

Utilisation de la vision assistée par ordinateur pour valider le contenu d'annonces immobilières : Quebec-RealEstate (QRE)

par

Roland Wowo GAMY

RAPPORT DE PROJET DE 15 CRÉDITS PRÉSENTÉ À L'ÉCOLE DE  
TECHNOLOGIE SUPÉRIEURE COMME EXIGENCE PARTIELLE À  
L'OBTENTION DE  
LA MAITRISE EN GÉNIE LOGICIEL  
M. Ing

MONTREAL, LE 8 MAI 2020

ÉCOLE DE TECHNOLOGIE SUPÉRIEURE  
UNIVERSITÉ DU QUÉBEC



Roland Wowo Gamy, 2020



Cette licence [Creative Commons](https://creativecommons.org/licenses/by-nc-nd/4.0/) signifie qu'il est permis de diffuser, d'imprimer ou de sauvegarder sur un autre support une partie ou la totalité de cette œuvre à condition de mentionner l'auteur, que ces utilisations soient faites à des fins non commerciales et que le contenu de l'œuvre n'ait pas été modifié.

**PRÉSENTATION DU JURY**

CE RAPPORT DE PROJET A ÉTÉ ÉVALUÉ

PAR UN JURY COMPOSE DE :

Professeur Alain April, directeur de projet,  
Département de Génie Logiciel et TI à l'École de technologie supérieure

Professeur Abdelaoued Gherbi, président du jury  
Département de Génie Logiciel et TI à l'École de technologie supérieure



## **AVANT-PROPOS**

Depuis l'avènement des applications Web dynamique où les utilisateurs peuvent être sollicités à saisir des données de toutes sortes (textes, images, vidéos, etc..), il est parfois très difficile de valider immédiatement ces données surtout lorsqu'il s'agit d'image ou de vidéo. Dans les applications Web de publication d'annonces par exemple, les administrateurs sont obligés d'avoir un modérateur qui valide manuellement chaque annonce avant qu'elle ne soit rendue disponible au public. Ceci engendre généralement des délais et ceux qui décident de publier directement l'annonce sans vérification humaine au préalable s'exposent au risque d'avoir des contenus inappropriés.

Or de nos jours, avec la maturité acquise dans les domaines de l'Intelligence artificielle, du traitement de mégadonnées (Big-Data) et de l'apprentissage machine, il est possible d'utiliser certaines de ces techniques pour valider automatiquement et efficacement le contenu des annonces.

Pour des applications avec un fort trafic et un nombre élevé d'utilisateurs, l'utilisation du Big-Data combiné avec un algorithme d'apprentissage machine, pourrait permettre d'identifier rapidement et bannir les utilisateurs récidivistes de l'application sans avoir systématiquement besoin d'intervention humaine.

Cette recherche appliquée vise à investiguer la possibilité d'utiliser ces techniques dans une étude de cas.



## **REMERCIEMENTS**

J'aimerais, de tout cœur, remercier profondément le professeur Alain April, directeur de ce projet et professeur au département de Génie Logiciel et TI à l'École de Technologie Supérieure d'avoir accepté de me superviser et de m'encadrer patiemment tout au long de la réalisation de ce projet. Le professeur April a su mettre à ma disposition toutes les infrastructures logicielles et matérielles nécessaires pour réaliser ce projet de recherche appliquée avec succès. Ses conseils, son expertise et son champ de compétence font de lui un professeur de référence dans le domaine de la gestion et le traitement de mégadonnées.

J'aimerais aussi remercier ma famille pour le soutien moral et financier qu'elle n'a cessé de manifester à mon égard et remercier particulièrement ma conjointe, qui en dépit de sa grossesse, a pu me supporter durant toutes ces nuits blanches et m'apporter réconfort quand besoin y était.

Je ne saurais finir sans remercier mon employeur, qui a pu mettre à ma disposition un accès gratuit aux solutions infonuagiques sous licence de Microsoft et à sa plateforme infonuagique (Microsoft Azure).





# Utilisation de la vision assistée par ordinateur pour valider le contenu d'annonces immobilières : Quebec-RealEstate (QRE)

ROLAND WOWO GAMY

## RÉSUMÉ

De nos jours les applications Web à fort trafic permettant à leur utilisateur de poster des contenus de toutes sortes (c.-à-c. textes, images, vidéos, etc.) et de les afficher quasi instantanément, peinent à valider en temps réels le contenu des informations saisies par les utilisateurs. C'est le cas par exemple des réseaux sociaux tel que Facebook, Twitter ou Instagram. Une des solutions qui permettrait de ne pas bloquer temporairement les contenus téléchargés des utilisateurs, tout en garantissant un contenu adéquat, serait d'utiliser une intelligence artificielle capable de reconnaître du contenu inapproprié et de l'exclure immédiatement. C'est le cas par exemple du réseau social Facebook qui dévoila à sa conférence de technologie F8 du 1er mai 2019 [1], une nouvelle approche améliorée [2] de détection et de reconnaissance d'objet basée sur la segmentation d'images [3] ainsi que le traitement du langage naturel.

L'application de ces techniques à leur modèle d'intelligence artificielle a permis de l'améliorer de façon significative, aidant du coup à empêcher les utilisateurs de diffuser des contenus de tueries de masse, de pornographie ou d'actes de violence pour ne citer que quelques exemples. Malgré le fait que l'ordinateur soit plus rapide et plus apte que l'humain à détecter des anomalies, il arrive parfois que certains contenus soient non détectés ou mal détecté dû aux volumes importants de données à traiter ou à une problématique reliée à l'échantillonnage des données d'entraînement des algorithmes d'intelligence artificielle.

Cette recherche expérimente avec l'utilisation d'algorithmes d'intelligence artificielle combinée à des techniques de traitement des mégadonnées dans une étude de cas d'un système d'annonce immobilière afin de pré valider automatiquement le contenu d'annonces téléchargées par les utilisateurs. Pour ce faire, l'essai d'un modèle de reconnaissance d'image à large spectre est expérimenté, par la suite, un modèle basé sur l'intelligence artificielle,

X

spécifiquement adapté pour le domaine de l'immobilier est proposé et entraîné avec une grande quantité de données (c.-à-d. photos, images de propriétés, etc..) que les utilisateurs publieront dans le système.

Le résultat de cette expérience démontre qu'il est possible de détecter la nature des éléments que publient les utilisateurs sur les plateformes Web et du même coup valider en temps réel (presque quasi instantanément) ces éléments.

Mots-clés : L'informatique en Nuage, Intelligence artificielle, Vision assistée par ordinateur, Algorithme de Reconnaissance d'image, Apprentissage machine, Mégadonnées

# Use of computer vision to validate the content of real estate ads: Quebec-RealEstate (QRE)

ROLAND WOWO GAMY

## ABSTRACT

Nowadays high-traffic Web applications that allow users to post content of all kinds (text, images, videos, etc.) and display them almost instantaneously, struggle to validate in real time the content of information entered by the users. This is the case for social networks such as Facebook, Twitter or Instagram. One of the solutions that would not temporarily block the "posts" of users while ensuring adequate content is to use an artificial intelligence capable of recognizing inappropriate content and exclude them immediately. For example, Facebook unveiled at its F8 technology conference on May 1, 2019 [1], a new and improved approach [2] to object detection and recognition based on image segmentation [3] and natural language processing.

The application of these techniques to their artificial intelligence model has significantly improved it, helping to prevent users from broadcasting mass murder, pornography or violence content, to mention only a few examples. Despite the fact that the computer is faster and more capable than humans in detecting anomalies, it happens sometimes that some cases are undetected or poorly detected due to large volumes of data to be processed or poor sampling of data used to train model.

In this research report, we have used the artificial intelligence approach combined with Big-Data techniques in a case study of a real estate ad system to validate the content of ads. Note that the approach used could be generalized to other areas such as: retail sales, advertising sites in general, dating sites, etc. To do this, we have experimented with a broad-spectrum image recognition model, then build our own artificial intelligence model specifically dedicated to real estate and trained with the mass of data (i.e. photos, images of properties, etc.) that users will publish in the system.

## XII

The results of this experimentations show that, it is possible to detect the nature of the elements that users publish on Web platforms and at the same time validate in real time these elements.

Keywords: Cloud Computing, Artificial Intelligence, Computer Vision, Algorithm for Image Recognition, Machine Learning, Big Data

## TABLE DES MATIÈRES

		Page
1.1	Problématiques .....	3
1.2	Objectif.....	4
1.3	Approche proposée .....	4
1.4	Portée de l'expérimentation.....	5
1.5	Conclusion.....	5
2.1	Revue de l'état de l'art .....	7
2.1.1	Avant l'utilisation de l'Intelligence artificielle .....	7
2.1.2	Avec l'utilisation de l'IA .....	8
2.2	Infonuagique.....	8
2.2.1	Définition .....	8
2.2.2	Principaux fournisseurs infonuagiques.....	9
2.2.3	Avantages et Inconvénients de l'infonuagique .....	10
2.2.4	L'infonuagique dans le cadre de ce projet .....	11
2.3	Intelligence artificielle .....	11
2.3.1	Définition .....	11
2.3.2	Type d'intelligence artificielle .....	12
2.3.3	Principaux champs d'application .....	13
2.3.4	L'apprentissage machine.....	14
2.3.5	Traitement du langage naturel.....	15
2.3.6	La vision.....	15
2.3.7	La robotique .....	15
2.3.8	Les véhicules autonomes .....	16
2.3.9	L'Intelligence artificielle dans le cadre du projet.....	16
2.4	Web Services, Web API et Application Web.....	16
2.4.1	Introduction .....	16
2.4.2	Définition d'un Web Services .....	17
2.4.3	Types de Web Services.....	18
2.4.4	Services Web Restful.....	19
	2.4.4.1 Méthodes Restful .....	19
	2.4.4.2 Architecture Restful .....	21
2.4.5	Avantages et Inconvénients des services Web .....	21
2.4.6	Web Services dans le cadre du projet.....	22
2.5	Conclusion.....	22
3.1	Microsoft Azure Cognitive Services : Vision API .....	23
3.1.1	Présentation .....	23
3.1.2	Fonctionnalités .....	24
3.1.3	Utilisation.....	24
3.2	Notre modèle Custom (QRE Model) .....	27
3.2.1	Présentation .....	27
3.2.2	Fonctionnalités .....	27
3.2.3	Utilisation.....	28
3.3	Conclusion.....	33

4.1	Architecture haut niveau.....	35
4.2	Description des différentes couches.....	36
4.2.1	Clients .....	36
4.2.2	Applications Web (QRE – Azure WebSites) .....	38
4.2.3	Services Cloud (QRE – Azure Cloud Services) .....	39
4.2.4	Gestion de Stockage (QRE – Azure Cloud Storage) .....	41
4.2.5	Traitement de données et Analytique (QRE- Azure Data Processing & Analytics) .....	42
4.3	Interaction entre les modules .....	43
4.3.1	Diagramme de séquence d’ajout d’une annonce valide.....	43
4.3.2	Diagramme de séquence d’ajout d’une annonce non valide .....	44
4.3.3	Diagramme de séquence du processus de validation d’une image.....	45
4.4	Structure de l’architecture du programme dans Visual Studio .....	45
4.5	Structure de l’architecture du programme dans le nuage .....	50
4.6	Conclusion .....	51
5.1	Résultats - Azure Cognitive Services : Vision API .....	53
5.2	Résultats - Azure Cognitive Services : Custom Vision API.....	59
5.3	Conclusion .....	63

## LISTE DES TABLEAUX

Page

Tableau 2-1 : Comparatif des différents types de déploiement [14]**Error! Bookmark not defined.**





## LISTE DES FIGURES

	Page
Figure 1-1 : Statistiques sur les agences immobilières québécoises (OACIQ) [4] .....	3
Figure 2-1 : Facteurs principaux du Nuage [12] .....	9
Figure 2-2 : Les différents embranchements de l'Intelligence artificielle [21] .....	13
Figure 2-3 : Les 5 principaux champs d'application de l'intelligence artificielle [21] .....	14
Figure 2-4 : Diagramme d'architecture général d'un Web Services [27] .....	18
Figure 2-5 : Principales méthodes utilisées en REST [21] .....	20
Figure 3-1 : Quelques Ressources nécessaires à l'utilisation de Azure Custom Vision .....	29
Figure 3-3-2 : Les différentes catégories utilisées pour entraîner le modèle IA .....	29
Figure 3-3: Résultat d'évaluation du modèle IA de base (1 heure d'entraînement) .....	30
Figure 3-4 : Performance du modèle IA pour certains libellés .....	31
Figure 3-5 : Quelques images de la catégorie « Bathrooms » .....	32
Figure 3-6 : Quelques images de la catégorie « Bedrooms » .....	32
Figure 4-1 : Architecture Principale en couche .....	35
Figure 4-2 : Client Web adapté pour Mobile (1), Tablette (2) et Ordinateur (3) .....	37
Figure 4-3 : Couche Applicative Web .....	38
Figure 4-4 : Couche des principaux services cloud utilisés .....	40
Figure 4-6 : Services de stockage de données .....	41
Figure 4-7 : Couche de traitement des mégas donnés en lot .....	42
Figure 4-8 : Architecture en couche dans Visual Studio .....	46
Figure 4-9 : Azure Function Image Resizer .....	46
Figure 4-10 : Couche des Modèles POCO .....	47
Figure 4-11 : Repositories de l'application .....	47

## XVIII

Figure 4-12 : Couche de Search de l'application .....	48
Figure 4-13 : Couche des services de l'application.....	49
Figure 4-14 : Application Web MVC .....	50
Figure 4-15 : Liste de toutes les ressources instanciées pour le projet.....	51

## LISTE DES ABRÉVIATIONS, SIGLES ET ACRONYMES

QRE.....	Quebec-RealEstate
OACIQ.....	Organisme d’Autoréglementation du Courtage Immobilier du Québec
IA.....	Intelligence Artificielle
ML.....	Machine Learning
API.....	Application Program Interface
REST.....	Representational State Transfer
SDK.....	Software Development Kit
MVC.....	Model View Controller
HTTP.....	HyperText Transfer Protocol
SOAP.....	Simple Object Access
VS.....	Visual Studio
POCO.....	Plain Old CLR Object







## INTRODUCTION

Le 15 mars 2019, un individu diffusa en temps réel une vidéo d'une tuerie de masse sur le réseau social Facebook. Le 2 juillet 2019, Facebook écope d'une amende de 3 millions de dollars pour ne pas avoir supprimé à temps des commentaires haineux de son réseau [11]. Le réseau social Twitter doit faire retirer ou supprimer constamment des commentaires à caractères racistes, injurieux ou violents de son application. Ces exemples sont autant de situations qui démontrent l'importance de se prévaloir de valideur de contenu rapide et automatisé sur les réseaux sociaux et sur les sites Web « grand public » à fort trafic.

Avec la montée en flèche du nombre d'utilisateurs sur les plateformes internet, et le nombre grandissant d'objets connectés, il devient impératif de trouver des moyens de contrôler ce qui est téléchargé par les utilisateurs afin de ne pas enfreindre les lois, blesser ou heurter à la sensibilité de certains utilisateurs. Cette situation s'applique aussi bien aux sites d'annonces immobilières, de location et de vente de biens immobiliers. C'est dans cette optique que ce travail de recherche appliquée est effectué. Une étude de cas a été menée pour intégrer dans une application de gestion d'annonce immobilière ouverte au public, un pré valideur de contenu avant publication.

Le but de ce rapport de recherche est donc de décrire toutes les étapes de conception, de validation ainsi que le processus d'essai pour atteindre cet objectif. Ce rapport présente donc la problématique de cette étude, l'approche prise pour l'expérimentation et quelques recommandations et adaptations possibles de la même approche à d'autres domaines d'affaires. Le rapport est divisé en 5 grandes parties :

1. La mise en contexte qui vise à expliquer la problématique, l'approche retenue pour l'expérimentation, les essais effectués et leurs limites ;
2. La revue de la littérature, expliquant les technologies et les différents concepts techniques couramment utilisés et qui peuvent régler cette problématique ;
3. La conception du prototype, décrivant les différentes analyses et les raisons qui ont motivé le choix d'une approche de conception versus une autre ;

4. Présenter l'implémentation de la solution proposée dans une étude de cas dans un logiciel nommé : Québec Real Estate (QRE) ;
5. Présenter les résultats de l'expérimentation, les résultats obtenus et quelques travaux futurs potentiels et pistes d'amélioration.



# CHAPITRE 1

## Mise en contexte

Ce chapitre a pour but d'introduire le sujet de cette recherche. Il détaille les problématiques qui ont mené à la réalisation de cette étude de cas, l'objectif final recherché et l'approche principale proposée afin de tenter de répondre aux problématiques énumérées.

### 1.1 Problématiques

Investir dans l'immobilier est reconnu comme étant un investissement solide et garanti. Ainsi il y a une grande quantité d'agences immobilières et de sites de gestion et d'annonce de biens immobiliers à travers le monde. Au Québec seulement, voir figure 1.1, on ne dénombre pas moins de 984 agences immobilières enregistrées selon l'Organisme d'Autoréglementation du Courtage immobilier du Québec (OACIQ).

Courtier immobilier	11514
Courtier immobilier agréé	2699
Courtier hypothécaire	814
Agence immobilière	986
Agence hypothécaire	46
<b>Total</b>	<b>16059</b>

*\* Excluant les titulaires de permis dont le permis est suspendu.*

Figure 1-1 : Statistiques sur les agences immobilières québécoises (OACIQ) [4]

La plupart de ces agences possèdent leur propre site Web. Parmi ces sites Web, certains sont ouverts au public (c'est-à-dire que l'annonce en soie est directement publiée par les utilisateurs) tandis que d'autres ne le sont pas.

Pour ceux qui sont ouverts au public, une approche populaire est de permettre aux utilisateurs de poster leur annonce à tout moment. Cette annonce ne devient visible en ligne qu'après un certain temps, c'est-à-dire le temps qu'une personne physique valide le contenu de l'annonce et s'assure que ce contenu est un contenu approprié.

Cette approche manuelle de validation cause un délai de réactivité dans la publication des annonces et cause un problème, car la personne ou les personnes chargées de valider les annonces peuvent se retrouver débordées par le nombre d'annonces à valider advenant un nombre élevé de publications en un court laps de temps.

Comment alors pré valider les contenus de ces annonces sans avoir besoin d'une assistance humaine ? Comment aussi s'assurer que des personnes mal intentionnées ne téléchargent pas des images de nudités par exemple ou du contenu inapproprié sur une application censée recevoir que du contenu immobilier ?

Dans le cas où le site deviendrait une plateforme incontournable avec un nombre considérable d'utilisateurs et un nombre croissant d'images et de contenus textuels, comment s'assurer de toujours pouvoir bien valider les images sans grand délais ?

## **1.2 Objectif**

L'objectif de cette recherche appliquée est donc de pouvoir valider le contenu des images téléchargées, en temps réel, par les utilisateurs depuis la plateforme Web sans avoir besoin d'intervention humaine ou du moins utiliser l'intervention humaine qu'en dernier recours. Et pour les plateformes avec un trafic élevé et un gros volume d'image, trouver un moyen de les valider rapidement avec un degré de latence minimum.

## **1.3 Approche proposée**

L'approche proposée est donc d'utiliser la vision assistée par ordinateur, qui est en quelque sorte une intelligence artificielle capable de reconnaître des objets et aussi capable de

comprendre et d'analyser le contenu de message textuel pour valider les contenus. Dans le cas de système avec un trafic élevé, utiliser l'approche de traitement de mégadonnées pour le stockage et le traitement de ces données. En bref, conjuguer les atouts de l'intelligence artificielle aux techniques de traitement des mégadonnées pour pré valider les contenus des annonces.

#### **1.4 Portée de l'expérimentation**

L'expérimentation effectuée dans cette recherche appliquée portera sur une première phase de développement d'un prototype logiciel qui consiste à valider toutes les images téléchargées par les utilisateurs afin de s'assurer que ces images sont bien des images de bâtiments, d'immeubles de jardin et autres. Pour ce faire un algorithme d'intelligence artificielle à large spectre, qui vise à reconnaître et identifier différents types d'objets est expérimenté. L'objectif de cet essai est d'expérimenter avec les services infonuagiques de Microsoft Azure pour notre cas d'étude (c.-à-d. un système Web QRE).

À noter que les contenus textuels, vidéos et/ou audios, contenus dans ces annonces téléchargées ne seront pas validés lors de cette première phase de conception du prototype logiciel. De plus, concernant les aspects nécessitant des techniques de mégadonnées, l'architecture logicielle qui pourrait être mise en place sera conçue, mais ne sera pas implémentée durant cette première phase du projet.

#### **1.5 Conclusion**

Ce chapitre a mis en évidence la problématique qui a donné lieu à cette recherche appliquée, les différentes approches proposées pour atteindre l'objectif visé et la portée de l'expérimentation. Le prochain chapitre fait état de la littérature des notions, des termes, des techniques et des technologies utilisées dans le cadre de cette étude.



## **CHAPITRE 2**

### **Revue de la littérature**

Ce chapitre a pour but de présenter un survol des technologies utilisées dans le domaine de cette recherche. Il vise à introduire quelques notions théoriques ainsi que des techniques prometteuses qui permettent aux lecteurs une compréhension de base du domaine. La revue de l'état de l'art vise aussi à identifier quelques approches et façons de faire qui ont un grand potentiel concernant la validation de contenus publiés sur des plateformes Web.

