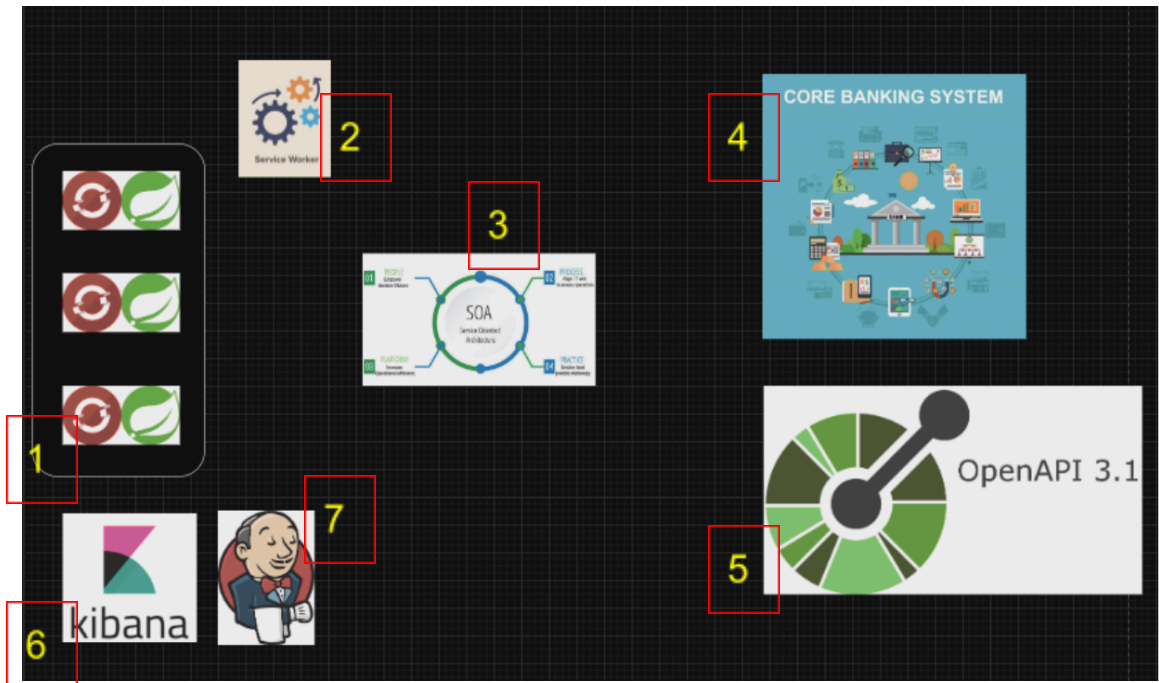


1. Case : PT. XYZ adalah sebuah perusahaan fintech yang ingin mengembangkan mobile apps mereka, dalam upaya menjangkau pengguna yang lebih luas mereka ingin mengembangkan aplikasi pinjaman online.

Dengan Arsitektur dibawah ini:



