

# MESSAGE QUEUE

## POURQUOI ?



- Permet de mettre en oeuvre des patterns plus avancés que le simple RPC (pub/sub, routing, topics, ...)
- Permet de mettre en place des stratégies back-to-back!  
*Quand il n'y a pas d'Humain qui clique:*
  - Que faire quand le service n'est pas dispo ?
  - Que faire quand le service ne répond pas ?
- Facilite l'architecture backend
  - La découvrabilité des services est simplifiée
  - Permet un couplage faible entre les services
  - Simplifie grandement les problèmes de gestion d'identité & droits

### Solutions:

RabbitMQ est sûrement la solution leader open source. C'est une très bonne solution, complète et avec un coût assez faible pour débiter

Apache Kafka est une autre solution populaire, le ticket d'entrée est beaucoup plus coûteux

zeroMQ est une solution de niche assez intéressante et en croissance, y compris chez de grands comptes, mais plus bas niveau en terme d'UX

Tous les cloud providers GCP, AWS, Azure ont leur solution vendor lock

Les principales solutions ESB (IBM, TIBCO, Software AG, Oracle, Red Hat) intègrent une message queue

MIAGE M2 - QUALITÉ DU SI - THOMAS HAESSLÉ & QUENTIN BURG

# LE VRAI VISAGE D'UN SI MICROSERVICES

## UNE ARCHITECTURE FRONT-BACKS ZERO-TRUSTED NETWORK (LA PLUS COURANTE)