#### **2.1 Revue de l'état de l'art**

##### **2.1.1 Avant l'utilisation de l'Intelligence artificielle**

Les problèmes liés à la validation des contenus, soumis par les utilisateurs sur les plateformes Web, ont toujours été un sujet préoccupant pour les gestionnaires et administrateurs de sites Web. Avant la montée en popularité de l'intelligence artificielle, l'une des façons de procéder pour cette validation des contenus, était d'utiliser les services d'une personne physique (c.-à-d. un humain) appelée « Modérateur » dont le rôle principal était de s'assurer que le contenu des données soumises par les utilisateurs s'aligne bien avec la vocation du site. Tant et aussi longtemps que le travail du modérateur n'a pas été fait, les données soumises par l'utilisateur restent en quelque sorte privées et ne sont pas visibles par les autres utilisateurs du site. Une autre approche populaire est d'utiliser l'intelligence collective [5], en quelque sorte, compter sur la contribution des utilisateurs du site Web pour rapporter auprès d'un modérateur les contenus qu'ils jugent inappropriés ou inadéquats. Enfin, une autre variante de solution publiée serait en effet de combiner les deux précédentes approches pour réduire le nombre de cas non détectés. Cependant avec la montée en flèche du nombre d'utilisateurs qui naviguent sur internet, toutes ces différentes approches sont maintenant caduques ou désuètes [6].

### **2.1.2 Avec l'utilisation de l'IA**

De nos jours, avec l'avènement de l'intelligence artificielle et son nombre élevé de libraires disponibles librement (c.-à-d. du domaine du logiciel libre) ainsi qu'avec l'avancée notable du succès de l'utilisation d'algorithmes de classification et de traitement du langage naturel, l'utilisation de techniques d'intelligence artificielle devient de plus en plus populaire pour la validation de contenu sur le Web. Lorsqu'il s'agit de données textuelles, par exemple pour empêcher les utilisateurs malicieux de répandre des rumeurs ou de la fausse information à grande échelle sur le Web, des modèles d'apprentissage machine basés sur la compréhension du langage naturel sont utilisés pour identifier les contenus de désinformation (« Fake News ») [7] [8]. Cette même technique est également utilisée dans les réseaux sociaux pour détecter les comptes des utilisateurs qui ont été piratés [9]. Lorsqu'il s'agit de contenu de type image, là aussi des avancées notables dans les algorithmes de type vision assistée par ordinateur ont eu lieu pour détecter par exemple des contenus de nudité de nature pornographique de ceux qui ne le sont pas [10]. Ce sont donc des approches similaires à ces dernières qui vont être mises de l'avant dans le cadre de cette étude de cas.

## **2.2 Infonuagique**

### **2.2.1 Définition**

L'infonuagique est un ensemble d'équipements informatiques, de raccordements réseau et de logiciels fournissant des services que des individus ou des collectivités peuvent exploiter depuis n'importe où dans le monde. Ce nuage technologique permet donc l'utilisation de services ou de serveurs informatiques distants par l'intermédiaire d'un réseau (c.-à-d. typiquement Internet) pour stocker des données et/ou les exploiter [12].

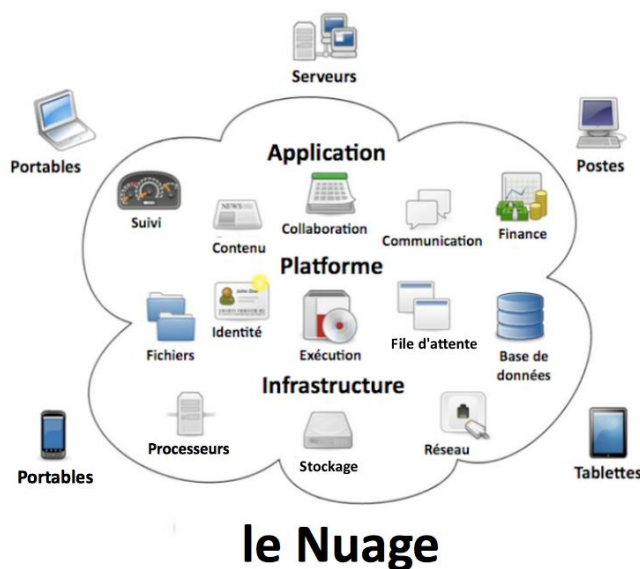


Figure 2-1 : Facteurs principaux du Nuage [12]

Les fournisseurs d'infonuagique publique offrent différents services d'intelligence artificielle faciles d'utilisation permettant aux développeurs de créer des applications intelligentes sans avoir des connaissances approfondies en matière d'intelligence artificielle ou de science des données. Notre but est de tirer profit d'un de ces services pour bâtir un modérateur intelligent de contenu de site Web et un modèle d'intelligence artificielle adapté spécifiquement pour notre cas de figure précis.

### 2.2.2 Principaux fournisseurs infonuagiques

Il existe plusieurs fournisseurs dans ce domaine. Les trois principaux (c.-à-d. les plus grands fournisseurs) sont : Amazon, Microsoft, Google.

La firme de recherche et de conseil « Gartner » a publié, en 2018, son rapport des entreprises qui sont actuellement leaders mondiaux de technologies infonuagiques en termes de rapidité d'exécution, de développement et de vision à long terme. Ces entreprises sont situées dans le quadrant en haut à droite de la figure 2-2.

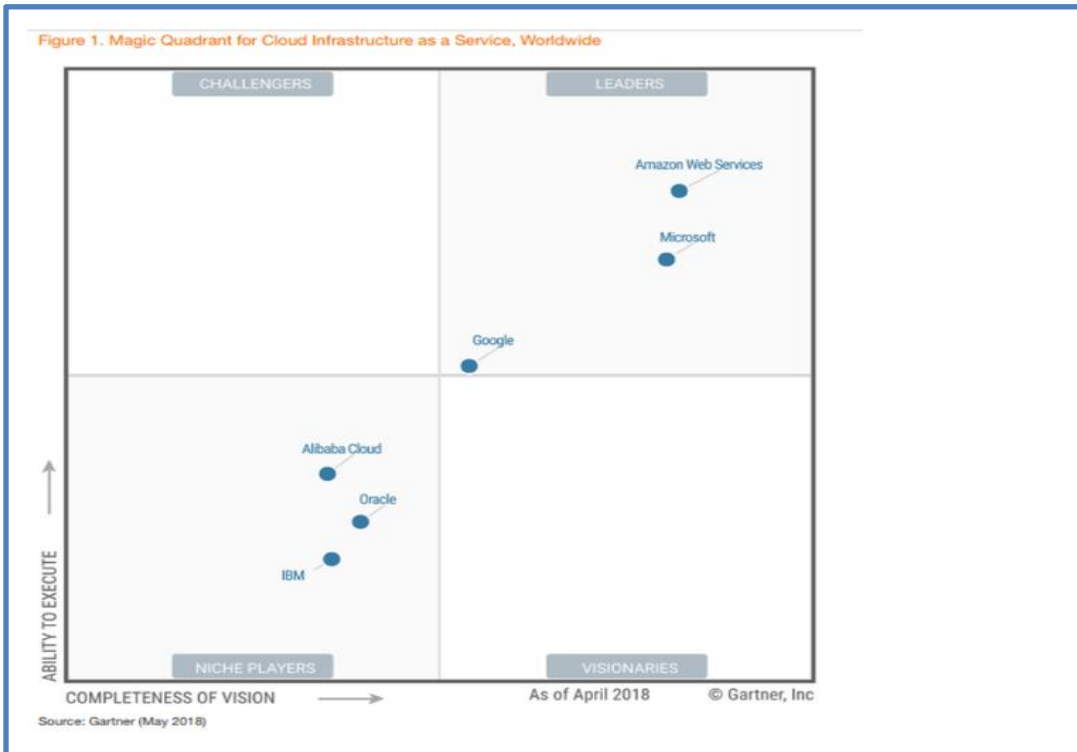


Figure 2-2 : Rapport annuel de 2018 des entreprises leadeures du Nuage selon Gartner [18]

### 2.2.3 Avantages et Inconvénients de l'infonuagique

Les avantages de l'utilisation de l'infonuagique augmentent continuellement avec l'ajout de nouveaux services, la recherche et développement continu et la création d'innovations qui repoussent sans cesse les limites de cette technologie. D'un point de vue « Architecture d'entreprise », les avantages de l'utilisation de services infonuagiques sont :

- Un bon rapport coût-efficacité ;
- La flexibilité dans les choix des infrastructures à utiliser ;
- La facilité de la mise à l'échelle (c.-à-d. l'élasticité) ;



- Le contrôle de la vitesse d'exécution ;
- La facilité d'intégration aux applications existantes ;
- L'assurance d'audits et de conformité des services ;
- La planification simple de la continuité des activités.

Bien que l'infonuagique offre de grandes opportunités, elle peut également présenter des défis pour les ingénieurs logiciels et les départements informatiques. Les inconvénients les plus cités sont que l'utilisation de l'infonuagique expose le client à des problèmes de sécurité et de performances incohérentes. La bonne nouvelle est que ces défis peuvent être surmontés avec une architecture infonuagique spécialement conçue pour supporter des charges de travail spécifique, et cela en tenant compte de bien choisir son fournisseur infonuagique (c.-à-d. fiable et efficace) [19].

#### **2.2.4 L'infonuagique dans le cadre de ce projet**

Dans le cadre de ce projet, deux différentes techniques d'intelligence artificielle visant la détection et la reconnaissance d'image ont été expérimentées. Ces techniques sont toutes hébergées dans le nuage. Afin de pouvoir bien les utiliser et en tirer un maximum de performance, une connaissance approfondie des services infonuagique, de leur architecture et de la façon de les utiliser et/ou de les instancier est initialement nécessaire ([voir l'annexe I](#)).

### **2.3 Intelligence artificielle**

#### **2.3.1 Définition**

L'intelligence artificielle (IA) est un ensemble des théories et des techniques mises en œuvre en vue de réaliser des machines capables de simuler l'intelligence humaine. Elle correspond donc à un ensemble de concepts et de technologies plus qu'à une discipline autonome [20]. Chez Microsoft, l'IA est perçue comme un moyen de permettre aux entreprises d'accomplir plus en utilisant des logiciels intelligents. Cette technologie maintenant disponible facilement, permet aux algorithmes d'apprendre un comportement à partir de grandes

quantités de données. Par exemple, une de ces techniques peut arriver à comprendre le langage et répondre à des questions. Une autre de ces techniques peut parvenir à voir et interpréter des images comme un humain.

### 2.3.2 Type d'intelligence artificielle

Les techniques d'IA peuvent être classées de plusieurs façons. Il existe principalement deux types de classification [21].

- Type 1
  1. **Techniques étroites** : ces techniques se concentrent sur des tâches précises et limitées : les techniques qui ne sont pas trop intelligentes pour faire leur propre travail peuvent être construites de manière à paraître intelligentes. Un exemple serait un jeu de poker où une machine bat un humain dans lequel toutes les règles et tous les mouvements sont introduits dans l'algorithme. Ici, chaque scénario possible doit être préalablement saisi manuellement ;
  - **Techniques puissantes** : les techniques capables de penser et d'exécuter des tâches seules, comme un être humain. Il n'existe pas d'exemples appropriés à ce sujet, mais certains chefs de file de l'industrie souhaitent ardemment se doter d'une IA forte, ce qui a permis des progrès rapides.
- Type2 (basé sur les fonctionnalités)
  1. **Machine réactive** : Est une des formes de base de l'IA. Il n'y a pas de mémoire conservée et ne peut pas utiliser les informations apprises pour ses actions futures. Exemple : un programme d'échecs IBM qui a battu Garry Kasparov dans les années 1990 ;
  2. **Mémoire limitée** : les systèmes d'intelligence artificielle peuvent utiliser les expériences passées pour éclairer les décisions futures. Certaines des fonctions de prise de décision dans les voitures autonomes ont été conçues de cette façon. Les

observations servent à informer des actions se déroulant dans un avenir pas si lointain, comme une voiture qui a changé de voie. Ces observations ne sont pas stockées de manière permanente, pas plus que le « Chatbot Siri de l'entreprise Apple » ;

3. **Théorie de l'esprit** : Ce type d'intelligence artificielle devrait être en mesure de comprendre les émotions, les croyances, les pensées, les attentes et les attentes des gens, et d'être en mesure d'interagir socialement bien que de nombreuses améliorations aient été apportées dans ce domaine ;
4. **Conscience de soi** : des techniques qui simulent la conscience, l'intelligence supérieure, la conscience de soi et la sensibilité (c.-à-d. en termes simples, un être humain complet). Bien sûr, ce type de bot n'existe pas non plus et, s'il est réalisé, il constituera l'un des jalons du domaine de l'Intelligence artificielle.

### 2.3.3 Principaux champs d'application

L'intelligence artificielle peut être utilisée de nombreuses façons, la figure 2-10 ci-dessous illustre quelques-unes d'entre elles [21] :

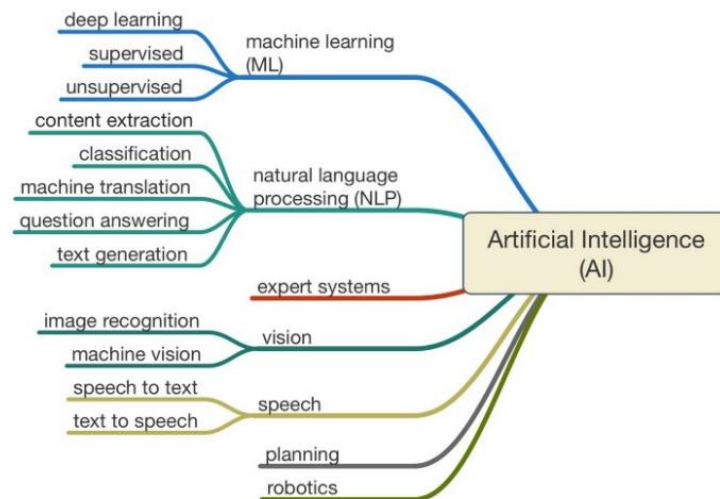


Figure 2-2 : Les différents embranchements de l'Intelligence artificielle [21]

De plus, parmi ces embranchements, on en dénombre 5 qui sont plus importants :

1. L'apprentissage Machine ;
2. Le traitement du langage naturel ;
3. La vision assistée par ordinateur ;
4. La robotique ;
5. Les véhicules autonomes.

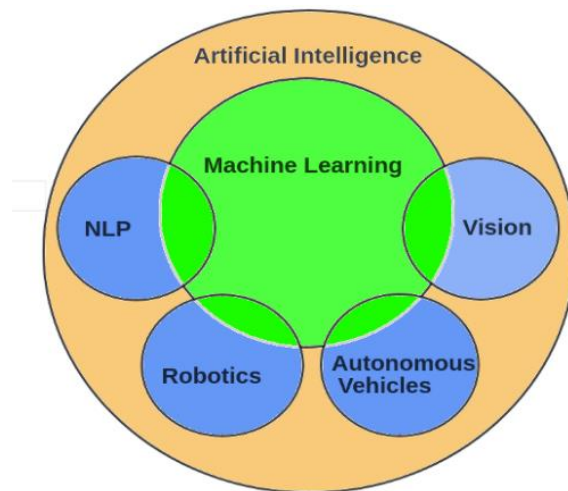


Figure 2-3 : Les 5 principaux champs d'application de l'intelligence artificielle [21]

Dans le cadre de cette étude de cas, l'accent est principalement axé sur des techniques de vision assistée par ordinateur et d'apprentissage machine.

### 2.3.4 L'apprentissage machine

L'apprentissage machine est un ensemble de méthodes où l'objectif (objectif) est défini et les étapes pour l'atteindre sont apprises par l'algorithme lui-même par formation (c.-à-d. l'acquisition d'expérience). Par exemple, pour identifier un objet simple tel qu'une pomme ou une orange. L'objectif n'est pas atteint en spécifiant explicitement les détails et en le

codant, mais comme en l'enseignant à un enfant en lui montrant de multiples images et en permettant ainsi à la machine de définir les étapes pour l'identifier comme une pomme ou une orange.

### **2.3.5 Traitement du langage naturel**

Le traitement du langage naturel est défini au sens large comme la manipulation automatique du langage naturel à l'aide d'un logiciel, tel que la parole et le texte. L'un des exemples les plus populaires est la détection d'un courriel indésirable.

### **2.3.6 La vision**

Les techniques de vision peuvent être vues comme un domaine de recherche qui permet aux machines de voir comme un humain. La vision artificielle saisit et analyse les informations visuelles à l'aide d'une caméra, d'une conversion analogique numérique et d'un traitement de signal numérique. Cela peut être comparé à la vue humaine, mais il n'est pas lié par la limitation humaine qui peut lui permettre de voir à travers les murs. L'apprentissage automatique permet d'obtenir les meilleurs résultats possibles, ce qui permet de dire que ces deux domaines est lié.

### **2.3.7 La robotique**

La robotique est un domaine de l'ingénierie centré sur la conception et la fabrication de robots. Les robots sont souvent utilisés pour effectuer des tâches difficiles à exécuter ou à réaliser de manière cohérente par l'homme. Les exemples incluent les chaînes de montage de voitures, dans les hôpitaux, les employés de bureau, les repas et la préparation des aliments dans les hôtels, les patrouilles dans les fermes et même en tant que policiers. Récemment, l'apprentissage machine a été utilisé pour obtenir de bons résultats dans la construction de robots à interaction sociale (c.-à-d. [Sophia](#)).

### **2.3.8 Les véhicules autonomes**

Ce domaine de l'IA a attiré beaucoup d'attention. La liste des véhicules autonomes comprend : les voitures, bus, camions, trains, navires, sous-marins et drones pilotés par pilote automatique.

### **2.3.9 L'Intelligence artificielle dans le cadre du projet**

Lors de cette recherche appliquée, une technique d'intelligence artificielle du type « Vision » sera expérimentée. Le modèle qui a été entraîné avec une grande quantité de données de tout genre devra être à mesure de reconnaître et de détecter tous types d'objets seront mis à contribution pour identifier le contenu des annonces (c.-à-d. des images) soumises par les utilisateurs.

## **2.4 Web Services, Web API et Application Web**

### **2.4.1 Introduction**

Les ingénieurs logiciels utilisent diverses plateformes de programmation pour concevoir et développer des applications Web. Par exemple, ces applications peuvent être développées à l'aide de technologies Java, d'autres en .Net, d'autres en Angular JS, Node.js, et bien d'autres. Le plus souvent, ces technologies hétérogènes nécessitent une sorte de communication entre elles. Comme elles sont construites avec différents langages de développement, il devient très difficile d'assurer une communication précise entre les applications. C'est là que l'utilisation des services Web devient importante. Les services Web constituent une plateforme commune qui permet à plusieurs applications basées sur différents langages de programmation de communiquer entre elles. Les lignes qui suivent détaillent un peu plus les concepts entourant les services Web, les différents éléments qui les constituent et les principes entourant son architecture.

## 2.4.2 Définition d'un Web Services

Un service Web est un support normalisé permettant de propager la communication entre les applications client et serveur sur internet. C'est en fait un module logiciel conçu pour effectuer un certain ensemble de tâches. Les services Web peuvent être recherchés sur le réseau internet et peuvent également être appelés en conséquence. Lorsqu'ils sont appelés, les services Web fournissent des fonctionnalités au client qui les invoque [21].

Dinarle Ortega, Elluz Uzcátegui et Maria M. Guevara [26] définissent un service Web comme étant, « la forme la plus utilisée pour la mise en œuvre d'une SOA (Service-Oriented Architecture). Un service Web est un système logiciel conçu pour prendre en charge l'interaction interopérable de machine à machine sur un réseau. Il possède une interface décrite dans un format pouvant être traité par une machine (en particulier WSDL). D'autres systèmes interagissent avec le service Web de la manière spécifiée par sa description à l'aide de messages SOAP, généralement acheminés via HTTP avec une sérialisation XML en conjonction avec d'autres normes relatives au Web. Les services Web exécutant des tâches utiles présentent souvent les propriétés suivantes :

1. Découvrable : L'une des exigences essentielles d'un service Web est de fournir le service à d'autres utilisateurs. Il doit donc être découvert et accessible par les consommateurs (utilisateurs humains ou autres services Web) ;
2. Communicable : il s'agit souvent d'une messagerie asynchrone, par opposition à la messagerie synchrone ;
3. Conversation : une conversation implique l'envoi et la réception de documents dans un contexte. Cela implique des interactions complexes entre les services Web et implique plusieurs étapes de communication liées les unes aux autres ;
4. Sécurisé et gérable : la sécurité, la gestion, la disponibilité et la tolérance aux pannes sont essentielles pour un service Web commercial. »

Le diagramme ci-dessous montre une vue synthétique du fonctionnement typique d'un service Web. Le client invoquerait une série d'appels de service Web via des demandes adressées à un serveur qui hébergerait le service Web réel.

