

## CIO JOURNEY

# Fannie Mae

*Transforming Critical Financial Infrastructure  
Behind the U.S. Economy*

---

|                 |                    |                   |               |
|-----------------|--------------------|-------------------|---------------|
| <b>Company:</b> | Fannie Mae         | <b>Revenue:</b>   | \$110 billion |
| <b>Sector:</b>  | Financial Services | <b>Employees:</b> | 7,200         |
| <b>Driver:</b>  | Bruce Lee          | <b>Countries:</b> | 1             |
| <b>Role:</b>    | CIO                | <b>Locations:</b> | 8             |

**Company IT Footprint:** Fannie Mae has 10,000 people and over 10,000 servers in one or two data centers. It connects to about 2,500 banks and institutions, and to about 40 market providers of data or other types of services. Fannie Mae manages petabytes of data and 400 different applications. The complexity of its systems and infrastructure is not on the order of an international bank, but it is vital—it's systemically important.

---

*“The point of this journey is to increase the resiliency of the company.”*

**Bruce Lee, Former Chief Information Officer, Fannie Mae**

# Fannie Mae Journey Overview



## Business Objectives

- Increase resiliency
- Establish clear IT strategy direction: agility, velocity, cloud
- Reduce app configuration complexity
- Shift to Office 365
- “Fix the foundations”



## The Solution

- Engage legal team for buy-in
- Create collaborative DevOps ecosystem
- Transform network and security infrastructures
- Establish direct-to-internet connectivity with local internet breakouts
- Standardize on cloud operating models, configurations



## Impact

- Reduced complexity: from 38 data-replication models to two
- Better user experience, less latency, no backhauling
- Successfully deployed Office 365
- Improved control, visibility
- Greater flexibility, resiliency
- Enhanced security

Fannie Mae is one of the major financial services organizations that underpins the economy of the United States. Bruce Lee, formerly the Chief Information Officer at Fannie Mae, describes the steps the organization took as it transformed its IT practices to a cloud model.

## In the words of Bruce Lee:

---

“

When we began our digital transformation journey here at Fannie Mae, we took a long hard look at what kind of company we were.

We are a Fortune top 25 company with a three trillion dollar balance sheet and 14 billion dollars in profit.

I have been an IT person my whole career. I started off creating trading applications for banks in London. We were the disruptors back then, using PCs instead of mainframes. From there, my career has followed the disruptions in the financial space; in the derivatives world of interest rates and then foreign exchange. I was at HSBC until about 2012, when I got an opportunity at the New York Stock Exchange, and I thought, “Well if you really believe in technology’s power to transform the market, we’re witnessing that with high-speed trading.” So I joined an industry in transition.

I came to Fannie Mae in 2014, when I saw that the mortgage market was transforming. The way that mortgages were created, serviced, and securitized was changing. The mortgage industry was probably the last of the financial services industries to get a real dose of technology transformation. In the past four years, we have been fundamentally rewriting the mortgage industry in the United States from our position as a secondary mortgage provider.

We have both the trading side and a B2B side to manage. The ecosystem is a big platform that does not look dissimilar to an Uber or Airbnb in that our job is to connect excess capacity—the world’s financial capital—to excess demand. We just had to take that platform and renovate it. That’s what we’ve been doing for the last few years.

At Fannie Mae, we are deemed part of the nation’s critical financial infrastructure because we move so much money, we connect so many things. We have an out-sized commercial impact, yet we are not that large in terms of people and servers. We manage petabytes of data and 400 different applications.

When it comes to starting down this path of digital transformation, I don’t think CIOs spend enough time answering the questions: “Where are we today? Where are we today? Where do we want to be? And, how do we start the journey from here to there?”

---

*“I don’t think CIOs spend enough time answering the questions: ‘Where are we today? Where do we want to be? And, how do we start the journey from here to there?’”*

---

When I joined, we had a lot of software development being done in the classical waterfall approach. We had five separate projects in which we were investing 100 million dollars a year, each. When you look at the track record of such large IT projects, you find there is a 96% failure rate, making them extraordinarily risky.

