We Are the Systems of the Future: Creating Legacy-Based Cloud Infrastructure Design, Not Just Code

Nigel Dsouza

We revere the new in tech—the most recent releases, the newest frameworks, and the state-of-the-art in edge computing.

Here's a bold idea, though: what if cloud computing platforms were constructed similarly to cathedrals?

Not for today. Not for this quarter's KPIs. But for the engineers and architects who will continue to use our code long after we are gone.

As someone who oversaw the architecture of enterprise systems at Fidelity Investments that handled billions of transactions, I've come to the conclusion that the most underappreciated stakeholder in contemporary cloud design is the one who hasn't been created yet: **the engineer who follows you**.

How Are They Going to View You?

Will a developer silently thank you—or curse your name—when they access your Terraform files in five years?

Will your IAM policies be self-explanatory? Will your deployment pipeline make sense without tribal knowledge? Will your system crumble with mystery or deteriorate gracefully?

The majority of us write code as if it were a temporary script.

In reality, cloud infrastructure has become institutional memory.

Archaeology is the new thing.

The Infrastructure Philosophy

Stop acting as though our systems are disposable. They're not.

They undergo cloning, forking, dockerization, scaling, and transmission.

Thus, it is imperative that we write with legacy in mind—not in the sense of "technical debt," but in the sense of **moral obligation**.

The way we refer to founding architects, elders, and forefathers.

And here is the beautiful but terrifying question:

Which values are ingrained in the design of your system?

Culture as Code

My team at Fidelity develops cloud-native alternative investment tools. We have pipes across continents. We have cross-failover zones in our DR plans.

But integrating culture into code is harder than the technology.

Do we make thoughtful comments? Do we record decisions in addition to endpoints? Do we leave wisdom—or just tools—behind?

Even though a cloud function takes only 15 milliseconds to execute, the effects can last for years.

Your Prospective Teammate is Observing

Consider writing each line of code as if it were a bottle with a message.

To the engineer who comes after. The next lead. The future you.

Not merely: "Is it effective?"
But: "Will it last?"
More importantly: "Does it teach?"

Infrastructure as a Chronological Record

We frequently discuss automation, scaling, and optimization.

But what about purpose?

The most effective solutions I've ever encountered weren't just fast or clever.
They showed **compassion**.

It makes sense. They were created by someone who cared about the inheritor.

You are not just an engineer.

You are an ancestor.

About the Author

Nigel Dsouza serves as a Principal Software Engineer and Technical Lead at Fidelity Investments, creating global cloud solutions that transcend toolchains and trends. According to him, legacy—not performance—is the most potent indicator in engineering.