

Transactions in a message broker

event-oriented architecture

message queue

robustness

microservices

Author: Héctor Valls
<https://www.hvalls.dev>

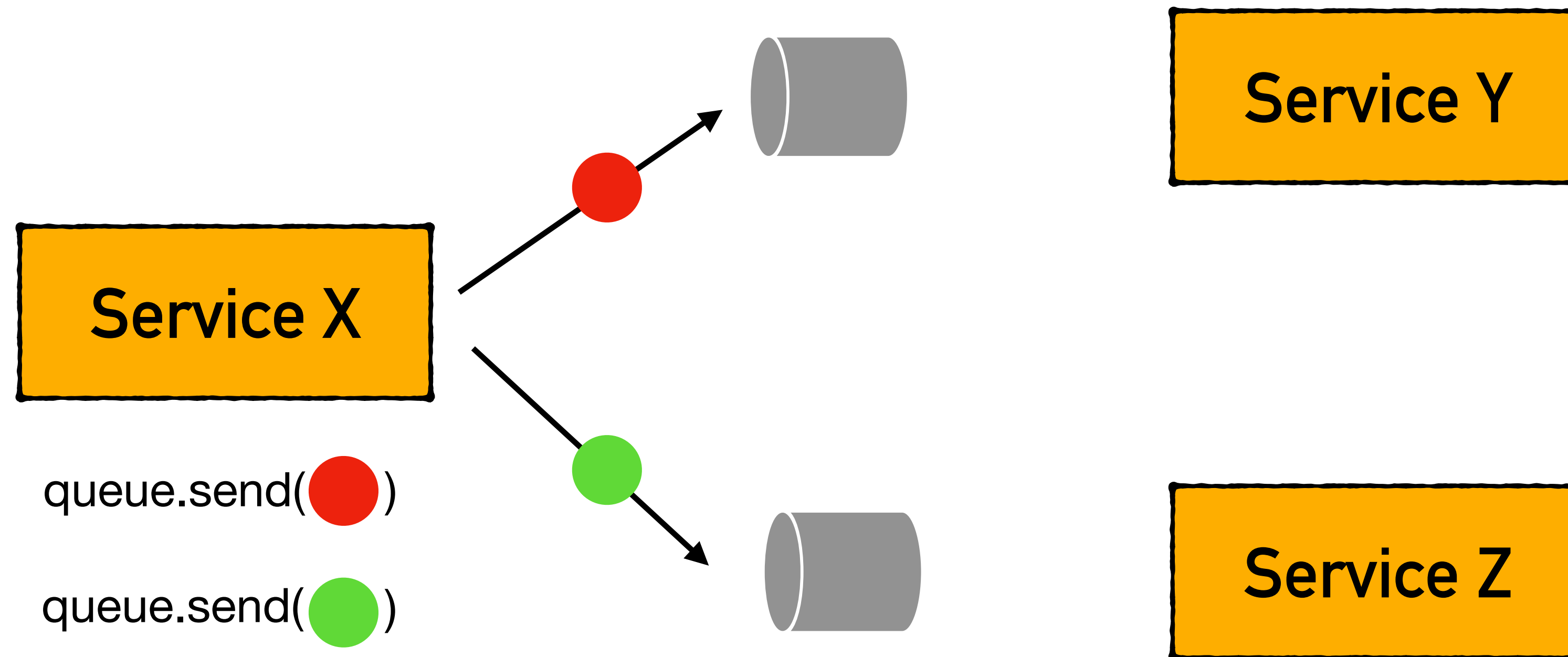
We have three services

Service X

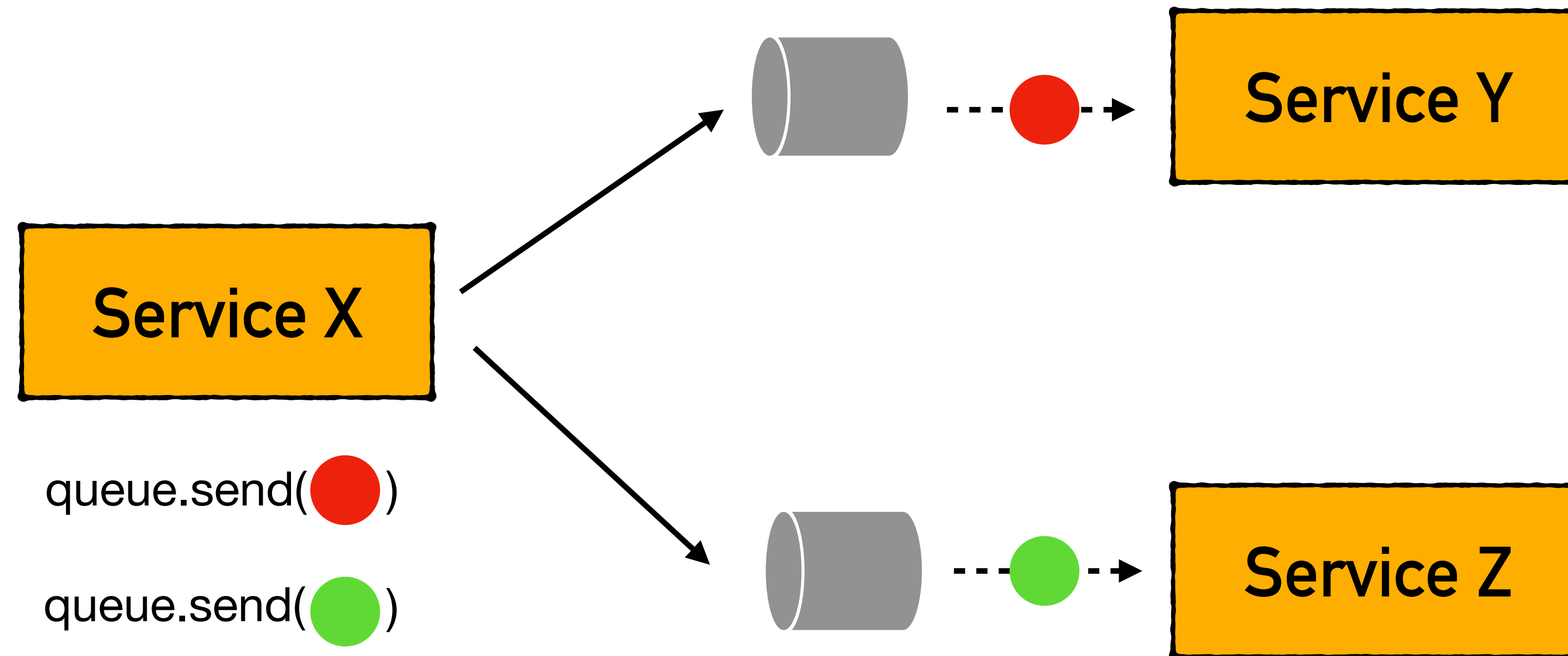
Service Y

Service Z

Service X sends messages to the other services