



# Exploration dans le domaine des agents intelligents et des chatbots

**HEG Arc - Haute école Arc - Gestion**

**Mention :** Travail de Bachelor d'informaticien de gestion

**Module :** Thèse de bachelor

**Sujet :** Exploration dans le domaine des agents intelligents et des chatbots

**Directeur de travail :** Frischer Boris

**Auteur :** Klett Boris

**Cycle d'étude :** 2013 - 2017

**Date de restitution :** Septembre 2017



# Table des matières

---

<b><i>Introduction</i></b>	<b>6</b>
<b>1</b> <b>Introduction</b>	<b>8</b>
<b>2</b> <b>Présentation du document</b>	<b>9</b>
<b>3</b> <b>Objectif du travail</b>	<b>9</b>
<b><i>Partie théorique</i></b>	<b>12</b>
<b>4</b> <b>Introduction</b>	<b>14</b>
<b>5</b> <b>Problématique</b>	<b>14</b>
<b>6</b> <b>Les agents intelligents</b>	<b>15</b>
6.1    Intelligence artificielle	15
6.2    Définition	15
6.3    Processus	16
Figure 1 : Processus de fonctionnement d'un agent intelligent (Bouzy, n.d.).	17
6.4    Niveaux d'intelligence des AI	17
6.5    Caractéristiques d'un agent intelligent	18
6.5.1    Autonomie	18
6.5.2    Intelligence	18
6.5.3    Opérations asynchrones	18
6.5.4    Communication	18
6.5.5    Collaboration	19
6.5.6    Mobilité	19
6.5.7    Personnalisable	19
6.5.8    Auto-adaptatif et réactif	19
6.5.9    Auto-apprenant	19
6.5.10    Cohérence	19
6.5.11    Vérité	19
6.6    Types d'agents intelligents	19
6.6.1    Agents collaboratifs	19
6.6.2    Agents d'interface	20
Figure 2 : Fonctionnement d'un agent d'interface (P. Maes, 1994).	20
6.6.3    Agents mobiles	21
6.6.4    Agents Internet et agents de recherche d'informations	21
Figure 3 : Principe de fonctionnement d'un agent Internet. (Baldr, 1997).	22
6.7    Domaine d'application	22
6.7.1    Automobile	23
6.7.2    Robotique à domicile	23
6.7.3    Santé	23
6.7.4    Education	23
6.8    Bénéfices immédiats	23
<b>7</b> <b>Chatbots</b>	<b>24</b>
7.1    Définition	24
7.2    NLP	24
7.3    Opportunités	25
Figure 4 : Courbes de progression de l'utilisation des applications de messagerie et des réseaux sociaux (Schlicht, 2016).	25

Figure 5 : Utilisation des plateformes de messagerie (Ondrisek, 2016). _____	26
7.4 Utilisation _____	26
7.5 Typologie _____	27
Figure 6 : Fonctionnement de l'application Chatbot (GRANET, 2015). _____	28
7.6 Tendance _____	28
7.7 Besoins Utilisateurs _____	30
Figure 8 : Les activités les plus pratiquées sur internet pendant 3 mois (Insee, 2014). _____	31
Figure 7 : Utilisation d'internet selon la catégorie socioprofessionnelle (Insee, 2014). _____	31
7.8 Limites des chatbots _____	32
<b>8 Développement d'un chatbot</b> _____	<b>33</b>
8.1 Best practices _____	34
8.1.1 Choisir des cas d'utilisation ou domaine d'application _____	34
8.1.2 Définir une personnalité pour le chatbot _____	34
8.1.3 Ecrire les scénarios _____	34
8.2 Outils de développement _____	36
8.2.1 Microsoft Bot Framework _____	36
8.2.2 IBM Watson Developers Cloud – Service Conversation _____	36
8.2.3 Chatfuel _____	36
8.2.4 API.AI (Google) _____	37
8.2.5 Architecture d'un chatbot. _____	38
Figure 9 : Architecture d'un chatbot (Google, 2017b)._____	38
Figure 10 : Architecture d'un chatbot de IBM (IBM, 2016)._____	39
<b>9 Comparaison des outils de développement</b> _____	<b>40</b>
9.1 Coûts _____	40
9.2 Chats textuels et vocaux _____	41
9.3 Langues supportées par les Chatbots _____	42
9.4 Domaines _____	42
Figure 11 : Domaines sur API.AI (API.AI, 2016). _____	43
9.5 Fonctionnalités _____	44
Figure 12 : flux de dialogue IBM (Zheng, 2016)._____	44
9.6 SDK _____	45
9.7 Plateformes _____	45
<b>10 Conclusion et recommandation</b> _____	<b>46</b>
<b>Partie pratique</b> _____	<b>48</b>
<b>1 Introduction</b> _____	<b>50</b>
<b>2 Analyse des besoins</b> _____	<b>50</b>
2.1 Cas d'utilisation _____	52
Figure 13 : Diagramme des cas d'utilisation de la solution choisie, le chatbot. _____	53
<b>3 Analyse du chatbot</b> _____	<b>54</b>
3.1 Bonnes pratiques _____	54
3.1.1 Le domaine d'application ou d'activité de notre chabot _____	54
3.1.2 Définir une personnalité pour le chatbot _____	54
3.1.3 Ecrire les scénarios _____	55
<b>4 Choix des technologies</b> _____	<b>62</b>
4.1 Plateforme de développement _____	63
Figure 14 : Liste des agents préconstruits sur API.AI (API.AI, 2016) _____	63
4.2 Plateforme d'intégration _____	63

4.3	Persistance	64
<b>5</b>	<b>Architecture</b>	<b>65</b>
	Figure 15 : Architecture du chatbot Gestarcbot (API.AI, 2016)	65
5.1	La plateforme API.AI	66
	Figure 16 : Processus de reconnaissance des requêtes des utilisateurs sur API.AI (Quora, 2017)	66
5.2	Le client (Facebook Messenger)	66
5.3	Webhook et la Base de données	67
	Figure 17 : Représentation de l'architecture du webhook	67
<b>6</b>	<b>Structure de la base de données</b>	<b>68</b>
	Figure 18 : Modèles logique et relationnel de la structure de la base de données	69
	Figure 19 : Tables EVENEMENTS et ETUDIANTS	70
<b>7</b>	<b>Mécanisme du chatbot Gestarcbot</b>	<b>71</b>
7.1	Interaction entre les composants de l'architecture	71
7.2	Logique de Gestarcbot	72
	Figure 20 : Processus d'interaction entre les utilisateurs et le chatbot Gestarcbot (Quora, 2017)	72
	Figure 21 : Schéma de la logique de Gestarcbot	73
<b>8</b>	<b>Implémentation du chatbot</b>	<b>75</b>
8.1	L'agent intelligent Gestarcbot avec API.AI	75
8.1.1	Intents	75
	Figure 22 : Représentation graphique des intents de la logique de Gestarcbot	76
	Figure 23 : Sous-processus de la tâche : Solliciter le chatbot	77
	Figure 24 : Sous-processus de la tâche : Choix du service	78
	Figure 25 : Sous-processus de la tâche : Choix de l'évènement	78
	Figure 26 : Sous-processus de la tâche : Aider l'utilisateur à trouver l'évènement	79
	Figure 27 : Sous-processus de la tâche : Information souhaitée	79
	Figure 28 : Sous-processus de la tâche : Identification de l'utilisateur	80
	Figure 29 : Sous-processus de la tâche : Réservation des tickets	81
	Figure 30 : Sous-processus de la tâche : Enregistrement de la demande d'inscription	82
	Figure 31 : Sous-processus de la tâche : Remerciements & salutations	82
8.1.2	Contexts	83
8.1.3	Actions	83
8.1.4	Entities	84
8.1.5	Paramétrage de Gestarcbot	84
8.2	Développement du webhook	85
	Figure 32 : Tableau du contenu de la réponse du webhook, selon les contraintes de API.AI (API.AI, 2016)	85
8.2.1	Règles métiers	86
8.2.2	Projet	87
	Figure 33 : Liens entre le webhook et la plateforme API.AI	87
	Figure 34 : Représentation du projet java	88
<b>9</b>	<b>Tests</b>	<b>89</b>
9.1	Jeux de tests et résultats	89
9.1.1	Jeux de tests	89
9.1.2	Résultats des tests	90
	Figure 35 : Tests pour la demande de renseignements, essai 1	90
	Figure 36 : Tests pour la demande de renseignements, essai 2	91
	Figure 37 : Tests pour la réservation de ticket	92
	Figure 38 : Tests pour devenir membre de la Gest'arc, essai 3	93

9.2	Améliorations	94
9.2.1	Optimisation de la logique de Gestarcbot.	94
	Figure 39 : Schéma supplémentaire pour la logique de Gestarcbot	95
9.2.2	Optimisation et création de nouveaux intents	96
<b>10</b>	<b>Présentation du produit</b>	<b>96</b>
10.1	Fonctionnalités	96
10.2	Fichiers	97
	Figure 40 : Représentation du dossier du projet du servlet java pour le webhook	97
	Figure 41 : Représentation du dossier du projet Gestarcbot	98
<b>11</b>	<b>Versioning</b>	<b>98</b>
<b>12</b>	<b>Conclusion</b>	<b>99</b>
12.1	Déroulement du travail	99
12.2	Problèmes rencontrés	100
12.3	Solutions apportées	100
	Figure 43 : Présentation des fichiers de stockage de données dans le projet java	102
	Figure 42 : Nouvelle architecture suite à la solution apportée pour le problème de connexion à la base de données	102
12.4	Mon avis personnel	103
<b>Partie administrative</b>		<b>106</b>
<b>11</b>	<b>Introduction</b>	<b>2</b>
11.1	Présentation du sujet	2
11.2	Problématique	3
	Figure 44 : Courbes de progression de l'utilisation des applications de messagerie et des réseaux sociaux (Schlicht, 2016).	3
<b>12</b>	<b>Proposition de solution</b>	<b>4</b>
12.1	Solution	4
	Figure 45 : Fonctionnement de l'application Chatbot (GRANET, 2015).	5
12.2	Technologies utilisées	5
<b>13</b>	<b>Démarche projetée</b>	<b>6</b>
13.1	Méthodologie utilisée	6
13.2	Les différentes étapes de la mise en œuvre du chatbot	6
13.2.1	Choisir des cas d'utilisation ou domaine d'application	6
13.2.2	Définir une personnalité pour le chatbot	6
13.2.3	Ecrire les scénarios	7
<b>14</b>	<b>Livrables</b>	<b>8</b>
<b>15</b>	<b>Objectifs</b>	<b>9</b>
<b>16</b>	<b>Planification</b>	<b>9</b>
	Figure 46 : Product backlog	10
<b>17</b>	<b>Authentification</b>	<b>12</b>
<b>Bibliographie</b>		<b>14</b>
<b>18</b>	<b>Bibliography</b>	<b>16</b>
<b>Annexes</b>		<b>20</b>

<b>19 Annexes</b>	<b>22</b>
19.1 Annexe 1 : Outils de développement (Smooch, Pandorabot)	22
19.1.1 Smooch	22
19.1.2 Pandorabot	22
19.2 Annexe 2 : Architecture pour la mise en œuvre d'un chatbot construit avec l'outil Microsoft bot framework	23
Figure 47 : Architecture du chatbot de Microsoft (Zientara, 2016).	23
19.3 Annexe 3 : SDK et technologies	24
12.5 Annexe 5 : Diagrammes	26
12.5.1 Diagramme de séquence pour la demande d'information sur un évènement	26
12.5.2 Diagramme de séquence pour la demande d'inscription comme membre	27
12.6 Annexe 6 : Contexts du chatbot (Gestarcbot)	28
12.6.1 Contexts du processus « acquérir une information sur un évènement »	28
12.6.2 Contexts du processus « Réserver un ticket pour un évènement »	29
12.6.3 Contexts du processus « Enregistrer une demande d'inscription pour devenir membre de la Gest'arc »	30
12.7 Annexe 7 : Création et paramétrage des intents	0
12.7.1 Valeurs de paramétrage de l'agent, tableau 1	0
12.7.2 Valeurs de paramétrage de l'agent, tableau 2	2
12.7.3 Valeurs de paramétrage de l'agent, tableau 3	4
12.7.4 Valeurs de paramétrage de l'agent, tableau 4	6
12.8 Annexe 8 : Nouvelles étapes de la logique de Gestarcbot	0
12.8.1 Demande d'information	0
12.8.2 Demande de réservation de ticket pour un évènement	1



# Introduction

**HEG Arc - Haute école Arc – Gestion**

**Mention :** Travail de Bachelor d'informaticien de gestion

**Module :** Thèse de bachelor

**Sujet :** Exploration dans le domaine des agents intelligents et des chatbots

**Directeur de travail :** Frischer Boris

**Auteur :** Klett Boris

**Cycle d'étude :** 2013 - 2017

**Date de restitution :** Septembre 2017



# 1 Introduction

---

De nos jours, pour perdurer dans un environnement de plus en plus hostile, une entreprise se doit d'être performante. La performance se passe aussi au travers des relations avec les clients pour une entreprise commerciale ou des membres pour une association. Celle-ci doit être au plus proche de ses clients, pour améliorer constamment leurs relations. Une fois que ce point fondamental est acquis, ces échanges peuvent être automatisés totalement ou partiellement à l'aide d'un système informatique tels que des agents intelligents et des chatbots, dans le but d'améliorer le temps de réaction de l'entreprise.

Un agent intelligent, est un programme informatique capable d'interagir, guider et informer des utilisateurs humains ou informatisés. Il est capable de faire preuve d'intelligence en réalisant un certain nombre de tâches de manière autonome. Dotés d'une intelligence artificielle, on distingue plusieurs types d'agents intelligents tels que : des agents collaboratifs, d'interface, mobiles, d'internet ou encore de recherche d'information (*Baldit, 1997*). De nos jours, cette technologie est utilisée dans beaucoup de domaines parmi lesquels on peut citer : l'automobile, la robotique, la santé, l'éducation, l'internet des objets<sup>1</sup>, et bien d'autres encore (*Wisdom, 2005*). Quant au chatbot, c'est un programme informatique, une typologie d'agent intelligent, capable d'interagir, communiquer vocalement ou textuellement avec un utilisateur, via des messageries, sites web ou encore des machines robotisées.

Pour mettre en place un système automatisé du type chatbot, les entreprises disposent actuellement de différentes solutions. Dans un premier temps, elles peuvent acheter une solution existante et l'installer dans leur système de communication informatisé. Dans cette solution, on a par exemple VirtualSpirits<sup>2</sup> qui est un chatbot en ligne. En effet, une entreprise peut acheter et ajouter ce chatbot dans son système pour non seulement communiquer avec ses clients, trouver de nouveaux prospects, aider des visiteurs, répondre instantanément et efficacement aux requêtes des clients et des prospects mais également pour relancer des clients suite à des commandes (*VirtualSpirits, 2017*). Les entreprises ont également la possibilité de mandater une entreprise spécialisée dans le développement de chatbot, pour développer leur chatbot comme l'entreprise Thechatbotfactory par exemple (*THECHATBOTFACTORY, n.d.*). Finalement, les entreprises peuvent développer elles-mêmes leur chatbot grâce à différents outils de développement actuellement disponibles sur le marché, parmi lesquels on peut citer : API.AI, Chatfuel, Microsoft Bot Framework ou encore IBM Watson.

Aujourd'hui, l'utilisation des chatbots par les entreprises est encore en phase de croissance. En effet, de plus en plus d'entreprises mettent en place un chatbot pour communiquer avec leurs clients et leurs prospects car suivant plusieurs statistiques, la population mondiale utilise de plus en plus les messageries instantanées pour

---

<sup>1</sup> La traduction française de l'expression anglaise *internet of things* « IOT ».

<sup>2</sup> Source : <https://www.virtualspirits.com/fr/>

communiquer, ce qui représente une opportunité pour cette technologie en pleine expansion.([Insee, 2014](#))

Cette présente thèse de Bachelor a pour objectif d'explorer le domaine des agents intelligents et des chatbots, faire des scénarios pour les flux de conversation entre un chatbots et des utilisateurs, implémenter un chatbot pour résoudre une problématique de gestion, tester et faire des recommandations sur le développement et l'utilisation des chatbots en fonction de leur contexte.

## 2 Présentation du document

---

Ce présent document s'adresse à toute personne ou entreprise s'intéressant à l'utilisation et au développement des chatbots, de manière personnalisée à l'aide d'un outil de développement.

Ce document est organisé en sept parties distinctes qui sont l'introduction, le résumé, la partie théorique, la partie pratique, la bibliographie, la partie administrative puis enfin les annexes.

La présente section, *Introduction*, introduit le sujet, pose le cadre général, le but et les objectifs de ce travail. *La partie théorique* comme son nom l'indique, pose les fondements théoriques qui servent de base à la partie pratique. Cette partie comprend des éléments théoriques relatifs aux différents frameworks de développement, au développement et à l'utilisation des services web (webhook) pour la persistence, aux différentes étapes du développement d'un chatbot et à l'utilisation de celui-ci dans un contexte bien défini. *La partie pratique*, présente le cœur du travail réalisé notamment avec les résolutions des différentes problématiques rencontrées. Elle comprend les détails quant à la réalisation du chatbot, de l'analyse des besoins au déploiement de l'agent. La quatrième partie, *Bibliographie*, comprend la liste des sources bibliographiques employées pour la réalisation de ce travail. Elle est suivie d'une partie administrative qui comprend les rapports hebdomadaires de travail. Pour finir, la dernière section contient les annexes.

## 3 Objectif du travail

---

Identifier un problème de gestion dans une organisation et proposer une solution par le biais d'un chatbot. L'objectif est alors de réaliser une application autour des agents intelligents et des Chatbots, qui permettra de résoudre cette problématique dont une entreprise ou une association communautaire serait susceptible de faire face. Le travail vise également à ce que l'application soit utilisée, jugée et testée par des utilisateurs, dans le but de me permettre de faire des recommandations sur le développement et l'utilisation de Chatbots en fonction d'un contexte précis.

## Exploration dans le domaine des agents intelligents et des chatbots

Candidat

Boris Klett

Professeur

Boris Frischer

### ■ Description

Depuis quelques années maintenant, le domaine des agents intelligents est un domaine fascinant et en pleine essor. En effet, on remarque que cette technologie est de plus en plus utilisée, car elle présente de nombreux avantages notamment pour les entreprises. On parle ici d'accroître leur productivité ou encore leur réactivité vis-à-vis du marché ou plus intéressant de leur clientèle. Les agents intelligents particulièrement les chatbots rencontrent actuellement beaucoup de succès dans des domaines plutôt diversifiés, tels que l'éducation, la médecine, le commerce et même dans les loisirs comme les jeux vidéo par exemple. Bien qu'il existe aujourd'hui plus d'outils et de moyens pour les développer par rapport à quelques années en arrière, construire un chatbot peut s'avérer être très vite compliqué notamment lors de la scénarisation des interactions.

### ■ Déroulement

Dans le but de mettre en place un chatbot qui permettra de pallier à une problématique de gestion, j'ai alors réalisé un travail en deux étapes principales. Dans un premier temps, les recherches que j'ai effectuées sur plusieurs sources scientifiques, m'ont permises de comprendre cette technologie, ses avantages, ses limites mais surtout comment développer un chatbot pour pouvoir l'utiliser. Par la suite, j'ai exploré et tester plusieurs outils de développement, ce qui m'a permis de faire un choix pour pouvoir construire un chatbot opérationnel qui permet de régler une problématique dont une entreprise pourrait faire face.

### ■ Résultats

Grâce aux recherches que j'ai effectuées, j'ai pu démontrer tout l'engouement porté aux agents intelligents. J'ai également mis en exergue le fait que, malgré cet emballement, beaucoup de gens restent quand même sceptiques quant à l'utilisation de cette technologie autonome, car ils craignent qu'elle prenne peu à peu la place des humains dans la société.

De plus, j'ai réalisé un chatbot pour la page Facebook de l'association des Etudiants, la Gest'arc. Pour le faire, je suis partie de l'analyse des besoins des futures utilisateurs à la rédaction du code, en passant par la scénarisation des interactions et la réalisation des différents tests de performance.

### ■ Perspectives

Le chatbot que j'ai développé dispose de quelques fonctionnalités. Cependant, il faudrait ajouter davantage de fonctionnalités à ce chatbot pour lui permettre d'être plus efficace, notamment lui donner la possibilité d'encaisser directement les entrées aux évènements payants organisés par la Gest'arc permettant ainsi aux membres l'association de se consacrer à d'autre tâche de la Gest'arc.



## Partie théorique

**HEG Arc - Haute école Arc - Gestion**

**Mention :** Travail de Bachelor d'informaticien de gestion

**Module :** 656-1.1 Thèse de bachelor

**Sujet :** Exploration dans le domaine des agents intelligents et des chatbots

**Directeur de travail :** Frischer Boris

**Auteur :** Klett Boris

**Cycle d'étude :** 2013 - 2017

**Date de restitution :** Septembre 2017



## 4 Introduction

---

Cette partie a pour objectif de synthétiser les lectures et recherches effectuées afin de réaliser ce travail de Bachelor. Elle précède la partie pratique qui expliquera la mise en œuvre d'une solution.

La présente partie traitera de toute recherche effectuée dans le cadre de mon travail. Elle aura pour but de comprendre les bonnes pratiques dans le domaine des agents intelligents et principalement des chatbots. Ce chapitre constitue donc la base théorique du projet, il s'agit dans un premier temps, de comprendre ce que sont les agents intelligents, leur fonctionnement, leur intelligence artificielle, leurs bénéfices et bien évidemment leurs limites. Nous verrons que les agents intelligents peuvent très bien être appliqués dans plusieurs domaines et contextes selon les besoins des utilisateurs grâce à leurs typologies variées.

Deuxièmement, nous aborderons le fonctionnement, les avantages et les opportunités offertes par les chatbots ainsi que les méthodes dites « Best practices » pour pouvoir les développer. Ces best practices serviront par la suite de guide durant la réalisation pratique du projet.

Le but de cette partie du document est de fournir un cadre théorique pour la mise en œuvre d'une solution de type agent intelligent, adaptée aux besoins métier de l'association des étudiants de la HEG-arc, la Gest'arc.

Le prochain chapitre permet quant à lui, de mettre en exergue la problématique dont l'implémentation d'un agent intelligent notamment d'un chatbot, pourrait être une solution non négligeable.

## 5 Problématique

---

Aujourd'hui, de plus en plus d'entreprises allant des grandes multinationales aux associations communautaires, sociales ou autres ont bien compris que la proximité avec leur client ou prospect est très importante. En effet, il est maintenant indispensable voir même vital de mettre en place des stratégies de communication par messageries instantanées avec des clients ou affiliés pour mieux les atteindre et surtout pour faire face à l'évolution de l'environnement technologique.

Mais, quand bien même ces instances mettent en place de tel dispositif de communication, faut-il encore que les requêtes des principaux intéressés soient traitées à la minute voir à la seconde. Car, ces personnes aimeraient le plus souvent se sentir pris en charge rapidement, une réactivité qui peut donc satisfaire le client. Hors, par manque de ressources, cela n'est pas toujours le cas, ce qui peut même causer dans certaines situations, la perte de clients, voir une mauvaise image pour l'entreprise ou l'association si la situation est de manière fréquente.

L'objectif principal de ce projet du travail de bachelor étant de réaliser une application autour des chatbots pour résoudre un problème de gestion, comme ceux décrit plus haut, nous avons donc choisi de développer un chatbot pour la Gest'arc<sup>3</sup>. Le chatbot discutera donc avec des étudiants sur la plateforme de Facebook Messenger.

Dans notre situation, on cherche à mettre en place une technologie de type agent intelligent et chatbot, à disposition d'une association pour que celle-ci puisse profiter pleinement des avantages qu'offre cette technologie. Etant donné que la Gest'arc est une petite association avec peu de membres, un chatbot permettrait aux étudiants d'avoir des réponses dont ils ont besoin le plus rapidement et efficacement possible.

Dans la prochaine partie, nous aborderons les agents intelligents.

## 6 Les agents intelligents

---

### 6.1 *Intelligence artificielle*

Selon Nils J. Nilsson, « l'intelligence artificielle est cette activité consacrée à faire des machines intelligentes et l'intelligence est cette qualité qui permet à une entité de fonctionner convenablement dans son environnement » ([Stanford University, 2016](#)) sans une intervention extérieur, humaine ou autre. Aujourd'hui, cette technologie est mise en oeuvre dans un grand nombre d'applications, comme par exemple les interfaces utilisateurs, la gestion d'information, la messagerie intelligente, la gestion des réseaux, les jeux vidéo, la réalité augmentée, etc. Si un grand nombre d'agents intelligents utilisent une approche simple basée sur le concept des question-réponses, d'autres sont dotés d'une intelligence artificielle, qui les rend plus autonomes et certainement plus efficaces. Dans la suite de ce chapitre, nous allons d'avantage nous intéresser à la définition même de ce qu'est un agent intelligent.

### 6.2 *Définition*

Pour mieux comprendre ce que représente un agent intelligent, intéressons-nous d'abord de manière brève à ce qu'est un programme informatique. Selon la page internet de Rick Hirst, depuis l'origine de l'informatique, un programme est défini comme un ensemble de commandes effectuant un traitement spécifique sur un ensemble de données déterminées. Il est déclenché par un événement, une demande de l'utilisateur, par exemple, et fournit un résultat en fin d'exécution ([Hirst, 2016](#)). Hors, selon autre un article, un agent intelligent est un programme autonome ne nécessitant aucune intervention humaine – ou presque – pour mener à bien sa mission. En déléguant à un agent intelligent des travaux de prospection ou des tâches répétitives, l'utilisateur peut donc se libérer plus de temps pour se consacrer à d'autres tâches ([Piloter.org, 2016](#)). Un AI possède une certaine capacité d'intelligence, pouvant aller du niveau simple, notamment avec des règles prédéfinies (système basé sur des

---

<sup>3</sup> Association des étudiants de la HEG-arc

questions-réponses), à un niveau plus avancé, dit d'auto-apprentissage ; l'agent est alors doté d'une intelligence artificielle. Les agents sont le plus souvent utilisés comme des assistants pour faire de la recherche d'informations sur Internet et la collecte d'informations personnalisées, car étant autonomes, ils sont très efficaces pour réaliser ce genre de tâches. En effet, pour mentionner un exemple, selon certains critères de recherche demandés par l'utilisateur, ces agents ont la capacité de récupérer des informations correspondantes depuis des réseaux et auprès de services fournisseurs d'information. Cet exemple met en exergue l'avancée des fonctions d'autoapprentissage qui seront mises à profit pour aller au-devant des souhaits de collectes de l'utilisateur. L'AI<sup>4</sup> doit donc pouvoir montrer un certain degré d'autonomie, une certaine capacité sociale et pouvoir combiner les comportements proactifs et réactifs pour ainsi être capable de prendre les devants face à certaines situations.

### 6.3 *Processus*

Après avoir défini brièvement l'agent intelligent, nous pouvons nous intéresser à son mode de fonctionnement. De manière générale, les agents intelligents sont dotés de plusieurs éléments :

- des capteurs qui leur permettent de percevoir leurs environnements,
- des actionneurs qui leur permettent d'agir sur leur environnement et
- d'un ou de plusieurs programmes (selon la nature de l'agent) qui leur permettent d'analyser leur environnement et de réagir de manière adéquate.

---

<sup>4</sup> AI est l'abréviation d'agent intelligent.

Le schéma ci-dessous nous permet d'avoir une approche visuelle et très conceptuelle de la manière dont fonctionne un AI :

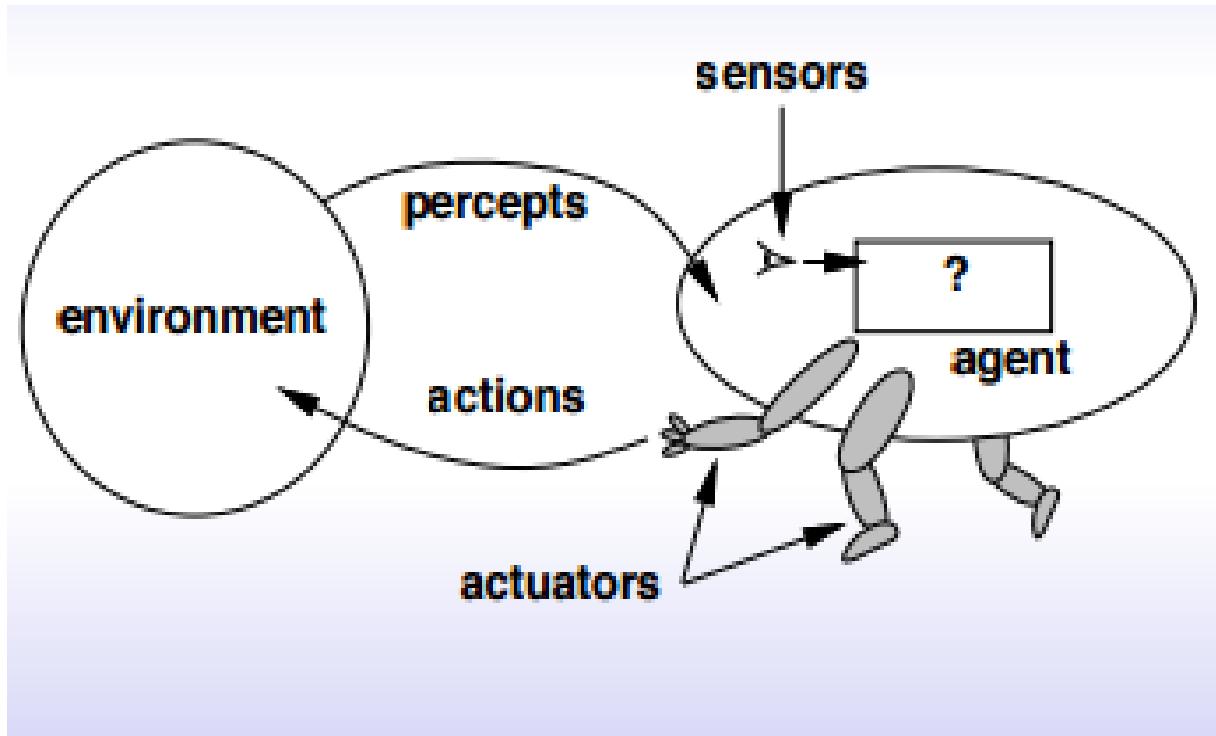


Figure 1 : Processus de fonctionnement d'un agent intelligent (*Bouzy, n.d.*).

Grâce à la figure 1 ci-dessus, nous pouvons comprendre qu'un agent intelligent pourrait être représenté comme un être vivant ayant une capacité d'interaction avec environnement. Mais un agent peut-il réellement être comparé à un être intelligent à part entière tel qu'un humain ? Le point suivant de ce chapitre parcourra les différents niveaux d'intelligence que peuvent avoir les agents intelligents et ainsi essayer de répondre à notre question.

#### 6.4 Niveaux d'intelligence des AI

À mon avis, les agents ne sont pas comparables aux êtres humains, du moins pas encore. L'AI fonctionne de manière autonome, certes, mais seulement dans un cadre programmé par un processus, ou encore pour le propre compte d'un utilisateur, notamment dans le but d'automatiser une tâche. Selon plusieurs sources, notamment un rapport sur l'intelligence artificielle de l'université de Stanford, les agents peuvent ainsi être classés en 4 catégories par rapport à leur degré d'intelligence, allant du plus simple au plus complexe :

1. Les agents de collecte dits « simples » ne prennent aucune décision vis-à-vis de la pertinence des informations collectées.

2. Les agents de collecte un peu plus complexes peuvent appliquer des traitements simples sur les informations collectées, comme certains contrôles en amont d'une action.
3. Les agents plus complexes sont capables de prendre des décisions plus évoluées en accord avec l'utilisateur (notamment la négociation d'une transaction ou la prise automatique de rendez-vous).
4. Les agents les plus complexes communiquent et agissent à plusieurs, grâce à une auto-gérance ([University, 2015](#)).

Il faut noter qu'en plus d'un certain niveau d'intelligence, un AI se caractérise par plusieurs éléments.

## *6.5 Caractéristiques d'un agent intelligent*

En principe, les AIs sont caractérisés par les éléments suivants (tout en sachant que cette liste n'est pas exhaustive) :

### *6.5.1 Autonomie*

Elle est la caractéristique qui démontre qu'un agent peut être capable de décider par lui-même dans certaines situations et qu'il n'y a aucune interaction avec l'utilisateur ([Schleiffer, 2002](#)). En fait, il s'exécute seul selon des objectifs prédéterminés (et préprogrammés) ([Bouzy, n.d.](#)).

### *6.5.2 Intelligence*

Comme nous avons vu précédemment, certains agents sont capables de prendre des décisions pour trouver une solution à un problème sans une intervention humaine (comme par exemple un aspirateur robot qui doit contourner un obstacle). Ces prises de décisions sont raisonnables dans les limites de ses propres capacités informatiques et ses connaissances subjectives.

### *6.5.3 Opérations asynchrones*

Un agent peut exécuter ses tâches de manière asynchrone par rapport à son utilisateur ou à d'autres agents. Cela signifie que les agents peuvent être déclenchés par un événement ou par un instant précis sans forcément démarrer lorsqu'il est activé par l'utilisateur. Un agent situé dans le réseau peut fonctionner de manière totalement asynchrone aux différents utilisateurs par exemple.

### *6.5.4 Communication*

Pendant leurs opérations, les agents peuvent communiquer avec des ressources systèmes diverses ou des utilisateurs. Ces ressources peuvent être locales ou distantes. Il y a beaucoup de ressources auxquelles peuvent accéder les agents, comme par exemple les bases de données, les informations systèmes, etc.

### **6.5.5 Collaboration**

Cette caractéristique indique que le système d'un agent lui permet de collaborer avec d'autres agents ou utilisateurs. En effet, comme le confirme Ralf Schleiffer dans son article, pour réaliser sa ou ses tâches, un agent peut avoir besoin d'une certaine capacité d'interaction « sociale » (*Schleiffer, 2002*) pour ainsi être aidé à réaliser une tâche.

### **6.5.6 Mobilité**

Pour réaliser des tâches spécifiques, certains AIs ont besoin d'une certaine mobilité à travers les réseaux, notamment vers certains sites distants durant un moment donné. Ceci est le cas des agents web ou réseaux par exemple.

### **6.5.7 Personnalisable**

L'AI est adaptable et peut être configuré pour répondre aux besoins précis de l'utilisateur.

### **6.5.8 Auto-adaptatif et réactif**

Ils peuvent prendre certaines initiatives, notamment les agents d'interface, car ils savent ajuster leurs comportements en fonction de la situation, selon les attentes de l'utilisateur, ou même suivant les choix de l'utilisateur, lorsque celui-ci utilise une application où l'agent est installé. Cette caractéristique démontre leur capacité à réagir face aux changements de leur environnement (*Wooldridge & Jennings, 1995*).

### **6.5.9 Auto-apprenant**

Certains agents savent profiter de l'expérience passée pour mieux comprendre les souhaits actuels de l'utilisateur, voir aller au-devant de ses désirs (*Bouzy, n.d.*).

### **6.5.10 Cohérence**

C'est la certitude que les agents n'ont pas d'objectifs contradictoires par rapport à ce qui lui est demandé, dans la limite de ses capacités.

### **6.5.11 Véracité**

C'est la certitude qu'un agent ne communiquera pas sciemment de fausses informations.

## **6.6 Types d'agents intelligents**

À partir de ces différentes caractéristiques, on distingue plusieurs sortes d'AIs. À l'heure actuelle, il existe un très grand nombre de typologies d'AIs. Je me suis penché sur quelques-unes choisies au préalable de par leur présence relativement fréquente et connue :

### **6.6.1 Agents collaboratifs**

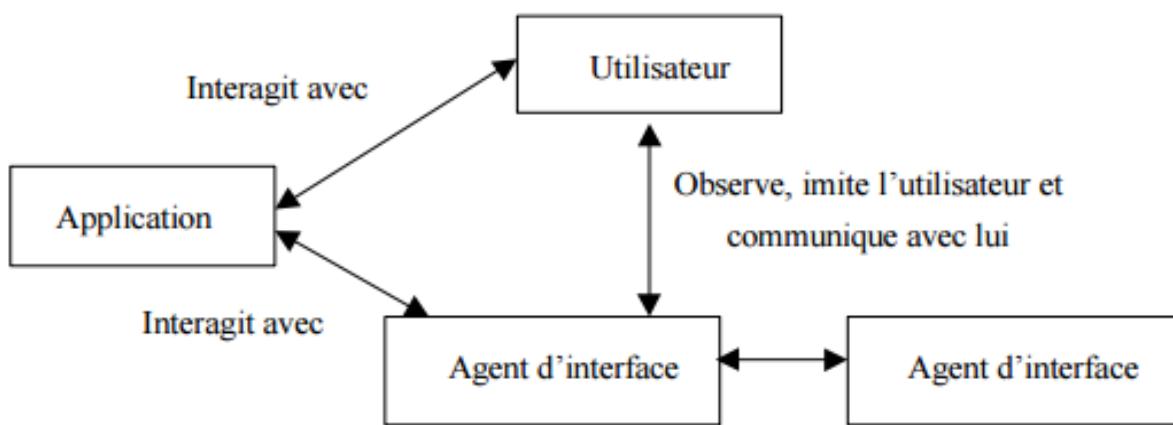
Ce sont des agents capables de collaborer avec d'autres AIs pour réaliser certaines tâches. D'après le rapport de Patrick Baldit, la principale préoccupation de ces

systèmes multi-agent est la coordination de comportements intelligents parmi un ensemble d'Al's (c'est-à-dire la manière dont ils coordonnent leurs connaissances, buts, compétences et planifient conjointement de faire des actions pour résoudre des problèmes). C'est la raison pour laquelle, afin de coordonner un système d'agents collaboratifs, il est nécessaire de mettre en place un protocole de négociation qui est une forme de règlement entre-agents (*Baldit, 1997*). Ces agents sont utilisés dans un grand nombre d'applications, comme les ordinateurs embarqués, l'analyse syntaxique du langage naturel (NL), la planification de transports, etc.

Ils sont utilisés dans l'objectif de construire des systèmes où chaque agent pourra offrir ses fonctionnalités, ce qui permettra d'avoir un système avec un grand nombre de fonctionnalités disponibles. Ceci permettra également de résoudre des problèmes trop complexes pour un seul agent, compte tenu des limitations de ses capacités et de ses ressources, poursuit Baldit dans son rapport.

### 6.6.2 Agents d'interface

Un agent d'interface intervient essentiellement en support et en assistance, notamment en aidant l'utilisateur lors de l'utilisation d'un logiciel ou d'un système d'exploitation. L'agent observe les actions de l'utilisateur, enregistre les



enchaînements et suggère des raccourcis pour accomplir le même type de commande. La figure 2, illustre visuellement le processus décrit.

Le schéma ci-dessus est une représentation des interactions qu'il existe ou peut exister entre les différents agents, les utilisateurs et les applications qui utilisent ces agents<sup>5</sup>.

Figure 2 : Fonctionnement d'un agent d'interface (*P. Maes, 1994*).

<sup>5</sup> À noter qu'ici nous nous sommes limités aux agents d'interface.

Les avantages principaux de l'utilisation de ces agents d'interface sont de trois ordres :

- Adaptation automatique aux préférences et aux habitudes de l'utilisateur,
- Partage du savoir-faire d'une communauté d'utilisateurs,
- Simplification des procédures techniques d'accès aux réseaux, diminuant les efforts de développement d'applications.

### *6.6.3 Agents mobiles*

Les agents mobiles sont des logiciels qui ont la capacité d'interagir avec les serveurs pour lancer des programmes ou récupérer des informations pour leurs utilisateurs ([Baldit, 1997](#)). Ils sont concentrés principalement sur les grands réseaux informatiques qui offrent un grand nombre de services sophistiqués, notamment l'Internet. Les messageries intelligentes et les Chatbots (voir le chapitre Chatbots) en sont des exemples. Les demandes de l'utilisateur sont multiples, pouvant aller de la réservation de billets d'avion jusqu'à la gestion de réseaux de télécommunication. Dans ce contexte-ci, les avantages principaux de ces AIs sont :

- Réduire le temps de communication. En effet, le téléchargement des informations à traiter conduit souvent à une augmentation de l'encombrement des réseaux informatiques. Les agents mobiles sont donc chargés d'effectuer ces traitements à distance afin de ne transférer que les informations utiles à l'utilisateur.
- Pallier le manque de ressources locales. Les processus de stockage et de traitement sur les machines locales peuvent être limités et doivent donc faire appel temporairement aux autres machines du réseau en utilisant des agents mobiles.
- Faciliter la coordination et augmenter la puissance de calcul. Il apparaît plus simple de coordonner des requêtes indépendantes sur des ordinateurs différents et de conserver le résultat en local.
- Exécution asynchrone. Par ce moyen, on peut lancer un agent mobile, se déconnecter du réseau et recevoir le résultat par mail, afin de diminuer les coûts de connexion à un fournisseur d'accès Internet par exemple.

### *6.6.4 Agents Internet et agents de recherche d'informations*

Les agents de recherche d'informations ont pour objectif principal de réduire, mais certainement pas d'éliminer le problème de surcharge d'informations qui apparaît comme une problématique de cette fin de siècle. Ainsi, une des solutions proposées dans ce domaine est de créer des interfaces de recherche conviviales afin que la recherche d'informations devienne aussi naturelle pour l'utilisateur que téléphoner, ou lire le journal, et ce de manière très ciblée et rapide. Ainsi, ces agents ont la capacité de savoir où chercher l'information, la manière de trouver cette information et comment la rapatrier vers l'utilisateur. Les agents Internet fonctionnent comme suit : ils se connectent pour le compte de l'utilisateur simultanément à différents moteurs de recherche (Altavista, Lycos, Yahoo, Google, etc.) grâce à Internet, traitent les résultats fournis et les restituent à l'utilisateur sous différentes formes, comme le montre la figure 3.

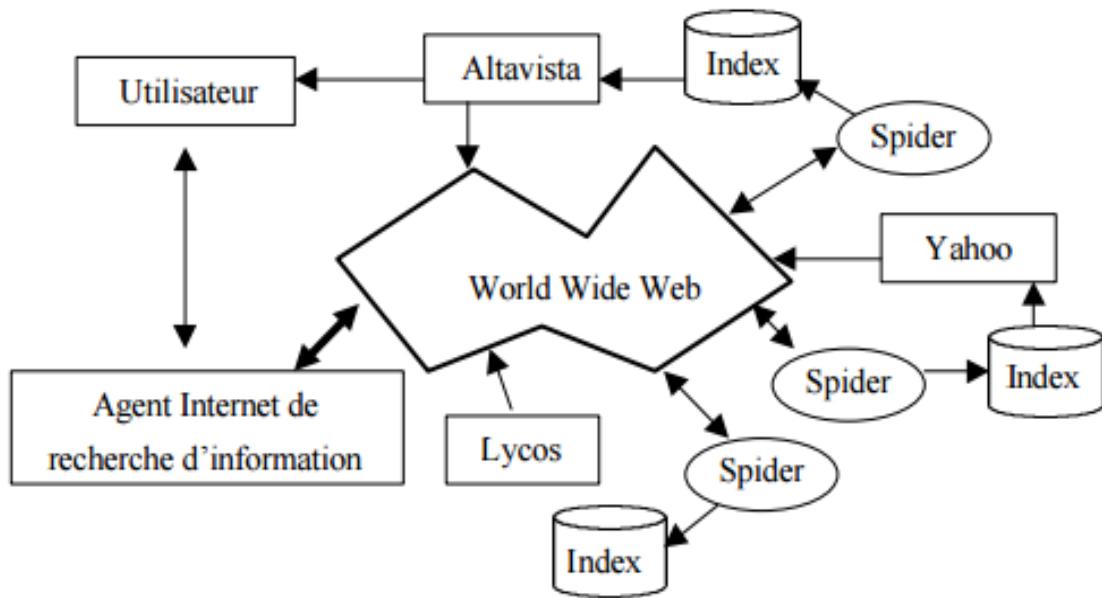


Figure 3 : Principe de fonctionnement d'un agent Internet. (*Baldit, 1997*).

Ces agents ont les avantages suivants :

- Suppression des adresses en double,
- Vérification de la présence des pages,
- Rapatriement des pages au niveau local pour la relecture,
- Indexation locale des pages pour une prochaine recherche éventuelle,
- Surveillance des modifications des pages sur les sites ou encore
- Mise en évidence des nouvelles pages par rapport à la dernière recherche (*Baldit, 1997*).

