Florent VANHOLLEBEKE

**Projet P8 : TRIOF**

**Partie I : Questions préliminaires**

**1) Généralités sur l’intégration d’une IA dans une application existante**

1) L’architecture client/serveur porte sur l’organisation d’échanges d’informations (le plus souvent par le biais d’un réseau tel qu’internet, hors cas du client/serveur sur la même machine). Le client (machine en demande) formule des requêtes au serveur, machine rendant un service au client en réponse. Cet échange d’informations se fait sur la base de protocoles de communication, lesquels diffèrent selon la nature du serveur (par exemple protocole http pour un serveur web, ou le protocole FTP pour un serveur de fichiers). Il existe différentes sortes d’architecture client/serveur. Les plus simples sont le peer-to-peer (la même machine sera le client et serveur tour à tour selon si elle reçoit ou fournit les données) ou l’architecture à deux niveaux selon laquelle le serveur fournit la ressource directement sans faire appel à une tierce machine. Il existe toutefois des architectures plus complexes telles que l’architecture à trois niveaux où une tierce machine appelée middleware sert d’intermédiaire entre le client et le serveur détenteur de la ressource, le middleware (ou aussi dit serveur d’application) étant lui-même client du serveur dit de données. Les clients eux-mêmes peuvent faire l’objet de catégorisations entre client léger, lourd ou riche selon le degré de sophistication et de travail leur étant demandé (emploi ou non de javascript par exemple). Avant l’apparition de cette architecture, les réseaux informatiques étaient configurés autour d’un ordinateur central seul capable de traitements, les terminaux des utilisateurs n’étaient que de simples écrans. Les avantages de l’architecture client/serveur non centralisée sont notamment le fait de pouvoir supporter une charge de travail importante (nombreuses requêtes de clients) si les données sont redondées sur différents serveurs, une plus haute disponibilité des données, la communication possible entre clients et un moindre coût de maintenance.

2) Un service web est un protocole d’interface informatique permettant l’échange de données entre applications et systèmes divers dans le cadre d’une architecture distribuée (ou autrement dit client/serveur). Pour l’heure, le protocole repose essentiellement sur TCP (http). Charque service web dispose de son URI (Uniform Resource Identifier) c’est-à-dire son adresse unique à l’instar de l’URL d’un site classique.

Dans le cadre d’un service web, nous parlons de demandeur et de fournisseur. Ces termes remplacent respectivement le client et le serveur tout en conservant le même esprit. Le demandeur produit des agents de requêtes alors que le fournisseur forme des agents. Agent et agent de requête communiquant ensemble avec la même sémantique (exemple JSON).

Il existe plusieurs technologies de services web, les principales étant les services web de type REST et de type WS-\*.

Parmi les avantages d’un service web, il y a le fait de pouvoir recevoir des informations d’un serveur distant sans devoir stocker toutes les données sur sa machine (gain de temps et de mémoire). La communication peut se faire par le biais de plateformes très différentes, les formats sont standardisés. Par ailleurs, ce service distant peut recevoir un grand nombre de requêtes à la fois. Enfin, les échanges peuvent être chiffrés (par exemple le SSL présent dans le protocole HTTPS) dans le cadre de transactions e-commerce.

3) Il est possible de sauvegarder un modèle d’intelligence artificielle dans un script inclus dans le code d’un service web. Cela permet à un client (agent de requête) de requêter le service web qui délivrera en réponse (agent) des informations ayant fait l’objet d’un traitement dudit modèle.

**2) Généralités sur les solutions Cloud de Computer Vision**

1) Custom Vision de Azure Cognitive Services est une solution cloud de Microsoft Azure. Custom Vision permet la classification d’images, mais également la détection d’objets.

Il permet d’entrainer ses propres modèles d’IA à contrario du service Computer Vision qui est le fruit intégral du travail de Microsoft et solution clé en main. Ici nous pouvons fournir nos données (par exemple nos images), créer le modèle et l’entrainer soi-même tout en bénéficiant d’une base réalisée par Microsoft, et enfin déployer son modèle.

Ensuite, le modèle peut être consommé à partir d’une application cliente en utilisant son ID de modèle avec le point de terminaison (REST) et la clé de ressource dans laquelle vous l’avez déployé.

La tarification est à l’utilisation. Il y a deux types d’instance :

* Gratuit : 2 transactions par seconde (5000 images gratuites, 1h d’entrainement par mois, 10000 prédictions mensuelles, 2 projets maximum)
* Standard : 10 transactions par seconde (100 projets maximum). Surcoût possible : 2$/ toutes les 1000 transactions, $10 par heure de calcul, $0.70 toutes les 1000 images.

2) Amazon Web Services dispose d’un service de vision par ordinateur appelé Amazon Rekognition Custom Labels. Pour sa part, Google propose AutoML Vision. Ces deux services permettent d’inclure ses propres images.

3)

**3) L'application Triof : analyse et compréhension du problème**

Il s’agit d’une application basée sur le framework Flask.

Il y a 4 pages différentes sur l’application : home, insert, type, et confirmation. Le fichier python triof\_app.py est le script général comprenant les routes (url).

Le fichier utils.py regroupe les fonctions pour réaliser les traitements de l’api.

Le répertoire templates comprend le code HTML des 4 pages. Le javascript est inclus dans le code HTML.

Le répertoire assets sert à améliorer le visuel de l’application en comprenant les fichiers CSS.

Le répertoire caméra comprend les photos pour tester l’api.