

Architecture applicative TP

Gaël Roustan (Argonauts)

2024-02-05

Abstract

Ce document contient les étapes pour la réalisation du TP d'introduction aux architectures.

Architectures

Introduction

Le projet BIB est une application de gestion de bibliothèques avec achat possible et consultation des livres.

Objectif

L'objectif du TP est d'illustrer la mise en œuvre des différentes architectures et de mettre en évidence les avantages et inconvénients de chacune.

Définition détaillée du besoin

La gestion de bibliothèque est déléguée à un ou plusieurs modérateurs. Votre application doit suivre les principes du paradigme du développement orienté objet.

Pour différentes entités de votre solution, comme le livre, il faut pouvoir :

- Créer un livre
- Mettre à jour un livre
- Consulter un livre
- Supprimer un livre

Les caractéristiques minimales d'un livre sont :

- auteur
- titre
- contenu

La librairie et la bibliothèque sont 2 ensembles distincts.

Les utilisateurs doivent pouvoir ajouter des livres papier ou numérique dans la bibliothèque.

A l'ajout, il faut vérifier l'exactitude et la cohérence de l'auteur et du titre.

Il n'y a pas de quantité limite sur le stock numérique de livre.

Il n'y a pas de contenu pour un livre uniquement papier.

Il faut laisser la possibilité d'acheter un livre numérique si déjà en papier dans la bibliothèque

Fonctionnalités avancées

Les fonctionnalités avancées sont à intégrer uniquement si le core de base de la solution est implémentée.

- Lecture du code IBAN
- Prise en photo de la couverture du livre
- OCR sur la couverture et la 4ème de couverture
- Validation des revues
- Evaluation positive ou négative en fonction de la critique laissée par l'utilisateur
- Vérifier la consistance des données
- Avertir l'utilisateur si l'auteur est mort avant d'avoir terminé
- Proposer des livres soit sur des genres identiques ou bien activer l'option, proposer des genres connexes

Modalités

La pédagogie ici consiste à implémenter chacune des architectures.

Les étapes importantes sont :

1. prise de connaissance du besoin métier
2. traduction du besoin en cahier des charges
3. validation du cahier des charges auprès du client
4. méthode MERISE pour obtenir le schéma de la base de données (au moins 1 première version)

Différents niveaux d'architecture

Les différentes architectures que nous mettrons en œuvre sont les suivantes :

1. Architecture monolithique centralisée
2. Architecture Client/Serveur 2-tier basée sur TCP
3. Architecture 3-tiers
4. Architecture orientée services basée sur HTTP/JSON
5. Architecture MicroServices basée sur gRPC et un répartiteur de messages

Ces architectures sont mises en place au fur et à mesure que le développement avance afin de fournir le plus rapidement possible une solution.

Tests

Ne négligez pas les tests.

CI

Ne négligez pas la chaîne d'intégration continue.

Frameworks

L'introduction du framework Django et/ou d'autres dépendances est tout à fait possible à partir du niveau 3.

