

# Plan intervention

## Pré-requis d'installation :

- JDK 11
- IDE
- Git
- Docker

## Table des matières

Jour 1 (présentiel) : Etat des lieux.....	2
Output.....	2
Jour 2,3,4 (présentiel) : Architecture micro-services.....	3
Introduction.....	3
Exemple micro-service.....	3
Design patterns.....	3
Jour 5 (présentiel).....	4
Jour 6 et 7.....	4
Jour 8 à 11.....	4

# Jour 1 (présentiel) : Etat des lieux

## Environnement de développement

- IDE, plugins
- Outil de build
- Stack technologique, Frameworks
- Pratique des tests unitaires
- Cycle Code/ Deploy / Test

## Gestion des sources

- Dépôt utilisés, branches,
- workflow de collaboration

## Pilotage

- Planification et Suivi des travaux
- Implication du métier

## Qualité :

- Revue de code
- Analyse statique qualité
- Analyse de vulnérabilité

## Releasing

- Processus de release
- Packaging
- Dépôts d'artefacts

## Tests d'intégration

- Environnement d'intégration
- Types de test d'intégration

## Procédures de déploiement

- Relation avec les gestionnaire d'infra
- Procédure de déploiement
- Gestion des migrations de schéma

## Surveillance et observabilité

- Mécanismes de monitoring
- Statistiques d'utilisation
- Feed-back utilisateurs finaux
- Gestion des issues

## Output

- Document de synthèse
- Identification d'une application BCEAO typique et objectifs d'évolution : Migration socle 3, micro-service ?

# Jour 2,3,4 (présentiel) : Architecture micro-services

## Introduction

Objectifs d'une architecture micro-services  
Contraintes et facteurs de réussite  
Décomposition en micro-services  
Design patterns  
Services transverses nécessaires : Framework ou Infra

## Exemple micro-service

Présentation d'une architecture micro-service reliés à une problématique métier BCEAO  
Présentation de 2 micro-services développés avec SpringBoot 3  
Revue de code, Test, Packaging (jar, docker, natif)  
Environnement de développement d'un service :

- Mocking des dépendances
- Services de support

## Design patterns

Configuration externalisée et centralisée

### Interactions entre services

Introduction  
Interactions RPC  
Messaging

### Cohérence des données et transactions

Saga Pattern

### Logique métier

Transactional Script Pattern  
Patterns orienté objet  
Event Sourcing Pattern

### Requêtage

API Composition Pattern  
CQRS Pattern

### API Externe

Gateway Pattern

### Sécurité

- Stratégies
- Propagation de Jeton
- Client-credentials

- Sécurisation Message broker

#### Observabilité

- Health
- Métriques
- Tracing

## **Jour 5 (présentiel)**

Identification du POC BCEAO

Définition de l'architecture

Définition des APIs

Répartition des groupes projet

## **Jour 6 et 7**

Démarrage de projet, organisation des tâches

Choix des starters

## **Jour 8 à 11**