République du Cameroun
***
Paix-Travail-Patrie
***
Ministère de l'Enseignement Supérieur
***
Université de Maroua
***
Ecole Nationale Supérieure Polytechnique de Maroua

Republic of Cameroon
***
Peace-Work-Fatherland
***
Ministry of Higher Education
***
The University of Maroua
***
National Higher Polytechnic School of Maroua

INFORMATIQUE ET TELECOMMUNICATIONS

CONCEPTION ET MISE EN PLACE D'UN CHATBOT SECURISE D'ENTREPRISE : CAS DE WORLD VOICE GROUP

Mémoire présenté et soutenu en vue de l'obtention du Diplôme

D'INGENIEUR DE CONCEPTION EN

CRYPTOGRAPHIE ET SECURITE INFORMATIQUE

par

KENMENE ARCELE CRESSON

Licence en Mathématique-informatique

Matricule: 17Y244P

sous la direction de

Dr. BOUDJOU Hortense (encadreur académique) et

Dr. MELATAGIA Paulin (encadreur professionnel)

Devant le jury composé de :

Président :

Rapporteur :

Examinateur :

Invité :

ANNÉE ACADÉMIQUE: **2018/2019**

DEDICACE A

Mon père FOUTSOP David et ma mère FOUAMENE Jeannette

RESUME

L'entreprise WORLD VOICE GROUP SARL est une société de prestation de services informatique, et qui fait dans la distribution digital. Sa satisfaction et celle de ses clients est l'objet de sa quête. Pour celà elle dispose d'un support client qui fonctionne actuellement à travers plusieurs plate-formes : E-mail, Appel téléphonique et Whatsapp. Cette methode implique l'excès d'utilisation des ressources humaines de l'entreprise et un engorgement dans le traitement des requetes des clients. Pour palier à ça, WORLD VOICE GROUP propose de mettre sur pied un outil permettant d'automatiser une bonne partie des taches du support client, afin d'y reduire l'implication de l'être humain.

De nos jours, bon nombre d'internautes utilisent au quotidien une messagerie instantanée. Les sites web et les reseaux sociaux (Facebook par exemple) constituent les premiers canaux de contacts des clients. Pour être proche de ceux-ci et de l'intelligence artificielle, WORLD VOICE GROUP décide donc de mettre en place un chatbot d'entreprise.

Nous appelons Chatbot (agent conversationnel), une application pouvant dialoguer avec une personne via une solution de chat (Facebook Messenger, Skype, ...), et comprendre ses demandes en langage naturel, pour proposer des réponses ou lancer des actions. Dans notre contexte, le chatbot comprend la requête du client et lui repond, accéde aux systèmes d'information de l'entreprise via des API (Application Programming Interface) sécurisées pour puiser les données et répondre d'une façon fiable. En cas d'incapacité de répondre, il dirige la requête vers un conseiller client.

Pour sécuriser le lien (API) entre le chatbot et le système d'information de l'entreprise, nous utilisons le protocole d'authentification et d'autorisation OAuth2 (Open Authorization 2), et le modèle de controle d'accès RBAC (Role Based Access Control).

Mots clés : Chatbot, API, OAuth2, modèle de controle d'accès

ABSTRACT

The company WORLD VOICE GROUP SARL is a company providing IT services, and doing digital distribution. His satisfaction and that of his customers is the object of his quest. For this it has a customer support that currently works across multiple platforms: E-mail, Phone call and Whatsapp. This method involves the excessive use of the company's human resources and a bottleneck in the processing of customer requests. To overcome that, WORLD VOICE GROUP proposes to set up a tool allowing to automate a good part of the tasks of the customer support, in order to reduce the implication of the human being there.

Nowadays, many Internet users use instant messaging. Websites and social networks (Facebook for example) are the first channels of contact for customers. To be close to these and artificial intelligence, WORLD VOICE GROUP decides to set up a company chatbot.

We call Chatbot (conversational agent), an application that can interact with a person via a chat solution (Facebook Messenger, Skype, ...), and understand its requests in natural language, to propose answers or launch actions. In our context, the chatbot understands the customer's request and responds to it, accesses the company's information systems via Application Programming Interface (API) to retrieve data and respond reliably. If unable to answer, he directs the request to a client advisor.

To secure the link (API) between the chatbot and the enterprise information system, we use the authentication and authorization protocol OAuth2 (Open Authorization 2), and the access control model RBAC (Role Based Access Control).

Keywords : Chatbot, API, OAuth2, access control model

REMERCIEMENTS

Nous remercions d'abord Dieu le père tout puissant pour tous ses bienfaits. En suite, le présent travail n'aurait sans doute pas été réalisé sans le soutient de certaines personnes à qui je tiens à exprimer ma profonde gratitude et mes sincères remerciements. Nous tenons donc ainsi à remercier :

- Le président du jury ... pour avoir accepté de présider le jury ;
- L'examinateur ... pour avoir accepté d'examiner notre travail ;
- l'encadreur académique ... pour sa disponibilité, sa rigeur dans le travail et les remarques apportées à ce travail ;
- Notre Chef de département Dr. KALADZAVI GUIDEDI pour tous ses conseils, sa disponibilité et les efforts consentis à notre égard ;
- L'encadreur professionnel Dr. MELATAGIA PAULIN pour sa disponibilité, son encadrement et pour les remarques apportées à notre travail pendant notre stage ;
- Tous les personnels de WORLD VOICE GROUP avec lesquels nous avons collaboré durant notre stage ;
- Nos enseignants du Département d'informatique et télécommunications pour les enseignements dispensés et les conseils prodigués ; En particulier Dr.BOUDJOU HORTENSE et M. LAOUKOURA CHARLES ;
- Mon papa FOUTSOP DAVID et ma maman FOUAMENE JEANNETTE pour tout l'amour, le soutien et l'attention qu'ils nous portent ;
- Ma très chère soeur KENNE KELVINE pour son soutien et son attention ;
- Mes frères et soeurs pour leur soutien moral et affectif ;
- Mes amis pour le soutien moral, affectif et les conseils prodigués ;
- Tous mes camarades de promotion pôur les conseils et les moments passés ensemble ainsi que l'ambiance conviviale qui a toujours régné entre nous ;
- Tous ceux qui ont contribué de près ou de loin à la réalisation de ce travail.

Table des matières

RESUME i

ABSTRACT ii

REMERCIEMENTS iii

INTRODUCTION GENERALE 1

1 DESCRIPTION DU PROBLEME et OBJECTIFS 4
- 1.1 introduction . . . 4
- 1.2 Description de l'entreprise WORLD VOICE GROUP SARL . . . 4
  - 1.2.1 Présentation de WORLD VOICE GROUP . . . 4
  - 1.2.2 Organisation de WORLD VOICE GROUP . . . 5
  - 1.2.3 Organigramme de WORLD VOICE GROUP . . . 7
- 1.3 Description de l'existant . . . 8
  - 1.3.1 par mail . . . 8
  - 1.3.2 Par site web : formulaire de contact . . . 10
  - 1.3.3 Whatsapp business . . . 12
  - 1.3.4 CallCenter . . . 14
- 1.4 Limites de l'existant . . . 16
- 1.5 Solution proposée et objectifs . . . 16
- 1.6 Conclusion . . . 17

2 AGENT CONVERSATIONNEL (CHATBOT) 18
- 2.1 Introduction . . . 18
- 2.2 Intelligence artificielle . . . 18
  - 2.2.1 Machine learning . . . 19
  - 2.2.2 chatbot . . . 19
- 2.3 Importance d'un chatbot pour l'entreprise . . . 19
- 2.4 Modèle d'architecture pour un chatbot . . . 20
- 2.5 Technologie de mise en oeuvre d'un chatbot . . . 22
- 2.6 Conclusion . . . 22

3 SECURISATION DES API **23**
3.1 Introduction . . . 23
3.2 C'est quoi une API REST . . . 23
3.2.1 API . . . 23
3.2.2 API REST . . . 24
3.3 Sécurité informatique . . . 24
3.4 Protocoles d'authentification et d'autorisation . . . 26
3.4.1 Le protocole Open Authorization (OAuth2) . . . 26
3.4.2 Protocole OpenID Connect . . . 28
3.4.3 Le protocole SAML . . . 32
3.4.4 Un focus sur JSON Web Token (JWT) . . . 32
3.4.5 choix du protocole d'authentification et d'autorisation . . . 34
3.5 Modèle de contrôle d'accès . . . 35
3.5.1 Contrôle d'accès . . . 35
3.5.2 Discretionary Access Control (DAC) . . . 36
3.5.3 Mandatory access control (MAC) . . . 37
3.5.4 Role based access control (RBAC) . . . 37
3.5.5 Organization Based Access Control (OrBAC) . . . 41
3.5.6 Attributes Based Access control (ABAC) . . . 43
3.6 Conclusion . . . 44

4 ANALYSE, CONCEPTION ET REALISATION **45**
4.1 Introduction . . . 45
4.2 Choix du cycle de vie de developpement logiciel . . . 45
4.3 Orientation et faisabilité . . . 47
4.4 Analyse des besoins . . . 48
4.4.1 besoins fonctionnels . . . 48
4.4.2 Besoins non fonctionnels . . . 49
4.5 Budgetisation . . . 49
4.6 Conception architecturale . . . 50
4.7 Conception detaillée . . . 50
4.7.1 présentation du langage UML . . . 50
4.7.2 Modélisation avec le langage UML . . . 51
4.8 Implémentation . . . 55
4.8.1 Environnement de développement . . . 55
4.8.2 développement du bot . . . 56
4.8.3 développement de l'API sécurisée . . . 56
4.9 Résultat . . . 57
4.10 Conclusion . . . 57

