

# Integration Lead Interview Response Guide

Sicong (Kevin) Jiang

January 2, 2026

## **Question 1: Tell me about a time you led an integration effort across teams. What challenges did you face, and how did you resolve them?**

Sure — at Spark NZ, I was leading the backend work for a new self-service feature in our mobile app, which sees over half a million users daily. The feature needed to integrate tightly with real-time backend systems hosted on Azure.

The tricky part was getting multiple teams — backend, mobile, DevOps — to stay aligned. We ran into issues with mismatched payloads and different sprint cycles that slowed us down and caused test failures.

So, I kicked off daily stand-ups with all the key folks and got everyone aligned on API contracts using versioning and shared mocks. I also helped build out a staging environment that mirrored prod closely, and we set up automated integration tests.

In the end, we launched on time, cut integration bugs by about 80%, and saw a 40% drop in support calls. It really showed the power of tight collaboration and clean interfaces.

## **Question 2: How have you incorporated Azure services or DevOps practices into a recent project?**

Yeah, we used Azure DevOps quite a bit at Spark. One of my main focuses was improving our CI/CD pipeline for a key backend service. The process was a bit slow and had too many manual steps, so I rebuilt the pipeline using Azure Pipelines with automated tests, approval gates, and multi-stage environments.

We were also managing infrastructure using Terraform — stuff like Azure Function Apps, App Services, and Blob Storage. I tied it all together with role-based access controls using Azure AD to keep things secure and consistent.

This overhaul shaved about 30% off our deployment times, gave us more confidence in releases, and made it way easier for the team to deploy small, safe updates. It also became the standard setup for other teams later on.

## **Question 3: How do you approach defining and executing an integration strategy for a large organization?**

I usually start by understanding the business goals and where the data needs to flow. At Spark, we had a shared backend platform used by multiple mobile teams, and I helped define the integration approach to keep things decoupled but cohesive.

I'm a fan of domain-driven design and building APIs or event streams that are clear, versioned, and scalable. I also look at where things might break down — like system coupling or unclear contracts — and design around that.

Execution-wise, I break things down into deliverable phases, bake in CI/CD and observability early, and keep stakeholders in the loop. I also hold regular architecture catchups and help mentor others so the strategy sticks.

That approach helped us reduce production issues and made our backend services easier to scale and reuse.

## **Question 4: Can you describe a time when you mentored or led a team to adopt new technologies or practices?**

Yeah, at Invenco, I was part of a team moving from a Java backend to Node.js. A lot of the team was new to JavaScript, so I created some starter docs, did pair programming sessions, and ran weekly knowledge shares. That helped everyone get up to speed quickly, and our new services were faster and easier to maintain.

At Spark, I also mentored junior developers on cloud-native development — especially with Azure DevOps and Terraform. I did regular PR reviews, ran workshops, and generally tried to make DevOps feel less scary for newer folks.

What I loved about that experience was seeing others become confident and independent — it made our team stronger and let me focus more on the bigger-picture architecture stuff.