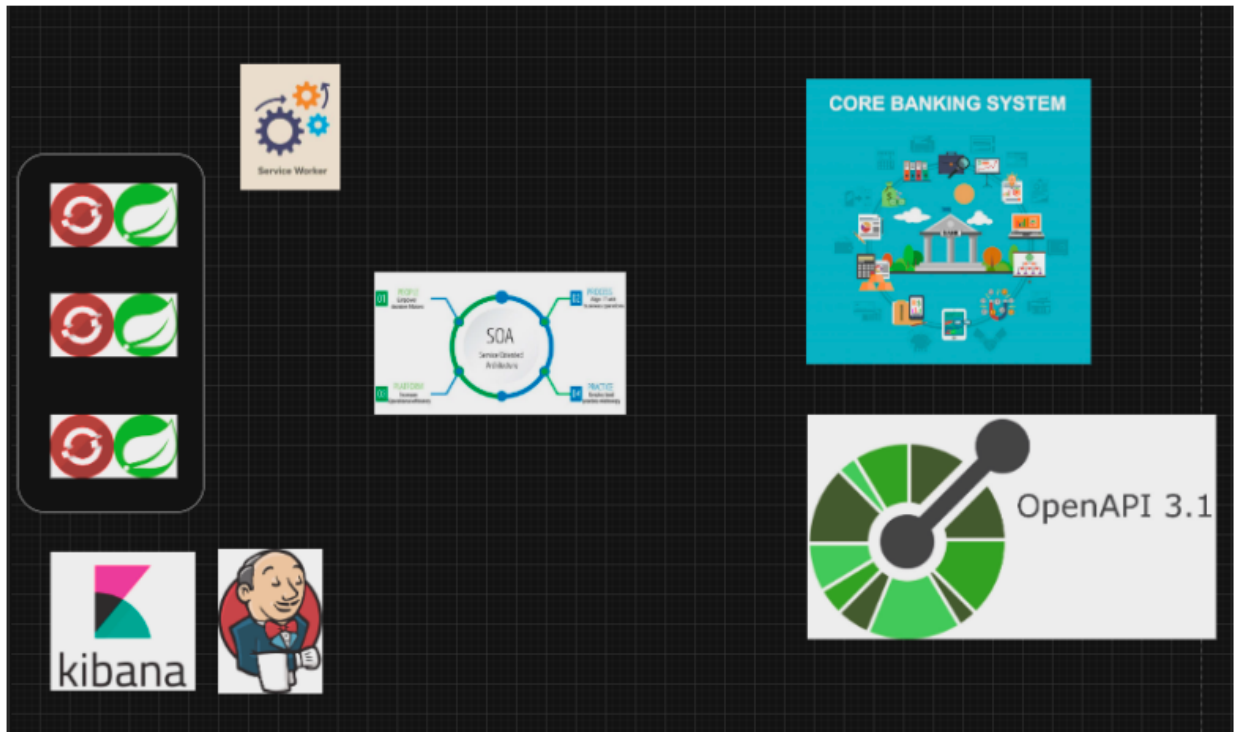


1. PT XYZ merupakan perusahaan fintech yang sedang membangun aplikasi pinjol, untuk membuat aplikasi tersebut maka PT XYZ menggunakan arsitektur dan tech stacks sebagai berikut :



- a. Microservice Development menggunakan Spring Boot dan Openshift
  - i. Microservice Development : Spring Boot merupakan salah satu frame work yang powerful dalam membangun aplikasi berbasis arsitektur microservice
  - ii. Penggunaan Container (Containerization) : Memanfaatkan teknologi container dengan menggunakan openshift untuk deployment
- b. CI/CD Pipeline
  - i. Mengimplementasikan Jenkins Pipeline untuk proses automasi build, testing, dan deployment serta mengintegrasikan testing tools didalam CI/CD Pipeline
- c. Logging with Kibana
  - i. Mengimplementasikan logging pada microservice untuk memonitor service dan mendapatkan informasi yang relevan

- ii. Kibana menjadi salah satu tools untuk memberikan solusi dalam memonitoring log analysis dan visualisasi, serta dapat digunakan untuk memonitoring health performance dari microservice
- d. Service Oriented Architecture
  - i. Penerapan Service Oriented Architecture (SOA) untuk memastikan komunikasi antar service dengan interface yang jelas
- e. API Management dengan OpenAPI
  - i. Pemanfaatan OpenAPI dalam development untuk menentukan standar Interface untuk RestFul API ( request and response), serta pembuatan dokumentasi API

### **Kesimpulan :**

Core Banking System dibangun dengan menggunakan arsitektur microservice sebagai fondasi dengan menggunakan prinsip SOA. Workflow dimulai dari design menggunakan OpenAPI untuk standar Interface yang digunakan, kemudian development menggunakan SpringBoot serta pemanfaatan teknologi Container menggunakan Openshift. Kemudian mengimplementasikan CI/CD Pipeline dengan Jenkins lalu dilanjutkan dengan proses monitoring dan logging dengan Kibana.

## 2. Rancangan API untuk peminjaman online:

- a. **POST**  
/api/loan/apply: Pengajuan pinjaman dengan total tagihan dan tenor pembayaran
- b. **GET**  
/api/loan/{user\_id}: Mengembalikan data pinjaman yang diajukan oleh user berdasarkan user id
- c. **GET**  
/api/bills/{user\_id}: Mengembalikan data tagihan yang harus dibayarkan (total dan batas pembayaran)

d. **POST**

/api/loan/{loan\_id}/bayar: Melakukan pembayaran tagihan berdasarkan id peminjaman.