CONCLUSION GENERALE ET PERSPECTIVES **57**

BIBLIOGRAPHIE **60**

ANNEXES **60**

Table des figures

1.1 Organigramme de WORLD VOICE GROUP SARL 7
1.2 Organigramme du fonctionnement actuel pour Email 9
1.3 Organigramme du fonctionnement actuel pour Formulaire de contact . . . 11
1.4 Organigramme du fonctionnement actuel pour Whatsapp business 13
1.5 Organigramme du fonctionnement actuel pour CallCenter 15

2.1 Modèle d'architecture pour un chatbot [5] 20

3.1 Fonctionnement d'une API REST [14] 24
3.2 Séquencement OAuth2 [17] 27
3.3 Algorithme Authorization code flow [17] 31
3.4 un exemple de payload 33
3.5 un exemple JWT 33
3.6 un exemple JWS [7] 34
3.7 Role-based access control [10] 38
3.8 Le noyau RBAC [16] 39
3.9 Exemple de hiérarchie de role 39
3.10 Schéma présentant les interactions entre les entités du modèle OrBAC [9] . 41
3.11 MetaModel du ABAC [11] 43

4.1 Modèle de cycle de vie en V [1] 47
4.2 Architecture du système 50
4.3 Diagramme de séquence pour "alerter un conseiller" 52
4.4 Diagramme de séquence pour "acceder au SI" 53
4.5 Diagramme de séquence général pour "conversation entre le client et le chatbot" 54
4.6 Diagramme de déploiement 55

INTRODUCTION GENERALE

« On peut apprendre à un ordinateur à dire : "Je t'aime", mais on ne peut pas lui apprendre à aimer. »

– De Albert Jacquard

Le mot chatbot est combiné des termes chat (qui signifie Discuter) et bot (qui signifie Robot). Il est un programme informatique qui simule une conversation avec une personne alors qu'il s'agit d'intelligence artificielle. Auparavant, les applications de messagerie ne servaient qu'à converser entre amis. Mais avec la montée de l'intelligence artificielle, les entreprises peuvent désormais converser avec leurs clients, et ce, de façon automatique grâce aux bots. Les chatbots sont montés en puissance.

Aujourd'hui une entreprise peut faire d'un chatbot un support client, un service client, marketing, un outil pour les commandes et achats en ligne, etc. Les clients accèdent à ceci à travers les sites web et les réseaux sociaux tels que Facebook Messenger, Skype, Telegram, Slack, ... Nous ne pouvons pas ignorer que presque tous les clients/prospects des entreprises passent le plus souvent leurs temps sur des réseaux sociaux. En 2018, Facebook comptait 2,8 millions d'utilisateurs camerounais. Sachant qu'en 2016, il était à 1,4 million d'utilisateurs camerounais [**4**], on constate donc une progression de 100% en deux ans, ce qui est énorme vu la base déjà conséquente d'utilisateurs en 2016.

Un support client est un lien entre l'entreprise et le client. C'est à travers ce support que les clients vont pouvoir communiquer directement avec l'entreprise, leur remonter des informations, exprimer leur ressenti, ... Ceci en accompagnant le client tout au long de sa navigation, de sa commande, de sa livraison, et bien évidemment après la réception de son colis ou de l'utilisation du service. Générer de la donnée pour comprendre et anticiper les attentes et les besoins des visiteurs/clients est une mission du service client. Et lorsque c'est le cas, c'est une vraie mine d'or qui peut s'ouvrir à l'entreprise, si elle décide d'exploiter ce potentiel.

Pour faire fonctionner le support client, l'entreprise **WORLD VOICE GROUP SARL** est partie du modèle classique qui est l'Email, appel téléphonique ; et va maintenant devenir beaucoup plus intelligent en restant proche de ce que les gens font (les réseaux sociaux) et de l'intelligence artificielle. Elle a donc accordé une importance considérable au chatbot, afin de satisfaire ses clients et elle-même en moins de temps.

La sécurité est une préoccupation de plus en plus importante aujourd'hui, que ce soit au niveau personnel ou au niveau de l'entreprise. Nous définissons la sécurité comme l'ensemble des dispositions prises pour se protéger ou encore protéger ses biens contre d'éventuelles attaques. Dans les entreprises, la sécurité passe par la prise en compte des facteurs humains d'où un certain accent doit être mis sur l'homme car celui-ci peut divulguer volontairement ou pas des informations sensibles sur l'entreprise ou encore porter atteinte au patrimoine de l'entreprise d'une manière directe ou pas et c'est donc pour cette raison que la plupart des entreprises aujourd'hui pour protéger leur patrimoine restreignent l'accès à celui-ci en adoptant des systèmes de sécurité.

L'entreprise WORLD VOICE GROUP dispose des systèmes d'informations qui sont de temps en temps utilisés par le support client pour mieux satisfaire les clients. Et si le chatbot remplace ce support client, alors le robot de ce chatbot doit pouvoir accéder aux systèmes d'informations via des API ; pour cela ces API doivent être sécurisées, afin que le robot soit authentifié avant d'être autorisé à accéder à une quelconque information.

Pour mener à bien ce travail, nous l'avons subdivisé en plusieurs chapitres : le premier chapitre est consacré à la description du problème et aux objectifs de notre travail, le second chapitre au fontionnement d'un chatbot, le troisième chapitre à la sécurisation des API d'un système d'information et enfin le quatrième chapitre est dédié à la conception et réalisation de notre chatbot sécurisé d'entreprise.

Chapitre 1

DESCRIPTION DU PROBLEME et OBJECTIFS

1.1 introduction

Dans ce chapitre nous présenterons l'entreprise où nous avons effectué notre stage académique à savoir WORLD VOICE GROUP. Nous présenterons en suite la description de l'existant relatif à notre sujet de stage, nous terminerons par une mise en évidence des limites liés à ce contexte et les objectifs de notre sujet.

1.2 Description de l'entreprise WORLD VOICE GROUP SARL

1.2.1 Présentation de WORLD VOICE GROUP

WORLD VOICE GROUP est une entreprise de services du numérique qui a été créée en 2008 au Cameroun sous le nom de WORLD VOICE CAMEROUN avant de prendre le nom actuel en 2016 (création en Février 2016 et démarrage des activités en Mars 2016). Elle est spécialisée dans le développement et l'intégration logicielle, l'intégration et l'agrégation d'infrastructures réseaux et de télécommunication, la distribution digitale, l'audit, le conseil et la formation en informatique et en télécommunication. WORLD VOICE GROUP est une Société à Responsabilité Limitée (SARL) dont le promoteur est M. Richard Bilau KENFACK et ayant 15.000.000 FCFA de capital. Ce capital est détenu à 80 % par M. Richard Bilau KENFACK et à 20 % par Mme. Léonie CHOUAMO NOUMBIBOU.

WORLD VOICE CAMEROUN était spécialisée dans le développement et l'intégration logicielle, l'intégration et l'agrégation d'infrastructures réseaux et de télécommunication, l'audit, le conseil et la formation en informatique et en télécommunication. Le changement majeur qui a donné naissance à WORLD VOICE GROUP a été en 2016 la mise en avant de l'activité de distribution digitale. Cette activité qui est au cœur de WORLD VOICE GROUP n'était initialement pas prévue pour être un domaine dans lequel toute l'équipe de

l'entreprise se déploierait ; WORLD VOICE CAMEROUN avait alors pour ambition dans ce domaine de fournir des plateformes pour les distributeurs déjà présents au Cameroun.

Le siège de WORLDVOICE GROUP se trouve à Biyem-Assi - Acacia, à Yaoundé, plus précisément à l'immeuble « Digital Center » ; immeuble qui par ailleurs appartient à M. Richard Bilau KENFACK. WORLDVOICE GROUP y loue quatre (4) espaces : un espace pour l'administration, un autre espace pour son call-center, un troisième espace pour le travail collaboratif, et un quatrième espace qui est utilisé par ses ingénieurs et ses techniciens. En plus de ce siège social, WORLDVOICE GROUP dispose de points de vente, notamment un point de vente à Douala au carrefour Idéal à Akwa, un point de vente au Rond-point express à Biyem Assi à Yaoundé, un point de vente à l'école des postes à Yaoundé, un autre point de vente à l'école des travaux à Yaoundé, un point de vente à chapelle Obili à Yaoundé et un point de vente au quartier général à Yaoundé.

WORLD VOICE GROUP est une entreprise en constante évolution pour s'adapter aux mutations de l'environnement juridique, économique et technologique au Cameroun. L'un des aspects de cette évolution est une ouverture du capital dans les cinq ans à venir. Cette ouverture concernera à la fois des investisseurs privés et le personnel de l'entreprise.

1.2.2 Organisation de WORLD VOICE GROUP

WORLD VOICE GROUP est une entreprise de services du numérique qui a été créée en 2016 au Cameroun à la suite de WORLD VOICE CAMEROUN qui a mené ses activités de 2008 à 2016. Elle est actuellement dirigée par M. Richard Bilau KENFACK – Gérant de WORLD VOICE SPRL basée en Belgique qui était l'un des actionnaires de WORLD VOICE CAMEROUN.

La société WORLD VOICE GROUP est spécialisée dans le développement et l'intégration logicielle, l'intégration et l'agrégation d'infrastructures réseaux et de télécommunications, la distribution digitale, l'audit, le conseil et la formation en informatique et en télécommunications. Elle a développé récemment une nouvelle spécialité qui est l'informatique décisionnelle recouvrant la fouille et l'extraction des connaissances (pour l'aide à la décision) dans des données.

WORLD VOICE GROUP, depuis sa création, développe pour elle-même et pour ses partenaires et clients des plateformes logicielles pour leur gestion, pour leur processus de travail avec, dans la plupart des cas, l'utilisation intensive de smartphones (Android, iOS, Windows Phone) comme plateforme client pour accéder à des services. Elle a notamment participé à la conception, à la réalisation et à l'essor d'applications pour le paiement électronique de produits et services en ligne, le mobile Banking en l'occurrence une plateforme de services prépayées bancaires, les systèmes d'information décisionnels et la fouille de données en télécommunication.