Les paliers

Architecture monolithique centralisée

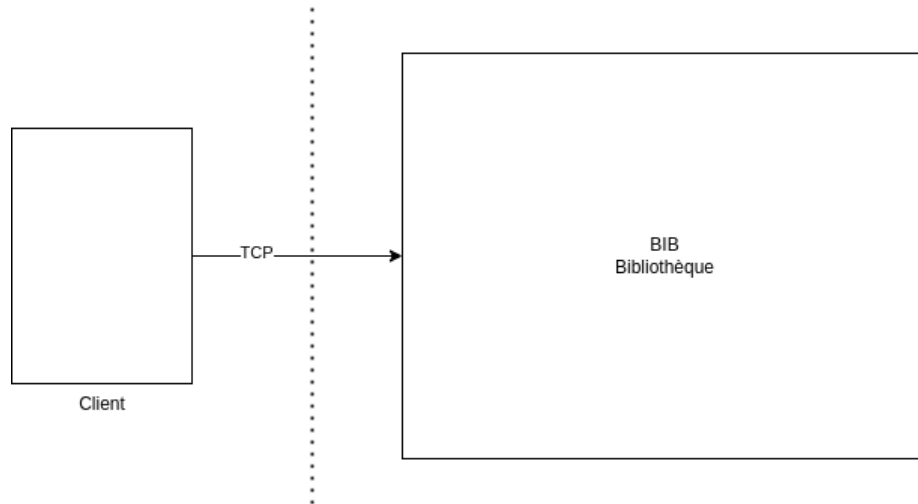


Bibliothèque local terminal

- Affichage dans le terminal
- Créer un livre avec un nom, un tag, une image (ASCII en noir et blanc)
- Supprimer un livre
- Afficher la liste de tous les livres
- Afficher le détail d'un livre
- Sauvegarder la bibliothèque dans un fichier plat au format JSON, le format JSON attendu est libre (possibilité d'utiliser le module

json ou pickle)

Architecture Client/Serveur 2-tier basée sur TCP

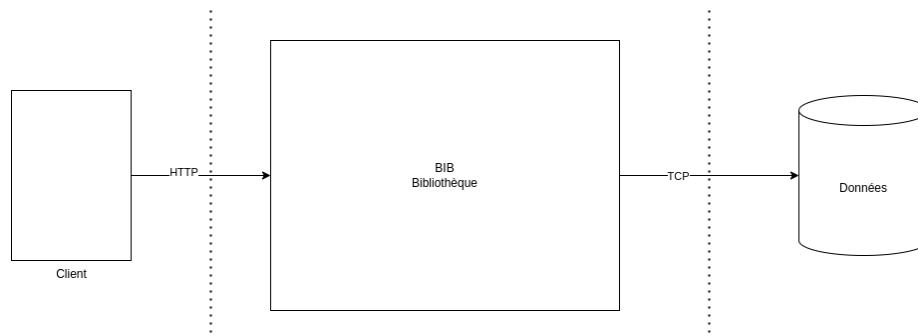


Bibliothèque Notations en réseau avec d'autres lecteurs (TCP)

- Implémenter à présent un client, toujours en console, qui communique en TCP avec le serveur développé dans les paliers précédents
- Le développement effectué dans les paliers précédents doit donc maintenant accepter des requêtes TCP

Pour l'implémentation en python : <https://docs.python.org/3/howto/sockets.html>

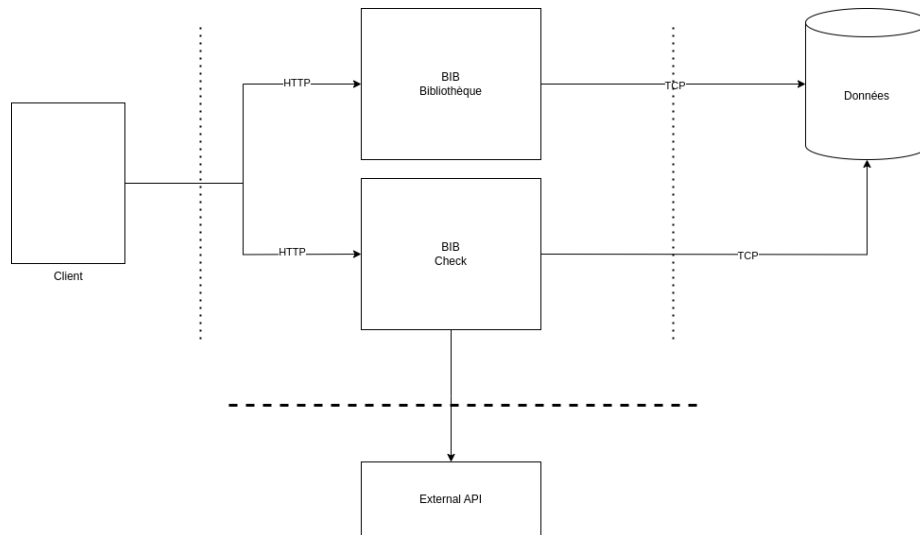
Architecture 3-tiers



Pour l'ajout de la 3ème couche et pour réaliser les tests, utiliser la base de données relationnelle SQLite3. Bibliothèque local terminal : 1h (sauvegarde dans base de données SQLite)