using a message queue

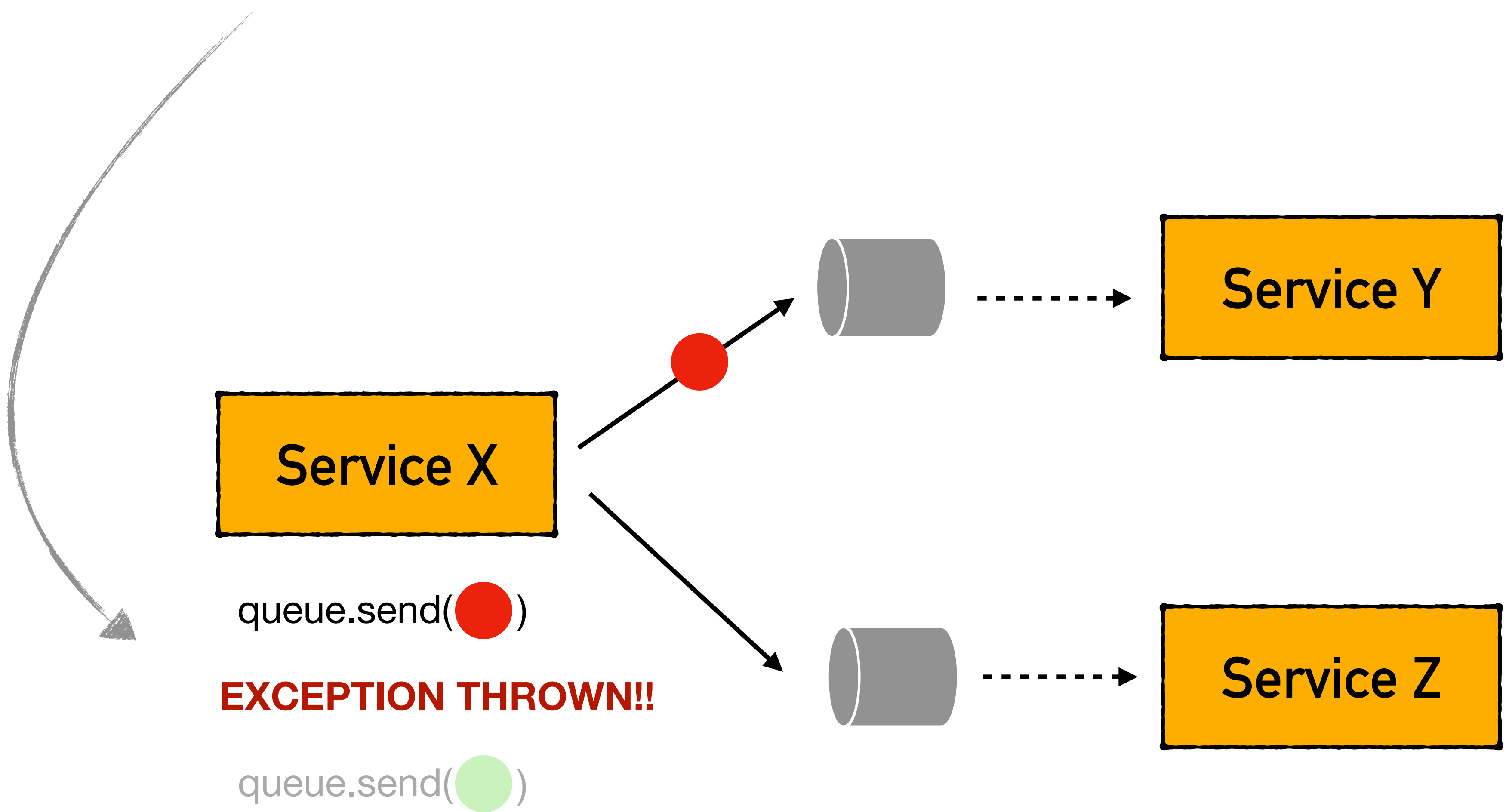


Service Y and Service Z consume their corresponding message from their corresponding message queue