M. Richard Bilau KENFACK, expert en technologie et voix sur IP, est le fondateur de WORLDVOICE GROUP, connu précédemment sous le nom de WORLD VOICE CAMEROUN. Lorsqu'il crée cette société en 2008, son objectif est de transférer auprès des ingénieurs camerounais, l'ensemble des technologies, des connaissances, des compétences, qu'il a acquises dans les télécommunications depuis 2004 où il a créé la société SPRL WORLD VOICE en Belgique. Ce transfert devait être effectué de manière à ce qu'à partir du Cameroun, de nouvelles offres de services, de nouvelles technologies, de nouvelles applications informatiques et en télécommunication soient proposées à des sociétés tant européennes qu'américaines, asiatiques et bien évidemment des sociétés camerounaises. Il ambitionne également de faire de cette société au Cameroun un pôle d'excellence en TIC, capable d'apporter des solutions informatiques aux problèmes auxquels est confrontée la diaspora camerounaise en particulier et la diaspora africaine de manière générale en rapport avec les TIC.

1.2.3 Organigramme de WORLD VOICE GROUP

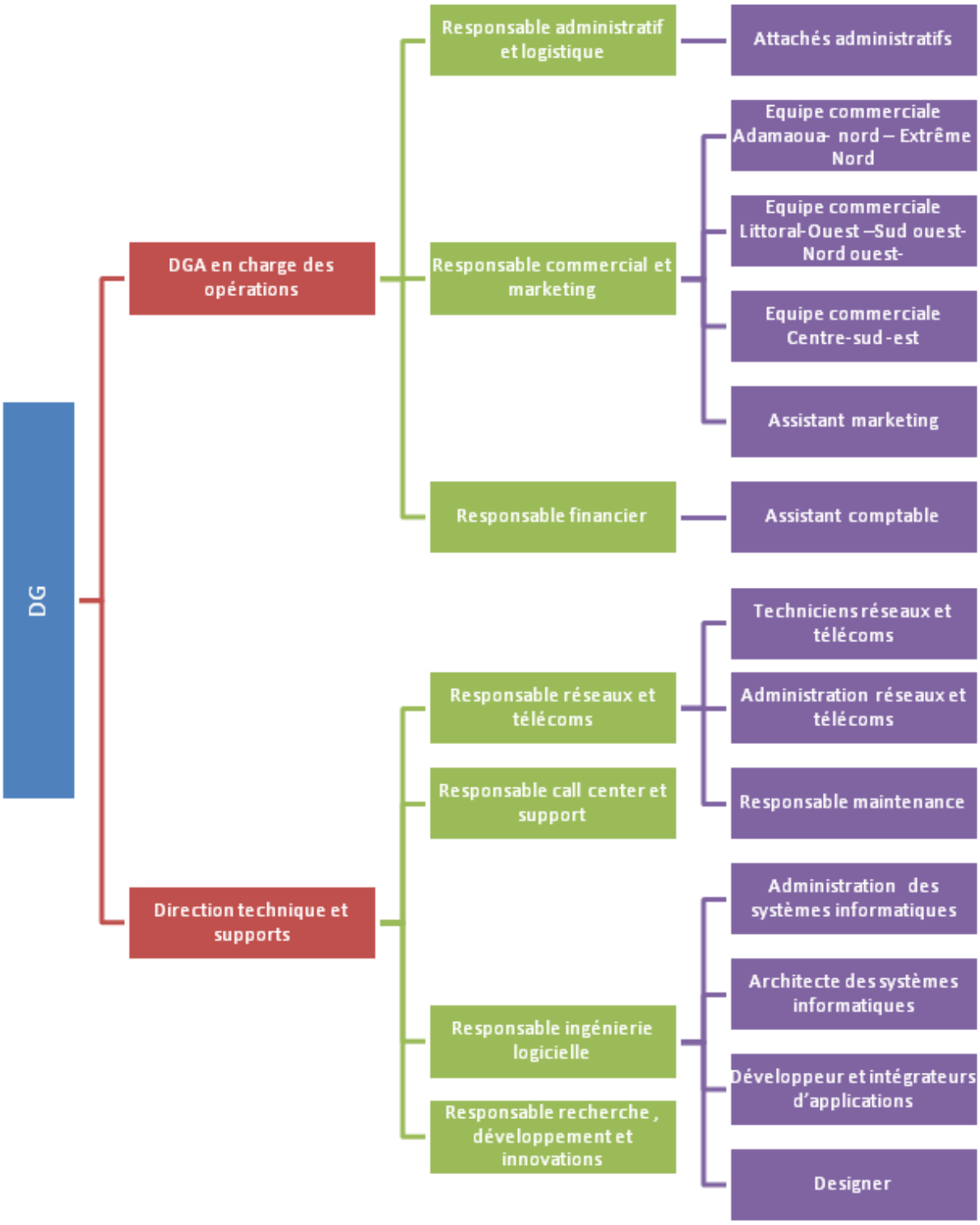


Figure 1.1 – Organigramme de WORLD VOICE GROUP SARL

1.3 Description de l'existant

WORLDVOICE GROUP pour gérer son support technique, commerciale, d'aide à l'utilisation de ses solutions, a mis en place des solutions de communication avec ses clients : Site web (formulaire de contact), adresse email, Whatsapp Business, CallCenter. Derrière ces plates-formes se trouvent les personnels de l'entreprise qui traitent tous les messages reçus.

1.3.1 par mail

Certains clients connaissent les contacts mail de WORLDVOICE GROUP et les utilisent pour poser leurs problèmes.

Une fois l'email envoyé par le client, un superviseur reçoit le mail, répond ou qualifie la requête à un personnel approprié, ce dernier peut traiter cette requête ou la qualifié à un autre personnel plus approprié. Ainsi de suite jusqu'à ce que le problème du client soit résolu.

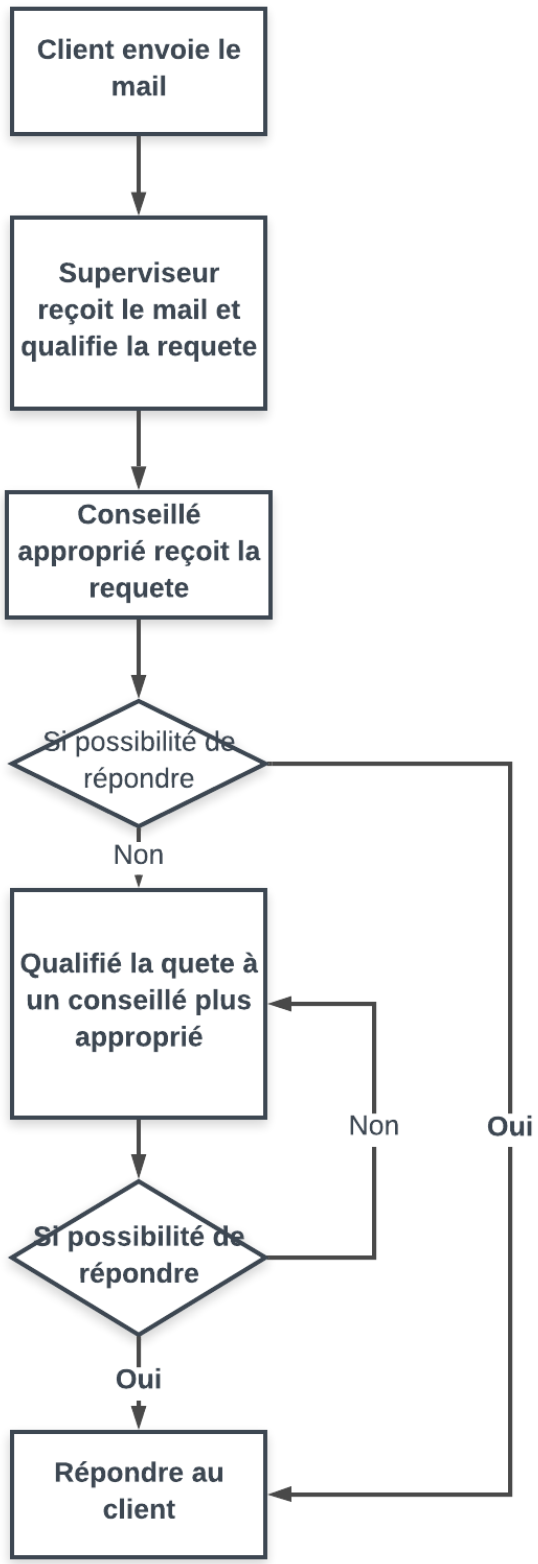


Figure 1.2 – Organigramme du fonctionnement actuel pour Email

Cette solution permet de tracer facilement les mails. Mais les requêtes des clients ne sont pas toujours structurées comme voulu par celui qui les traite. Celui qui traite des requêtes se trouve parfois en train faire la redondance des réponses. Par ailleurs, cette solution ne permet pas de gérer les urgences.

1.3.2 Par site web : formulaire de contact

WORLDVOICE GROUP dispose de plusieurs sites web, et sur ces sites web se trouve un formulaire de contact qu'un client peut utiliser pour envoyer sa requête.

Le client entre les informations suivantes : Adresse email, Téléphone, Objet, contenu du message. Ces informations sont envoyées et reçu sous forme de mail. A partir d'ici, la requête est traité de la même façon que précédemment (c'est-à-dire par le canal email).

