Jessica Long-Heinicke

6.23.25

CSD 380

Module 6

Summary of Chapter 13 Case Study

The case study contrasts monolithic and microservice architectures using Amazon’s evolution as a key example. Amazon initially operated a monolithic application called Obidos. This approach offered simplicity, easy database joins, a single schema, and efficient resource use at small scales. However, as the system grew, it suffered from increased component coupling, poor and inflexible scaling (only vertical, all or nothing), difficult performance turning, and cumbersome “all or nothing” schema management. Obidos became too tangled and complex, hindering independent scaling and evolution.

In response, Amazon transitioned to a service-oriented architecture (SOA), essentially microservices, between 2001 and 2005. This new approach involved breaking the monolith into independent, modular services with isolated persistence, forming a graph rather than rigid tiers. Key benefits of this microservices model include simplicity of individual units, independent scaling and performance tuning, independent testing and deployment, and the ability to optimize techniques like caching per service. However, it also introduces challenges, such as managing many cooperating units and small repositories, requiring sophisticated tooling and dependency management, and dealing with network latencies. Amazon’s shift was a significant innovation at the time, moving from a two tier monolith to a fully distributed, decentralized services platform.

Amazon’s transformation yielded three crucial lessons. First, rigorously applying strict service orientation is an excellent technique for achieving isolation between components, granting unprecedented levels of ownership and control. Second, prohibiting direct database access by clients is essential. This isolation allows service teams to make scaling and reliability improvements to the underlying data store without requiring changes or coordination with clients. Third, adopting a service oriented architecture greatly enhances both development and operational processes. It enables the creation of small, focused teams that take full responsibility for service’s entire lifecycle, from design to build operation. This structure fosters rapid innovation with strong customer focus. The impact of these lessons is dramatically evidenced by Amazon’s deployment velocity, which skyrocketed from approximately 15,000 deployments per day in 2011 to nearly 136,000 deployments per day by 2015.