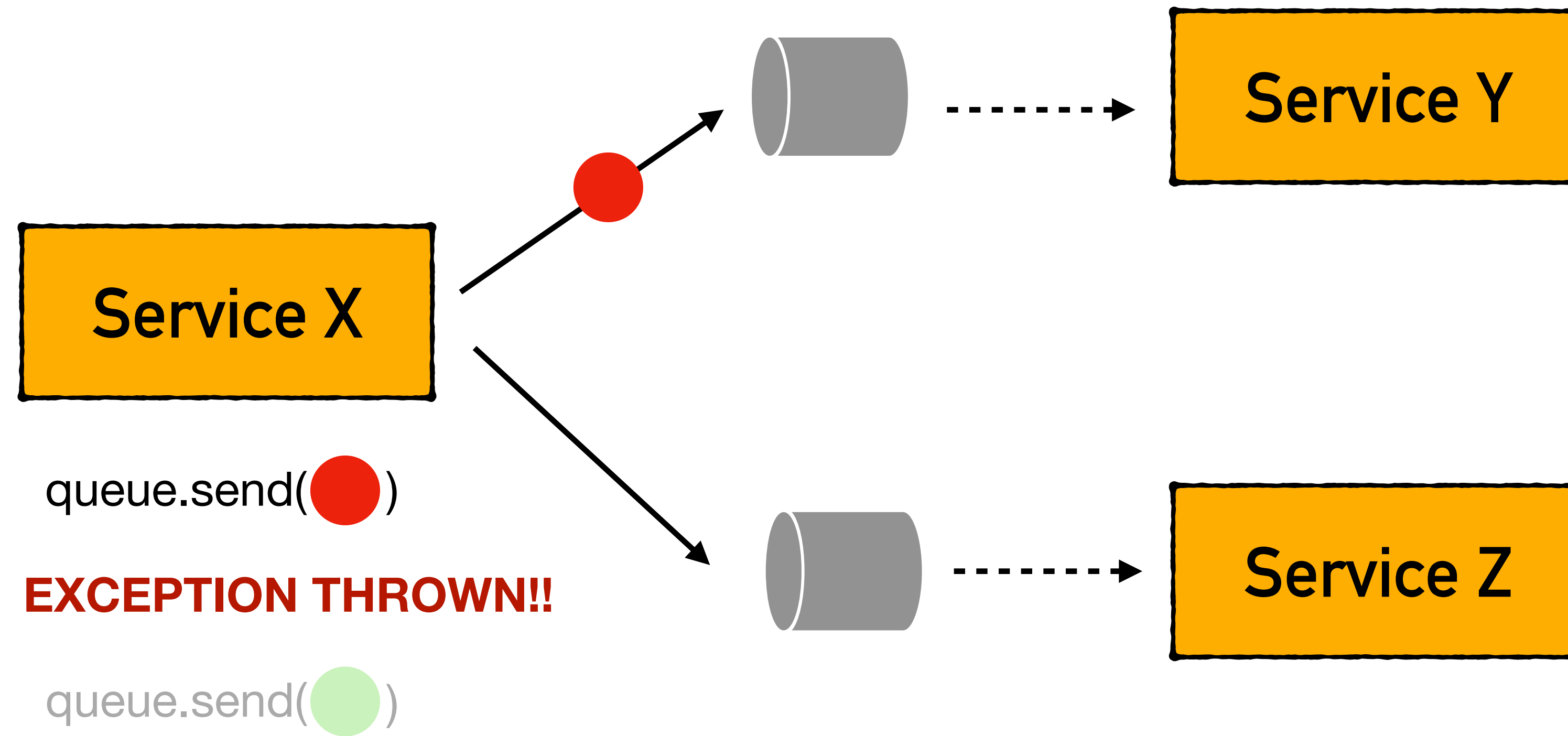


So far, so good...
BUT

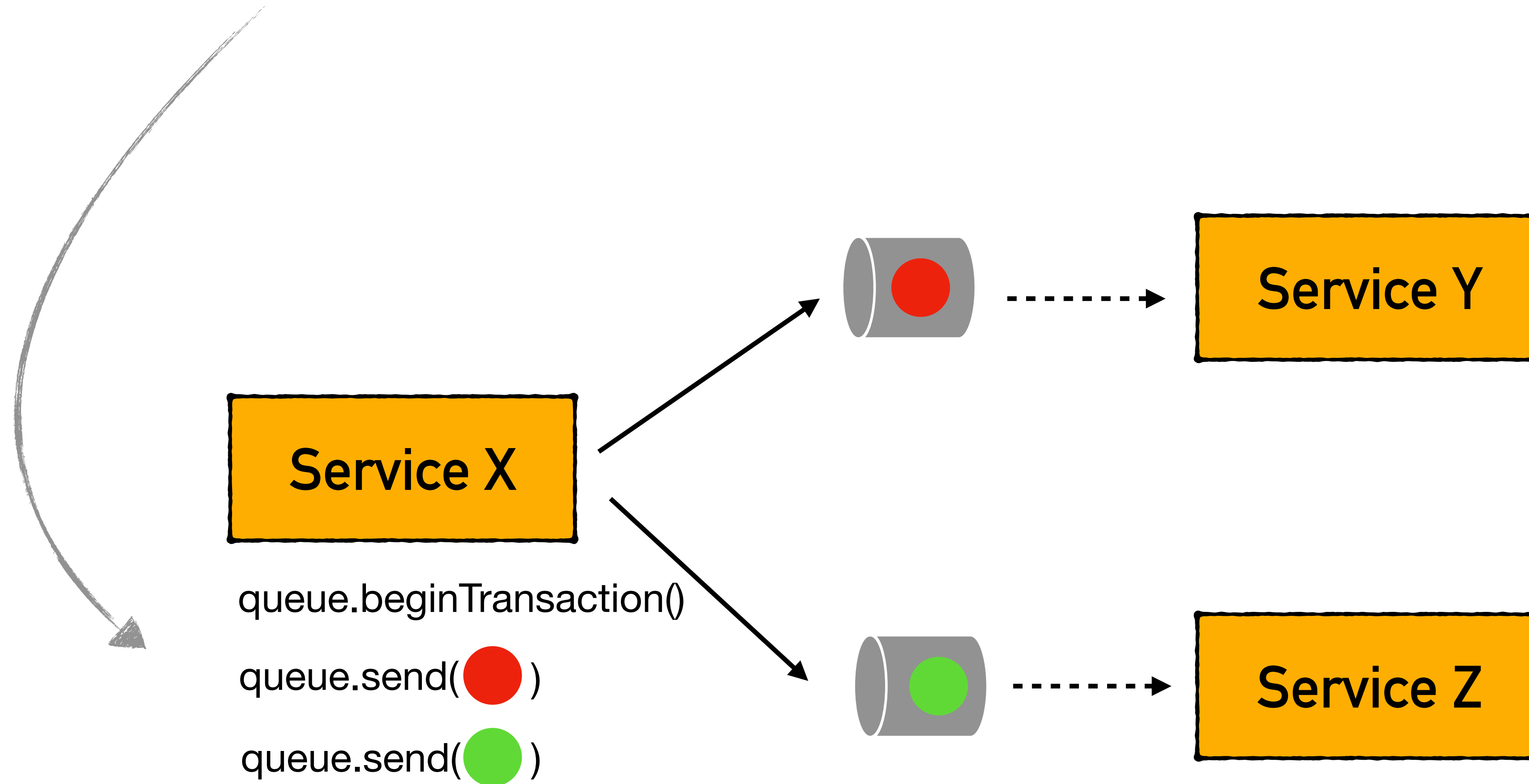
What happens if Service X throws an in-process exception after sending ● and before sending ● ?



