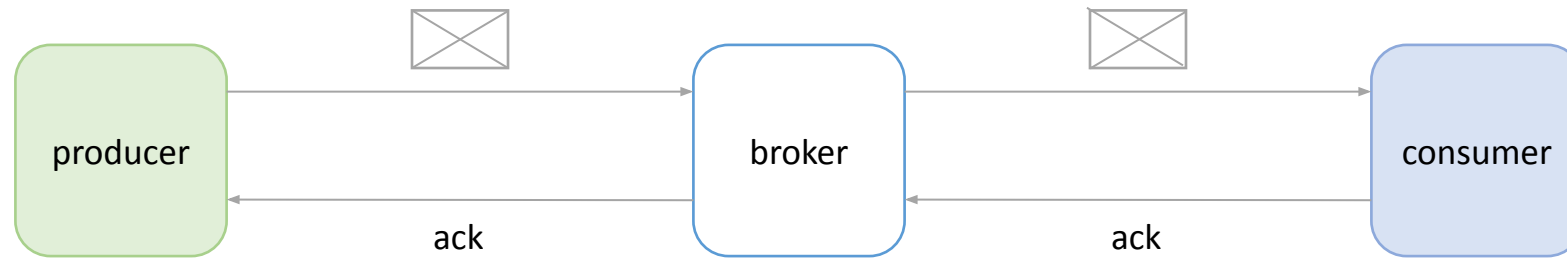
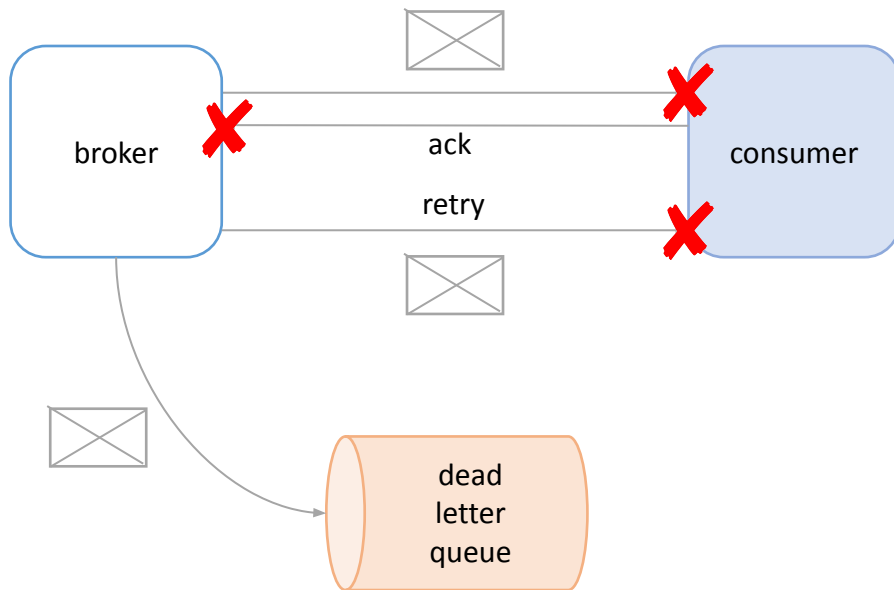