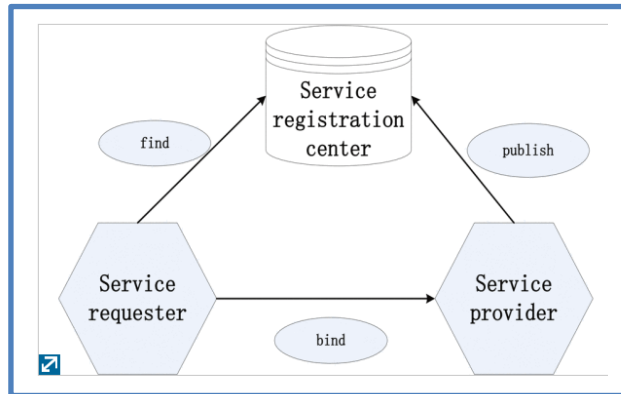


Figure 2-4 : Diagramme d'architecture général d'un Web Services [27]

### 2.4.3 Types de Web Services

Il existe principalement deux types de technologies derrière le terme services Web :

- Les services Web de type « Representational State Transfer (REST) » exposent entièrement ces fonctionnalités comme un ensemble de ressources identifiables par un URI et accessibles par la syntaxe et la sémantique du protocole HTTP. Les services Web de type REST sont donc basés sur l'architecture du Web et ses standards de base : HTTP et URI ;
- Les services Web de type « WS-\* » exposent ces mêmes fonctionnalités sous la forme de services exécutables à distance. Leurs spécifications reposent sur les standards SOAP et WSDL pour transformer les problématiques d'intégration héritées du monde des intergiciels en objectif d'interopérabilité. Les standards WS-\* sont souvent décriés comme risquant de générer une course à la performance technologique. Toutefois leur robustesse dans le milieu des services entre professionnels est reconnue, et ils restent largement utilisés. Aussi on préfère les faire évoluer.



Cela dit, l'emphase sera plus sur les services Web de type REST, étant donné que c'est ce type de service qui sera utilisé dans la réalisation de cette étude de cas.

#### **2.4.4 Services Web Restful**

REST permet de créer des services Web légers, faciles à gérer et évolutifs. Un service qui repose sur l'architecture REST est appelé service RESTful. Le protocole sous-jacent pour REST est HTTP, qui est le protocole Web de base. REST signifie « REPresentational State Transfer ». En fait, les services Web ont vraiment parcouru un long chemin depuis leur création. En 2002, le consortium Web avait publié la définition des services Web WSDL et SOAP. Cela constituait le standard de la mise en œuvre des services Web. En 2004, le consortium Web publia la définition de la norme RESTful. Et au cours de ces dernières années, cette norme est devenue très populaire. Et est utilisée par de nombreux sites Web populaires tels que Facebook et Twitter. REST est un moyen d'accéder aux ressources situées dans un environnement particulier. Par exemple, vous pourriez avoir un serveur hébergeant des documents importants, des images ou des vidéos. Ce sont tous des exemples de ressources. Si un client, par exemple un navigateur Web, a besoin de l'une de ces ressources, il doit envoyer une demande au serveur pour accéder à ces ressources. REST définit à présent un moyen d'accéder à ces ressources.

##### **2.4.4.1 Méthodes Restful**

Le diagramme ci-dessous montre principalement tous les méthodes ou verbes (POST, GET, PUT et DELETE) et un exemple de ce qu'ils signifieraient. Supposons qu'un service Web RESTful soit défini à l'emplacement. <http://demo.monapi.com/employee>. Lorsque le client adresse une demande à ce service Web, il peut spécifier l'un des verbes HTTP normaux de GET, POST, DELETE et PUT. Vous trouverez ci-dessous ce qui se produirait si les verbes respectifs étaient envoyés par le client.

1. POST - Ceci serait utilisé pour créer un nouvel employé à l'aide du service Web RESTful ;

2. GET - Ceci serait utilisé pour obtenir une liste de tous les employés utilisant le service Web RESTful ;
3. PUT - Ceci sera utilisé pour mettre à jour tous les employés utilisant le service Web RESTful ;
4. DELETE - Ceci sera utilisé pour supprimer tous les employés utilisant le service Web RESTful.

Que faire pour cibler une ressource en particulier ? Supposons qu'il existe un enregistrement d'employé avec le numéro d'employé 1. Les actions suivantes auraient les significations respectives :

1. POST - Ceci ne serait pas applicable, car ce sont les données de l'employé 1 déjà créé qui sont récupérées ;
2. GET - Ceci sera utilisé pour obtenir les détails de l'employé avec l'employé n ° 1 en utilisant le service Web RESTful ;
3. PUT - Ceci sera utilisé pour mettre à jour les détails de l'employé avec l'employé n ° 1 en utilisant le service Web RESTful ;
4. DELETE - ceci est utilisé pour supprimer les détails de l'employé dont le numéro d'employé est 1.

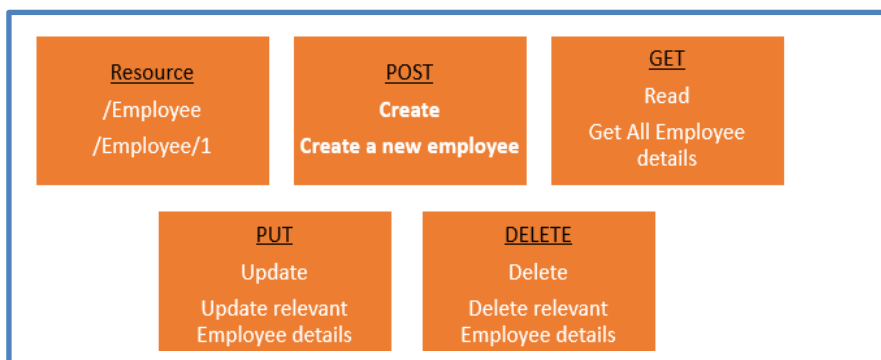


Figure 2-5 : Principales méthodes utilisées en REST [21]

#### **2.4.4.2 Architecture Restful**

Une application ou une architecture considérée comme étant de type RESTful ou REST présente les caractéristiques suivantes :

1. L'état et la fonctionnalité sont divisés en ressources distribuées - cela signifie que chaque ressource doit être accessible via les commandes HTTP normales de GET, POST, PUT ou DELETE. Ainsi, si quelqu'un souhaite obtenir un fichier d'un serveur, il doit pouvoir émettre la demande GET et obtenir le fichier. S'ils veulent mettre un fichier sur le serveur, ils doivent pouvoir émettre la demande POST ou PUT. Enfin, s'ils souhaitent supprimer un fichier du serveur, ils émettent la demande DELETE ;
2. L'architecture est client / serveur, sans état « Stateless », en couches et supporte la mise en cache ou « caching ».

L'architecture client-serveur est l'architecture typique où le serveur peut être le serveur Web hébergeant l'application et le client peut être aussi simple que le navigateur Web. Finalement, la notion de « Stateless » signifie que l'état de l'application n'est pas maintenu dans REST. Par exemple, si vous supprimez une ressource d'un serveur à l'aide de la commande DELETE, vous ne pouvez pas vous attendre à ce que les informations de suppression soient transmises à la demande suivante. Afin de vous assurer que la ressource est supprimée, vous devez émettre la demande GET.

#### **2.4.5 Avantages et Inconvénients des services Web**

L'utilisation de services Web comporte des avantages [22] :

- Les services Web fournissent l'interopérabilité entre divers logiciels fonctionnant sur diverses plateformes ;
- Les services Web utilisent des standards et protocoles ouverts ;
- Les protocoles et les formats de données sont au format texte dans la mesure du possible, facilitant ainsi la compréhension du fonctionnement global des échanges ;

- Les outils de développement, s'appuyant sur ces standards, permettent la création automatique de programmes utilisant les services Web existants.

Ils comportent aussi des inconvénients [22] :

- Les normes de services Web dans certains domaines sont actuellement récentes ;
- Les services Web souffrent de performances faibles comparées à d'autres approches de l'informatique répartie telles que le RMI, CORBA, ou DCOM.

#### **2.4.6 Web Services dans le cadre du projet**

Dans le cadre de ce projet, l'intelligence artificielle de détection et de reconnaissance d'image est expérimentée. Les techniques d'intelligence artificielle étant toutes hébergées dans le nuage, elles ne sont invocables que via des services Web de type REST et des Web API. La connaissance des principes sous-jacents de cette technologie est un incontournable au succès de notre expérimentation.

### **2.5 Conclusion**

Ce chapitre passe en revue les principales notions du domaine de l'infonuagique, de l'intelligence artificielle, du fonctionnement des services Web et applications Web en général, il met aussi en lumière quelques façons actuelles de faire pour approcher la problématique décrite au chapitre 1. Le prochain chapitre présente les différents algorithmes d'IA expérimentés dans cette recherche appliquée.

## CHAPITRE 3

### Analyse des algorithmes

Ce chapitre a pour but de décrire le service infonuagique de vision offert par Microsoft Azure. Ce service de vision assistée par ordinateur est expérimenté dans le cadre de ce projet. Il vise à décrire l'API et l'algorithme d'IA permettant de traiter les images et de les interpréter.

#### 3.1 Microsoft Azure Cognitive Services : Vision API

##### 3.1.1 Présentation

Azure Cognitive Services sont des API, des kits SDK et des services disponibles pour aider les développeurs à créer des applications intelligentes sans avoir des connaissances approfondies en matière d'intelligence artificielle ou de science des données. Le catalogue de services dans Azure Cognitive Services peut être divisé en cinq principaux piliers : Vision API, Speech API, Language API, Search API et Decision API. [24]. L'API Vision est celle explorée dans cette recherche.

Azure Vision API offre aux développeurs un accès à des algorithmes avancés qui traitent les images et renvoient des informations. Pour analyser une image, vous pouvez télécharger une image ou spécifier une URL de l'image. Les algorithmes de traitement des images peuvent analyser le contenu de différentes manières, en fonction des caractéristiques visuelles qui vous intéressent. Par exemple, Computer Vision API peut déterminer si une image contient un contenu adulte ou raciste, ou détecter tous les visages humains d'une image.

### 3.1.2 Fonctionnalités

De tous les algorithmes analysés, Azure Vision API est celle qui semble la plus complète, offrant essentiellement 5 différents services :

- **Computer Vision** : Le service Computer Vision donne accès à des algorithmes avancés pour le traitement des images et le retour des informations ;
- **Custom Vision Services** : Le service Custom Vision permet de créer des classificateurs d'image personnalisés ;
- **Face API** : Face API fournit un accès à des algorithmes de visage avancés, permettant la détection et la reconnaissance des attributs de visage ;
- **Form Recognizer** : Form Recognizer identifie et extrait les paires clé-valeur et les données de table des documents de formulaire et génère ensuite des données structurées, y compris les relations dans le fichier d'origine ;
- **Ink Recognizer** : Ink Recognizer permet de reconnaître et d'analyser les données de trait d'encre numérique, les formes et le contenu manuscrit, et de générer une structure de document avec toutes les entités reconnues ;
- **Video Indexer** : Video Indexer permet d'extraire des informations de votre vidéo.

### 3.1.3 Utilisation

L'intégration de l'intelligence artificielle dans l'application peut se faire en utilisant un SDK natif ou en appelant directement l'API REST. Des prérequis sont à respecter par rapport à l'URL de l'image, le format, la taille et la dimension de l'image. Les fonctionnalités d'analyse de l'image sont présentées en deux options :

- **Option 1** : Analyse délimitée - Analyser uniquement un modèle donné ;
- **Option 2** : Analyse élargie - Analyser avec la taxonomie des 86 catégories pour fournir des détails supplémentaires.

Après avoir instancié le service dans le nuage, une clé d'abonnement est automatiquement fournie et qui devra être incluse dans toutes les requêtes faites au service. Le format de l'adresse de l'API REST à invoquer est la suivante :

**`https://Location.api.cognitive.microsoft.com/vision/v2.0/analyze?visualFeatures=Description,Tags&subscription-key=<Your subscription key>`**

où:

- Location : représente l'endroit géographique où le service a été instancié dans le nuage
- Your subscription key : la clé d'abonnement fournie par le fournisseur du nuage.

Exemple :

POST

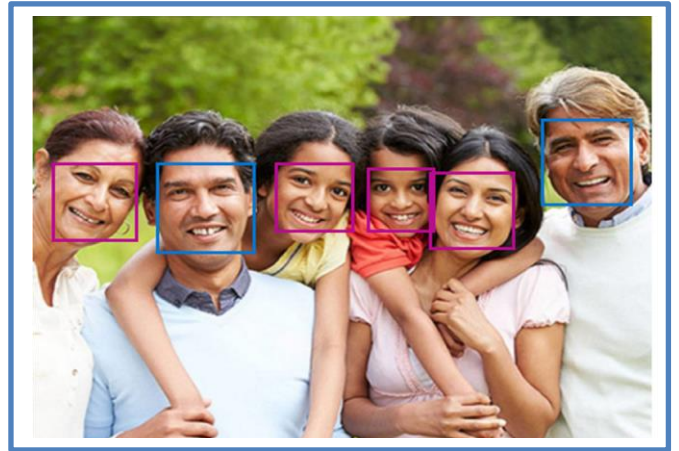
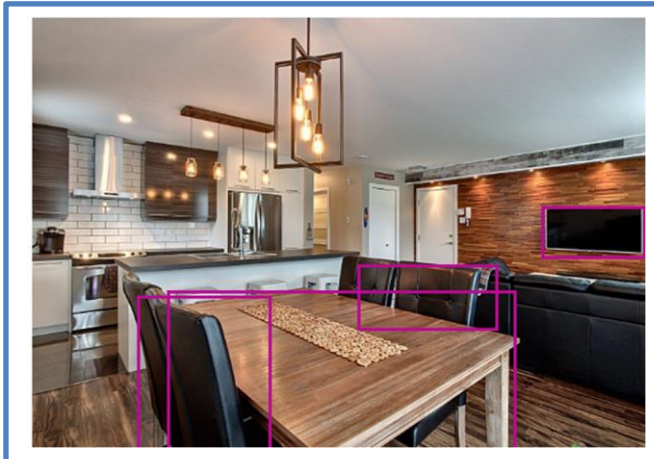
`https://westus.api.cognitive.microsoft.com/vision/v2.0/describe&subscription-key=8AXB52363ARTY85963214ASSS_9`

Le résultat de l'analyse de l'image est un document JSON avec plusieurs caractéristiques dont les principales sont :

- Une liste d'objets : un tableau JSON qui liste tous les objets identifiés dans l'image avec un degré de confiance (probabilité) ainsi que les coordonnées exactes de l'emplacement de l'objet ;
- Une liste de libellés : un tableau JSON qui liste les libellés des objets détectés dans l'image ;
- Description : en fonction des objets détectés et des libellés, l'algorithme génère une description complète du contenu de l'image comme le ferait un humain avec un degré de confiance ;
- Catégories : l'algorithme associe une catégorie à l'image parmi un catalogue de catégorie avec un score ;
- Adult Content : indique si l'image contient du contenu pour adulte seulement avec un degré de confiance ;
- Faces : un tableau JSON de toutes les faces d'humain détectées dans l'image, leur emplacement, le sexe et l'âge de la personne.

L’algorithme est également capable de détecter les écritures manuscrites dans les images.

Ci-dessous deux exemples d’images traitées par l’algorithme avec leur résultat respectif :



FEATURE NAME:	VALUE
Objects	[ { "rectangle": { "x": 1327, "y": 500, "w": 270, "h": 144 }, "object": "television", "parent": { "object": "display", "confidence": 0.581 }, "confidence": 0.575 }, { "rectangle": { "x": 848, "y": 659, "w": 365, "h": 183 }, "object": "chair", "parent": { "object": "seating", "confidence": 0.555 }, "confidence": 0.5 }, { "rectangle": { "x": 276, "y": 740, "w": 360, "h": 446 }, "object": "chair", "parent": { "object": "seating", "confidence": 0.794 }, "confidence": 0.792 }, { "rectangle": { "x": 346, "y": 728, "w": 919, "h": 456 }, "object": "dining table", "parent": { "object": "table", "confidence": 0.713 }, "confidence": 0.708 } ]
Tags	[ { "name": "indoor", "confidence": 0.9879653 }, { "name": "table", "confidence": 0.9777049 }, { "name": "chair", "confidence": 0.948350549 }, { "name": "floor", "confidence": 0.933600843 }, { "name": "coffee table", "confidence": 0.922964 }, { "name": "interior", "confidence": 0.8885645 }, { "name": "hotel", "confidence": 0.858062863 } ]
Description	{ "tags": [ "indoor", "table", "room", "chair", "window", "kitchen", "living", "wooden", "counter", "sitting", "area", "furniture", "large", "modern", "dining", "bedroom", "bench", "desk", "island", "computer", "bed", "stove", "white" ], "captions": [ { "text": "a modern kitchen with an island in the middle of a room", "confidence": 0.858062863 } ] }
Adult score	0.000279246044
Racy	false
Racy score	0.000432123867
Categories	[ { "name": "indoor_room", "score": 0.55078125 } ]
Faces	[ ]

FEATURE NAME:	VALUE
Objects	[ { "rectangle": { "x": 203, "y": 87, "w": 121, "h": 375 }, "object": "person", "confidence": 0.551 }, { "rectangle": { "x": 460, "y": 58, "w": 140, "h": 402 }, "object": "person", "confidence": 0.859 }, { "rectangle": { "x": 4, "y": 100, "w": 182, "h": 355 }, "object": "person", "confidence": 0.835 }, { "rectangle": { "x": 53, "y": 102, "w": 266, "h": 358 }, "object": "person", "confidence": 0.839 }, { "rectangle": { "x": 878, "y": 294, "w": 121, "h": 375 }, "object": "person", "confidence": 0.551 } ]
Tags	[ { "name": "smile", "confidence": 0.997360468 }, { "name": "outdoor", "confidence": 0.997002363 }, { "name": "person", "confidence": 0.994868755 }, { "name": "human face", "confidence": 0.9812523 }, { "name": "posing", "confidence": 0.9580809 }, { "name": "clothing", "confidence": 0.9436915 }, { "name": "woman", "confidence": 0.855067849 }, { "name": "girl", "confidence": 0.855067849 } ]
Description	{ "tags": [ "outdoor", "person", "posing", "photo", "grass", "group", "standing", "man", "people", "woman", "young", "holding", "dress", "white" ], "captions": [ { "text": "Bern Collaco et al. posing for a photo", "confidence": 0.990609467 } ] }
Adult content	false
Adult score	0.00137113838
Racy	false
Racy score	0.00465372251
Categories	[ { "name": "people_group", "score": 0.97265625 } ]
Faces	[ { "age": 43, "gender": "Male", "faceRectangle": { "top": 159, "left": 118, "width": 95, "height": 95 } }, { "age": 54, "gender": "Male", "faceRectangle": { "top": 111, "left": 490, "width": 91, "height": 91 } }, { "age": 56, "gender": "Female", "faceRectangle": { "top": 155, "left": 878, "width": 121, "height": 375 } } ]



## **3.2 Notre modèle Custom (QRE Model)**

### **3.2.1 Présentation**

Le modèle IA personnalisé QRE spécifiquement dédié aux images du domaine de l'immobilier est bâti en utilisant Azure Custom Vision API, qui est un service cognitif qui permet de créer, déployer et améliorer ses propres classificateurs d'images. Un classificateur d'images est un service d'IA qui applique des étiquettes (qui représentent des classes) aux images, en fonction de leurs caractéristiques visuelles. Contrairement au service Computer Vision, Azure Custom Vision API permet de déterminer les étiquettes à appliquer. [28]

L'avantage avec Azure Custom Vision API est qu'il permet de faire des distinctions plus fines en utilisant les détails des images du domaine personnalisé, aussi il permet d'éviter de confondre le modèle IA avec des images similaires, mais non pertinentes.

### **3.2.2 Fonctionnalités**

Les fonctionnalités offertes par Azure Custom Vision API peuvent être divisées en deux fonctionnalités principales :

- La classification d'image
- Et la détection d'objet

La classification d'image applique une ou plusieurs étiquettes à une image. La détection d'objets est similaire, mais elle renvoie également les coordonnées dans l'image où les étiquettes appliquées peuvent être trouvées.

L'algorithme de classification d'image cherche à affecter une image d'entrée à l'une d'un ensemble fixe de catégories ou d'étiquette. Chaque affectation reçoit une probabilité basée sur la confiance du modèle dans sa prédiction. À première vue, cela semble être un problème relativement simple à résoudre. Néanmoins, il existe des défis considérables pour un algorithme de vision par ordinateur, car il faut considérer la variation de l'image d'entrée avec laquelle l'image est soumise, les changements de point de vue et d'échelle, la déformation des images, l'obscurcissement partiel du sujet à analyser et toute une gamme de conditions

d'éclairage et de distractions de fond différentes. Un bon modèle de vision par ordinateur doit être capable de faire des prédictions correctes entre des images qui correspondent aux différentes catégories sans être distrait par une variation pertinente comme celles qui viennent d'être décrites. Plusieurs variantes d'algorithme de type « k-nearest neighbors algorithm » sont testées avec les données d'entraînement et celui qui à le meilleur résultat en termes de « Precision », « Recall » et « AP » est retenu pour bâtir le modèle final. Ces différents termes sont décrits à la section 3.2.3 qui suit.

### **3.2.3 Utilisation**

Azure Custom Vision API est disponible sous la forme d'un ensemble de SDK natifs ainsi que via une interface Web sur la page d'accueil de Azure Custom Vision Service. Il est possible de créer, entraîner, évaluer et tester un modèle IA en utilisant l'une des méthodes ou les deux ensembles.