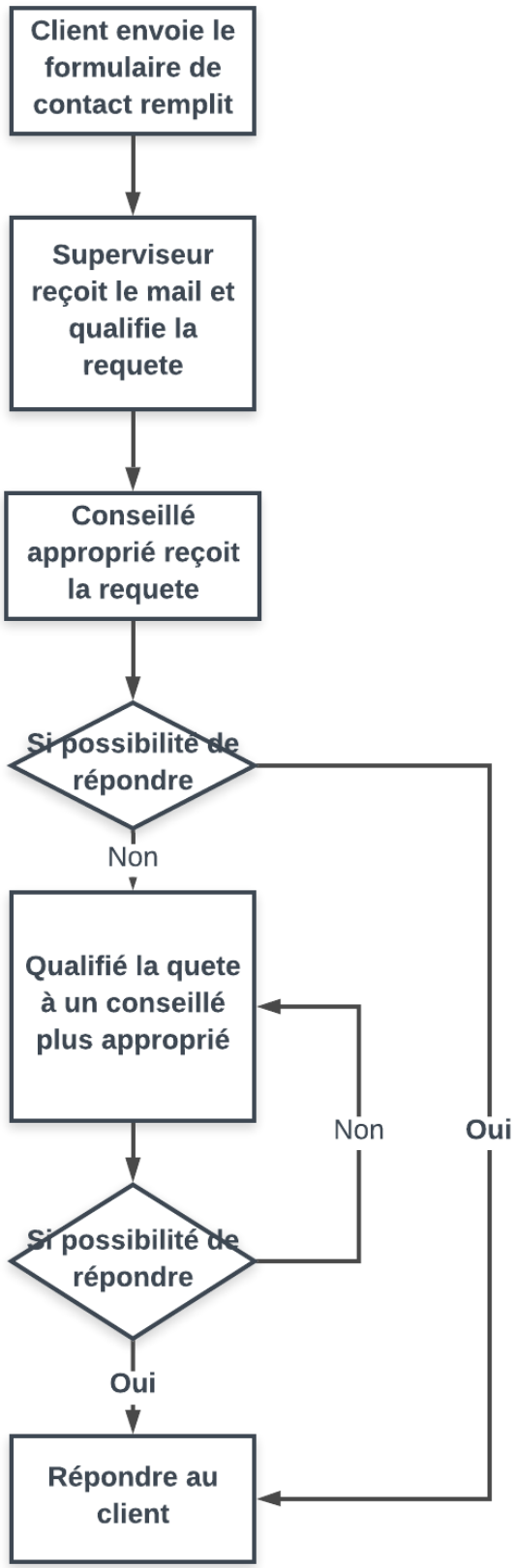


Figure 1.3 – Organigramme du fonctionnement actuel pour Formulaire de contact

Cette solution a les mêmes inconvénients que celle vu précédemment sauf que ici le formulaire est bien structuré et comporte des champs d'entrée obligatoires, ce qui permet d'avoir plus d'informations et de donner moins de travail dans la compréhension de la requête.

1.3.3 Whatsapp business

WORLDVOICE GROUP a une plateforme Whatsapp Business permettant de résoudre les problèmes des clients. Cette plate-forme a deux fonctionnalités : gestion des utilisateurs et gestion des messages ou tickets.

Pour la gestion des messages ou des tickets, tout commence par l'envoie d'un message sur whatsapp par client, ce qui donne lieu à l'ouverture d'un ticket sur la plateforme. L'administrateur reçoit la requête, la traite, ou transfère le ticket à un utilisateur de la plateforme ou à un département (commercial, technique, administratif, ...). Un utilisateur peut ensuite répondre à ce ticket et le client reçoit la réponse directement sur whatsapp. Le client peut aussi répondre par whatsapp et l'utilisateur reçoit sa réponse sur le même ticket.
Après plusieurs réponses, le problème du client peut être résolu, l'administrateur peut alors décider de fermer le ticket.
Lorsqu'un utilisateur est incapable de résoudre le problème d'un client, il qualifie aussi la requête à quelqu'un d'autre. Ce dernier peut aussi qualifier à un autre, ainsi de suite jusqu'à ce que le problème du client soit résolu.

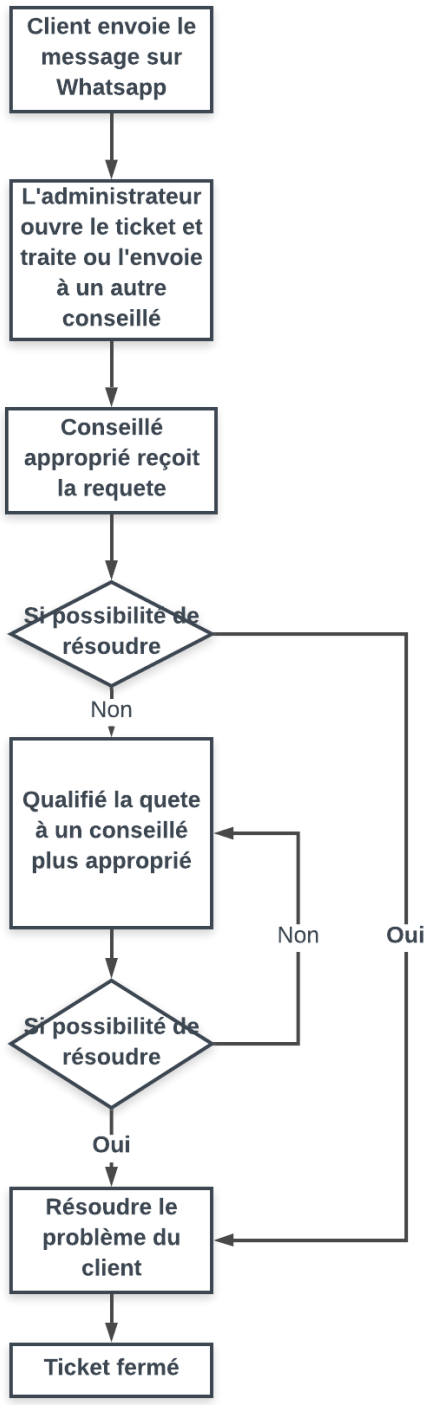


Figure 1.4 – Organigramme du fonctionnement actuel pour Whatsapp business

Cette méthode est beaucoup plus accessible que celles qui précèdent. Whatsapp est une application où l'on peut facilement s'adapter et est installer dans presque tous les

smartphones. Mais cette solution n'empêche pas la redondance des réponses.

1.3.4 CallCenter

WORLDVOICE GROUP met à la disposition de ses clients un numéro pour appel téléphonique.

Lorsque le client émet l'appel, un IVR (Serveur vocal interactif, qui est un système permettant de communiquer avec un utilisateur par téléphone) reçoit l'appel et demande au client de faire un choix sur ce qu'il lui propose (Par exemple, tapez 1 pour le service DigiPOS, tapez 2 pour avoir le service X, tapez 3 pour Y).
L'IVR peut lui-même répondre au client, si ce dernier a fait un choix correspondant aux problèmes dont la solution se trouve dans la base de connaissance de l'IVR.
En fonction du choix du client, l'IVR peut diriger l'appel vers un personnel approprié de l'entreprise. Le personnel analyse, comprend et traite la requête.
En cas d'incapacité de résolution de la requête par le personnel, ce dernier qualifie la requête à un autre personnel plus approprié. Ainsi de suite jusqu'à ce que le problème du client soit résolu.

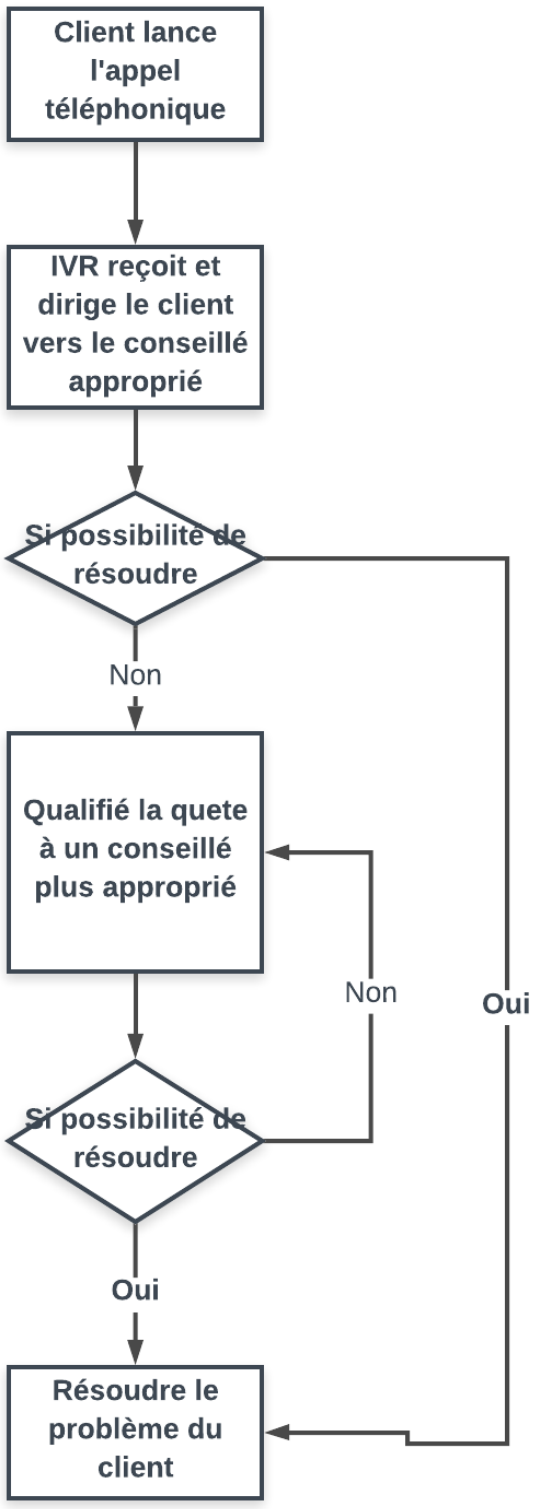


Figure 1.5 – Organigramme du fonctionnement actuel pour CallCenter

La méthode par appel téléphonique peut être plus importante que les autres (vu précédemment) dans la mesure où elle permet de gérer les urgences. Le client peut facilement et rapidement atteindre le conseiller client par appel téléphonique. Mais elle n'empêche pas la redondance des réponses et ne permet pas de tracer les conversations.

