

Application d'architectures distribuées

Rapport de mi-semestre

Travail à réaliser

Lors de ce projet il est demandé de développer une application mobile permettant de créer un lecteur de musique commandé par la voix. Il sera nécessaire d'implémenter 4 applications distinctes communiquant entre-elles.

- une application mobile centrale servant de base pour le lecteur de musiques,
 - une application de retranscription des commandes vocales en transformant des commandes vocales en texte,
 - Une application de compréhension de la parole, qui doit permettre d'interpréter le texte généré par l'application de retranscription et en extraire des actions à réaliser.
 - Une quatrième application permet de gérer une base de donnée de musiques et de faire du streaming de ces musique à la demande des utilisateurs.
- Chacunes de ces applications doivent être implémentées sur des serveurs et accessibles depuis l'application mobile.

Travail réalisé

J'ai développé une partie de l'application mobile en java sous android Studio, celle-ci me permet d'enregistrer la commande vocale à envoyer au serveur de transcription et de la stocker temporairement dans la mémoire du téléphone, lors de l'appui sur un bouton.

J'ai développé sur un serveur nodeJs, en utilisant le framework ExpressJs l'application de retranscription des commandes audio vers du texte. Je fais appel à l'api "Speech-to-Text" de Google, qui prend en entrée un fichier audio et fournit en retour de requête, le texte correspondant à la transcription de la commande audio passée en paramètre.

J'ai développé en java l'application de compréhension de la parole, permettant d'identifier les commandes demandées par l'utilisateur. En passant un texte correspondant à une commande en entrée, cette application identifie l'action à réaliser en conséquence, ainsi que certains paramètres en relation avec cette commande.

Plusieurs commandes sont reconnaissables:

- Jouer une musique (action: jouer musique, paramètre: musique)
- Jouer les musiques d'un artiste (action: jouer musique, param: artiste)
- Jouer les musiques d'un album (action: jouer musique, param: album)
- Afficher les musiques (action afficher musiques)

- Afficher les musiques d'un album (action afficher musiques, param: album)
- Afficher les musiques d'un artiste (action afficher musiques, param: artiste)
- Afficher les albums (action afficher albums)
- Afficher les albums d'un artiste (action afficher albums, params:)
- Musique suivante (action musique suivante)
- Musique précédente (action musique précédente)
- Augmenter le son
- Diminuer le son
- Couper le son
- Remettre le son
- Arrêter la musique
- Mettre la musique en pause
- Reprendre la musique
- Recommencer la musique

Cette reconnaissance est effectuée à l'aide d'expressions régulières en sortie, et retourne l'action à réaliser ainsi que les paramètres correspondants.

Travail qu'il me reste à réaliser

Pour le reste du semestre, il me reste à approfondir l'application mobile android, en proposant des écrans différents, notamment pour permettre d'afficher la liste des différents albums et musiques en fonction des commandes de l'utilisateur, ainsi qu'une vue sur laquelle sera affichée la musique en cours d'écoute et ses informations.

Il faudra que j'implémente l'application en charge de la base de donnée des musiques et du service de streaming de ces musiques. Pour cette application, je pourrai utiliser la technologie ICE pour faire le lien entre l'application mobile sous android et le serveur, ainsi qu'une base de donnée MongoDB.

De plus il faudra que je mette en place un modèle de communication par message à l'aide d'un serveur GlassFish en utilisant des JMS pour relier l'application android et le serveur de reconnaissance des commandes à réaliser. Pour améliorer l'application de compréhension des commandes, je pourrai mettre en place un système d'Intelligence Artificielle, pour reconnaître directement si le titre donné lors d'une demande d'écoute ou d'affichage correspond à une musique, un album ou un artiste. Pour ce faire, je prévois de mettre en place une base de donnée PostgreSQL et des EntityBeans, pour stocker et gérer les différentes correspondances.

Il me reste également à développer un modèle de communication entre l'application mobile et le système de retranscription de commande audio en texte, en utilisant un système de requête http, par lequel j'enverrai le fichier audio généré par l'enregistrement de la commande de l'utilisateur.