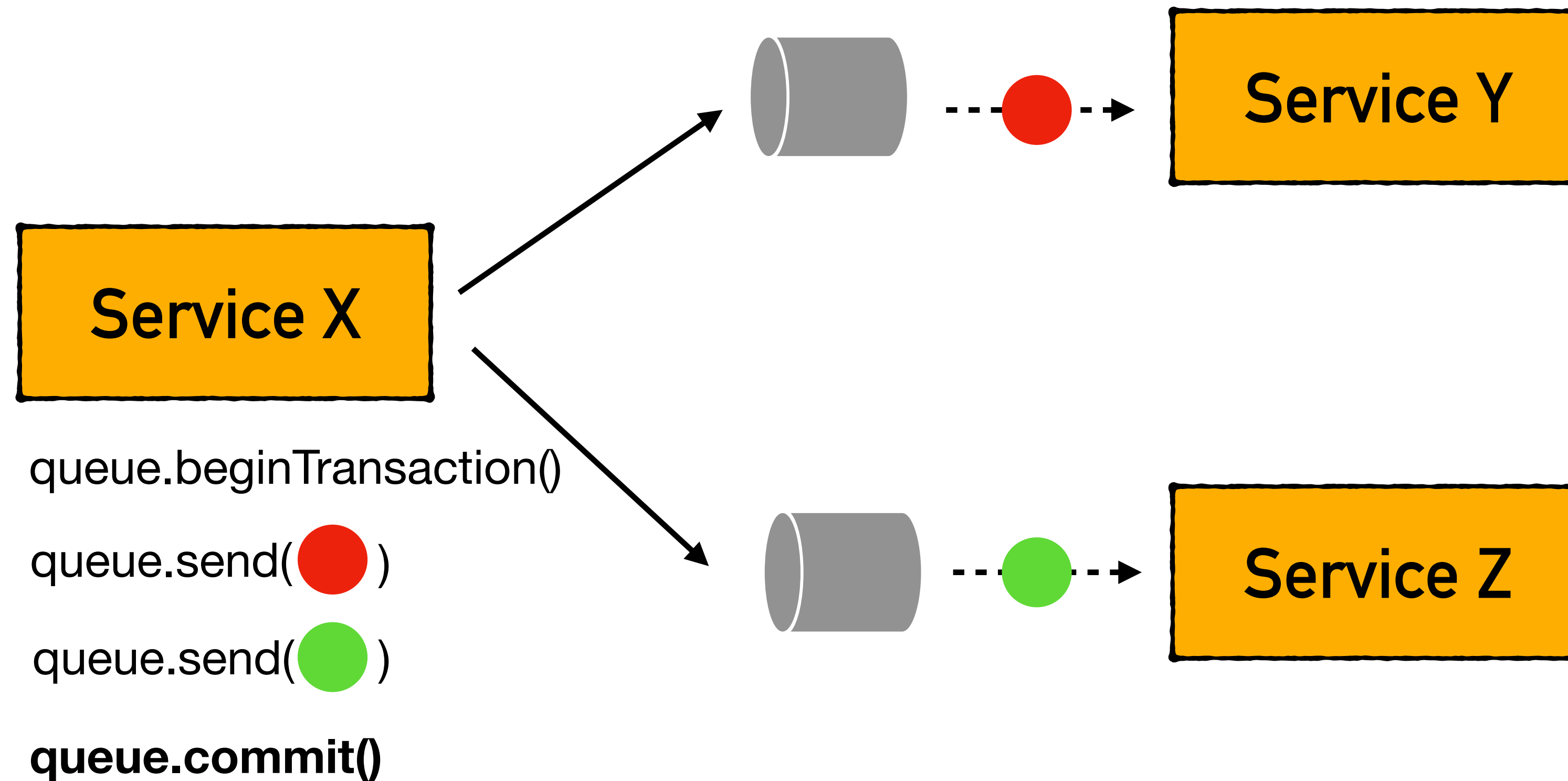
● is sent but ● is not. It may drive to an inconsistent system state.
So, how to solve it?