Dans le cadre de ce projet, compte tenu du nombre important de catégories (libellé) relatives au domaine de l'immobilier et du volume de données d'entraînement, les deux techniques ont dû être utilisées, l'interface Web pour prébâtir le modèle IA de base et ensuite avec les données accumulées dans la base de données et les différents comptes de stockage, entraîner, évaluer et tester le modèle via le SDK .NET.

L'architecture de l'application ainsi que les différents diagrammes de séquences décrits au chapitre 4 illustrent comment tous les modules sont orchestrés ensemble pour arriver au but visé.

Tout comme la plupart des services cloud, pour pouvoir créer, entraîner, évaluer et tester un modèle IA personnalisé via le SDK .NET, Azure Custom Vision nécessite de fournir une clé et une série de ressource pour pouvoir interagir avec les services. En outre on peut citer :

- Une Clé de ressource
- Un Endpoint (Url)
- Un Project Id

### Project Settings

**General**

**Project Name\***

**Project Id**

**Description**

**Usage:** ⓘ

527 training images uploaded; 99473 remain

24 tags created; 476 remain

**Resources:**

qrecustomvision  
 Subscription: Azure pour les étudiants  
 Resource Group: qregrp  
 Resource Kind: Custom Vision Training

**Key:**

**Endpoint:**

**Resource Id:**

**Pricing Tier: 50**

Change Pricing Tier

1 projects created; 99 remain

Figure 3-1 : Quelques Ressources nécessaires à l'utilisation de Azure Custom Vision

Les catégories retenues dans le cadre de cette expérience (étiquette ou libellé) sont les suivantes :

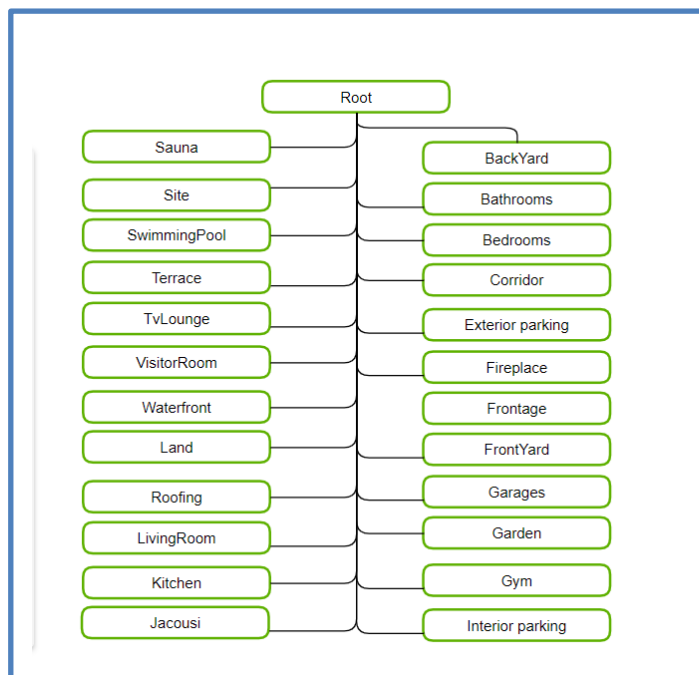


Figure 3-3-2 : Les différentes catégories utilisées pour entrainer le modèle IA

Ci-dessous, les résultats obtenus après entraînement du modèle avec les données de base pendant une heure environ :

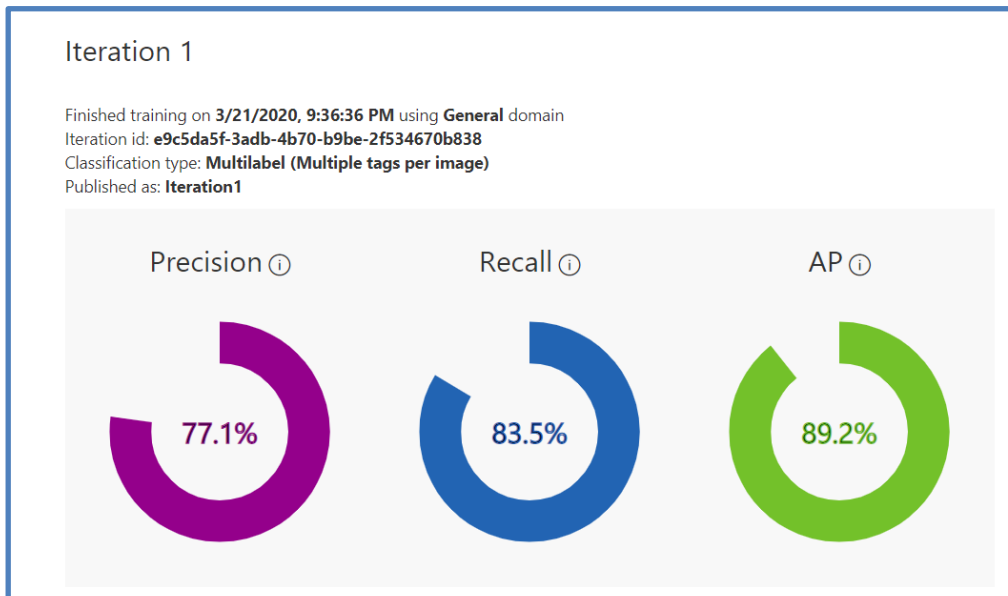


Figure 3-3: Résultat d'évaluation du modèle IA de base (1 heure d'entraînement)

Dans l'évaluation d'un modèle IA les termes suivants décrivent :

- Precision : Si un libellé est prédit par le modèle IA, quelle est la probabilité que cela soit exact ?
- Recall : Parmi les libellés à prédire correctement, quel pourcentage le modèle a-t-il correctement trouvé ?
- AP : Une mesure des performances du modèle, récapitule la « Precision » et le « Recall » à différents seuils.

Ci-dessous, quelques performances du modèle IA par libellé :











Tag	Precision <sup>^</sup>	Recall	A.P.	Image count
VisitorRoom	100.0%	80.0%	82.9%	23 
Sauna	100.0%	100.0%	100.0%	20 
Roofing	100.0%	100.0%	100.0%	20 
Land	100.0%	100.0%	100.0%	19 
InteriorParking	100.0%	75.0%	100.0%	19 
Garages	100.0%	100.0%	100.0%	26 
Fireplace	100.0%	100.0%	100.0%	20 
ExteriorParking	100.0%	25.0%	39.0%	18 
Corridor	100.0%	100.0%	100.0%	25 
Bedrooms	85.7%	100.0%	100.0%	30 

Figure 3-4 : Performance du modèle IA pour certains libellés

Ci-dessous, quelques images utilisées pour bâtir le modèle par catégorie (libellé) :

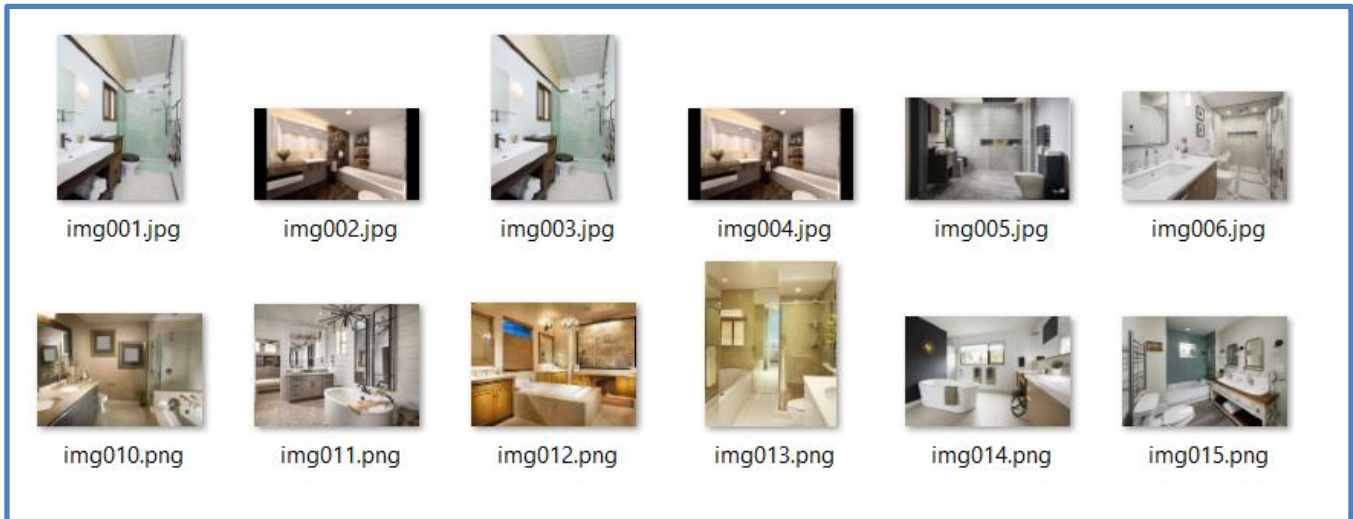


Figure 3-5 : Quelques images de la catégorie « Bathrooms »

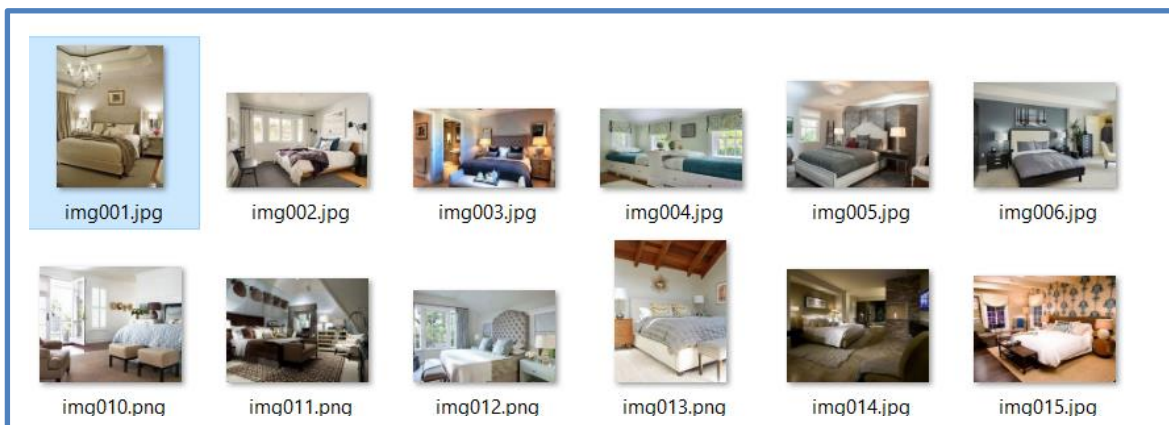


Figure 3-6 : Quelques images de la catégorie « Bedrooms »

À noter qu'une centaine d'image mise de côté par catégorie a été utilisée pour évaluer le modèle IA.

### **3.3 Conclusion**

Ce chapitre présentait les deux principaux algorithmes d'IA utilisés, il a été aussi présenté en détails les différentes fonctionnalités offertes par chacun d'eux et comment les utiliser. Pour bâtir un modèle IA plus solide et robuste, capable de détecter et identifier correctement un maximum d'objet, il est souhaitable de faire une combinaison des deux approches : la « Vision API » à large spectre et la « Custom Vision API » plus spécifique. Le prochain chapitre présente l'architecture logicielle de toute la solution, les différents modules et l'interaction entre ces modules.





## CHAPITRE 4

### Architecture et implémentation de la solution

Ce chapitre a pour but de présenter l'architecture logicielle du prototype expérimental qui a été expérimenté. Il décrit à travers des diagrammes et des schémas, le contenu des différents services et modules mis en place ainsi que les interactions qui ont lieu entre eux.

#### 4.1 Architecture haut niveau

L'architecture proposée pour ce premier prototype logiciel est hébergée en grande partie sur la plateforme Microsoft Azure. La vue d'ensemble de l'architecture logicielle proposée est décrite à la figure 4.1.

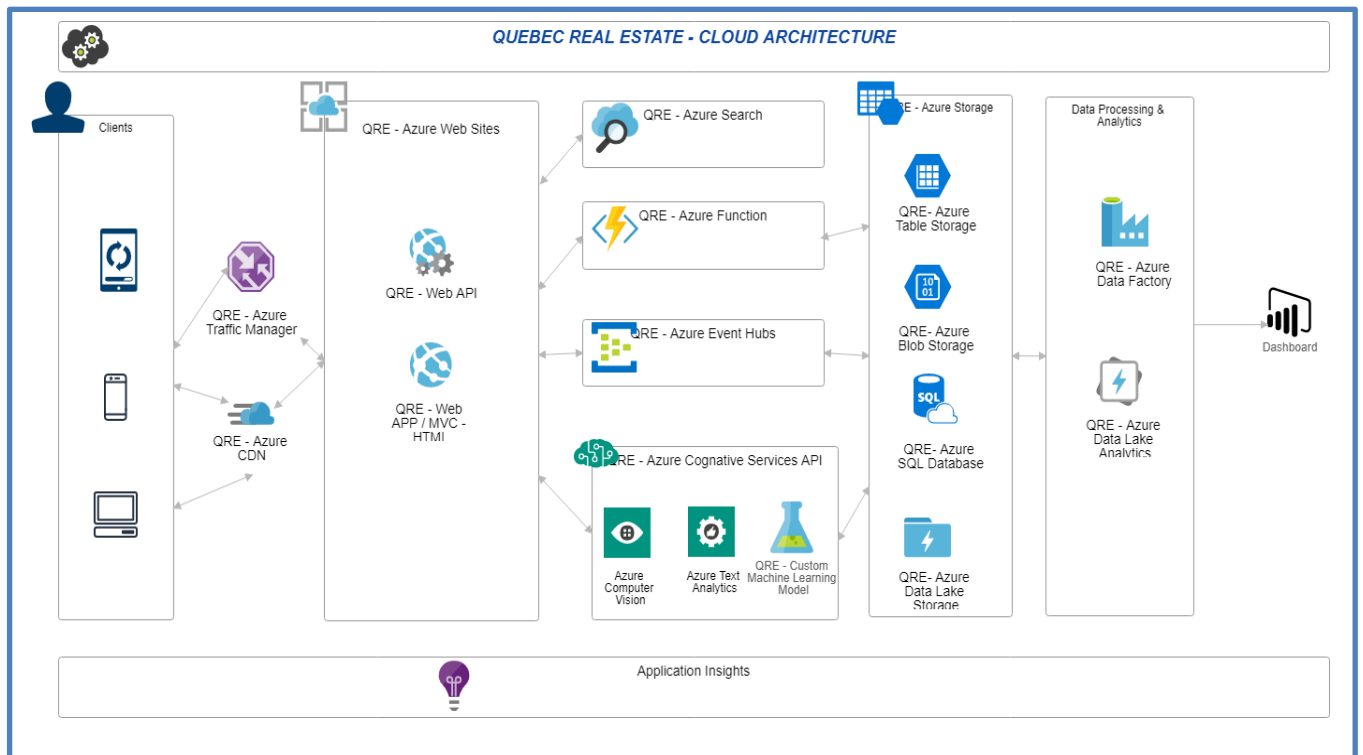


Figure 4-1 : Architecture Principale en couche

Cinq (5) principales couches applicatives composent l'architecture de la solution :

- La couche client ou Interface utilisateur :
- La couche pour le Serveur applicatif / Server Web :
- La couche de Services offerts par la structure infonuagique de Microsoft Azure :
- La couche de stockage des données :
- La couche d'analyse et de traitement en lot.

La plateforme Azure de Microsoft s'intègre évidemment bien avec les technologies de Microsoft, c'est la raison pour laquelle les cadres et langages de programmation utilisés au niveau des différentes couches sont ceux de Microsoft. Quelques-uns parmi eux sont :

1. ASP.NET / MVC, pour tout ce qui est application Web et serveur applicatif ;
2. ASP.NET/ Web API, pour tout ce qui est service Web et API ;
3. C# (Csharp), comme langage principal de programmation ;
4. Entity Framework, pour tout ce qui est accès à la base de données ou ORM (Object-Relational Mapping) ;
5. Azure SQL Database, qui est une base de données Microsoft SQL Server dans le nuage.

## **4.2 Description des différentes couches**

### **4.2.1 Clients**

La couche « Client » est essentiellement Web, faite en HTML/CSS standard, dynamisé avec du JavaScript. L'interface utilisateur est conçue de sorte à être « Responsive » (adaptée pour tous types d'écrans). L'application peut donc être consultée via un ordinateur <sup>3</sup>, une tablette <sup>2</sup> ou un téléphone intelligent <sup>1</sup>.

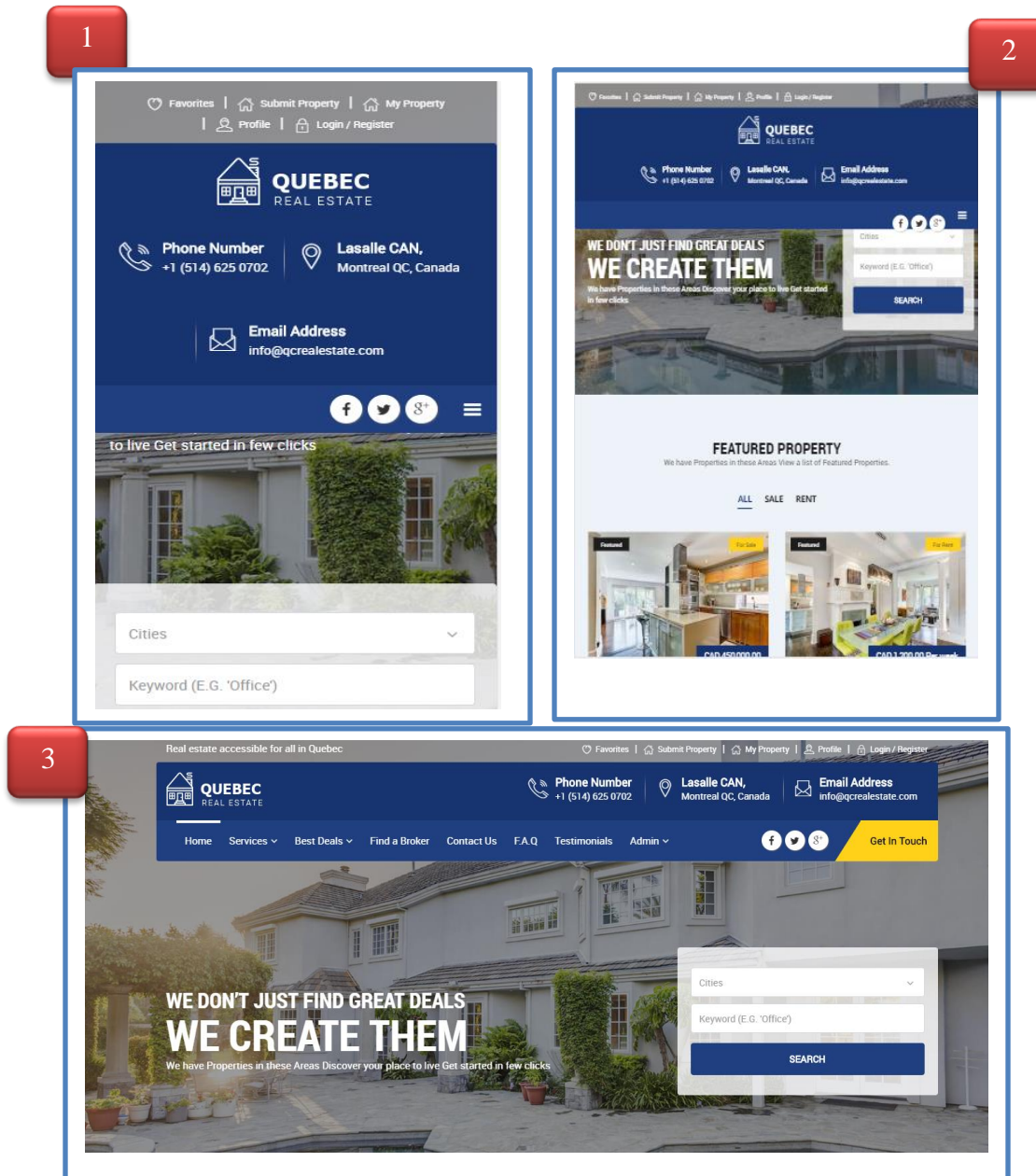


Figure 4-2 : Client Web adapté pour Mobile (1), Tablette (2) et Ordinateur (3)

#### 4.2.2 Applications Web (QRE – Azure WebSites)

Cette couche représente la couche Web Applicative du système, conçue avec le Framework Web ASP.NET MVC de Microsoft, elle est chargée de traiter toutes les requêtes Web. Deux applications Web sont déployées dans cette couche :

1. Une application Web MVC standard avec tous les concepts de Model, Vue, Contrôleur chargé d'opérer toute la logique nécessaire d'ajout et de modification des annonces et de générer le rendu HTML/CSS et JavaScript retourné au client ;
2. Une application Web spécialement exposée comme une API (un service Web REST), exposant tous les services du Système à des composantes ou applications externes. Cette API devient utile lorsque l'on veut offrir les services de publication d'annonce via une application mobile native ou hybride par exemple.