We had a lot of departmental Sun boxes running Solaris, which meant a lot of application concentration onto individual servers. Most of our people were no longer programmers, but rather had become vendor managers because the development had been outsourced. We’d lost the core ability to engineer and architect. We’d become captive to our vendors in a very dysfunctional way.

That hard look was the start of our journey. While we found many areas we needed to improve, we found pockets of people who still have that imaginative view of what the future can be. I listened to them, as a new CIO must.

### **Defining a strategy**

The main message was that we needed clarity of direction. We set out and made five bold statements about our IT strategy. One of them was that we would partner more closely with the business and make releases for structural applications every six months. This excited the agile team, but it scared the waterfall guys to death—but it also got everyone to a place in their heads where they said, “We’ve got to go faster.”

Another goal of the new strategy was that we were going to embrace the cloud where it made sense. Stating that helped overcome the objections of the traditional IT forces internally.

A third objective was to build a team internally that could power our digital transformation by being core to the business. That means acquiring the skills, bringing in talent, pushing our vendors and outsourcers further away from us, and internalizing more of the work. This is a typical arc you will see in most agile digital transformations: you have to own more of the people yourself and you have to be more self-sufficient in software engineering and design. We have that as a goal.

We put another goal first that we called “fixing the foundations.” That meant putting in place fundamental security and architectures that recognized how critical data

was. At the end of the day, data is what matters most to us. Beyond cloud, beyond security, beyond everything. Who has that data? How accurate is it? And how can it be relied on? What intrinsic value does it have that a company like Fannie Mae can stand behind? These foundational improvements included partnering and agile delivery. It was adopting the cloud. It was sorting out the data and building the team to do it all.

## Adopting SaaS

Moving to Salesforce led to interesting conversations with our business. The business wanted better Customer Relationship Management (CRM) tooling, but they were looking at it through what I call an old-world view. We pushed them to realize that they did not want a better tool for creating tickler notices to call a customer—what they really needed was a customer engagement tool. You want an environment where interested customers can find what they need on their own. You want to create a world that allows our own internal data view to intersect with the customer’s view of themselves.

These conversations occurred when we were executing on our objective of IT partnering with the business. We evaluated other tools but ultimately went with Salesforce for CRM. We adopted Salesforce and immediately had to learn a valuable lesson, to learn to resist the temptation to over-customize SaaS tools—to try to make them fit our old way of thinking. We even had a group that redesigned the way the Salesforce interface worked until the end users asked, “Why does it scroll back and forth instead of up and down?” We learned to abandon that stuff and work with what Salesforce delivered.

## Migrating internal applications

As we embraced the cloud more generally, we had to look at how to effectively move our own applications from our data centers to AWS. One of the challenges people have is that they underestimate a long-held core corporate tenet: that the infrastructure will be perfect, and applications can be written with that assumption. We will provide highly available clusters, and automatic disk mirroring, monitoring, and redundancy at the hypervisor level. We’ll have transaction integrity maintained on the database backend, and the VMs will never go down. Because of that, application developers did not have to code resiliency into their applications.

In my experience, your approach to the cloud has to be that something can go wrong. VMs are easier to move around. They need to automatically recover. You have to worry about the state of your data and the multiple states it can be in. Basically, you have to think about all aspects of not having a perfect infrastructure. You can't rely on speed, for instance, as it will vary. In the internal corporate network, you spent a lot of time tuning everything to make sure that a transaction will never take more than 100 milliseconds. Because of that, you could guarantee what throughput would look like or know that two updates will be done close enough together and you won't have a data integrity problem.

With cloud, you can't take any of that for granted. You have to program for what it is and what happens if it slows down. The Intel updates for the Specter and Meltdown bugs that Amazon rolled out are a great example. Everything slowed down, and you had to adjust for that.