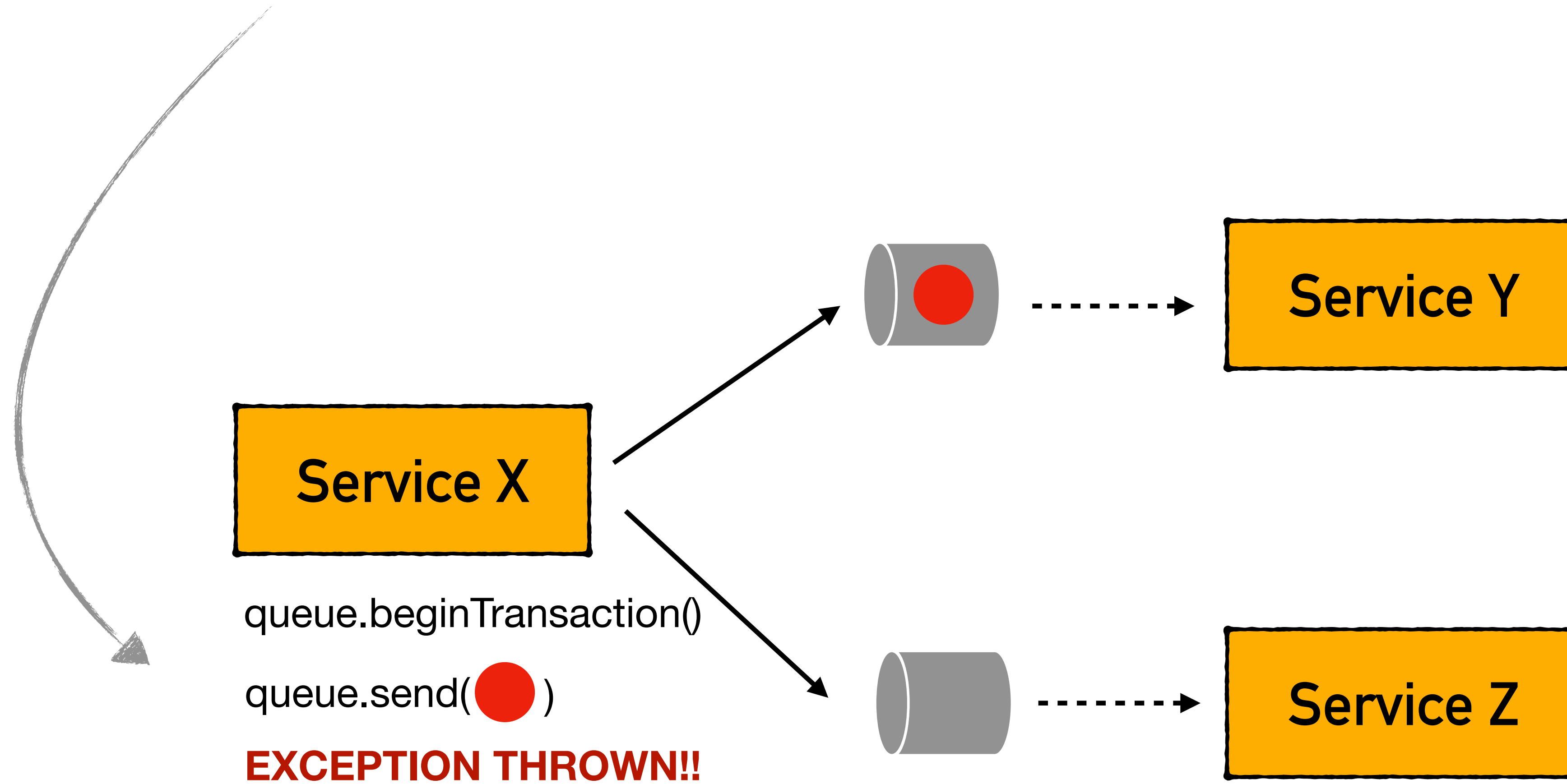
Using a transaction. Not a database transaction but a broker one.