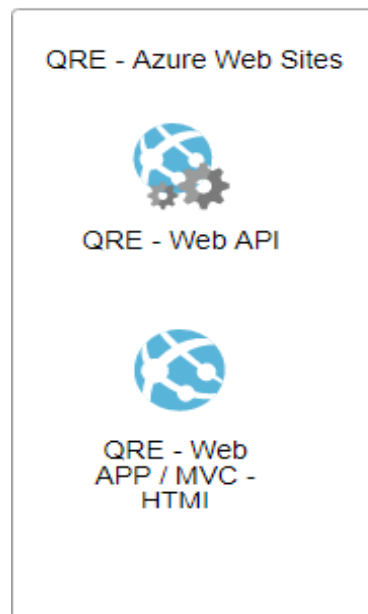


Figure 4-3 : Couche Applicative Web

### 4.2.3 Services Cloud (QRE – Azure Cloud Services)

Cette couche regroupe tous les services cloud offerts par la plateforme d'azure, parmi ceux-ci, il y a par exemple :

- Azure Search : est le seul service de recherche en nuage doté de fonctionnalités d'intelligence artificielle (AI) intégrée qui enrichit tous les types d'informations pour identifier et explorer facilement les contenus pertinents à grande échelle. Il utilise la même pile de langage naturel intégrée à Microsoft utilisée par Bing et Office depuis plus de dix ans, ainsi que des API prédéfinies pour la vision, le langage et la parole. Il est utilisé dans le cadre de ce projet pour l'indexation et la recherche de propriétés ;
- Azure Function : est un service de calcul sans serveur qui vous permet d'exécuter du code à la demande sans avoir à provisionner ou gérer explicitement l'infrastructure. Il est utilisé pour redimensionner en temps réels les images soumises par les utilisateurs ;
- Azure Cognitive API : sont des API, des kits SDK et des services disponibles pour aider les développeurs à créer des applications intelligentes sans avoir des compétences approfondies en intelligence artificielle. Azure Cognitive Services permet aux développeurs d'ajouter facilement des fonctionnalités cognitives à leurs applications. Ces services de vision assistée, de détection, de reconnaissance d'objet et de création de modèles d'intelligence artificielle « Custom » sont utilisés dans le cadre de ce projet pour filtrer les images soumises par les utilisateurs.

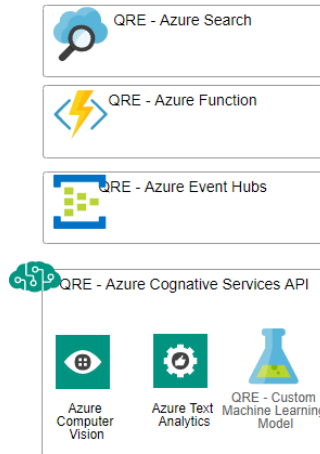


Figure 4-4 : Couche des principaux services cloud utilisés

Il est important de noter que le fait d’avoir une architecture modulaire facilite l’intégration et les tests de nouveaux modules. Ainsi pour voir intégrer et tester les autres services d’intelligence artificielle offerts par Google et Amazon, il a fallu adapter seulement la couche des services.

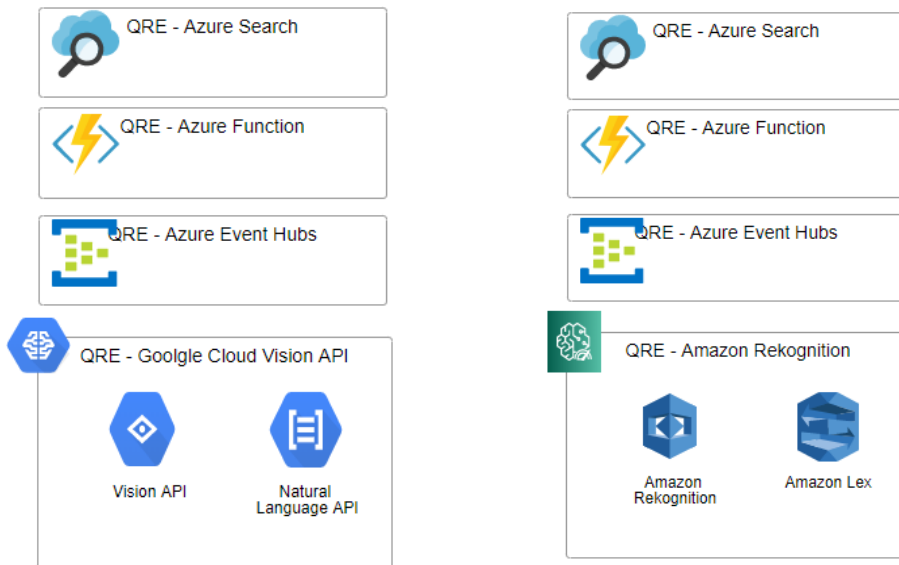


Figure 4-5 : Intégration des API de Google et d’Amazon à l’architecture

#### 4.2.4 Gestion de Stockage (QRE – Azure Cloud Storage)

Différents services de stockage sont utilisés dans le cadre ce projet, les principaux services de stockage sont :

- Azure Blob Storage : qui sert en fait de stockage de tout ce qui est image en cours de traitement postée par les utilisateurs.
- Azure SQL Database : qui sert de base de données transactionnelle au système
- Azure Data Lake : qui sert d'entrepôt de toutes les données brutes qui transitent par l'application (textes, images, audio, vidéo, etc.). C'est également là que sont sauvegardés les résultats des longs processus de traitement de données (méga donné).

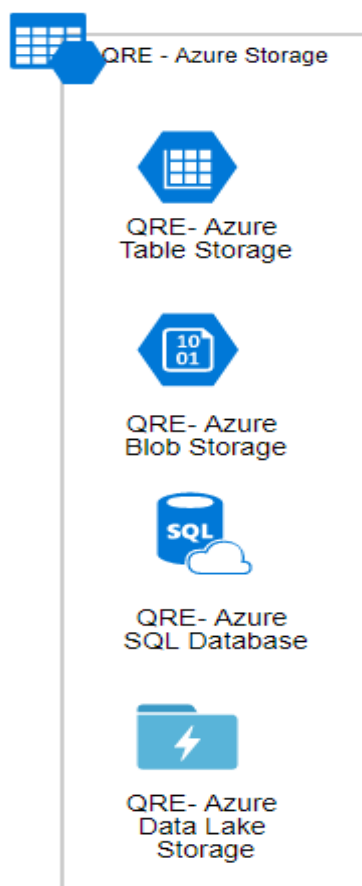


Figure 4-5 : Services de stockage de données

#### 4.2.5 Traitement de données et Analytique (QRE- Azure Data Processing & Analytics)

La couche traitement de données et analytiques est dédiée à tout ce qui a trait au traitement en lots. Toutes les données transactionnelles (c.-à-d. images, vidéos, audios) postées par les utilisateurs sont transférées régulièrement vers le stockage qui sert d'entrepôt de données (nommé Azure Data Lake). Les données supprimées par les utilisateurs ne sont pas physiquement supprimées dans le stockage, elles sont stockées dans un répertoire spécifique et seront utilisées au même titre que les données non supprimées pour l'entraînement du modèle d'intelligence artificielle « Custom » de l'application. Un processus de traitement en lot est lancé régulièrement pour traiter les données inadéquates soumises par les utilisateurs dans le but d'identifier les utilisateurs récidivistes et les polluposteurs et procéder automatiquement à la désactivation de leurs accès. Éventuellement, dans les phases subséquentes du projet, il sera possible de générer un rapport d'activité pour le modérateur du système. Les deux principaux services permettant d'orchestrer une série de traitement de mégadonnées et d'analyse, sont : Azure Data Factory et Azure Data Lake Analytics.

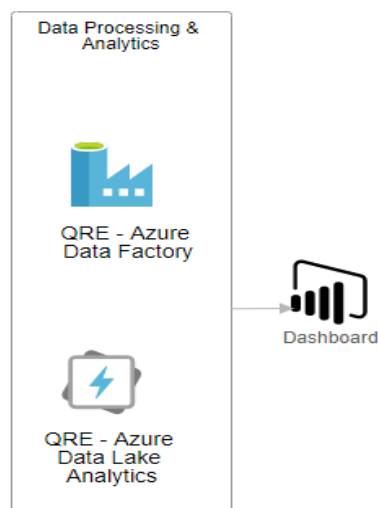


Figure 4-6 : Couche de traitement des mégas donnés en lot

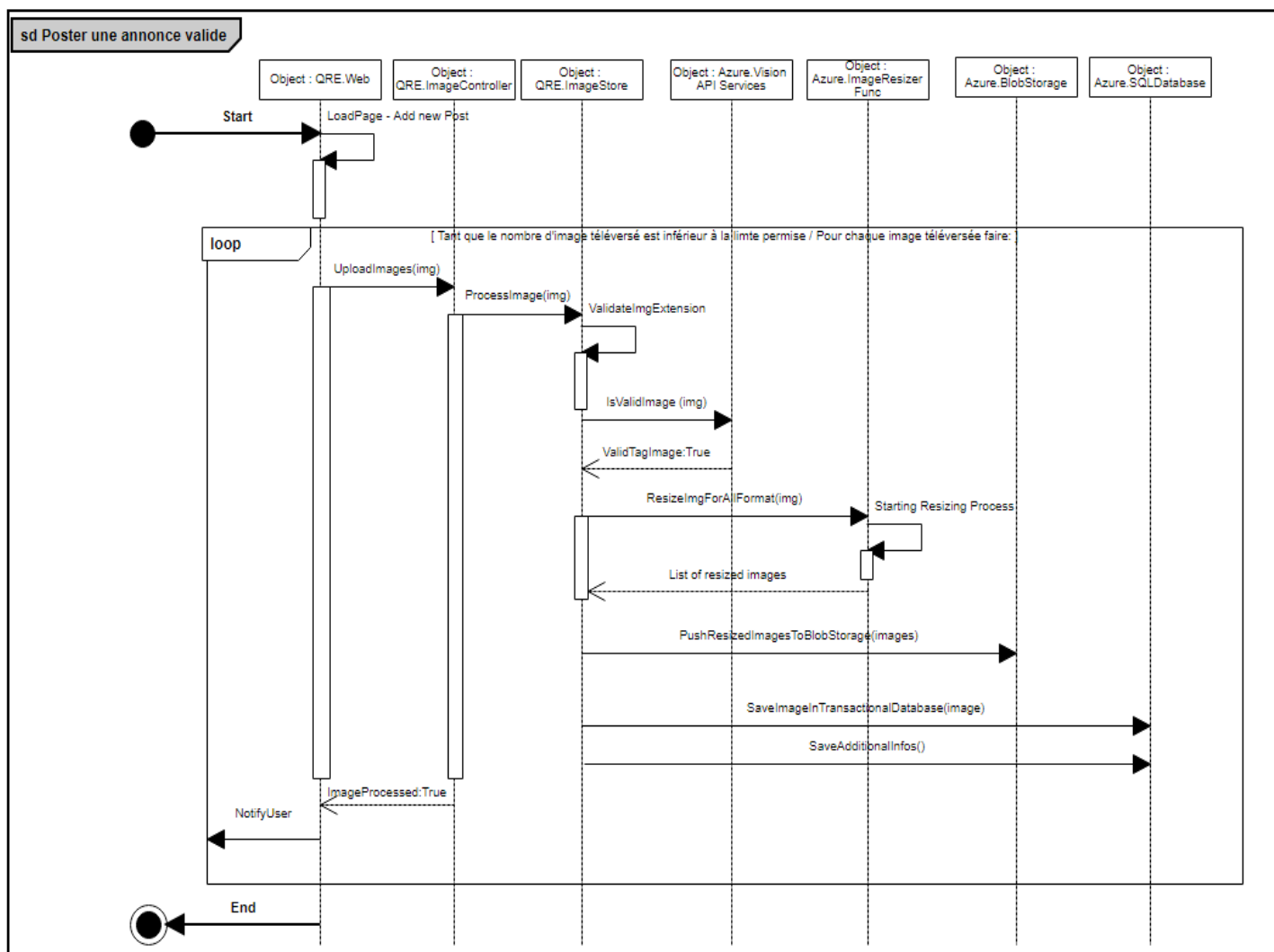
Une couche de gestion des « logs » et de surveillance de l'application, via un service offert par la plateforme Azure, a été instanciée à l'aide de l'application « Insight ».



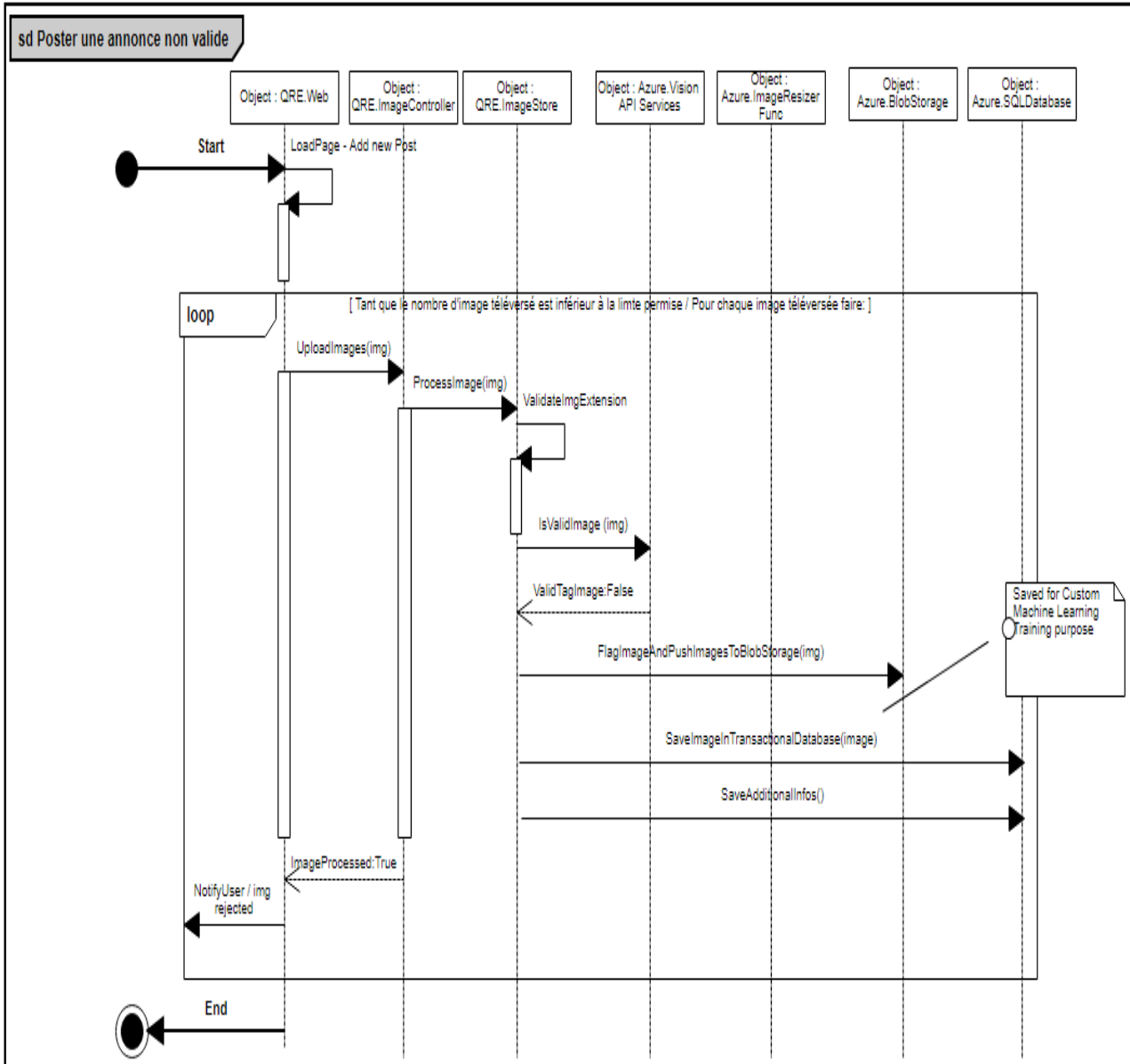
### 4.3 Interaction entre les modules

Quand l'utilisateur est connecté au système applicatif il peut ajouter une annonce. Cette annonce sera validée à l'aide des processus décrits aux section 4.3.1, 4.3.2 et 4.3.3.

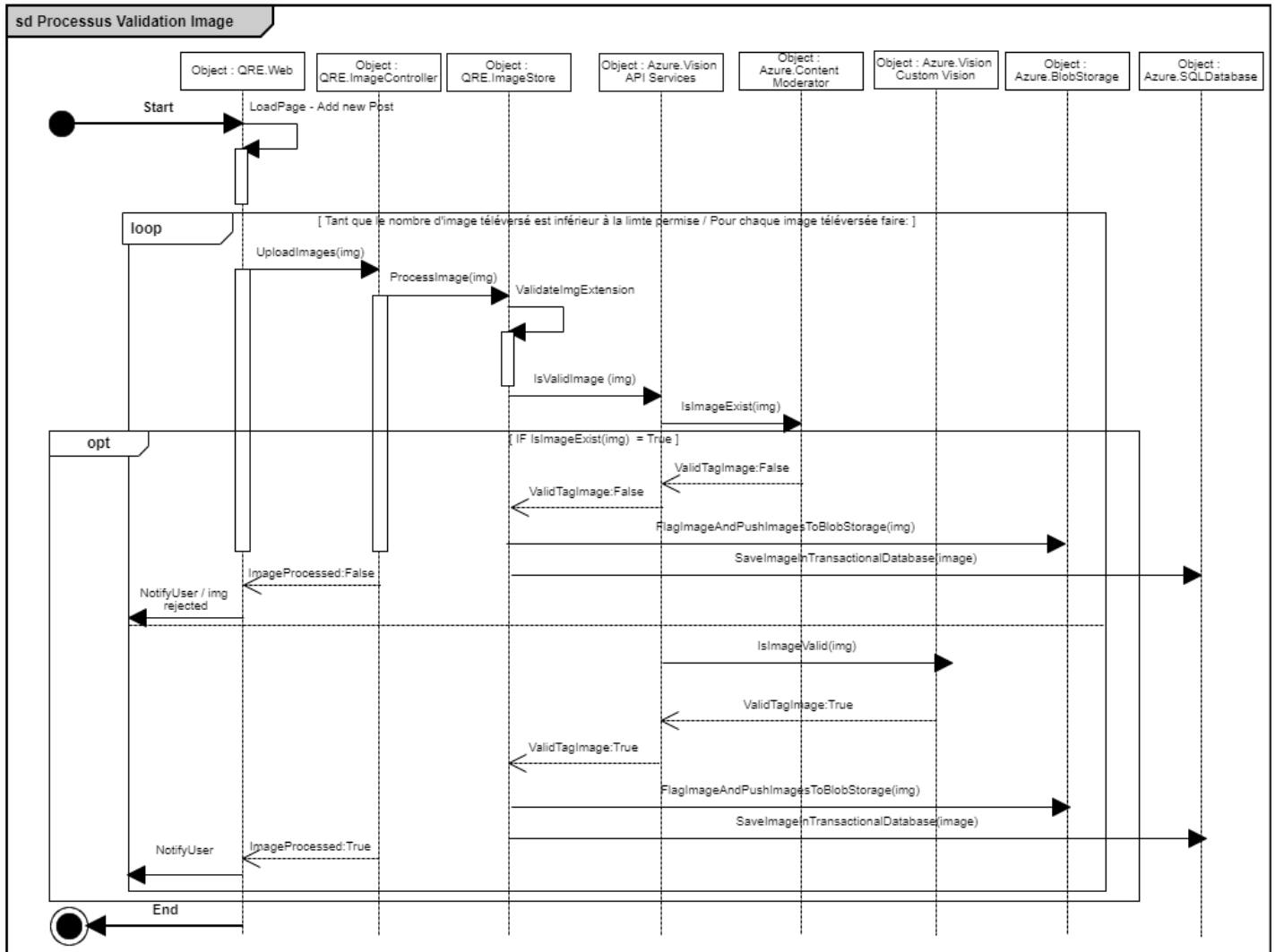
#### 4.3.1 Diagramme de séquence d'ajout d'une annonce valide



### 4.3.2 Diagramme de séquence d'ajout d'une annonce non valide



### 4.3.3 Diagramme de séquence du processus de validation d'une image



### 4.4 Structure de l'architecture du programme dans Visual Studio

Dans Visual Studio, les couches de l'architecture logicielle du prototype sont implémentées comme étant soit des libraires codées en C# ou des applications qui s'arriment bien avec l'environnement Microsoft. La figure ci-dessous montre les différentes couches dans l'IDE Visual Studio :

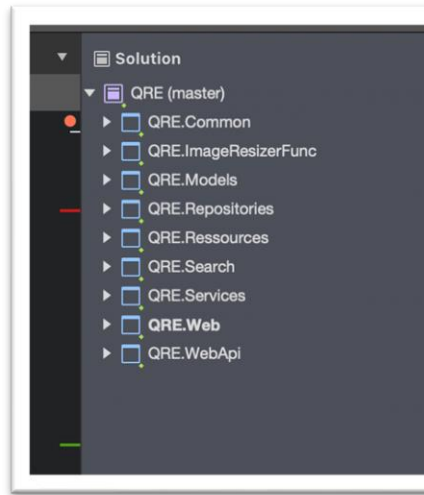


Figure 4-7 : Architecture en couche dans Visual Studio

- QRE.Common (au haut de la figure 4-8): Librairie en C# qui contient toutes les classes utilitaires, les constantes et les énumérations utilisées dans l'application. Cette librairie est référencée par toutes les autres couches ;
- QRE.ImageResizerFunc : Librairie en C# qui est responsable de redimensionner les images soumises par les utilisateurs pour les adapter au format de l'application. Cette librairie est déployée dans Azure via le Module « Azure Function ».

