



# SKILLIO

**Plateforme intelligente de formation & gestion  
des compétences**

Module : Applications Web Distribuées

**Présenté par: GenZ**

# SOMMAIRE

- 1. Introduction & présentation du projet**
- 2. Objectifs et vision fonctionnelle**
- 3. Architecture globale du système**
- 4. Description des microservices**
- 5. Communication & flux**
- 6. Infrastructure et déploiement**
- 7. Conclusion**





## 1-INTRODUCTION & PRÉSENTATION DU PROJET

**Skillio** est une plateforme intelligente dédiée à la formation et au développement des compétences.

Elle permet de centraliser les formations, de suivre les compétences des utilisateurs et de faciliter les interactions professionnelles.



## 2- OBJECTIFS ET VISION FONCTIONNELLE

**AUTHENTIFICATION  
ET GESTION DES  
ACCÈS**



**FORMATIONS ET  
CONTENUS**



**CONTENU ET  
COMMUNICATION**



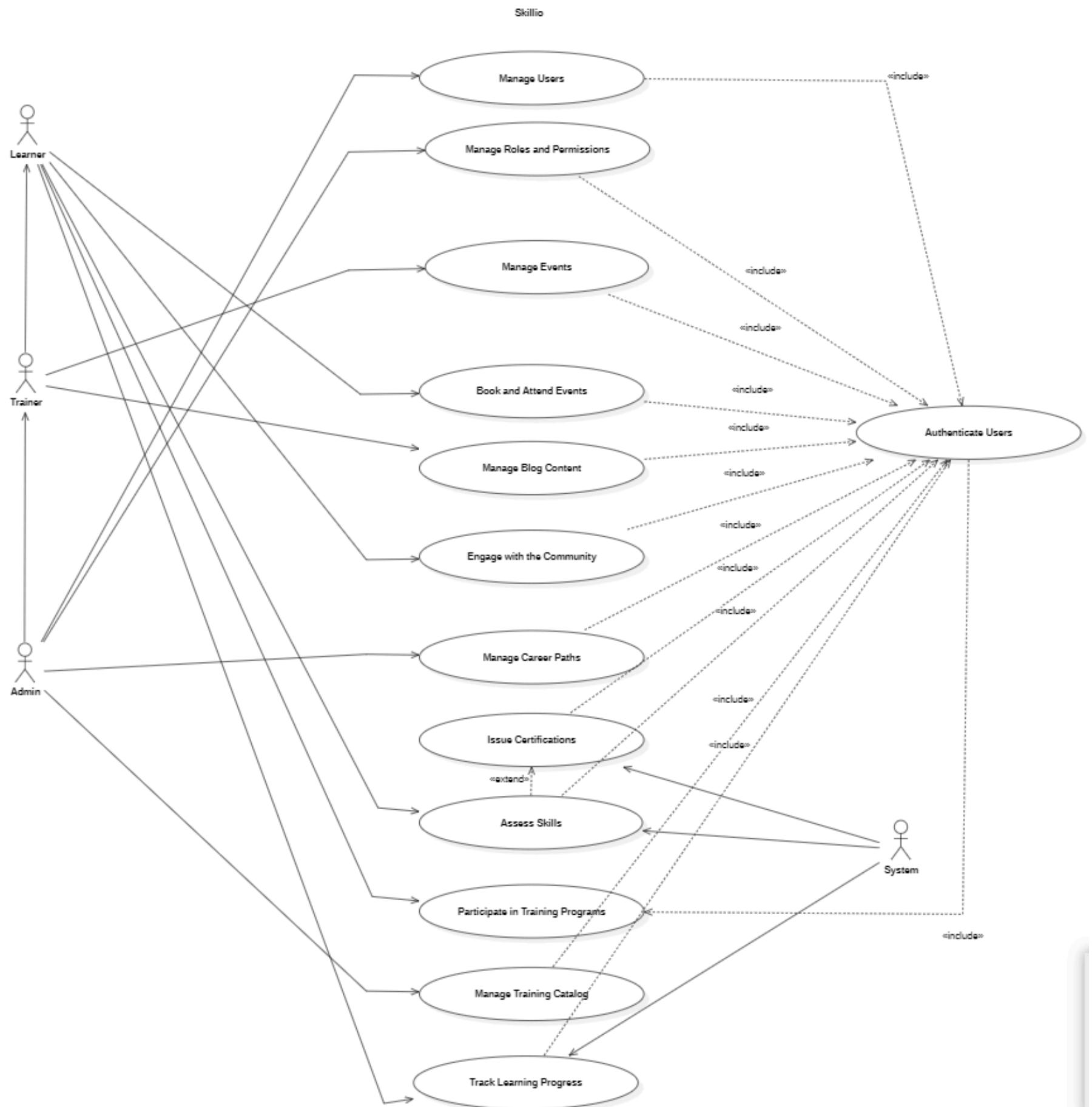
**ÉVÉNEMENTS ET  
ENGAGEMENT**



