Payment processing self critique

Tam Vo

My group proposed a 3-tiered architecture. I was thinking that this would be a good starting point for a distributed payment processing system because I was thinking about a few users. I was not thinking this would be more than 100 requests per second. Alongside this, I felt that the architecture would scale better if I addressed the missing core components, such as a load balancer into a compute cluster, and a distributed database for storing payment information.

I felt that this solution was still an open architecture as it allows for addition for new components such as a load balancer and distributed cluster compute without an architectural overhaul.

With my additional reading, I was thinking in the lines of, how can I make the process not fail or drop in between each stage. I think that task management is key to architecting a good payment processor. The thing is that payments do not necessarily have to hit the account immediately. So I am wondering, how I can make this eventually consistent without being bottlenecked by an ACID database.