Quel qu'en soit le canal de communication, toute ressource humaine intervenant dans le traitement d'une requête peut accéder au système d'information (SI) de l'entreprise pour mieux répondre au client.
La complexité de traitement d'une requête peut à un moment donné amener la ressource humaine à continuer la communication par mail pour garder la traçabilité de la communication.

1.4 Limites de l'existant

Le processus actuel nous fait noter plusieurs conséquences néfastes à ne pas négliger :

- La ressource humaine est très utilisée pour traiter les requêtes des clients
- La ressource humaine accède au SI de l'entreprise pour apporter des réponses ; ce qui implique un engorgement car un personnel peut se trouver en train de qualifier une requête à un autre personnel, ce dernier peut aussi se trouver en train de qualifier à un autre, ainsi de suite. De ce fait on note une énorme perte de temps.
- le temps de réponse dépend aussi du nombre de requêtes reçus
- les heures de travail sont fixées de 8h du matin à 18h, donc aucun employé ne sera disponible pour répondre en dehors de cette période.
- Le processus actuellement adopté n'assure pas la disponibilité immédiate des réponses aux clients au temps voulu.

1.5 Solution proposée et objectifs

L'idée est de proposer une aide active pour automatiser une bonne partie du support client et sans monopoliser un grand nombre de conseillers. Cette aide devrait être proposée au client au moment voulu et non quelque temps après. D'où vient l'idée de créer **un Chabot (un agent conversationnel)**, ce dernier peut guider le client rapidement afin de lui trouver la solution sur Facebook Messenger, Skype, Telegram, sites web de WORLD VOICE GROUP, ... De ce fait, on note une optimisation de temps, de coût et de qualité des réponses.

Objectifs :

- Créer un chatbot qui va analyser, comprendre et traiter la requête du client.
- un chatbot qui va identifier la bonne ressource humaine vers lequel router la requête en cas d'incapacité de résolution.
- Pemettre au bot d'accéder au SI de l'entreprise (via les API) pour puiser les données et donner une réponse fiable au client ;
- **Sécuriser les accès aux SI de l'entreprise.**

L'entreprise WORLDVOICE GROUP exhortera ses clients à communiquer avec le chatbot pour tout préoccupation, et en cas d'incapacité de résolution le chatbot les dirigera vers un conseiller client.

1.6 Conclusion

Dans ce chapitre nous avons présenté le contexte et la problématique liés à notre travail. En recensant les problèmes rencontrés par l'entreprise WORLD VOICE GROUP dans le support client, nous proposons de résoudre par la mise en place d'un chatbot sécurisé d'entreprise. Dans le chapitre suivant nous présenterons le fonctionnement d'un chatbot.

Chapitre 2

AGENT CONVERSATIONNEL (CHATBOT)

2.1 Introduction

Un chatbot, aussi appelé « **agent conversationnel** », est un programme informatique capable de simuler une conversation avec un ou plusieurs humains par échange vocal ou textuel.

Cet outil est aujourd'hui très utilisé sur Internet par les services clients de marques ou de commerçants en ligne à travers la messagerie instantanée.
Nous allons dans ce chapitre présenter l'architecture, l'importance et les technologies de mise en oeuvre d'un chatbot d'entreprise

2.2 Intelligence artificielle

L'intelligence artificielle (IA, ou AI en anglais pour Artificial Intelligence) consiste à mettre en œuvre un certain nombre de techniques visant à permettre aux machines d'imiter une forme d'intelligence réelle. L'IA se retrouve implémentée dans un nombre grandissant de domaines d'application.
L'intelligence artificielle est devenue omniprésente dans nos vies et elle est présentée comme la solution à tous nos problèmes.

L'IA est en réalité une discipline jeune d'une soixantaine d'années, qui réunit des sciences, théories et techniques (notamment logique mathématique, statistiques, probabilités, neurobiologie computationnelle et informatique) et dont le but est de parvenir à faire imiter par une machine les capacités cognitives d'un être humain [2].

Les spécialistes préfèrent en général employer le nom exact des technologies concrètement en œuvre (qui relèvent aujourd'hui essentiellement de l'apprentissage automatique – machine learning) et sont parfois réticents à employer le terme d' « intelligence » car les

résultats, bien qu'extraordinaires dans certains domaines, demeurent encore modestes au regard des ambitions entretenues.

2.2.1 Machine learning

Le Machine Learning, aussi appelé apprentissage automatique en français, est une forme d'intelligence artificielle permettant aux ordinateurs d'apprendre sans avoir été programmés explicitement à cet effet. Cette technologie permet de développer des programmes informatiques pouvant changer en cas d'exposition à de nouvelles données. Découvrez la définition, le fonctionnement et les secteurs d'applications du Machine Learning.
Le Machine Learning est une méthode d'analyse de données permettant d'automatiser le développement de modèle analytique. Par le biais d'algorithmes capables d'apprendre de manière itérative, le Machine Learning permet aux ordinateurs de découvrir des insights cachées sans être programmés pour savoir où les chercher.[3]

2.2.2 chatbot

Fondamentalement un **chatbot** est une application conversationnelle qui permet à un utilisateur de dialoguer, que ce soit via les reseaux sociaux (Facebook Messenger, WhatSapp, Skype, etc.) ou directement sur un site internet. Le terme **chatbot** est la contraction de deux parties : le **"chat"** désigne une discussion, et le **"bot"** désigne un robot.
Un chatbot est un logiciel robotisé dialoguant avec un individu par le biais de l'automatisation. **Ce robot "analyse" et "comprend"** les messages à travers sa bibliothèque de questions-réponses. Un chatbot fonctionne de différentes manières. Il réagit à la voix de l'utilisateur ou lorsque celui-ci tape quelque chose sur son clavier. Aussi bien à partir du texte que de la voix, le chatbot va décrypter les mots-clés de la requête pour y répondre.

Le chatbot est donc une partie de l'intelligence artificielle qu'on appelle intelligence conversationnelle ; et le robot qui est derrière la conversation est appélé agent conversationnel.

2.3 Importance d'un chatbot pour l'entreprise

Le service à la clientèle : Les clients posent beaucoup de questions aux marques/société, et souvent les mêmes. Le service à la clientèle peut être fastidieux. Et dans ce cas, le chatbot peut être d'une aide précieuse et augmenter le taux de satisfaction. Attention tout de même à bien traiter les interactions. Une réponse hors sujet pourrait avoir l'effet contraire : le mécontentement.

Générer des transactions : Le chatbot peut se transformer en vendeur, car contrairement à une simple publicité, la conversation établie avec le chatbot est personnalisée. Il peut en effet orienter un utilisateur vers un produit selon ses goûts, ses préférences,

son âge, etc. Ainsi, une conversation peut aboutir habilement à une vente, sans inonder l'utilisateur de publicités.

Partager du contenu : Finalement, une messagerie instantanée est très proche d'un réseau social. L'utilisateur discute et partage du contenu avec ses amis. Ainsi, le chatbot peut pousser l'utilisateur à partager et propager du contenu à travers son réseau.

Une expérience inédite : Si l'on parvient à faire de son chatbot une expérience nouvelle, l'engagement sera grandement amélioré. Ne pas dupliquer une fonctionnalité sur le site ; l'intérêt est d'apporter un regard nouveau avec ce nouveau mode de communication. Par exemple, un chatbot demande à l'utilisateur son humeur actuelle. Puis, trois questions lui sont posées : le type et la taille de son animal préféré et sa préférence à être seul ou en groupe. Selon les réponses, le chatbot envoie une courte vidéo animalière. Après avoir regardé la vidéo, le chatbot lui redemande son humeur à l'aide d'émoticônes. Nous constatons donc que le but est d'améliorer l'humeur des utilisateurs à travers de courtes vidéos sur la nature. Ce qui est une expérience inédite.

2.4 Modèle d'architecture pour un chatbot

Le schema suivant présente le mobèle d'architecture d'un chatbot.

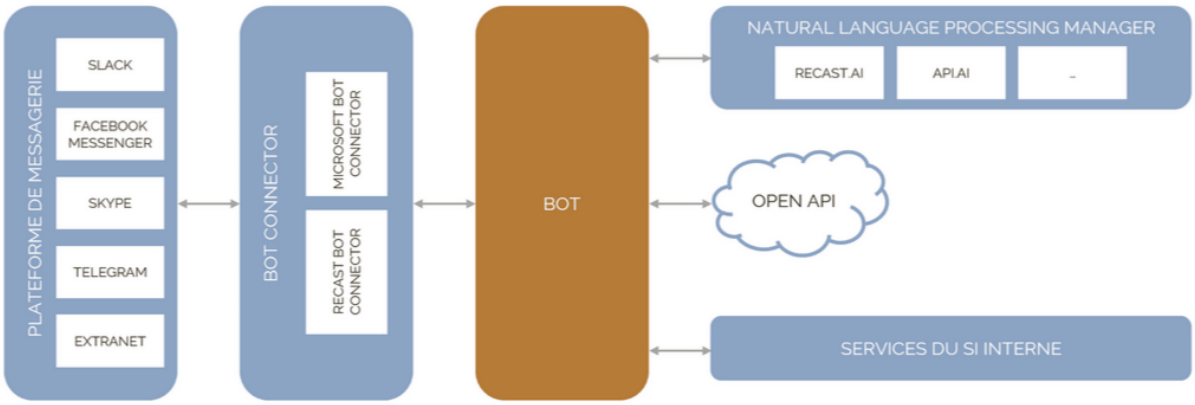


Figure 2.1 – Modèle d'architecture pour un chatbot [5]

Explications :

- **La plateforme de messagerie (côté frontal) :** Slack, Facebook Messenger, Skype, Telegram, ou un extranet de l'Entreprise.
  Ce sont les différents moyens de conversation que l'utilisateur peut choisir pour échanger avec le bot. Chaque plateforme utilise un protocole dédié, qu'il soit public

ou non. S'il est plutôt simple de s'interfacer avec une seule plateforme, offrir à l'utilisateur le choix de son canal de communication exige de multiplier les interfaces, et la maîtrise de tous les protocoles devient réellement complexe.

