Explain that implementing a message queue is like organizing work on a restaurant: by separating the order emission (waiters) from the order processing (cooks), we can increase our capacity for taking orders, while keeping quality high

The experience of the restaurant is known by most people

In constrained places like a cafe where someone does multiple roles you can experience the bottle neck

ChatGPT: *Imagine you run a restaurant that serves 100 customers a day. You take everyone's order at 6 PM, then try to cook and serve all 100 meals at once. That worked when you had only 20 customers—it was a rush, but manageable. Now, it's chaos. The kitchen gets overwhelmed, the servers are backed up, and everyone's waiting. You can try to speed up cooking a little, but it won't solve the real problem."*
"Now imagine instead you switch to a buffet. As each dish is ready, it goes out on the line, and customers serve themselves when they're ready. The kitchen keeps cooking steadily, and the load is spread out. No traffic jams, no bottlenecks."
"Right now, our system is the restaurant doing a big nightly rush—it overwhelms the notifications system. Moving to a message queue is like setting up the buffet. We smooth out the workload, letting the system handle each notification when it's ready, instead of flooding it all at once.

Imagine you need to peel a lot of potatoes. You could put them all in one bag and haul them over to where you need to work, but eventually your bag isn't big enough. Better to have a conveyer belt bring you the potatoes one by one

job is getting too big is like the potato bag getting to large