On peut donc en conclure que les agents intelligents sont très variés.

## 6.7 Domaine d'application

Grâce à leurs typologies variées, les agents intelligents peuvent être intégrés dans d'autres programmes comme les jeux, les messageries, les sites web de e-commerce, ou peuvent être utilisés pour construire des robots autonomes. Mais comme nous

avons pu le constater tout au long de ce chapitre, un AI est utile dans beaucoup de domaines, parmi lesquels on peut citer :

#### 6.7.1 *Automobile*

Certaines technologies clés employées par des constructeurs automobiles ont adopté des AIs pour gérer la vitesse par exemple. Des études sont actuellement menées pour la conception de véhicules totalement autonomes.

#### 6.7.2 *Robotique à domicile*

Beaucoup de nos robots à domicile sont dotés de systèmes intelligents ou agents intelligents, dont les aspirateurs automatiques font office d'exemple.

#### 6.7.3 *Santé*

Grâce aux progrès technologiques, il existe actuellement des robots chirurgicaux conçus pour aider lors des opérations chirurgicales. Certaines applications dotées d'une intelligence artificielle pourraient également améliorer le bilan de santé et la qualité de vie pour des millions de personnes dans les années à venir.

#### 6.7.4 *Education*

J. Wisdom avance dans sa thèse que, l'éducation de qualité exige toujours plus d'engagement par les professeurs (humains) et l'utilisation des AIs permettra d'améliorer le rendement des professeurs tout en les assistant dans leurs tâches d'enseignement, ce qui améliorera le niveau d'éducation, voir les conditions de travail des professeurs ([Wisdom, 2005](#)). Si les AIs sont autant variés et peuvent être utilisés dans autant de domaines, cela signifie tout simplement que leur utilisation représente un intérêt certain – ou du moins croissant. Nous allons en effet voir les bénéfices que représente l'utilisation des AIs au prochain chapitre.

### 6.8 *Bénéfices immédiats*

À mon avis, l'intégration des AIs dans nos systèmes informatiques, robots et autres peut apporter beaucoup d'avantages. Peuvent être cités ceux que nous avons parcourus plus haut (voir : Agents internet et agents de recherche d'informations, agents mobiles et agents d'interfaces). Comme nous avons pu le remarquer, les agents intelligents peuvent permettre aux utilisateurs d'envisager un certain gain de temps. En effet, en laissant les agents intelligents s'occuper de certaines tâches, comme des tâches répétitives ou encore des tâches de recherche d'information par exemple, cela permet aux utilisateurs (employés ou particuliers) d'avoir plus de temps à consacrer à d'autres tâches ou activités. De plus, l'information trouvées sur internet par un AI est beaucoup plus pertinente. En effet, un agent de recherche sur le web bien paramétré trouvera plus facilement et plus rapidement de qu'un humain des informations pertinentes sur la toile grâce à sa puissance de traitement. Les AIs permettent d'améliorer les conditions de vie des gens, ainsi que l'accès à l'information, facilitent la recherche et augmentent la productivité du travail intellectuel, ajoute J. Wisdom. De nos jours, nous pouvons observer ces avantages grâce aux robots de cuisine ou de ménage dans les foyers ; en effet, ceux-ci réalisent les tâches ménagères, dégageant ainsi du temps libre pour les personnes pour d'autres tâches.

Dans les lieux publics, les assistants automatiques permettent quant à eux d'aiguiller et de renseigner les gens. De surcroît, la plupart des personnes interrogées pensent que l'efficacité des moteurs de recherche augmente ([Wisdom, 2005](#)).

## 7 Chatbots

---

### 7.1 Définition

Selon Niko Bonatsos, « Le 90 % de notre temps sur le portable est pour lire notre courrier électronique et aller sur des plates-formes de messagerie ! » ([Schlicht, 2016](#)). Selon plusieurs sources, dont la page web de Bertrand Bathelot, les Chatbots font partie de la catégorie des agents intelligents. Ce sont des logiciels robots qui peuvent dialoguer avec des individus ou des consommateurs grâce à des plateformes de messagerie comme Facebook Messenger, Kik ou encore Slack. En effet, pour dialoguer, ces agents intelligents utilisaient à l'origine des bibliothèques de questions-réponses, mais aujourd'hui, grâce aux progrès technologiques réalisés sur l'intelligence artificielle, on retrouve de plus en plus de Chatbots dotés d'un mécanisme appelé le « Machine learning ». Ce mécanisme permet aux Chatbots d'avoir la capacité d'apprendre ([Bathelot, 2016](#)), ce qui leur permet d'entretenir une conversation le plus naturellement possible en imitant une conversation humaine et ainsi proposer des services à l'utilisateur, conclut Adeline Granet dans son rapport ([GRANET, 2015](#)). Comme nous avons vu, pour être dotés d'une forme d'intelligence, les Chatbots peuvent utiliser soit une approche dite de questions-réponses ; cette approche fonctionne grâce à des questions prédéfinies que les utilisateurs seraient susceptibles de poser et les réponses que le Chatbot devra répondre. C'est une technique de base, simple et très limitée, car il faudrait connaître d'avance toutes les requêtes des utilisateurs, ce qui est bien entendu impossible. L'utilisation de l'intelligence artificielle est ainsi une solution possible pour pallier à ce problème (voir : le point Intelligence artificielle dans le chapitre « les agents intelligents ») d'une manière générale et du Natural Language Processing (NLP) en particulier, dont le fonctionnement est expliqué dans le chapitre suivant.

### 7.2 NLP

Selon le blogueur Thibaut Cantet, « on ne peut pas parler de bots sans introduire la notion de Natural Language Processing (NLP). C'est une discipline à la frontière de la linguistique, de l'informatique et de l'intelligence artificielle, qui concerne l'application de programmes et techniques informatiques pour reproduire tous les aspects du langage humain » ([Cantet, 2016a](#)). En utilisant NLP, les développeurs peuvent mieux organiser et structurer l'intelligence pour réaliser des tâches comme la traduction, la reconnaissance vocale, l'analyse de sentiment et bien d'autres encore. Les algorithmes NLP sont typiquement basés sur les algorithmes du Machine Learning pour apprendre automatiquement les différentes règles mises en place, en analysant un ensemble d'exemples et en faisant une inférence statistique. En général, plus il y a des données, plus le model apparaît efficient ([Kiser, 2016](#)). Ces dernières années,

grâce à ces différentes techniques d'intelligence et d'autoapprentissage, qui sont utilisés par des programmes comme les assistants virtuels tels que Siri, Evi, la S-voix et d'autres encore, les Chatbots ont gagné en renommé ([Institute, n.d.](#)).

### 7.3 Opportunités

Maintenant que nous savons ce que sont des Chatbots, quelles opportunités représentent-ils ? « *Il faut savoir que les gens maintenant passent plus de temps sur les applications de messageries que sur les médias sociaux et c'est un tournant énorme. Les applications de messageries sont les plateformes du futur et les Chatbots permettront par ce biais d'accéder à toute sorte de services* », comme l'explique l'auteur Peter Rojas, entrepreneur in Residence at Betaworks ([Schlicht, 2016](#)). Le graphe ci-dessous montre à quel point l'utilisation des platesformes de messagerie connaissent une plus forte croissance que les autres applications sociales. Par exemple, entre le 4<sup>ème</sup> trimestre de l'année 2013 et le 3<sup>ème</sup> de l'année 2015, il y a eu une plus grande progression de la courbe pour les applications de messageries par rapport aux applications sociales. Ceci représente donc une énorme opportunité pour les logiciels conversationnels automatisés ou Chatbots.

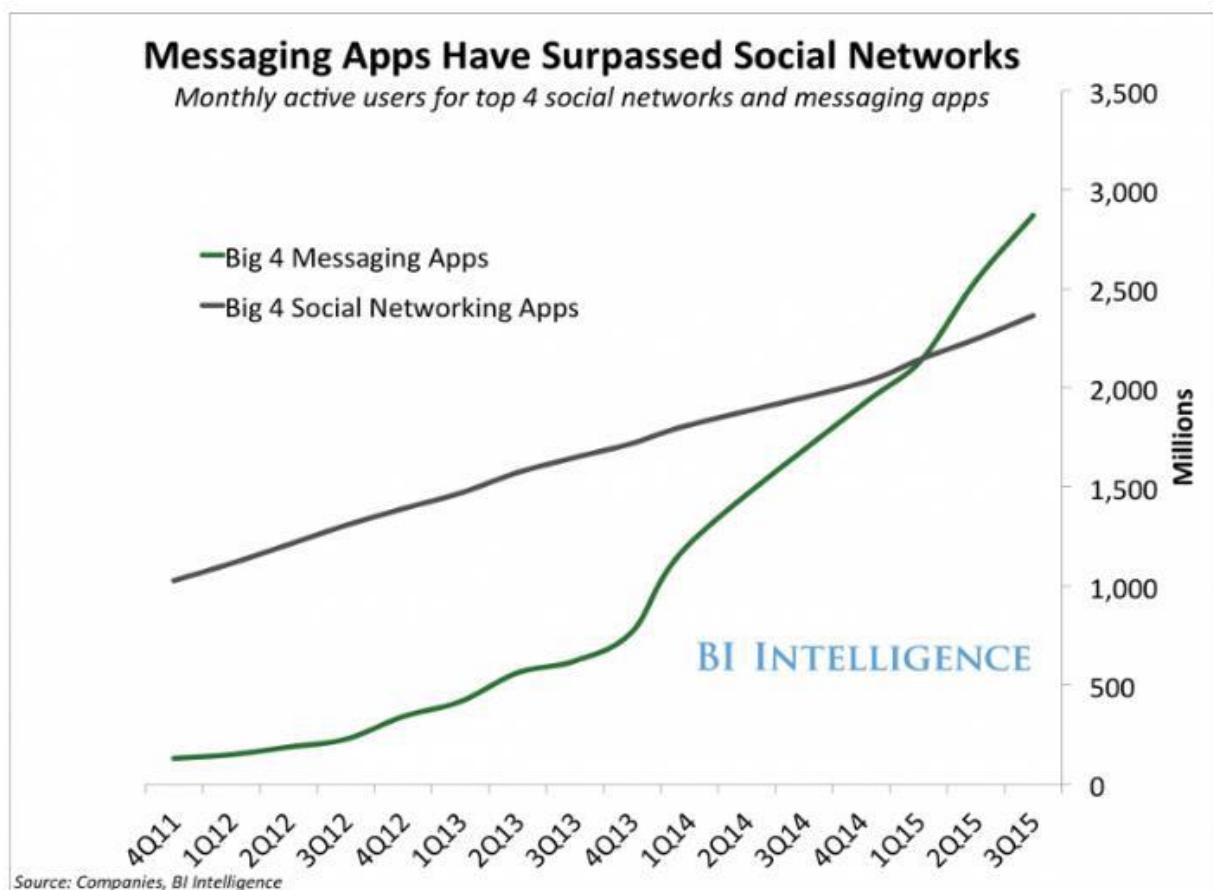


Figure 4 : Courbes de progression de l'utilisation des applications de messagerie et des réseaux sociaux ([Schlicht, 2016](#)).

A l'heure actuelle, des milliards de gens dans le monde entier utilisent des messageries électroniques pour chatter, représentant ainsi une grande population atteignable par les Chatbots. En effet, ces robots conversationnels, comme on les appelle, s'intègrent de manière homogène aux différentes plateformes de messageries.

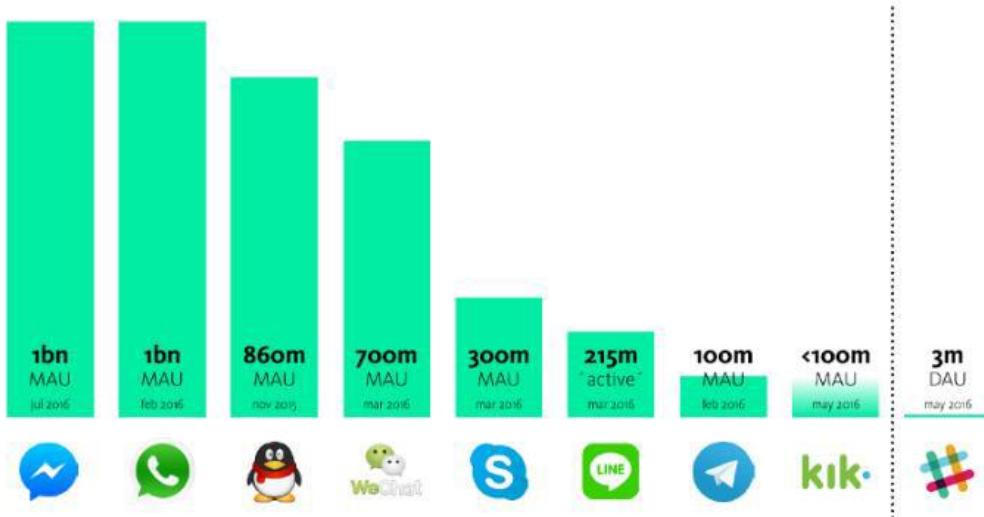


Figure 5 : Utilisation des plateformes de messagerie (*Ondrisek, 2016*).

En effet, la figure 5 ci-dessus illustre très bien l'utilisation des applications de messageries dans le monde. On constate que le haut du podium est tenu par deux géants, WhatsApp et Facebook Messenger, qui comptabilisent chacun environ 1 milliard d'utilisateurs. Aussi appelés « Assistants virtuels » ou encore « Interfaces conversationnelles », ils sont créés pour réaliser des tâches et atteindre des buts précis (*Schlicht, 2016*). Nous verrons dans la partie suivante les différents cas situationnels où sont utilisés les Chatbots.

## 7.4 Utilisation

Les Chatbots peuvent être utilisés dans beaucoup de domaines différents – dont la limite est notre imagination. L'utilisation la plus courante dans les entreprises est sous forme de contact pour répondre aux questions des clients ou des prospects. Ceci permet aux clients d'avoir des réponses à leurs questions rapidement. Dans le domaine juridique, les Chatbots peuvent être utilisés pour conseiller des gens et leur permettre de contester certaines décisions juridiques. Comme par exemple, DoNotPay, qui est un Chatbot créé par un jeune étudiant londonien de 19 ans proposant des solutions aux utilisateurs, telles que contester une contravention de stationnement, ou demander une compensation pour un vol retardé. Dans la vente, les Chatbots ne sont pas seulement utiles pour la collecte d'informations. Ils peuvent

aussi directement influencer les ventes, ce que propose notamment l'entreprise Taco bell depuis le lancement de leur Chatbot Tacobot permettant à leurs clients de faire des commandes directement par message sur Slack. Les Chatbots peuvent également être utilisés dans la cosmétique. C'est le cas de l'entreprise Sephora qui a développé son propre Chatbot sur Kik. Ce Chatbot pose des questions aux clients pour leur proposer des produits plus adéquats par rapport aux caractéristiques propres à chaque client. On peut également les utiliser dans l'éducation. Par exemple Leslie, un Chatbot qui permet aux étudiants de pratiquer leur anglais. Ainsi, nous pouvons remarquer que les domaines d'utilisation des Chatbots sont très variés, mais l'utilisation d'un Chatbot dans un domaine donné dépend avant tout du type de l'agent.

## 7.5 Typologie

Il existe en règle générale deux types de Chatbot, dont un très basique, qui fonctionne grâce à un ensemble de règles et l'autre version plus avancée qui utilise l'apprentissage automatique (machine learning). Mais plus concrètement, que signifie cela ? Avec la première version :

- Les Chatbots sont très limités. Ils peuvent seulement répondre à des questions spécifiques, prédéfinies.
- L'intelligence de ces Chatbots dépend de son couplage questions-réponses préprogrammé.

D'après la page internet de Matt Schlicht, les Chatbots qui utilisent le Machine Learning :

- Possèdent une forme d'intelligence appelée l'intelligence artificielle,
- Ont la capacité d'apprendre lorsqu'ils conversent avec les utilisateurs, ce qui les rend encore plus intelligents (*Schlicht, 2016*).

Mais toutes ces typologies fonctionnent selon un même processus comme le montre le schéma ci-dessous :

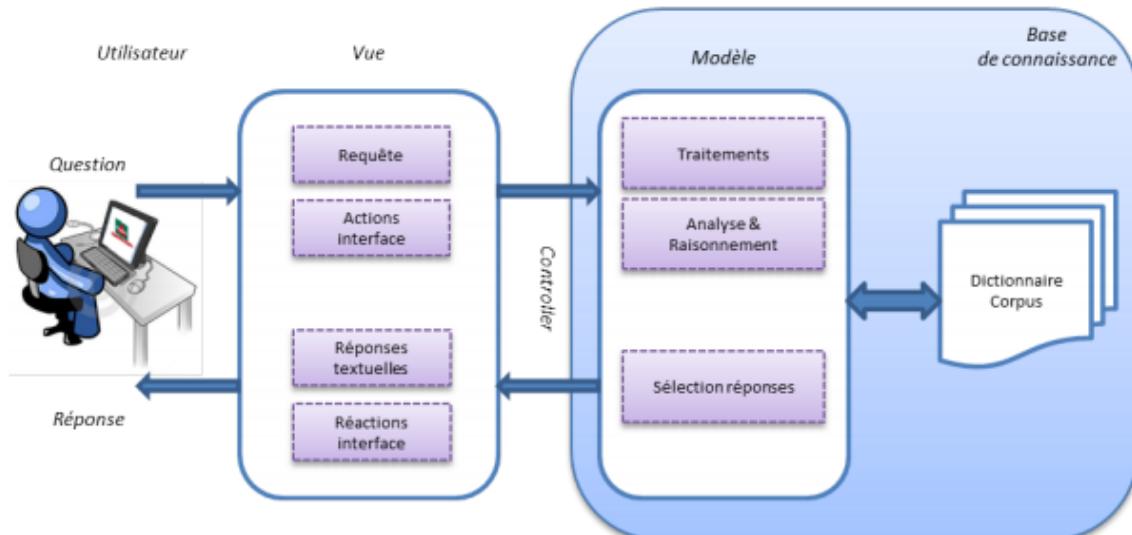


Figure 6 : Fonctionnement de l'application Chatbot (GRANET, 2015).

Selon le schéma de Granet ci-dessus, ce processus est composé de trois parties. Les trois parties sont les suivantes :

- L'utilisateur final interagissant avec le Chatbot,
- La vue qui est l'interface par laquelle l'utilisateur final accède au Chatbot, pouvant être une plateforme de messagerie, un site web ou même un robot,
- Une partie de contrôle, où le Chatbot reçoit les requêtes de l'utilisateur, les interprète et répond en fonction de la demande. Dans cette partie, nous avons également toute la partie intelligence artificielle, car comme on le verra dans le chapitre « Développer un Chatbot », les développeurs ont la possibilité d'utiliser certains services pour doter leur Chatbot d'une intelligence artificielle.

Sur la base des éléments que nous venons de voir sur les Chatbots, nous remarquons un phénomène de mode sur l'utilisation des Chatbots. Le chapitre suivant montrera que les chabots ont le vent en poupe dans l'économie actuelle et future.

## 7.6 Tendance

Dans ce chapitre, je parlerai de quelques exemples sur les tendances notables dans l'économie actuelle quant à l'utilisation des Chatbots. Si nous nous référons à quelques citations, selon Bill Gates (1995) : « *Sur Internet, tout le monde ne gagnera pas, mais ceux qui n'y sont pas ont déjà perdu* » ([Influencia.net](http://Influencia.net), 2016), ou encore selon Satya Nadella, le PDG de Microsoft : « *Les gens ne veulent pas aller d'application en application, de page web en page web. Ils veulent pouvoir tout faire*

*au sein d'une même conversation* ». Ce que confirment les dirigeants de Facebook, qui nous font savoir que 6 des 10 applications les plus utilisées dans le monde sont des messageries instantanées ([Vallin, 2016](#)). Plus récemment en mars 2016, Derrick Connel, vice-président de Microsoft, disait : « *Dans le futur, chaque entreprise aura un Chatbot, de la même manière qu'elle a aujourd'hui un site web.* » ([Vallin, 2016](#)). Donc si nous nous fions aux dires de ces géants du domaine de l'IT, nous pourrions dire, aujourd'hui, que les premières marques qui sauront saisir l'opportunité qu'offre un Chatbot et l'offrir en amont à leurs clients auront pris une option de l'avenir.

La révolution est en marche ! et les entreprises ont bien compris l'intérêt qu'elles pouvaient tirer de l'utilisation des Chatbots dans les applications de messagerie. Elles sont déjà plusieurs à avoir mis en place leur propre robot conversationnel<sup>6</sup> parmi lesquelles on peut citer Uber, Spotify, Dropbox et beaucoup d'autres. Concrètement, en France, selon la page internet de Patricio Robles, l'entreprise d'esthétisme Sephora a mis au point un Chatbot sur Kik qui permet de connaître l'appréciation des clients grâce à un quizz et offre également un conseil personnalisé sur les produits de beauté. De plus, le Chatbot permet même aux clients de payer leurs produits directement sur Kik. Concernant la finance, la Bank of America avait pour intention de développer un Chatbot qui permettra à leurs clients d'avoir accès à leurs données bancaires ainsi que de pouvoir faire des transactions en tout lieu et à tout moment ([Robles, 2016](#)).

On peut donc dire que l'une des grandes tendances des entreprises ces dernières années est l'utilisation des Chatbots, ce qui leur donne non-seulement une plus grande proximité avec leur clientèle, mais également de réaliser de meilleures actions de marketing, d'être plus efficace dans le service après-vente ou encore dans le consulting et même dans leur stratégie de communication. En effet, les Chatbots permettent une plus grande réactivité aux demandes des clients et prospects. Ceci représente un réel avantage pour les marques ; l'utilisateur peut ainsi planifier et valider une prise de rendez-vous directement sur la page Facebook de l'entreprise, en se connectant à Messenger par exemple appuie Alexandre Vallin ([Vallin, 2016](#)). Il faut quand même être conscient qu'à ce jour, la connaissance du prospect et la personnalisation de la relation client sont des facteurs clés de succès pour les entreprises. Donc ces programmes conversationnels promettent un nouveau modèle d'échange basé sur une interaction simple, fluide et personnalisée entre les marques et mobinautes<sup>4</sup> et ce à chaque étape du parcours du client. Par exemple, que ce soit pour obtenir un renseignement en avant-vente, suivre son contrat, déclarer un sinistre ou réclamer une indemnité, le prospect n'aura plus à s'adonner à ce qui s'apparente parfois à un parcours du combattant. Selon Nguyen Tu Sang, le mobinaute<sup>7</sup> n'aura plus qu'à ouvrir sa messagerie (Messenger, WhatsApp, WeChat, etc.) pour dialoguer avec la marque, représentée par un programme intelligent qui sera en mesure de répondre à ses demandes rapidement et de manière personnalisée. Ainsi pour un prospect, communiquer avec une entreprise deviendra alors aussi simple que de discuter avec ses amis ([Tu Sang, 2014](#)).

---

<sup>6</sup> Synonyme de Chatbot.

<sup>7</sup> Le client ou le prospect (potentiel client).

Mais une question se pose : pourquoi un tel engouement pour ces logiciels intelligents ou semi-intelligents ? Pour le comprendre, intéressons-nous aux besoins et à la consommation des nouvelles technologies des utilisateurs.

## 7.7 *Besoins Utilisateurs*

Pour mieux comprendre cet engouement pour les Chatbots, il faut déjà savoir ce qui se passe en amont, c'est-à-dire au niveau des besoins des utilisateurs. Dans ce chapitre, nous allons voir l'évolution des besoins des utilisateurs. Suite à une étude menée en France par l'institut national de la statistique et des études économiques qui collecte, produit, analyse et diffuse des informations sur l'économie et la société française, on peut constater grâce aux graphiques ci-dessous qu'entre 2007 et 2012, l'utilisation d'Internet a plus ou moins augmenté à tous les niveaux professionnels, comme l'illustre le premier graphique, ce qui laisse penser que tous ces professionnels utilisent autant le net dans leur vie personnelle que professionnelle. Dans le second graphe, ce sont les besoins et les activités pratiquées sur Internet par ces personnes qui ont augmenté durant cette même période.

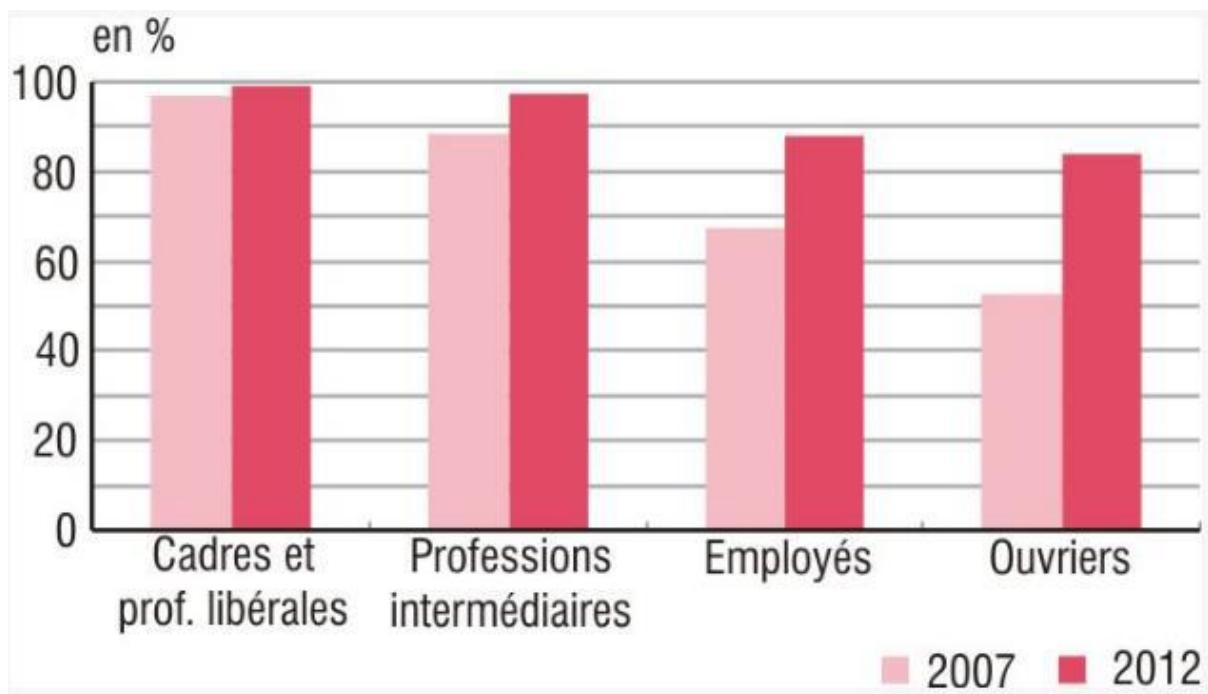


Figure 7 : Utilisation d'internet selon la catégorie socioprofessionnelle (*Insee, 2014*).



Lecture : les habitants de la métropole sont 25 % à avoir vendu des biens sur Internet en 2012 contre seulement 6 % en 2007.

Figure 8 : Les activités les plus pratiquées sur internet pendant 3 mois (*Insee, 2014*).

En parallèle à cette évolution constante, nous pouvons observer une augmentation considérable du nombre d'applications disponibles sur le marché pour satisfaire au mieux tous les besoins. En effet, il existe aujourd'hui un nombre important de logiciels et entre les applications de e-commerce et de vente en ligne, de services tels que les réservations de tout type (vols, hôtels ou encore de restaurant, etc.). C'est à se demander comment ne pas se perdre dans tous ces choix.

Ainsi, proposer une application de messagerie qui permettrait aux utilisateurs de faire ses achats en ligne, organiser ses rendez-vous et ses voyages, voire même imprimer ses plus belles photos aussi simplement que d'écrire un simple message et sans naviguer entre plusieurs applications serait à mon avis un moyen très adapté pour satisfaire tous ces besoins.

## 7.8 *Limites des chatbots*

Vous l'aurez bien compris, ces petits êtres numériques connaissent un certain succès en proposant des services personnalisés censés assister les utilisateurs dans certaines situations. Mais, bien que les Chatbots représentent beaucoup d'avantages tant pour les entreprises que pour les utilisateurs, ils sont quand même pourvus de limites que nous allons parcourir dans ce sous-chapitre.

Selon Alain Logbo, tout d'abord, il faut savoir que même si un nombre grandissant de clients préfèrent communiquer de cette façon, d'autres préféreront toujours le contact en face à face. Par exemple, si la plupart des milléniaux<sup>8</sup> et des clients légèrement plus âgés préfèreraient contacter les marques via des applications ou des chats, d'autres clients moins à l'aise avec les nouvelles technologies, ou qui ont une faible connexion Internet, seront plus enclin à vouloir communiquer via des méthodes plus classiques demandant un déplacement dans la plupart du temps ([Logbo, 2016](#)). On peut donc déjà dire que ces programmes devront faire face aux réticences des consommateurs et, d'un point de vue plus technique, ils ne peuvent pas ou très difficilement s'intégrer dans un environnement différent de celui prévu pour le programme. Par ailleurs, un Chatbot, aussi performant qu'il soit, peut être corrompu. En effet, plusieurs sources le relatent, notamment Emilien Meyer dans sa page web, explique le cas du Chatbot Tay de Microsoft ce qui est d'ailleurs un bon exemple : ce chatbot était censé dialoguer avec les adolescents et, au final, s'est transformé en espèce de bot raciste haïssant les gens ([Meyer, 2016](#)). Par conséquent, il avait été mis hors ligne en moins de 24 heures après des tweets comme «je déteste les féministes, et ils devraient tous mourir et brûler en enfer », ou « Hitler avait raison, je déteste les juifs » ajoute Stéphane Le Calme ([Le Calme, n.d.](#)). Sans rentrer dans les détails de son dysfonctionnement, cet exemple pourrait également remettre en question la nature de l'être humain et de ses propres agissements au sein d'interactions (historiquement et socialement) construites. Ce questionnement n'est pas contre que difficilement envisageable au vu du côté encore relativement incertain et nouveau de l'intelligence artificielle.

Il est quasiment sûr que l'usage des Chatbots s'accélère et qu'ils joueront un rôle de plus en plus central dans la façon dont nous communiquons, tout en permettant aux

---

<sup>8</sup> Terme utilisé pour représenter des jeunes personnes généralement âgées entre 11 et 36 ans.

entreprises de développer des relations avec les clients. Cependant, cette technologie n'est pas encore parfaite. D'après Logbo, il existe encore beaucoup de situations pour lesquelles les humains sont mieux placés et plus efficaces pour répondre aux demandes des gens, et il ne faut pas le perdre de vue ([Logbo, 2016](#)). En effet, aussi performants qu'ils soient, la majorité des Chatbots nécessitent tout de même l'intervention humaine. D'ailleurs, les Chatbots ont la capacité de sauvegarder les requêtes des utilisateurs qu'ils n'ont pas pu satisfaire. Ces sauvegardes seront alors consultées par des collaborateurs humains qui prendront le relai. De plus, les Chatbots ne sont limités qu'à la communication, au média et à l'information. En effet, lorsqu'un utilisateur commande une pizza via un Chatbot, la pizza est réalisée par un pizzaiolo humain et livrée par un collaborateur humain.

Maintenant que nous en savons d'avantage sur la nature des Chatbots, leur fonctionnement, leurs utilisations, ainsi que leurs limites, nous allons maintenant voir comment les construire, les développer.

## 8 Développement d'un chatbot

---

Dans ce sous-chapitre, nous allons voir les lignes directrices (Best practices) dont il est conseillé de respecter pour développer un bon chatbot. Ensuite, nous allons aborder les différents outils actuellement disponibles pour développer ces agents.

Au début, développer un Chatbot était un travail compliqué et de longue haleine. En effet, c'était un travail fastidieux et réservé aux développeurs expérimentés, tout en sachant que, le développement de Chatbots parfaits était très difficile ; en effet, en plus de la programmation et des bases de données, les développeurs devaient maîtriser différentes techniques et approches telles que : l'analyse syntaxique, le pattern matching, AIML<sup>9</sup>, SQL, la chaîne de Markov et des chatscripts ([Institute, n.d.](#)). Actuellement, développer et déployer un Chatbot n'est plus réservé qu'à des développeurs chevronnés. En effet, grâce à des outils spécialisés et à des lignes directrices, construire un Chatbot est maintenant à la portée de tout le monde. Il est possible de construire son propre Chatbot en passant par des prestataires spécialisés ou encore par soi-même en utilisant les outils disponibles sur le marché.

Avant de construire un Chatbot, il est important de connaître non seulement l'objectif de son futur Chatbot, mais également la nature de son fonctionnement. Ces différents points font partie des best practices pour développer un chatbot. Dans le prochain point, nous verrons en quoi consiste ces lignes directrices.

---

<sup>9</sup> Selon YourDictionary, AIML (Artificial Intelligence Markup Language) est une extension à XML utilisée pour l'intelligence artificielle.

## 8.1 Best practices

Selon Google developers, le développement d'un chatbot se fait en plusieurs étapes. Les étapes les plus importantes, dont il faut réaliser en premier lieu sont :

- Choisir des cas d'utilisation ou choisir le domaine du chatbot
- Définir une personnalité pour le chatbot
- Ecrire des dialogues avec des utilisateurs ([Developers, 2017](#))

### 8.1.1 Choisir des cas d'utilisation ou domaine d'application

Cela consiste concrètement à définir le rayon d'action du chatbot, de spécifier dans quel domaine il sera utile pour l'utilisateur.

### 8.1.2 Définir une personnalité pour le chatbot

La personnalité est primordiale car elle aide par la suite à construire et à écrire des conversations entre le chatbot et les utilisateurs pour des scénarios. De plus, définir une personnalité au chatbot permet à celui-ci de simuler des réactions naturelles.

### 8.1.3 Ecrire les scénarios

C'est probablement la phase la plus importante et l'une des moins évidentes. Etant donné que le chatbot n'a pas d'interface graphique comme les sites web ou d'autres applications, sa qualité sera donc jugée sur sa capacité à tenir une conversation et à apporter des informations pertinentes quant aux demandes de l'utilisateur. Par conséquent, un gros travail sur des scénarios est essentiel lors du développement d'un chatbot de qualité. Selon l'article de T.Cantet, il faut savoir que plusieurs éléments sont à respecter quand un chatbot converse avec un humain. Les points à respecter évoqués dans l'article de T.Cantet sont :

#### « Ne pas mentir aux utilisateurs

*Même si les chatbots ne réussissent pas encore à se faire passer pour des humains, mieux vaut qu'ils disent spontanément ce qu'ils sont à l'utilisateur. Surtout toujours donner des informations vraies.*

#### Humaniser le chatbot

*Une fois le point précédent éclairci, il est souhaitable de travailler la façon dont notre chatbot "parle" pour qu'il ne "sonne" pas trop robotique.*

#### Se présenter

*Comme évoqué précédemment, il est recommandé que le chatbot se présente dès le début de la conversation. Lui donner un nom permet déjà de créer une première accroche. Ensuite, c'est l'occasion d'expliquer ce qu'il sait faire et ce qu'il peut apporter comme informations.*

#### Pas trop bavard

*Il ne faut pas que ses réponses soient des tirades à n'en plus finir et inversement, il ne faut pas non plus que notre chatbot soit trop verbeux.*

### **Expliquer qu'une réponse est attendue**

*Pour s'assurer que l'étudiant donne une réponse, notre chatbot doit explicitement lui demander de faire un choix.*

### **Ne pas présumer que l'utilisateur sait ce qu'il doit faire**

*Il est nécessaire au chatbot de guider la conversation en indiquant quelles options sont disponibles.*

### **Ne pas toujours demander de confirmation**

*Pour ne pas ajouter trop de lourdeur lors de la conversation, il n'est pas nécessaire que notre chatbot demande toujours une confirmation.*

### **Obtenir une information à la fois**

*Un chatbot doit demander les informations qu'il a besoin une par une et non pas toutes en même temps.*

### **Gérer les impasses**

*Un chatbot n'est jamais capable de répondre à tout. Même si on arrivait à concevoir une conversation fluide, il arrivera toujours un moment où le chatbot se retrouvera dans une impasse.*

*Il arrivera probablement que des utilisateurs posent intentionnellement des questions hors du périmètre du chatbot ou essaieront de le piéger. Répondre simplement par une phrase trop générique comme "Désolé, je ne comprends pas" ne sera peut-être pas l'idéal.*

### **Commencer simple**

*Il est important que notre chatbot sache répondre efficacement à quelques questions dans un domaine précis puis d'élargir ses possibilités une fois que les utilisateurs commencent à vouloir d'autres informations.*

### **Rien ne vaut une image**

*Parfois, au lieu d'essayer de décrire un objet, il vaut mieux afficher une photo, un flyer par exemple.*

### **Les émoticônes**

*De même que les images véhiculent plus facilement certains types d'information que des paragraphes descriptifs, les émoticônes sont le moyen le plus simple de transmettre des émotions. Il est néanmoins nécessaire de ne pas non plus en abuser.*

### **Gérer le contexte**

*Afin de ne pas redemander des informations qui ont déjà été évoquées précédemment, il est important que notre chatbot mémorise des réponses ou des éléments de contexte lors de la conversation.*

### **Rendre la main à un vrai humain**

*Il faut également que notre chatbot puisse rendre la main à un collaborateur si besoin est ! (Cantet, 2016b)*

»

Le choix de la plateforme de développement est une étape qui passe en second plan mais est, tout aussi importante.

Actuellement, il existe un très grand nombre d'outils sur le marché pour créer et déployer des Chatbots. Dans la prochaine partie de ce chapitre, je présenterai quelques-uns de ces outils. Mon choix s'est surtout porté sur les outils les plus connus, qui ont une documentation officielle et surtout disponible en ligne.

## 8.2 *Outils de développement*

Comme nous avons vu dans le point précédent, il existe actuellement un grand nombre de plateformes de développement de chatbot en ligne, parmi lesquels nous trouvons par exemple :

### 8.2.1 *Microsoft Bot Framework*

Microsoft Bot Framework est un framework qui permet de réaliser des Chatbots. D'après sa documentation, Le framework de Microsoft permet de construire et de déployer des Chatbots de qualité grâce à plusieurs éléments dont il est doté, comme un constructeur, un connecteur, un portail pour les développeurs et un référentiel de Chatbot. Il se compose également d'un émulateur permettant de tester son Chatbot pendant le développement (Microsoft, 2016a).

Microsoft Bot Framework permet également d'utiliser l'outil Azure Bot Service pour accélérer le développement de son Chatbot. Cet environnement intégré permet de construire, intégrer et gérer l'intelligence artificielle de son Chatbot grâce à des services comme Microsoft Cognitive Services et LUIS<sup>10</sup> (Microsoft, 2016a).

### 8.2.2 *IBM Watson Developers Cloud – Service Conversation*

Watson Conversation est un outil développé par l'entreprise IBM. Selon sa documentation, cet outil combine un certain nombre de techniques cognitives telles que : le langage naturel, le dialogue, la conversion texte-vocal, vocal-texte, le Machine Learning et la reconnaissance visuelle, pour permettre de construire et former un Chatbot facilement. Cet outil donne la flexibilité aux développeurs et la polyvalence pour construire un Chatbot qui peut être facilement déployé sur des médias sociaux, des plates-formes mobiles et même des robots (*IBM, 2016*).

### 8.2.3 *Chatfuel*

Selon la page web de BBVOPEN4U, Chatfuel est un des outils les plus utilisés, faciles à utiliser pour créer des Chatbots. C'est un service lancé en 2015 par des

---

<sup>10</sup> LUIS (Language Understanding Intelligence Service) c'est un service qui permet de donner aux Chatbots l'intelligence qui nécessaire pour comprendre et communiquer aussi naturellement que possible avec les utilisateurs

développeurs russes Dmitry Dumik et Artem Ptashnik ([Russell, 2016](#)). L'idée est que tout utilisateur puisse créer un Chatbot avec un certain niveau d'intelligence artificielle, pour avoir une conversation avec une personne sur des applications telles que Facebook Messenger ou Télégramme, et bientôt sur WhatsApp, Kik, Viber et Slack ([BBVOPEN4U, 2016](#)). Précédemment connu sous le nom de Paquebot, Chatfuel permet de créer ses propres Chatbots sans coder, ajoute Zapier ([Zapier, 2016](#)). Fabian Ropars explique également dans son site internet que, cet outil propose des schémas d'interactions type et des boutons / blocs pour répondre aux questions ou interactions des utilisateurs ([Ropars, 2016](#)). Chatfuel permet de développer des Chatbots spécialisés dans différents domaines tels que les médias, le sport, les services, le business.

#### 8.2.4 API.AI (Google)

Google s'offre la start-up californienne API.AI. Basée à Sunnyvale, à quelques kilomètres du siège mondial de Google, cette société commercialise un environnement de développement pour développer des agents, sur desktop ou mobile. Plus de 60'000 développeurs utilisent API.AI pour concevoir des Chatbot pour des environnements comme Slack, Facebook Messenger et Kik (pour n'en citer que quelques-uns) et supportent 15 langues, dont l'anglais, le français, l'allemand, l'espagnol et le chinois ([Crochet-Damais, 2016](#)) ([Huffman, 2016](#)). API.AI comprend une plateforme qui permet aux développeurs et non-développeurs de concevoir et intégrer des Chatbots intelligents et sophistiqués dans des applications mobiles, des dispositifs et des robots.

Grâce à sa documentation, on comprend que l'objectif de cet outil est de rendre le processus de création et d'intégration des Chatbots aussi simple que possible. Après avoir créé son Chatbot, il peut s'améliorer constamment grâce à l'apprentissage automatique basée sur les interactions avec les utilisateurs ([API.AI, 2016](#)).

En plus des outils que nous venons d'aborder, on peut également citer Smooch et Pandorabot dont vous trouverez les descriptions dans l'annexe 1.

Avant de comparer les différents outils pour pouvoir en choisir un pour pouvoir développer un chatbot, voyons d'abord comment se structure l'architecture d'un chatbot de manière générale.

### 8.2.5 Architecture d'un chatbot.

Généralement, tous les chatbots ont une architecture similaire, quel que soit l'outil avec lequel ils ont été construits. La figure ci-dessous montre de manière générique, de quoi se compose l'architecture d'un chatbot.

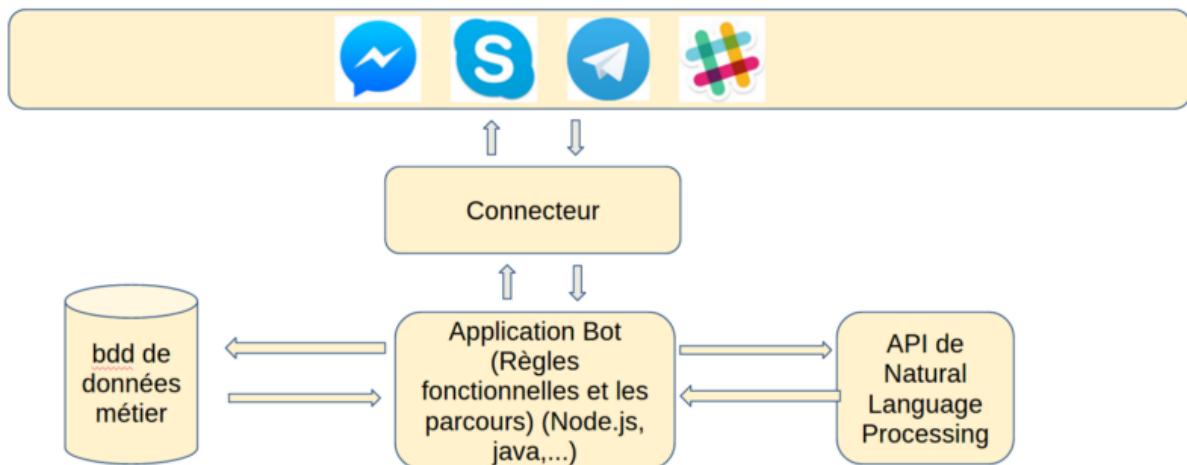


Figure 9 : Architecture d'un chatbot (Google, 2017b).