The interplay between hardware and software is much more loosely coupled in the cloud. That's what developers have to program for.

During 18 months of "test and learn", we tried to take a lot of corporate standards and design principles into AWS, and it was a disaster. We had to regroup after 18 months and implement a native cloud model rather than try to duplicate what we had in the data center.

We had anticipated this learning curve when we started, but we still had to go through it so people's hearts and minds would come over. They had to experience why it was difficult, why their paradigm doesn't work, and why they had to learn a new one.

I think of cloud migration for applications in the following ways. Corporate applications like HR systems and payroll systems should go to the cloud. Get the things that are not core to your business into the cloud. It's painful if it slows down, but it's not a problem. Then you have the other end of the spectrum, which is highly variable compute. Lots of compute, lots of data, but highly variable loads. That's another use case where you should definitely go to the cloud.

The challenge is the migration of your core transactional systems, your legacy Sun Solaris, Oracle transaction flows, the things that go up and down and are interlinked

---

*“You may leave your data where it is in the data center, but you have to be smart about all the hops it takes to get to it.”*

---

end to end across the whole company. There may be as many as 40 to 50 applications in a single business value chain. Decomposing that so the pieces can move to the cloud and be programmed in such a way that their variable performance does not impact your SLAs is the key. We are only now getting to see just how hard that is.

On top of creating a DevOps ecosystem, you have to figure out support. Your cloud infrastructure providers may not call you for two hours when they have a problem. In the meantime, you race around with your own troubleshooting only to discover the glitch was on their end.

### **Transforming the network and security infrastructures**

When we started our cloud transformation, we performed a network hop analysis. This application spoke to this data set internally by making several hops. It would go from the application stack through two global load balancers and down to the storage arrays and find its data. It was a two- or three-hop technical journey from the view of the execution memory to the data we needed. When we put the same data set in AWS, because it was going to be used for analytics, we discovered we increased the number of network hops by a factor of three, to nine.

As you move applications to the cloud, you have to be aware of the network paths the data is going to take. You may leave your data where it is in the data center, but you have to be smart about all the hops it takes to get to it. You will invariably be adding layers and hops. You have to engineer that carefully. That may mean doubling down on the quality and depth of the networking team in your organization.

We embraced security as a journey that ran in parallel with our exploration of our applications, data, and transaction processing. We had phases of what we would allow along the journey and what we could support. And then we looked at what comes next, and what comes after that. We were fortunate in that the CISOs were of the mind to “make it work.” We did not experience that typical battle with the security folks. They took the time to learn the AWS security stack, learn the way it works, and in many cases implement what was needed. We have assets in AWS, but we still connect back to our data centers and then out through our security stack before getting to the internet. The next step of our evolution is to put that security stack in Amazon, as well, to allow direct connections from inside Amazon to other places.

## Establishing local breakouts

We are on the verge of pulling the trigger to allow our 10,000 employees to go directly to the internet from wherever they are. We use Zscaler Internet Access (ZIA) for that. The driver is Office 365. When you make that shift to Office 365, while still backhauling everyone's traffic to HQ, the bandwidth usage goes through the roof.

The point of this journey is to increase the resiliency of the company. Moving to Office 365 means that if we have issues with our systems, our employees can still get to email and SharePoint. Because of that, we went with Azure's hosted Active Directory, removing one more thing that could fail. If people cannot authenticate, they would not be able to get to Office 365.

## Things to avoid

- Avoid saying it will be quick and easy to move to the cloud. It won't be. Just tell people it is much harder than they think.
- Try to avoid contention between developers and infrastructure people. Developers tend to jump to the cloud due to impatience with the controls in place. They don't want to wait for a server to be provisioned. They try to make the case that using the cloud is just easy.

