**Architecture de L’ERP SR-Manager**

**Approche Micro Service**

|  |  |
| --- | --- |
| * L’approche micro services est une approche d’architecture et de développement d’une App composée de plusieurs micro services * L’idée est de découper un grand problème en plusieurs petits voir micro unité implémentée sous forme de micro service * Chaque micro service répond à une fonctionnalité * Chaque micro service est développé, testé et déployé séparément et indépendant des autres. * Chaque MS peut être développé avec une technologie différente des autres(Java, C++, C#, PHP, Python…) * Chaque MS tourne dans un processus séparé * La communication être les MS est régit par un système de communication léger basé sur REST | **BD A**  **BD B**  **BD C**  **App Web**  **App Mobile**  Gateway  **BD ERP** |

**Architecture de chaque micro service**

Backend

COUCHE DAO

COUCHE METIER

COUCHE WEB

-Paramsdb

-Marchedb

* Soustraitancedb
* Mscontactdb
* Atelierdb
* Grhdb
* Logistiquedb
* Materieldb
* Achatdb
* Tresoreriedb
* Decompte

Frontend

Angular11,

HTML,

SCSS,

Bootstrap,

JavaScript,

jQuery

Les Bases de données

**Les bases de données :**

* Toutes les bases de données des micros service sont des base de données **MongoDb** non Sql.
* Chaque micro service a sa propre base de données.
* Toutes Les données de ERP seront ensuite centralisées sur une grande base de données **MongoDb** pour des raisons d’analyse et d’intelligence artificielle

**Frontend :**

* Application web développée avec le framwork Angular 11,HTML5, Bootstrap4, jQuery...

**Backend : Java**

* Solution distribuée 100% java développée avec une approche micro services :
  + (Ms-Parametres, Ms-Marche , Ms-Contact, Ms-Soutraitance, Ms-Atelier, Ms-Grh, Ms-Logistique, Ms-Materiel, Ms-Achat, Ms-Tresorerie, Ms-Decompte…)
* Chaque application est développée sous forme de micro service.
* Chaque micro service est compose des couches suivantes :
  + **Couche DAO** :
    - Elle permet de gérer l’accès a la base des données
    - Elle est implémenté en utilisant les outils suivants  : Spring Data, Reactive Mongo Repository
  + **Couche Métier ou service :**
    - Elle encapsule tous les objets métiers du micro service.
    - Elle centralise toutes la logique métier, tous les traitements et les calculs du micro service.
    - Elle constitues le noyau du micro service.
  + **Couche Web ou contrôleur:**
    - Elle offre une interface d’accès à l’application via des contrôleurs Rest Api

**Service d’enregistrement : Java**

* L’enregistrement et la publication des adresses (URL et Port) des micros service est assurée par un **micro service java** dédie en utilisant :
  + Spring Eureka Server
  + Spring Eureka Client

**Service geteway : Java**

* La distribution, le routage et l’acheminement des requetés venants de toutes les composantes de l’application et assurée par un **micro service java** dédie en utilisant :
  + Spring geteway
  + Spring Cloud