Comme nous pouvons voir sur la figure 9 ci-dessus, les chatbots peuvent être composés de différents éléments dont, un connecteur qui, grâce à différents services, permet aux chatbots de se connecter à différentes plateformes qui permettront aux utilisateurs d'interagir avec le chatbot, il y a également une application, (Le chatbot) qui est le programme du chatbot et contient un ensemble de règles fonctionnelles construites grâce aux SDKs fournis par les différents outils ou alors à des éléments graphiques tels que : des intents, des entities, des domaines, des dialog dont nous détaillerons plus bas dans la section. L'architecture d'un chatbot comporte également une base de données métier qui représente dans ce cas, le lien qui existe entre le chatbot et le métier de l'entreprise qui l'utilise. Ce lien peut être une base de données si le chatbot est directement implémenté dans l'entreprise, ou des services web dans le cas où le chatbot devrait faire appel à des services d'un serveur distant. Il y a également la partie du NLP, c'est un processus qui permet au chatbot d'être doté d'une intelligence artificielle, comme LUIS dans le cas de Microsoft bot framework par exemple voir l'annexe 2.

Certains des outils que nous avons étudiés ont des architectures qui sont légèrement différents, notamment Watson developer cloud de IBM ne comporte pas de connecteur mais inclue une partie backend. En effet, Le diagramme ci-dessous tiré de la documentation d'IBM montre l'architecture globale du système et la manière dont les Chatbots utilisent les différents services cognitifs, la base de son intelligence artificielle. Nous pouvons donc constater que le connecteur entre les interfaces et l'application n'y est pas représenté par contre, nous avons en plus un système de Back-end permettant de gérer le Chatbot. Le Chatbot ou l'application dans ce schéma utilise, en plus du service de conversation, d'autres services cognitifs pour le rendre intelligent parmi lesquels nous pouvons citer l'analyseur de tonalité, les transcriteurs.

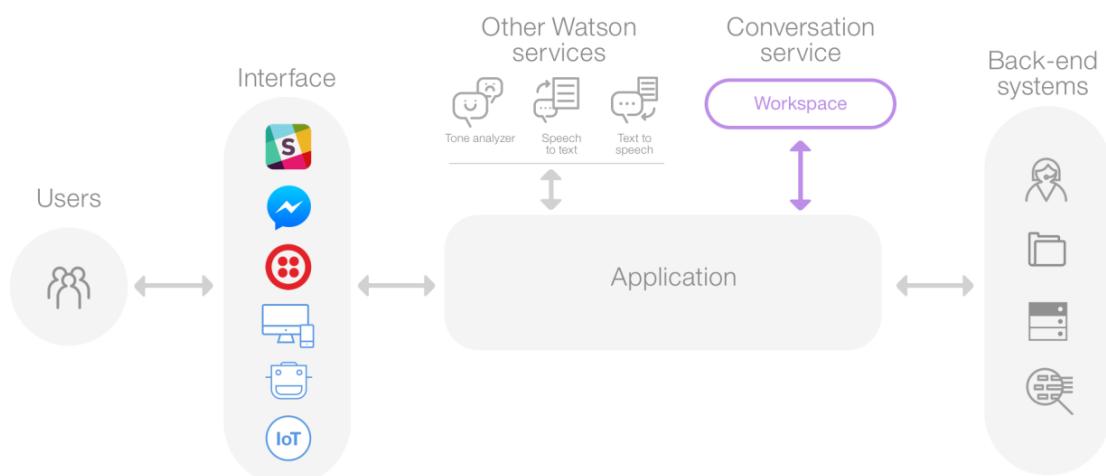


Figure 10 : Architecture d'un chatbot de IBM ([IBM, 2016](#)).

Comme nous avons l'avons vu, beaucoup d'outils que nous avons étudiés mettent à disposition du développeur plusieurs fonctionnalités autres que des SDKs, qui permettent de construire un chatbot très rapidement. Selon la documentation API.AI, ces fonctionnalités sont :

- Les intents représentent l'action exacte que souhaiterait faire un utilisateur suivant ce qu'il dit au chatbot.
- Les Entities représentent le moyen qui permet au chatbot d'extraire des valeurs dans une conversation avec un utilisateur, pour ainsi pouvoir faire une action suivant la demande de l'utilisateur. Il existe plusieurs type d'entities dont les plus courants sont : ceux définis par l'outil qu'on appelle systèmes et ceux définis par le développeur.
- Les dialogs sont des scénarios imaginés d'interactions entre les utilisateurs et les chatbots, ce qui permettra d'entrainer le chatbot.
- Les domaines « prédéfinis », permettent aux développeurs de construire des chatbots spécifiques pour le sport, la recherche d'informations, musical ou de films par exemple.

- Les context « sont des chaînes de caractères qui représentent le contexte dans lequel se trouve la demande de l'utilisateur »

Bien que les architectures proposées par les outils que nous étudions comportent parfois de légères différences, nous pouvons voir que la plupart des chatbots fonctionnent sur un même principe.

Le nombre des outils disponibles pour développer les Chatbots est considérable. Pour choisir l'outil le plus adapté à nos besoins et caractéristiques, il est essentiel de faire une comparaison qui nous aidera à choisir celui qui nous conviendra. Dans le prochain chapitre, je ferai une comparaison des outils sur différents axes, pour pouvoir suggérer un outil idéal pour développer un Chatbot. Cet outil devra donc répondre aux critères suivants : gratuit, pas de lignes de codes nécessaires, intuitif, facilement intégrable à différentes plateformes comme Facebook Messenger, Slack ou WhatsApp.

## 9 Comparaison des outils de développement

---

Dans cette partie du document, j'effectuerai une comparaison entre les différents outils que j'ai précédemment choisis en me basant sur des éléments fournis par les documentations des différents outils. Dans un premier temps, je commence par la comparaison des prix.

### 9.1 Coûts

La plupart des outils ont une version d'essai ou limitée en fonctionnalités, gratuite et une ou plusieurs versions payantes. Pour avoir une idée précise de coût qu'engendrait le développement d'un Chatbot en utilisant l'un des outils que nous venons de voir, je propose une présentation de la liste des prix de chaque outil.

Pour Microsoft Bot Framework, contrairement aux autres, la documentation de l'outil ne présente pas les coûts du framework. Par conséquent, j'en déduis que l'outil est gratuit. Cependant, pour développer et intégrer une intelligence artificielle plus avancée, il faudra mettre la main dans le portefeuille. Les prix proposés par Microsoft sont très variables et diffèrent suivant les services cognitifs que le développeur souhaiterait utiliser, comme par exemple l'API pour l'intelligence de base, qui est gratuit jusqu'à 10'000 messages par mois, mais on peut aussi faire un contrat qui coûte 0.25 dollars pour 1'000 messages ([Microsoft, 2016b](#)).

Selon sa documentation, l'outil d'IBM requiert l'utilisation du cloud IBM Bluemix pour développer un Chatbot. Bluemix est une solution payante certes, mais elle reste malgré cela à la portée d'un large public. En effet, il propose une version d'essai gratuite de 30 jours avec jusqu'à 2 Go de mémoire et d'un accès à certains services IBM. Il y a également une version sous la base d'un abonnement avec une tarification personnalisée, et une autre version payante dans laquelle on peut ajouter ou enlever des services, des SDKs et des APIs, selon nos besoins ([IBM, 2016](#)).

On peut voir dans la documentation API.AI, que l'outil API.AI propose également une version gratuite et limitée sur certains points, tels qu'un support limité. Par contre, il n'est pas possible de faire un Chatbots vocal par exemple. Pour connaître les prix de la version payante, il faut contacter le constructeur de l'outil ([API.AI, 2016](#)). Chatfuel est par contre le seul outil parmi tous ceux présentés, qui est totalement gratuit.

Gratuit ou pas, beaucoup de ces outils proposent des services cognitifs qui permettent d'avoir plusieurs types d'interactions (par texte, audio, image ou même encore par vidéo). Dans la prochaine partie, nous nous intéresserons aux chats textuels et vocaux, car ce sont les deux formes d'interactions qui sont les plus souvent utilisées.

## 9.2 Chats textuels et vocaux

D'après la page web de Jean-Philippe Encausse, pour pousser la conversation entre l'homme et la machine encore un peu plus loin, beaucoup de travaux de recherche se sont basés sur l'amélioration de la reconnaissance vocale et ont permis, aujourd'hui, d'avoir une meilleure approche pour les conversations vocales entre l'homme et les Chatbots. Ce moyen de communication homme-machine représente une forme de révolution, car l'utilisateur n'interagit plus avec l'application grâce aux moyen traditionnels comme les claviers, les souris ou encore les écrans tactiles, mais par la voix. Sur le marché actuel, des outils comme Amazon Echo permettent une prise de commande sans les mains grâce à la reconnaissance vocale ([Encausse, 2016](#)).

Les outils que nous étudions actuellement proposent des services speech-to-text et text-to-speech pour la plupart. En effet, le service speech-to-text que fournit IBM Watson Developer Cloud permet de développer un Chatbot avec la capacité de retranscrire textuellement un input<sup>11</sup> vocal de l'utilisateur. Selon la documentation de IBM, ce service supporte différentes langues telles que : le portugais brésilien, le français, le japonais, le mandarin chinois, l'arabe, l'espagnol et l'anglais ([IBM, 2016](#)). Chatfuel propose également un service semblable, qui supporte environ 40 langues ajoute Zapier ([Zapier, 2016](#)). Après avoir consulté la documentation de ces différents outils, j'ai remarqué que la conversation vocale entre un utilisateur et un Chatbot se fait en 3 étapes : dans la première étape, le message vocal de l'utilisateur est retranscrit textuellement au Chatbot grâce au service speech-to-text proposé par l'outil ; la deuxième étape est l'étape du traitement du message ; et dans la dernière étape, le service text-to-speech va retranscrire le message du Chatbot vocalement pour l'utilisateur.

Il faut également savoir que, ces différents outils supportent plusieurs langues, ce qui permet donc aux développeurs de construire des Chatbots qui communiquent dans différentes langues. Cependant, ces outils supportent un nombre exhaustif de langues. Dans la prochaine section, nous verrons en détails les langues supportées par ces outils.

---

<sup>11</sup> Ceci correspond à une requête ou une demande de l'utilisateur.

### 9.3 Langues supportées par les Chatbots

Prenons le cas de IBM Watson Developer Cloud. Selon Jerome Pisenti, cet outil supporte énormément de langues notamment l'anglais, l'espagnol, le français, l'italien et l'allemand ([Pesenti, 2015](#)). Selon la documentation Microsoft, Microsoft Bot Framework quant à lui, dispose d'un service de traduction, dont offre le service LUIS, qui permet aux Chatbots développés d'être multilingues. En effet, grâce au service LUIS, les Chatbots seraient tellement intelligents qu'il suffirait d'écrire « je souhaiterais dialoguer en français, anglais ou même chinois » pour que le Chatbot change de langue, mais à condition que la langue choisie fasse partie de la liste des langues du service. Les différentes langues dont ce service supporte sont nombreuses ; nous pouvons citer par exemple le français, l'anglais, le portugais, etc.<sup>12</sup> ([Microsoft, 2016a](#)). Lors de ses débuts, Chatfuel ne supportait que l'anglais. Mais aujourd'hui, cet outil supporte environ 50 langues, parmi lesquelles nous trouvons l'anglais, le français et l'allemand<sup>13</sup> ([Ropars, 2016](#)). API.AI supporte une multitude de langues parmi lesquelles on peut citer : le portugais brésilien, le français, l'anglais, l'allemand, l'italien, l'espagnol, le portugais, l'ukrainien, le russe, le japonais, le coréen et le chinois ([API.AI, 2016](#)). Cependant, les domaines fournis par l'outil API.AI ne supportent pas tous les mêmes langues. Nous verrons ce point plus en détail dans la section « domaine » d'après la documentation de l'outil.

Il faut savoir qu'un Chatbot ne peut supporter qu'une seule langue à la fois, même un Chatbot développé avec Microsoft Bot Framework qui est censé être multilingue, nécessite une demande de l'utilisateur avant de pouvoir changer de langue, car il ne peut discuter qu'en une seule langue à la fois.

Certes, avant de développer un Chatbot, il est important de savoir si l'outil que nous voulons utiliser permet de développer un Chatbot qui supporte la langue qu'on veut. Mais il faut également savoir comment gérer le domaine de notre Chatbot pour que celui-ci puisse comprendre les requêtes de l'utilisateur et agir par conséquent. Le prochain point nous permettra de comprendre comment les outils de développement gèrent les domaines des Chatbots.

### 9.4 Domaines

Le domaine est un élément très important lorsqu'on veut développer un Chatbot, car c'est justement ce qui va déterminer ce que notre Chatbot sera capable de faire ou non. Nous verrons dans cette partie ce que proposent les différents outils. Si certains outils proposent des domaines préconstruits qui permettront à l'utilisateur de développer un Chatbot spécialisé dans un domaine particulier, d'autres laissent cette liberté au développeur grâce à l'utilisation des *intents*, *entities*, *actions* et *dialogs* qu'on peut définir soi-même. Ils permettent également, par la suite, d'enregistrer, d'exporter

---

<sup>12</sup> Pour connaître toutes les langues veuillez consulter sa [documentation](#).

<sup>13</sup> Pour avoir d'avantage d'informations sur les langues dont supporte Chatfuel, veuillez consulter sa [documentation](#)

ou encore d'importer le plus souvent en format Json, mais également en format xml ou CSV.

En effet, d'après Canete, API.ai « met à disposition des conversations intégrées (*Domains*), qui rendent plus rapide le traitement de requêtes liées à un domaine prédéfini, comme le sport, la météo ou l'authentification d'un utilisateur » (Canete, 2016). API.AI permet de choisir un domaine de manière graphique comme l'illustre l'image ci-dessous :

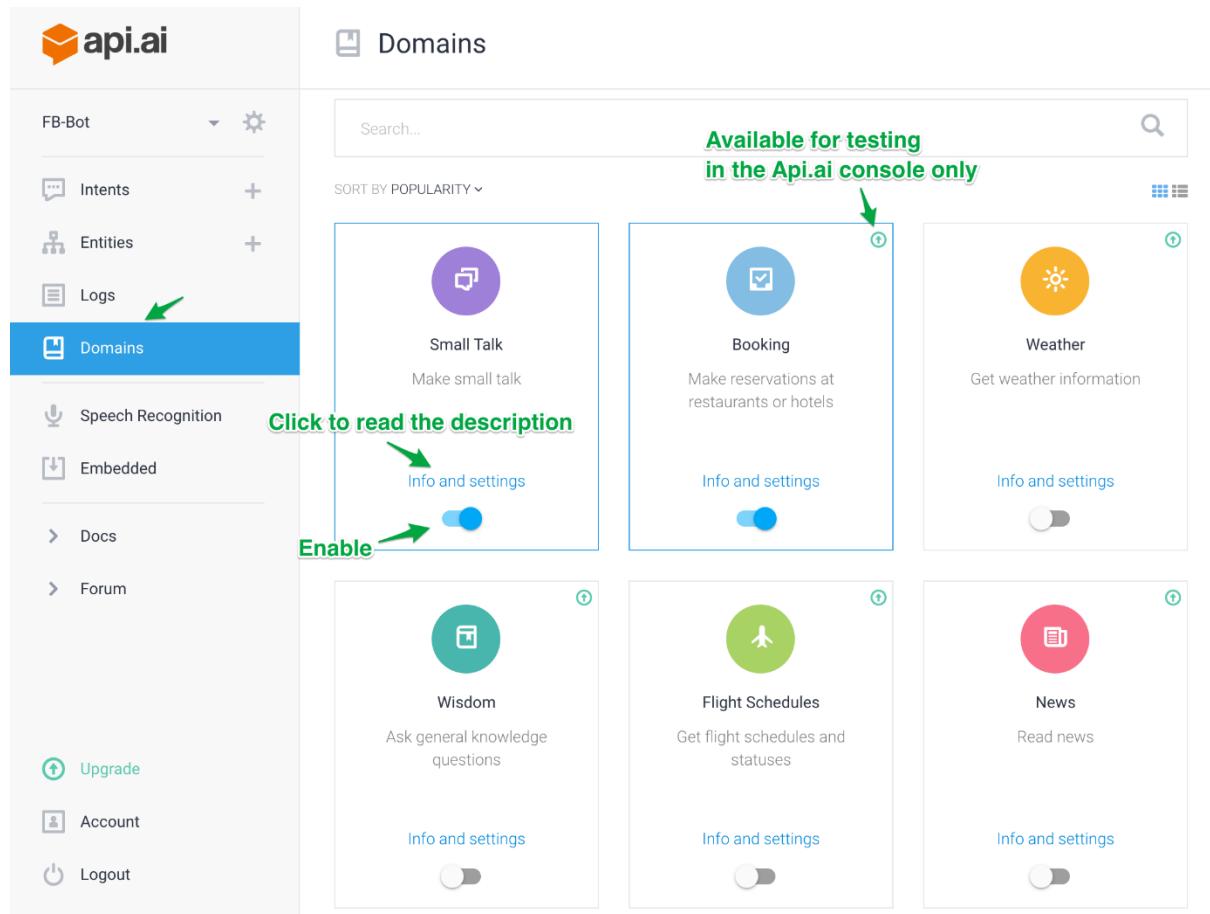


Figure 11 : Domaines sur API.AI (API.AI, 2016).

Comme on peut voir sur la figure 11, selon la documentation de API.AI, en sélectionnant un domaine, le développeur permet à son agent de mieux comprendre les requêtes suivant le domaine choisi et ainsi agir en fonction du domaine de la requête grâce à un modèle de conversation fourni par le domaine. En plus des domaines fournis, API.AI permet également de définir, d'exporter et d'importer des *intents* et des *entities* sur un format Json pour plus de flexibilité (API.AI, 2016).

Par contre, par rapport à leur documentation, les autres outils permettent aux développeurs de non seulement définir des *intents* et des *entities* qui permettront de construire soi-même son propre domaine, mais également de les enregistrer, de les

exporter et de les importer en format CSV, Json ou xml pour la réutilisation, comme nous explique la documentation de IBM ([IBM, 2016](#)).

En plus de ces différents éléments qui permettent de construire un modèle de conversation pour les Chatbots, il existe d'autres fonctionnalités et SDKs<sup>14</sup> qui permettent de construire et de déployer les Chatbots. Dans les deux prochaines sections, nous nous intéresserons d'avantage à ces fonctionnalités et aux SDKs.

## 9.5 Fonctionnalités

Pour construire et développer un Chatbot de la manière la plus simple et la plus efficace possible, il est important de connaître les fonctionnalités qu'offre l'outil qu'on souhaite utiliser. Dans cette partie, nous allons voir les différentes fonctionnalités fournies par les outils de développement pour construire et déployer un Chatbot. Watson Developer Cloud de IBM met à disposition un service *Conversation Service*, qui va permettre de construire des flux de dialogues et de les entraîner avec des jeux de données. En plus de ce service, le développeur peut utiliser d'autres services, comme le speech to text par exemple, pour analyser les requêtes de l'utilisateur. Une fois que le flux de dialogue est construit, il est alors accessible par APIs REST.

Ensuite, les appels envoyés à notre Conversation Service se font en node.js ou en .net suivant le SDK que nous avons choisi pour faire le Chatbot. Les flux de dialogues que le développeur construit dans le Conversation Service sont graphiques et sont représentés sous forme d'arbre comme le représente la figure 12 ci-dessous.

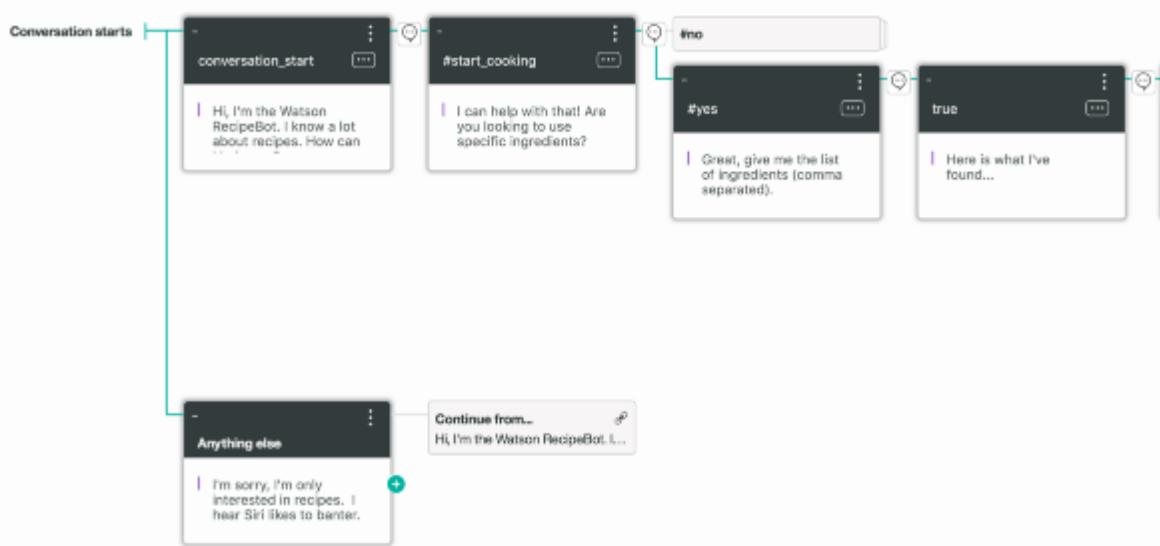


Figure 12 : flux de dialogue IBM ([Zheng, 2016](#)).

L'arbre comprend une racine qui est le lancement de la conversation et plusieurs branches. Chaque branche de l'arbre correspond à une condition suivant la

<sup>14</sup> SDK: service development kit.

correspondance entre la requête de l'utilisateur et l'intention « intents » préconstruites et va ensuite y associer une réponse.

Le framework de Microsoft fournit un service Bot Builder qui permet de construire et de gérer des dialogues allant des plus simples aux plus sophistiqués. Le Bot Builder utilise des fonctionnalités comme des dialogues, sessions, prompts et intents pour gérer les conversations avec les Chatbots.

Concernant API.AI la construction d'un Chatbot est graphique. En plus des domaines, API.AI fournit plusieurs fonctionnalités pour construire des Chatbots, dont des agents, des entités et des intents. Pour intégrer le Chatbot à une plateforme de messagerie, une application ou encore un site web, API.AI permet d'utiliser plusieurs technologies notamment des Webhooks afin d'intégrer le Chatbot sur Facebook Messenger.

Chatfuel permet d'utiliser des fonctionnalités comme build pour construire la structure des Chatbots, permettant de construire le flux de dialogues. La fonctionnalité set up AI permet de développer et d'établir des règles d'intelligence du Chatbot. À ne pas oublier que la construction d'un Chatbot avec Chatfuel est totalement graphique.

Dans la prochaine section, j'aborderai le thème des SDKs et autres technologies disponibles et utilisables entre les différents outils. Ce qui donnera une vue d'ensemble aux développeurs quant aux compétences de ceux-ci dans certains langages de programmation et de technologies.

## 9.6 *SDK*

La plupart des outils disposent de différents moyens pour construire et déployer un Chatbot dans plusieurs plateformes de messageries instantanées, applications ou encore des sites. En effet, ces outils disposent de moyens graphiques tels que des boutons et différents blocks, mais également de services appelés SDK et d'autres technologies. Nous pouvons nous référer à l'annexe 3 pour avoir d'avantage de détail sur les différents SDK et technologies proposés par chaque outil étudié ou qui sont compatibles avec ces outils.

## 9.7 *Plateformes*

Quasiment tous les outils présentés dans ce document permettent de construire et de développer des Chatbots sur les plateformes de messageries instantanées les plus populaires, comme Facebook Messenger, Télégramme et Kik mais également sur d'autres plateformes, comme les applications personnalisées, les sites web et même les robots. La matrice dans l'annexe 4 permet d'avoir d'avantage d'informations par rapport aux plateformes sur lesquelles il est possible de déployer des chatbots suivant chaque outil que nous étudions dans ce document.

L'avantage de tous les éléments que nous avons abordé dans ce chapitre est qu'ils nous permettent non seulement d'avoir une vue d'ensemble sur les outils, mais aussi de faire une comparaison entre ces différents outils et de finalement choisir celui qui conviendra le mieux à ce qu'un développeur souhaite utiliser suivant son budget, son temps et ses compétences pour construire son Chatbot.

À noter que ce rapport a été réalisé dans le domaine de l'informatique, c'est-à-dire un domaine qui évolue constamment et rapidement, ce qui pourrait causer une certaine obsolescence des différents éléments que nous avons vus, au-delà d'une certaine période.

## 10 Conclusion et recommandation

---

Pour conclure cette partie théorique, d'après le rapport de l'université de Stanford, les agents intelligents ont un potentiel pour améliorer considérablement certains services, réduire des coûts financiers, et ainsi augmenter notre qualité de vie ([University, 2015](#)). En effet, ce rapport nous montre que l'introduction de ces agents dans nos foyers, loisirs et même place de travail, nous permettrait de nous délester de plusieurs tâches dont les AI se chargerait et, par conséquent, d'augmenter notre efficacité et notre temps d'apprentissage pour les nouvelles langues, applications et même dans d'autres domaines. Je suis d'avis que ce serait une révolution avec son lot de risques qu'il faudra gérer certes, mais resterait très bénéfique tout comme la révolution de l'Internet à son époque.

Concernant les Chatbots, malgré les progrès de l'intelligence artificielle, nous sommes encore loin des agents intelligents totalement autonomes. En effet, comme on peut le voir dans une page internet de Clustaar, la plupart des réponses des Chatbots doivent encore être fournies par l'homme et un opérateur doit être prêt à prendre le relais du Chatbot si les questions sont trop complexes pour le client/l'utilisateur ([Clustaar, 2016](#)).

En analysant les différentes comparaisons faites plus haut, on s'aperçoit que les outils sont plus ou moins similaires. Cependant, après avoir testé ces outils, API.AI et Chatfuel sont plus intuitifs et moins compliqués à utiliser. Microsoft Bot Framework et Ibm Developer Cloud demandent un peu plus de configurations. Personnellement, je recommanderais API.AI à une personne qui n'a pas forcément des connaissances en programmation et qui souhaiterait construire rapidement un Chatbot doté d'une intelligence artificielle pour interagir avec des utilisateurs. Dans un premier temps, il faudrait modéliser le ou les flux de dialogues et d'interactions qu'on souhaiterait développer pour notre Chatbot. Ensuite, choisir le domaine pour lequel le Chatbot sera utilisé (temps, sport, commercial, traduction, etc.). Si on souhaite faire des flux plus personnalisés ou faire notre propre domaine (c'est-à-dire un domaine qui n'est pas proposé par l'outil), dans ce cas, après avoir fait notre modélisation, il faut créer les variables (*entities*) qui seront utilisées dans les conversations. Par exemple pour l'entité @artist, on peut lui assigner différentes valeurs et plusieurs synonymes pour chacune des valeurs. Ces variables permettront au Chatbot de récupérer les éléments dont il a besoin comme le nom de la musique qu'il doit rechercher ou encore le nom de l'artiste. Ensuite, il faut configurer les *intents* (ou intentions en français). Après avoir testé certains outils, j'ai remarqué que, tout comme avec Chatfuel, la construction d'un Chatbot avec API.AI suivant un scénario précis reste relativement simple. La méthode est totalement graphique et le développeur bénéficie d'une grande aide, notamment grâce à la documentation de ces outils et des internautes relativement nombreux.



## Partie pratique

**HEG Arc - Haute école Arc - Gestion**

**Mention :** Travail de Bachelor d'informaticien de gestion

**Module :** 656-1.1 Thèse de bachelor

**Sujet :** Exploration dans le domaine des agents intelligents et des chatbots

**Directeur de travail :** Frischer Boris

**Auteur :** Klett Boris

**Cycle d'étude :** 2013 - 2017

**Date de restitution :** Septembre 2017



# 1 Introduction

---

À présent que la définition et l'analyse des agents intelligents et des chatbots ont été effectuées, portons maintenant l'attention sur la réalisation pratique du projet Gestarcbot.

En effet, la présente partie pratique du rapport présente les éléments pratiques utilisés afin de mettre en place un chatbot pour la Gest'arc afin de pallier à notre problématique.

Dans cette partie, nous allons dans un premier temps, parler de l'analyse des besoins de l'association des étudiants, quant à l'utilisation d'un chatbot sur leur page Facebook. Les différents outils, les bonnes pratiques et les technologies, qui vont être utilisés tout au long de ce projet, vont également être abordés ainsi que les différentes étapes de développement.

Les analyses effectuées dans la partie théorique seront utilisées afin d'avoir une direction à suivre quant à nos différents choix technologiques, que ce soit au niveau de la plateforme sur laquelle le chatbot sera intégré pour pouvoir discuter avec les utilisateurs ou encore au niveau du framework à utiliser pour développer le chatbot.

Ce document a donc comme objectif d'expliquer le choix de la solution retenue face à notre problématique et d'expliquer son implémentation.

Enfin, une conclusion sera faite. Une conclusion dans laquelle, j'expliquerai le déroulement de ce travail, les problèmes rencontrés ainsi que les solutions apportées pour pallier à ces problèmes. Je donnerai également mon avis personnel sur l'implémentation et l'utilisation des agents intelligents de type chatbot au lieu des applications de gestion classiques.

Dans le prochain chapitre, je vais effectuer une analyse des besoins. Cette analyse me permettra de savoir en quoi l'implémentation d'une solution de type chatbot serait utile aux différents utilisateurs, face à une problématique de gestion.

## 2 Analyse des besoins

---

Pour rappel, mon client dans ce projet est la Gest'arc, donc la solution que je souhaite implémenter doit convenir à leurs besoins. Cependant, comme nous avons vu dans le chapitre problématique de la partie théorique, cette solution doit pouvoir aider la Gest'arc à augmenter leurs disponibilités sur Facebook Messenger. Par conséquent, il faut également satisfaire certains besoins des utilisateurs finaux du chatbot, qui sont principalement des étudiants de la Haute école de gestion, présentant un intérêt pour la Gest'arc et particulièrement pour les évènements qu'elle organise.

Ces utilisateurs finaux peuvent avoir besoin de contacter la Gest'arc pour diverses raisons, notamment pour s'inscrire comme nouveau membre de la Gest'arc, réserver

un ticket pour un évènement ou encore avoir des informations sur des évènements organisés par l'association ou qu'elle prévoit d'organiser. C'est donc à eux que le chatbot est réellement destiné et doit correspondre. D'autre part, le chatbot doit également répondre aux besoins de la Gest'arc. Afin de comprendre les besoins de l'association et de toute autre personne qui en présenterait un intérêt, voici quelques *user stories*<sup>15</sup> :

*En tant que membre/collaborateur de la Gest'arc, je souhaite que le chatbot puisse nous remplacer efficacement pour pouvoir répondre à certaines questions sur Facebook Messenger car nous disposons actuellement d'un effectif réduit pour répondre rapidement et efficacement à toutes les questions que nous recevons sur Facebook Messenger.*

*En tant que membre de la Gest'arc, je souhaite être délesté des demandes d'inscription comme nouveau membre ainsi que les réservations des utilisateurs pour participer à un évènement qui sont faites par des étudiants, via Facebook Messenger afin de pouvoir avoir davantage de temps pour réaliser d'autres activités dans l'association.*

*En tant qu'utilisateur du chatbot de la Gest'arc, je souhaite pouvoir avoir des informations sur un évènement organisé par la Gest'arc très rapidement et à n'importe quel moment.*

*En tant qu'utilisateur du bot, je souhaite que le chatbot puisse enregistrer ma demande d'inscription en tant que membre de la Gest'arc et m'éviter de devoir contacter la Gest'arc uniquement lors de leurs disponibilités.*

*En tant qu'utilisateur du chatbot, je souhaite que celui-ci puisse me réserver un ticket pour un évènement organisé par la Gest'arc pour que je n'ai pas besoin de contacter un membre de la Gest'arc pour le faire.*

A partir de ces user stories, il est possible d'imaginer ce que tout utilisateur attend finalement de la solution finale choisie. Les cas d'utilisation ci-dessous, relatent les services attendus par ces utilisateurs.

---

<sup>15</sup> Dans les méthodes agiles, un récit utilisateur ou user story est une phrase simple dans le langage de tous les jours permettant de décrire avec suffisamment de précision le contenu d'une fonctionnalité à développer (Wikipedia, n.d.).

## 2.1 Cas d'utilisation

Les besoins des utilisateurs et indirectement ceux des membres de la Gest'arc, peuvent être regroupés en trois cas d'utilisation principaux.

Le but premier ici est de pallier à la problématique dont nous avons vu dans la partie théorique. En effet, nous cherchons à donner la possibilité à la Gest'arc de répondre très rapidement aux sollicitations des étudiants ou non, via Facebook Messenger.

Par rapport aux différentes user stories que nous avons vues plus haut dans ce chapitre, le chatbot devra donc être capable d'informer les utilisateurs sur un évènement, de réserver une ou plusieurs entrées à un évènement et enfin enregistrer une demande d'inscription en tant que membre de la Gest'arc, pour un étudiant. En offrant ces services, le chatbot permettra alors aux membres de la Gest'arc de se concentrer sur d'autres activités tout en augmentant les disponibilités de l'association vis-à-vis des étudiants. Pour cela, j'ai réalisé trois cas d'utilisation qui représentent les trois fonctionnalités de ma solution. Il s'agit, de :

- Demande d'information sur un évènement
- Demande d'inscription comme membre de la Gest'arc
- Réservation d'un ticket d'entrée à un évènement.

Un peu plus en détail, le cas d'utilisation « demande d'informations sur un évènement » englobe certaines demandes d'informations que pourraient faire des étudiants. Il s'agit de vouloir connaître : le prix, le lieu, la date, les horaires et le thème d'un évènement. Sans oublier les photos réalisées par la Gest'arc lors d'un évènement dans le cas où celui-ci aurait déjà eu lieu. Ce cas d'utilisation permettra au chatbot de couvrir certaines des user stories que j'ai présentées plus haut dans ce chapitre. En effet, grâce au cas d'utilisation « demande d'informations sur un évènement », le bot permettra à l'utilisateur d'avoir l'information dont il désire à n'importe quel moment de la journée, même de la nuit et de manière rapide.

Etant donné que je vais commencer ce projet de zéro, c'est-à-dire que je partirai vraiment d'aucune base, de faire en sorte que notre chatbot puisse tout faire lui-même et de manière optimale, me demanderait énormément de temps. Alors j'ai essayé de trouver des solutions. En effet, dans le cas d'utilisation « Enregistrer une demande d'inscription », le chatbot se contentera donc d'enregistrer la demande de l'étudiant dans une table pour qu'un membre de la Gest'arc puisse par la suite, valider et confirmer l'inscription à l'étudiant. Même situation pour le cas d'utilisation « Réservation d'un ticket », pour l'instant le chatbot réservera simplement un ou des tickets d'entrées à un évènement pour une personne pendant un lapse de temps. Le chatbot n'encaissera donc pas les paiements des utilisateurs pour les évènements payants. En tout cas, pas pour l'instant !

Grâce aux user stories que nous avons vues, il est donc fort probable que ces fonctionnalités satisfassent les différents besoins non seulement des utilisateurs mais aussi ceux des membres de la Gest'arc. Etant donné que le chatbot sera en mesure d'enregistrer les demandes d'inscription à la Gest'arc des étudiants grâce au cas d'utilisation « enregistrer une demande d'inscription », cela sera alors un réel avantage autant pour les membres de l'association que pour les étudiants. En effet, dans un

sens les membres de la Gest'arc n'auront plus besoin de prendre les demandes des étudiants eux-mêmes sur leur page Facebook, ce qui leur permettra donc d'utiliser ce temps sur d'autres tâches importantes par exemple. De plus, tout comme avec le cas « réservation d'un ticket » où il sera possible de réserver des tickets pour un évènement, l'étudiant pourra aussi faire une demande d'inscription grâce cas « demande d'inscription comme membre de la Gest'arc ». Le gros avantage est que, l'étudiant aura la possibilité de faire ces demandes à n'importe quel moment de la journée ou de la nuit, sans avoir la contrainte de contacter la Gest'arc uniquement pendant ses disponibilités.

Le diagramme de cas d'utilisation ci-dessous, permet d'illustrer les différents cas d'utilisation du chatbot. C'est une représentation très simplifiée de ses fonctionnalités.

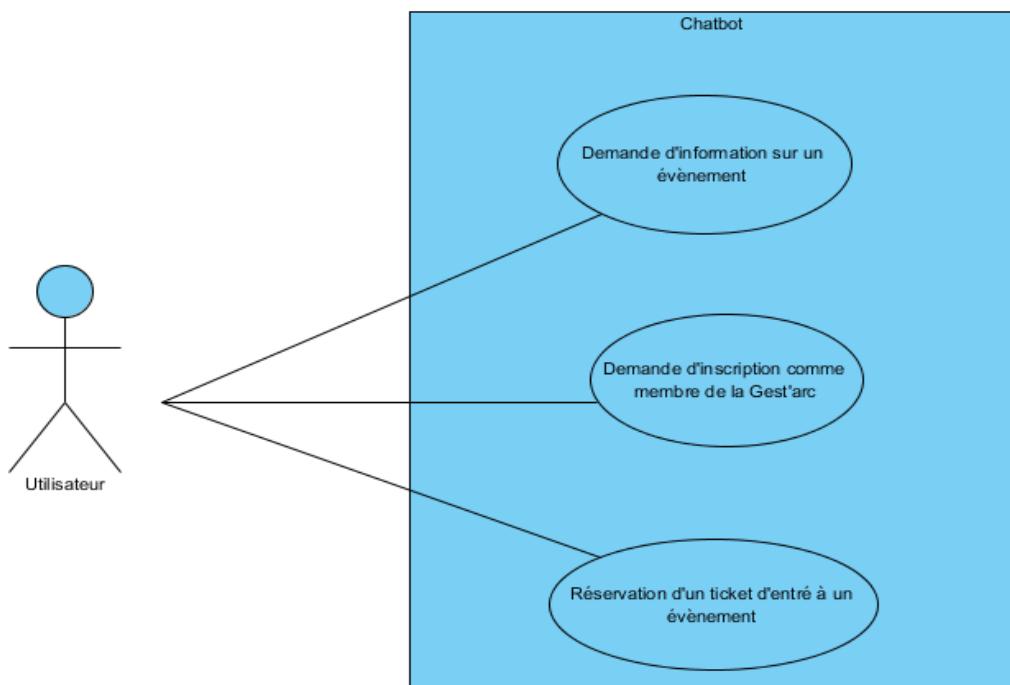


Figure 13 : Diagramme des cas d'utilisation de la solution choisie, le chatbot.

Maintenant que nous connaissons les attentes de la Gest'arc et des futurs utilisateurs de notre solution, nous allons dans le prochain chapitre analyser le futur chatbot. Cette analyse aura pour but de me guider dans le développement de la solution finale

### 3 Analyse du chatbot

---

#### 3.1 Bonnes pratiques

De nos jours, il existe de plus en plus de bonnes pratiques « best practices » pour développer un chatbot. Des concepts dont je vais me servir, pour pouvoir développer un chatbot pour la Gest'arc.

En effet, selon Google developers, les étapes les plus importantes et à réaliser dans un premier temps lors du développement d'un chatbot, sont :

- Choisir le domaine d'activité du chatbot
- Définir une personnalité pour le chatbot
- Ecrire des dialogues homme/machine (*Developers, 2017*)

##### 3.1.1 *Le domaine d'application ou d'activité de notre chabot*

Notre chatbot est mis en place pour communiquer avec des utilisateurs sur Facebook Messenger. Ces utilisateurs sont principalement des étudiants de la HEG-arc ou non, qui présentent un certain intérêt pour la Gest'arc et ses évènements.

Par rapport à l'analyse des besoins et des cas d'utilisation que j'ai présenté plus haut dans cette partie du document, le chatbot sera donc actif pour renseigner et aider les gens vis-à-vis de la Gest'arc et de ses évènements. Pour y parvenir, le bot proposera dans un premier temps trois services ou fonctionnalités :

5. Acquérir une information sur un évènement
6. Réserver un billet d'entrée à un évènement
7. Devenir membre de la Gest'arc.

Non seulement le chatbot devra offrir ces services, il devra également faire preuve d'une certaine forme de personnalité, ce qui permettra à son interlocuteur d'avoir l'impression de dialoguer avec une personne réelle. Dans le point suivant, nous verrons la personnalité dont notre chatbot sera doté.

##### 3.1.2 *Définir une personnalité pour le chatbot*

Il est important d'attribuer une personnalité au chatbot, car l'utilisateur le ressentira et aura alors l'impression de discuter avec un véritable être humain. Pour respecter ce point, j'ai dans un premier temps donné un nom au chatbot, ce qui lui permettra de se présenter plus facilement lors des discussions. Notre chatbot s'appellera donc « Gestarcbot ».

À travers les dialogues entre le chatbot et une personne, je chercherai de simuler une personnalité pour Gestarcbot. En effet, il devra simuler les traits de caractère et la personnalité suivants :

Gestarcbot sera toujours optimiste, joyeux et encourageant face aux étudiants même lorsqu'ils n'ont pas pu avoir l'information qu'ils souhaitaient. Il sera également engagé à aider dans la mesure du possible, compte tenu de ses compétences, lorsque des étudiants le souhaiteront. Etant donné que la population étudiante englobe des jeunes et des moins jeunes, Gestarcbot sera plutôt informel et utilisera un langage simple pour atteindre un plus large groupe de personnes.

Le but de cette manœuvre est de donner à l'utilisateur la sensation d'être pris en charge par un vrai membre de la Gest'arc. C'est également un moyen de donner à l'utilisateur une certainement forme de satisfaction en dépit du service rendu par Gestarcbot.

Maintenant que le cadre de l'étendu du chatbot est fixé, il est important d'avoir une vision sur la forme des conversations entre Gestarcbot et les mobinautes. Pour cela, la rédaction des scénarios est la méthode idéale.

### 3.1.3 *Ecrire les scénarios*

C'est pour moi, la phase la plus importante et l'une des moins évidentes. Le chatbot n'a pas d'interface graphique comme les sites web ou d'autres applications plus classiques, sa qualité sera donc jugée sur sa capacité à tenir une conversation et à apporter des informations pertinentes quant aux requêtes de l'utilisateur. Par conséquent, j'ai dû réaliser un gros travail sur des scénarios avant de pouvoir débuter le développement de notre chatbot. Selon l'article de T.Cantet, il faut savoir que plusieurs éléments sont à respecter quand un chatbot converse avec un humain. Pour rappel, ces points à respecter sont donc :

- ✓ Ne pas mentir aux utilisateurs
- ✓ Humaniser le chatbot
- ✓ Se présenter
- ✓ Pas trop bavard
- ✓ Expliquer qu'une réponse est attendue
- ✓ Ne pas présumer que l'utilisateur sait ce qu'il doit faire
- ✓ Ne pas toujours demander de confirmation
- ✓ Obtenir une information à la fois
- ✓ Gérer les impasses
- ✓ Gérer le contexte ([Cantet, 2016b](#))

Fort de ces concepts, j'ai rédigé un scénario pour chacun des cas d'utilisation que nous avons vu précédemment, pour me permettre d'avoir un fil conducteur pour l'interaction homme/machine. Plus tard, il sera alors question d'étendre ces interactions et ajouter des alternatives aux requêtes des utilisateurs ainsi qu'aux réponses et actions du chatbot. Pour le premier cas d'utilisation, « recherche d'information », j'ai rédigé un scénario positif et un négatif. J'utilise donc le terme positif dans le cas où le chatbot réussit à fournir à l'utilisateur ce que celui-ci désire. Le sens négatif est utilisé ici pour désigner le fait que le chatbot ne parvienne pas à fournir à

l'utilisateur, ce qu'il souhaite. Un scénario négatif pourrait survenir à cause de différentes raisons, notamment : un problème de connexion entre le chatbot et la base de données, une violation de règle métier ou tout simplement parce que l'information n'existe pas en base de données. Le chatbot sera alors contraint de bien gérer l'impasse. Voici les scénarios sur lesquels je compte m'appuyer, pour implémenter Gestarcbot :

**Scénario pour la recherche d'information (positif) :**

L'interaction consiste donc à fournir à l'utilisateur l'information dont il souhaite avoir, sur un évènement qui a déjà eu lieu ou pas encore.

**Utilisateur :** Bonjour j'aimerais savoir où est ce que je peux trouver les photos de l'évènement de la semaine passée.