Message delivery guarantees

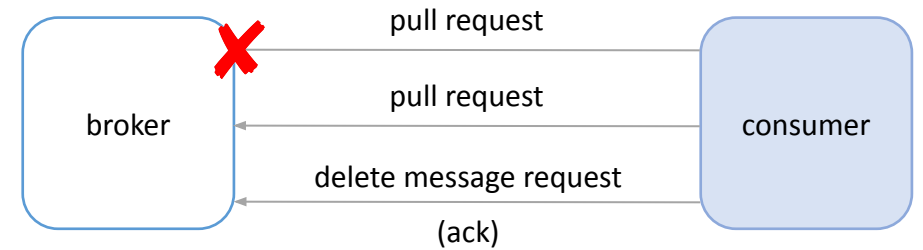


Message delivery guarantees

push



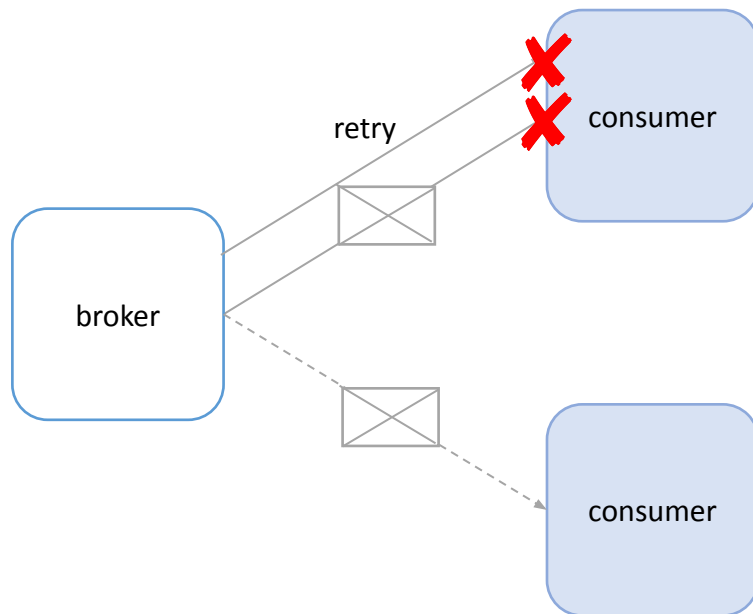
pull



Message delivery guarantees

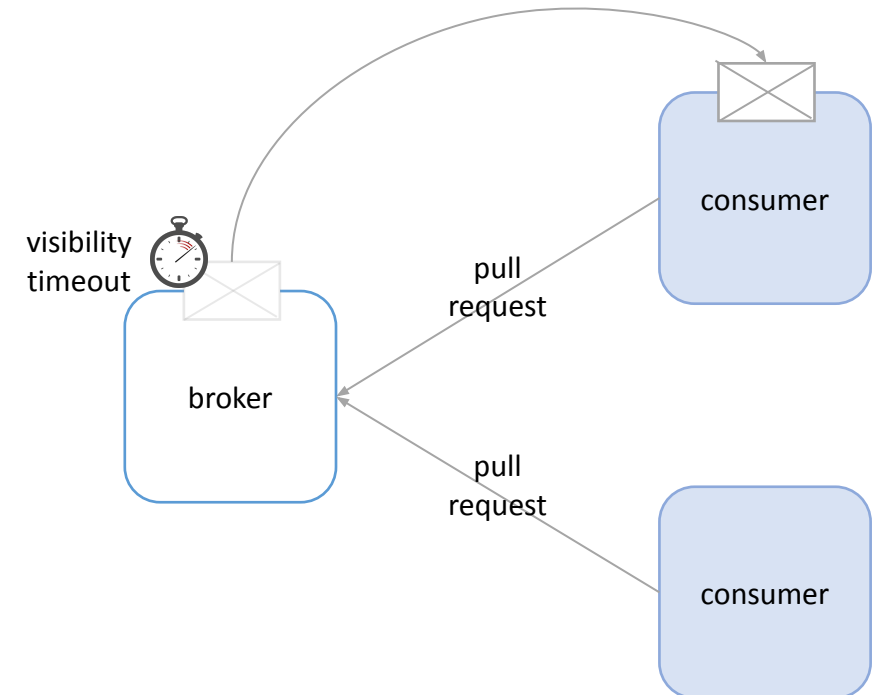
push

(with competing consumers)



pull

(with competing consumers)



