Tema 2: MOM

Sistemas Distribuidos

2022-2023

MOM: *Message Oriented Middleware*

Concepto

Desacoplar clientes y servidores, productores y consumidores.

- Las RPCs son como llamadas telefónicas (el otro extremo debe estar activo)
- La mensajería es como el SMS (el mensaje se almacena hasta que puede ser entregado)

El *Broker*

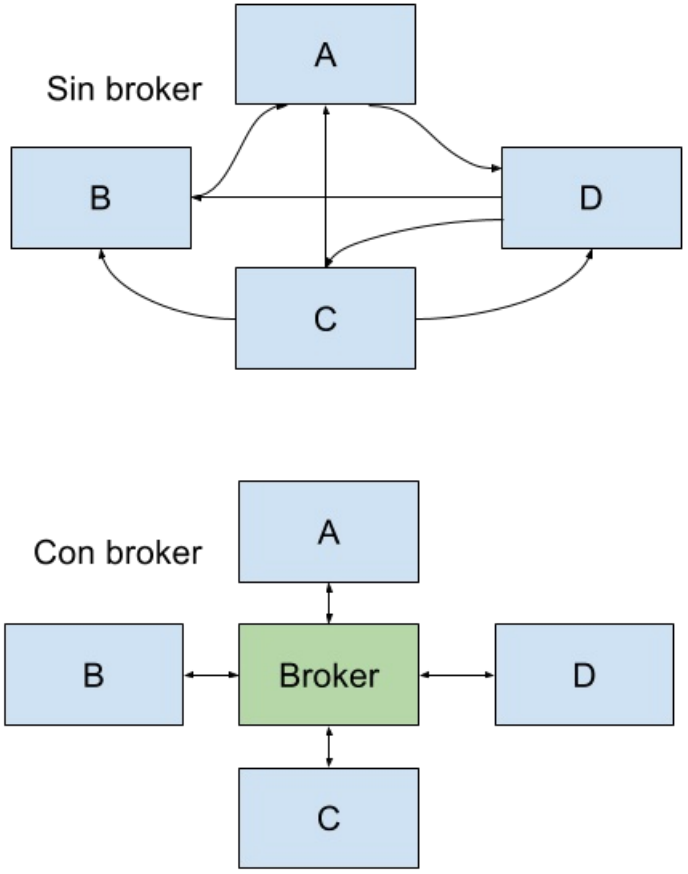
Es un intermediario entre las partes del sistema que queremos comunicar.

Se ocupa de:

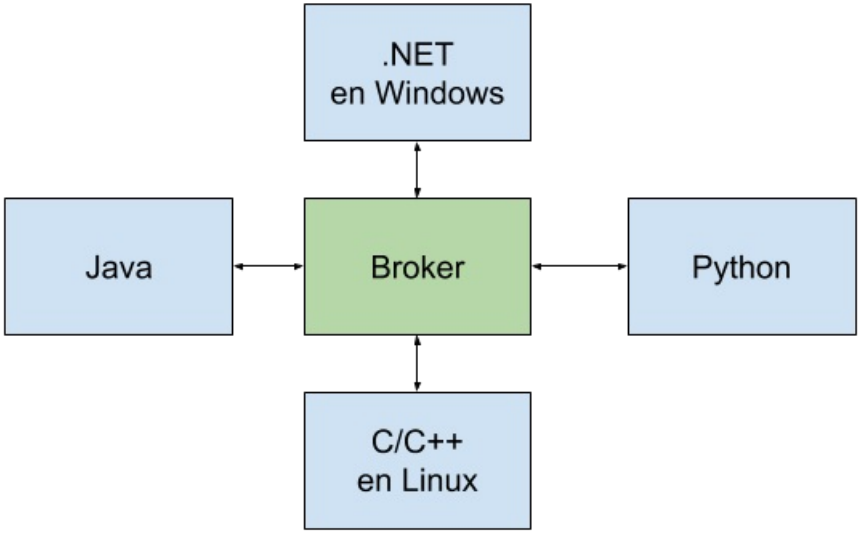
- Transformar la información (mensajes). Agrupar, partir, etc.
- Enrutarla a su(s) destino(s)
- Permitir diferentes patrones de reparto (uno-a-uno, *broadcast*, suscripciones)
- Asegurar la llegada (almacenamiento persistente)