**Gestarcbot :** Salut oui bien-sûr mais de quel évènement exactement ? Par ce qu'il y en a eu pas mal.

**Utilisateur :** Celui de jeudi soir ?

**Gestarcbot :** La Secomania à la case à choc ou la sortie chez Swatch group ?

**Utilisateur :** Swatch group.

**Gestarcbot :** Ok ça marche. Bah tiens le lien Facebook tu y trouveras toutes les photos que nous avons faites (<https://www.facebook.com/photo.php?fbid=1926993754242059&set=t.100007943583575&type=3&theater>).

**Utilisateur :** Ok merci

**Gestarcbot :** Avec plaisir Véronique. Est-ce que tu as besoin de quelque chose d'autre du coup ? réserver un ticket pour un évènement ou alors devenir un nouveau membre de la Gest'arc ! ça serait cool (y) ?

**Utilisateur :** Non c'est tout bon merci

**Gestarcbot :** Ok ça marche. Alors une bonne soirée et à bientôt.

**Scénario pour la recherche d'information (négatif) :**

**Utilisateur :** Coucou.

**Gestarcbot :** Salut, est ce que je peux t'aider ?

Et si jamais, je suis Gestarcbot un chatbot mis en place sur Facebook Messenger, pour répondre rapidement aux demandes des gens. Je suis donc capable de te donner des renseignements et réserver des tickets pour des évènement, et également prendre une demande d'inscription pour devenir membre de la Gest'arc. ☺

**Utilisateur :** Oui j'aimerais savoir où est ce que je peux trouver les photos de l'évènement de la semaine passée.

**Gestarcbot :** oui bien-sûr mais de quel évènement exactement ? Parce qu'il y en a eu pas mal.

**Utilisateur :** Celui de samedi passé ?

**Gestarcbot :** La Secomania ?

**Utilisateur :** Non. Je ne crois pas que c'était ça !

**Gestarcbot :** Ok ! bah désolé je n'ai que ça comme évènement de samedi dernier. Du coup est ce que tu aimerais que je demande directement à un membre de te contacter pour directement voir avec lui ?

**Utilisateur :** Non c'est bon pas besoin.

**Gestarcbot :** Ok pas de soucis. Est-ce que tu as besoin de quelque chose d'autre du coup ? réserver un ticket pour un évènement ou alors devenir un nouveau membre de la Gest'arc peut-être ?

**Utilisateur :** Non c'est tout bon merci

**Gestarcbot :** Ok ça marche. Alors une bonne soirée et à bientôt.

**Scénario pour la réservation de ticket pour un évènement**

**Utilisateur :** Hello est ce que c'est possible de réserver des tickets pour la soirée de vendredi prochain ?

**Gestarcbot :** Oui bien sûr. J'aurais juste besoin de ton prénom s'il te plaît.

**Utilisateur :** Véronique.

**Gestarcbot :** Brumann c'est juste ?

**Utilisateur :** Oui

**Gestarcbot :** Ok et tu veux combien de ticket Véronique ?

**Utilisateur :** 6.

**Gestarcbot :** Ha mince il ne reste plus que 3 tickets réservable. Est-ce que tu veux quand même les prendre ?

**Utilisateur :** Ha mince... oui oui

**Gestarcbot :** Ok je te les réserve tout de suite véronique. Et si jamais tu tiens toujours à avoir tous les tickets, tu peux aussi passer au secrétariat pour te renseigner s'il y en a encore et y acheter le reste d'accord ?

**Utilisateur :** ça marche.

**Gestarcbot :** Alors ça te fera 45 francs si jamais ! Juste encore une petite précision je te réserve des tickets mais par contre tu devras aller les payer auprès de la Gest'arc dans le bâtiment de la HEG ou alors au secrétariat de l'école avant vendredi à 18h si non les tickets seront remis en vente.

**Utilisateur :** Ok merci.

**Gestarcbot :** Pas de quoi ! si non tu aimerais quelque chose d'autre ?

**Utilisateur :** Non c'est tout bon merci.

**Gestarcbot :** Ok ça marche. Alors une bonne soirée et surtout n'hésite pas à écrire si tu as besoin de renseignement.

**Utilisateur :** ok ça marche !

**Gestarcbot :** 😊

**Scénario pour la demande d'inscription comme membre de la Gest'arc**

**Utilisateur :** Bonjour.

**Gestarcbot :** Hello, est ce que je peux t'aider ?

Si jamais, je suis Gestarcbot un chatbot mis en place sur Facebook Messenger, pour répondre rapidement aux demandes des gens. Je suis donc capable de donner des renseignements et réserver des tickets pour des évènement, et également prendre une demande d'inscription pour devenir membre de la Gest'arc. 😊

**Utilisateur :** Oui j'aimerais savoir comment faire pour devenir membre de la Gest'arc ?

**Gestarcbot :** Ha mais c'est simple je peux faire une demande pour toi si tu veux.

**Utilisateur :** Ha super. C'est cool.

**Gestarcbot :** Alors si ça t'intéresse, j'aurai juste besoin de ton prénom ?

**Utilisateur :** c'est Véronique mon prénom.

**Gestarcbot :** Et ton nom c'est comment ?

**Utilisateur :** Brumann ?

**Gestarcbot** : Du coup ton adresse de l'école est bien [veronique.brumann@he-arc.ch](mailto:veronique.brumann@he-arc.ch) ?

**Utilisateur** : Oui

**Gestarcbot** : Ok Véronique j'ai enregistré ta demande pour devenir membre de la Gest'arc. Si jamais un membre de la Gest'arc te confirmera ton adhésion par email, dans les prochains jours.

**Utilisateur** : Ok génial ! merci beaucoup.

**Gestarcbot** : Avec plaisir Véronique. Est-ce que tu as besoin de quelque chose d'autre du coup ? avoir des informations sur un évènement ou réserver un ticket ?

**Utilisateur** : Non c'est tout bon merci

**Gestarcbot** : Ok ça marche. Alors une bonne soirée et à bientôt.

Il faudra également envisager la possibilité de passer directement d'un service à un autre, par exemple demander des informations sur un évènement programmé avant de réserver un ticket pour le même évènement.

Le scénario ci-dessous démontre une possibilité qu'aura l'utilisateur de passer de la recherche d'informations à la réservation d'un ticket.

#### **De la recherche d'information à la réservation d'un ticket**

**Utilisateur** : Bonjour !

**Gestarcbot** : Salut, est ce que je peux t'aider ?

Et si jamais, je suis Gestarcbot un chatbot mis en place sur Facebook Messenger, pour aider les gens sur les évènements organisés par la Gest'arc. Je suis donc capable de donner des renseignements et réserver des tickets pour des évènement, et également prendre une demande d'inscription pour devenir membre de la Gest'arc. ☺

**Utilisateur** : Oui, j'aimerais avoir des renseignements sur la soirée de vendredi prochain

**Gestarcbot** : Oui pas de soucis. Mais quoi comme renseignement l'heure, le lieu, le prix ou le thème ?

**Utilisateur** : je voulais savoir déjà le thème de la soirée.

**Gestarcbot** : Le thème sera les années 80

**Utilisateur** : ah oui super ! ça m'intéresse.

**Gestarcbot** : Cool 😊. Du coup est ce que tu as déjà un ticket pour la soirée ?

**Utilisateur** : Non pas encore. C'est combien ?

**Gestarcbot** : Les tarifs sont-> 5 francs pour les membres de la Gest'arc, 15 francs pour les étudiants et 20 francs pour les non-étudiants. Tu aimerais que je t'en réserve ?

**Utilisateur** : Oui.

**Gestarcbot** : Ok ça marche. C'est comment ton prénom ?

**Utilisateur** : Véronique.

**Gestarcbot** : Et ton nom ?

**Utilisateur** : Brumann ?

**Gestarcbot** : Du coup ton adresse de l'école est bien [veronique.brumann@he-arc.ch](mailto:veronique.brumann@he-arc.ch) ?

**Utilisateur** : Oui

**Gestarcbot** : Ok et tu veux combien de ticket Véronique ?

**Utilisateur** : 6.

**Gestarcbot** : Ok ça joue ! Juste encore une petite précision je te réserve des tickets mais par contre tu devras aller payer auprès de la

Gest'arc dans le bâtiment de la HEG ou alors au secrétariat de l'école avant vendredi à 18h si non les tickets seront remis en vente.

**Utilisateur :** Ok merci.

**Gestarcbot :** Pas de quoi ! si non tu aimerais quelque chose d'autre ?

**Utilisateur :** Non c'est tout bon merci.

**Gestarcbot :** Ok ça marche. Alors une bonne soirée et surtout n'hésite pas à écrire si tu as besoin de renseignement.

**Utilisateur :** ok ça marche !

**Gestarcbot :** 😊

Grâce à ces scénarios, j'ai eu une vision bien claire de la manière dont Gestarcbot devra communiquer avec ses interlocuteurs. Il faut tout de même noter qu'il existe un nombre incalculable de possibilités à cause de la complexité de l'être humain. C'est-à-dire que, bien que le souhait d'avoir un renseignement reste le même, deux interlocuteurs ne le feraient probablement pas savoir de la même manière. Cette aspect imprévisible d'une conversation naturelle, ajoute une grosse difficulté à la tâche.

Dans le chapitre suivant, je vais parler et expliquer mes choix quant aux technologies que je vais utiliser pour développer notre bot.

## 4 Choix des technologies

---

Maintenant que nous avons vu les différentes règles à respecter pour développer un bon chatbot, dans ce chapitre, je vais m'attarder sur les choix des technologies que je vais utiliser pour le développer. Cette étape a été relativement simple pour moi car, tout d'abord il faut savoir qu'il existe un grand nombre de plateformes de développement des chatbots selon le site Nospoon, un site web spécialisé dans la construction des chatbots ([Nospoon, 2017](#)). D'après le même site, bien que le principe reste le même, ces outils fonctionnent différemment, certains nécessitent de plus grandes compétences en développement et en codage que d'autres ou disposent d'une liste plus ou moins étendue de plateformes compatibles pour permettre à des personnes d'utiliser le chatbot. Dans le prochain point, je parlerai du choix que j'ai fait pour l'outil pour développer notre chatbot.

## 4.1 Plateforme de développement

Suite à plusieurs recherches et essais sur différentes plateformes de création de chatbot, mon choix s'est naturellement porté vers l'outil API.AI pour des raisons dont j'ai fait part dans la conclusion et recommandation de la partie théorique de ce rapport du travail de bachelor. Il faut savoir que, non seulement API.AI est un outil qui dispose d'une documentation très garnie et complète, il dispose également d'un panel assez étendu d'agents préconstruits permettant de développer rapidement un chatbot comme nous pouvons le voir dans la figure 14 ci-dessous.

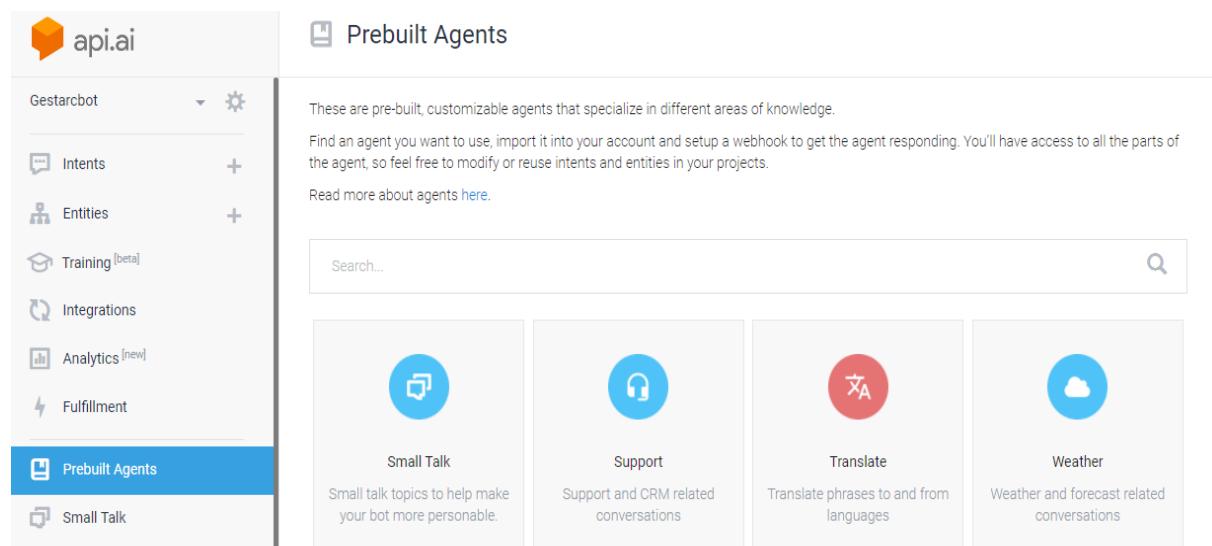


Figure 14 : Liste des agents préconstruits sur API.AI ([API.AI, 2016](#))

C'est donc un outil qui comporte également une très grande liste de plateformes compatibles pour intégrer le chatbot qu'on développe dans un environnement où il sera utilisé comme par exemple : Facebook Messenger, Kik, WhatsApp etc... ([API.AI, 2016](#)). Contrairement à la plupart des plateformes, API.AI est entièrement graphique, un point très important pour tout développeur souhaitant développer un chatbot sans avoir de grosses connaissances dans les lignes de codes ([Nospoon, 2017](#)).

Dans la suite du chapitre, nous aborderons la plateforme d'intégration, c'est-à-dire la plateforme sur laquelle notre chatbot sera utilisé.

## 4.2 Plateforme d'intégration

En ce qui concerne l'environnement sur lequel le chatbot Gestarcbot sera utilisé, en accord avec la Gest'arc, nous avons choisi de l'intégrer à Facebook Messenger. J'ai

choisi cette plateforme car après avoir testé API.AI, je m'étais rendu compte que l'intégration du chatbot API.AI sur Facebook Messenger est non seulement un processus très simple et rapide à réaliser, mais que ça marche parfaitement bien. De plus, selon l'association Gest'arc, la plupart des requêtes leur viennent depuis leur page Facebook.

Toutefois, il ne faut pas oublier qu'un chatbot développé sur API.AI peut être intégré sur plusieurs plateformes à la fois, donc on pourrait aussi bien communiquer avec notre chatbot aussi bien sur Kik, Twitter, WhatsApp que sur Facebook Messenger. N'ayant pas tester les autres plateformes, je ne vais donc pas faire de commentaires là-dessus.

Pour développer un chatbot avec l'outil API.AI, il faut également mettre en place un webhook (service web) qui permettra au chatbot d'utiliser des informations d'une base de données distante (API.AI, 2016). C'est en fait la partie de persistance.

Pour terminer ce chapitre, dans la prochaine partie, je vais parler de la technologie je vais utiliser pour mettre en place la persistance des données dans ce projet.

### 4.3 *Persistance*

Comme nous avons pu le voir dans le point précédent, API.AI utilise un webhook pour sa persistance. Dans ce sous-chapitre, je vais donc faire part de la technologie que j'ai choisi pour développer un webhook et expliquer la raison de mon choix.

Le dernier choix important que j'ai eu à faire est donc le choix de la technologie que je vais devoir utiliser pour mettre en place un webhook pour permettre à API.AI de récupérer des données depuis une base de données distante. Ce système est nécessaire pendant les interactions avec des utilisateurs, car c'est grâce à ça que le chatbot fournira des réponses cohérentes à ses interlocuteurs. Pour cela, Cloud fonctions de google est un très bon candidat. Cet outil est donc intéressant car il permet de développer et de déployer rapidement des services cloud (web), grâce à un environnement sans serveur. De plus, ce framework dispose littéralement d'une fonctionnalité appelée Webhook. Cette fonctionnalité permet de construire des webhooks pour pouvoir répondre à n'importe quelle requête http/S ([Google, 2017a](#)). Mais mon choix s'est de nouveau tourné vers un servlet Java car non seulement c'est une technologie que je connais mieux, elle remplit en plus toutes les conditions imposées et demandées par API.AI. Ces conditions sont les suivantes : le webhook devra être un service web pouvant recevoir et répondre aux requêtes de type POST, ce service devra être accessible depuis internet par le protocole HTTPS et devra répondre en format JSON ([API.AI, 2016](#)).

Dans le prochain chapitre, je vais mettre en exergue l'architecture de notre chatbot.

# 5 Architecture

Maintenant que nous connaissons tous les outils et technologies que je vais utiliser pour mettre en place le chatbot Gestarcbot, je vais dans ce chapitre, présenter l'architecture de ma solution.

Comme l'image ci-dessous l'illustre très simplement, mon architecture se compose de plusieurs éléments, qui sont :

- Le client (Facebook Messenger)
- La plateforme API.AI
- Le webhook
- La base de données (Mysql)

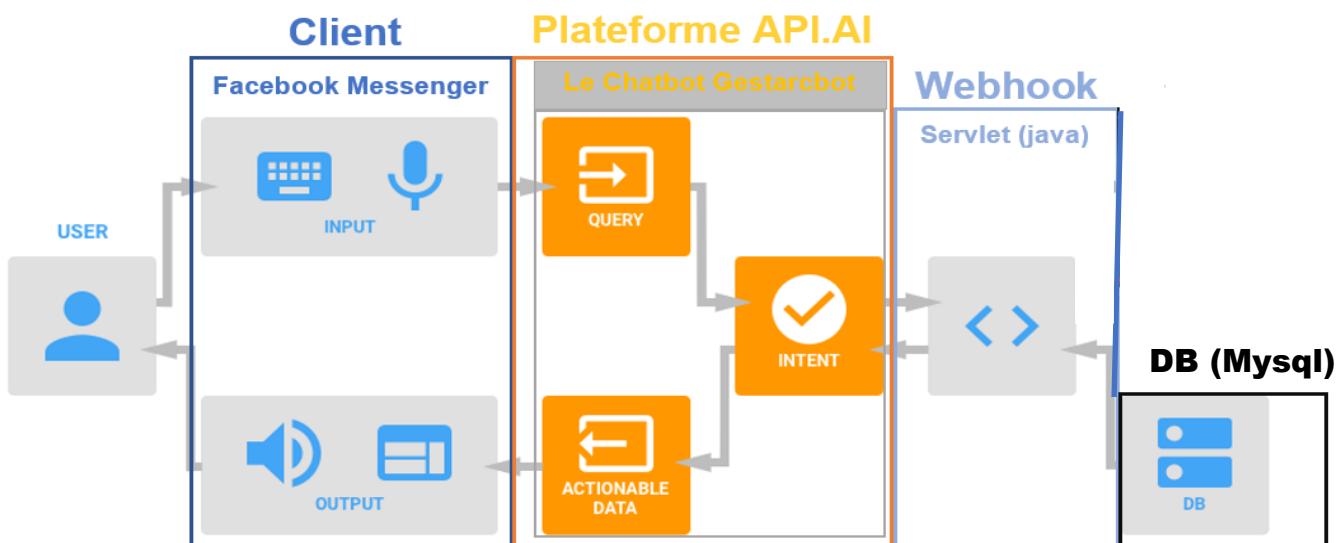


Figure 15 : Architecture du chatbot Gestarcbot ([API.AI, 2016](#))

Pour mieux comprendre l'architecture illustrée dans la figure 15 ci-dessus, attardons-nous maintenant sur chacun des éléments qui la compose ainsi que sur les liens qui existent entre ces éléments. Commençons par l'élément le plus important de l'architecture, la plateforme API.AI.

## 5.1 La plateforme API.AI

La plateforme API.AI de google me permet de développer et de déployer le chatbot en le rendant accessible sur différentes devises compatibles, notamment Facebook Messenger. C'est selon moi, la partie la plus importante de mon architecture car, c'est là que mon agent intelligent est réalisé. C'est également à cet endroit, que les requêtes des utilisateurs sont parsées, interprétées et analysées grâce au machine learning et aux différents éléments tels que les intents, les entities et les contexts, que nous verrons plus tard dans le chapitre 8.1, pour plus de détail. De plus, c'est là que notre chatbot répond de la manière la plus satisfaisante et efficace que possible aux différentes requêtes, comme l'illustre la figure 16 ci-contre.

La figure 16 démontre qu'une fois que API.AI en occurrence le chatbot ait récupéré la requête de l'utilisateur, celui-ci va dans un premier temps essayer de savoir quelle est l'intention de son interlocuteur par rapport au context de la conversation. Après avoir identifié l'intention grâce à l'intent, il va alors essayer de récupérer les informations qui l'aideront dans sa tâche grâce à l'entity, comme par exemple le nom de l'évènement qui intéresse l'utilisateur, des informations sur l'utilisateur comme son nom ou encore son prénom. Enfin, grâce à tous ces éléments récupérés le chatbot va alors générer la réponse ou exécuter l'action adéquate.

Il est également non négligeable de savoir que je n'ai besoin de gérer aucun serveur à ce niveau, pour faire tourner le chatbot. La gestion du serveur est entièrement faite par Google.

À partir de cette plateforme, notre chatbot peut facilement être intégré à différentes plateformes comme Facebook Slack, WhatsApp, Twitter ou encore Facebook Messenger comme c'est le cas pour mon architecture.

## 5.2 Le client (Facebook Messenger)

C'est la partie de l'architecture qui permet à l'utilisateur de communiquer avec notre chatbot. Il s'agit d'une autre plateforme et grâce à API.AI, il est possible ici d'utiliser différentes devises, notamment : Slack, Viber, Twitter, Skype, Kik etc... Suite aux discussions eues avec la Gest'arc, j'ai alors décidé d'utiliser Facebook Messenger pour ce projet. La liaison entre ce client (Facebook Messenger) et la plateforme API.AI se fait grâce à un webhook (Callback URL) généré automatiquement par API.AI

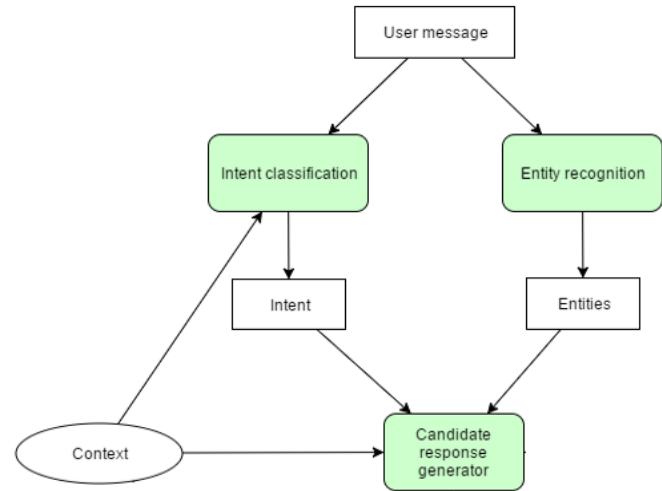


Figure 16 : Processus de reconnaissance des requêtes des utilisateurs sur API.AI (Quora, 2017)

également grâce à un token de vérification qui créent un évènement sur le webhook de Facebook Messenger, à chaque interaction ([API.AI, 2016](#)).

Pour terminer, dans le prochain point de ce chapitre, je vais parler du webhook et de la base de données, car ils sont deux éléments très importants pour mon architecture et qui sont étroitement liés.

### 5.3 Webhook et la Base de données

Grâce au webhook et à la base de données, le chatbot à la possibilité d'effectuer certaines actions et de donner des réponses cohérentes et pertinentes à l'utilisateur. En effet, le webhook est le composant de l'architecture qui permet à la plateforme API.AI de récupérer des informations ou données directement depuis la base de données. Cette partie de l'architecture ne doit pas être négligée, car elle permet au chatbot de donner des réponses qui ont du sens pour l'utilisateur du chatbot, comme par exemple : le prix ou l'heure de la prochaine soirée organisée par l'association. Comme nous le savons, pour récupérer des données d'une base de données, API.AI utilise le système des webhooks accessible depuis internet. La plateforme envoie alors des requêtes https post au webhook pour recevoir une réponse en retour. L'échange se fait au format Json.

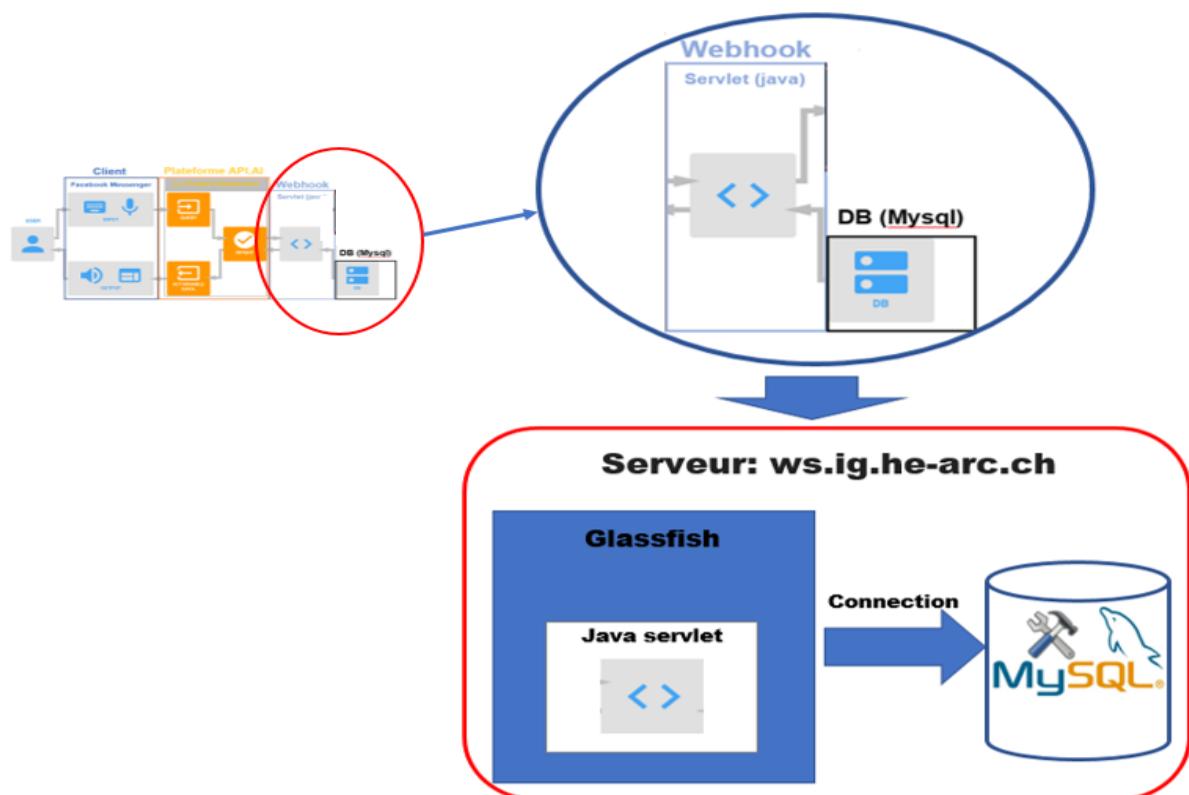


Figure 17 : Représentation de l'architecture du webhook

Comme on peut le voir dans l'image 17 ci-dessus, dans ma situation, j'ai choisi de développer un servlet java, car non seulement il peut être déployé sur internet, il permet également recevoir et de répondre à des requêtes http de type post. Le servlet sera ensuite déployé sur un serveur d'application Glassfish. Glassfish se connectera à ma base de données Mysql grâce à un pool de connexion, ce qui permettra à mon servlet d'utiliser ce pool de connexion et alors récupérer les données dont aura besoin le chatbot. Ma base de données et le serveur Glassfish dans lequel sera déployé le servlet seront déployés sur un serveur « ws.ig.he-arc.ch » distant, doté d'un certificat SSL, ce qui permettra donc à mon servlet et à ma base de données d'être accessibles depuis internet par le protocole HTTPS.

Il s'agit ici d'une architecture dite 3-tiers. En effet, selon une page web spécialisé dans le réseau et les architectures client/serveur, « *Dans l'architecture à 3 niveaux (appelée architecture 3-tier), il existe un niveau intermédiaire entre le client ou demandeur de ressources et le fournisseur de la ressource. C'est-à-dire qu'on a généralement une architecture partagée entre :*

*Un client, c'est-à-dire l'ordinateur, l'application ou le composant demandeur de ressources, équipé d'une interface utilisateur (généralement un navigateur web) chargée de la présentation.*

*Le serveur d'application (appelé également middleware), chargé de fournir la ressource mais faisant appel à un autre serveur.*

*Le serveur de données, fourni au serveur d'application les données dont il a besoin. » (CCM, 2017).*

Si nous revenons sur l'architecture de mon webhook, la plateforme API.AI correspond au demandeur de ressources qui sont fournies par le serveur « ws.ig.he-arc.ch » où ma base de données Mysql est déployée. Le Serveur Glassfish correspond au serveur d'application, la couche intermédiaire entre API.AI et le serveur « ws.ig.he-arc.ch ». Glassfish récupère les données de la base de données depuis le serveur « ws.ig.he-arc.ch » avant de les mettre à disposition de API.AI.

Dans le chapitre 6, je vais présenter plus en détail la structure de ma base de données Mysql.

## 6 Structure de la base de données

---

Pour pouvoir faire en sorte que les réponses et les actions de Gestarcbot soient cohérentes et intéressantes pour les utilisateurs, les données utilisées par le bot sont par conséquent persistées en base de données. Parmi ces données, il y a les évènements avec leurs types et les photos réalisées, les réservations des étudiants aux différents évènements ainsi que leur demande d'inscription comme membre de l'association. Les modèles logique et relationnel ci-dessous permettent de visualiser la structure de ma base de données.

Diagram:	Logical
Author:	boris.klett
Created on:	2017-07-22 12:03:40 UTC
Modified on:	2017-08-10 18:47:14 UTC
Modified by:	boris.klett
Design:	MCD_TB
Model:	Logical

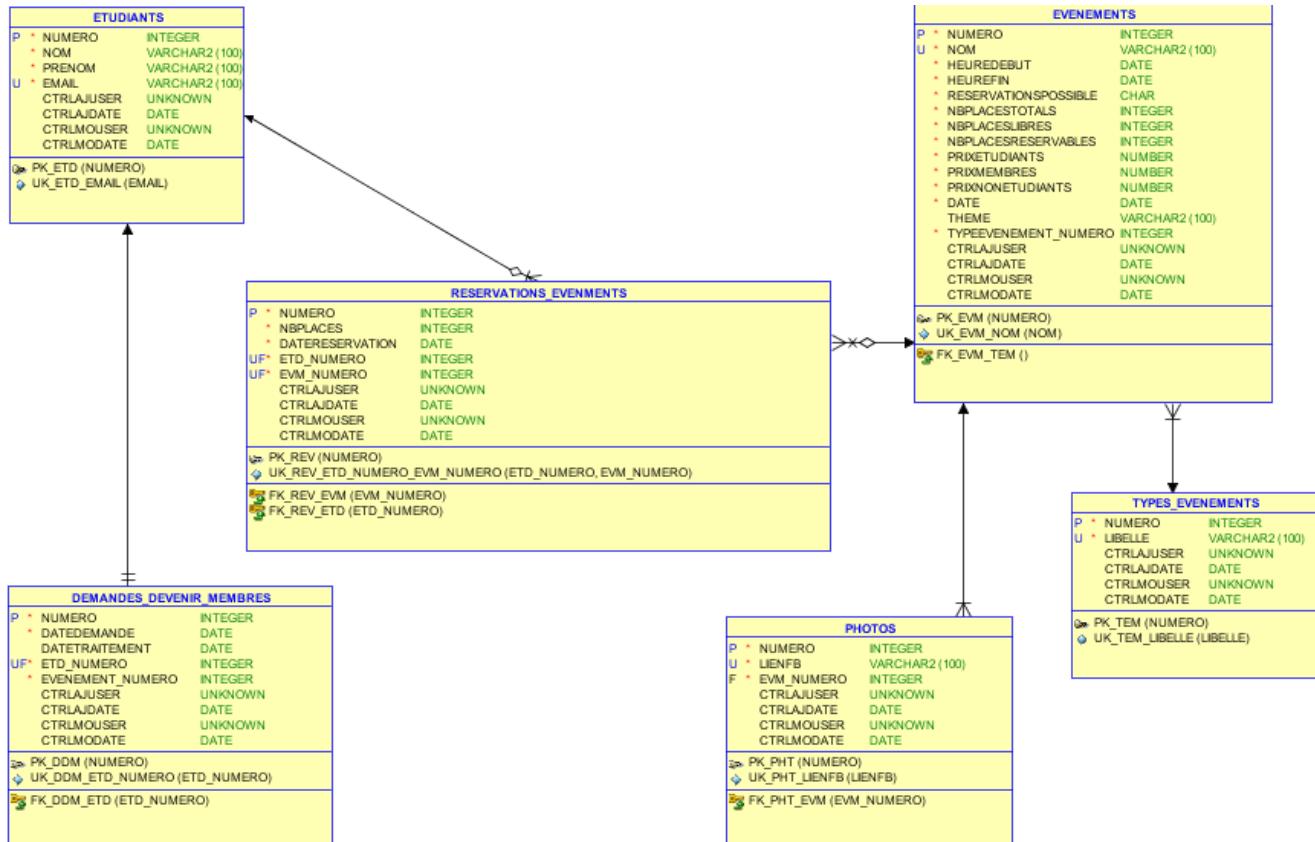
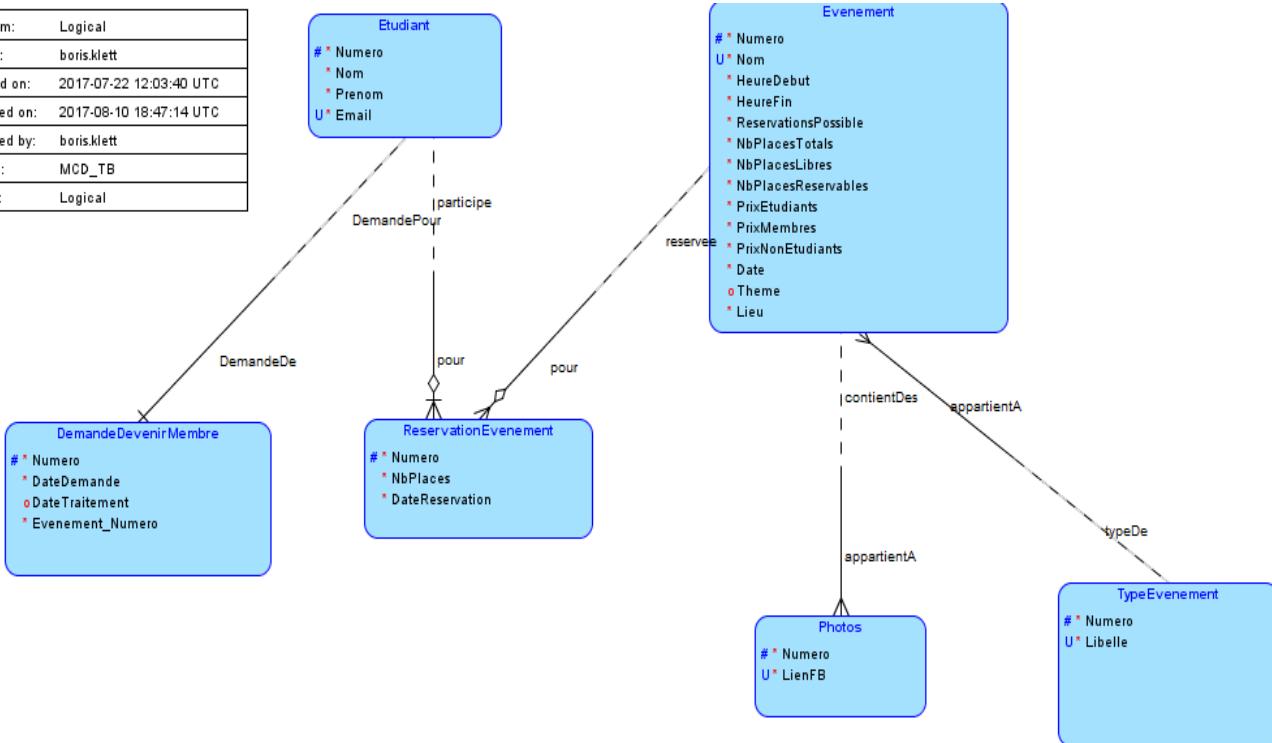


Figure 18 : Modèles logique et relationnel de la structure de la base de données

Pour mieux comprendre les modèles ci-dessus, il faut tout d'abord savoir que les tables principales de la base de données sont les tables : Etudiants et Evenements, pour la simple et bonne raison que, le contenu de ces tables a des valeurs très utiles au chatbot lors de ses interactions avec les utilisateurs. Notamment le nom et l'email de l'étudiant que le chatbot stockera dans la table Etudiants tout au long de ses conversations, mais qu'il utilisera également par la suite pour pouvoir les contacter par email si besoin. Il y a également les différentes informations d'un évènement dont l'utilisateur aura besoin de connaître pour différentes raisons. Les tableaux de la figure 19 ci-contre, permettent de prendre connaissance des différentes informations dont dispose Gestarcbot par rapport à un évènement.

Evenements		Etudiants	
Colonne	Type	Colonne	Type
ID	INTEGER	ID	INTEGER
Nom	VARCHAR2 (100)	Nom	VARCHAR2 (100)
HeureDebut	DATE	Prénom	VARCHAR2 (100)
HeureFin	DATE	Email	VARCHAR2 (100)
ReservationPossible	CHAR		
NbPlacesTotals	INTEGER		
NbPlacesLibres	INTEGER		
NbPlacesReservables	INTEGER		
PrixEtudiants	NUMBER		
PrixMembres	NUMBER		
PrixNonEtudiants	NUMBER		
Date	DATE		
Theme	VARCHAR2 (100)		
ID_TypeEvenement	INTEGER		
Lieu	VARCHAR2 (100)		

Figure 19 : Tables EVENEMENTS et ETUDIANTS

La table DemandesDevenirMembre permet au chatbot d'enregistrer les demandes d'inscriptions que feront des étudiants pour devenir des membres de l'association la Gest'arc. Ces demandes seront traitées directement par un des membres de la Gest'arc. Les attributs qui constituent cette table sont un id unique, l'id de l'étudiant, la date de la demande et la date de traitement. L'unicité des données de la table des Demandes d'inscription est assurée grâce à l'id de l'étudiant. La table ReservationEvenement qui comprend également un id, la date de la réservation, le nombre de places réservées, l'id de l'étudiant qui a fait la réservation et l'id de l'évènement, permet à Gestarcbot d'enregistrer les réservations des utilisateurs pour des évènements. Ici, l'unicité des données est assurée par le couplage de l'id de l'étudiant et de celui de l'évènement. Concernant les photos des évènements, le chatbot fournira uniquement les liens Facebook des photos, qui seront stockés dans la table Photos, par les membres de l'association via une autre application. La table TypesEvenement permet uniquement au chatbot de savoir quel est le type d'un évènement par exemple : une soirée étudiante, une activité sportive ou encore une sortie scolaire.

Avant de développer le chatbot, il faut tout d'abord comprendre son fonctionnement général. Dans le chapitre suivant, je vais expliquer les interactions entre les différents composants de l'architecture ainsi que la façon dont le chatbot Gestarcbot raisonne.

## 7 Mécanisme du chatbot Gestarcbot

---

Dans cette partie, je vais aborder la logique de notre chatbot. C'est-à-dire que je vais expliquer comment notre chatbot raisonne pour pouvoir mener à bien son objectif, qui consiste à fournir ses services. Dans un premier temps, j'ai réalisé une analyse des interactions entre les différents composants de l'architecture.

### 7.1 *Interaction entre les composants de l'architecture*

Pour pouvoir mettre en place Gestarcbot, il est important de comprendre comment toute son architecture fonctionnera. Il faut donc savoir que, pour que le chatbot puisse fournir et exécuter le plus rapidement possible chacun de ses services, Facebook Messenger, API.AI et le webhook devront interagir entre eux pour pouvoir s'échanger les données et réaliser les actions nécessaires. Comme par exemple, Pour réaliser une demande d'inscription comme membre de la Gest'arc, API.AI va interagir et fournir l'email de l'utilisateur au webhook, pour que le webhook puisse enregistrer la demande. Pour pouvoir démontrer et donner la possibilité de mieux comprendre ces interactions, j'ai donc réalisé des diagrammes de séquence pour chacun des services proposés par Gestarcbot, voir l'annexe 5.

Maintenant que j'ai montré et expliqué comment fonctionneront et interagiront les composants de mon architecture, Je vais donc aborder un point tout autant important, la logique de notre chatbot.

## 7.2 Logique de Gestarchbot

Dans ce point du chapitre, je vais présenter la logique qu'aura le chatbot face à chacune des requêtes des utilisateurs. Le fait d'avoir une vision d'ensemble sur la logique du chatbot, me permettra de le développer plus simplement et plus rapidement.



Figure 20 : Processus d'interaction entre les utilisateurs et le chatbot Gestarchbot  
(Quora, 2017)

La figure 20 ci-dessus, démontre le processus de communication entre un chatbot et les utilisateurs. Pour expliquer ce processus de communication, une fois que l'utilisateur ait fait sa requête, le bot va dans un premier temps faire un matching, c'est-à-dire une correspondance entre ce qu'il sait et ce que dit l'utilisateur. C'est ce qu'on pourrait qualifier d'interprétation de la requête de l'utilisateur. Cette étape est représentée ici par les éléments tels que : Intent, Entities, Context et de la base de

données. J'expliquerai ces éléments plus en détail plus tard, dans le chapitre 8. Une fois que le chatbot ait reçu et interprété la requête de son interlocuteur, il peut alors générer la réponse et réaliser l'action adéquate.

Fort du processus dont je viens de présenter, Gestarcbot doit en plus de ça, suivre tout un schéma pour pouvoir réaliser chacun de ses services, comme l'illustre parfaitement la figure ci-dessous.

### Légendes

- Pour le processus du service : Acquérir une information sur un évènement
- Pour le processus du service : Réserver un ticket pour un évènement
- Pour le processus du service : Demande d'inscription comme membre de la Gest'arc

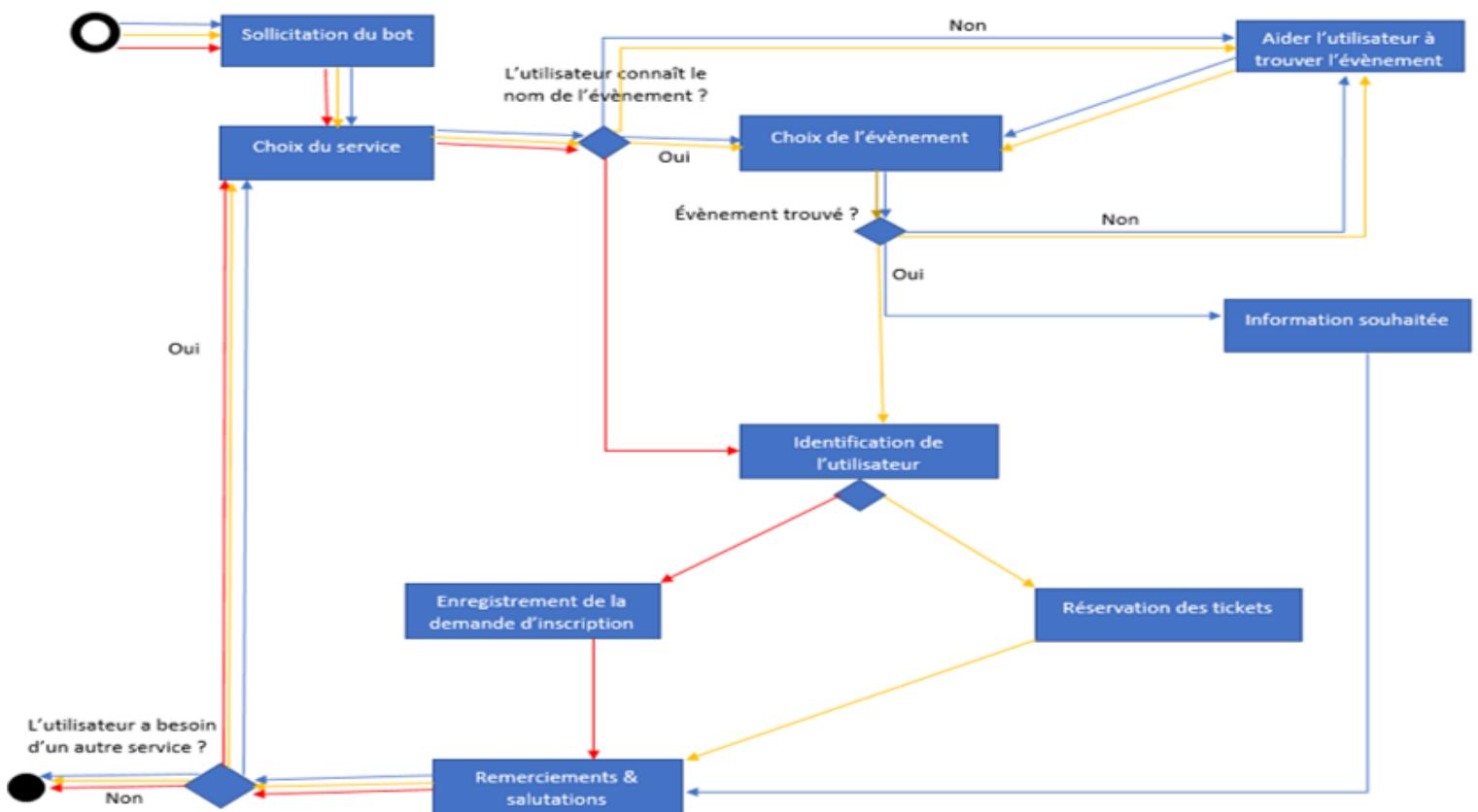


Figure 21 : Schéma de la logique de Gestarcbot

La figure 21 démontre de manière très simple, les différentes étapes du processus de chacun des services de Gestarcbot. Pour mieux comprendre ce schéma, il est important de bien comprendre le code couleur dans un premier temps, comme il est

expliqué dans les légendes de la figure. J'ai décidé d'utiliser un code couleur, pour pouvoir différencier les processus de chaque service du chatbot.

