

Kafka vs RabbitMQ

Avantajları

- Tek bir protokol kullanılıyor. (Native Kafka Protocol)
- Mesajlar kalıcı (persistency),
- Kolay ölçeklenebilir Cluster yapısı,
- Consumer'ların offline olduğunda mesajın kaybolmaması istenirse,
- Çok fazla sayıda mesajlaşma yapılıyorsa (saniyede 100k ideal ancak gerekli konfigürasyon ile saniyede bir kaç milyon mesajlaşmanın üstesinden gelebilir),
- Stream Processing yapıyorsanız,
- En önemlisi Cursor'ın Queue 'da nerede kalındığını consumer'ın bilmesini ve yönetmesini (consumer-centric) istenirse tercih edilmeli.
- Farklı protokoller kullanmak için (AMQP, MQTT vs.),
- Mesajın ulaşmasını garantilemek istenirse,
- Yüksek performansa ve kolay entegrasyon istenirse (dokümantasyonu oldukça güçlü, bir çok client library'si mevcut),
- Geliştirme ortamınız .Net ise (resmi .NET destekli)
- Kafka'ya göre daha az sayıda mesajlaşma yapılıyorsa (saniyede 20k ideal ancak saniyede bir milyon mesajlaşmayı da kaldırabilecek biçimde geliştirildi),
- Farklı topolojilere uyumlu olmasını istenirse,
- Cursor'ın queue'da nerede kaldığını broker'ın belirlemesini (broker-centric) istiyorsanız

Dezavantajları

- Consumer'lar mesaj almak için broker'ı tetikleyecekse (Event odaklı mimari),
- Mesajın iletilmesini garanti etmez. (acknowledge yok)
- Queue'da filtreleme veya veriyi güncelleme doğrudan yapılamaz. Data flow veya processing frameworklerine ihtiyaç var. (Apache Beam, Spark, Flink vs.)
- Consumer'ların online olduğunu var sayar ve acknowledge gelmediği sürece mesaj durumu beklemede olarak işaretler. (mesaj iletimi)
- Geliştirme ortamınız .NET ise piyasadaki SDK'lardan birini kullanmak zorundasınız. Resmi olarak destek vermiyorlar.
- Yüksek hacimli mesajlaşma için uygun olmayabilir.
- Mesajların kalıcı olması için persistent modunda kullanılmalı. Yoksa RabbitMQ servisi restart olduğunda tüm mesajlar kaybolacaktır.