> Simplifica el clásico problema de *muchos se comunican con muchos*

El *Broker* simplifica las comunicaciones



El *Broker* puede "pegar" diferentes entornos y lenguajes



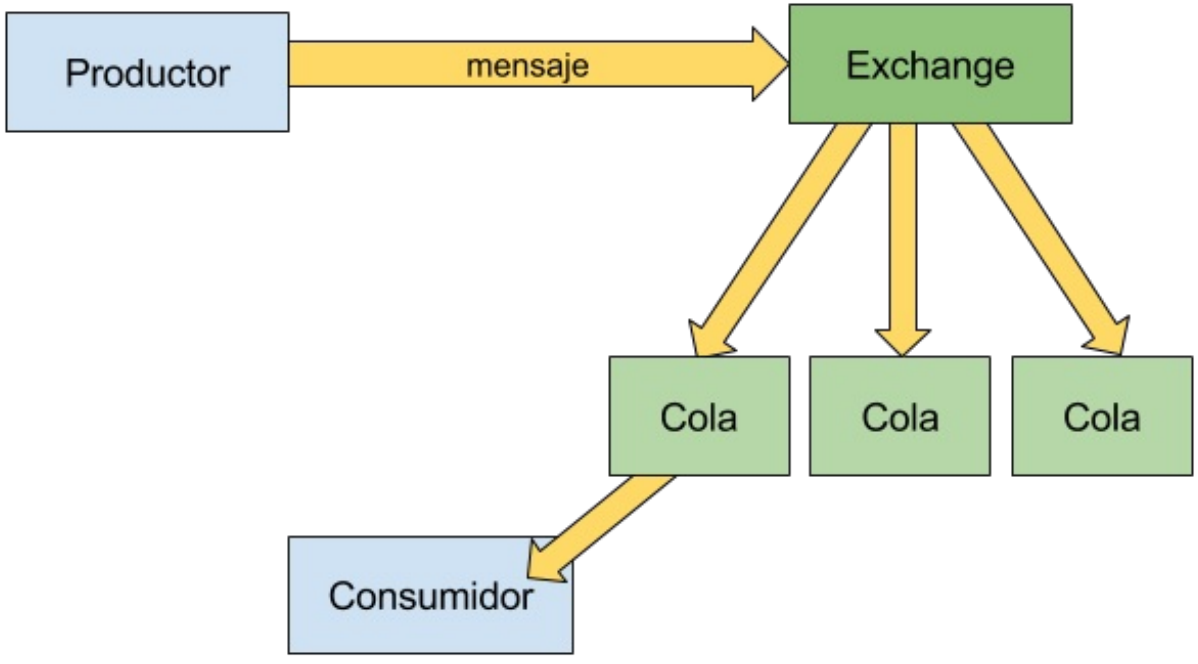
Estandarización

- Durante años los MOM se usaban básicamente en entornos financieros, para "pegar componentes"
- Las soluciones eran cerradas y propietarias (1993: IBM MQseries, 1997: Microsoft MSMQ)
- Java aporta su solución en 2001: JMS
- Finalmente en 2004 se crea el estándar AMQP
- En 2007 aparece la primera implementación abierta: RabbitMQ
- Hay más implementaciones (Apache ActiveMQ, Apache Apollo, etc.)

Elementos de la mensajería

- Procesos (se comunican por *canales*)
  - Productor
  - Consumidor
- Mensajes
  - Cabecera
  - Contenido (paquete de datos más o menos grande)
- Elementos intermedios (*broker*)
  - *Exchanges* (centralitas): Recibe mensajes del productor, los envía a una...
  - Colas: Almacenan mensajes y los envían a consumidores
  - *Bindings*: Asocian *exchanges* con colas.

Elementos de la mensajería (2)



El *Exchange* soporta muchos tipos de *binding* (directo, *fan-out*, por tópico, etc.) para permitir diferentes arquitecturas

Problemas

- Modelo de programación *asíncrono* y *orientado a eventos* (más difícil que el síncrono, secuencial)
- Difícil de adaptar a un modelo síncrono (tipo RPC)
- El *broker* es un punto único de fallo, y cuello de botella
- Rendimiento peor que RPCs
- Un suscriptor lento puede afectar a los demás (al causar que el *broker* limite la velocidad del productor)