Ryan Monnier

CSD 380

13-Sep-2025

Chapter 13 Case Study - Architecture at Amazon

Amazon’s early system, known as Obidos, was built as a monolithic application where every feature, from product recommendations to customer reviews, was crammed into a single structure. At first this approach worked, but as the company expanded, the application became tangled and resistant to change. The lack of flexibility made it almost impossible to add new features or scale individual parts of the platform without affecting everything else.

Realizing this was unsustainable, Amazon leaders began rethinking their architecture. Instead of continuing to stack more logic on the old foundation, they embraced a service-oriented design. Each function of the business was reimagined as an independent service with its own boundaries, allowing development teams to work in parallel without stepping on each other’s toes. Just as important, Amazon restricted direct access to databases, forcing all interactions to flow through services. This made it easier to improve reliability and performance behind the scenes without disrupting clients.

The shift didn’t just change the technology, it reshaped the company’s development culture. Teams gained end-to-end ownership of their services, from design through deployment, which sped up innovation and kept them closely tied to customer needs. The payoff was huge: by 2011 Amazon was already pushing out around 15,000 deployments every day, and within just a few years that number jumped into the hundreds of thousands.

The Amazon case shows that moving away from monolithic systems and adopting strict service boundaries can unlock flexibility, scalability, and speed. Just as importantly, it highlights how architecture choices influence organizational structure and team productivity, not just technical performance.