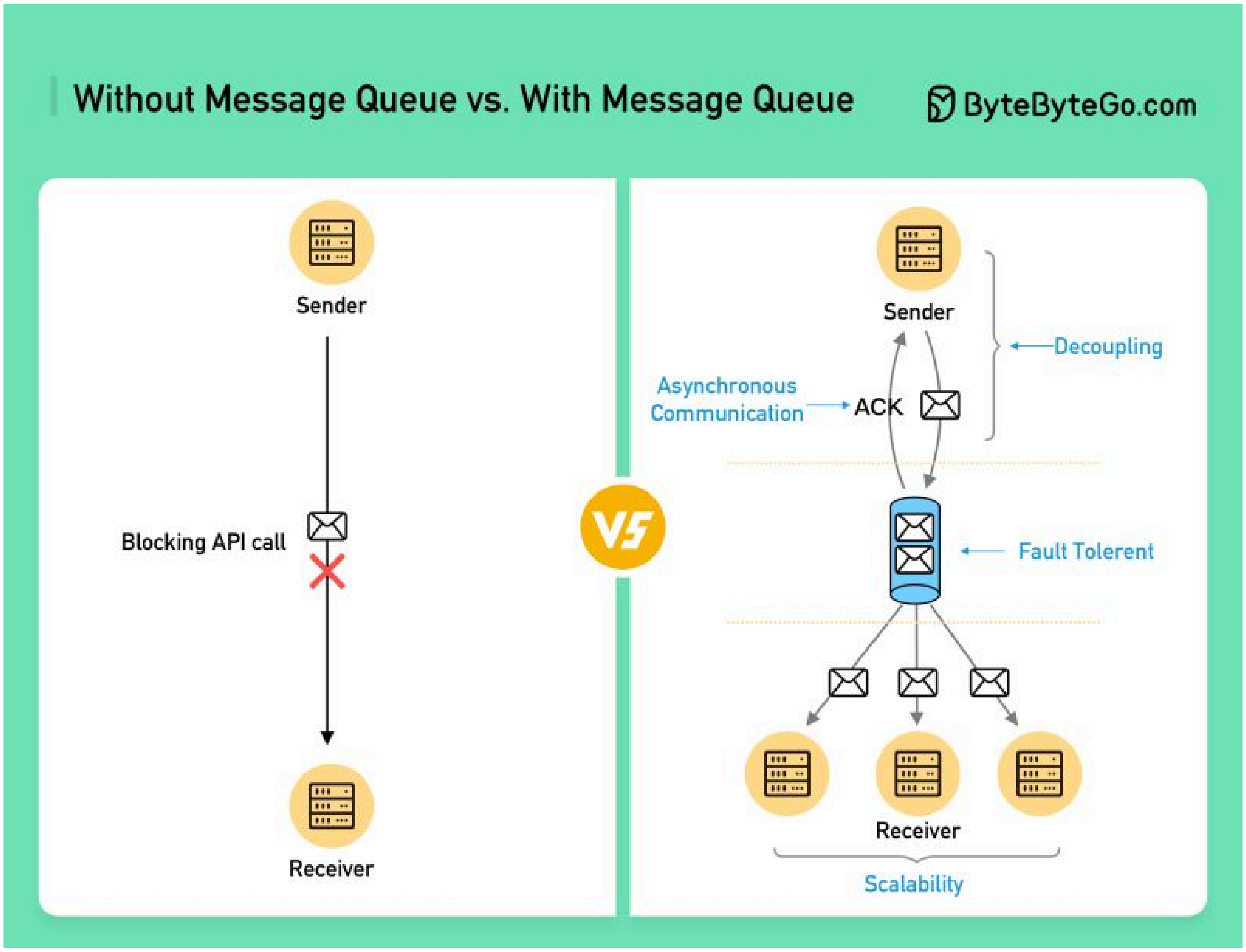
Why do we need message brokers 👇?

Message brokers play a crucial role when building distributed systems or microservices to improve their performance, scalability, and maintainability.



- Decoupling: Message brokers promote independent development, deployment, and scaling by creating a separation between software components. The result is easier maintenance and troubleshooting.
- Asynchronous communication: A message broker allows components to communicate without waiting for responses, making the system more efficient and enabling effective load balancing.
- Message brokers ensure that messages are not lost during component failures by providing buffering and message persistence.
- Scalability: Message brokers can manage a high volume of messages, allowing your system to scale horizontally by adding more instances of the message broker as needed.

To summarize, a message broker can improve efficiency, scalability, and reliability in your architecture. Considering the use of a message broker can greatly benefit the long-term success of your application. Always think about the bigger picture, and how your design choices will affect the overall project.

Over to you: there is a term called pub/sub. Are you familiar with their meanings?