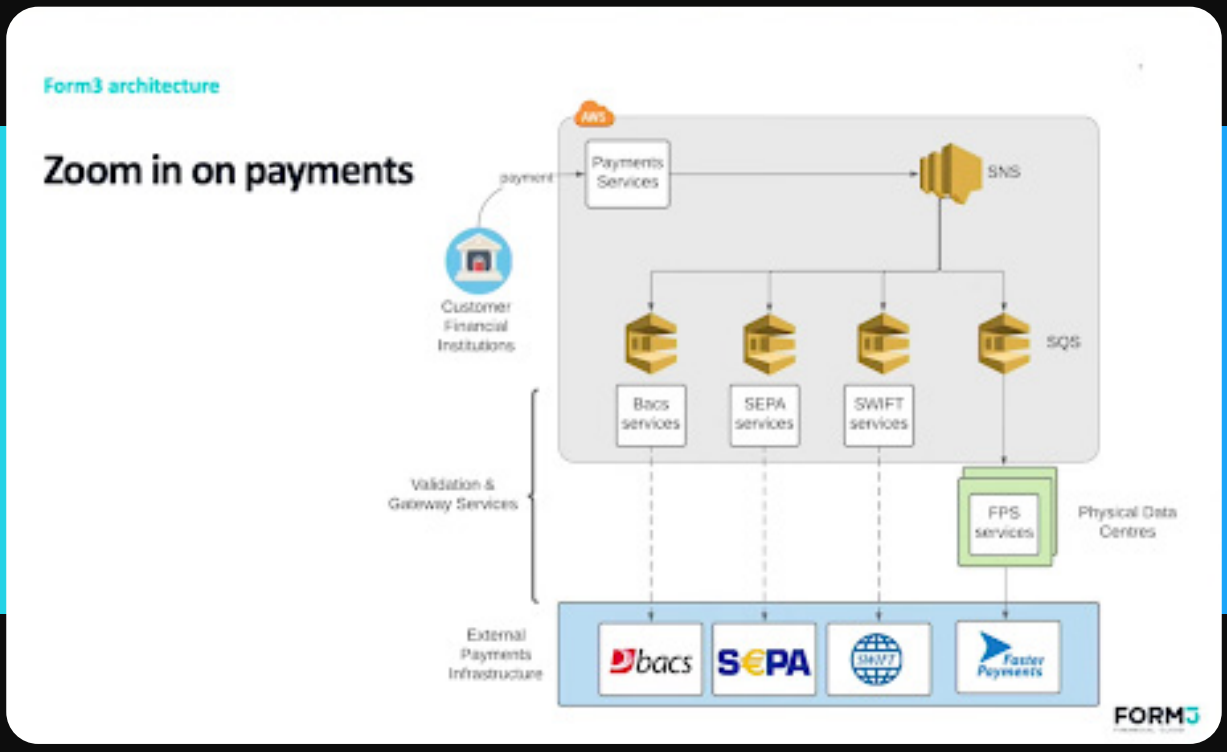
Architecturally, putting payments processing into the cloud makes it much simpler for Form3 to stand up and expand a truly distributed application infrastructure. This would push them closer to end users, reduce latency, and improve availability. Unlike legacy payments processing technology operations that must build multiple redundancies at the hardware layer, Form3 could design for even greater redundancy and resilience in the cloud. Because Form3 engineers maintain and develop the platform, clients get the benefit of continuous improvements, unlike older platforms that rarely upgrade. By designing the Form3 application with a cloud-native architecture running on Kubernetes, with loosely coupled microservices tied together via APIs, Form3 could more quickly innovate and incrementally iterate on different aspects of their platform without impacting platform performance or end users. Form3 wanted to build their application with as many open source components as possible, like the Kubernetes container orchestration application and Linkerd, an open source service mesh.

SYNADIA

The Form3 Application Architecture

The application is written in Go and composed of multiple services running on Elastic Kubernetes Service (EKS) with the Linkerd service mesh handling microservice routing, observability, and performance guarantees. Customer financial services institutions and fintechs send a payment processing request to the Form3 API, which is hosted in AWS. Form3 translates the request into their internal request format and uses AWS' Simple Notification Service (SNS) and SQS to create a paired pub-sub event bus for processing messages. In this system, payment requests "fan out" to one of the four supported external payment processing systems (SWIFT, BACS, SEPA, Faster Payments). Form3 used the AWS Simple Queuing Systems as a pub-sub system to define rules for processing the request, including recipient, retries, and more. Form3 validates each payment request and provides a secure gateway connecting to those services.

For Faster Payments, the regulations require integration to the scheme from physical Data Centers. Form3 is therefore required to lease physical communication lines and hardware. "We previously had a third-party doing that for us but as we were growing, we wanted more control over our ability to scale up. To do this, we made our own physical data centers with leased lines in it." Both in AWS and the FPS-dedicated data center, the Form3 application runs on Kubernetes.



