

Comparatif de Kafka vs. RabbitMQ :

Points communs :

- Middleware orienté message (MOM)
- Même ordre de grandeur de message consommable par seconde (un peu + côté Kafka)
- Open source
- Modèle d'intégration : Publish / Subscribe (+ point-to-point côté RabbitMQ)

Domaines d'applications :

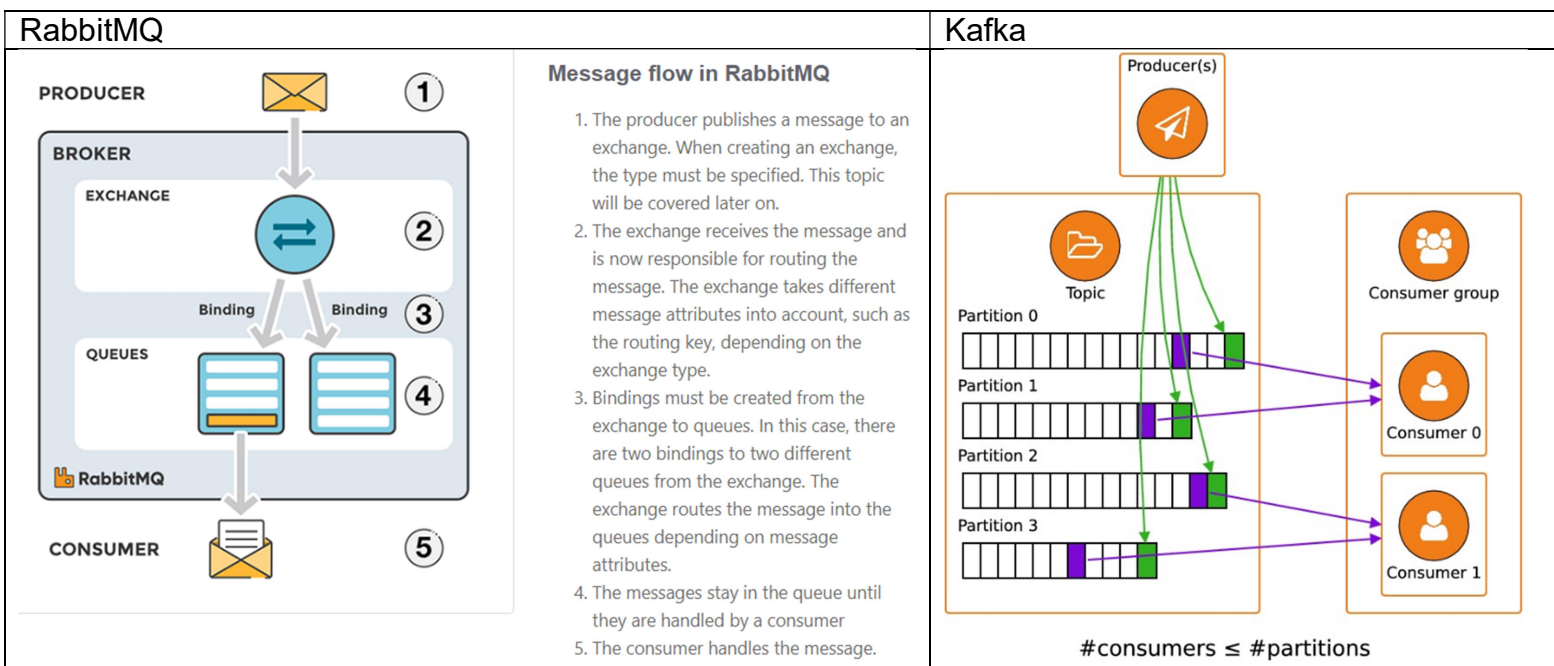
- Collecte d'infos d'IoT
- Transmission de métriques, logs
- Envoi de notifications, mails, messagerie instantanée...
- Transmission d'action utilisateurs
- Actions compte à compte bancaire

Infos générales / Principales différences :

RabbitMQ	Kafka
Depuis 2007	Depuis 2011
Destiné au début : composante primaire dans les messageries SOA ; Maintenant : Flux	Pour le scénario de flux
Structure de données : FIFO ; Optimal quand les messages sont livrés rapidement	Structure de données : Log (le consommateur gère l'offset) Possibilité de relire des données, les messages sont conservés selon un temps paramétré
Inclus les protocoles : MQTT, AMQP, STOMP ; Facilité pour la communication avec d'autres solutions implémentant AMQP ;	Routage simple (clé de routage). Les messages sont sur des « topics » (les consommateurs s'abonnent aux topics voulus)
Mode de délivrance des messages : <i>at least once</i>	Modes de délivrance des messages : <i>at least once</i> et <i>exactly once</i>

Architecture :

	RabbitMQ	Kafka
Stockage	Dans une base Mnésia. Quand saturation sur la base Mnésia, stockage sur disque	Sur disque, dans des fichiers (tailles équiv.), logs. Cluster de servers, dans des topics (Durable).
Routage	Flexible	Basique
Structure de données	File FIFO	Log



Approche :

RabbitMQ	Kafka
Approche push-based	Approche pull-based
Distribuer les messages individuellement et rapidement ;	Consommateur récupère les lots souhaités, à partir d'un offset spécifique ; Mise en commun longue

Messages :

	RabbitMQ	Kafka
Maintien de l'ordre	Dans une même chaîne (TCP multiplexée)	Dans une même partition
Temps de vie	Jusqu'à que le message soit consommé (et retour du consommateur reçu)	Délai paramétré (ou si saturation atteinte)
Priorités	Possibilité de définir un degré de priorité des messages	-

Préférence selon le cas d'usage

RabbitMQ	Kafka
Besoin de routages élaborés ; Utiliser des protocoles STOMP, MQTT ou AMQP... ; Suivi de métriques opérationnelles ;	Besoin de routages simples ; Conserver (sur temps donné) et relire des messages ; Mise à l'échelle ; Capture d'évènement induisant un changement d'état (dans une base de données ou autre) ; Besoins transactionnels ; Traiter les données en parallèle.

Sitographie :

<https://www.upsolver.com/blog/kafka-versus-rabbitmq-architecture-performance-use-case> [en]

<https://blog.ippon.fr/2018/03/27/comparatif-rabbitmq-kafka/> [fr]

<https://www.cloudamqp.com/blog/2015-05-18-part1-rabbitmq-for-beginners-what-is-rabbitmq.html> [en - avec l'image pour l'architecture RabbitMQ]

<https://openclassrooms.com/fr/courses/4451251-gerez-des-flux-de-donnees-temps-reel/4451521-metamorphosez-vos-applications-temps-reel-avec-kafka> [en - avec l'image pour l'architecture Kafka]