- **Le BotConnector :** Le BotConnector répond à cette complexité et permet de s'intégrer simplement dans plusieurs plateformes de conversations au travers d'une seule et même interface. Il évite donc de devoir développer (et maintenir) un bot multi plateformes ou une version spécifique du bot adaptée aux spécificités de chacune des plateformes possibles. Les solutions de BotConnector sont récentes (2016). Elles permettent aux développeurs de bots de toucher rapidement et sans effort un large public.

- **Natural Language Processing Manager (NLP) (côté back)** Le "Natural Language Processing Manager" (Le moteur de traitement automatique de langage naturel) est un domaine de l'intelligence artificielle centré autour de la logique conversationnelle. Il regroupe différents concepts qui donnent au bot sa capacité à comprendre le sens d'une phrase, quelles que soient les fautes de frappe ou les errances orthographiques et grammaticales de l'auteur. Cette phase d'interprétation requiert une étape d'entraînement pendant laquelle le NLP Manager est confronté à des cas réels. Il s'agit de le corriger à chaque erreur afin qu'il puisse **"apprendre"** à reconnaître le sens des phrases qui lui seront soumises.

- **Le bot (côté central) :** Le bot est avant tout un workflow (processus permettant d'automatiser la circulation des flux d'information dans une entreprise) conversationnel, qui est déroulé pendant l'échange avec l'utilisateur afin de lui apporter les réponses adaptées à ses demandes. Le bot est au centre de l'architecture, interfacé avec le NLP Manager qui lui donne le sens de la demande utilisateur et les API qui lui remontent l'information dont il a besoin.

- **Open API :** Il s'agit d'API publiques qui permettent à tout un chacun de consommer des données dites ouvertes, proposées par des acteurs tels que Météo France, SNCF ou RATP. Beaucoup de bots profitent de ces APIs pour construire des réponses riches et diversifiées à l'utilisateur sur ses préoccupations quotidiennes.

- **Services du SI : (côté back)** Chargé de répondre à des questions plus complexes requiert un accès au Système d'Information et l'exploitation des données personnelles qu'il contient.

  Par exemple on peut parler de consulter le solde de son compte bancaire ou d'exécuter un virement.

On se frotte alors aux problématiques sensibles d'identification, d'authentification et d'autorisation de l'utilisateur pour accéder à ses données confidentielles.

2.5 Technologie de mise en oeuvre d'un chatbot

Ce n'est qu'une fois que l'on a parfaitement défini le but principal du chatbot qu'il est dans la bonne voie pour choisir la technologie adéquate. C'est en comparant les avantages et les inconvénients des différentes technologies que nous trouverons celle qui répondra à toutes nos exigences. Il existe une panoplie de technologies, dont quelques unes sont : Botnation IA, Recast .IA, Manychat, Chatfuel, Botsify, Snatchbot, Quriobot [20]. Ces technologies ont chacun des avantages et inconvénients. Notre choix pour la mise en place du chatbot sera porté sur Snatchbot ; Ceci parce qu'elle couvre tous les modules du modèle d'architecture présenté précédemment et aussi pour des raisons groupées dans le tableau suivant :

Sophistication de la technologie et des usages	Machine Learning
Ergonomie et approche pédagogique	Logigramme du bot. Test en ligne du chatbot
Facilité de développement	Interaction simple ou extraction d'infos : mail, url, téléphone, date, adresse, durée, traduction, humain ou JSON (intégration du robot dans l'API). Message de bot : insertion lien, message différé, opération arithmétique, obtention d'attributs facebook(prenom, nom, photo de profil, lieu, fuseau horaire, sexe).
Facilité d'intégration et d'ajout	Site internet Messenger, Slack, email, skype, twilio ou même SMS
Coût	Gratuit
langues supportées	plusieurs y compris le français

2.6 Conclusion

Le chatbot est devenu une technique fiable de marketing, du support client en ligne et autres. Nous avons vu en détail dans ce chapitre le fonctionnent d'un chatbot, et en particulier la communication entre un bot et un système d'information (SI) via des API. Nous pensons ainsi à un point indispensable qui sera présenté dans le chapitre suivant : **la sécurisation des API**.

Chapitre 3

SECURISATION DES API

3.1 Introduction

La **sécurité d'un système d'information** est une discipline qui se veut de protéger l'intégrité et la confidentialité des informations stockées dans un système d'information. Dans notre contexte, l'accès à une donnée ou information se fait à travers des **API** qui sont aujourd'hui omniprésentes, elles permettent la communication et l'échange de données entre applications et systèmes hétérogènes. Vient alors la question de la sécurisation de l'échange de ces données. Plusieurs éléments que nous allons présenter dans ce chapitre entrent en jeux dans la sécurisation d'une API à savoir les protocôles d'authentication et d'autorisation, les modèles de contrôle d'accès. .

3.2 C'est quoi une API REST

3.2.1 API

L'**API**, pour **Application Programming Interface**, est la partie du programme qu'on expose officiellement au monde extérieur pour manipuler celui-ci. L'API est au développeur ce que l'User Interface est à l'utilisateur. Cette dernière permet d'entrer des données et de les récupérer à la sortie d'un traitement. Initialement, une API regroupe un ensemble de fonctions ou méthodes, leurs signatures et ordre d'usage pour obtenir un résultat.

La mise en place d'une API permet d'opérer une séparation des responsabilités entre le client et le serveur. Cette séparation permet donc une portabilité et évolutivité grandement améliorées. Chaque composant peut évoluer séparément car il n'y a aucun logique du côté du serveur. Ainsi on peut imaginer une refonte totale de la charte graphique du site web sans devoir modifier le code côté serveur ou sur les autres clients (mobiles par exemple).

3.2.2 API REST

Representational State Transfer (REST) est un type d'architecture basé sur le protocole HTTP utilisé pour les API. C'est une solution permettant à un client d'accéder à des services web.
Une API REST se doit d'être sans état ou stateless en anglais. La communication entre le client et le serveur ne doit pas dépendre d'un quelconque contexte provenant du serveur. Ainsi, chaque requête doit contenir l'ensemble des informations nécessaires à son traitement. Cela permet au serveur de traiter indifféremment les requêtes de plusieurs clients via de multiples instances de serveurs.
une API REST pour manipuler les ressources, utilise les méthodes HTTP suivantes :

- **GET** : Récupération d'une ressource ;
- **POST** : Ajout d'une ressource ;
- **PUT** : Mise à jour d'une ressource ;
- **DELETE** : Suppression d'une ressource ;
- **HEAD** : Similaire à GET, mais permet uniquement de récupérer les en-têtes HTTP.

La figure suivante permet de mieux comprendre les échanges entre un client REST et une API REST :

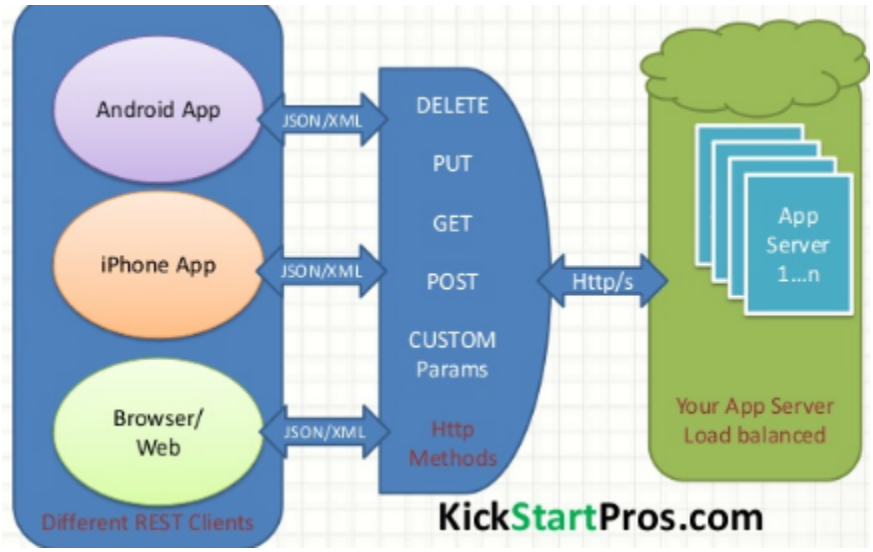


Figure 3.1 – Fonctionnement d'une API REST [14]

3.3 Sécurité informatique

Le **risque** en terme de sécurité est généralement caractérisé par **les menaces, les vulnérabilités et les contre-mesures**.

La **menace** (en anglais « threat ») représente le type d'action susceptible de nuire dans l'absolu, tandis que **la vulnérabilité** (en anglais « vulnerability », appelée parfois faille ou brêche) représente le niveau d'exposition face à la menace dans un contexte particulier. Enfin la **contre-mesure** est l'ensemble des actions mises en oeuvre en prévention de la menace.

Les **contre-mesures** à mettre en oeuvre ne sont pas uniquement des solutions techniques mais également des mesures de formation et de sensibilisation à l'intention des utilisateurs, ainsi qu'un ensemble de règles clairement définies.

Le **système d'information** est généralement défini par l'ensemble des données et des ressources matérielles et logicielles de l'entreprise permettant de les stocker ou de les faire circuler. Le système d'information représente un patrimoine essentiel de l'entreprise, qu'il convient de protéger.
La **sécurité informatique**, d'une manière générale, consiste à assurer que les ressources matérielles ou logicielles d'une organisation sont uniquement utilisées dans le cadre prévu.