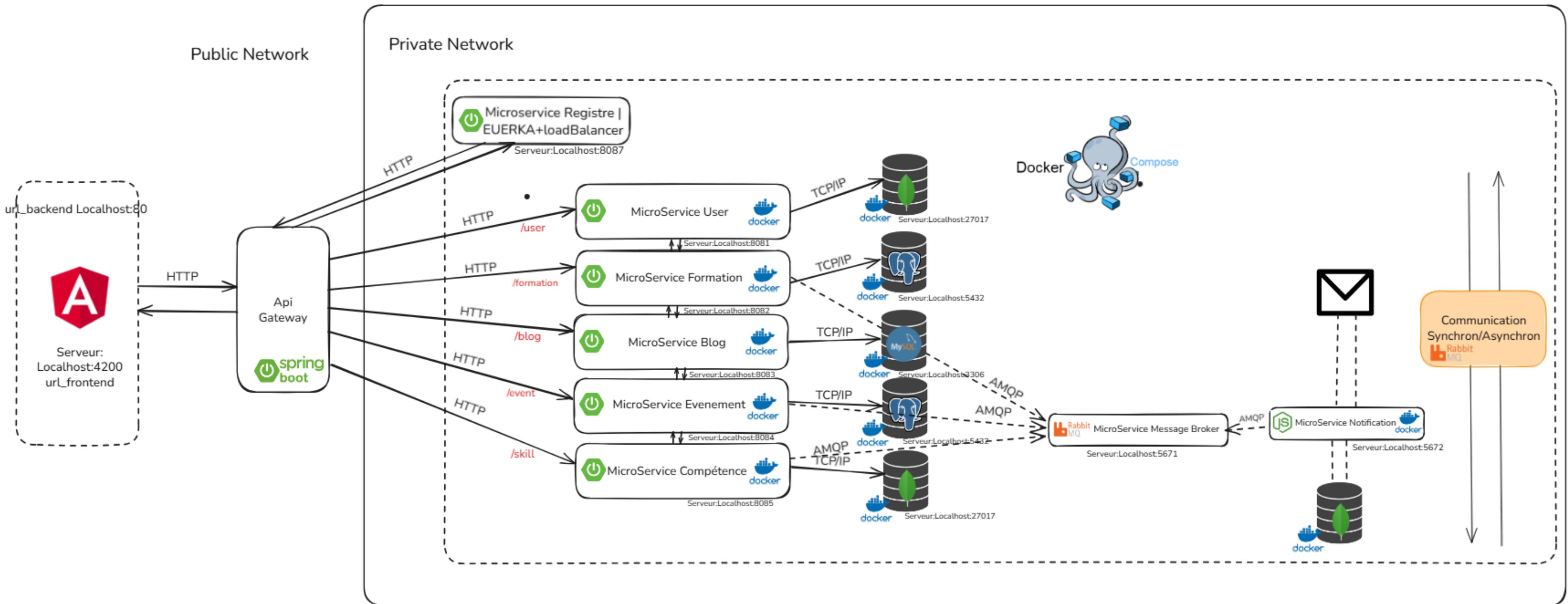
**COMPÉTENCES ET  
CERTIFICATIONS**



# DIAGRAMME D'USE CASE



# 2-ARCHITECTURE GLOBALE DU SYSTÈME



# 4-DESCRIPTION DES MICROSERVICES

Microservice	Responsabilité	Technologie
MicroService User	Gestion utilisateurs, authentification, rôles & permissions	Spring Boot + Docker
MicroService Formation	Gestion catalogue formations, participation, progression	Spring Boot + Docker
MicroService Blog	Gestion contenu éditorial, interactions communauté	Spring Boot + Docker
MicroService Evenement	Gestion événements, réservations	Spring Boot + Docker
MicroService Compétence	Évaluation compétences, certifications, parcours professionnels	Spring Boot + Docker
API Gateway	Routage, sécurité, point d'entrée unique	Spring Boot
Service Registry	Découverte services, Load Balancing	Eureka
Message Broker	Communication asynchrone, notifications	RabbitMQ
Notification Service	Envoi notifications (email, push)	Spring Boot + Docker