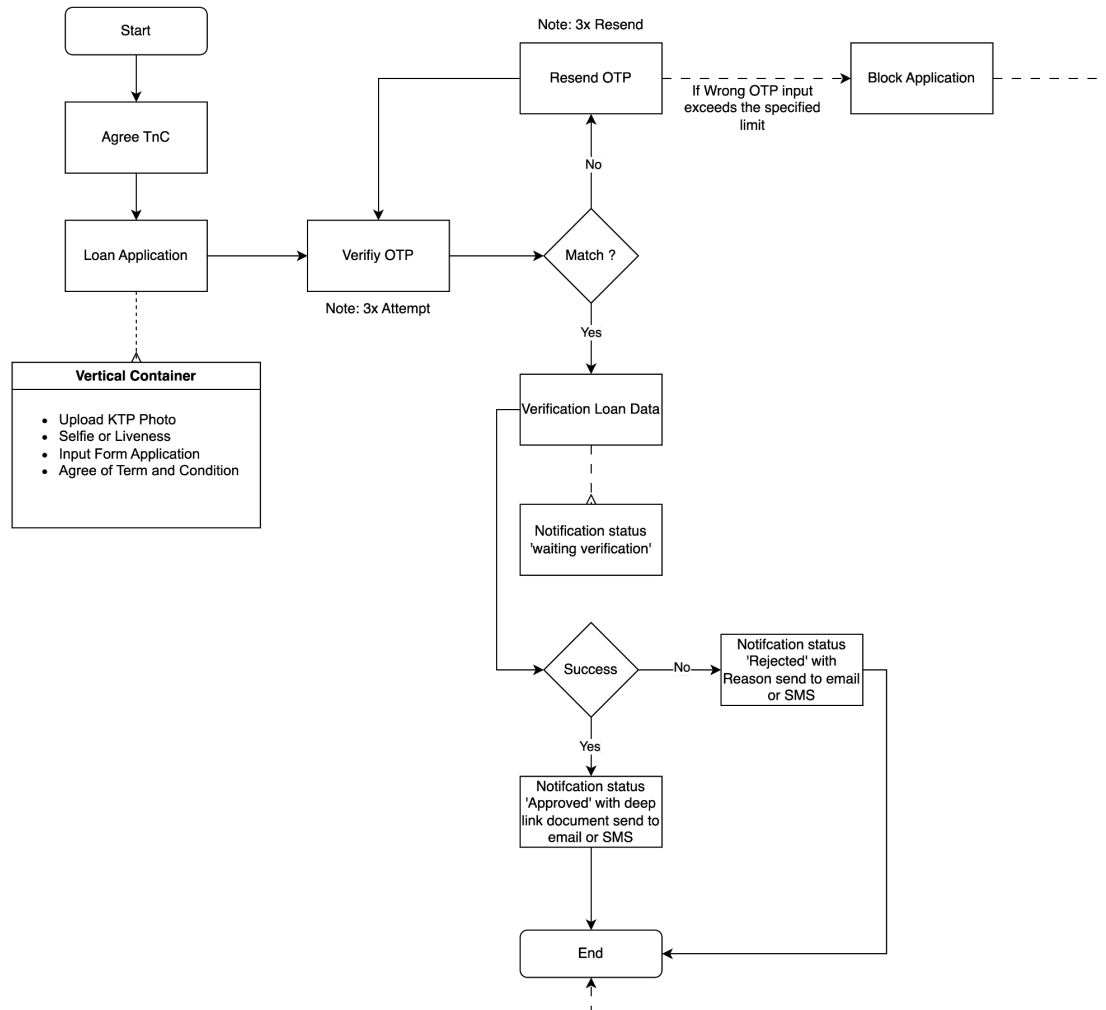
Dari arsitektur yang digunakan diatas untuk memaksimalkan aplikasi ada beberapa integrasi yang digunakan:

1. **Spring Boot dan GitLab:** Spring Boot merupakan kerangka kerja utama untuk mengembangkan backend aplikasi pinjaman online. Ini akan digunakan untuk membangun layanan-layanan web yang mengelola logika bisnis, integrasi dengan database, dan menyediakan antarmuka API untuk aplikasi klien. Sedangkan GitLab akan digunakan untuk mengelola code aplikasi pinjaman online. Developer menggunakan GitLab untuk mengelola code, sehingga dapat melakukan kolaborasi, dan mengotomatisasi alur kerja pengembangan dengan CI/CD atau penggabungan secara bersamaan dalam repository serta untuk proses deployment selanjutnya yaitu staging maupun production environment.
2. **Service Worker:** Service Worker akan digunakan dalam aplikasi klien (mobile apps) untuk menyediakan kemampuan offline atau kemampuan data secara lokal. Ini memungkinkan pengguna untuk mengakses aplikasi dan melakukan transaksi bahkan ketika tidak terhubung dengan internet.
3. **SOA (Service-Oriented Architecture):** SOA akan digunakan untuk membagi aplikasi menjadi layanan-layanan terpisah yang dapat diakses melalui antarmuka yang didefinisikan dengan jelas. Ini memungkinkan fleksibilitas dalam pengembangan, penyesuaian, dan skala aplikasi. Seperti :

- a. Pemisahan service, service ini dapat dipisahkan berdasarkan fungsi dari setiap service.
 - b. User Interface Pengguna, layanan pengajuan pinjaman dapat memiliki antarmuka yang mendefinisikan operasi-operasi seperti mengajukan pinjaman, melihat status pinjaman.
 - c. Internal Gateway, biasanya digunakan untuk komunikasi dengan service pengguna dengan microservice yang dibangun.
 - d. Rendahnya ketergantungan antar service, karena setiap service dibangun sesuai dengan fungsi masing-masing dan tidak bergantung.
4. **Core Banking System:** Core Banking System akan terhubung dengan backend aplikasi pinjaman online untuk memproses transaksi keuangan, manajemen rekening, dan manajemen risiko. Ini memastikan integritas data dan keamanan transaksi.
5. **OpenAPI 3.1:** Digunakan untuk mendefinisikan dan mendokumentasikan antarmuka API aplikasi pinjaman online yang akan diakses oleh aplikasi klien. Ini memastikan konsistensi dan pemahaman yang jelas tentang cara mengakses akses backend. Yang biasanya digunakan untuk integrasi dengan merchant jika dibutuhkan.
6. **Kibana:** Kibana akan digunakan untuk memantau kinerja aplikasi pinjaman online, termasuk metrik penggunaan, kinerja transaksi, dan masalah yang mungkin muncul yang dapat divisualisasikan dengan dashboard yang akan dibangun.
7. **Stakeholder:** Stakeholder dapat melihat report dan pemantauan kinerja aplikasi dengan kibana serta melakukan analisa data yang dapat digunakan untuk menentukan conclusion dan dapat digunakan untuk memprediksi proses berikutnya yang dapat dilakukan untuk mengembangkan product.

