



Java Message Service

JMS är ett Java message-oriented middleware API som skickar meddelande mellan flera olika klienter med hjälp av message broker. Det finns flertalet att använda, bl.a. ActiveMQ, Kafka och RabbitMQ. Det behöver inte vara enbart Java till Java, utan Python, C++ eller andra språk som stöder messaging protocol kan skicka ut på olika kanaler som kan lyssnas av.





Queue

För att skicka meddelande mellan applikationer måste en kö först skapas. Den kan vara namngiven, eller gå ut på broadcast så alla kan lyssna på meddelandet.

En applikation blir publisher, dvs den som skickar meddelandet. Övriga applikationer bli subscribers som kopplar upp sig på kön och lyssnar på meddelande.

JMS kö fungerar enligt FIFO principen, och kan bara bli skickad en gång.

Point-to-point model



Fungerar som en privat kanal, och enbart applikationer som är kopplade till denna kö, kommer kunna få ta del av meddelande.

Det kan finnas flera Producers som skickar till kön, och flera Consumers kopplade till kön, men när en Consumer har läst meddelandet, tas meddelandet bort, även om det finns fler Consumers kopplade till kön.

Finns inga Consumers, stannar meddelandet i kön tills dess en Consumer har kopplats upp alternativt om meddelandets tid löper ut.

Publish-and-subscribe model



Skickar meddelande med ett speciellt meddelande ämne.

Consumers lyssnar efter meddelande med en speciell topic. Producers och Consumers har ingen vetskap om varandra, de har ingen kö de skickar ut på som de båda är uppkopplade på. Därför kan det vara 0 som lyssnar, eller X som lyssnar.

Consumers måste aktivt lyssna, om den är offline, kommer den missa meddelandet, om den inte konfigureras som en Durable Subscriber då alla meddelande cachas och sen visas när applikationen startas.