- Ajouter un mode de sauvegarde SQLite différent au moment du démarrage, ce nouveau mode de sauvegarde doit être SQLite

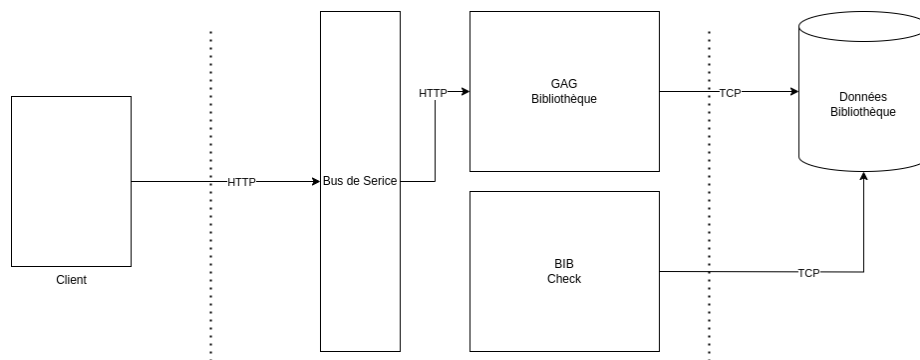
Architecture N-tiers



Restructurer l'application pour que les tiers suivants sont clairement identifiables :

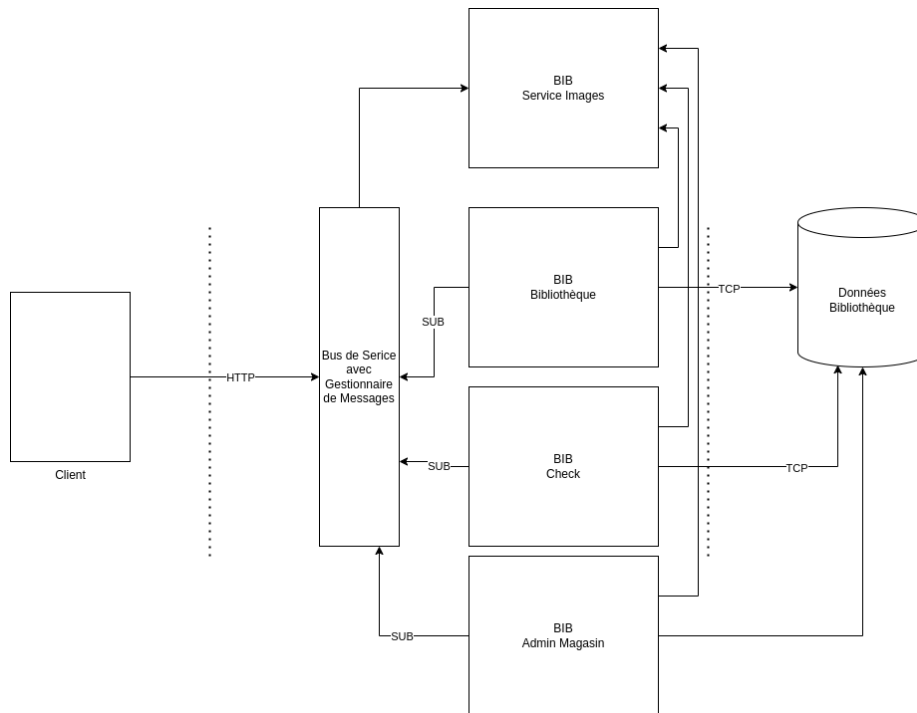
- Client console
- Serveur Bibliothèque avec la base de données
- Serveur Magasin avec la base de données

Architecture orientée services basée sur HTTP/JSON



Pour permettre la création de plusieurs services identiques pour assurer la redondance, intégrer un bus de service qui fera l'intermédiaire.

Architecture MicroServices basée sur gRPC et un répartiteur de messages



- Basée sur http en JSON
- Basée sur gRPC, utiliser la bibliothèque

Livrable attendu

Un repo GIT contenant le projet BIB avec une branche au minimum par type d'architecture.