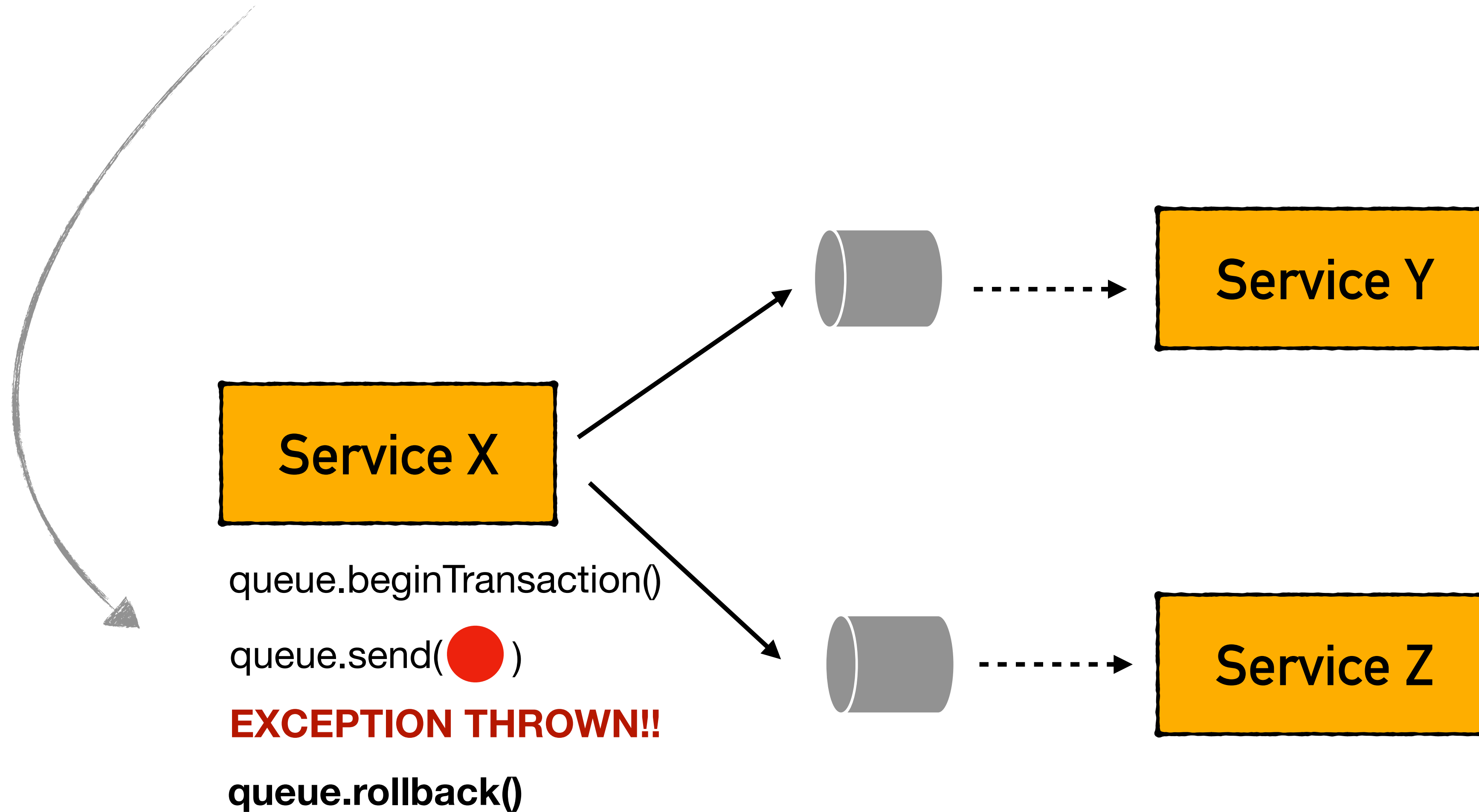
Messages are queued but they are not consumed until transation commits



And if an error occurs...



... you can simply rollback the transaction and messages pending to be consumed will be dismissed



Most of the message brokers provides a transaction system.

Learn yours and make your architecture more robust 🦵

**Find more content like this on
<https://www.hvalls.dev>**