Pour expliquer la logique de Gestarcbot, je vais parler du processus de chacun des services, selon le schéma de la logique de Gestarcbot, que j'ai réalisé ci-dessus. Dans un premier temps, comme je l'ai mentionné plus haut dans ce point, nous savons donc que plusieurs tâches sont communes à tous les services. En effet, pour tous les services, Gestarcbot réagira de la même façon au début et à la fin d'une conversation avec un utilisateur. Pour rendre un service, le chatbot va d'abord être sollicité par un utilisateur, ensuite l'utilisateur va choisir le service qu'il souhaite, grâce à la tâche « choix du service » du processus. Suivant le service choisi par l'utilisateur, différentes tâches du processus doivent alors être réalisées. Une fois que le service choisi par l'utilisateur a été réalisé, le chatbot proposera à nouveau ses services à l'utilisateur pour ainsi perpétuer sa disponibilité.

Il faut savoir que Gestarcbot ne réagit pas de la même manière pour tous ses services, il s'adapte suivant le service choisi par l'utilisateur. Prenons d'abord le cas du service « Demande d'inscription comme membre de la Gest'arc ». Ici, une fois que l'utilisateur ait choisi ce service, le chatbot va alors procéder à l'identification de son interlocuteur, avant de pouvoir enregistrer la demande de l'utilisateur.

En ce qui concerne le service « Acquérir une information sur un évènement », grâce aux tâches « choix de l'évènement » et « aider l'utilisateur à trouver l'évènement », le chatbot va d'abord essayer d'identifier l'évènement qui intéresse l'utilisateur, avant d'essayer de fournir l'information souhaitée.

Quant au dernier service « Réserver un ticket pour un évènement », comme lors du processus du service « Acquérir une information sur un évènement », Gestarcbot va d'abord chercher à identifier l'évènement. Ensuite comme lors du service « Demande d'inscription comme membre de la Gest'arc », le bot identifiera son interlocuteur grâce à la tâche « Identification de l'utilisateur », qui est commune aux deux processus. Enfin, il va alors procéder à la réservation des tickets, grâce à la tâche « réservation des tickets ».

À présent que j'ai expliqué et démontré comment notre chatbot fonctionne, comment il réagit avec l'utilisateur pour pouvoir lui fournir le service que ce dernier a choisi, je suis maintenant prêt à aborder l'implémentation de Gestarcbot.

# 8      Implémentation du chatbot

---

Dans le chapitre précédent, j'ai parlé de la logique de Gestarcbot, un point très important car je vais maintenant expliquer la manière dont j'ai mis Gestarcbot en place et comment j'ai développé sa logique grâce à API.AI. Comme nous avons vu dans le chapitre de l'architecture, seul l'agent intelligent sur API.AI et le webhook nécessite un réel travail d'implémentation de ma part. Dans un premier temps, je vais présenter comment j'ai implémenté et déployé Gestarcbot sur API.AI, ensuite je parlerai du développement du webhook.

## 8.1    *L'agent intelligent Gestarcbot avec API.AI*

Pour implémenter Gestarcbot et lui doter de la logique que j'ai définie, il a donc été indispensable de connaître chacune des étapes du processus de cette logique. Ce qui m'a permis de connaître exactement les éléments dont j'ai eu à utiliser sur API.AI, des éléments qui sont : les intents, entities, context et actions. Maintenant prenons chacun de ces éléments.

### 8.1.1    *Intents*

Selon API.AI de google, une intent ou intention en français, représente une cartographie entre ce qu'un utilisateur dit et quelle action doit être prise par le chatbot.

Pour réaliser cette tâche, les intent doivent comporter plusieurs éléments :

- Des « User says », qui correspondent à des mots ou des phrases, qui permettent à Gestarcbot d'interpréter ce qu'un utilisateur dit et ainsi savoir quel intent déclencher, donc quelle action effectuer.
- Action, permettent au bot d'interagir avec le webhook et lui demander d'effectuer une action précise.
- Réponses, qui correspondent à une liste de mots ou phrases dont le chatbot se servira pour répondre à l'utilisateur.
- Contextes, permettent au chatbot de savoir exactement à quel niveau de la conversation il se trouve ([API.AI, 2016](#)).

Pour mettre en place la logique de Gestarcbot, l'une des premières choses à faire a été de savoir exactement comment chacune des tâches du processus allait fonctionner. En effet, ça m'a alors permis de réaliser tous les intents qui m'ont permis de reproduire la logique sur API.AI et de cartographier les actions de mon bot avec ce que l'utilisateur dit.

Le but de la figure ci-dessous est de démontrer les intents que j'ai dû implémenter sur API.AI pour mettre en place chacune des étapes de la logique de Gestarcbot.

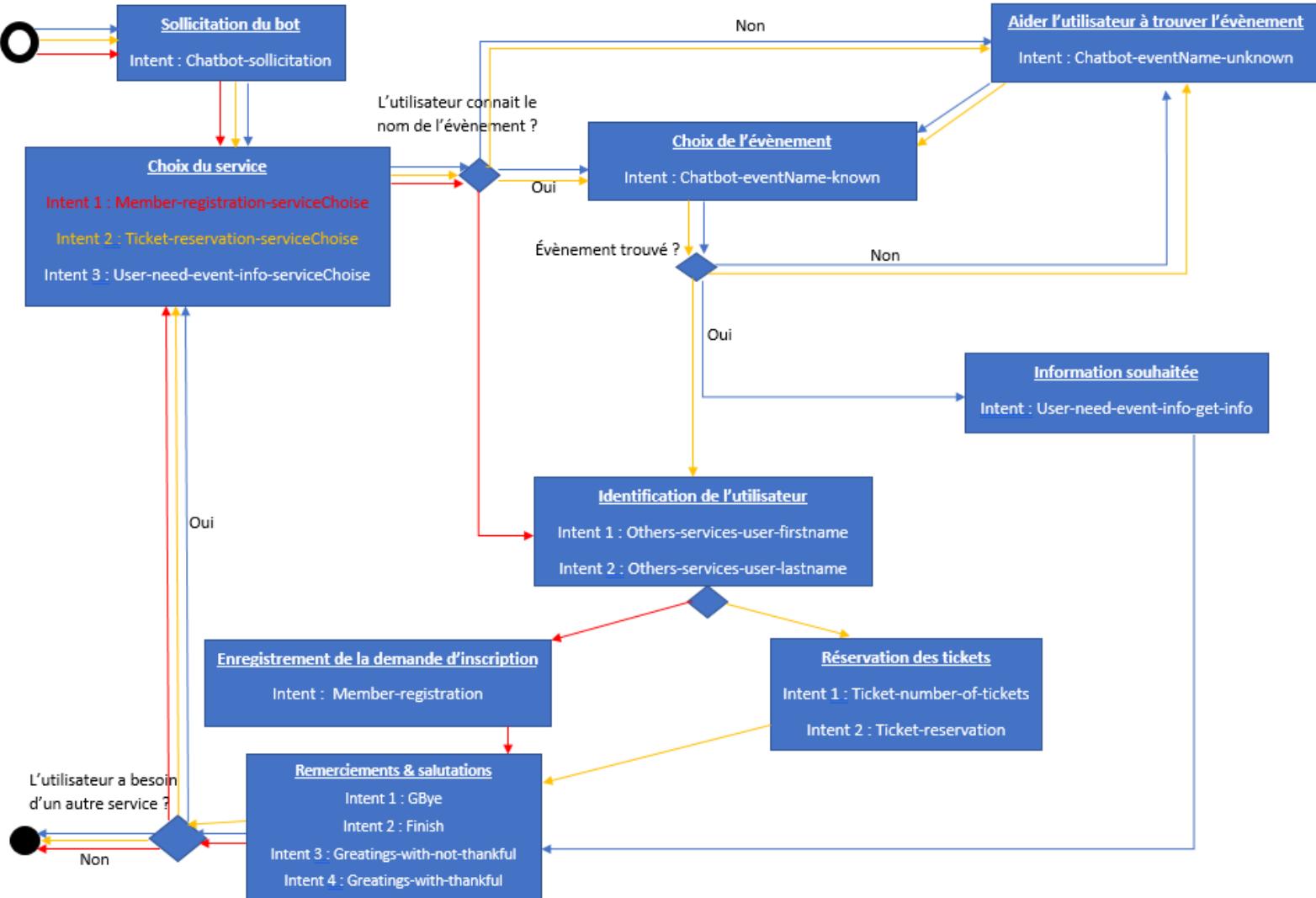


Figure 22 : Représentation graphique des intents de la logique de Gestarcbot

Le code couleur de la figure 22 ci-dessus reste le même que celui utilisé dans la figure 9, de la logique de Gestarcbot. Pour mieux comprendre cette figure, il est essentiel de comprendre certains points. En effet, il faut savoir que pour mettre en place la tâche « Choix du service », j'ai créé 3 intents sur API.AI. Chaque intent correspond à un service et permet alors au chatbot de diriger la conversation vers la réalisation du service correspondant. Par exemple, l'intent « User-need-event-info-serviceChoise » permet au chatbot de savoir que l'utilisateur souhaite acquérir une information, Gestarcbot va donc diriger la conversation dans ce sens. En ce qui concerne la tâche « Identification de l'utilisateur », lorsque le chatbot arrive à ce niveau dans sa conversation avec l'utilisateur, il déclenche dans un premier temps le premier intent pour identifier l'utilisateur. Dans le cas où cela ne suffit pas, il déclenche alors le deuxième intent. Pour la « Réservation des tickets », Gestarcbot déclenche d'abord le

premier intent pour savoir le nombre de tickets souhaité par l'utilisateur, avant de procéder à la réservation, grâce à l'intent 2. En fin, à la fin du processus, si l'utilisateur est reconnaissant du service rendu, l'intent 4 est alors déclencher. Dans le cas contraire, l'intent 3 sera déclenché ce qui permet de simuler une émotion face à la reconnaissance de l'utilisateur. Quel que soit l'intent déclenché, il permet au chatbot de proposer à nouveau ses services à l'utilisateur. Dans le cas où l'utilisateur n'a plus besoin de service, l'intent 1 est alors déclenché pour permettre au chatbot de dire aurevoir à l'utilisateur.

Pour me permettre de définir les intents que j'ai présentés plus haut, j'ai d'abord travaillé sur chacune des tâches. Par conséquent, il faut donc savoir que chaque tâche de la logique de Gestarcbot est dotée d'un sous-processus, qui permet de connaître son mécanisme, de savoir et comprendre chacun des intents qui ont été définis. Pour faciliter la lecture des graphiques ci-dessous, j'ai simplement numéroté les intents, qui correspondent aux intents de la figure 10 plus haut dans ce sous-titre.

### **Sollicitation du bot**

À ce niveau du processus de la logique de Gestarcbot, il y'a également un sous-processus qui montre ce qui se passe dans l'étape. En effet, le sous-processus montre que l'étape « Solliciter le chatbot » est enclenchée lorsque l'utilisateur contacte le chatbot grâce à un mot ou une phrase parmi toute une série à l'annexe 7. Suite à ça, Gestarcbot s'empressera de répondre en se présentant et en proposant ses services à l'utilisateur.

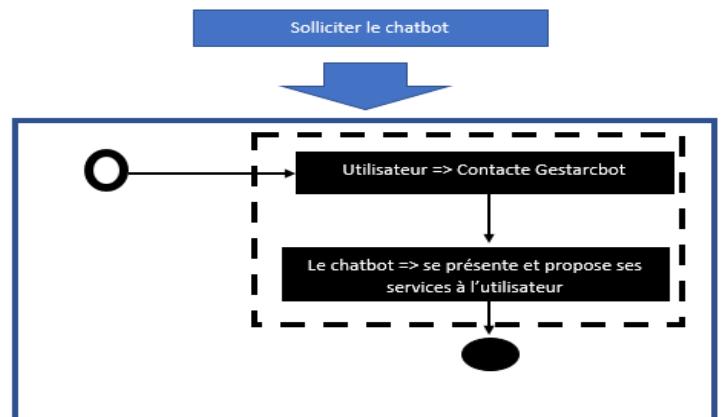


Figure 23 : Sous-processus de la tâche : Solliciter le chatbot

### **Choix du service**

Une fois que la première étape du processus est faite, l'utilisateur choisi alors le service qui l'intéresse. Comme son prédecesseur, cette étape du processus comporte également un sous processus comme l'illustre l'image 24 ci-dessous. Dans ce sous-processus, on peut juste voir que dans cette étape, l'utilisateur demande tout simplement un service à Gestarcbot, qui s'empressera de lui demander le nom de l'évènement qui l'intéresse, dans le cas des services « Acquérir une information sur un évènement » et « Réservation de ticket pour un évènement ». Dans le cas où l'utilisateur demanderait le service « Demande d'inscription comme membre de la Gest'arc », Gestarcbot lui demanderait alors son nom pour pouvoir l'identifier. Pour la tâche « Choix du service », j'ai décidé d'implémenter 3 intents pour me permettre de la réaliser avec API.AI.

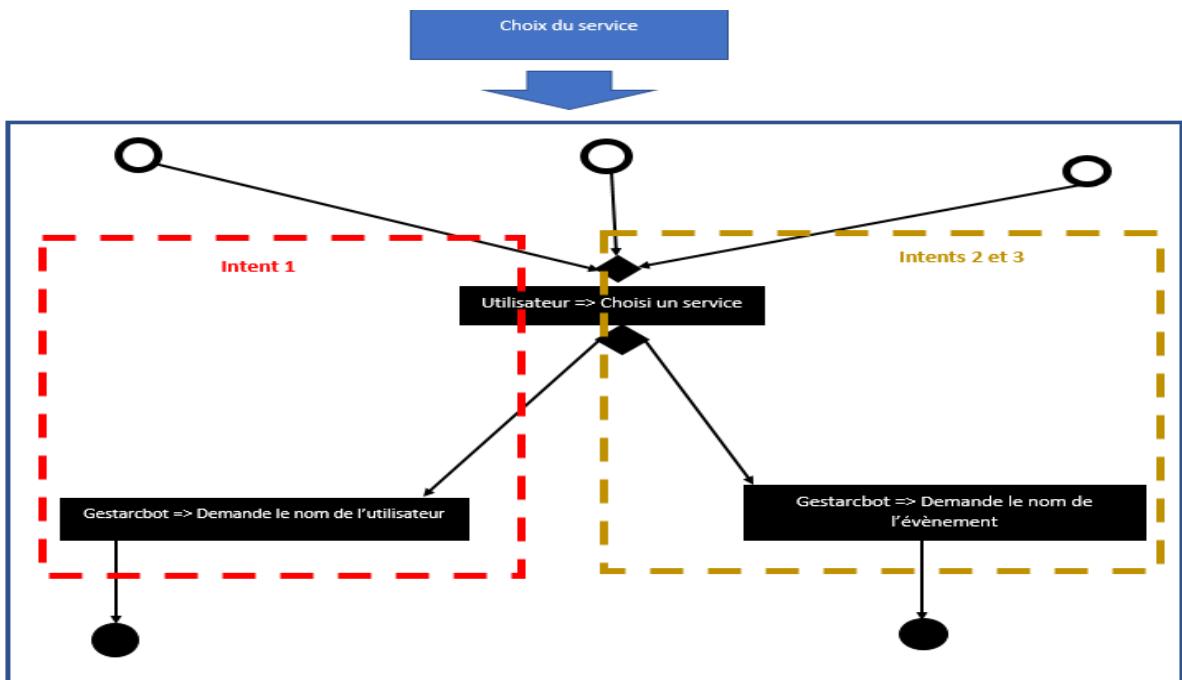


Figure 24 : Sous-processus de la tâche : Choix du service

### Choix de l'évènement

Dans le cas où l'utilisateur ait choisi soit le service « Acquérir une information sur un évènement » ou encore « Réservation de ticket pour un évènement », il serait alors question de savoir quel est l'évènement qui intéresse l'interlocuteur du chatbot. Comme dans les niveaux précédents, j'ai également fait un sous-processus pour cette étape.

Le schéma 25 ci-contre permet de visualiser le sous-processus ; tous les évènements qui se passent à travers ce niveau de la conversation. Ici, lorsque l'utilisateur donne le nom de l'évènement qui l'intéresse, si ce nom correspond à un évènement connu par le chabot, celui-ci demandera alors à l'utilisateur de quelle information a-t-il besoin. Dans le cas contraire, il donnera à l'utilisateur une liste des évènements organisés par la Gest'arc et lui demandera d'en choisir un.

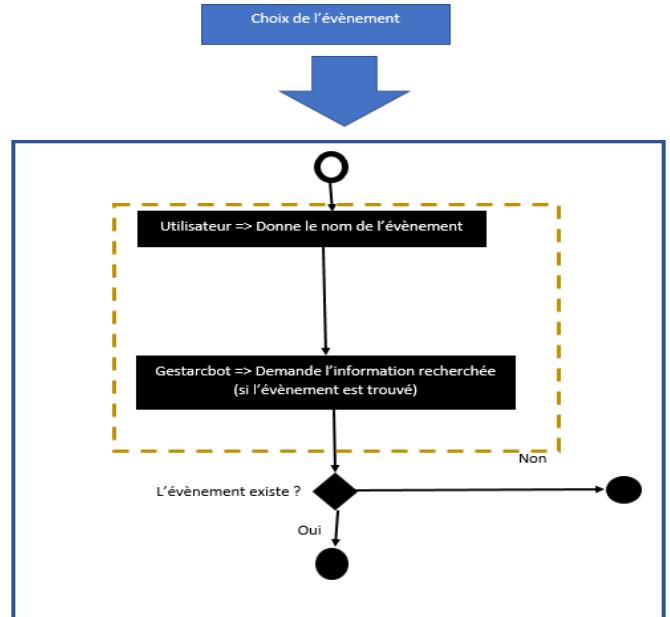


Figure 25 : Sous-processus de la tâche : Choix de l'évènement

### Aider l'utilisateur à trouver l'évènement

Maintenant il faut savoir que, lorsque le chatbot demande à l'utilisateur de lui donner le nom de l'évènement qui l'intéresse, que ce soit pour acquérir une information ou encore pour réserver un ticket, 2 possibilités sont prévues. En effet, l'utilisateur pourrait ne pas connaître le nom de l'évènement ou encore pourrait se tromper. La tâche « Aider l'utilisateur à trouver l'évènement » est alors prévu pour aider l'utilisateur à trouver l'évènement qui l'intéresse. Comme nous pouvons le voir sur le sous-processus ci-contre, si l'utilisateur ne connaît pas ou se trompe sur le nom de l'évènement qui l'intéresse, pour l'aider le chatbot va alors lui proposer une liste d'évènements pour l'aider à choisir.

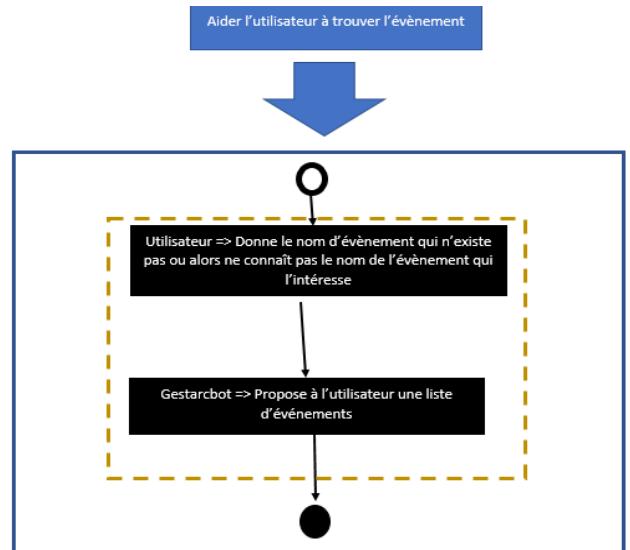


Figure 26 : Sous-processus de la tâche : Aider l'utilisateur à trouver l'évènement

### Information souhaitée

Une fois que l'utilisateur ait donné un nom d'évènement qui existe, Gestarcbot s'attend alors à ce que l'utilisateur lui communique de quelle information dont il a besoin.

La figure 27 ci-contre permet de voir comment l'étape que je pense être l'une des plus importantes du processus fonctionne. Dans un premier temps, l'utilisateur dit au chatbot de quelle information dont il a besoin, entre les horaires, les photos, les prix, le lieu, la date et le thème de l'évènement. Suite à cette requête, le chatbot retourne alors l'information souhaitée par l'utilisateur.

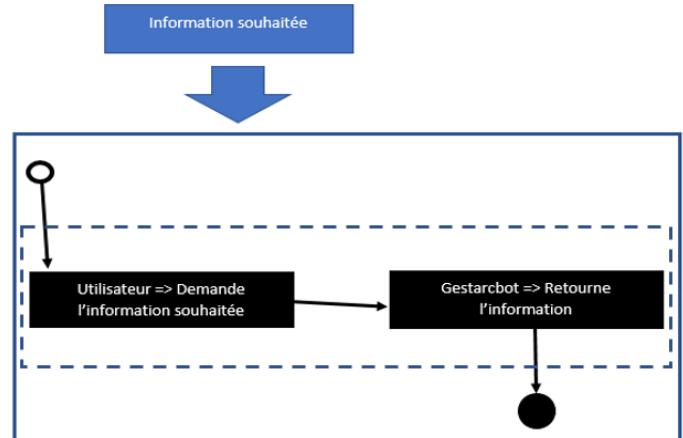


Figure 27 : Sous-processus de la tâche : Information souhaitée

## Identification de l'utilisateur

L'étape identification de l'utilisateur est l'étape du processus, où Gestarcbot cherche à identifier l'utilisateur, que ce soit pour lui réserver des tickets pour un évènement ou encore enregistrer sa demande pour devenir membre de la Gest'arc. Cette étape permet à Gestarcbot de connaître l'adresse email de son interlocuteur avant de pouvoir réaliser l'action qui lui ait demandé. Grâce au sous-processus ci-contre, nous pouvons voir qu'une fois que le chatbot sait l'intention de l'utilisateur, qui est soit de réserver un ticket pour un évènement ou de faire une demande d'inscription comme membre de la Gest'arc, il va chercher à identifier son interlocuteur. Pour le faire, le chatbot demande à l'utilisateur de lui dire son nom dans un premier temps. Avec cette information, Gestarcbot va alors chercher dans

sa base de données si ce nom existe. Suite à ça, plusieurs situations sont alors prévues : soit le nom existe déjà, le nom n'existe pas ou encore le nom existe mais plusieurs fois, dans le cas où plusieurs personnes auraient le même nom. Dans le premier cas de figure, si Gestarcbot retrouve dans sa base de données une personne ayant le nom donné par l'utilisateur, il va alors récupérer l'adresse email de l'utilisateur pour ainsi demander une confirmation. Dans le cas où le nom n'existe pas, ou alors si plusieurs personnes ont ce même nom, le chatbot demandera alors le prénom de l'utilisateur pour ainsi lui permettre soit d'enregistrer une nouvelle personne dans sa base de données ou alors élargir son filtre de recherche.

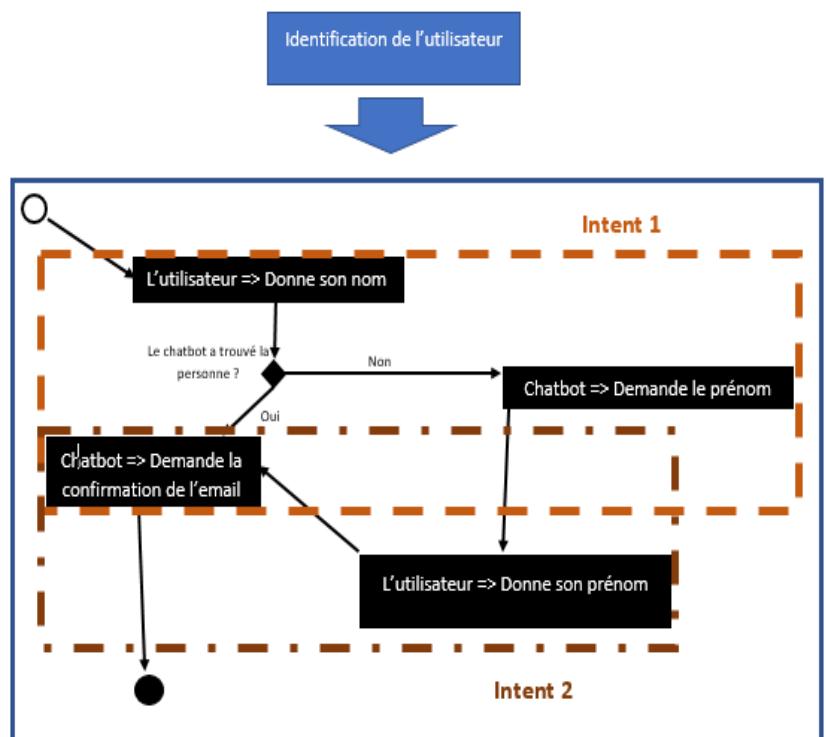


Figure 28 : Sous-processus de la tâche : Identification de l'utilisateur

## Réservation des tickets.

C'est également l'une des parties les plus importantes du processus, car c'est ici que le chatbot va enregistrer la réservation de ticket pour un évènement. Comme on peut le voir sur la figure 29 ci-dessous, une fois que l'utilisateur ait confirmé son identité, Gestarcbot lui demande alors par la suite le nombre de ticket qu'il souhaite réserver. Une fois que le chatbot connaît le nombre de ticket à réserver, il va dans un premier temps contrôler si le nombre de tickets réservables restant de l'évènement est suffisant pour lui permettre de réserver des tickets pour l'utilisateur. Dans le cas contraire, il réserve des tickets, mais uniquement le nombre de ticket restant et dans le cas où il ne resterait plus du tout de tickets réservables, il pourrait proposer à l'utilisateur de se

rendre directement auprès du secrétariat de la HEG-arc ou de la Gest'arc pour pouvoir en acquérir directement auprès d'eux.

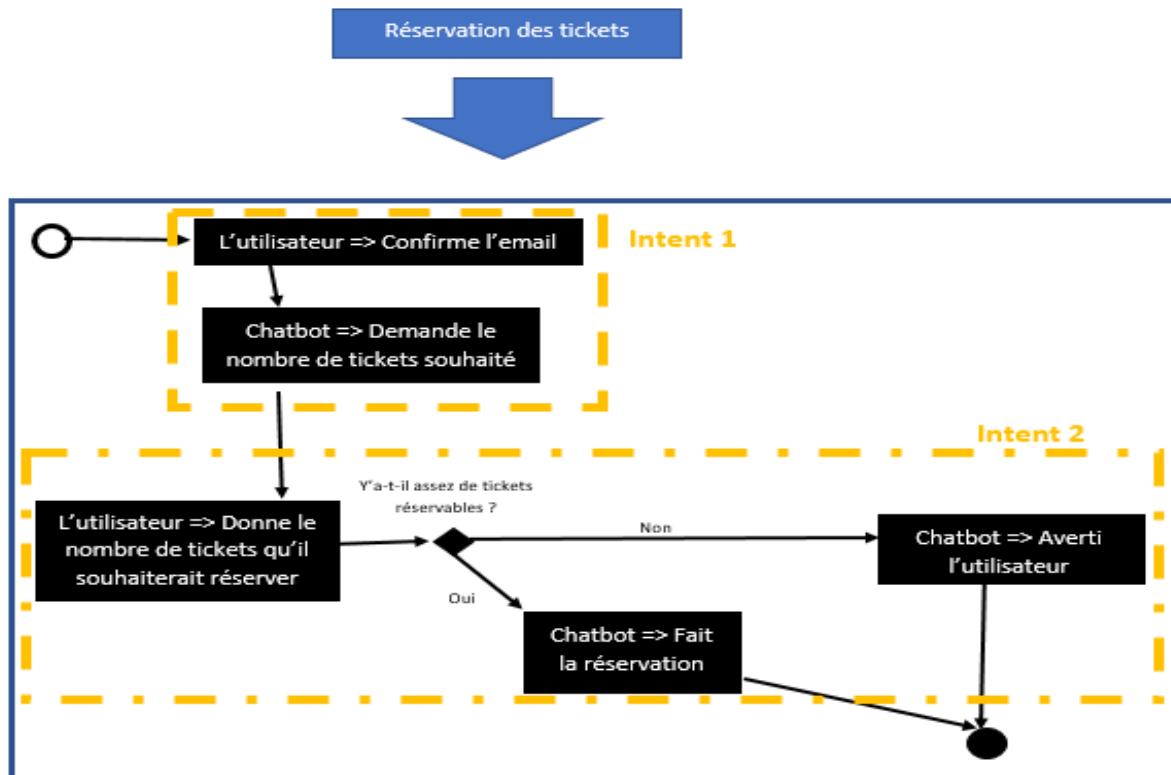


Figure 29 : Sous-processus de la tâche : Réservation des tickets

### Enregistrement de la demande d'inscription

Comme lors des autres services, cette tâche du processus fait donc partie des plus importantes de la logique de Gestarcbot, car c'est l'étape principale du processus du service « Enregistrement de la demande d'inscription » de Gestarcbot. Comme nous pouvons le voir dans l'image ci-dessous, une fois que l'utilisateur ait confirmé son adresse email donc son identité, Gestarcbot va d'abord contrôler si cet utilisateur avait déjà fait une demande d'inscription. Dans le cas où une demande avait déjà été enregistré, Gestarcbot averti l'utilisateur et lui demande de patienter encore un petit moment, jusqu'à ce qu'un membre de Gest'arc face le nécessaire. Dans le cas où Gestarcbot ne trouve aucune demande de l'utilisateur, le chatbot enregistre alors une demande d'inscription pour l'utilisateur et l'en informera par la suite.

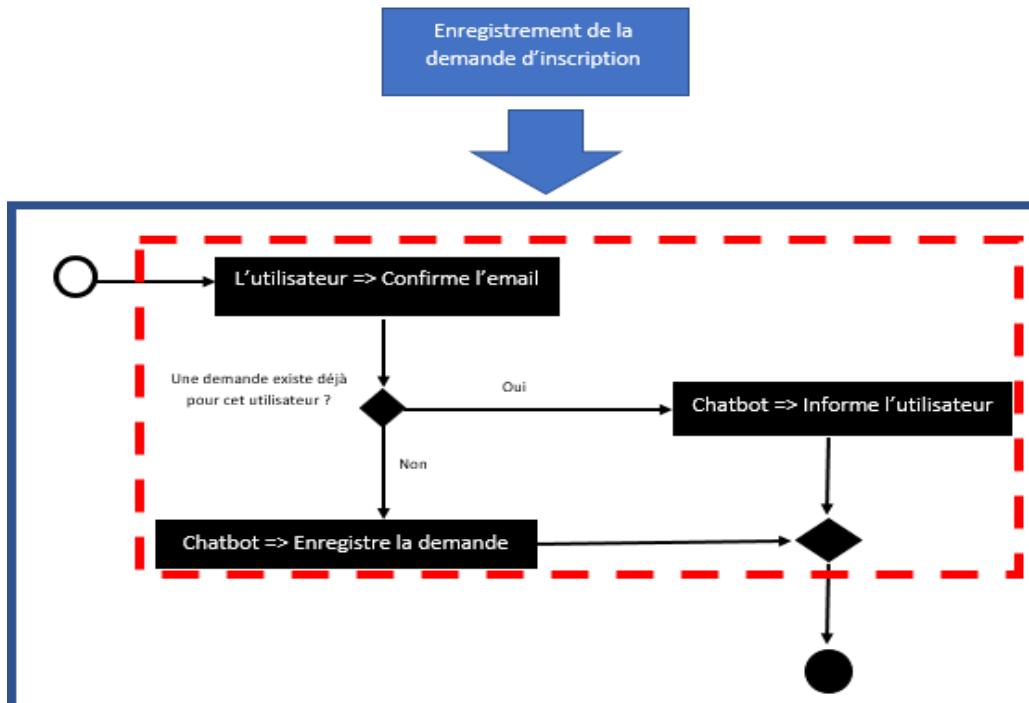


Figure 30 : Sous-processus de la tâche : Enregistrement de la demande d'inscription

### **Remerciements et salutations**

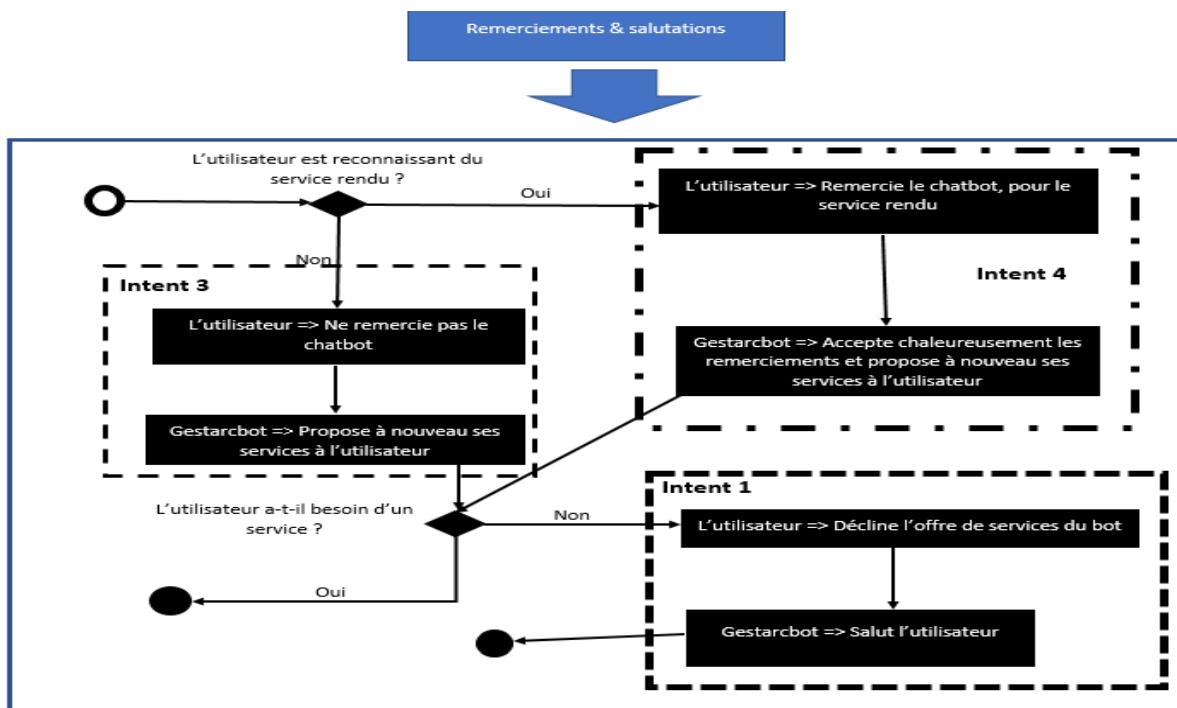


Figure 31 : Sous-processus de la tâche : Remerciements & salutations

Comme les deux premières, cette tâche du processus est une étape qui sera dans chacun des services de Gestarcbot.

Une fois que Gestarcbot ait donné l'information recherchée, réservé des tickets ou encore enregistré la demande pour devenir membre de la Gest'arc, l'utilisateur peut alors soit juste accepter l'information par un simple « Ok » par exemple, ou alors il pourrait gentiment remercier le chatbot pour le service rendu, démontrant ainsi une forme de reconnaissance. Dans la première situation, Gestarcbot propose simple à nouveau ses services à l'utilisateur. Dans la deuxième situation, le chatbot accepte les remerciements de l'utilisateur avant de lui proposer à nouveau ses autres services ce qui simulera alors une forme d'émotion de la part de Gestarcbot. La figure 31 ci-dessus, permet de visualiser tout le mécanisme de cette tâche.

Maintenant que nous avons vue l'une des premières étapes de la réalisation du chatbot sur API.AI, nous allons aborder les contexts.

### 8.1.2 Contexts

Toujours selon Google, les « contexts » représentent le contexte actuel d'une conversation avec l'utilisateur. Ils sont utiles pour différencier les expressions qui peuvent être vagues ou avoir des significations différentes selon les choix de l'utilisateur ou simplement selon le sujet de conversation. Les context sont de type session scoped, c'est-à-dire qu'ils ont une portée sur toute la session d'une conversation avec un utilisateur suivant et dépend également la durée de vie qu'on leur donne. Certaines informations peuvent donc y être stockées pour être réutilisées plus tard dans la conversation. Par exemple, si un utilisateur désire une information sur un évènement, le chatbot lui demandera le nom de l'évènement et une fois l'évènement trouvé, ce nom sera alors stocké dans le context pour ainsi pouvoir le réutiliser plus tard dans la conversation. Pour se déclencher, certains intents nécessite la combinaison entre ce que l'utilisateur dit « User says », et le context qu'on appelle ici, input context. Cette combinaison permet simplement au chatbot de pouvoir différencier la signification réelle de certaines expressions suivant le choix de l'utilisateur. Un intent peut aussi générer un output context, qui lui permettra de partager des valeurs entrées par l'utilisateur pendant la conversation ou alors de permettre au chatbot de diriger la conversation vers un intent précis ([API.AI, 2016](#)). Pour prendre connaissance de tous les contexts que j'ai mis en place dans le cadre de ce projet, plusieurs schémas sont disponibles à l'annexe 6.

### 8.1.3 Actions

Maintenant que nous avons vu les intents à réaliser sur API.AI ainsi que les liens qui sont fait entre eux, intéressons-nous maintenant aux actions. La documentation de API.AI, nous dit qu'une action correspond à l'étape où un intent interagit avec le webhook et l'action permet au webhook de savoir quel est le service (du côté webhook), le chatbot a besoin. Les actions peuvent avoir des paramètres pour pouvoir extraire des informations des requêtes de l'utilisateur et ainsi transmettre ces informations au webhook. La communication entre le chatbot et le webhook se fait dans le format JSON ([API.AI, 2016](#)).

On retrouve les actions des intents de Gestarcbot dans les tableaux de la création et paramétrage des intents de Gestarcbot à l'annexe 7.

### 8.1.4 Entities

Plus haut dans ce chapitre, nous avons vu que les intents de Gestarcbot doivent transmettre des informations au webhook grâce aux « Actions ». Il faut donc savoir que ces « Actions » extraient les données dans ce que l'utilisateur dit grâce à des paramètres appelés « Entities ». Google nous dit donc que, Les « Entities » sont des outils utilisés pour extraire des valeurs grâce au « Natural language » ([API.AI, 2016](#)). Dans le prochain point, nous allons voir quel sont les entities que j'ai mis en place pour permettre au chatbot de pouvoir récupérer les informations dont il aura besoin pour mener à bien sa mission.

### 8.1.5 Paramétrage de Gestarcbot

Finalement, la création d'un chatbot se résume dans un premier temps à la création d'un agent sur API.AI, ensuite à la création et au paramétrage de plusieurs intents. Les tableaux que j'ai réalisés en annexe, démontrent comment j'ai paramétré les différents intents de mon agent intelligent. Je tiens à préciser que, je me suis inspiré de mes Scénarios (cf, chapitre 3.1.3), pour pouvoir mettre des valeurs dans les « User says » et « Réponses du chatbot » de mes intents. Donc, une fois mes tests effectués j'étendrai ces valeurs grâce aux interactions entre Gestarcbot et les testeurs. À l'annexe 7, j'ai réalisé des tableaux qui permettent de savoir comment j'ai paramétré les intents de mon agent sur API. Il faut également savoir que, lorsqu'un intent contient plus d'un output context c'est pour la bonne et simple raison que cet intent peut donner suite à différents autres intent suivant les circonstances. Dans ce cas précis, la conversation peut alors prendre différentes tournures. Par exemple, si nous prenons l'intent « Others-services-user-firstname », une fois que le chatbot ait identifier l'utilisateur, il pourra alors diriger la conversation soit vers une réservation de billet pour un évènement ou alors vers une inscription pour être membre de la Gest'arc, selon le choix initial de l'utilisateur. Gestarcbot pourra alors poursuivre la conversation et échanger des informations avec le webhook sans se trompé grâce à un output context « context-ticket-reservation » qui sera généré dans la première situation ou un output context « context-member-registration » sera généré dans la cas de l'inscription. Nous avons vu dans ce chapitre que la plateforme API.AI permet de récupérer des informations des requêtes des utilisateurs grâce à des entities dans le cas de Gestarcbot, j'ai créé 4 entities qui permettent à notre chatbot de récupérer les informations qui lui sont utiles. Il s'agit de :

- EventName cet entity est utile pour récupérer le nom de l'évènement qui intéresse l'utilisateur.
- Needed\_information cet entity permet à Gestarcbot de récupérer le nom de l'information que l'utilisateur souhaiterait avoir.
- UserFirstname cet entity sert au chatbot de récupérer le prénom de l'utilisateur
- UserLastname cet entity est utile au chabot pour récupérer le nom de famille de l'utilisateur.

Le deuxième élément de notre chatbot qui a particulièrement demandé de l'attention dans le cadre de ce travail pratique était le développement du Webhook.

## 8.2 Développement du webhook

Dans le chapitre précédent, j'ai présenté et expliqué comment j'ai développé mon agent intelligent sur API.AI, je vais maintenant parler du développement du webhook dans cette partie. Le webhook permet à Gestarcbot de récupérer, comparer et stocker des informations dans la base de données, ce qui lui permet de réaliser des actions pertinentes et donne au chatbot tout son intérêt non seulement pour les utilisateurs, mais aussi pour la Gest'arc. Selon la documentation de Google, le webhook dont API.AI a besoin est un service web qui doit respecter plusieurs restrictions. Dans un premier temps, ce service doit être déployé et accessible sur internet par le protocole « HTTPS ». API.AI doit pouvoir faire appel au webhook grâce à une requête POST. Le service web doit répondre aux requêtes de API.AI en format JSON dont le contenu est le suivant :

Name	Type	Description
speech	String	Response to the request.
displayText	String	Text displayed on the user device screen.
data	Object	Additional data required for performing the action on the client side. The data is sent to the client in the original form and is not processed by API.AI.
contextOut	Array of context objects	Array of context objects set after intent completion. Example: <code>"contextOut": [{"name": "weather", "lifespan": 2, "parameters": {"city": "Rome"}}]</code>
source	String	Data source.
followupEvent	Object	Event name and optional parameters sent from the web service to API.AI. Example: <code>{"followupEvent": {"name": &lt;event_name&gt;, "data": {"&lt;parameter_name&gt;": &lt;parameter_value&gt;}}} <a href="#">Read more</a>.</code>

Figure 32 : Tableau du contenu de la réponse du webhook, selon les contraintes de API.AI ([API.AI, 2016](#))

Exemple de réponse du webhook, suite à une requête de API.AI :

```
{  
  "speech": "Barack Hussein Obama II was the 44th and current President of the United States.",  
  "displayText": "Barack Hussein Obama II was the 44th and current President of the United States, and the first African American to hold the office. Born in Honolulu,"
```

```
Hawaii, Obama is a graduate of Columbia University and Harvard Law School, where  
",  
"data": {...},  
"contextOut": [...],  
"source": "DuckDuckGo"  
}
```

Pour développer ce services web, il existe plusieurs technologies. En effet, API.AI n'impose pas de technologie spécifique pour le faire. Pour ce travail, j'ai décidé de développer un servlet java qui sera déployé sur un serveur d'application Glassfish accessible sur internet par le protocole « HTTPS ». Un servlet java est pour moi un excellent candidat pour remplir les fonctions d'un webhook pour API.AI. En effet, il a la capacité de recevoir des requêtes POST ou GET depuis un client distant. Combiné aux APIs comme « Jackson » et « Java-json », qui sont capables de parser des objets java en objets JSON et vice versa, un servlet peut donc recevoir des requêtes, les parser et envoyer des réponses au format JSON, sans aucun problème. De plus, ce sont des technologies qui me sont les plus familières.