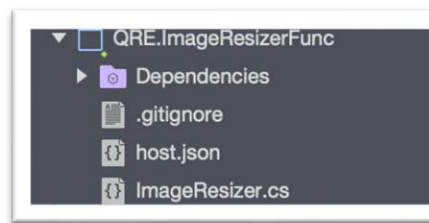


Figure 4-8 : Azure Function Image Resizer

1. QRE.Models : Librairie qui contient tous les objets de type POCO de l'application. Typiquement ces objets sont la représentation des tables de la base de données.

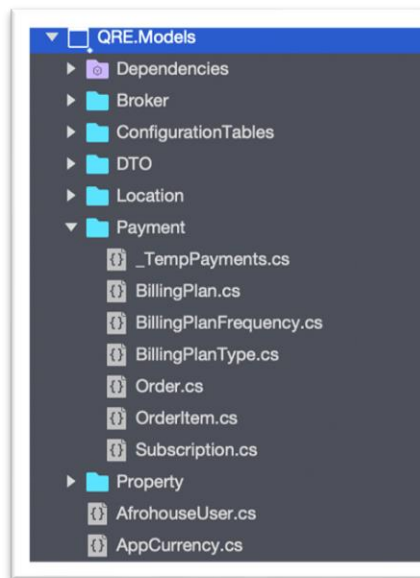


Figure 4-9 : Couche des Modèles POCO

2. QRE.Repositories : Librairie qui est responsable d'être l'intermédiaire entre les services et l'accès à la base de données.

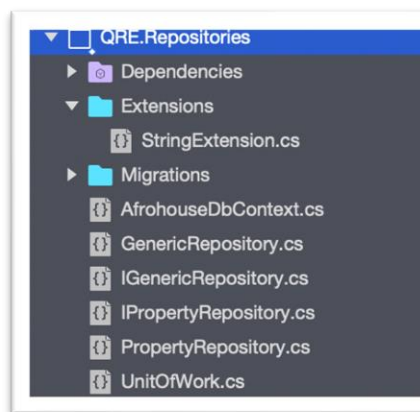


Figure 4-10 : Repositories de l'application

3. QRE.Search : Librairie qui est responsable de tous les aspects recherche de propriété de l'application. Elle utilise étroitement le module Azure.Search pour fonctionner.

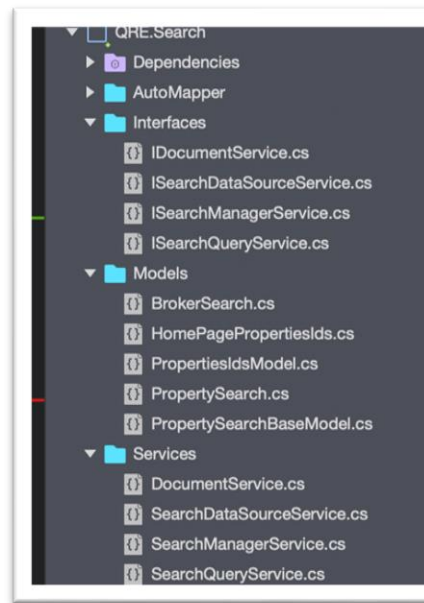


Figure 4-11 : Couche de Search de l'application

4. QRE.Services : La librairie qui contient toute la logique applicative du système, elle expose les services offerts par l'application via des interfaces. Les appels aux différents services de détection et de reconnaissance d'image des différents nuages sont gérés dans cette couche.

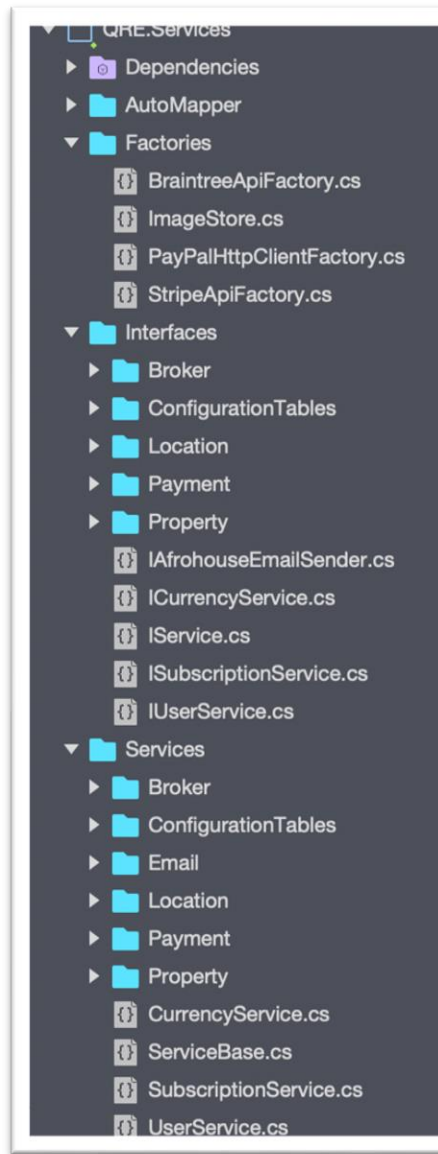


Figure 4-12 : Couche des services de l'application

5. QRE.Web et WebAPI : sont deux applications MVC qui respectivement gèrent tout ce qui est Web et Service Web de l'application.

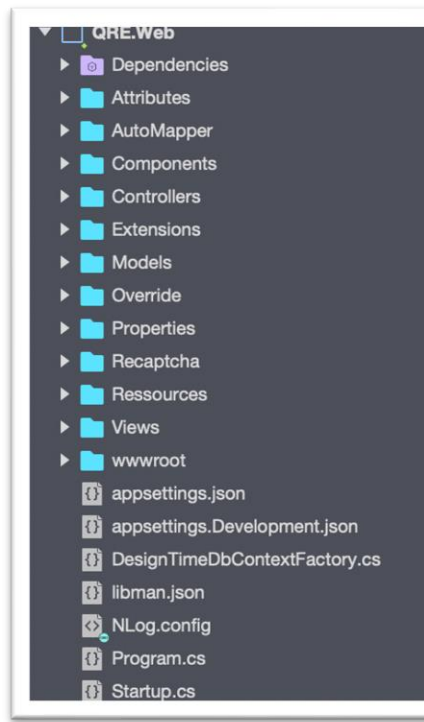


Figure 4-13 : Application Web MVC

#### 4.5 Structure de l'architecture du programme dans le nuage

Dans le nuage les ressourcesinstanciées peuvent être classées en 6 groupes distincts :

- Les ressources de type « APP Service » qui regroupe tout ce qui est application Web, Web API et Web Services ;
- Les ressources de type « Cognitive Services » qui regroupe tout ce qui est Intelligence artificielle et Modèle d'apprentissage « Custom » ;
- Les ressources de type « Search Service » qui regroupe tout ce qui module de « search » ;
- Les ressources de type « Data Lake » qui regroupe tout ce qui est stockage de méga donné et traitement en lots ou en temps réel ;



- Les ressources de type « SQL Database » pour tout ce qui est base de données et serveur de base de données ;
- Les ressources de type « Storage Account » pour tout ce qui est stockage des blobs (images, vidéos, etc..) pour traitement.

The screenshot shows the 'All resources' page in the Azure portal. The page title is 'All resources' and there is a 'Documentat' link in the top right. Below the title, there are several action buttons: '+ Add', 'Edit columns', 'Refresh', 'Export to CSV', 'Assign tags', 'Delete', 'Feedback', 'Preview info', and 'Leave preview'. There is also a search bar 'Filter by name...' and several filter buttons: 'Subscription == all', 'Resource group == all', 'Type == all', and 'Location == all'. Below the filters, it says 'Showing 1 to 13 of 13 records.' and 'Show hidden types'. The main content is a table with the following columns: NAME, TYPE, RESOURCE GROUP, LOCATION, and SUBSCRIPTION. The table contains 13 rows of resources, all belonging to the 'admingrp' resource group and the 'Azure pour les étudiants' subscription. The resources are: qcrealestate (App Service), ASP-admingrp-b05f (App Service plan), qcrealestate (Application Insights), qrecustomvision (Cognitive Services), qrecustomvision\_Prediction (Cognitive Services), qrevisionapi (Cognitive Services), qreadf (Data factory (V2)), qredatalake (Data Lake Analytics), qredatalakeadls (Data Lake Storage Gen1), qresearch (Search service), qre\_db (SQL database), qredbserver (SQL server), and qrestore (Storage account).

NAME	TYPE	RESOURCE GROUP	LOCATION	SUBSCRIPTION
qcrealestate	App Service	admingrp	Canada Central	Azure pour les étudiants
ASP-admingrp-b05f	App Service plan	admingrp	Canada Central	Azure pour les étudiants
qcrealestate	Application Insights	admingrp	Canada Central	Azure pour les étudiants
qrecustomvision	Cognitive Services	admingrp	East US	Azure pour les étudiants
qrecustomvision_Prediction	Cognitive Services	admingrp	East US	Azure pour les étudiants
qrevisionapi	Cognitive Services	admingrp	East US	Azure pour les étudiants
qreadf	Data factory (V2)	admingrp	East US	Azure pour les étudiants
qredatalake	Data Lake Analytics	admingrp	East US 2	Azure pour les étudiants
qredatalakeadls	Data Lake Storage Gen1	admingrp	East US 2	Azure pour les étudiants
qresearch	Search service	admingrp	Canada Central	Azure pour les étudiants
qre_db	SQL database	admingrp	Canada Central	Azure pour les étudiants
qredbserver	SQL server	admingrp	Canada Central	Azure pour les étudiants
qrestore	Storage account	admingrp	Canada Central	Azure pour les étudiants

Figure 4-14 : Liste de toutes les ressources instanciées pour le projet

## 4.6 Conclusion

Ce chapitre présentait l'implémentation technique de la solution, il a présenté les différentes structures, les différents modules et les différentes couches de l'application. Il fait ressortir l'utilisation de plusieurs micro-services dans l'implémentation de la solution. La stratégie de stockage et de traitement de gros volumes de données est aussi abordée. L'architecture en couche MVC « Model View Controller » a été privilégiée pour bâtir la portion applicative Web de la solution. Le prochain chapitre présente les différents tests et les résultats obtenus à la suite de l'implémentation de la solution.



## CHAPITRE 5

### Tests et résultats

#### 5.1 Résultats - Azure Cognitive Services : Vision API

Le modèle IA de Azure Vision API, étant un modèle de reconnaissance à large spectre, un seuil de confiance d'au moins 85% est requis pour ne pas rejeter l'image, ce seuil de 85% est configurable dans l'application. Les catégories acceptables d'images sont celles qui sont de type « building, indoor, outdoor » seulement. Pour tester la robustesse et la fiabilité de l'application, différents scénarios de tests ont été effectués.

- Utilisation d'images ne cadrant pas avec le secteur de l'immobilier ;
- Utilisation d'images relatives au secteur de l'immobilier, mais avec des personnes dénudées ;
- Utilisation d'images relatives au secteur de l'immobilier avec des personnes correctement habillées et bien identifiées ;
- Utilisation d'images relatives au secteur de l'immobilier contenant du texte à caractère raciste ou pour adulte ;
- Utilisation d'images relatives au secteur de l'immobilier contenant des argots ;
- Validation de la langue dans laquelle la description de l'annonce est faite ;
- Validation des mots contenus dans la description de l'annonce ;
- Utilisation d'images relatives au secteur de l'immobilier avec plusieurs personnes de différents âges, correctement habillées et bien identifiées.

Ci-dessous, les résultats de certains des scénarios :

- Utilisation d'images ne cadrant pas avec le secteur de l'immobilier ;



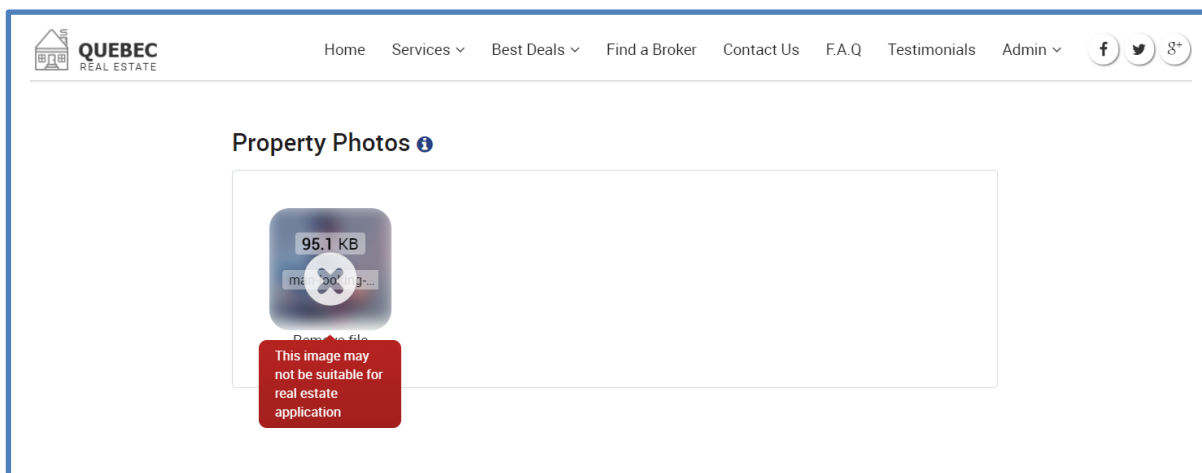
Par exemple, pour cette image, le modèle IA l'identifie à la catégorie « people » avec le plus haut score de confiance qui est de 0.73. En se fiant aux critères et au seuil établi, cette image sera donc rejetée.

```

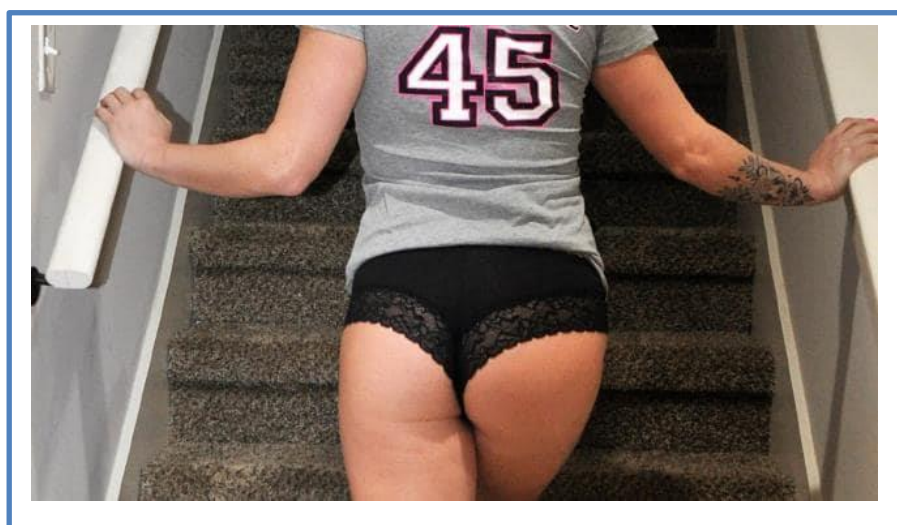
public async Task<ImageAnalysisResult> AnalyzeImageAsync(string imageUrl)
{
    var imageAnalysis = await this._computerVisionClient.AnalyzeImageAsync(imageUrl, this.GetVisualFeatureTypes().ToList());
    var result = new ImageAnalysisResult
    {
        IsImageCategory = imageAnalysis.Categories.Count > 0,
        IsAdult = imageAnalysis.AdultInfo.IsAdultContent,
        IsRacy = imageAnalysis.AdultInfo.IsRacyContent,
        Categories = imageAnalysis.Categories
    };
    return result;
}

```

The image shows a Visual Studio IDE with a debugger window open. The debugger window displays the state of the `imageAnalysis` object. The `Score` property is highlighted with a red arrow and shows the value `0.73046875`. The `Name` property is also visible, showing `"people_show"`. The `Categories` property is expanded to show a list of three categories.



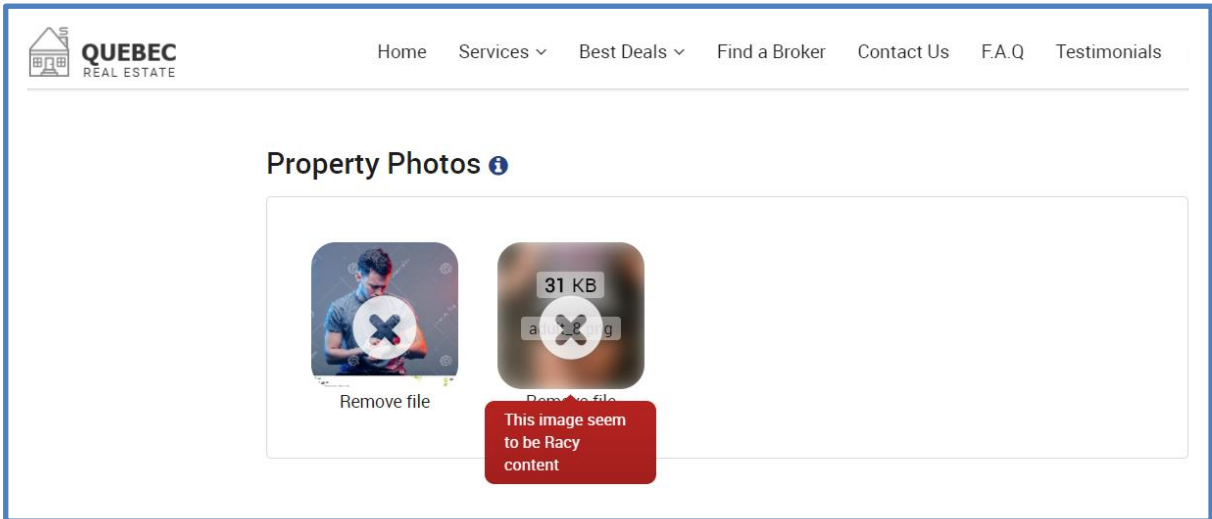
- Utilisation d'images relatives au secteur de l'immobilier, mais avec des personnes dénudées ;



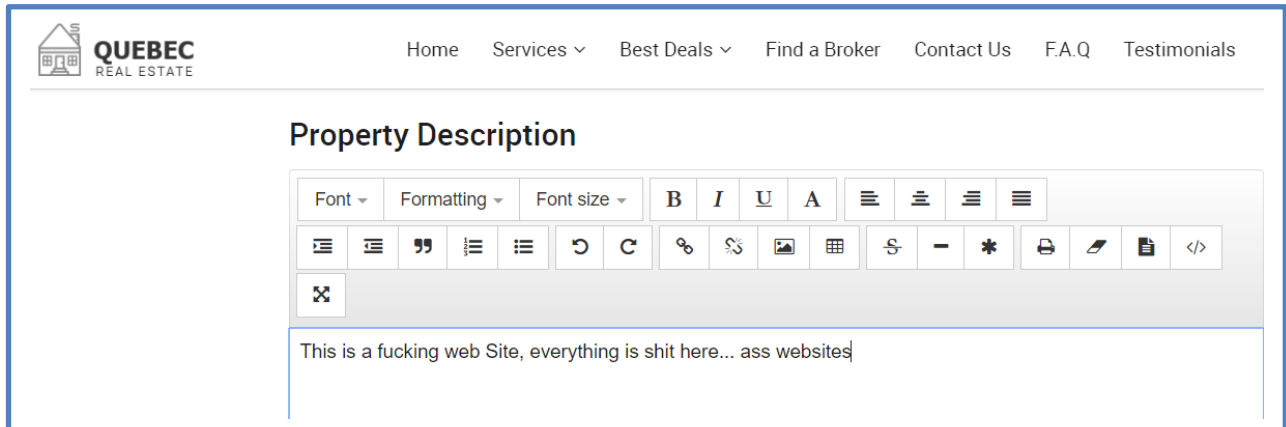
Pour cette image, le modèle IA accorde le score le plus élevé (0.53) à la catégorie « people » et classifie cette image comme ayant du contenu racisé avec une confiance de 0.91. Cette image est donc elle aussi rejetée.

```
public async Task<ImageAnalysisResult> AnalyzeImageAsync(string imageUrl)
{
    var imageAnalysis = await this._computerVisionClient.AnalyzeImageAsync(imageUrl, this.GetVisualFeatureTypes().ToList());
    var result = new ImageAnalysisResult
    {
        Adult = imageAnalysis.AdultInfo,
        Brands = imageAnalysis.Brands,
        IsImageCategoryA = imageAnalysis.Categories,
        IsAdult = imageAnalysis.IsImageAdult,
        IsRacy = imageAnalysis.IsImageRacy,
        Categories = imageAnalysis.Categories;
    };
    return result;
}
```

```
public async Task<ImageAnalysisResult> AnalyzeImageAsync(string imageUrl)
{
    var imageAnalysis = await this._computerVisionClient.AnalyzeImageAsync(imageUrl, this.GetVisualFeatureTypes().ToList());
    var result = new ImageAnalysisResult
    {
        IsImageCategoryA = imageAnalysis.Categories,
        IsAdult = imageAnalysis.IsImageAdult,
        IsRacy = imageAnalysis.IsImageRacy,
        Categories = imageAnalysis.Categories;
    };
    return result;
}
```



- Validation des mots contenus dans la description de l'annonce ;



Avec une extension faite au modèle IA, il a été aussi possible de vérifier que les mots contenus dans un texte ne contiennent pas d'argots et même déterminer la langue dans laquelle le texte est écrit. Dans l'exemple ci-dessus, le modèle est parvenu à détecter trois mots à rejeter.

```

public async Task<TextAnalysisResult> AnalyzeTextAsync(string inputText)
{
    if (!string.IsNullOrEmpty(inputText))
    {
        var lang = await this._ContentModeratorClient.TextModeration.DetectLanguageAsync("text/html", new MemoryStream(EncodedText));
        var textAnalysis = await this._ContentModeratorClient.TextModeration.ScreenTextAsync("text/html", new MemoryStream(EncodedText));
        lang.Detect
        return new TextAnalysisResult
        {
            IsValidText = true,
            ScreenInfo = null,
            Language = lang.Detect.Language;
        };
    }
    //Return default
    return new TextAnalysisResult
    {
        IsValidText = true,
        ScreenInfo = null,
        Language = "en";
    };
};

```

The screenshot shows the execution of the `AnalyzeTextAsync` method. The `lang.Detect` property is expanded, showing the following details:

- `AutoCorrectedText`: null
- `Classification`: (Microsoft.Azure.CognitiveServices.ContentModerator.Models.Classification)
- `Language`: "eng"
- `Misrepresentation`: null
- `NormalizedText`: " fucking web Site, shit ... ass websites"
- `OriginalText`: "This is a fucking web Site, everything is shit here... ass websites"
- `PII`: null
- `Status`: (Microsoft.Azure.CognitiveServices.ContentModerator.Models.Status)
- `Terms`: Count = 3

The `Terms` property is further expanded, showing a list of detected terms:

- [0]: (Microsoft.Azure.CognitiveServices.ContentModerator.Models.DetectedTerms)
- [1]: (Microsoft.Azure.CognitiveServices.ContentModerator.Models.DetectedTerms)
- [2]: (Microsoft.Azure.CognitiveServices.ContentModerator.Models.DetectedTerms)

The `IsValidText` property is set to `true`, `ScreenInfo` is `null`, and `Language` is `en`. The `Term` property is expanded, showing the detected term: "fucking".