We had to fight a lot of that at the beginning. It's natural for the developers to want to avoid working within the constraints of IT, but it always comes back to haunt you. Eventually, they have to interact with you, the security people, the data team, and network people. The myth that cloud is what drives developer productivity falls apart when you try to run anything in production for real; and run it at sustained levels; and have monitoring; and make sure it has the right backups; and that the resiliency is in place and that you've tested it; and the network doesn't get crowded out by something else.

To me, it means you are just shifting the pain to a different part of the organization: off the developers and onto their infrastructure colleagues. Developers, infrastructure, and security all have to be on the same page from the beginning.

There is no shortcut when you are building the hard stuff—the things that address real business and customer problems require integration across silos.



---

*“There is no shortcut when you are building the hard stuff — the things that address real business and customer problems require integration across silos.”*

---

## Things to do

- **The main point is to realize that your cloud migration journey, like your digital migration journey, is going to be multifaceted.** You should not think of it as one thing. You have your pure SaaS projects, your platforms like Salesforce and ServiceNow, and you have your office automation shelf like Office 365 and SharePoint. You can progress on one thing separate from the others. Moving Exchange to the cloud is a lot easier than moving a mortgage underwriting system to the cloud. One took nine months to organize, the other is going to take five years to complete.
- **Be precise in language usage.** Infrastructure as a service is different than platform as a service versus software as a service. They are all very different with different paths to success. Be especially careful with SharePoint. Customizations spread like wildfire and make it hard to migrate to the cloud. If I had to do it all over again I would have killed off SharePoint internally first.
- **The legal team has to adjust too.** They have to understand that the nature of the vendor-customer relationship has changed. They comb through contracts looking to customize them to the company’s benefit. But cloud contracts are one size fits all. The provider cannot modify them for each customer. Which brings up an important insight I had.

## Cloud introduces standards

A big “aha” moment for me was when I thought about the fact that we know intrinsically that systems—trade systems, cargo systems, any systems—all work better when you have standards. Think of railroads and standard gages. We know standards are good.

In corporate life, though, standards have become associated with avoiding the negative—security standards to prevent you from doing something stupid; database standards so you don’t do anything stupid. They’re seen as governance hurdles that limit creativity. Standards are somehow burdensome and bad. The beauty of the cloud is that it has managed to make standards sort of sexy, make them good things, because they free developers from the whole infrastructure nightmare. If your own infrastructure team tried to impose a whole bunch of standards on developers, they would hate it.

*“The beauty of the cloud is that it has managed to make standards sort of sexy, make them good things, because they free developers from the whole infrastructure nightmare.”*

Developers are OK with just ticking boxes on AWS when they set up a VM. They don't think of them as standards; they don't fret over the fact that there are only three choices of configuration. They forget that they used to specify hundreds of different configurations on the corporate side, insisting that their application is special, it's different; they need this, they need that. In the cloud, they are perfectly happy with limited choices and just tick which ones they want.

Somehow the cloud has managed to make standards acceptable. The cloud is not customized, it is not bespoke. It's a very standard environment. I used to have 38 flavors of data replication in one data center at Fannie Mae; 38 ways that application teams had to decide their applications would move data from their primary system to their backup system. 38 of them. We had to close the data center to get down to ten ways, and now we have another big project underway to get that number down to two.

This move to standardization is a good thing for the industry.



---

## Ready to transform your company?

Create business value with Zscaler today.

[CONTACT US](#)

[REQUEST DEMO](#)

<https://www.zscaler.com/company/contact>

<https://www.zscaler.com/custom-product-demo>

---

**About Zscaler** Zscaler was founded in 2008 on a simple but powerful concept: as applications move to the cloud, security needs to move there as well. Today, we are helping thousands of global organizations transform into cloud-enabled operations.

© 2019 Zscaler, Inc. All rights reserved. Zscaler™ is either (i) a registered trademark or service mark or (ii) a trademark or service mark of Zscaler, Inc. in the United States and/or other countries. Any other trademarks are the properties of their respective owners.