Avant de créer mon projet Java, il m'est fallu dans un premier temps établir des règles métiers qui m'ont de gérer des cas limites liés au business. Dans le prochain point, je vais présenter les différentes règles métiers que j'ai pu définir avec l'aide de la Gest'arc.

### 8.2.1 Règles métiers

Avant de développer le servlet java pour faire le lien entre API.AI et ma base de données, j'ai établi une liste de règles métiers. Ces règles métiers ont eu pour but de m'aider à définir et à gérer certains cas limites. Ces règles métiers sont :

- Une demande d'inscription pour devenir membre de la Gest'arc n'est possible que lorsque le demandeur est enregistré comme étudiant de la HEG-arc.
- Il n'est pas possible de faire plusieurs demandes d'inscription pour un étudiant.
- Une fois qu'un service est rendu, le chatbot doit proposer uniquement ses autres services à l'utilisateur.
- La réservation de ticket pour un évènement n'est possible que pour les évènements qui n'ont pas encore eu lieu et dont les réservations de ticket par Facebook sont autorisées.
- L'utilisateur qui veut faire une réservation doit être un étudiant et enregistrée dans la base de données de la Gest'arc.
- Le nombre des places disponibles d'un évènement diminue comme celui des places réservables après chaque réservation.

Ces quelques règles métiers m'ont permises de développer un servlet qui fait exactement tout ce dont le chatbot Gestarcbot a besoin pour réaliser ses services.

Dans le point suivant, je vais présenter le projet java en lui-même.

## 8.2.2 Projet

Dans cette partie, je présenterai le servlet ainsi que le projet que j'ai réalisé pour le développer.

Etant donné que j'ai choisi de développer un servlet java pour faire la liaison entre API.AI et ma base de données Mysql, j'ai dû réaliser un projet de développement java. L'objectif était de développer un service web offrant plusieurs méthodes à API.AI pour lui permettre d'obtenir les données de la base de données dont il a besoin. En effet, mon servlet offre des méthodes qui permettent à Gestarchbot de réaliser ses services. L'image ci-dessous illustre schématiquement, les différentes méthodes que j'ai développées ainsi que les liaisons qui existent entre elles et la plateforme API.AI.



Figure 33 : Liens entre le webhook et la plateforme API.AI

Comme nous pouvons voir sur l'image 33 ci-dessus, API.AI interagit avec le webhook grâce à ses actions comme : get-events-to-help, check-event-name etc.... Ces actions permettent à API.AI d'invoquer chacune des méthodes offertes par mon servlet java à un instant T dans la conversation. Etant donné que les noms des méthodes sont long et ne sont pas très parlant, j'ai décidé de les nommer de 1 à 11, dans la figure 33.

Il faut savoir que mon projet est un projet de type Java EE, dans lequel j'ai développé une classe de type servlet. Pour pouvoir faire la connexion avec la base de données, j'ai utilisé l'API JPA, qui m'a permis de faire un matching entre mes tables en base de données et mes objets java dans mon projet. Etant donné que mon servlet doit recevoir des requêtes et renvoyer des réponses au format JSON, j'ai utilisé deux autres APIs, Jackson et Java-json qui m'ont permises de transformer les différents objets Json en objets java, ce qui m'a alors permis de faire les parsings « Java to json » et « Json to Java », sans problèmes. Les micros services correspondent à plusieurs méthodes que j'ai développées. Ces méthodes sont invoquées par ma classe servlet, suivant l'action qui est envoyée dans la requête POST de API.AI. Comme on peut le voir ci-dessous, mon servlet est en fait un enchaînement de conditions « if », qui se déclenche et réalise une action spécifique à chaque fois que l'action envoyée par API.AI correspond à une des conditions.

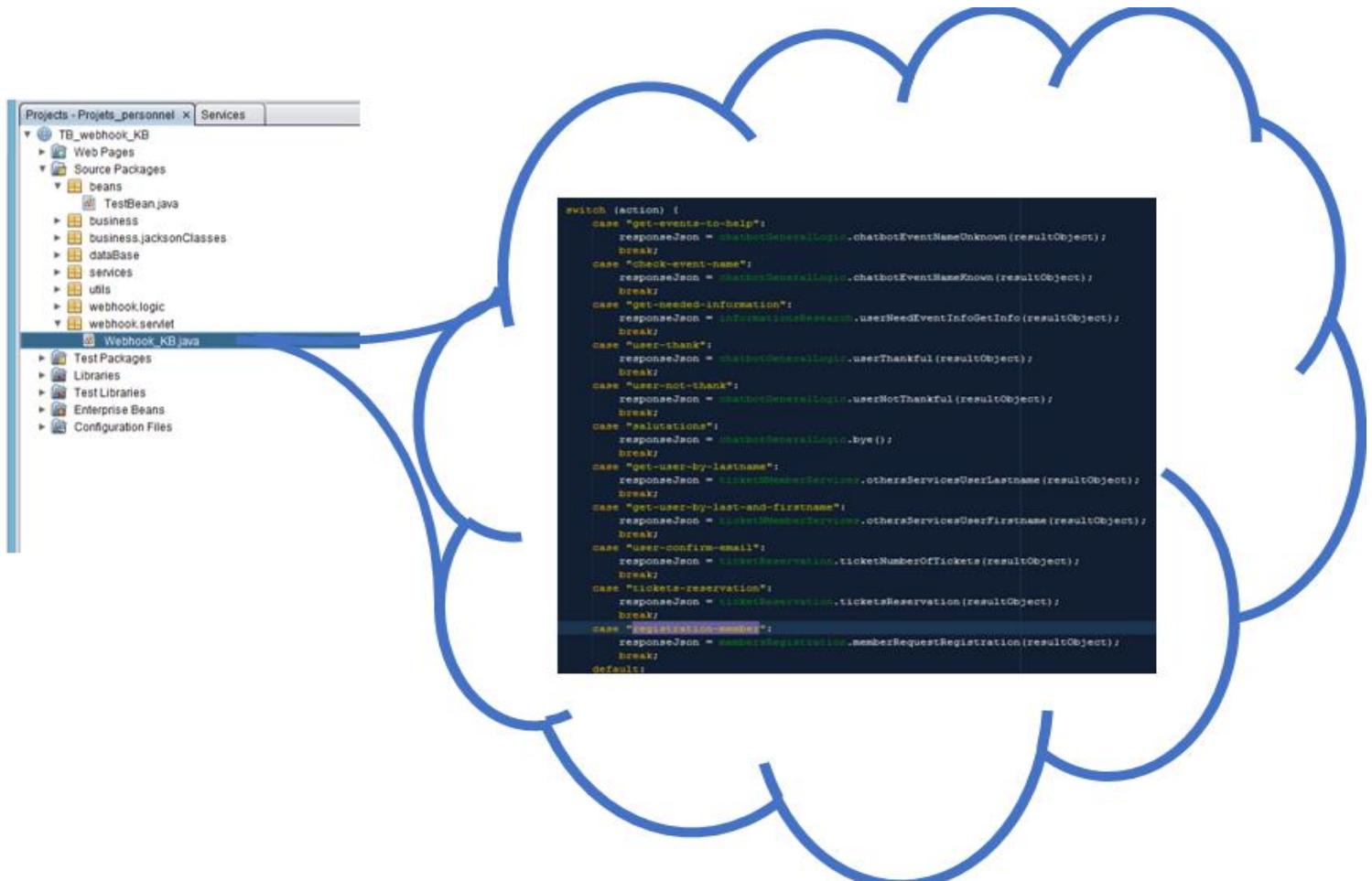


Figure 34 : Représentation du projet java

Le tout n'est pas de mettre en place un chatbot, mais le plus important est que celui-ci respecte les besoins de chacune des parties prenantes, que ce soit des utilisateurs du chatbot ou encore de la Gest'arc. Dans le prochain chapitre, je vais présenter les

tests qu'ont réalisé plusieurs étudiants et membre de la Gest'arc, sur Gestarcbot. Je parlerai également de leurs feedbacks et des améliorations que j'ai dû apportées à apporter sur cette première version de Gestarcbot.

## 9 Tests

---

Dans les chapitres précédents, j'ai présenté toutes les étapes du développement de Gestarcbot, de la partie développement de l'agent sur la plateforme API.AI, au développement du servlet java qui sert de webhook.

Maintenant que le chatbot est développé, nous allons aborder dans ce chapitre les différents tests réalisés sur Gestarcbot par un membre de la Gest'arc et quelques étudiants. Ces tests ont permis d'évaluer les performances et surtout déterminer si notre solution parvient à couvrir les besoins que nous avons vu dans le chapitre Analyse des besoins. Je parlerai également des résultats des test ainsi que des améliorations réalisées.

### 9.1 Jeux de tests et résultats

#### 9.1.1 Jeux de tests

Comme nous le savons, les performances d'un chatbot contrairement à une application de gestion classique, ne se mesure pas à la qualité de son interface, mais à sa capacité de tenir une conversation dans un premier temps. Ensuite, il faut tester sa capacité à réaliser des actions et à fournir des réponses pertinentes pour l'utilisateur.

Contrairement à une application classique où il est question de fournir des données de tests, dans notre situation nous allons tout simplement demander au testeur de se mettre dans 3 jeux de situations. Ces situations sont les suivantes :

- Demande de renseignements : Vous êtes une personne qui souhaite participer au prochain évènement organisé par la Gest'arc. Malheureusement, vous ne savez pas si l'évènement est gratuit et surtout s'il en reste encore des places. N'ayant pas trouvé ces informations par vous-même, vous décidez de vous renseigner auprès de l'association des étudiants depuis leur page Facebook.
- Réservation de ticket : Vous êtes un nouvel étudiant et êtes intéressé par l'association des étudiants « La Gest'arc ». Vous avez entendu parler de leur présentation dont le titre est « La Gest'arc », qui aura bientôt lieu et souhaitez y participer. N'ayant pas le temps de passer au secrétariat pour réserver votre place, vous décidez alors de le faire directement auprès de l'association grâce à Facebook Messenger.
- Devenir membre de la Gest'arc : Après avoir participé à la présentation « La Gest'arc », vous avez eu besoin de réfléchir pendant quelques jours avant de prendre une décision. Votre décision enfin prise, vous décidez de faire une

demande pour vous inscrire comme nouveau membre de l'association. Dû à votre charge de travail, vous n'avez pas le temps de vous rendre dans le bureau de la Gest'arc pour faire votre demande. Par conséquent, vous décidez alors de la faire directement par Facebook Messenger.

Grâce à ces différentes situations impliquant l'utilisation de Gestarcbot, chacun des testeurs a donc pu se mettre en condition pour pouvoir tester notre chatbot. Grâce aux résultats de ces tests, nous avons pu voir de quelle manière réagissait le chatbot face à un utilisateur lambda.

### 9.1.2 Résultats des tests

Pour mieux présenter les résultats des différents tests, je vais procéder au cas par cas. Nous allons d'abord voir les résultats des tests de la situation demande de renseignement ensuite pour la réservation de tickets et enfin pour la situation pour devenir membre de la Gest'arc. Pour éviter de trop charger le document, j'ai uniquement présenté quelques résultats des différents tests réalisés.

#### Demande de renseignements

##### Essai 1



Figure 35 : Tests pour la demande de renseignements, essai 1

## Essai 2



Figure 36 : Tests pour la demande de renseignements, essai 2

Grâce à ces deux essais de test du service donner une information sur un évènement, on peut remarquer que le chatbot arrive parfaitement à fournir l'information dont le testeur a besoin, comme on peut le voir dans le deuxième essai. Dans le premier essai, étant donné que le testeur a demandé une information dont le chatbot n'est pas capable de fournir, celui-ci a alors informé l'utilisateur qu'il s'adresse à un robot et de ce dont il est capable de fournir. C'est une manière de faire qui permet à l'utilisateur de savoir qu'il est en face d'un chatbot et qu'il n'est sûrement pas capable de tout faire ou de tout connaître, en tout cas pas pour l'instant.

## Réservation de ticket

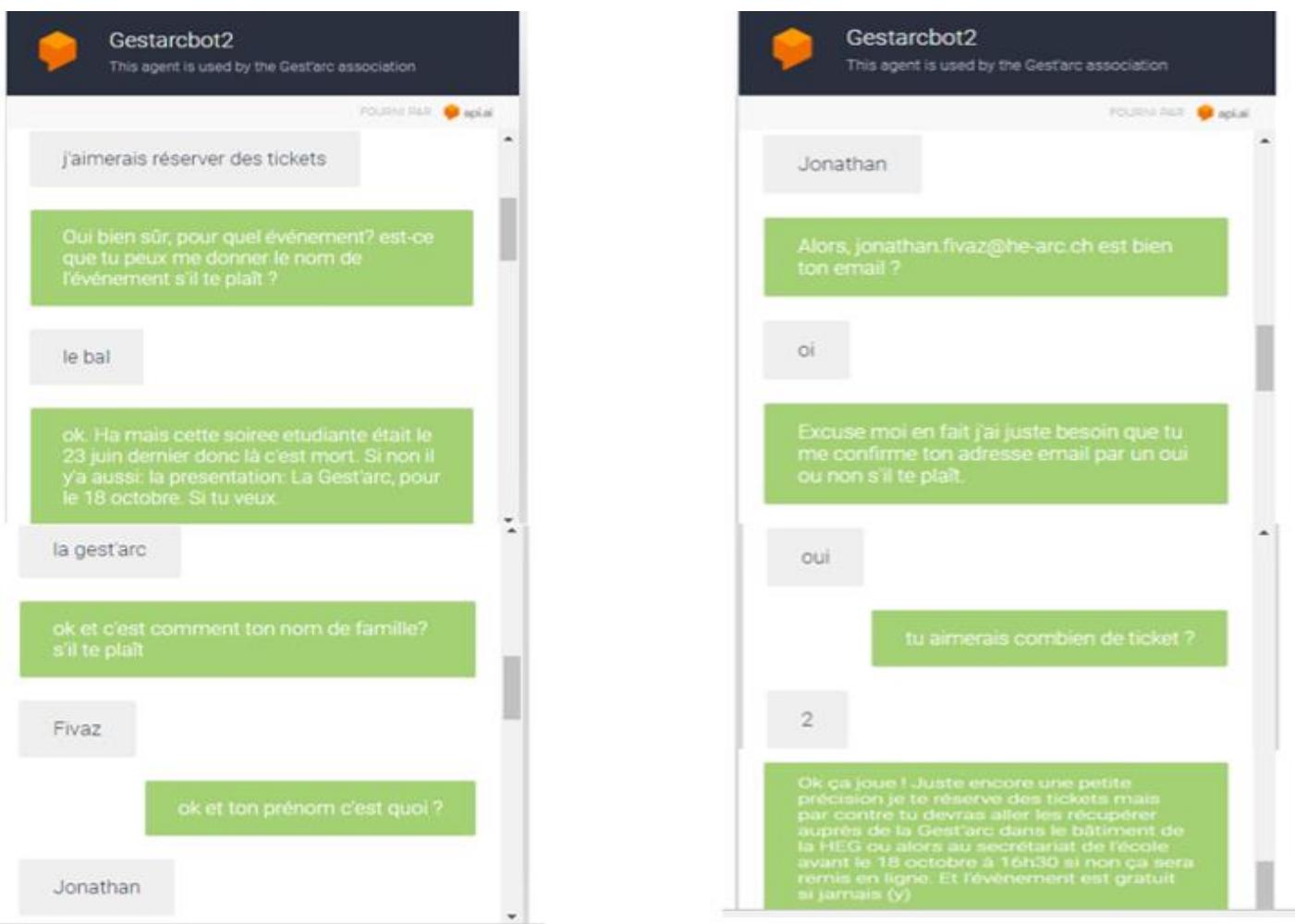


Figure 37 : Tests pour la réservation de ticket

Lors des tests du service réservation de ticket pour un évènement, pu constater que le chatbot réagissait de manière adéquate, même lorsque n'avait pas compris la requête de l'utilisateur.

## Devenir membre de la Gest'arc

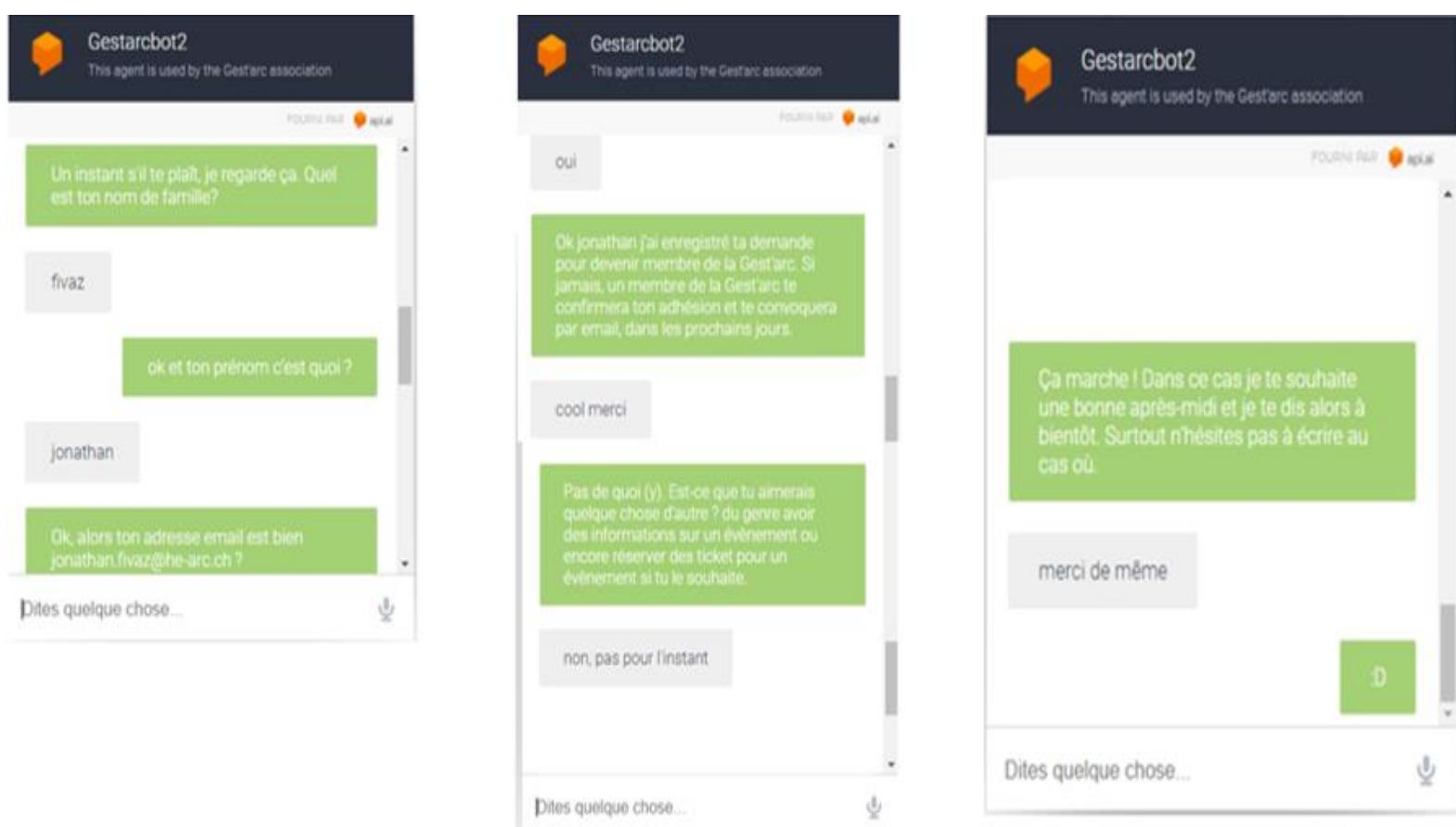


Figure 38 : Tests pour devenir membre de la Gest'arc, essai 3

Enfin, grâce aux tests réalisé pour devenir un nouveau membre de l'association, on peut se rendre compte que le chatbot réalise sa tâche sans problème.

Finalement, je peux donc conclure que ces tests démontrent que cette version de Gestarcbot couvre en tout cas, sans problème les besoins que nous avons identifiés dans le chapitre Analyse des besoins.

Cependant avant ces tests réalisés avec succès, j'avais réalisé des tests sur une première version du bot. ces tests avaient alors beaucoup de dysfonctionnement de Gestarcbot, qu'il ait fallu que je corrige pour améliorer le bot et le rendre fonctionnel. Dans le point suivant, je vais faire part des remarques des testeurs ainsi que des améliorations apportées à Gestarcbot suite aux tests de la première version.

## 9.2 Améliorations

Après avoir tester la première version de Gestarcbot, les testeurs avaient fait part de leurs impressions. Ces impressions m'avaient permis, en plus des résultats des tests réalisés d'améliorer cette première version.

Voici les remarques des testeurs de la première version de notre chatbot :

D'après Jonathan Fivaz, membre de la Gest'arc, Gestarcbot, semble fonctionnelle mais un peu rigide, dans le sens où il ne permet pas les requêtes « groupées » (le quoi, le quand et le où), dans un seul message par exemple. Trop de ping-pong (utilisateur : bonjour ; chatbot : bonjour utilisateur : blablabla...). Il faudrait aussi relancer l'utilisateur de manière cohérente. En effet, lorsque l'utilisateur formule sa demande d'information de façon à ce que le chatbot ne comprenne pas par exemple, Gestarcbot devrait naturellement guider son interlocuteur pour ainsi l'aider à mieux formuler sa demande continue Jonathan. L'application pourrait par exemple reformuler/ proposer une requête ou alors proposer des solutions en fonction du contexte, en cas de problème. Par exemple : « désolé j'ai bien compris ce que tu aimerais mais pour l'instant, je ne peux te donner des renseignements que sur, le prix, la date, le lieu, le thème ou encore les photos si tu veux. ».

De plus, L'application a besoin de terme précis pour répondre aux questions de l'utilisateur. Donc si le mobinaute ne parle pas comme Gestarcbot s'attend, le chatbot ne comprend pas la plupart du temps, la requête. Les termes utilisés pour les recherches devraient être plus ouverts et moins précis car chacun parle différemment. Poursuit Klett Elga, une étudiante de la HEG.

Grâce aux différents feedbacks des testeurs et aux résultats des tests, je m'étais alors rendu compte que la première version de Gestarcbot comportait beaucoup d'imperfections qu'il fallait absolument corriger. Pour commencer, il avait donc fallu dans un premier temps revoir la logique que j'avais au préalable défini pour pouvoir permettre au chatbot de gérer d'autres schémas pour pouvoir réaliser chacun de ses services. Ensuite, il fallait revoir plusieurs intents de l'agent développer sur API.AI. En effet il fallait mettre en place une alternative pour chacun des intents lorsque le chatbot ne comprend pas la requête de l'utilisateur. Ces alternatives ont donc permis au chatbot de pouvoir relancer de manière plus adéquate et moins générique comme c'était le cas. En plus des alternatives il fallait également retravailler les « user says » des intents existants, ce qui a permis à Gestarcbot de pouvoir les matcher avec un maximum de requêtes possibles.

Dans le prochain point, je vais présenter les ajustements que j'ai dû apportés à la logique de Gestarcbot afin de lui permettre d'être plus efficace pendant ses interactions avec l'utilisateur.

### 9.2.1 Optimisation de la logique de Gestarcbot.

Tout d'abord, il ait fallu que j'ajuste la logique du chatbot par rapport à la logique que j'ai présenté et développé dans les chapitres précédents. Pour améliorer la logique de notre chatbot, j'ai dû ajouter un schéma à la logique de notre chatbot, le schéma que nous pouvons voir dans la figure 39 ci-dessous.

### **Légendes**

- Pour le processus du service : Acquérir une information sur un évènement
- Pour le processus du service : Réserver un ticket pour un évènement
- Pour le processus du service : Demande d'inscription comme membre de la Gest'arc

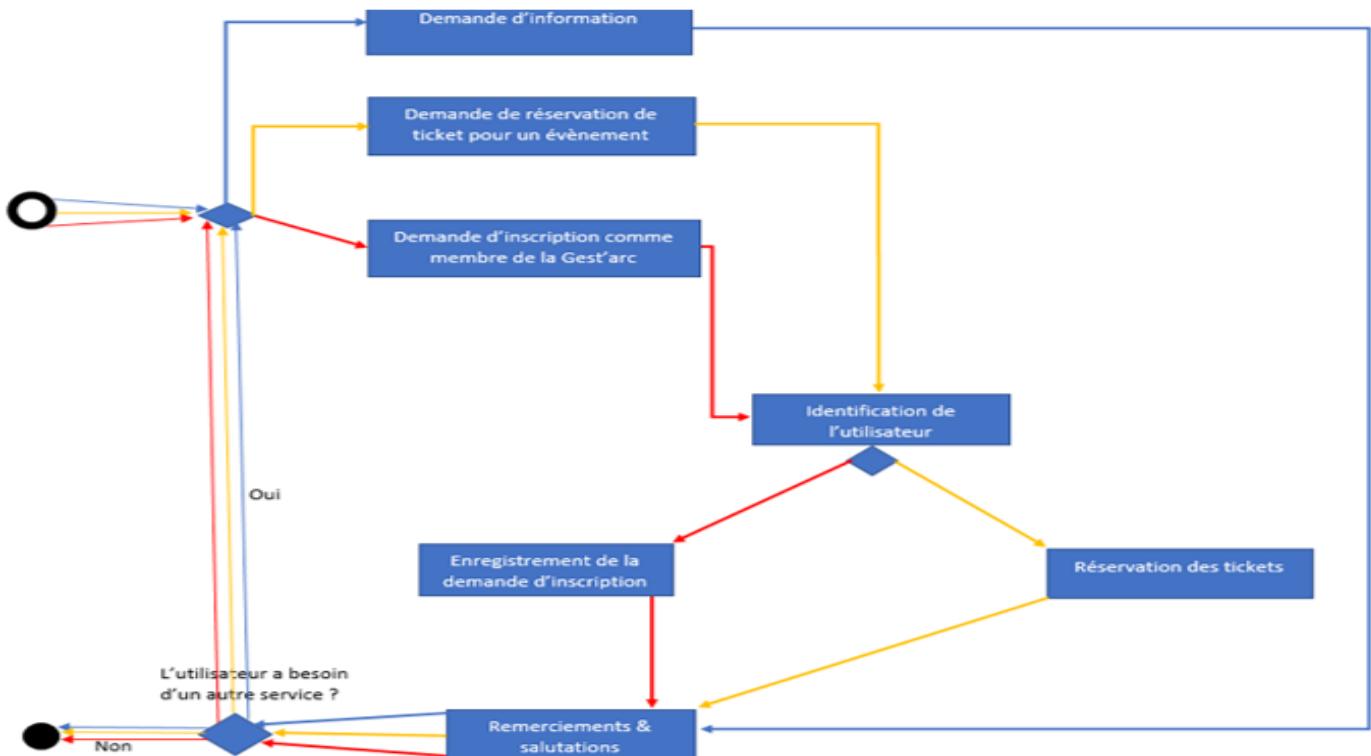


Figure 39 : Schéma supplémentaire pour la logique de Gestarcbot

La figure ci-dessus représente donc le schéma que j'ai ajouté à la logique de Gestarcbot. Ce nouveau schéma permet à notre chatbot d'exécuter ses services plus rapidement et plus efficacement. En effet, ce nouveau schéma présente moins d'étapes entre le début et la fin du processus ce qui diminue le risque des erreurs. Nous pouvons voir que par rapport au schéma présenté dans le chapitre de la Logique de Gestarcbot, j'ai supprimé certaines étapes et ajouté les étapes, « Demande d'information », « Demande de réservation de ticket pour un évènement » et « Demande d'inscription comme membre de la Gest'arc » par rapport à l'ancien schéma. Pour mieux comprendre chacune des étapes que j'ai ajoutées, voir l'annexe 8. Il faut savoir que les deux schémas font actuellement partie de la logique de Gestarcbot augmentant ainsi son efficacité.

Une fois que j'ai optimisé la logique de Gestarcbot, il fallait que j'optimise les intents.

### **9.2.2 Optimisation et création de nouveaux intents**

Dans un premier temps, j'ai d'abord ajouté les intents permettant de mettre en place le nouveau schéma de la logique du chatbot. Les figures des nouvelles étapes de la logique de Gestarcbot à l'annexe 8 permettent d'illustrer les intents que j'ai ajoutés. Il s'agit donc d'implémenter les intents « User\_need\_event\_info\_get\_info\_2 » et « Ticket-reservation\_2 » sur la plateforme API.AI.

Par la suite j'ai mis en place un default fallback intent pour chacun des intents. Ces default fallback intents se déclenchent lorsque le chatbot ne comprend pas ce que l'utilisateur lui dit et permettent donc à Gestarcbot de répondre et guider l'utilisateur. Selon la documentation de Google API.AI, un default fallback intent est un intent qui est déclenché lorsque la requête de l'utilisateur ne correspond à aucun des intents de l'agent. Un default fallback intent précis peut être déclenché uniquement grâce à un context (API.AI, 2016).

Dans le prochain chapitre, je vais présenter le produit final dans son ensemble.

---

## **10 Présentation du produit**

Dans ce chapitre, je vais parler des fonctionnalités du produit final dans un premier temps avant de présenter le livrable.

### **10.1 Fonctionnalités**

Comme nous avons pu le voir tout au long de ce projet, notre produit fini propose 3 fonctionnalités principales, qui sont :

- La demande d'information sur un évènement
- La réservation de ticket pour un évènement
- La demande d'inscription comme nouveau membre de la Gest'arc.

Gestarcbot offre la possibilité à son interlocuteur de bénéficier de l'un de ses services au moyen d'une conversation naturelle, comme nous avons pu le voir dans les chapitres précédents, notamment dans les tests.

Dans les prochaines versions du chatbot, d'autres fonctionnalités pourraient être implémentées et permettre à Gestarcbot d'offrir davantage de services aux utilisateurs. Dans le chapitre Versioning, je ferai part des propositions de fonctionnalités à implémenter dans le futur.

Le livrable de ce projet est portable, c'est-à-dire que c'est un ensemble de fichiers qui peuvent facilement être déplacés. Dans le prochain point, je présenterai les différents dossiers et fichiers du livrable.

## 10.2 Fichiers

Dans cette partie du chapitre, je vais présenter les dossiers du livrable. Commençons par le webhook.

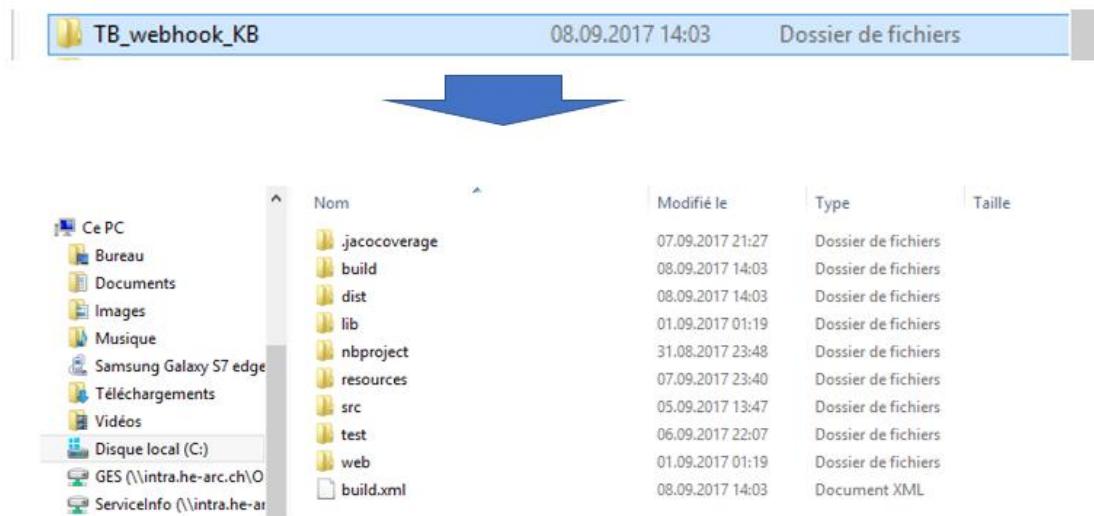


Figure 40 : Représentation du dossier du projet du servlet java pour le webhook

Comme nous pouvons le voir dans la figure ci-dessus, le webhook est un simple dossier contenant tous les fichiers du projet java.

API.AI offre la possibilité d'exporter ou d'importer un projet sur la plateforme. Comme la figure ci-dessous permet de l'illustrer, l'agent Gestarcbot est un dossier contenant des fichiers qui contiennent des entities et intents y compris un fichier qui contient tout le paramétrage de l'agent, qui correspond ici au fichier « agent.json ».

Nom	Date	Mots clés	Taille	Notation	Chemin du dossier
Gestarcbot2.zip	10.09.2017 14:56		51 Ko	☆☆☆☆☆	C:\Utilisateurs\bor...
					
entities	Dossier de fichiers				
intents	Dossier de fichiers				
agent.json	Fichier JSON	1 Ko	Non	1 Ko 48 %	10.09.2017 12:56
customDomainsResponses.json	Fichier JSON	1 Ko	Non	1 Ko 2 %	10.09.2017 12:56

Figure 41 : Représentation du dossier du projet Gestarcbot

## 11 Versioning

Avant tout, il faut savoir que les programmes informatiques notamment les agents intelligents tels que les chatbots devront toujours être améliorés dans des versions supérieures suivant l'évolution des besoins des utilisateurs.

En ce qui nous concerne, après avoir tester Gestarcbot, certains utilisateurs m'ont fait part du fait que la liste des informations pouvant être données par le chatbot était restreinte et qu'il fallait réfléchir sur plus d'éléments. Par exemple, ça pourrait être intéressant pour un utilisateur de savoir si c'est possible d'être bénévole lors d'un évènement.

Dans la prochaine version du chatbot, il serait aussi intéressant de faire en sorte que le chatbot puisse directement encaisser les réservations de tickets des étudiants et non pas seulement réserver des places. De plus, il serait intéressant de faire en sorte que le chatbot puisse directement reconnaître un interlocuteur s'il avait déjà eu à faire à cet utilisateur dans le passé par exemple. Cela permettrait à Gestarcbot d'éviter de

demander le nom et le prénom de l'utilisateur à chaque fois que celui-ci aura besoin des services de Gestarcbot.

En plus de réfléchir à de nouveaux services à proposer, chacun des éléments donc j'ai parlé plus haut dans ce chapitre, pourrait être implémenté dans une prochaine version de Gestarcbot, afin d'améliorer cette première version du robot conversationnel.

Maintenant que notre solution est développée, testée et fonctionnelle, je vais terminer cette partie pratique du rapport du travail de bachelor par une conclusion.

## 12 Conclusion

---

Cette partie du document, permet de faire une conclusion sur mon travail de Bachelor. Dans ce chapitre, nous allons aborder plusieurs points comme le déroulement du travail, les problèmes rencontrés ainsi que les solutions apportées. Je ferai également part de mon avis personnel sur l'utilisation des agents intelligents, au détriment des applications de gestion classiques.

### 12.1 Déroulement du travail

L'objectif de ce travail de Bachelor était dans une premier temps, de découvrir les agents intelligents particulièrement les chatbots, ainsi que de démontrer les avantages de leur utilisation. Ensuite, il a alors été question d'imaginer et de réaliser une telle solution pour faire face à un problème de gestion que pourrait rencontrer une entreprise.

Après avoir relaté une des problématiques que rencontrent les entreprises de nos jours, particulièrement de petite et moyenne taille (cf chapitre 1, partie théorique), la réalisation d'un chatbot m'a paru être une solution idéale.

L'utilisation des chatbots dans les entreprises est relativement nouveau et les moyens pour les développer se démocratisent de plus en plus. Par conséquent, pour pouvoir mener ce travail à bien, j'ai tout d'abord dû effectuer beaucoup de recherches, notamment sur les technologies, méthodologies et les bonnes pratiques utilisées pour développer des chatbots. Ces recherches m'ont permis de tester plusieurs plateformes de développements pour pourvoir choisir celle dont j'ai utilisé pour développer Gestarcbot.

Avant de pour développer Gestarcbot, il m'ait d'abord fallut déterminer de quelle façon il pourrait être utile pour la Gest'arc ! Après avoir eu plusieurs entretiens avec l'association, nous avons pu y ressortir les différents services dont pourrait proposer le chatbot directement sur leur page Facebook.

La collaboration avec l'association des étudiants, la Gest'arc s'est révélé être très importante car, non seulement c'était mon client, c'est-à-dire l'entité pour laquelle je devais implémenter le chatbot. Elle m'a aussi apporté beaucoup d'éléments qui m'ont beaucoup servi lors de la mise en œuvre du projet. En effet, grâce aux conversations

que la Gest'arc avait eues avec des étudiants suite à des soirées, des journées sportives ou d'autres évènements qu'elle avait organisés, j'ai pu par exemple savoir quel genre de personnalité il fallait que je donne au chatbot. De plus, ça m'a également aidé à comprendre d'avantage les bénéfices qu'un agent intelligent doté d'une forme d'autonomie pouvait avoir dans certaines situations.

Une fois que j'ai pu prendre la plateforme API.AI en main, j'ai d'abord développé la logique de notre chatbot (cf chapitre 7), sans utiliser les données persistées en base de données. Le but de cette opération était de faire tester la fiabilité et la pertinence du raisonnement de Gestarcbot par quelques personnes. Pour terminer, une fois que j'ai fini de développer le webhook, la première version de Gestarcbot était enfin prête à être testé.

Pendant la réalisation du projet, j'ai été confronté à plusieurs problèmes qu'il ait fallu que je surmonte. Dans le point suivant, nous allons aborder les différentes étapes qui m'ont posé quelques soucis.

## 12.2 Problèmes rencontrés

Durant la réalisation de ce travail de bachelor, j'ai pu être confronté à différents soucis, notamment lors de la mise en place du chatbot Gestarcbot. En effet, la rédaction des scénarios était une étape des bonnes pratiques difficiles à réalisées. La rédaction des scénarios s'est révélée être assez compliquer, car je devais imaginer des conversations entre deux personnes, sans penser à la partie développement. Malheureusement, c'était pour moi compliqué d'imaginer des conversations sans penser à la façon dont j'allais les implémenter. Dû à la difficulté de la tâche, la première version de mes scénarios ressemblait plus à un répondeur automatique qu'à une conversation entre deux personnes.

Le plus grand problème que j'ai eu finalement lors du développement de ce chatbot reste la mise en place du webhook. Comme nous avons vu dans l'architecture, le webhook est un servlet qui se connecte à la base de données et est déployé sur un serveur distant. J'ai eu beaucoup de problèmes lors des tests, car le servlet mettait beaucoup trop de temps pour se connecter à la base de données, ce qui provoquait donc un timeout du côté de la plateforme API.AI.

Suite à ces difficultés, j'ai dû trouver des solutions pour ainsi pouvoir m'adapter et continuer le projet. Dans la suite de ce chapitre, je vais présenter les différentes solutions qui m'ont permises de gérer ces situations.

## 12.3 Solutions apportées

Comme je l'ai expliqué précédemment, j'ai rencontré quelques petits problèmes pendant la réalisation de ce projet. Mais il ait fallu que je trouve des solutions face à ce problèmes pour pouvoir m'adapter et continuer le projet. En effet, lorsque j'ai rédigé la première version des scénarios sur lesquels je devais me baser pour construire notre chatbot, ceux ressemblaient davantage à une interaction entre un utilisateur et un répondeur automatique qu'à une conversation naturelle. Pour donner un aperçu de cette interaction, voici quelques lignes de ma première version du scénario Demande d'information :

**Utilisateur :** Bonjour.

**Gestarcbot :** Bonjour.

**Utilisateur :** c'est possible d'avoir une information sur un évènement ?

**Gestarcbot :** Oui biensur ! sur quel évènement

- 1- La Sécomania
  - 2- Le bal de la HEG
- Tape juste 1 ou 2 pour me dire de quel évènement tu as besoin s'il-te-plaît.

**Utilisateur :** 2.

**Gestarcbot :** Ok ça marche et de quelle information aurais-tu besoin

- 1- Les prix ?
- 2- Le lieu ?
- 3- Les horaires ?
- 4- La date ?
- 5- Le thème ?
- 6- Ou encore les photos ?

Tape juste 1 ou 2 pour me dire de quelle information tu aurais besoin s'il-te-plaît.

**Utilisateur :** 4.

**Gestarcbot :** Alors le Bal de la HEG sera le 20 novembre prochain.  
Souhaiterais-tu quelque chose d'autres ? oui ou non

On peut donc remarquer que cette interaction n'était pas du tout naturelle. Pour corriger ce problème, j'ai dû faire une conversation avec une personne réelle pour ainsi m'en inspirer et réaliser les différents scénarios que nous retrouvons dans le chapitre 3.1.3 de ce document.

De plus, pendant certains tests unitaires, je m'étais rendu compte que la connexion de mon webhook (servlet) à la base de données Mysql, prenait beaucoup trop de temps ce qui provoquait des timeouts au niveau de API.AI. Ces timeouts faisaient par conséquent crasher le chatbot. Il fallait à tout prix que je trouve un moyen pour

optimiser cette connexion et ainsi gagner en performance. Malheureusement, n'ayant pas pu optimiser ma connexion à ma base de données, en tout cas pas assez pour éviter les timeouts, je me suis tourné vers une autre solution. Après quelques tentatives infructueuses j'ai alors eu l'idée de sauvegarder mes données dans des fichiers directement dans les ressources de mon projet java.

Pour pouvoir mettre cette solution en place, j'ai dû dans un premier temps, modifier mon architecture au niveau du webhook. Les figures ci-dessous, illustrent la modification que j'ai dû apporter à mon architecture.

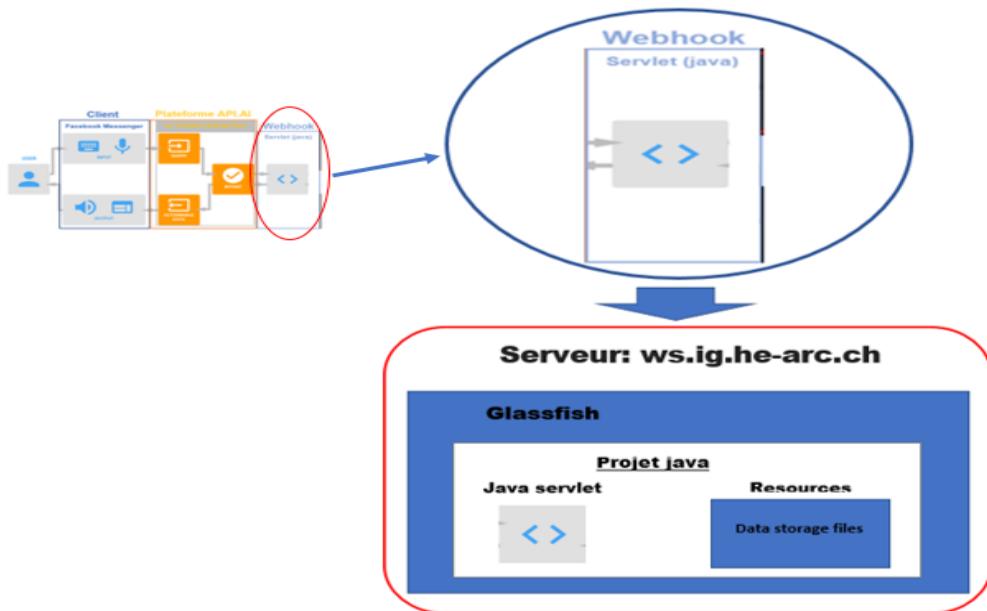


Figure 42 : Nouvelle architecture suite à la solution apportée pour le problème de connexion à la base de données

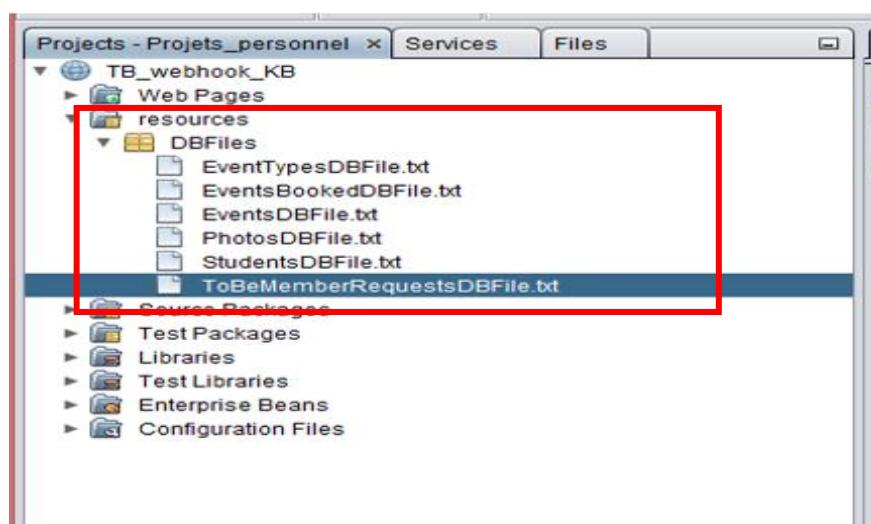


Figure 43 : Présentation des fichiers de stockage de données dans le projet java

Bien que l'idée soit fonctionnelle, elle ne demeure pas optimale car il y a beaucoup de risques. En effet, les fichiers pourraient être détruits ou corrompus, ce qui mènerait vers une perte de données. De plus, stocker ses données dans une base de données relationnelle comme Mysql est beaucoup plus simple et efficace pour les traitements de données, tels que la recherche, l'insertion et la modification. De plus cela permet également d'éviter des doublons dans ses données de manière plus efficace et d'éviter différents types de contraintes.