# 5-COMMUNICATION & FLUX

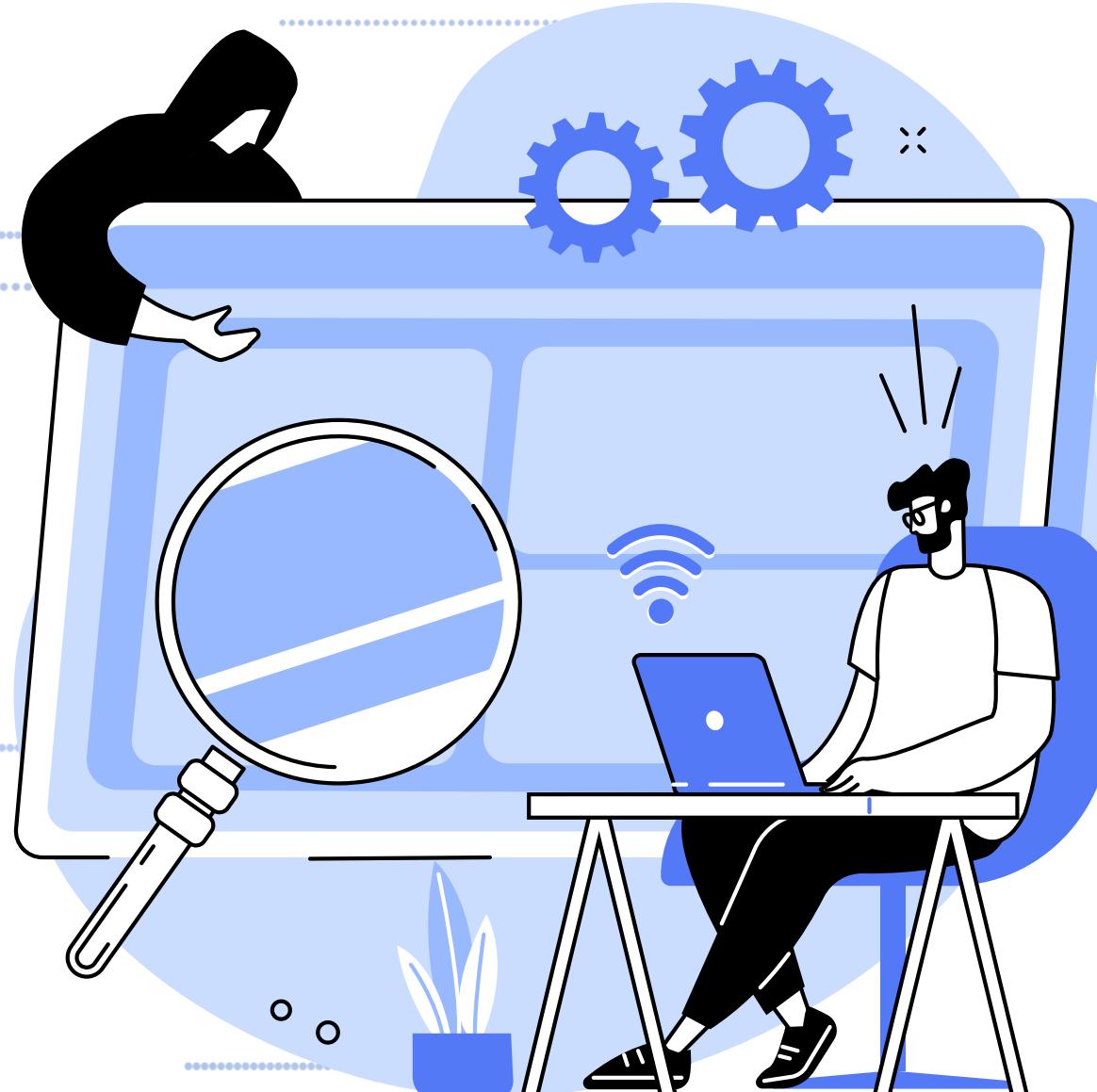
## TYPES DE COMMUNICATION

- **HTTP/REST:** Frontend ↔ API Gateway ↔ Microservices
- **TCP/IP:** Inter-microservices communication
- **AMQP:** Message Broker pour événements asynchrones