Les Objectifs de la sécurités :

La sécurité informatique vise généralement quatres principaux objectifs :

- **L'intégrité**, c'est-à-dire garantir que les données sont bien celles que l'on croit être ;
- **La confidentialité**, consistant à assurer que seules les personnes autorisées aient accès aux ressources échangées ;
- **La non répudiation**, permettant de garantir qu'une transaction ne peut être niée ;
- **L'authentification**, consistant à assurer que seules les personnes autorisées aient accès aux ressources.

Les causes de l'insécurité : [19]

On distingue généralement deux types d'insécurités :

- **l'état actif d'insécurité**, c'est-à-dire la non connaissance par l'utilisateur des fonctionnalités du système, dont certaines pouvant lui être nuisibles (par exemple le fait de ne pas désactiver des services réseaux non nécessaires à l'utilisateur)
- **l'état passif d'insécurité**, c'est-à-dire la méconnaissance des moyens de sécurité mis en place, par exemple lorsque l'administrateur (ou l'utilisateur) d'un système ne connaît pas les dispositifs de sécurité dont il dispose.

Une API REST a besoin d'être sécurisé dans la mesure où certaines ressources requiert une authentification et une autorisation d'accès.

3.4 Protocoles d'authentification et d'autorisation

En informatique un protocole définit les règles et les procédures permettant à deux processus informatiques d'échanger des données. Il existe plusieurs protocoles de sécurisation des APIs.

3.4.1 Le protocole Open Authorization (OAuth2)

OAuth2 spécifie un protocole de délégation d'accès. Son but principal est donc de décrire comment l'accès à des API sécurisées d'une application ou d'un site web (fournisseur) va être délégué à une autre application (consommateur).

Le protocole distingue 4 rôles principaux :

- **Resource Owner** : celui qui détient les ressources,
- **Resource Server** : serveur qui héberge les ressources protégées,
- **Client** : application cliente (front, back ou mobile) qui demande l'accès aux ressources,
- **Authorization Server** : serveur qui génère des jetons (tokens) pour le client et qui seront transmis lors des requêtes vers le serveur de ressources.

La notion de token

- **Access Token** : token permettant de valider un accès à un service sécurisé (une autorisation) avec une durée de vie définie. Il est indispensable.
- **Refresh Token** : token dit "de renouvellement" de longue durée permettant de demander la création d'un nouvel Access Token si ce dernier est expiré. Il est émis au même moment que l'Access Token mais n'est pas envoyé à chaque requête. Le Refresh Token doit être stocké par l'application cliente de manière sécurisée.

Enregistrement des clients

Il faut noter que dans le cadre du protocole OAuth2, chaque application cliente qui désire accéder à des ressources protégées doit au préalable s'enregistrer auprès du serveur d'autorisation (généralement via un formulaire). La spécification OAuth2 précise les paramètres standards que les clients doivent renseigner lors du processus d'enregistrement :

- **Application Name** : nom de l'application,
- **Redirect URI (ou Callback URL)** : URI (ou URL) de l'application cliente vers laquelle seront faites les redirections par le serveur d'autorisation, une fois l'accès aux ressources autorisé (ou bien lorsque l'accès sera refusé),

- **Grant Type(s)** : types d'autorisation qui pourront être utilisés par le client lors de la demande de ressources protégées.

Le serveur d'autorisation délivre en retour un couple client_id/client_secret :

- **client_id** : chaîne de caractères générée de façon aléatoire qui identifie une application cliente de manière unique,
- **client_secret** : chaîne de caractères représentant la clé secrète du client et qui sera utilisée lors de l'appel à certaines API nécessitant une entête HTTP Authorization.

Le flux de demande d'accès à des ressources sécurisées peut donc être représenté de la manière générique suivante :

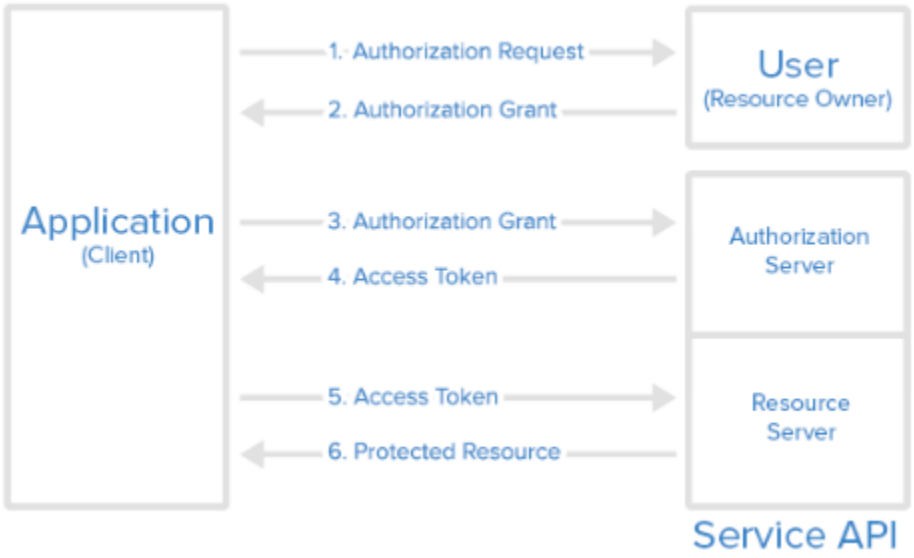


Figure 3.2 – Séquencement OAuth2 [17]

1. L'application cliente envoie une demande d'accès aux ressources protégées de l'utilisateur en précisant notamment son identité (client_id), le type d'autorisation et les scopes souhaités. Les scopes sont déterminés par le serveur d'autorisation au préalable. Plus l'API est découpée en scopes petits, et plus le profil de chaque client est précis (et donc limité). Chaque application cliente ne connaît que les scopes qu'elle peut utiliser.
2. Si l'utilisateur approuve la requête, un droit d'accès est renvoyé au client.
3. Le client demande alors un Access Token au serveur de jetons en fournissant son identité (ex : des credentials client), ainsi que son droit d'accès reçu précédemment.
4. Si l'identité est validée (le client est bien authentifié) et que le droit d'accès est valide, le serveur de jetons lui délivre un Access Token.

5. Le client peut ensuite demander l'accès aux ressources protégées au serveur de ressources en présentant son Access Token.

6. Si l'Access Token fourni est valide, les ressources demandées sont renvoyées au client.

HTTPS (HYPERTEXT TRANSFER PROTOCOL SECURED) : Pour tout type d'implémentation de sécurité, allant de l'authentification de base à une implémentation à part entière de OAuth2, HTTPS est un plus. Sans HTTPS, quelle que soit l'implémentation, la sécurité est susceptible d'être compromise.
Donc la spécification OAuth2 impose que tous ces échanges aient lieu en HTTPS.

Selon le type d'autorisation indiqué par le client, ce schéma sera implémenté de différentes façons. Dans la spécification OAuth2, on distingue 4 types d'autorisations :

- **Authorization Code** : utilisé si l'application cliente est située côté serveur. Cas le plus implémenté.

- **Implicit** : utilisé si l'application cliente est située côté client (ex : une application Javascript ou une application mobile) et qu'aucun autre type d'autorisation n'est utilisable. Ce mode est moins sécurisé car le token ne reste pas côté serveur, mais est exposé côté client et peut être intercepté.

- **Resource Owner Credentials** : les identifiants de connexion sont envoyés au client puis au serveur d'autorisation. Cela implique qu'il y ait une confiance absolue entre les deux. Souvent utilisé lorsque le client a été développé par la même entité que celle fournissant le serveur d'autorisation (ex : un accès à des ressources sécurisées d'un sous-domaine), ce type d'autorisation est très fortement déconseillé car OAuth a été pensé justement pour que les identifiants de connexion ne soient plus transmis aux applications tierces. De plus il n'y a pas de vérification de l'URL de callback.

- **Client Credentials** : utilisé lorsque le client est lui-même le propriétaire des données. Il n'y a donc pas d'autorisation spécifique à obtenir de la part de l'utilisateur. Les échanges commencent directement à l'étape 3.

Il est important de retenir qu'OAuth2 ne gère pas l'authentification. A aucun moment il n'est question d'informations utilisateur, de rôles ou d'habilitations. Afin d'avoir une solution d'identité complète, il est nécessaire d'utiliser le protocole OpenID Connect.

3.4.2 Protocole OpenID Connect

OpenID Connect est un protocole qui gagne en popularité car c'est une surcouche à OAuth2 (il est capable de répondre à tous ses cas d'utilisation), et ajoute de nouvelles fonctionnalités qui manquaient à OAuth2 :

- La prise en charge de **l'authentification**,

- La notion d'**ID Token**,
- La gestion de la **session SSO** (ex : le Single Logout),
- Une nouvelle API pour **récupérer les informations** utilisateur (User Info endpoint),
- **Standardisation** des informations utilisateurs,
- Un système de **découverte du serveur OpenID** afin de permettre aux clients de s'enregistrer par eux-mêmes.

La notion d'ID Token

Un ID Token est un jeton auto-portant qui contient l'identité d'un utilisateur. Il est véhiculé au format JWT et est constitué principalement :

- des paramètres de l'authentification :
  - date d'expiration,
  - date de création,
  - date d'authentification,
  - des moyens de contrôle permettant de valider l'ID Token et l'Access Token.
- des accréditations (rôles, habilitations) de l'utilisateur :
  - formalisme à déterminer par le fournisseur d'identité.
- des attributs (claims) de l'utilisateur.

Les attributs sont associés à des "scopes" :

- attributs **standards** :
  - scope profile : nom, prénom, surnom, date de naissance, ...
  - scope email : email, email vérifié
  - scope address : adresse
  - scope phone : numéro de téléphone, numéro de téléphone vérifié
- attributs **privés** : attributs proposés par le fournisseur d'identité. Il est nécessaire de les spécifier afin d'éviter toute collision avec des claims existants.