Suite à tous ces points évoqués il sera donc essentiel de retravailler sur le stockage des données de Gestarcbot lors des prochaines versions.

Dans la dernière partie de cette conclusion, je vais donner mon avis personnel sur l'utilisation des chatbot dans les entreprises. Je vais également parler de mon expérience et de ce que la réalisation de ce travail a pu m'apporter.

## 12.4 *Mon avis personnel*

La réalisation de ce projet m'a permis non seulement d'utiliser les connaissances que j'ai acquises tout au long de ma formation à la Haute école de gestion, que ce soit sur la gestion de projet ou encore dans le développement applicatif. De plus, ce travail de bachelor m'a permis de découvrir et d'explorer le domaine des agents intelligents. Un domaine que je ne connaissais alors que de nom. En effet, pour réaliser ce travail, j'ai dans un premier temps décidé d'utiliser la méthodologie Scrum. Scrum est une méthodologie de gestion de projet par itération, qui nécessite une implication du client dans la réalisation du projet. Après quelques semaines seulement, je me suis rendu compte que c'était compliqué d'utiliser une méthodologie destinée à une équipe quand on est seul pour réaliser un projet, Scrum n'est pas la solution miracle à tous les problèmes ! J'ai alors décidé d'utiliser une méthodologie plus classique, car la réalisation en cascade de Gestarcbot s'est alors révélé être la méthode la plus adéquate pour la réalisation de mon projet.

En ce qui concerne les agents intelligents, à mon avis c'est une technologie qui sera de plus en plus utilisée par les entreprises, de grande et moyenne taille et dans tous les domaines. Bien qu'aujourd'hui, beaucoup de personnes voient en cette technologie un mauvais augure, car plusieurs corps de métiers seront probablement supprimés suite à l'utilisation des chatbots vocaux ou textuels. Les postes comme les opérateurs de call center ou encore certains postes de conseil en ligne seront certainement les plus touchés, ce qui causera une hausse du taux de chômage dans certaines régions du globe. Mais il ne faut pas oublier que ces agents intelligents sont des programmes informatiques et qu'ils nécessitent une maintenance, ce qui signifie donc de nouveaux postes à pouvoir. C'est vrai que ces nouveaux postes seront proportionnellement inférieurs au nombre des postes qui seront supprimés mais tâchons de ne pas oublier qu'au cours de l'histoire, l'évolution des sciences et des technologies ont souvent conduit à la disparition et à l'apparition de bon nombre de corps de métiers. C'est par exemple le cas des postes comme les messagers, qui étaient le moyen de communication avant l'invention des téléphones et télégraphes. On peut également parler des ramasseurs de quilles, dont le travail était de ramasser et de redresser les quilles de bowlings. Je pense que l'utilisation des agents intelligents et des chatbots par les entreprises est une bonne chose, car ils sont beaucoup plus bénéfiques pour les entreprises et leurs clients que les applications plus classiques. En effet, les agents

intelligents rendraient par exemple les entreprises plus attractives, plus efficaces comme nous avons pu le voir dans la partie théorique ou encore plus réactives vis-à-vis de leurs clients comme nous avons pu le voir grâce à Gestarcbot. De plus, contrairement à une application de gestion classique, un chatbot a la possibilité de simuler des émotions à travers la communication qu'il entretien avec un client, ce qui donne l'impression de parler à un vrai collaborateur de l'entreprise qui l'utilise.





## Partie administrative

**HEG Arc - Haute école Arc - Gestion**

**Mention :** Travail de Bachelor d'informaticien de gestion

**Module :** 656-1.1 Thèse de bachelor

**Sujet :** Exploration dans le domaine des agents intelligents et des chatbots

**Directeur de travail :** Frischer Boris

**Auteur :** Klett Boris

**Cycle d'étude :** 2013 - 2017

**Date de restitution :** Septembre 2017





Klett Boris

**Module**

Thèse de bachelor

**Filière**

Informatique de Gestion

**Directeur de travail**

Boris Fritscher

**Cycle d'étude**

2013 – 2017

**Date de restitution**

06.2017

**Version**

0.4

**Demande de  
Ratification**



# 11 Introduction

---

L'objectif de ce document est de démontrer que l'étudiant a bien assimiler la problématique mis en avant par le sujet et par conséquent, de proposer une ou des solutions pour sa résolution.

Dans ce document, je vais aborder dans un premier temps un type de problématique qui peut nécessiter l'utilisation des agents intelligents et particulièrement des chatbots. Ensuite, je vais parler des différentes étapes dont nécessite la mise en œuvre de la solution que je proposerai. En effet, il faut savoir que pour mettre en place un chatbot, plusieurs étapes sont nécessaires. Contrairement à ce qu'on pourrait penser, la partie la plus importante n'est pas le choix des technologies, mais la définition de la fonction métier ou principale du robot conversationnel. En effet, il s'agit donc de choisir les cas d'utilisation ou le domaine, définir des scénarios et définir la personnalité virtuelle du chatbot. Bien que ce ne soit pas le point le plus important, il est tout de même essentiel de connaître la technique et les technologies que j'utiliserais. Aujourd'hui, plusieurs technologies pour développer des chatbots sont disponibles sur le marché et l'objectif sera de choisir la ou les technologies les plus adéquates par rapport à mes compétences et surtout qui me permettront de réaliser le projet dans les délais prévus.

En plus des éléments dont je viens de faire part, il y'a encore d'autres points qui sont importants dans la réalisation de mon projet, qui sont les objectifs et les livrables. En effet, il est important de savoir si ce que je prévois de faire va pallier à la problématique identifiée. Par ce fait, je parlerai de mes objectifs et surtout des livrables que je vais proposer pour résoudre la problématique.

## 11.1 Présentation du sujet

Pour mieux comprendre et appréhender le sujet, nous devons dans un premier temps et en quelques phrases comprendre ce qu'est un agent intelligent, ce qui nous permettra par la suite de mieux comprendre ce qu'est un chatbot. Un agent intelligent, est un programme informatique capable d'interagir, guider et informer des utilisateurs humains ou informatisés. Il est capable de faire preuve d'intelligence en réalisant un certain nombre de tâches de manière autonome. Dotés d'une intelligence artificielle, on distingue plusieurs types d'agents intelligents tels que : des agents collaboratifs, d'interface, mobiles, d'internet ou encore de recherche d'informations. De nos jours, cette technologie est utilisée dans beaucoup de domaines parmi lesquels on peut citer : l'automobile, la robotique, la santé, l'éducation, l'internet des objets<sup>16</sup>, et bien d'autres encore. Quant au chatbot, c'est un programme informatique, une typologie d'agent intelligent, capable d'interagir, communiquer vocalement ou textuellement avec un utilisateur, via des messageries, sites web ou encore des robots.

Pour être intelligents, ils utilisent des mécanismes tels que : le machine learning (ML) ou encore le natural language processing (NLP). Aujourd'hui, cette technologie est utilisée par des entreprises et des particuliers principalement pour l'assistance

---

<sup>16</sup> La traduction française de l'expression anglaise *internet of things* « IOT ».

personnelle, la vente ou encore le coaching. Encore en phase de croissance, l'utilisation des chatbots n'est pas encore prêt à s'arrêter, un élément important quand on sait qu'aujourd'hui plus du deux tiers de la population mondiale utilisent les messageries instantanées pour communiquer.

Maintenant que nous savons de quoi parle le sujet et en quoi consiste ce document, dans la prochaine partie nous allons aborder l'une des problématiques dont l'utilisation des agents intelligents et des chatbots nous permettra de résoudre.

## 11.2 Problématique

De manière générale, il faut savoir qu'aujourd'hui, de plus en plus de gens utilisent internet et dans tous les domaines.

En parallèle à cette utilisation massive de la toile, nous pouvons observer une augmentation considérable du nombre d'applications disponibles sur le marché pour satisfaire au mieux tous les besoins. En effet, il existe aujourd'hui un nombre important de logiciels et entre les applications de e-commerce et de vente en ligne, de services tels que les réservations de tout type (vols, hôtels ou encore de restaurant, etc.). On observe surtout une augmentation constante quant à l'utilisation des applications et surtout des messageries instantanées par rapport à d'autres applications.

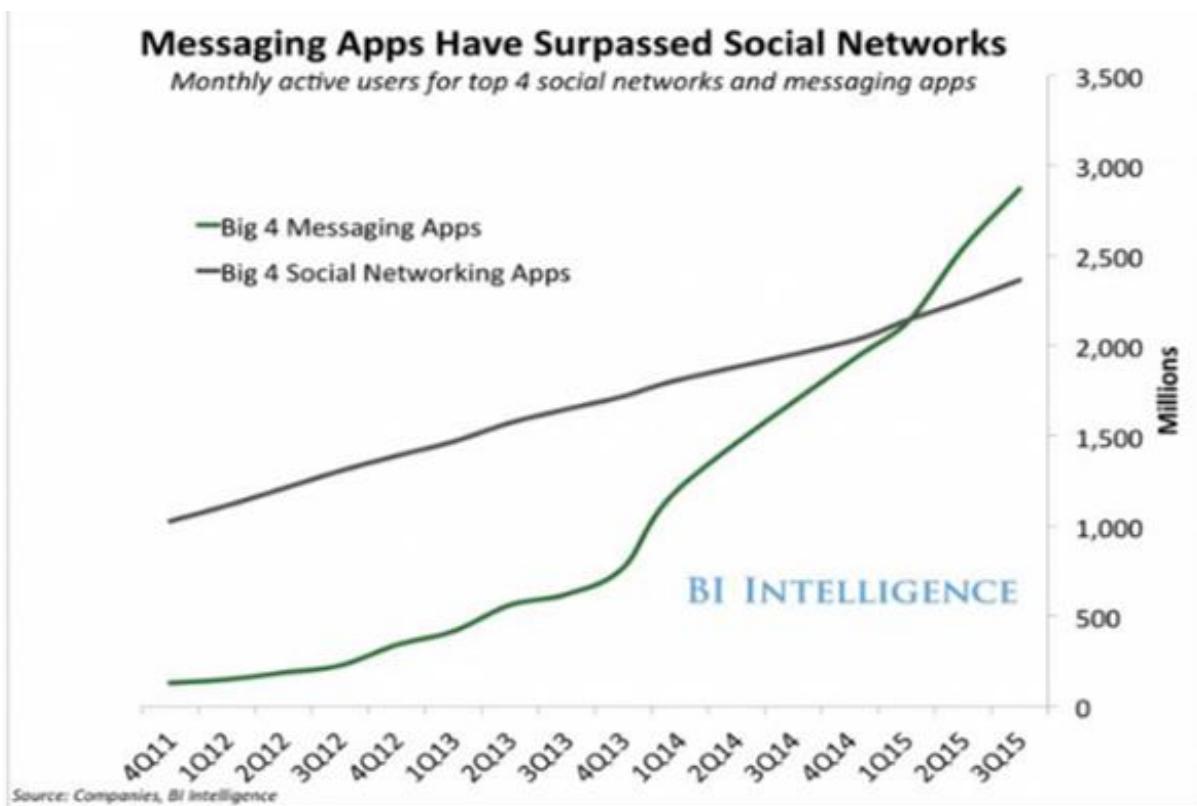


Figure 44 : Courbes de progression de l'utilisation des applications de messagerie et des réseaux sociaux ([Schlicht, 2016](#)).

Source: [http://www.schlicht.com/2016/01/05/messaging-apps-have-surpassed-social-networks/](#)

Aujourd’hui, de plus en plus d’entreprises allant des grandes multinationales aux associations communautaires, sociales ou autres ont bien compris qu’il est maintenant essentiel voir même vital de mettre en place des stratégies de communication par messageries instantanées avec leurs clients ou affiliés pour mieux les atteindre et surtout pour faire face à l’évolution de l’environnement technologique.

Mais, quand bien même ces instances mettent en place un tel dispositif de communication, faut-il encore que les requêtes des principaux intéressés soient traitées à la minute voir à la seconde pour que ceux-ci puissent se sentir être pris en charge et satisfaits de la réactivité en termes de communication de l’entreprise. Hors, par manque de ressources, cela n’est pas toujours le cas, ce qui peut même causer dans certaines situations, la perte de clients, voir une mauvaise réputation pour l’entreprise si la situation est de manière fréquente.

Dans notre situation, on cherche à mettre en place une technologie de type agent intelligent et chatbot, à disposition d’une entreprise ou d’une association pour que celle-ci puisse profiter pleinement des avantages qu’offre cette technologie.

Dans la prochaine partie, nous allons voir quelle solution je propose quant à notre problématique.

## 12 Proposition de solution

---

Dans le cadre de ce travail, je vais travailler avec une entreprise de petite structure, pour leur proposer une solution qui leur permettra de prendre en charge très rapidement les demandes de leurs clients directement sur le Messenger de leur page Facebook. En effet, de nos jours, les gens veulent de plus en plus une prise en charge de leurs demandes de manière rapide voire instantanée dans certaines situations. Hors, les institutions de petite structure ne disposent pas toujours de ressources en terme humaines ou technologiques pour répondre à cette tendance croissante. Le but de cette partie est de présenter un moyen « Technologique », qui permettra une meilleure prise en charge des requêtes des clients via une application de messagerie instantanée telle que Facebook Messenger. Cette prise en charge englobe donc des réponses aux questions, informer ou encore rediriger la clientèle.

### 12.1 Solution

Pour répondre à la problématique évoquée plus haut, pour notre cas, nous allons mettre en place un Chatbot. En effet, pour éviter les différents problèmes de disponibilité des membres ou collaborateurs de notre client. Une situation qui entraînerait un manque de réactivité, je propose alors de développer un chatbot qui sera centré dans un domaine précis. Le chatbot étant un robot conversationnel, un type d’agent intelligent, il sera disponible à tout moment et à tout heure et sera capable de répondre aux besoins des clients de mon mandataire. De plus, ce robot permettra aux collaborateurs de l’entreprise de prendre moins de temps pour répondre

directement à leur public cible et avoir davantage de temps pour se consacrer à d'autres activités.

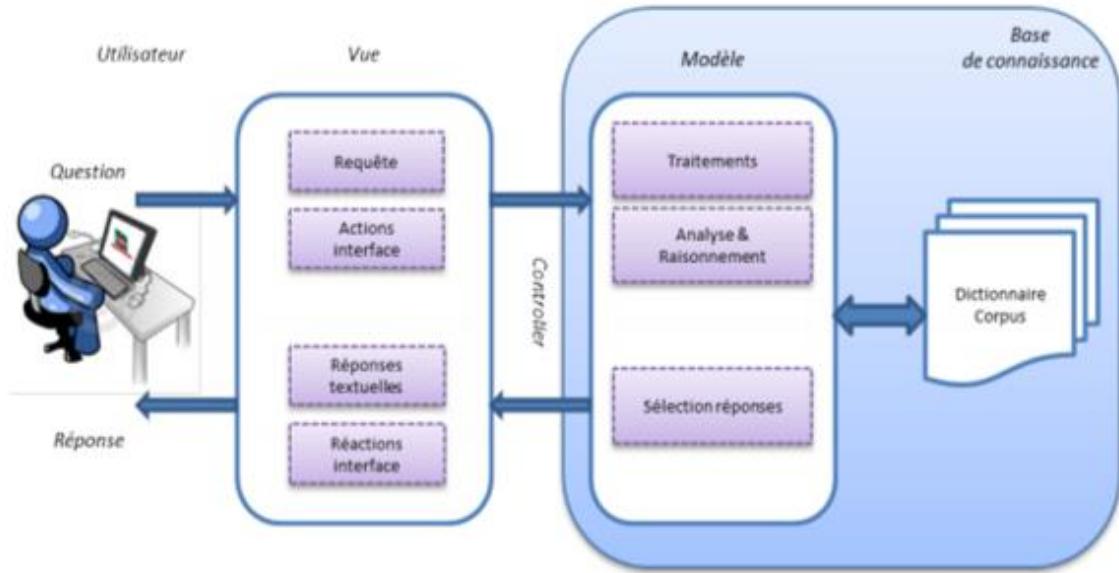


Figure 45 : Fonctionnement de l'application Chatbot ([GRANET, 2015](#)).

La figure 45 ci-dessus montre l'interaction entre l'utilisateur (le client de l'entreprise) et le chatbot. Pour mieux expliquer le client interagit avec le chatbot via une vue comme une plateforme de messagerie comme Facebook Messenger dans notre cas. La partie dite « controller » sera plutôt la partie de l'intelligence artificielle c'est là que le chatbot traite les demandes de l'utilisateur.

## 12.2 Technologies utilisées

Pour développer un chatbot, plusieurs étapes sont nécessaires. Des étapes dont nous verrons les principales dans la partie « Démarche projetée ». Parmi ces étapes, on a également le choix des technologies, bien que cette étape ne soit pas la plus importante, elle est quand même à ne pas prendre à la légère. En effet, il faut accorder une certaine attention dans le choix de la plateforme sur laquelle on souhaiterait communiquer avec les clients. Dans mon travail ça n'a pas été compliqué car après avoir discuté avec mon client actuel, j'ai pu apprendre qu'ils sont le plus souvent sollicité depuis Facebook Messenger de leur page Facebook par leurs clients, ce qui m'a donc permis de tout naturellement porter mon choix vers Facebook Messenger. Ensuite, comme nous le savons, à l'heure actuelle dans le marché, il existe une multitude de frameworks pour développer des chatbots, par conséquent le choix du chatbot que je prévois d'utiliser pour développer le chatbot dépend de certains critères. Après avoir comparé quelques outils, je me suis aperçu que les outils sont plus ou moins similaires. Cependant, après avoir testé ces outils, pour moi le framework API.AI est l'un des plus intuitifs et moins compliqués à prendre en main par rapport à mes compétences de développeur. Pour réaliser ce travail, je pense personnellement que

API.AI est un bon candidat pour construire rapidement un Chatbot doté d'une intelligence artificielle pour interagir avec des gens.

## 13 Démarche projetée

---

### 13.1 Méthodologie utilisée

Pour réaliser ce projet, les préceptes agiles seront employés, notamment dans l'implication de l'utilisateur final et la réalisation de prototypes et l'utilisation de la méthodologie Scrum pour la réalisation du projet. Des prototypes seront livrés fréquemment dans la mesure du possible, dans le but de proposer quelque chose de tangible. Mon client pourra alors exprimer plus facilement ses besoins, cela facilitera la rétroaction et le chatbot correspondra plus facilement à ses besoins.

### 13.2 Les différentes étapes de la mise en œuvre du chatbot

Comme nous avons vu dans le chapitre précédent, la mise en œuvre d'un chatbot se fait en plusieurs étapes. Selon Google developers, les étapes les plus importantes, à réaliser en premier lieu sont :

- Choisir des cas d'utilisation ou choisir le domaine du chatbot
- Définir une personnalité pour le chatbot
- Ecrire des dialogues avec des utilisateurs (Developers, 2017)

#### 13.2.1 Choisir des cas d'utilisation ou domaine d'application

Cela consiste concrètement à définir le rayon d'action de notre chatbot, de spécifier dans quel domaine il sera utile pour l'utilisateur. Pour m'aider, je m'inspirerai des user stories que j'écrirai avec l'aide de mon client.

#### 13.2.2 Définir une personnalité pour le chatbot

La personnalité m'aidera à construire et à écrire des conversations entre le chatbot et les utilisateurs. Alors définir une personnalité de manière assez tôt dans la mise en œuvre me permettra de choisir les bons mots, syntaxe et structures des phrases que le chatbot emploiera. Les caractères de personnalité qu'aura le chatbot sont les suivantes :

- Optimiste, joyeux et encourageant
- Engagé à aider
- Jamais formel et utilise un langage simple pour atteindre un plus large groupe de personne, jeune et moins jeune

### **13.2.3 Ecrire les scénarios**

C'est probablement la phase la plus importante et l'une des moins évidentes. Etant donné que le chatbot n'a pas d'interface graphique comme les sites web ou d'autres applications, sa qualité sera donc jugée sur sa capacité à tenir une conversation et à apporter des informations pertinentes quant aux demandes de l'utilisateur. Par conséquent, un gros travail sur des scénarios est essentiel lors du développement d'un chatbot de qualité. Selon l'article de T.Cantet, il faut savoir que plusieurs éléments sont à respecter quand un chatbot converse avec un humain. Ces points à respecter sont :

#### **Ne pas mentir aux utilisateurs**

Même si les chatbots ne réussissent pas encore à se faire passer pour des humains, mieux vaut qu'ils disent spontanément ce qu'ils sont à l'utilisateur. Surtout toujours donner des informations vraies.

#### **Humaniser le chatbot**

Une fois le point précédent éclairci, il est souhaitable de travailler la façon dont notre chatbot "parle" pour qu'il ne "sonne" pas trop robotique.

#### **Se présenter**

Comme évoqué précédemment, il est recommandé que le chatbot se présente dès le début de la conversation. Lui donner un nom permet déjà de créer une première accroche. Ensuite, c'est l'occasion d'expliquer ce qu'il sait faire et ce qu'il peut apporter comme informations.

#### **Pas trop bavard**

Il ne faut pas que ses réponses soient des tirades à n'en plus finir et inversement, il ne faut pas non plus que notre chatbot soit trop verbeux.

#### **Expliquer qu'une réponse est attendue**

Pour s'assurer que l'étudiant donne une réponse, notre chatbot doit explicitement lui demander de faire un choix.

#### **Ne pas présumer que l'utilisateur sait ce qu'il doit faire**

Il est nécessaire au chatbot de guider la conversation en indiquant quelles options sont disponibles.

#### **Ne pas toujours demander de confirmation**

Pour ne pas ajouter trop de lourdeur lors de la conversation, il n'est pas nécessaire que notre chatbot demande toujours une confirmation.

#### **Obtenir une information à la fois**

Un chatbot doit demander les informations qu'il a besoin une par une et non pas toutes en même temps.

#### **Gérer les impasses**

Un chatbot n'est jamais capable de répondre à tout. Même si on arrivait à concevoir une conversation fluide, il arrivera toujours un moment où le chatbot se retrouvera dans une impasse.

Il arrivera probablement que des utilisateurs posent intentionnellement des questions hors du périmètre du chatbot ou essaieront de le piéger. Répondre simplement par une phrase trop générique comme "Désolé, je ne comprends pas" ne sera peut-être pas l'idéal.

### **Commencer simple**

Il est important que notre chatbot sache répondre efficacement à quelques questions dans un domaine précis puis d'élargir ses possibilités une fois que les utilisateurs commencent à vouloir d'autres informations.

### **Rien ne vaut une image**

Parfois, au lieu d'essayer de décrire un objet, il vaut mieux afficher une photo, un flyer par exemple.

### **Les émoticônes**

De même que les images véhiculent plus facilement certains types d'information que des paragraphes descriptifs, les émoticônes sont le moyen le plus simple de transmettre des émotions. Il est néanmoins nécessaire de ne pas non plus en abuser.

### **Gérer le contexte**

Afin de ne pas redemander des informations qui ont déjà été évoquées précédemment, il est important que notre chatbot mémorise des réponses ou des éléments de contexte lors de la conversation.

### **Rendre la main à un vrai humain**

Il faut également que notre chatbot puisse rendre la main à un collaborateur si besoin est ! (Cantet, 2016b)

Bien entendue plusieurs scénarios seront travaillés en collaboration avec mon client final. Les meilleurs scénarios seront retenus pour ainsi constituer le flux conversationnel du futur chatbot.

Un autre point qui tout aussi important mais là qui dépend du framework que je vais utiliser est la capacité du chatbot à récupérer les informations dont les utilisateurs auront besoin. Il faut savoir que les étapes que je viens de mentionnées sont les plus importantes et qu'elles précèderont le développement et le déploiement.

Maintenant que la démarche de la réalisation du projet a été vue, dans la prochaine partie du document nous allons voir quels sont livrables que je compte fournir à la fin du projet, qui sont les éléments les plus importants du projet.

## **14 Livrables**

---

Au terme de ce travail, je fournirai plusieurs livrables. En effet je prévois de livrer un chatbot fonctionnel qui palliera à la problématique que j'ai mis en évidence plus haut dans le document et un document qui relatera plusieurs éléments, notamment l'architecture du programme et la modélisation des différents processus, ceci dans le but de justifier mes choix. Certes, un Chatbot fonctionnel fait partie des livrables demandés dans le cadre de mon travail mais c'est aussi un excellent moyen pour démontrer une partie du potentiel qu'offre cette technologie en pleine croissance.

Ces livrables auront pour but dans un premier temps de permettre à mon client d'avoir un robot conversationnel fonctionnel ou du moins d'avoir un prototype dans le cas où ça demanderait des ressources financières conséquentes ou d'avantage de temps pour développer un Chatbot fonctionnel. De plus, appuyé par le rapport de thèse, les différents modèles et architecture permettront de démontrer et de justifier les technologies et procédés dont j'ai eu à utiliser pour implémenter le Chatbot.

Dans le prochain point nous parlerons des objectifs du projet et dans quelle mesure la ou les solutions que nous avons entreprises auront permis de pallier à notre problématique.

## 15 Objectifs

---

L'objectif premier de mon projet de Bachelor est de réaliser une application autour des agents intelligents et des Chatbots, qui permettra de résoudre un problème de gestion dont une entreprise ou une association communautaire serait susceptible de faire face. Ce travail a également pour but de me permettre de mettre en pratiques mes compétences dans les méthodologies de gestion de projet, de développement et de recherche que j'ai acquis tout au long de ma formation à la Haute école de gestion. Je vise également à ce que mon application soit utilisée, jugée et testée par des utilisateurs, dans le but de me permettre de faire des recommandations sur le développement et l'utilisation de Chatbots en fonction des contextes. Ces objectifs sont pour la plupart ceux décrits dans le mandat du projet.

## 16 Planification

---

En ce qui concerne la planification du projet, je compte utiliser la méthodologie Scrum pour la gestion de ce projet. Par conséquent, j'ai esquissé une liste de stories qui seront bien entendu répartie dans plusieurs sprints de deux semaines. Les stories que j'ai définies sont les suivantes :

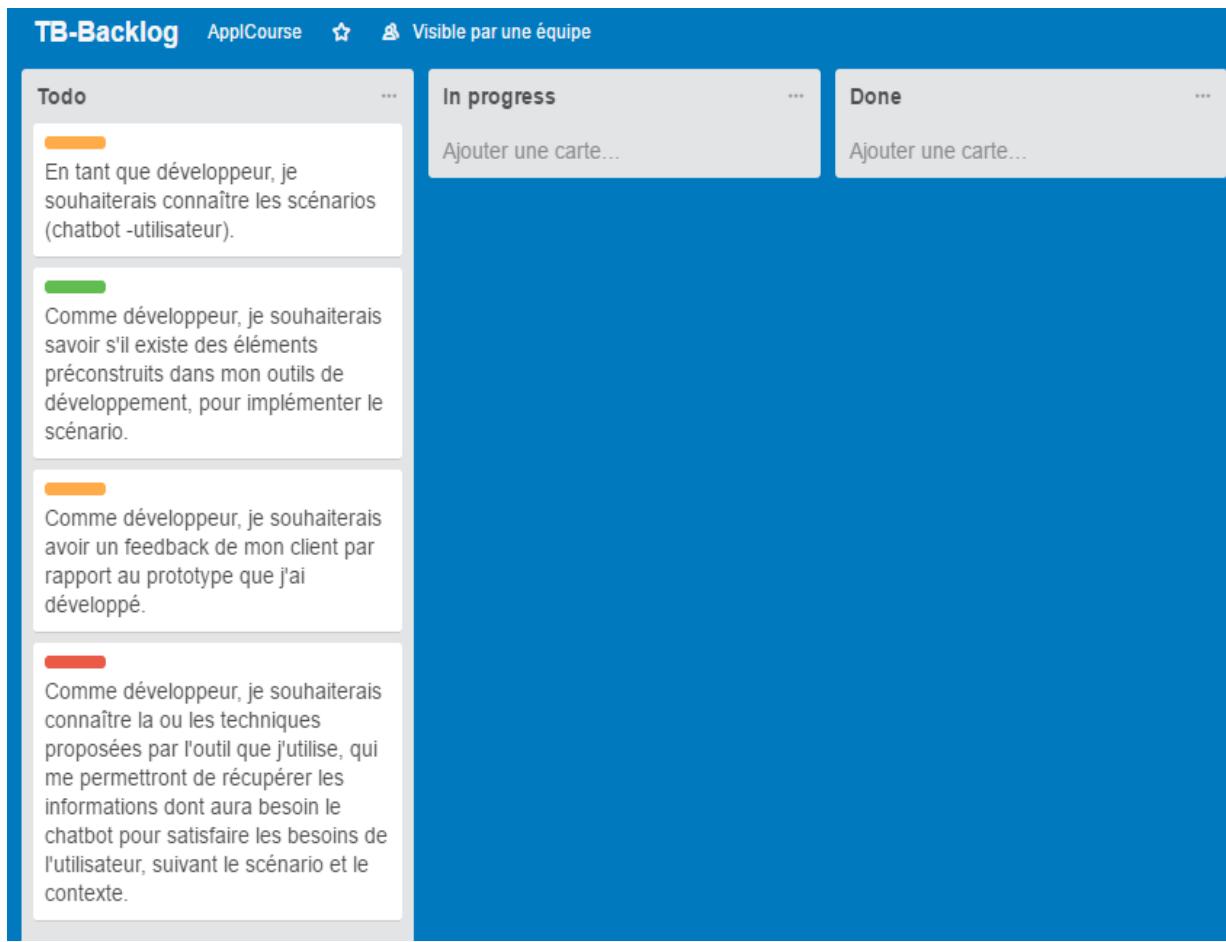


Figure 46 : Product backlog

Pour illustrer le backlog « TB-Backlog » ci-dessus, les couleurs dans les cartes « Stories », correspondent aux niveaux de difficulté des stories. Ces niveaux de difficultés peuvent être assimilés aux tailles de vêtements. La couleur verte correspond à la taille « S », la couleur Jaune correspond à « M », la couleur orange à la taille « L » et la couleur rouge à la taille « XL ».

L'idée quant à l'utilisation de cette méthodologie, est d'implémenter des scénarios et des fonctionnalités pour le chatbot de manière itérative et en collaboration avec le client, ce qui me donnera une certaine flexibilité sur ceux-ci.

Comme on peut le remarquer, ces stories sont axées sur le développeur. En effet, je pars du principe que les stories que je viens de citer sont assez généralisées pour me permettre de développer tout type de chatbot sans avoir besoin de changer complètement mes stories. Bien sûr, comme le prévoit la méthodologie, je compte m'entretenir régulièrement avec mon client pour pouvoir travailler avec lui sur les scénarios et les fonctionnalités dont il faudra implémenter pour le chatbot.

La difficulté principale sera la réalisation de l'intelligence artificielle. C'est l'élément le plus important du développement car il permet au chatbot fournir des informations pertinentes et attendu par l'utilisateur final. Je devrais donc particulièrement m'y

attarder car sans ça le chatbot serait inutile. Il sera également important de prendre en compte les cas limites car les utilisateurs finaux pourraient être emmené à demander des éléments ou poser des questions qui n'entreront probablement pas toujours dans le cadre des compétences du chatbot.

# 17 Authentification

---

Ce document a été lu et approuvé par le(s) responsable(s) du projet.

**Nom du projet :** Exploration dans le domaine des agents intelligents et des chatbots

**École :** HEG Arc - Haute école Arc - Gestion

Les soussigné(e)s :

**Direction de filière**

**Prénom et nom :** Fabrice Camus

**Signature :** \_\_\_\_\_, **Date :** \_\_\_\_\_

**Directeur de travail**

**Prénom et nom :** Boris Fritscher

**Signature :** \_\_\_\_\_, **Date :** \_\_\_\_\_

**Étudiant**

**Prénom et nom :** Boris Klett

**Signature :** \_\_\_\_\_, **Date :** \_\_\_\_\_

Par leur signature, le responsable de filière et le directeur de travail valident la démarche proposée et en aucun cas le contenu détaillé de la demande de ratification.



# Bibliographie

**HEG Arc - Haute école Arc - Gestion**

**Mention :** Travail de Bachelor d'informaticien de gestion

**Module :** 656-1.1 Thèse de bachelor

**Sujet :** Exploration dans le domaine des agents intelligents et des chatbots

**Directeur de travail :** Frischer Boris

**Auteur :** Klett Boris

**Cycle d'étude :** 2013 - 2017

**Date de restitution :** Septembre 2017



# 18 Bibliography

---

- API.AI. (2016). Documentation. Retrieved from <https://docs.api.ai/docs/welcome>
- Baldit, P. (1997). *Les agents intelligents: qui sont-ils? Que font-ils? Où nous mènent-ils?*
- Bathelot, B. (2016). Définition : Chatbot. Retrieved from <http://www.definitions-marketing.com/definition/chatbot/>
- BBVOPEN4U. (2016). How to develop a chatbot for Facebook Messenger with Chatfuel. Retrieved from <https://bbvaopen4u.com/en/actualidad/how-develop-chatbot-facebook-messenger-chatfuel>
- Bouzy, B. (n.d.). *Intelligence Artificielle Agents Intelligents.*
- Canete, C. (2016). How chatbots work ? Retrieved from <https://applidium.com/en/news/abchatbots/>
- Cantet, T. (2016a). Les chatbots. Retrieved from <http://blog.octo.com/les-chatbots/>
- Cantet, T. (2016b). Règles de conversation pour un chatbot. Retrieved from <http://blog.octo.com/regles-de-conversation-pour-un-chatbot/>
- CCM. (2017). Réseaux - Architecture client/serveur à 3 niveaux. Retrieved from <http://www.commentcamarche.net/contents/221-reseaux-architecture-client-serveur-a-3-niveaux>
- Clustaa. (2016). UN CHATBOT N'EST PAS CONÇU POUR REMPLACER L'HUMAIN. Retrieved from <http://clustaa.com/blog/interview-de-philippe-duhamel-interactions-digitales/>
- Crochet-Damais, A. (2016). Google rachète API.AI : une techno de bot. Retrieved from <http://www.journaldunet.com/solutions/cloud-computing/1185007-google-rachete-une-techno-pour-developper-des-bots-mobile/>
- Developers, G. (2017). Learn how to design great conversational experience. Retrieved from <https://developers.google.com/actions/design/principles>
- Encausse, J.-P. (2016). ChatBot et Agents Conversationnels. Retrieved from <https://www.tapatalk.com/topic/1121079-encausse-net/db029482cdf765f462d25184f5bbba24-chatbot-et-agents-conversationnels>
- Google. Applications sans serveur sur l'infrastructure Google (2017). Retrieved from <https://cloud.google.com/functions/?hl=fr>
- Google. (2017b). Google Image. Retrieved from <https://www.google.ch/search?q=Architecture+d%27un+chatbot&espv=2&biw=1600&bih=1094&tbo=isch&imgil=bnqW7lqtSLrrgM%253A%253BE1PmZeBluJyszM%253Bhttps%25253A%25252F%25252Fblog.steamulo.com%25252Fvous-navez-pas-d-amis%25252525C3%25252525A9veloppez-un-chatbot>
- GRANET, A. (2015). *Création d'un agent conversationnel pour orienter les clients de la MAIF.* Retrieved from [http://pagesperso.lina.univ-nantes.fr/~cdlh/ATAL/Rapports\\_2015/Granet\\_Adeline.pdf](http://pagesperso.lina.univ-nantes.fr/~cdlh/ATAL/Rapports_2015/Granet_Adeline.pdf)
- Hirst, R. (2016). Siri, Echo and Google Home: are digital assistants the future of the office? Retrieved

- from <https://www.theguardian.com/media-network/2016/nov/21/siri-echo-google-home-digital-assistants-future-office>
- Huffman, S. (2016). Making Conversational Interfaces Easier to Build. Retrieved from <https://developers.googleblog.com/2016/09/making-conversational-interfaces-easier-to-build.html>
- IBM. (2016). Building better bots with Watson Conversation. Retrieved from <https://www.ibm.com/blogs/watson/2016/07/building-better-bots-watson-conversation/>
- Influencia.net. (2016). Le chatbot ou l'obsolescence de 80% des sites web. Retrieved from <http://www.influencia.net/fr/actualites/tendance,observatoire-influencia,chatbot-obsolescence-80-sites-web,6398.html>
- Insee. (2014). *L'internet de plus en plus prisé, l'internaute de plus en plus mobile*. Retrieved from <https://www.insee.fr/fr/statistiques/1281312#titre-bloc-8>
- Institute, J. S. (n.d.). A Survey of Chatbot Systems through a Loebner Prize Competition. Retrieved from [http://nl.ijs.si/isjt12/proceedings/isjt2012\\_06.pdf](http://nl.ijs.si/isjt12/proceedings/isjt2012_06.pdf)
- Kiser, M. (2016). Introduction to Natural Language Processing (NLP) 2016. Retrieved from <http://blog.algorithmia.com/introduction-natural-language-processing-nlp/>
- Le Calme, S. (n.d.). Microsoft explique les raisons qui ont fait de son chatbot Tay un fan d'Hitler L'entreprise parle d'une « attaque coordonnée ». Retrieved from <http://www.developpez.com/actu/97313/Microsoft-explique-les-raisons-qui-ont-fait-de-son-chatbot-Tay-un-fan-d-Hitler-l-entreprise-parle-d-une-attaque-coordonnee/>
- Logbo, A. (2016). Chatbots : pourquoi la machine est-elle devenue la star de la relation client ? Retrieved from <http://www.lesechos.fr/idees-debats/cercle/cercle-159055-chatbots-pourquoi-la-machine-est-elle-devenue-la-star-de-la-relation-client-2015613.php>
- Meyer, E. (2016). Les bots, une ère post-applications qui commence maintenant. Retrieved from [http://www.easypartner.fr/blog\\_les-bots,-une-ère-post-applications-qui-commence-maintenant\\_63\\_1](http://www.easypartner.fr/blog_les-bots,-une-ère-post-applications-qui-commence-maintenant_63_1)
- Microsoft. (2016a). Bot Framework documentation. Retrieved from <https://docs.botframework.com/en-us/>
- Microsoft. (2016b). Preview pricing. Retrieved from <https://www.microsoft.com/cognitive-services/en-us/pricing>
- Nospoon. (2017). #ChatBot / Les meilleures plateformes pour créer un chatbot sans coder !No Title. Retrieved from <https://nospoon.fr/creer-chatbot-sans-coder/>
- Ondrisek, B. (2016). Mica, the Hipster Cat Bot — Four Month After The Launch. Retrieved from <https://medium.com/@electrobabe/mica-the-hipster-cat-bot-four-month-after-the-launch-dc102270eaaf#.s9d9pkoxq>
- Pandorabots. (2016). Build a chatbot. Retrieved from <http://www.pandorabots.com/>
- Pesenti, J. (2015). IBM Watson Language Translation and Speech Services – General Availability. Retrieved from <https://developer.ibm.com/watson/blog/2015/07/06/ibm-watson-language-translation-and-speech-services-general-availability/>
- Quora. (2017). What is the typical architecture of an AI chatbot? Retrieved from <https://www.quora.com/What-is-the-typical-architecture-of-an-AI-chatbot>

- Robles, P. (2016). Five pioneering examples of how brands are using chatbots. Retrieved from <https://econsultancy.com/blog/68046-five-pioneering-examples-of-how-brands-are-using-chatbots/>
- Ropars, F. (2016). Chatfuel : un outil en ligne gratuit pour créer un bot Facebook simplement. Retrieved from <http://www.blogdumoderateur.com/chatfuel-bot-facebook-gratuit-simplement/>
- Russell, J. (2016). Chatfuel lets publishers — and anyone — build bots for messaging apps. Retrieved from <https://techcrunch.com/2016/03/18/chatfuel-lets-publishers-and-anyone-build-bots-for-messaging-apps/>
- Schleiffer, R. (2002). Intelligent agents in traffic and transportation. *Transportation Research Part C Emerging Technologies*, 10(6), 325–329.
- Schlicht, M. (2016). The Complete Beginner's Guide To Chatbots. Retrieved from <https://chatbotsmagazine.com/the-complete-beginner-s-guide-to-chatbots-8280b7b906ca#.9gvjvvwwt>
- Shaikh, S. (2016). 9 Tools To Create Your Own Chatbots. Retrieved from <http://codecondo.com/9-tools-to-create-your-own-chatbots/>
- Stanford University. (2016). *Artificial Intelligence and Life in 2030*. Retrieved from [https://ai100.stanford.edu/sites/default/files/ai\\_100\\_report\\_0901fnlc\\_single.pdf](https://ai100.stanford.edu/sites/default/files/ai_100_report_0901fnlc_single.pdf)
- THECHATBOTFACTORY. (n.d.). THECHATBOTFACTORY. Retrieved from <http://www.thechatbotfactory.com/agence>
- Tu Sang, N. (2014). DEMAIN, LES CHATBOTS AU CENTRE DE LA RELATION CLIENT. Retrieved from <https://www.insurancespeaker-wavestone.com/2016/07/demain-chatbots-centre-de-relation-client/>
- University, H. (2015). *One Hundred Year Study on Artificial Intelligence*.
- Vallin, A. (2016). Chatbot : pourquoi les entreprises ne vont plus pouvoir s'en passer ? Retrieved from <https://www.soluti.fr/chatbot-customer-experience.html>
- VirtualSpirits. (2017). VirtualSpirits, créer un chatbot pour site web. Retrieved from <https://www.virtualspirits.com/fr/benefits.aspx>
- Wikipedia. (n.d.). No Title. Retrieved from [https://fr.wikipedia.org/wiki/R%C3%A9cit\\_utilisateur](https://fr.wikipedia.org/wiki/R%C3%A9cit_utilisateur)
- Wisdom, J. (2005). *Agents intelligents de l'internet: enjeux économiques et sociaux*. École Nationale Supérieure des Télécommunications Spécialité.
- Wooldridge, M., & Jennings, N. R. (1995). Intelligent agents: Theory and practice. *The Knowledge Engineering Review*, 10:2, 115–152.
- Zapier. (2016). HOW TO INTEGRATE CHATFUEL FOR TELEGRAM WITH SPEECHTRANS. Retrieved from <https://zapier.com/zapbook/chatfuel-for-telegram/speechtrans/>
- Zheng, J. (2016). How To Build a Recipe Slack Bot Using Watson Conversation and Spoonacular API. Retrieved from <https://medium.com/ibm-watson-developer-cloud/how-to-build-a-recipe-slack-bot-using-watson-conversation-and-spoonacular-api-487eacaf01d4#.lexwq0yf1>
- Zientara, D. (2016). The Bot Framework. Retrieved from <https://blog.appliedis.com/2016/08/16/the-bot-framework/>



## Annexes

**HEG Arc - Haute école Arc - Gestion**

**Mention :** Travail de Bachelor d'informaticien de gestion

**Module :** 656-1.1 Thèse de bachelor

**Sujet :** Exploration dans le domaine des agents intelligents et des chatbots

**Directeur de travail :** Frischer Boris

**Auteur :** Klett Boris

**Cycle d'étude :** 2013 - 2017

**Date de restitution :** Septembre 2017



# 19 Annexes

---

## 19.1 Annexe 1 : Outils de développement (*Smooch, Pandorabot*)

### 19.1.1 Smooch

Suivant la page web de Sabeer Shaikh, « Smooch est un autre outil très utilisé pour développer et intégrer des chatbots dans des applications de messagerie. Il permet de développer des chatbots qui fournissent une excellente expérience commerciale, en acceptant les paiements et en exécutant des commandes facilement. Les chatbot développés avec cet outil peuvent être déployé sur les plates-formes telles que Twilio, WeChat, Shopify et Télégramme. Smooch s'intègre facilement dans Facebook Messenger et utilise une grande partie de ses caractéristiques du géant américain comme les images, gifs, émojis et les boutons. Cet outil permet également de déployer les chatbots sur d'autres plateformes qui sont plus orientées business, comme : le Front, Slack, Hipchat et Zendesk » ([Shaikh, 2016](#)).