SYNADIA

Challenges of the Scaling, Latencies, and SLAs

The system worked well initially until Form3's rapid growth stressed some of the internal components. Form3 was processing over 1 million Faster Payments transactions a day. The company normally processes a payment within a few hundred milliseconds. "We started noticing that during high traffic periods the SQS queue was taking as much as 300 milliseconds to process. That is a big chunk of our SLA so we decided something had to be done because this would not scale. The volumes are going to continue growing," recalls Simion.

In addition, the AWS overall uptime SLA is 99.9%. This was not sufficient for critical payments infrastructure. Says Simion, "We could not risk losing our processing capability in a cloud outage." Creating a higher availability service would require creating a multi-cloud capability with the ability to process on another cloud in the case of an outage. As more customers came to rely on the Form3 infrastructure, the company knew that it had to create more resiliency and ability to deliver service from multiple clouds with near-instant failover.

Lastly, in the physical data center set up for its Faster Payment infrastructure Form3's engineers faced limitations in the compute footprint and capacity per node of applications. Simply bridging the AWS messaging systems into the physical data center would not work. "Physical data centers have limitations on what you can actually run on them. We needed a lightweight events bus and that's where we heard about NATS," explained Simion.

To summarize, Form3 needed an event bus that was faster to reduce latency and lightweight to run in physical data centers. To back up tight SLAs for customers, Form3 needed an event bus that enabled a cloud agnostic architecture, with replicas of the Form3 application running in multiple public clouds in an active-active configuration.

PROS AND CONS OF AWS ARCHITECTURE

Pros:

- ✓ Easy to add new schemes
- ✓ Scalable cloud architecture
- ✓ SQS and SNS fan out scales fairly well
- ✓ Managed architecture

Cons:

- ✓ Cloud provider dictates architecture
- ✓ SQS and SNS set performance limits
- ✓ Performance can be variable
- ✓ Tail latency in SQS at 60% of full SLA, creating risk
- ✓ Single cloud SLAs limited latency reduction and resiliency

NATS + JetStream: Reduced Latency, Better SLAs and Cloud Agnostic Service

The Form3 engineering team began looking for an alternative event bus and messaging service and learned about NATS from the Cloud Native Computing Foundation (where NATS is a member project). NATS is an open source data fabric sponsored by Synadia. Competing with streaming and messaging solutions like Kafka and RabbitMQ, the NATS server is written in the Go programming language, just like the Form3 application. The core design principles of NATS are performance, scalability, and ease of use, also mirroring the design ethos of Go.

NATS has a vibrant community of thousands of active users and contributors. Synadia offers both enterprise grade NATS support and NGS, a global utility for NATS-as-a-Service. The Synadia team helped Form3 explore NATS and provided technical guidance during the due diligence process. Because they were providing mission critical and high availability financial services, the Form3 team weighed heavily the long track-record of NATS, which is in production use at tens of thousands of organizations, including a number of the world's most sophisticated technology and finance firms. The NATS maintainer team also has deep roots in messaging and event streams and creating cloud native architectures: NATS creator and Synadia CEO Derek Collison helped design messaging systems at TIBCO and then later went on to design the CloudFoundry PaaS project.

SYNADIA

The NATS Difference

Unlike Kafka or other messaging products, NATS can provide high throughput and low latency on a small infrastructure footprint; it is even possible to run a fully functioning NATS server on a RaspberryPI unit. NATS is also extremely reliable and does not require a restart with changes to the topology of applications or clients. The NATS community has written clients for NATS in over 40 different languages, extending support to the majority of commonly used programming languages. To expand a NATS cluster is as simple as adding an additional server. NATS can easily bridge clusters with “Leaf Nodes” In addition, NATS does not require supporting technologies (like a JVM) and is wholly independent. NATS also does not require significant time for operations or tuning, making it painless for DevOps teams. NATS includes a persistence layer called JetStream.