The screenshot shows the top navigation bar of the Quebec Real Estate website. The logo is on the left, followed by a menu with items: Home, Services, Best Deals, Find a Broker, Contact Us, F.A.Q, Testimonials, and Admin. Social media icons for Facebook, Twitter, and a plus sign are on the right. Below the navigation is a horizontal menu with seven icons and labels: Infos, Images, Profile, Preview, Plans, Payment, and Confirmation. A red error message box is displayed, stating: "Validation Errors. The following words: 'shit,ass,fucking' used in the description are not allowed, please delete them or replace them with acceptable words." Below the error message is a large heading that says "ADD YOUR PROPERTY".

The screenshot shows the "Property Description" editor on the Quebec Real Estate website. The navigation bar is identical to the previous screenshot. The page title is "Property Description". Below the title is a red error message: "The following words: 'shit,ass,fucking' used in the description are not allowed, please delete them or replace them with acceptable words." Below the error message is a rich text editor toolbar with various icons for text formatting (bold, italic, underline, text color, background color, text alignment, list, link, unlink, image, table, link, unlink, print, undo, redo, code) and a text area. The text area contains the text: "This is a fucking web Site, everything is shit here... ass websites".



## 5.2 Résultats - Azure Cognitive Services : Custom Vision API

Sur plus de 1620 images analysées, de différentes catégories, avec des luminosités, contrastes et couleur de fond différente, le modèle IA personnalisé de QRE est parvenu à identifier et à classifier correctement 1580 images avec un degré de confiance de plus de 60 %.

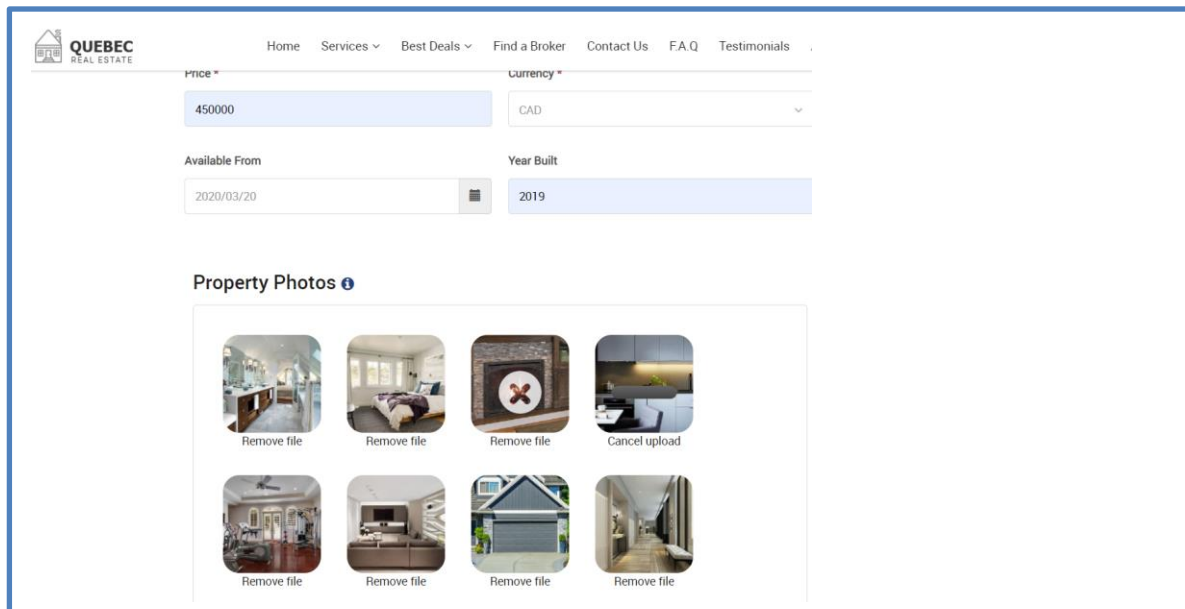
	Degré de confiance > 0.6	Degré de confiance < 0.6
Nombre d'images	1580	40
Ratio	0.97%	0.024%

Ci-dessous un exemple du processus de soumission d'image et de classification avec le modèle IA personnalisé :

1. Saisir les données obligatoires du formulaire et télécharger les images de l'annonce à publier

The screenshot shows the 'ADD YOUR PROPERTY' form on the Quebec Real Estate website. The form is titled 'ADD YOUR PROPERTY' and includes a 'Property Info' section. The 'Submit Property' button is highlighted in yellow. The form fields are as follows:

- Title \***: Maison De La Rue Test
- Region \***: Montréal Ile
- City \***: LaSalle



2. Pour chaque image téléchargée, déterminée si l'image est une image valide et la classifiée

Chaque image téléchargée est soumise à une analyse et une classification. Le modèle IA retourne une probabilité pour chaque catégorie, la catégorie ayant la plus forte probabilité, supérieure au seuil fixé (de 0.6) est retenue comme la catégorie à assigner à l'image.

```
public async Task<PropertyImageNatureEnum> ClassifyImage(string imageUrl)
{
    var predictionResult = await this._customVisionPredictionClient
        .ClassifyImageUrlAsync(this._projectId, this._modelName, new ImageUrl(imageUrl));

    if (predictionResult != null && predictionResult.Predictions != null
        && predictionResult.Predictions.Any())
    {
        var prediction = predictionResult.Predictions
            .Where(x => x.Probability >= 0.6)
            .OrderByDescending(x => x.Probability)
            .FirstOrDefault();

        if (prediction != null)
        {
            return prediction.TagName.GetEnumFromString();
        }
    }

    var elem = predictionResult.Predictions[0];

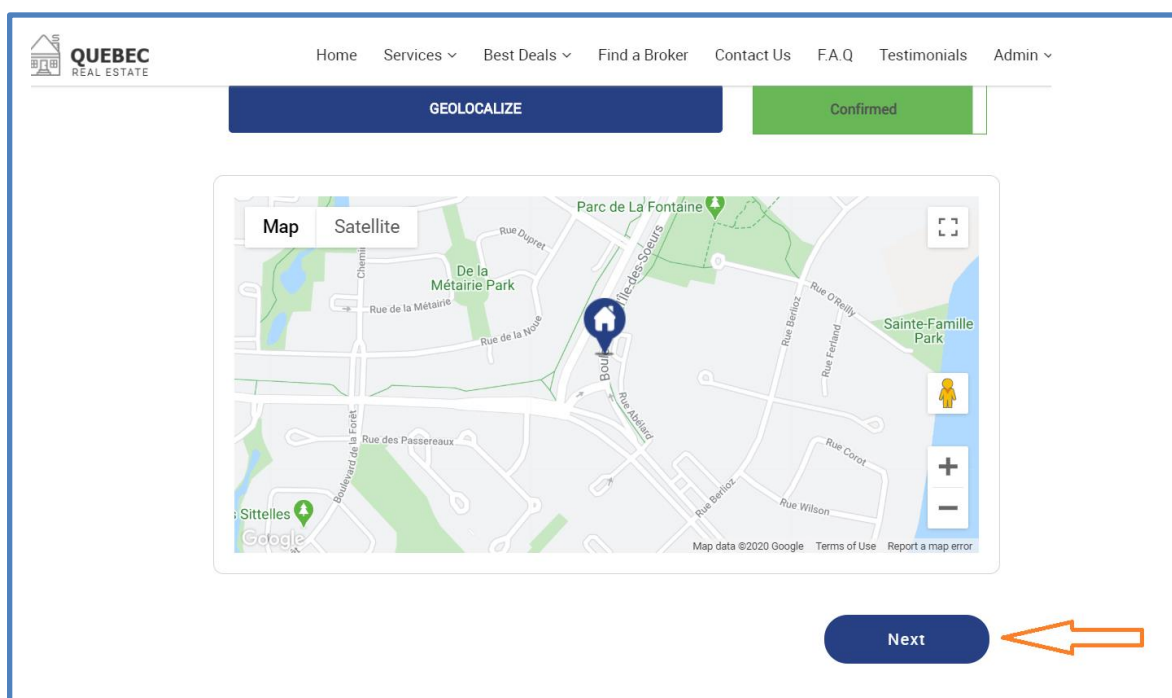
    return PropertyImageNatureEnum.NONE;
}
```

The screenshot also shows a Visual Studio debugger window displaying the results of the `predictionResult.Predictions` collection. The collection contains 14 items. The first item (index 0) is expanded, showing the following properties:

- BoundingBox**: null
- Probability**: 0.928477168
- TagId**: {12eb7208-4a86-44bb-9525-688e081aa7a6}
- TagName**: "Bathrooms"

An orange arrow points to the `predictionResult.Predictions` property in the code, and another orange arrow points to the first item in the debugger window.

S'assurer que la propriété est bien géolocalisée et cliquer sur le bouton suivant pour accéder à l'étape suivante.




3. Passer à l'étape suivante de sélection de l'ordre d'apparition des images et de la confirmation du type de l'image

À cette étape le résultat de classification des images téléchargées par l'utilisateur est affiché. À chaque image est assignée la catégorie que le modèle IA a déterminé. L'utilisateur a aussi la possibilité de choisir une autre catégorie à assigner à l'image. Après quoi le modèle IA sera ré-entraîné avec les modifications faites.

REAL ESTATE


Infos Images Profile Preview Plans Payment Confirmation

### ORDER IMAGE AND CONFIRM IMAGE TYPE




img021.jpg

01 Bathrooms



img002.jpg

02 Bedrooms




img01.

03 Gym


QUEBEC REAL ESTATE

Home Services Best Deals Find a Broker Contact Us F.A.Q Testimonials Admin




img020.jpg

04 TvLounge



img029.jpg

05 Garages



img005.jpg

06 Corridor

Previous Next

4. Suivre le flux d'enchaînement des écrans jusqu'à la mise en ligne finale de l'annonce

Cliquer soit sur le bouton suivant pour continuer le processus de mise en ligne de l'annonce ou sur le bouton précédant pour opérer des modifications aux étapes passées.

### **5.3 Conclusion**

Ce chapitre avait pour objectif de présenter les différents scénarios de tests mis en œuvre ainsi que les résultats obtenus. Il décrit également la performance des différents modèles IA utilisés, les seuils d'acceptabilité, les seuils de rejection et les données d'échantillonnage. Il décrit aussi le processus à suivre pour mettre en ligne une annonce valide. La section suivante est une conclusion du projet.



## CONCLUSION

L'objectif principal du présent projet de recherche appliquée est d'étudier la possibilité d'utiliser la vision assistée par ordinateur pour valider automatiquement, sans interventions humaines, le contenu des annonces publiées par les utilisateurs sur une plateforme Web immobilière. Pour ce faire, il a fallu en premier lieu analyser différents algorithmes de traitement d'images de différents fournisseurs infonuagiques, en retenir que quelques-uns, puis instancier les services nécessaires permettant de bâtir et entraîner les modèles IA.

Par la suite élaborer une architecture logicielle axée sur des micro-services permettant une meilleure évolutivité ou élasticité « Scalability » de la solution. Advenant un trafic utilisateur élevé, l'architecture de la solution a été conçue pour stocker et traiter de gros volume de données. Enfin une implémentation technique de la solution a été faite avec élaboration de différents scénarios de tests pour concrétiser la faisabilité du projet.

Maintenant, l'intelligence artificielle, le big-data et l'infonuagique font partie des domaines de la technologie en constante évolution, il peut être intéressant d'utiliser une combinaison des services offerts par « Google Cloud Vision API » et un autre fournisseur infonuagique pour bâtir des modèles IA plus robustes et plus efficaces. Par exemple le modèle générique de vision de Google Cloud permet de savoir si une image provient d'un autre site Web sur internet, cette fonctionnalité par exemple aurait permis de détecter les utilisateurs qui copieraient des images d'un site Web à l'autre et les bloquer.

Une analyse du contenu vidéo/audio et textuel plus approfondie en tenant compte de la langue de saisie pourrait aussi faire l'objet d'une autre phase d'implémentation et être une continuité au présent projet.

Pour terminer, il est important de souligner que les connaissances acquises dans le programme de Maîtrise en Génie Logiciel ont beaucoup contribué à l'élaboration de la planification, de la conception et de l'exécution de ce projet.



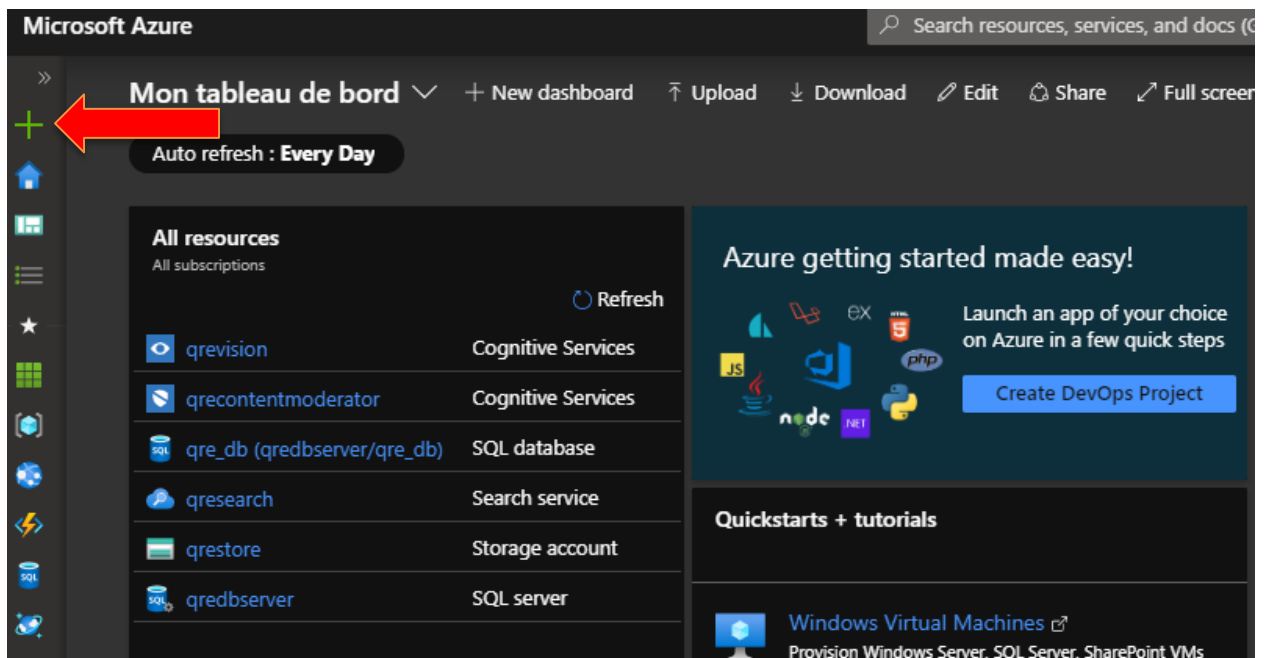


## ANNEXE I

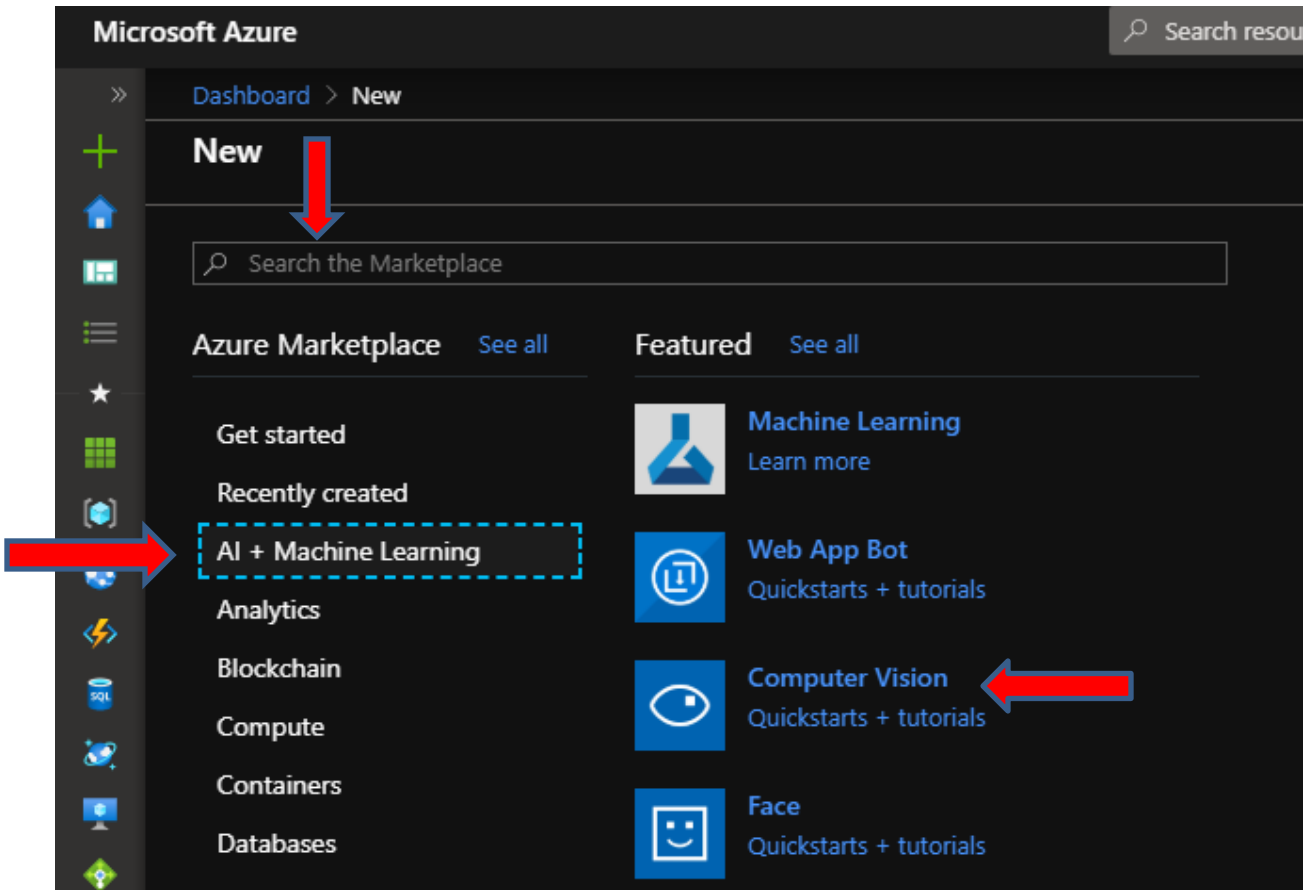
### Exemple d'instanciation de services dans Azure

Ci-dessous un exemple d'instanciation d'un service AI de détection d'objet dans Microsoft Azure :