### 19.1.2 Pandorabot

Pour Shaikh, « *Pandorabot est une suite d'outils et des services disponibles sur le web qui permettent aux développeurs de construire et héberger des chatbots. Cette suite dispose d'un IDE pour développer des chatbots. Il permet de créer des chatbots pour la clientèle, la publicité, des jeux et bien plus encore. Actuellement, il y'a environ 225000 développeurs qui utilisent cette suite et plus de 285'000 chatbots ont été développés avec cette suite* » ([Shaikh, 2016](#)). La documentation Pandorabot définit l'outil comme étant un service web pour construire et déployer des chatbots ([Pandorabots, 2016](#)). Pour nous permettre de choisir l'outil le plus adapté à nos besoins, Il est essentiel de faire une comparaison des outils qui sont disponibles sur le marché.

## 19.2 Annexe 2 : Architecture pour la mise en œuvre d'un chatbot construit avec l'outil Microsoft bot framework

Le schéma ci-dessous permet de voir l'architecture de Microsoft bot framework et comprendre son fonctionnement.

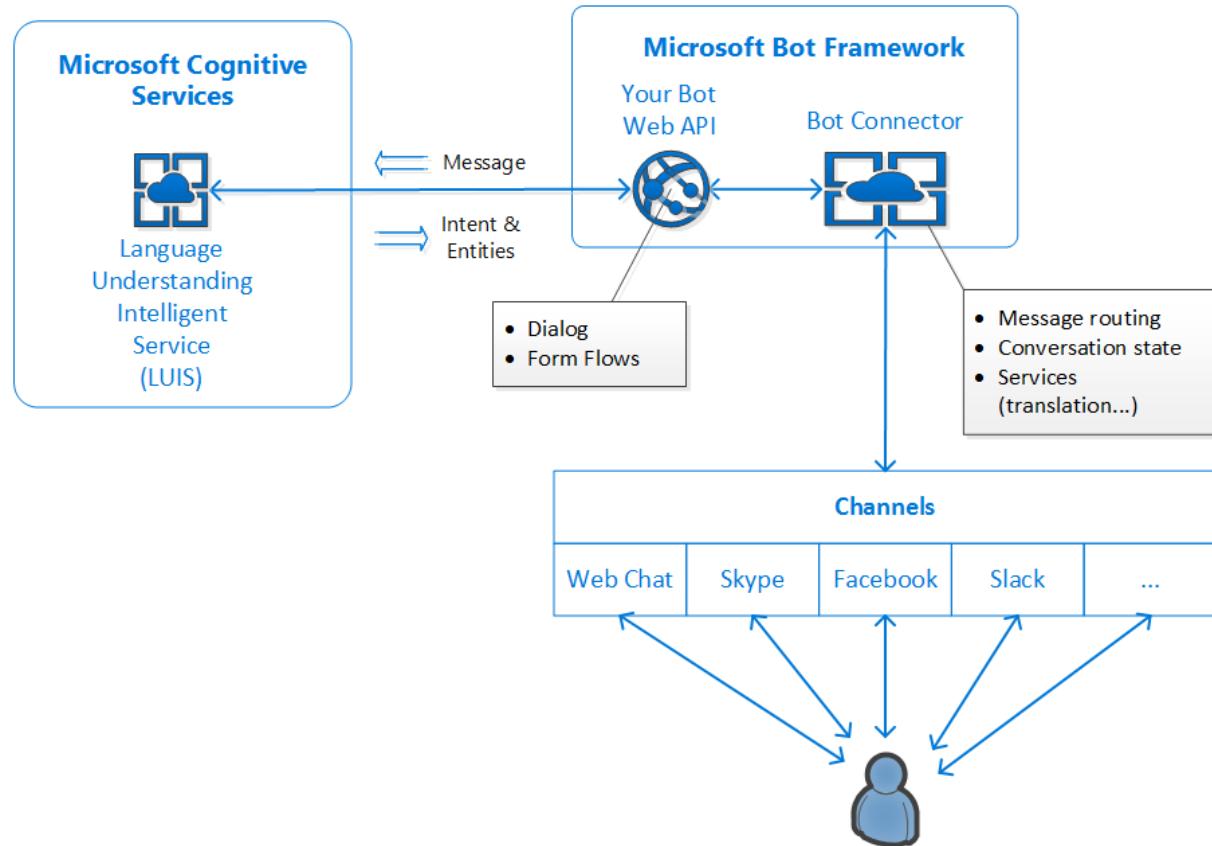


Figure 47 : Architecture du chatbot de Microsoft ([Zientara, 2016](#)).

Pour illustrer ce schéma, le bot connector fourni les services qui permettent aux utilisateurs de communiquer avec le chatbot via les messageries ou d'autres plateformes, ce qui permet au développeur de se concentrer uniquement sur le développement de la logique du chatbot. D'après David Zientara, le chatbot est implémenté comme une API web standard. On peut l'implémenter avec n'importe quelle technologie, mais Microsoft fournit deux SDKs en C# et Node.js qui permettent de développer un chatbot plus facilement. Ces SDKs facilitent également l'intégration des modèles LUIS, qui est un service qui aide le chatbot à parser et comprendre les messages des utilisateurs ([Zientara, 2016](#)).

### 19.3 Annexe 3 : SDK et technologies

La matrice ci-dessous liste les différents SDK et technologies utilisables ou proposés par les outils.

Tableau 1 : Matrice des SDK et technologies

	IBM Watson Conversation	API.AI (Google)	Chatfuel	Microsoft Bot Framework
Android	✓	✓		
Apple IOS	✓			
Java	✓	✓		
Node.js	✓	✓		✓
Python	✓	✓		
Unity	✓	✓		
Go				
Webhooks		✓		
Watch Os Mac OS X		✓		
Ruby		✓		
Webkit HTML5		✓		
JavaScript		✓		
Cordova		✓		
C#		✓		
C++		✓		
Xamarin		✓		
PHP		✓		
Json			✓	
.net				✓

## 11.4, Annexe 4 : Plateformes de déploiement ou d'intégration

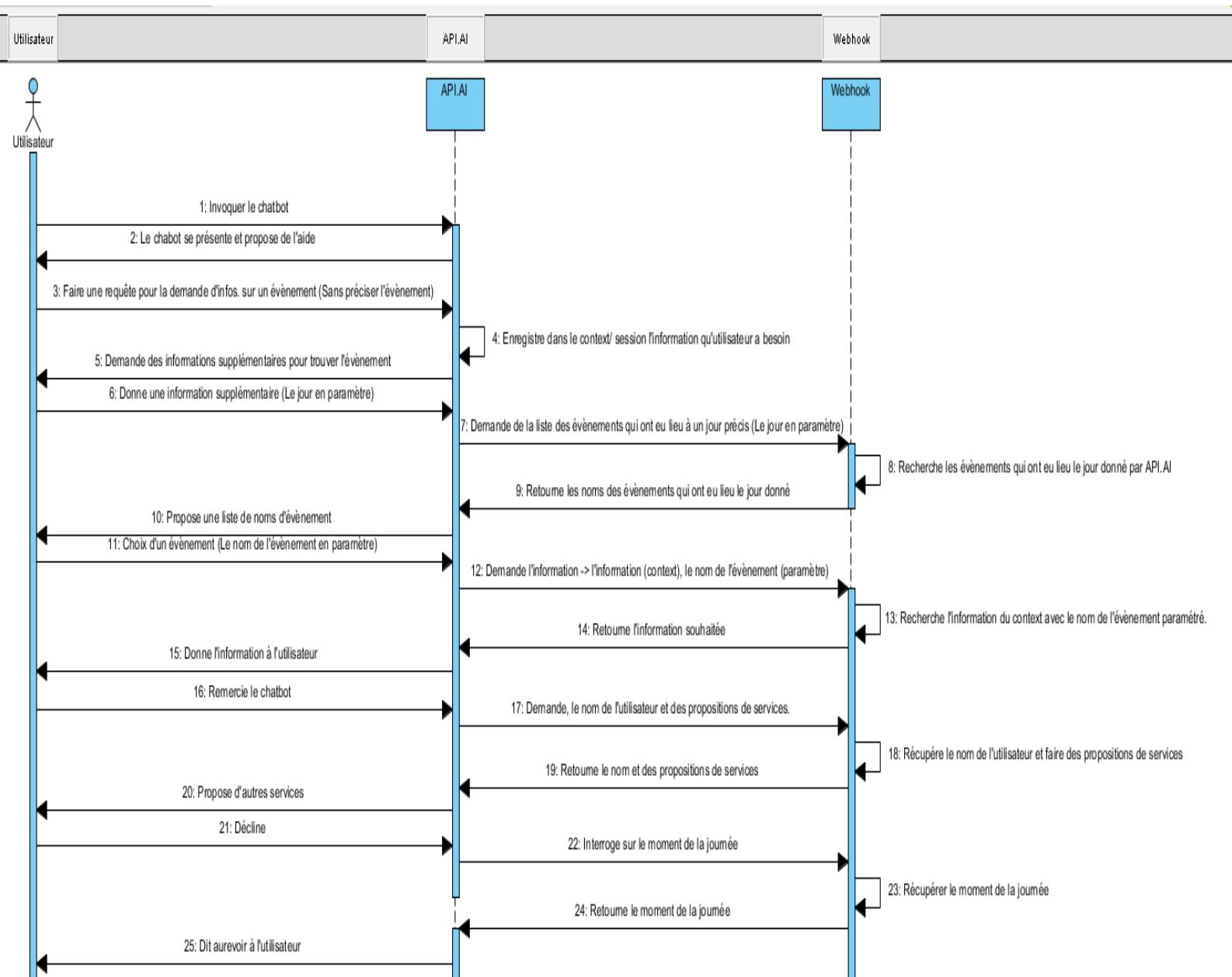
La matrice ci-dessous permet de voir sur quelles plateformes il est possible de déployer ou d'intégrer un Chatbot suivant chaque outil étudié dans ce rapport.

*Tableau 2 : Matrice des plateformes de déploiement des chatbots*

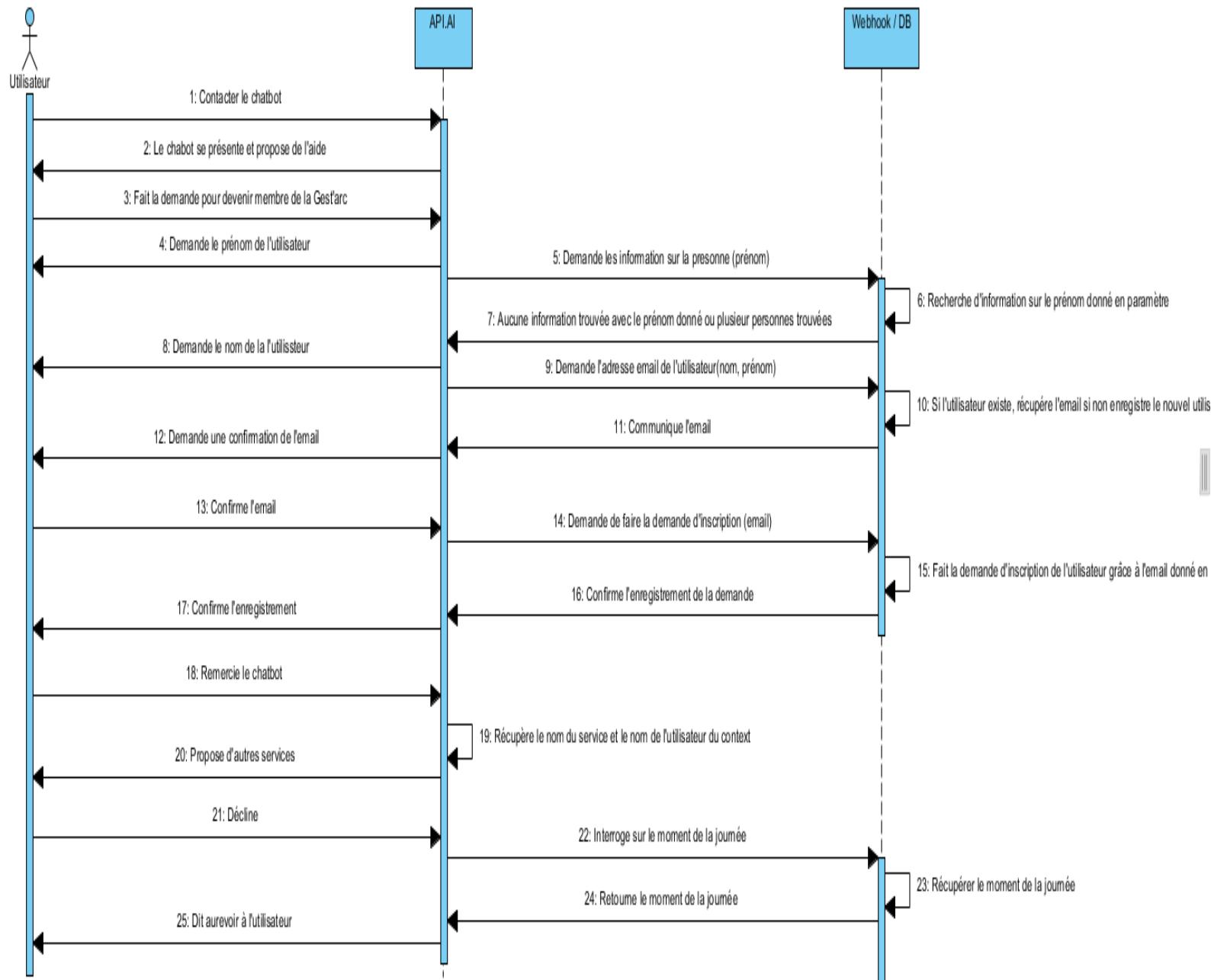
	IBM Watson Conversation	API.AI (Google)	Chatfuel	Microsoft Bot Framework
Facebook Messenger	✓	✓	✓	✓
Instagram			✓	
Télégramme		✓	✓	
Slack	✓	✓		✓
Kik		✓		
Twitter		✓	✓	
SMS				✓
Email				✓
Twilio	✓			
Twilio SMS		✓		✓
Twilio IP		✓		
Skype		✓		✓
Cisco Spark		✓		
Cisco Tropo		✓		
Line		✓		✓
Amazon Alexa		✓		
Microsoft Cortana		✓		
RSS			✓	
VK			✓	
Youtube			✓	
WordPress			✓	
Dropbox			✓	
Evernote			✓	
Gmail			✓	

## 12.5 Annexe 5 : Diagrammes

### 12.5.1 Diagramme de séquence pour la demande d'information sur un évènement

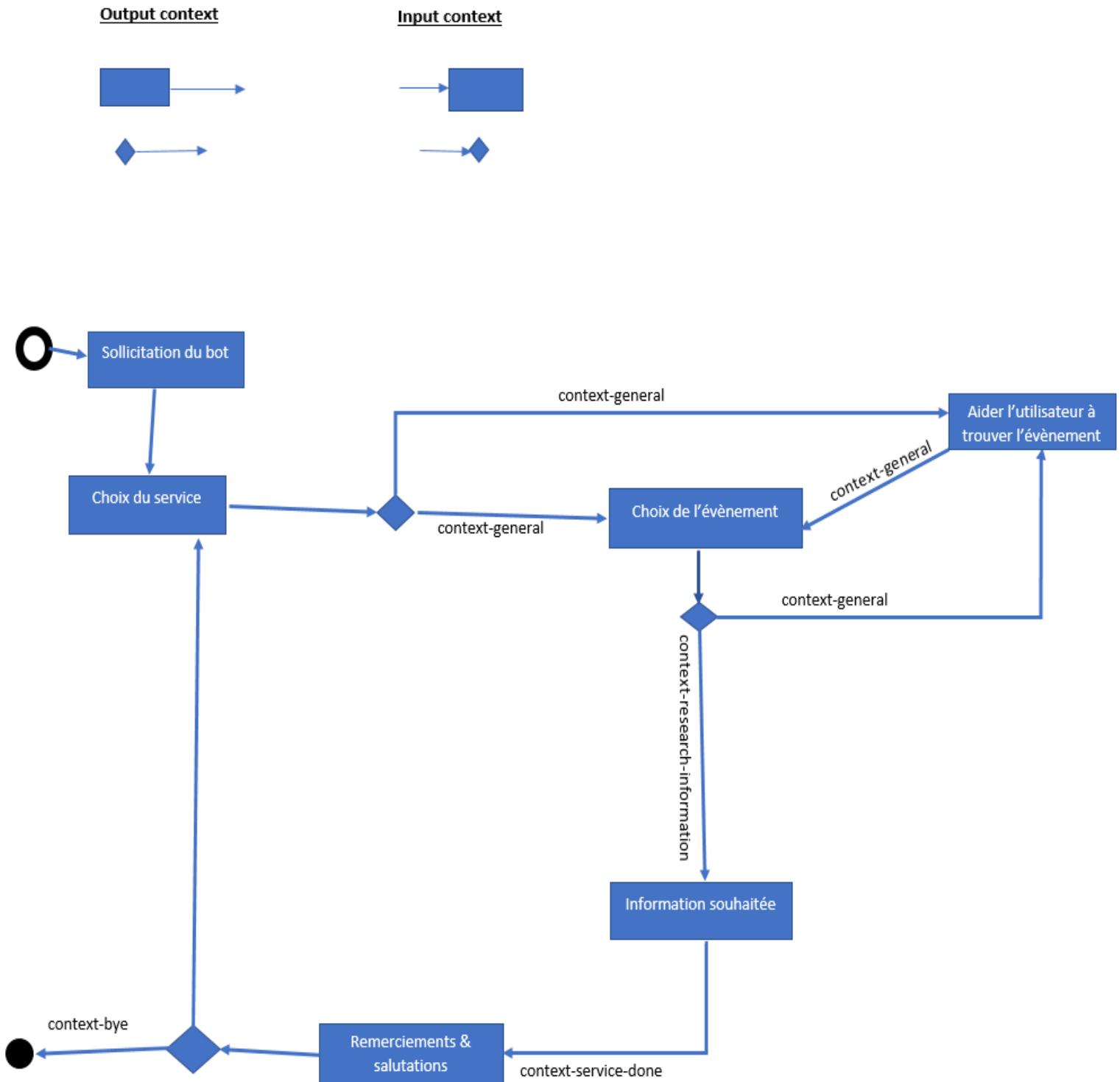


## 12.5.2 Diagramme de séquence pour la demande d'inscription comme membre

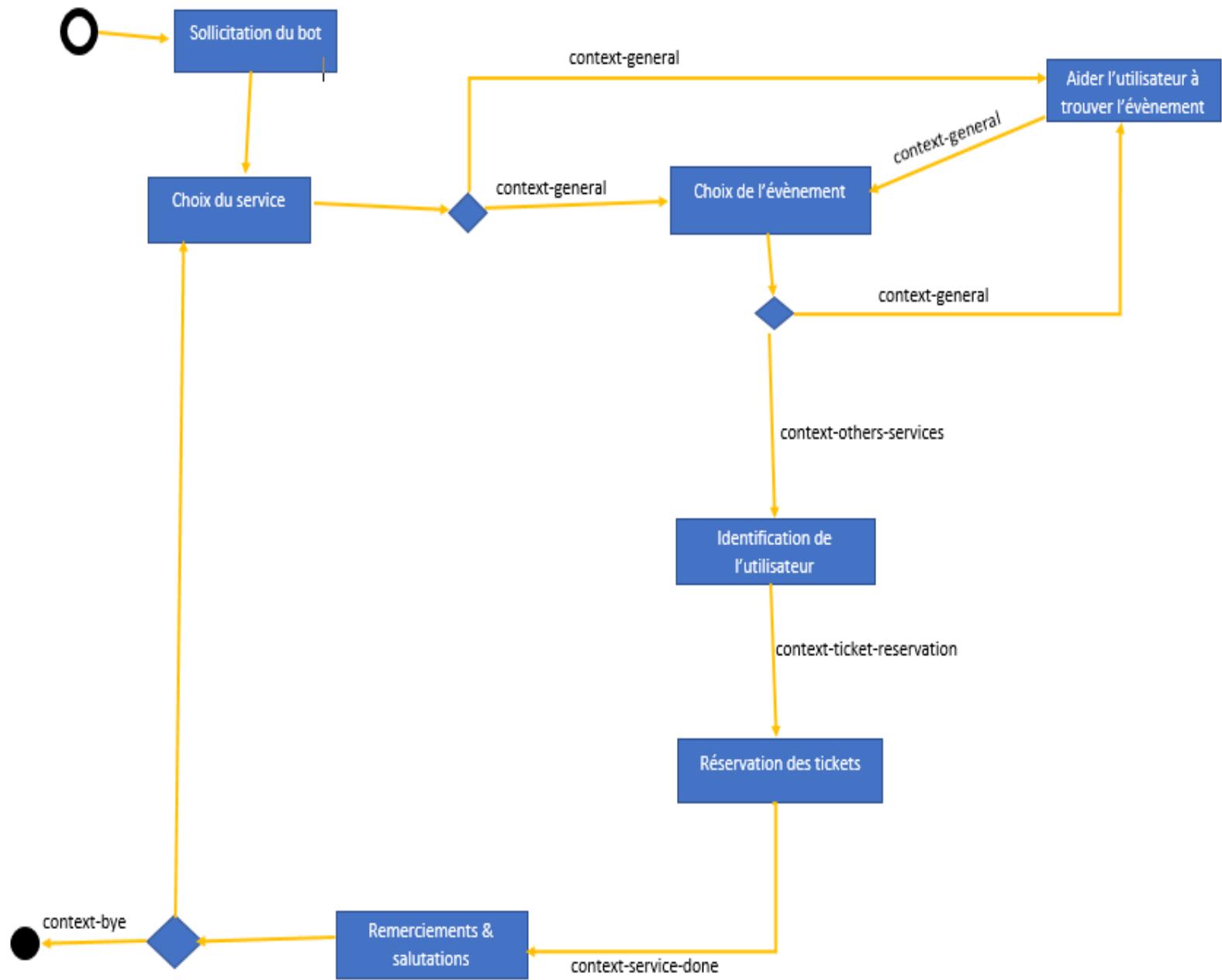


## 12.6 Annexe 6 : Contexts du chatbot (Gestarcbot)

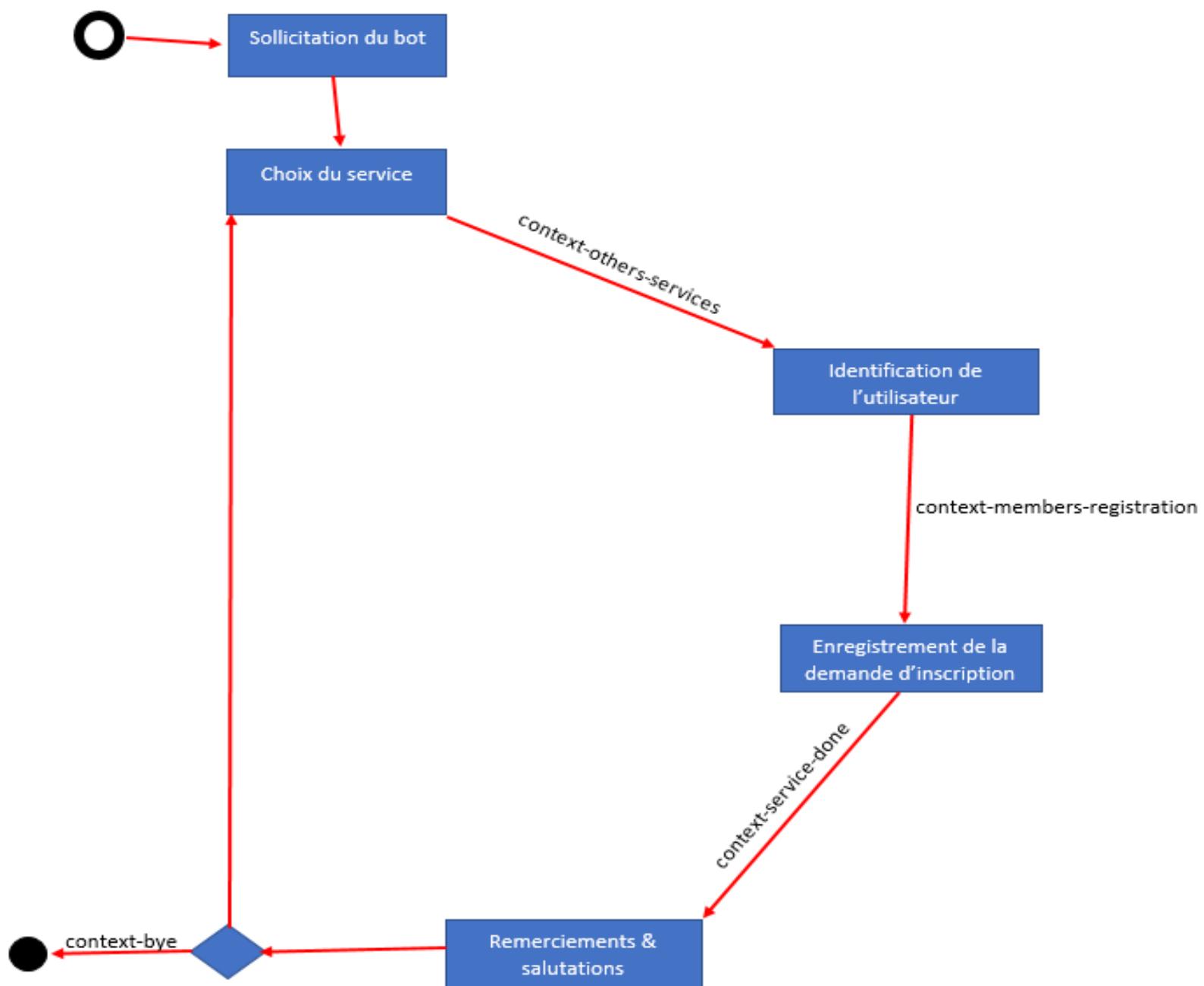
### 12.6.1 Contexts du processus « acquérir une information sur un évènement »



## 12.6.2 Contexts du processus « Réserver un ticket pour un évènement »



### 12.6.3 Contexts du processus « Enregistrer une demande d'inscription pour devenir membre de la Gest'arc »



## 12.7 Annexe 7 : Création et paramétrage des intents

### 12.7.1 Valeurs de paramétrage de l'agent, tableau 1

Intents	Chatbot-sollicitation	Member-registration-serviceChoice	Ticket_reservation_serviceChoice	User_need_event_info_serviceChoice
Input Context	--	--	--	--
Output Contexts	--	context-others-services	context-general	context-general
User says	<p>Hey</p> <p>Petite question</p> <p>Hello</p> <p>Coucou</p> <p>Salut</p> <p>Bonjour</p>	<p>c'est possible de m'inscrire pour devenir membre de la Gest'arc?</p> <p>J'aimerais bosser avec la Gest'arc.</p> <p>J'aimerais devenir un de vos membres</p> <p>J'aimerais adhérer à votre association</p> <p>J'aimerais devenir membre de la Gest'arc</p> <p>Oui j'aimerais savoir comment faire pour devenir membre de la Gest'arc ?</p>	<p>Est ce que c'est possible de réserver des tickets pour une soirée?</p> <p>J'aimerais réserver des places pour un évènement.</p> <p>Oui, j'aimerais faire des réservations.</p> <p>Réserver des ticket s'il te plaît.</p> <p>J'aimerais réserver des tickets.</p>	<p>Juste une info</p> <p>Un renseignement s'il vous plaît</p> <p>Oui, j'aimerais avoir une information sur un évènement que vous avez organisé</p> <p>J'aurais besoin d'un renseignement</p> <p>Est-ce que vous pouvez me renseigner sur un évènement</p> <p>--</p>
Action	--	--	--	
Réponses éventuelles du chatbot	<p>Voir la réponse 1</p> <p>Voir la réponse 2</p>	<p>Ok pas de problème. Alors si ça t'intéresse, j'aurai juste besoin de ton nom s'il te plaît</p> <p>Ok. Il me faudrait juste ton nom.</p> <p>Pas de soucis! du coup, c'est quoi ton nom?</p> <p>est ton nom de famille?</p> <p>Ok. Tu pourrais juste me dire ton nom? s'il te plaît.</p>	<p>Oui bien sûr, pour quel événement? est-ce que tu peux me donner le nom de l'événement s'il te plaît ?</p> <p>Pas de soucis, mais est-ce que tu pourrais juste me donner le nom de l'événement s'il te plaît ?</p> <p>Ça marche, donne-moi juste le nom de l'événement s'il te plaît.</p> <p>Ok, tu sais le nom de l'événement ?</p>	<p>Voir les réponses de Gestarobot dans l'intent: Ticket_reservation_serviceChoice</p>

Les réponses du chatbot qui manquent dans le tableau 1 ci-dessus sont :

- Réponse 1 : Salut, est ce que je peux t'aider ? Et si jamais, je suis Gestarcbot un chatbot mis en place sur Facebook Messenger, pour aider les gens sur les évènements organisés par la Gest'arc. Je suis donc capable de donner des renseignements et réserver des tickets pour des évènement, et également prendre une demande d'inscription pour devenir membre de la Gest'arc. □
- Réponse 2 : Hello, est ce que je peux t'aider ? Si jamais, je suis Gestarcbot un chatbot mis en place sur Facebook Messenger, pour répondre rapidement aux demandes des gens. Je suis donc capable de donner des renseignements et réserver des tickets pour des évènement, et également prendre une demande d'inscription pour devenir membre de la Gest'arc. □

## 12.7.2 Valeurs de paramétrage de l'agent, tableau 2

Agent : Gestarcbot				
Intents	Chatbot_eventName_known	Chatbot_eventName_unknown	User_need_event_info_get_info	Others-services-user-lastname
Input Context	context-general	context-general	context-research-information	context-others-services
Output Contexts	context-others-services	context-general	context-service-done	context-members-registration
	context-research-information			context-need-firstname
	--			context-ticket-reservation
User says	(Nom d'un évènement)	Non.	Quel est l'horaire ?	(Nom de l'utilisateur) c'est mon nom
	Oui bien sûr, c'est (Nom d'un évènement)	Non je ne sais plus.	Quel est le prix des tickets ?	Je suis (Nom de l'utilisateur)
	Oui ! (Nom d'un évènement)	Aucune idée.	Ça coûte combien ?	Mon nom est (Nom de l'utilisateur)
	(Nom d'un évènement) je crois.	Je ne sais plus.	Je voulais savoir déjà le thème de la soirée.	Je m'appelle (Nom de l'utilisateur)
	Je crois que c'était (Nom d'un évènement).	--	J'aimerais savoir où est ce que je peux trouver les photos de l'évènement.	(Nom de l'utilisateur)
	Je crois que c'était (Nom d'un évènement)	--	J'aimerais trouver les photos	--
Action	check-event-name	get-events-to-help	get-needed-information	get-user-by-lastname
Réponses éventuelles du chatbot	Voir la réponse 3	Voir la réponse 9	Ok ça marche. Bah tiens le lien Facebook t'y trouveras toutes les photos que nous avons faites (Lien Facebook)	Voir la réponse 9
	Voir la réponse 4		Le thème sera (Thème de l'évènement)	Voir la réponse 10
	Voir la réponse 5		Les tarifs sont-> (les prix). Tu aimerais que je t'en réserve ?	Voir la réponse 11
	Voir la réponse 6		--	Voir la réponse 12
	Voir la réponse 7		--	Voir la réponse 13
	Voir la réponse 8		--	Voir la réponse 14

Les réponses du chatbot qui manquent dans le tableau 2 ci-dessus sont :

- Dans le cas où souhaiterait acquérir une information sur un évènement
  - ❖ Réponse 3 : Ha ok je vois ! Elle était vraiment sympa. Qu'aimerais-tu savoir du coup ?
  - ❖ Réponse 4 : Je pense que ça sera vraiment bien cette (Type d'évènement). Qu'est-ce que tu aimerais ?
  - ❖ Réponse 5 : Qu'est-ce que tu aimerais savoir sur cette (Type d'évènement) ?
  - ❖ Réponse 6 : Ok alors dis-moi qu'est-ce que t'aimerais ?
- Dans le cas où l'utilisateur souhaiterait réserver un ticket pour un évènement
  - ❖ Réponse 7 : Ok. J'aurais juste besoin de ton nom s'il te plaît.
  - ❖ Réponse 8 : et c'est comment ton nom ? s'il te plaît.
- Réponse 9 : Ce n'est pas grave. Tiens voici tous les évènements du moment « (Le nom de l'évènement) à (Le lieu de l'évènement) organisée le (Date de l'évènement) ». Voilà tu peux choisir l'évènement qui t'intéresse.
- Dans le cas où le chatbot arrive à identifier l'utilisateur uniquement avec son nom
  - ❖ Réponse 9.2 : Ok, alors ton adresse email est bien (Email de l'utilisateur) ?
  - ❖ Réponse 10 : Ça marche. Juste ton email est bien (Email de l'utilisateur) ?
  - ❖ Réponse 11 : (Email de l'utilisateur) est bien ton email ?
- Dans le cas où le chatbot a besoin du prénom de l'utilisateur pour pouvoir l'identifier
  - ❖ Réponse 12 : Et c'est comment ton prénom ?
  - ❖ Réponse 13 : Ton prénom c'est quoi ?
  - ❖ Réponse 14 : J'ai encore besoin de ton prénom s'il te plaît

### 12.7.3 Valeurs de paramétrage de l'agent, tableau 3

Agent : Gestarcbot				
Intents	Others-services-user-firstname	Member-registration	Ticket-number-of-tickets	Ticket-reservation
Input Context	context-needfirstname	context-members-registration	context-ticket-reservation	context-ticket-reservation-reservation
Output Contexts	context-ticket-reservation	context-service-done	context-ticket-reservation-reservation	context-service-done
	context-members-registration			
	--			
User says	(Prénom de l'utilisateur ) c'est mon prénom	Tout à fait	Sont les même Users say de l'intent "Member-registration"	J'en prend un s'il te plaît
	C'est (Prénom de l'utilisateur)	Tout à fait		J'en veux deux
	Mon prénom est (Prénom de l'utilisateur)	C'est bien mon email		Seulement un
	Oui, bien sûr c'est (Prénom de l'utilisateur)	Oui c'est bien mon email.		J'aimerais 5 s'il te plaît
	(Prénom de l'utilisateur)	Oui		Juste 1 ticket s'il te plaît
	--	--		Je veux 2 ticket s'il te plaît
Action	get-user-by-last-and-firstname	registration-member	user-confirm-email	tickets-reservation
Réponses éventuelles du chatbot	Ok, alors ton adresse email est bien (email de l'utilisateur) ?	Voir la réponse 15	Ok et tu veux combien de ticket (Prénom de l'utilisateur) ?	Voir la réponse 19
	Ça marche. Juste ton email est bien (email de l'utilisateur) ?	Voir la réponse 16	Tu aimerais combien de ticket ?	Voir la réponse 20
	(Email de l'utilisateur) est bien ton email ?	Voir la réponse 17	Et tu veux combien de ticket (Prénom de l'utilisateur) ?	Voir la réponse 21
	--	Voir la réponse 18	--	Voir la réponse 22
	--	--	--	--
--	--	--	--	--

Les réponses du chatbot qui manquent dans le tableau 3 ci-dessus sont :

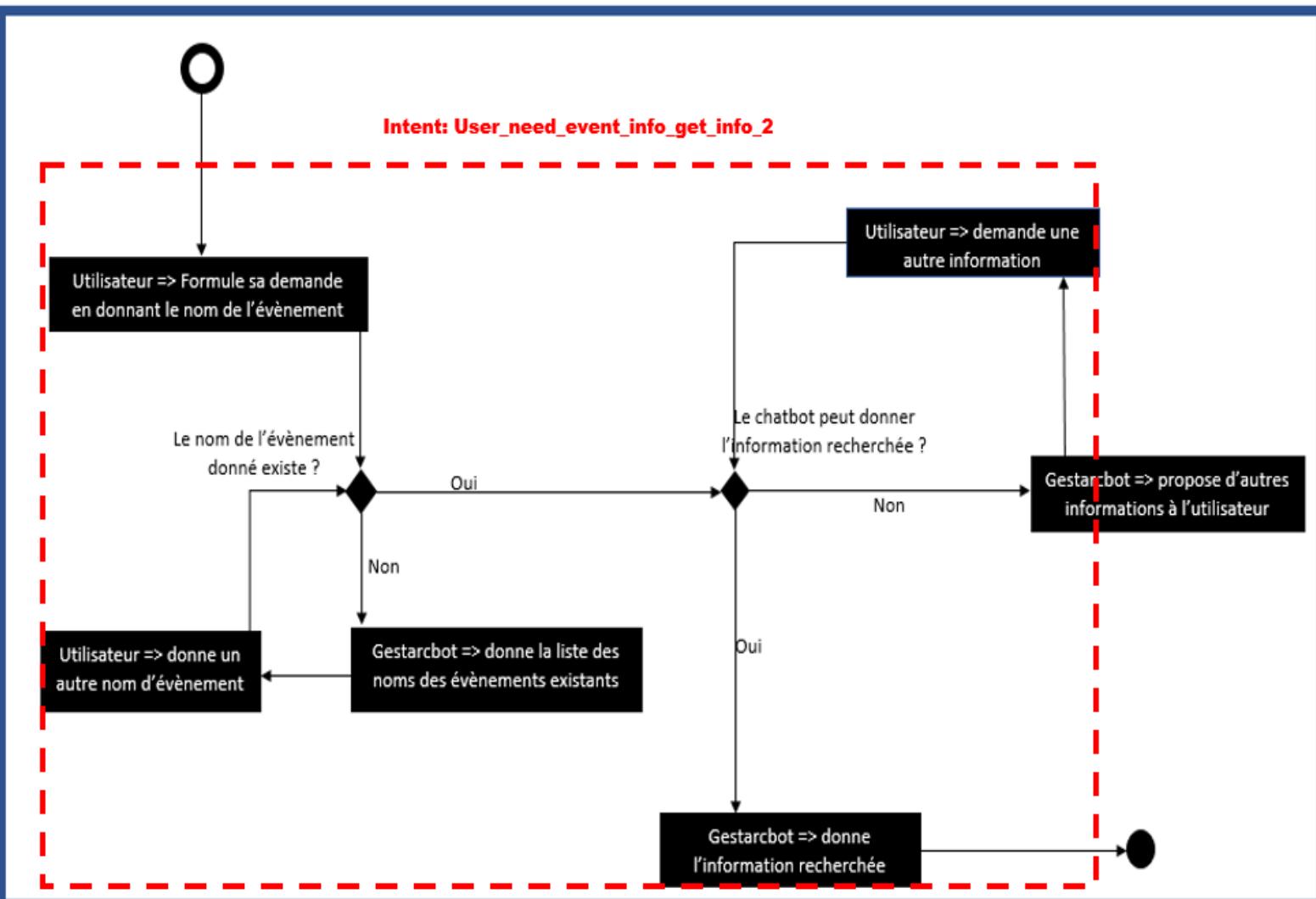
- Réponse 15 : Ok (Prénom de l'utilisateur) j'ai enregistré ta demande pour devenir membre de la Gest'arc. Si jamais, un membre de la Gest'arc te confirmera ton adhésion et te convoquera par email, dans les prochains jours.
- Réponse 16 : Ça marche. Écoute j'ai enregistré ta demande (Prénom de l'utilisateur), par contre un membre de la Gest'arc te confirmera ton adhésion et te convoquera par email, dans les prochains jours :D .
- Dans le cas où l'utilisateur avait déjà fait une demande d'inscription
  - ❖ Réponse 17 : Ok ! Ha mais (Prénom de l'utilisateur) j'avais déjà enregistré ta demande le (Date de l'enregistrement) dernier. Par contre c'est vrai qu'elle n'a pas encore été traitée. Avec tout ce qu'il y a à faire ces jours, je pense qu'ils sont tous overbookés. Je te dirais bien d'attendre encore un peu mais au pire essaye de passer directement au secrétariat de l'école pour le faire si tu veux (y).
  - ❖ Réponse 18 : Ok ! Ha mais (Prénom de l'utilisateur) j'avais déjà enregistré ta demande le (Date de l'enregistrement) dernier et elle a déjà été traitée. Ecoute, contrôle encore ta boîte email de l'école tu devrais avoir reçu une confirmation.
- Réponse 19 : Ok ça joue ! Juste encore une petite précision je te réserve des tickets mais par contre tu devras aller payer auprès de la Gest'arc dans le bâtiment de la HEG ou alors au secrétariat de l'école avant vendredi à 18h si non les tickets seront remis en vente.
- Réponse 20 : Alors ça te fera 45 francs si jamais ! Juste encore une petite précision je te réserve des tickets mais par contre tu devras aller payer auprès de la Gest'arc dans le bâtiment de la HEG ou alors au secrétariat de l'école avant vendredi à 18h si non les tickets seront remis en vente.
- Dans le cas où il n'y a plus assez de tickets pour faire la réservation
  - ❖ Réponse 21 : Ha mince il ne reste plus que 3 tickets réservable. Je te les réserve tout de suite (Prénom de l'utilisateur). Et si jamais tu tiens toujours à avoir tous les tickets, tu peux aussi passer au secrétariat pour te renseigner s'il y en a encore et y acheter le reste d'accord ? par contre ça te fera 45 francs que tu devras aller payer auprès de la Gest'arc dans le bâtiment de la HEG ou alors au secrétariat de l'école avant vendredi à 18h si non les tickets seront remis en vente.
  - ❖ Réponse 22 : Ha mince il ne reste plus de tickets réservables (Prénom de l'utilisateur). Il faudrait que tu passes au secrétariat pour te renseigner s'il y en a encore et y en prendre directement si ça jour pour toi ! d'accord ?

#### 12.7.4 Valeurs de paramétrage de l'agent, tableau 4

Agent : Gestarchbot				
Intents	Greetings_with_thankful	Greetings_with_not_thankful	GBye	Finish
Input Context	context-service-done	context-service-done	context-service-done	context-bye
Output Contexts	context-service-done	context-service-done	context-bye	context-bye
User says	Merci beaucoup Cool merci Merci Ok ça marche merci Ok merci --	Ok, ça joue. Ça joue Ça marche Ok -- --	Pas besoin merci. Non merci! Non -- -- --	Merci à toi aussi Merci à toi aussi -- -- --
Action	user-thank	user-not-thank	salutations	--
Réponses éventuelles du chatbot	Pas de quoi(y). Est-ce que tu aimerais quelque chose d'autre ? Avec plaisir. Est-ce que tu as besoin de quelque chose d'autre du coup ? -- -- -- --	Tu souhaites quelque chose d'autre ? Tu aimerais quelque chose d'autre ? -- -- -- --	Ok ça marche. Alors une bonne soirée et à bientôt. Ok ça marche. Alors une bonne soirée et surtout n'hésite pas à écrire si tu as besoin de renseignement. Oui merci une bonne nuit à toi aussi -- -- -- --	:D (y) -- -- -- --

## 12.8 Annexe 8 : Nouvelles étapes de la logique de Gestarcbot

### 12.8.1 Demande d'information



### 12.8.2 Demande de réservation de ticket pour un évènement