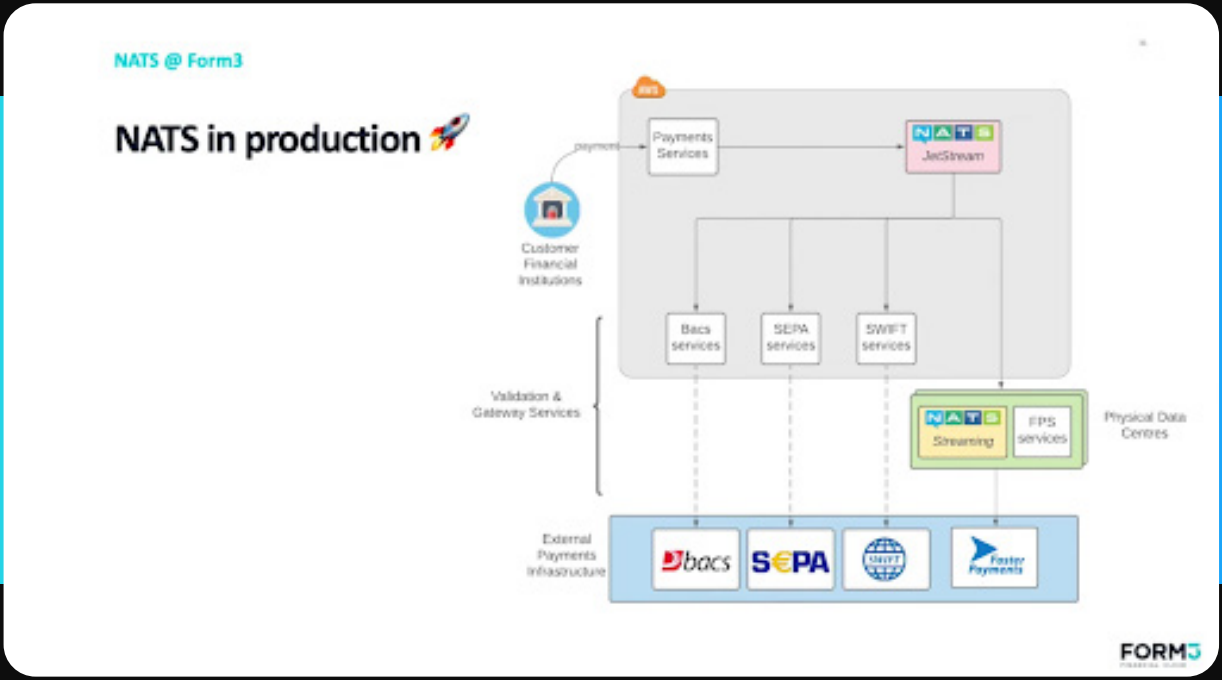
Because NATS is so simple to set up, has such low latency and high capacity per server, NATS works equally well in almost any environment and in any cloud. As a complete pub-sub system, NATS can replace both SQS and SNS with a single service. NATS also includes JetStream, an advanced, more flexible and easier to deploy persistence layer. JetStream features also include a distributed key/value database, data-at-rest encryption by default, and horizontal scalability.

Form3 Benefits and Results

- ✓ >6x reduction in latency
- ✓ Reduced infrastructure footprint
- ✓ Replaced two paid services with a single piece of software
- ✓ Lightweight enough for limited server capacity in physical data center
- ✓ Improve resiliency and higher availability
- ✓ Cloud portability and a cloud-agnostic architecture
- ✓ Data at rest encryption and security

SYNADIA

How Form3 Deployed NATS



In March 2020, the Form3 engineering team tested out NATS and NATS Streaming as a replacement event bus for their SQS and SNS pairing. They quickly determined that NATS met their expectations in terms of latency, reliability, resilience, and compute requirements. Form3 worked closely with the Synadia team to ensure that the planned architecture would maintain low latency while scaling up our payments.

After validating the new application architecture with NATS, Form3 began its replacement of SQS and SNS with NATS. The top latency for messages immediately dropped to less than 50 milliseconds between locally hosted servers, an improvement of more than 6x. In fact, the Form3 engineering team found that NATS could process higher volumes of transactions faster with a much smaller infrastructure footprint (CPU and RAM)i.

Form3 also deployed NATS for the Faster Payments physical data centers. NATS fit into the limited footprint environment of the data center without difficulty and actually enabled Form3 to handle a greater volume of payments than the previous architecture, something that will simplify and reduce future scaling costs in this more expensive environment

SYNADIA

After the successful deployments and the realization of how much faster NATS was than the SQS-SNS pairing, Form3 realized they could actually use NATS to create a cloud-agnostic architecture leveraging JetStream and its highly configurable persistence layer. In the near future, Form3 would deploy an active-active configuration across multiple clouds, with the application linked via NATS and JetStream to create a true multi-cloud cluster. Workloads could easily shift from one cloud to the next, using NATS JetStream and Leaf Nodes, a method that NATS uses to easily bridge NATS servers across clouds, Kubernetes clusters, or other environments. (Leaf Nodes are also commonly used for edge computing applications).

This new Form3 architecture also replaces multiple PostGres databases hosted in AWS with CockroachDB, a cloud native database designed for multi-cloud deployments. NATS JetStream makes this architecture considerably more efficient by updating and storing payment processing status information in the actual event stream. This reduces the need to trombone data back and forth beetween the application to the database. “Every time something new happens to a payment, there will be a new message in NATS. There will be no hammering databases to constantly read updates,’ explains Simion. With this multi-cloud active-active architecture, Form3 will be able to achieve higher uptime than any single cloud could guarantee. “It lets us break free from the SLAs of any one cloud and create our own,” says Simion. “None of this would happen if we couldn’t have a real cloud agnostic event bus, communicating across multiple clouds.”

> *“It lets us break free from the SLAs of any one cloud and create our own. None of this would happen if we couldn’t have a proper event bus across the clouds.”*
>
> Adelina Simion, a Technology Evangelist with Form3