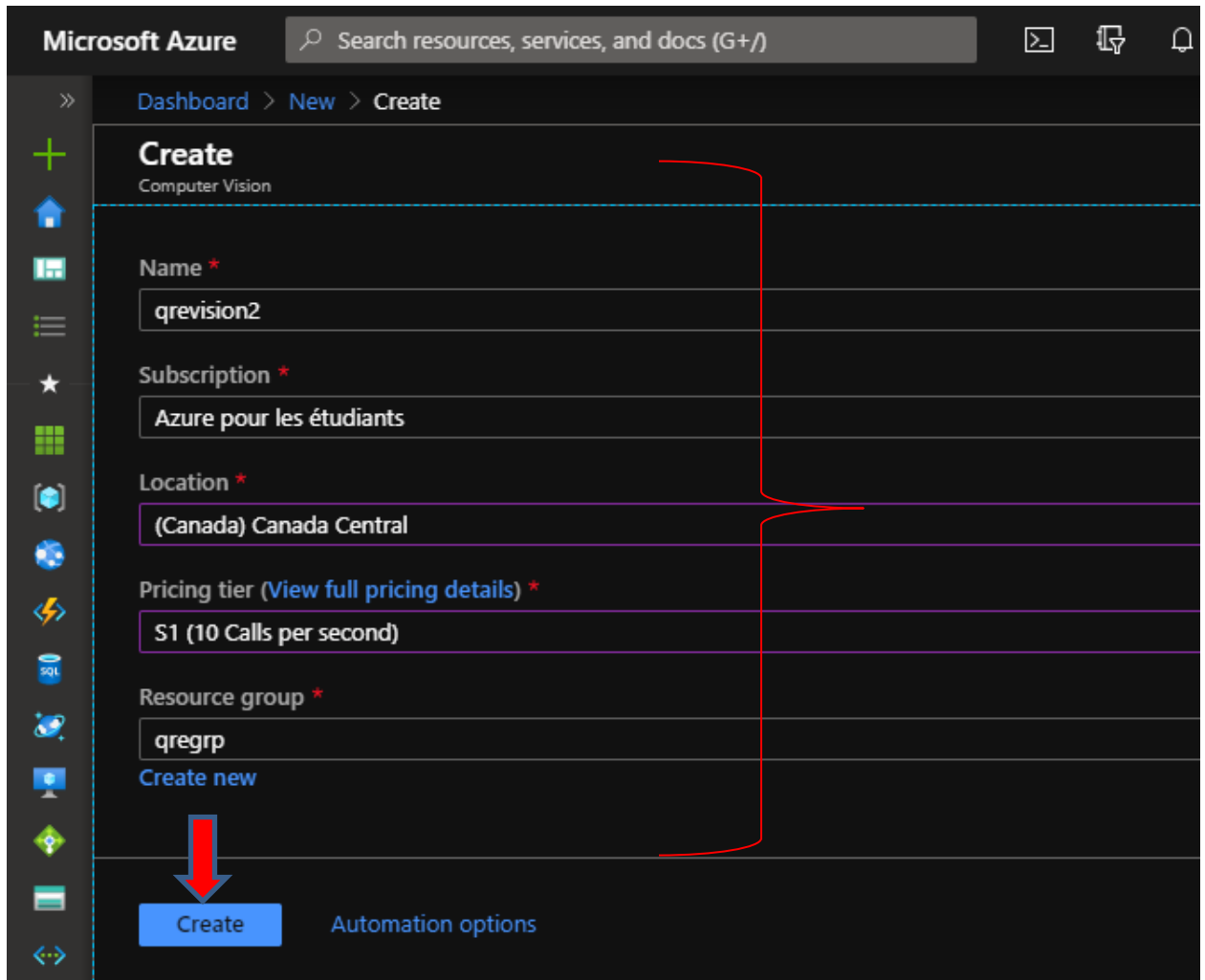
1. Se connecter sur le portail Web de Microsoft Azure : <https://portal.azure.com/>



2. Cliquer sur l'icône « plus » en vert pour ajouter une nouvelle ressource et choisir la catégorie de la ressource à ajouter. Dans notre cas, le service « Computer Vision » que nous voulons instancier se retrouve dans la catégorie « AI + Machine Learning ». Une barre de recherche existe aussi pour faciliter la recherche de service.



3. Cliquer sur le service recherché et remplir le formulaire des champs obligatoires puis créer le service.



4. Une fois le service créé, vous avez toutes les informations vous permettant d'utiliser et de gérer le service.

Microsoft Azure Search resources, services, and docs (G+)

Dashboard > qreversion - Quick start

### qreversion - Quick start

Cognitive Services

Search (Ctrl+/)

- Overview
- Activity log
- Access control (IAM)
- Tags
- Diagnose and solve problems

RESOURCE MANAGEMENT

- Quick start
- Keys and Endpoint
- Pricing tier
- Virtual network
- Billing By Subscription
- Properties

Explore the Quickstart guidance to get started with Computer Vision.

# 1

Get the API Key and endpoint to authenticate your applications and start sending calls to

Key1 ⓘ

Endpoint ⓘ

`https://qreversion.cognitiveservices.azure.com/`

All Computer Vision calls and Docker container activations require a key. Specify the key either in the re

# 2

Try the service in the API console - requires an API Key and selecting your location: canad

Use the API Console to quickly try the API without writing code. Be sure to select the correct location fo

[API Console](#)

## ANNEXE II

## Les services instanciés dans le cadre du projet

All resources Documentat

[+](#) Add
 [≡](#) Edit columns
 [↻](#) Refresh
 [↓](#) Export to CSV
 [🏷](#) Assign tags
 [🗑](#) Delete
 [❤](#) Feedback
 [ℹ](#) Preview info
 [🚪](#) Leave preview

Filter by name...
 Subscription == all
 Resource group == all
 Type == all
 Location == all
 [+](#) Add filter

Showing 1 to 13 of 13 records.  Show hidden types

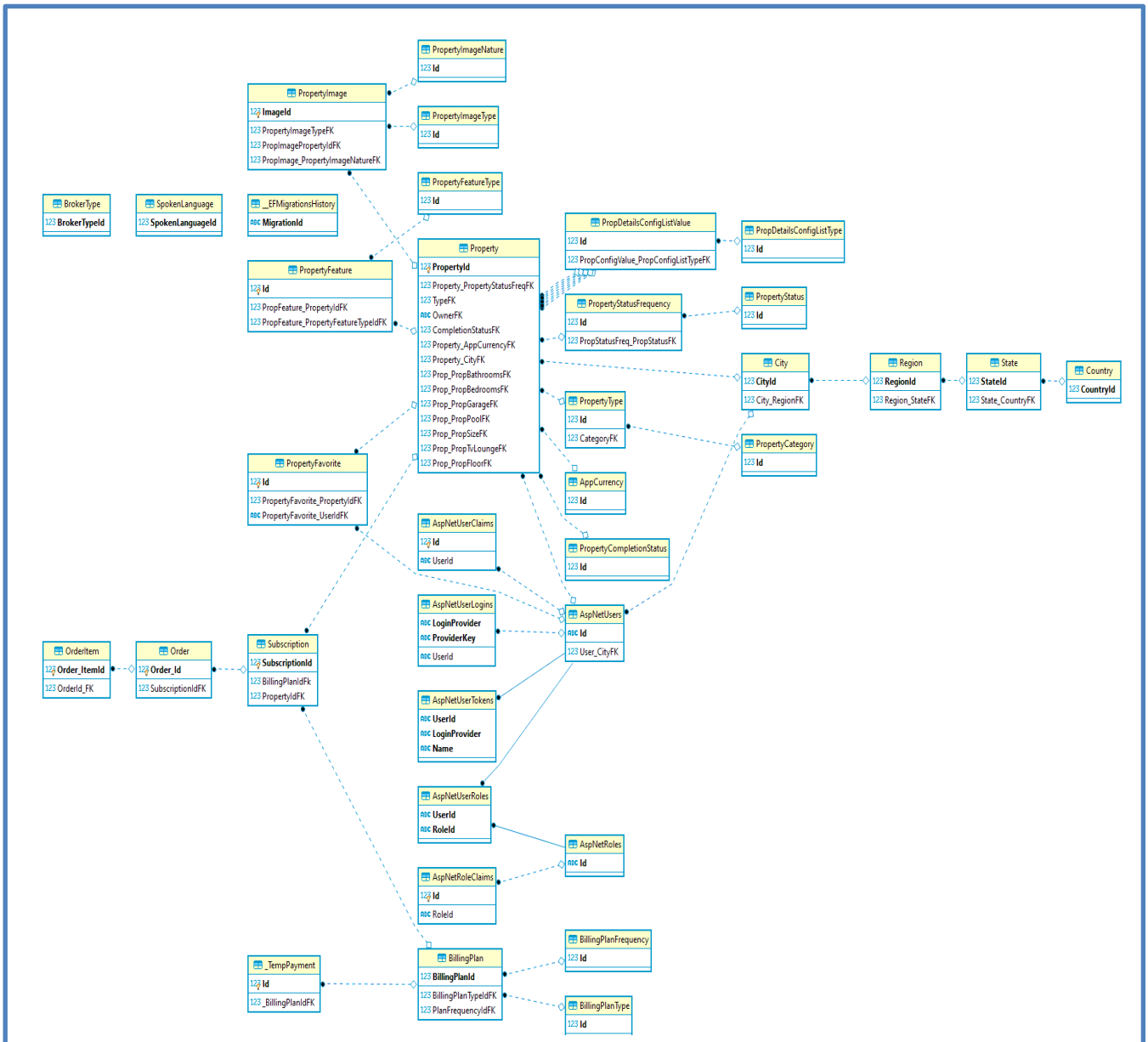
<input type="checkbox"/> NAME	<input type="checkbox"/> TYPE	<input type="checkbox"/> RESOURCE GROUP	<input type="checkbox"/> LOCATION	<input type="checkbox"/> SUBSCRIPTION
<input type="checkbox"/> qcrealestate	App Service	admingrp	Canada Central	Azure pour les étudiants
<input type="checkbox"/> ASP-admingrp-b05f	App Service plan	admingrp	Canada Central	Azure pour les étudiants
<input type="checkbox"/> qcrealestate	Application Insights	admingrp	Canada Central	Azure pour les étudiants
<input type="checkbox"/> qrecustomvision	Cognitive Services	admingrp	East US	Azure pour les étudiants
<input type="checkbox"/> qrecustomvision_Prediction	Cognitive Services	admingrp	East US	Azure pour les étudiants
<input type="checkbox"/> qrevisionapi	Cognitive Services	admingrp	East US	Azure pour les étudiants
<input type="checkbox"/> qreadf	Data Factory (V2)	admingrp	East US	Azure pour les étudiants
<input type="checkbox"/> qredatalake	Data Lake Analytics	admingrp	East US 2	Azure pour les étudiants
<input type="checkbox"/> qredatalakeadls	Data Lake Storage Gen1	admingrp	East US 2	Azure pour les étudiants
<input type="checkbox"/> qresearch	Search service	admingrp	Canada Central	Azure pour les étudiants
<input type="checkbox"/> qre_db	SQL database	admingrp	Canada Central	Azure pour les étudiants
<input type="checkbox"/> qredbserver	SQL server	admingrp	Canada Central	Azure pour les étudiants
<input type="checkbox"/> qrestore	Storage account	admingrp	Canada Central	Azure pour les étudiants

< Previous Page 1 of 1

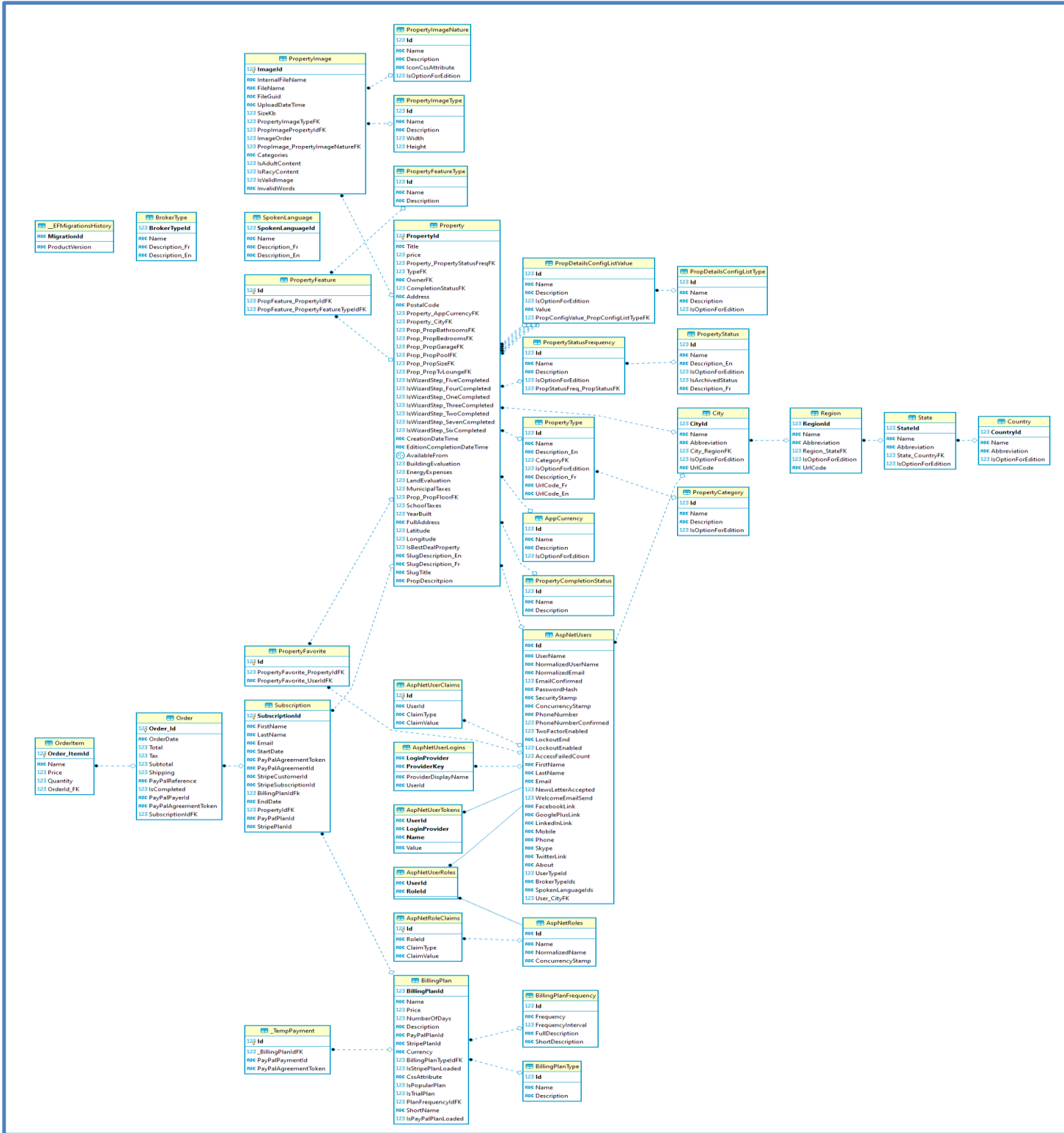
## ANNEXE III

### Schéma de la base de données transactionnelle

#### 1. Version simplifiée du schéma pour une meilleure lisibilité



## 2. Version complète avec toutes les propriétés



## LISTE DE RÉFÉRENCES BIBLIOGRAPHIQUES

- [1] Conférence F8 du 1<sup>er</sup> mai 2019 de Facebook : le résumé des nouvelles avancées en Intelligence Artificielle <<https://ai.facebook.com/blog/ai-recap-2019-f8/>>, consulté le 27 mars 2020.
- [2] Technique améliorée de détection et reconnaissance d'Objects <<https://ai.facebook.com/blog/advances-in-content-understanding-self-supervision-to-protect-people/>>, consulté le 27 mars 2020.
- [3] Kirillov, A., Girshick, R. et coll. (2019) Panoptic Feature Pyramid Networks, Facebook AI Research (FAIR), 10p. <<https://arxiv.org/pdf/1901.02446.pdf>>, consulté le 27 mars 2020.
- [4] Site Web de l'OACIQ des transactions immobilières et des statistiques <<https://www.oaciq.com/fr/oaciq/statistiques>>, consulté le 27 mars 2020.
- [5] Intelligence collective <[https://fr.wikipedia.org/wiki/Intelligence\\_collective](https://fr.wikipedia.org/wiki/Intelligence_collective)>, consulté le 27 mars 2020.
- [6] Trabelsi, S. et Bouafif H. (2013) «Abusing social networks with abuse reports: A coalition attack for social networks», *2013 International Conference on Security and Cryptography (SECRYPT)*, 29-31 juillet, Reykjavik, Iceland, 6p., <<https://ieeexplore.ieee.org/document/7223206>>, consulté le 27 mars 2020.
- [7] Tyagi, S., Pai, A., Pegado, J. et Kamath, A. (2019) « A Proposed Model for Preventing the spread of misinformation on Online Social Media using Machine Learning », *2019 Amity International Conference on Artificial Intelligence (AICAI)*, 4-6 février, Dubai, UAE, pp. 678—683, consulté le 27 mars 2020.
- [8] Al Assad, B. et Erascu, M. (2018) « A Tool for Fake News Detection », *2018 20th International Symposium on Symbolic and Numeric Algorithms for Scientific Computing (SYNASC)*, 20-23 septembre, Timisoara, Roumanie, pp. 379—386, consulté le 27 mars 2020.
- [9] Manuel E., Gianluca S., Christopher K. et Giovanni V. «Towards Detecting Compromised Accounts on Social Networks», *IEEE Transactions on Dependable and Secure Computing*, consulté le 27 mars 2020.
- [10] Danilo C. et Joseana M. « A Machine Learning Forensic Discriminator of Pornographic and Bikini Images», *2018 International Joint Conference on Neural Networks (IJCNN)*, consulté le 27 mars 2020.
- [11] Commentaires haineux: Facebook écope d'une amende de 3 millions de dollars <<https://ici.radio-canada.ca/nouvelle/1207275/facebook-amende-allemande-commentaires-haineux-europe>>, consulté le 27 mars 2020.
- [12] Nuage Computing <[https://fr.wikipedia.org/wiki/Cloud\\_computing](https://fr.wikipedia.org/wiki/Cloud_computing)>, consulté le 27 mars 2020.
- [13] A Brief History of Cloud Computing <<https://www.ibm.com/cloud/blog/cloud-computing-history>>, consulté le 27 mars 2020.



- [14] Top 4 Cloud Deployment Models <<https://www.sam-solutions.com/blog/four-best-cloud-deployment-models-you-need-to-know/>>, consulté le 27 mars 2020.
- [15] Cloud Computing – Types of Cloud <<https://www.esds.co.in/blog/cloud-computing-types-cloud/#sthash.XBHDqkzv.dpbs>>, consulté le 27 mars 2020.
- [16] What are the different types of cloud computing services?  
<<https://azure.microsoft.com/en-ca/overview/types-of-cloud-computing/>>, consulté le 27 mars 2020.
- [17] What is IaaS ? <<https://azure.microsoft.com/en-ca/overview/what-is-iaas/>>, consulté le 27 mars 2020.
- [18] Gartner Magic Quadrant for Cloud Infrastructure as a Service 2018 <<https://www.bmc.com/blogs/gartner-magic-quadrant-cloud-iaas/>>, consulté le 27 mars 2020.
- [19] The Pros and Cons of Cloud Computing <<https://www.ibm.com/blogs/cloud-computing/2014/05/13/the-pros-and-cons-of-cloud-computing/>>
- [20] Intelligence artificielle, <[https://fr.wikipedia.org/wiki/Intelligence\\_artificielle](https://fr.wikipedia.org/wiki/Intelligence_artificielle)>, consulté le 27 mars 2020.
- [21] Intelligence artificielle : Definition, Types, Exemples, Technologies <<https://medium.com/@chethankumargn/artificial-intelligence-definition-types-examples-technologies-962ea75c7b9b>>, consulté le 27 mars 2020.
- [21] Web Services Tutorial for Beginners, <<https://www.guru99.com/Web-services-tutorial.html>>, consulté le 27 mars 2020.
- [22] Service Web Wikipédia <[https://fr.wikipedia.org/wiki/Service\\_Web](https://fr.wikipedia.org/wiki/Service_Web)>, consulté le 27 mars 2020.
- [23] REST, Wikipédia <[https://fr.wikipedia.org/wiki/Representational\\_state\\_transfer](https://fr.wikipedia.org/wiki/Representational_state_transfer)>, consulté le 27 mars 2020.
- [24] Azure Cognitive Services <<https://docs.microsoft.com/en-us/azure/cognitive-services/welcome>>, consulté le 27 mars 2020.
- [25] Azure Vision API <<https://docs.microsoft.com/en-us/azure/cognitive-services/computer-vision/home>>, consulté le 27 mars 2020.
- [26] Dinarle O., Elluz U. et Maria M. Guevara. « Enterprise Architecture and Web Services », *2009 Fourth International Conference on Internet and Web Applications and Services*, consulté le 27 mars 2020.
- [27] Chuangwei Z. et Xu Y. « Design and implementation of single-service multi-function Webservice », *2011 International Conference on Computer Science and Service System (CSSS)*, consulté le 27 mars 2020.
- [28] Azure Custom Vision API <<https://docs.microsoft.com/en-us/azure/cognitive-services/custom-vision-service/home>>, consulté le 27 mars 2020.