📩 Email Microservice Architecture Overview

🏛️ Architecture Summary

Our email service is designed using **Spring Boot microservices** with **Eureka for service discovery**. It includes **rate limiting and API usage tracking** to support a pay-per-use model. The entire system is **containerized using Docker** for easy deployment.

Services & Responsibilities

Service	Responsibility
Eureka Server	Service discovery for microservices
API Gateway	JWT authentication, rate limiting, and request routing
User Service	Manages users, authentication, and subscriptions
Email Service	Sends plaintext & HTML emails, stores email logs
Billing Service	Tracks API usage, applies rate limits
Frontend (Dashboard)	Users track emails, view usage, and manage subscriptions

---

🔄 How It Works

1️⃣ **User logs in** → Receives JWT token for authentication 2️⃣ **User sends an email request** → API Gateway enforces rate limits 3️⃣ **Billing Service logs usage** → Tracks the number of emails sent per user 4️⃣ **Email Service processes request** → Sends plaintext or HTML emails using SMTP 5️⃣ **User views email history** → Dashboard retrieves logs from the Email Service 6️⃣ **Billing Service enforces limits** → Users are restricted based on their quota

---

⚡ Key Features

✅ **Rate Limiting** → Prevents excessive API usage based on subscription tier
✅ **API Usage Tracking** → Logs email requests for monitoring
✅ **Subscription Management** → Users can upgrade or downgrade plans manually
✅ **Dockerized Deployment** → All services run inside Docker containers for easy scaling and maintenance

---

🌐 Next Steps

✅ Deploy using **Docker Compose** for container orchestration
✅ Implement **detailed logging** for tracking and debugging
✅ Optimize API performance with **efficient database queries**