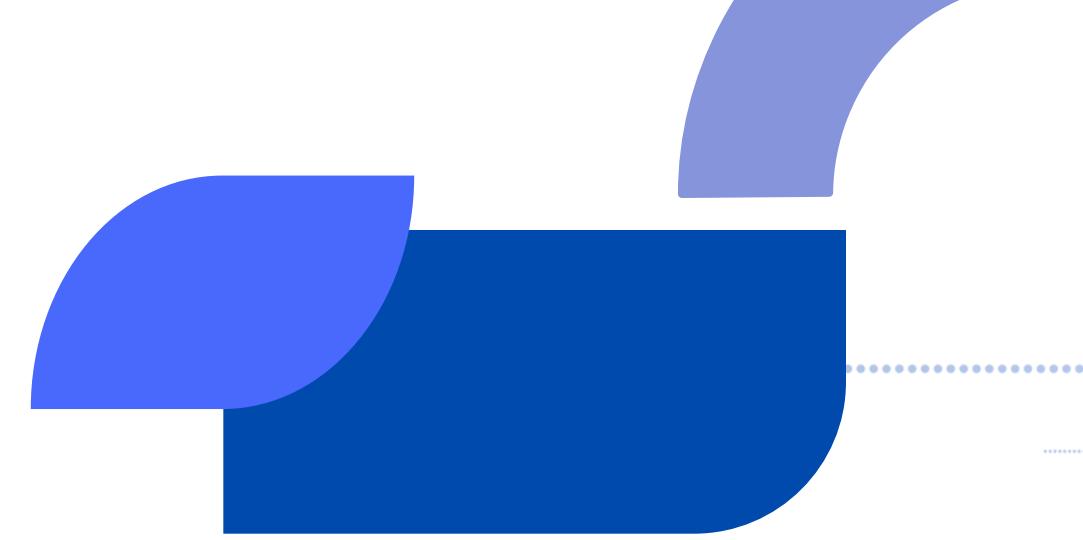
## FLUX PRINCIPAUX

- **Communication Synchrone (HTTP)**
- 1. Frontend → API Gateway → Microservices → Databases**
- User: MongoDB (27017)
  - Formation: PostgreSQL (5432)
  - Blog: MySQL (3306)
  - Evenement: PostgreSQL (5432)
  - Compétence: MongoDB (27017)
- **Service Discovery**
- 2. Microservices ↔ Eureka Registry (enregistrement + load balancing)**
- **Communication Asynchrone (AMQP)**
- 3. Événements métier → Message Broker → MS Notification**
- Nouvelle inscription formation
  - Certification obtenue
  - Rappel événement
- **Communication Inter-Services (TCP/IP)**
- 4. MS Formation ↔ MS Compétence (synchronisation skills)**
- 5. MS User ↔ Tous services (authentification/autorisation)**

# 6-INFRASTRUCTURE ET DÉPLOIEMENT



- **Conteneurisation:** Docker + Docker Compose
- **Bases de données:**
  - MongoDB (ports 27017) - MS User, MS Compétence
  - PostgreSQL (ports 5432) - MS Formation, MS Evenement
  - MySQL (port 3306) - MS Blog
- **Frontend:** Angular (localhost:4200)
- **Backend:** url\_backend (localhost:8088)
- **Message Queue:** RabbitMQ pour communication asynchrone
- **Service Discovery:** Eureka pour load balancing
- **Orchestration:** Docker Compose pour déploiement multi-conteneurs



## 7-CONCLUSION

**Skillio** est une plateforme moderne et scalable  
L'architecture microservices permet une évolution  
continue

Communication efficace via REST & RabbitMQ  
Infrastructure robuste et prête pour le cloud

Présenté par: GenZ