Message delivery guarantees

at-most-once

message will never be delivered more than once
messages might be lost

exactly-once

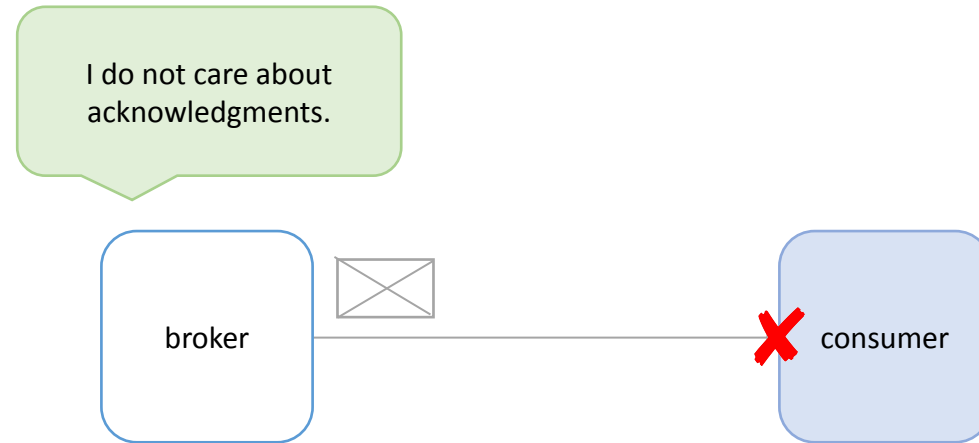
messages are processed exactly one time

at-least-once

retry the message until it is delivered
messages might be processed multiple times

Message delivery guarantees

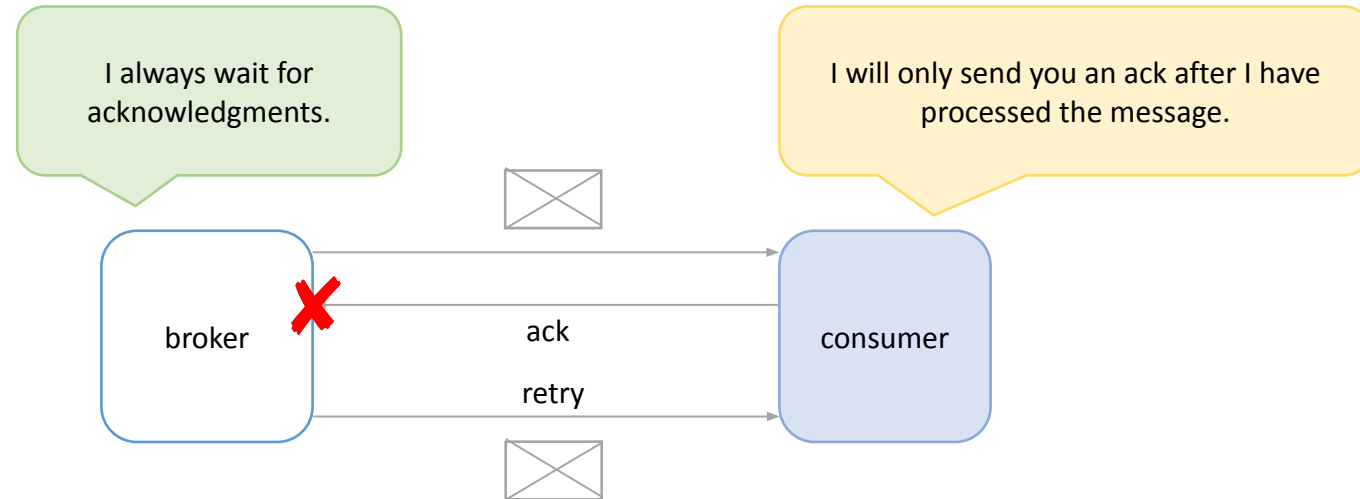
at-most-once



trades off higher throughput for reduced safety

Message delivery guarantees

at-least-once



Message delivery guarantees

exactly-once

- it is achievable (e.g. Kafka Streams, SQS FIFO queues)
- we can rely on either idempotency or distributed transaction to achieve exactly-once semantics
- if we rely on idempotency, it must be implemented on all levels (broker, consumer)
- due to many nuances, it is hard to build exactly-once systems in practice
- most of the messaging systems you will work with will likely be configured to provide at-least-once delivery guarantee