2. Buatlah rancangan spesification api yg relate untuk menghandle salah satu fitur pinjol ini (dalam bentuk apapun):

Berikut alur dalam proses pinjaman pinjol:



API yang dibutuhkan

1. Endpoint: /verify-otp

Deskripsi:

Endpoint ini digunakan untuk memverifikasi OTP yang dimasukkan oleh pengguna.

Metode HTTP: **POST**

Body Request:

```
{  
  "phone_number": "08123456789",  
  "otp_code": "123456"  
}
```

Response:

a. Success (HTTP Status 200 OK):

```
{  
  "status": "success",  
  "message": "Verifikasi OTP berhasil"  
}
```

b. Error (HTTP Status 400 Bad Request):

```
{  
  "status": "error",  
  "message": "Verifikasi OTP gagal, kode OTP tidak valid atau sudah  
kedaluwarsa",  
  "remaining_attempts": 2,  
  "can_resend": true,  
  "resend_attempts": 2  
}
```

Deskripsi Respons Gagal:

- *remaining_attempts*: Jumlah percobaan verifikasi OTP yang tersisa sebelum mencapai batas.
- *can_resend*: Menunjukkan apakah pengguna dapat mengirim ulang OTP.
- *resend_attempts*: Jumlah percobaan pengiriman ulang OTP yang tersisa sebelum mencapai batas.

2. Endpoint: /process-loan-application

Deskripsi:

Endpoint ini digunakan untuk memproses pengiriman aplikasi pinjaman setelah verifikasi OTP yang berhasil.

Metode HTTP: POST

Body Request:

```
{
  "phone_number": "08123456789",
  "otp_code": "123456",
  "user_data": {
    "name": "Dhesa Ardhiyanta",
    "email": "dhesa.ardhi@gmail.com",
    "dob": "1990-01-01",
    "address": "Main Street",
    "occupation": "Engineer",
    "bank_name": "Mandiri Bank",
    "account_number": "1234567890"
  },
  "loan_data": {
    "amount": 1000000,
    "term": 12,
    "interest_rate": 5
  },
  "selfie_image": "base64_encoded_image",
  "ktp_image": "base64_encoded_image"
}
```

Response:

- a. Success (HTTP Status 200 OK):

```
{
  "status": "success",
  "message": "Pengiriman aplikasi pinjaman berhasil diproses",
  "loan_application_status": "waiting_verification"
}
```

- b. Error (HTTP Status 400 Bad Request):

```
{
  "status": "error",
  "message": "Verifikasi OTP gagal, kode OTP tidak valid atau sudah kadaluwarsa",
  "remaining_attempts": 2,
}
```

```
"can_resend": true,  
"resend_attempts": 2  
}
```

3. Endpoint: /notify-loan-application-status

Deskripsi:

Endpoint ini digunakan untuk memberikan notifikasi kepada pengguna tentang status pengajuan pinjaman, baik itu "Approved" atau "Rejected", beserta alasan penolakan jika status adalah "Rejected". Dokumen akan dikirimkan melalui tautan dalam notifikasi "Approved".

Metode HTTP: POST

Body Request:

```
{  
  "loan_application_id": "789012",  
  "status": "Approved",  
  "reason": "Income verification successful",  
  "contact_method": "sms", // atau "email" tergantung method yang digunakan  
  // untuk pengiriman  
  "destination": "+1234567890", // Nomor telepon atau alamat email penerima  
  "document_link": "https://example.com/document/12345" // Tautan ke dokumen  
  // (hanya diperlukan jika status adalah "Approved")  
}
```

Response:

- a. Success (HTTP Status 200 OK):

```
{  
  "status": "success",  
  "message": "Notifikasi pengajuan pinjaman berhasil dikirim"  
}
```

b. Error (HTTP Status 400 Bad Request):

```
{  
  "status": "error",  
  "message": "Gagal mengirim notifikasi pengajuan pinjaman"  
}
```