

FRONTIER ENTERPRISE

A SPECIAL ISSUE BY FRONTIER-ENTERPRISE.COM

2026



FEATURES

**Standard Chartered:
Banking & Telemetry**

**Quantum Computing's
Next Frontier**



INTERVIEWS

**Prudential: Path to
Omni-Cloud Future**

FRONTIER ENTERPRISE 2026 ISSUE #1

www.frontier-enterprise.com



Photo courtesy of Prudential.

PRUDENTIAL

Prudential's Path to an Omni-Cloud Future

P

rudential's shift to the cloud led the company to reassess early assumptions about stability, cost, and simplicity. What began as a lift-and-shift migration has since become a multi-cloud architecture

shaped by abstraction, platform engineering, and increasing use of AI to reduce developer workload without removing human control. In this discussion with Frontier Enterprise, Alex Khaerov, Director of Cloud Foundation, outlines how Prudential's infrastructure has evolved and where he sees its next phase of cloud engineering heading.

What does Prudential's cloud architecture look like today?

Before the cloud era, and because we operate across multiple regions, we relied heavily on data centres of different types and sizes. Around six to eight years ago, we entered the cloud transformation phase. Like many enterprises, Prudential viewed the cloud as the natural next step. The first phase of our transformation was typical. We selected a cloud provider and conducted a full-scale migration as quickly as possible. This was mostly lift-and-shift work, replacing on-prem workloads with similar set-ups in the cloud without redesigning or refactoring. The intention was to reduce our on-prem footprint, modernise quickly, and take advantage of the expected cost and efficiency benefits of the cloud.

Many expectations existed at the time, such as the belief that the cloud would be inherently more cost-efficient or more stable, or that it would improve incident response and provisioning time. Some proved correct, especially around scalability and elasticity, but others were misconceptions, particularly the idea that stability and security would be assured by default.

What happened after the initial lift-and-shift migration?

After this migration, we entered the second phase, which I would describe as a more deliberate transformation within the cloud. This is where we started considering greenfield projects, cloud-native design, and Kubernetes. While the initial migration focused

on moving existing workloads, phase two centred on building new applications natively in the cloud.

We also realised that with so many moving parts and teams working autonomously, the old model of central provisioning had become a bottleneck. Even though the cloud offered click-ops convenience, it slowed us when everything continued to pass through traditional managed services teams.

So we shifted towards infrastructure self-service. For a large, federated enterprise, this meant defining a standardised infrastructure foundation describing identity, connectivity, default configurations, and reusable patterns. By reverse-engineering common application needs, we created blueprints representing the most repeatable infrastructure layouts. This allowed teams to operate end to end with automation, following GitOps principles and infrastructure as code.

The intention wasn't full automation without human involvement. There are still review points for security or subject-matter experts, but the process became self-paced with minimal intervention. We encouraged a fail-fast approach, where infrastructure blueprints could be improved through small, rapid changes.

This phase also brought cost optimisation and a deeper understanding of cloud usage. But it highlighted another major challenge: vendor lock-in. As teams increasingly wanted to use unique offerings like BigQuery, Cosmos DB, DynamoDB, or specialised storage, relying on a single cloud provider became limiting.

How did you address the risk of vendor lock-in?

This challenge led to the third phase: transitioning from hybrid cloud environments, meaning the cloud and on-premises, to multi-cloud.

Even after phase one and two, we still had on-prem workloads, some due to regulatory constraints, while others were latency-critical or unsuitable for the cloud. Therefore, a hybrid infrastructure was inevitable. The third phase introduced a second cloud provider, letting us choose hosting environments based on technical requirements, commercial considerations, or specific capabilities.

In multi-cloud, the estate isn't evenly divided; workloads are placed based on function. For Kubernetes, which we use extensively, capabilities across providers are similar, making cost a major deciding factor.

Multi-cloud brought new questions. Should we manage each cloud independently, or provide an abstraction layer to unify the developer experience? We chose abstraction. Developers shouldn't need to understand the underlying specifics of each provider. For commodity services like VMs or managed Kubernetes, abstraction makes the experience consistent and hides unnecessary complexity.

How did Prudential's abstraction approach evolve into a full platform engineering model?

This abstraction layer evolved into the fourth stage: infrastructure orchestrators and internal developer portals (IDPs). At this stage, infrastructure self-service becomes a full platform engineering discipline. Developers interact with cloud resources through reusable components presented in application-centric language rather than infrastructure-centric terms. Whether you are deploying Node.js or Java microservices, the platform ensures standardised underlying infrastructure without requiring developers to understand cloud details.

IDPs also help unify traditional and cloud-native workloads. Even for workloads like VDI, a service catalogue and an abstraction layer help users request resources without knowing what happens underneath.

How is Prudential applying AI to manage its cloud infrastructure?

Now we're entering the fifth stage, shaped by the emerging AI era. The goal is to reduce the cognitive load for developers and increase productivity by applying AI to infrastructure orchestration.

Tasks like summarising documentation, generating configuration, interpreting telemetry, or proposing changes can be handled by LLM-powered assistants. Instead of navigating catalogues manually, developers can interact through natural language. They describe their use case, and the system recommends suitable components. For example, they can say: "I need to host a static PDF for a specific business unit," and the system will suggest the right storage or CDN pattern.

Lifecycle management will also evolve. Developers express desired outcomes, and the system generates infrastructure-as-code templates or change requests. Humans still approve decisions, but AI accelerates analysis and reduces manual work. As patterns mature, agents will collaborate to preselect components, leaving developers to approve final recommendations.

How will Prudential's cloud infrastructure evolve to meet new challenges?

Alongside this evolution, hybrid clouds are becoming increasingly complex. Traffic flows across cloud providers, SaaS platforms, on-prem systems, and AI studios like Vertex AI or Azure AI Foundry. With the rise of SaaS, private connectivity becomes critical, especially in regulated industries where public internet connectivity is insufficient.

This progression leads toward what we call omni-cloud: a future where management and control planes are fully abstracted. Developers interact with a unified management plane, while multiple clouds and AI back ends act as interchangeable control planes. This resembles the direction of open telemetry: standards-driven interoperability with vendor-specific implementations underneath.

In omni-cloud, AI and LLM functions interact with the management layer rather than individual cloud implementations, enabling consistent security, policy enforcement, and developer experience across all platforms.

Ultimately, enterprise cloud has become highly complex, and it's unrealistic for developers to understand every layer of the stack. Abstraction, automation, AI-assisted orchestration, and standardised patterns are essential for productivity.

Even with this modernisation, human decision-making remains central. AI can summarise, analyse, and propose infrastructure changes, but the final decision, particularly in critical environments like financial services, must remain with people. ■