OpenID Connect propose plusieurs interfaces (endpoints) :

- **authorization** : pour authentifier un utilisateur,
- **token** : pour demander un token (access / refresh / ID),

- **user info** : pour récupérer des informations sur l'utilisateur (son identité, ses droits),
- **revocation** : pour supprimer un token (access / refresh),
- **introspection** : pour valider un token (access / refresh).

D'autres interfaces optionnelles sont disponibles pour l'enregistrement de clients, la découverte de fournisseurs OpenID Connect, etc.

La notion de Authorization code

OpenID Connect propose trois algorithmes pour déterminer comment retourner les tokens :

Authorization Code Flow :
L'algorithme retourne un code d'autorisation pour ensuite récupérer des tokens :

- les tokens sont retournés uniquement par l'interface token,
- la récupération d'un token d'accès s'effectue en deux étapes :
  - un code est retourné par l'interface authorization,
  - ce code est envoyé par le client à l'interface token.
- le client doit être enregistré auprès du fournisseur OpenID (via un identifiant et un secret),
- s'applique très bien aux applications mobiles, web et back-end,
- **algorithme le plus implémenté.**

Implicit Flow L'algorithme retourne directement les tokens.

- les tokens sont retournés directement par l'interface authorization (l'interface token n'est plus utilisée),
- il n'y a pas d'enregistrement des clients,
- il n'y a pas de notion de Refresh Token,
- les tokens à durée de vie longue ne sont pas autorisés,
- algorithme pour les applications type Javascript (sans back-end).

Hibrid Flow C'est un mix entre l'Authorization Code Flow et l'Implicit Flow.

- le séquencement est identique à l'Authorization Code Flow à l'exception du fait que l'interface authorization peut retourner le code, l'ID Token et l'Access Token,
- le Refresh Token quant à lui s'obtient par un appel à l'interface token,
- algorithme très peu utilisé.

Focus sur l'algorithme d'authentification Authorization Code Flow : Le diagramme ci-dessous présente l'algorithme d'authentification dans sa forme basique.
Les points suivants sont à noter :

- La spécification prévoit que l'OpenID Provider fournisse (ou délègue à un autre OpenID Provider) la mire d'authentification,
- La spécification prévoit également que l'OpenID Provider demande le consentement de l'utilisateur sur l'accès aux données de son identité (accès à des périmètres précis, mail, adresse, téléphone, ...).

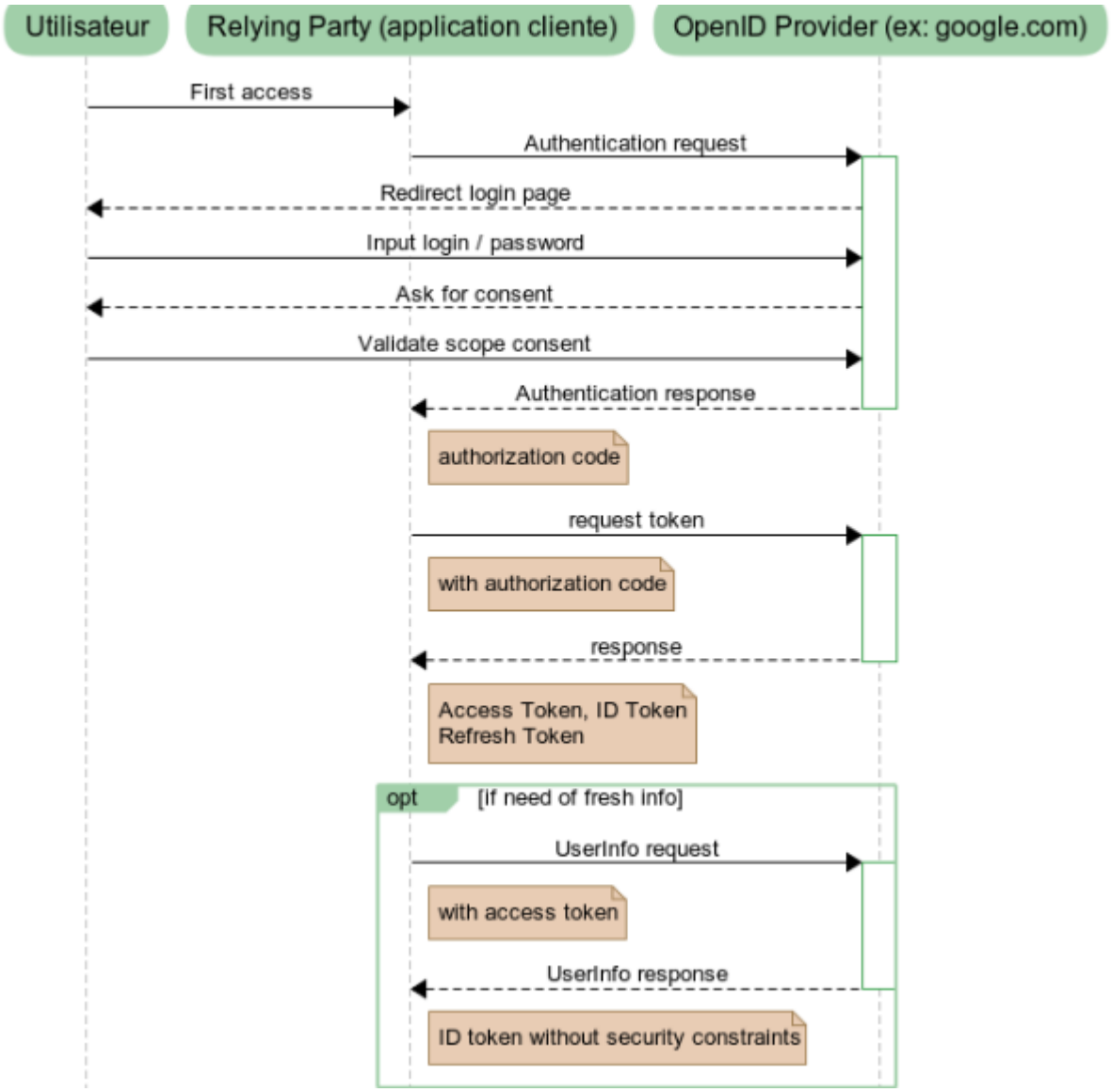


Figure 3.3 – Algorithme Authorization code flow [17]

Notons que la récupération des informations utilisateurs “UserInfo request” n'est pas obligatoire puisque l'ID Token peut être alimenté avec les informations utilisateurs lors de la demande de token.

3.4.3 Le protocole SAML

Le SAML « Security Assertion Markup Language » est un protocole ouvert et standardisé basé sur le langage XML pour échanger des informations d'authentification et d'autorisation entre des entités ou domaines de sécurité.[7]

Le SAML va gérer à la fois le format du message XML, appelé assertion, ainsi que les renseignements nécessaires à l'authentification et le process d'échange entre deux grands partenaires :

- Le SP (Service Provider) ou fournisseur de service, qui protège l'accès aux ressources demandées (sites web, applications etc) en appliquant une politique de sécurité. Par exemple, il bloque tout accès à un utilisateur non authentifié et le dirige vers son fournisseur d'identité.
- L'IdP (Identity Provider) est le fournisseur d'identité qui répond à la demande du SP. Il est chargé d'authentifier l'utilisateur et de forger la réponse contenant les informations associées à l'identité (groupe en général) et demandées par le SP.

3.4.4 Un focus sur JSON Web Token (JWT)

JWT est un format d'échange d'informations standard et sécurisé. Un token JWT est décomposé en trois parties, séparées par le caractère **.** :

header :
Le header (JOSE header) décrit l'algorithme utilisé pour signer ou chiffrer le token. Par exemple pour une signature utilisant l'algorithme HS256 (HMAC with SHA-256), nous aurions : **”alg” :”HS256”,”typ” :”JWT”** Le header doit ensuite être encodé en base64.

payload :
C'est le contenu du token. Par exemple notre ID Token décrit précédemment :

```
{
"jti": "f232b54cb285452db02770c9d16f8f212151",
"iss": "http://server.meritis.fr",
"sub": "24400320",
"aud": "s6BhdRkqt3",
"nonce": "n-0S6_WzA2Mj",
"exp": 1515604697,
"iat": 1515593897,
"name": "Matthieu Mabyre",
"given_name": "Matthieu",
"family_name": "Mabyre",
"gender": "male",
"email": "matthieu.mabyre@meritis.fr",
"acr": ["role1","role2", "role3"]
}
```

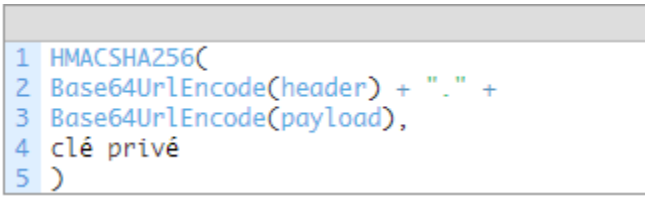
Figure 3.4 – un exemple de payload

Le champ **jti** (JWT id) correspond à l'identifiant du token. Par exemple un UUID généré aléatoirement.
Le payload doit également être encodé en base64 par la suite.

Signature du jeton :
Elle est effectuée en utilisant l'algorithme de signature (défini dans le header) à partir :

- du header au format base64 url encodé,
- du payload au format base64 url encodé,
- de la clé privée du serveur d'autorisation.

La signature des tokens vise à garantir que les tokens auto-portants générés par le serveur d'autorisation ont bien été générés par ce serveur, et qu'ils n'ont pas été altérés par un tiers.



```
HMACSHA256(
Base64UrlEncode(header) + "." +
Base64UrlEncode(payload),
clé privé
)
```

Figure 3.5 – un exemple JWT

Exemple :
Au final, nous obtenons un jeton JWS (S pour Signed) encodé en base64 (sa taille va dépendre de la taille du payload) :

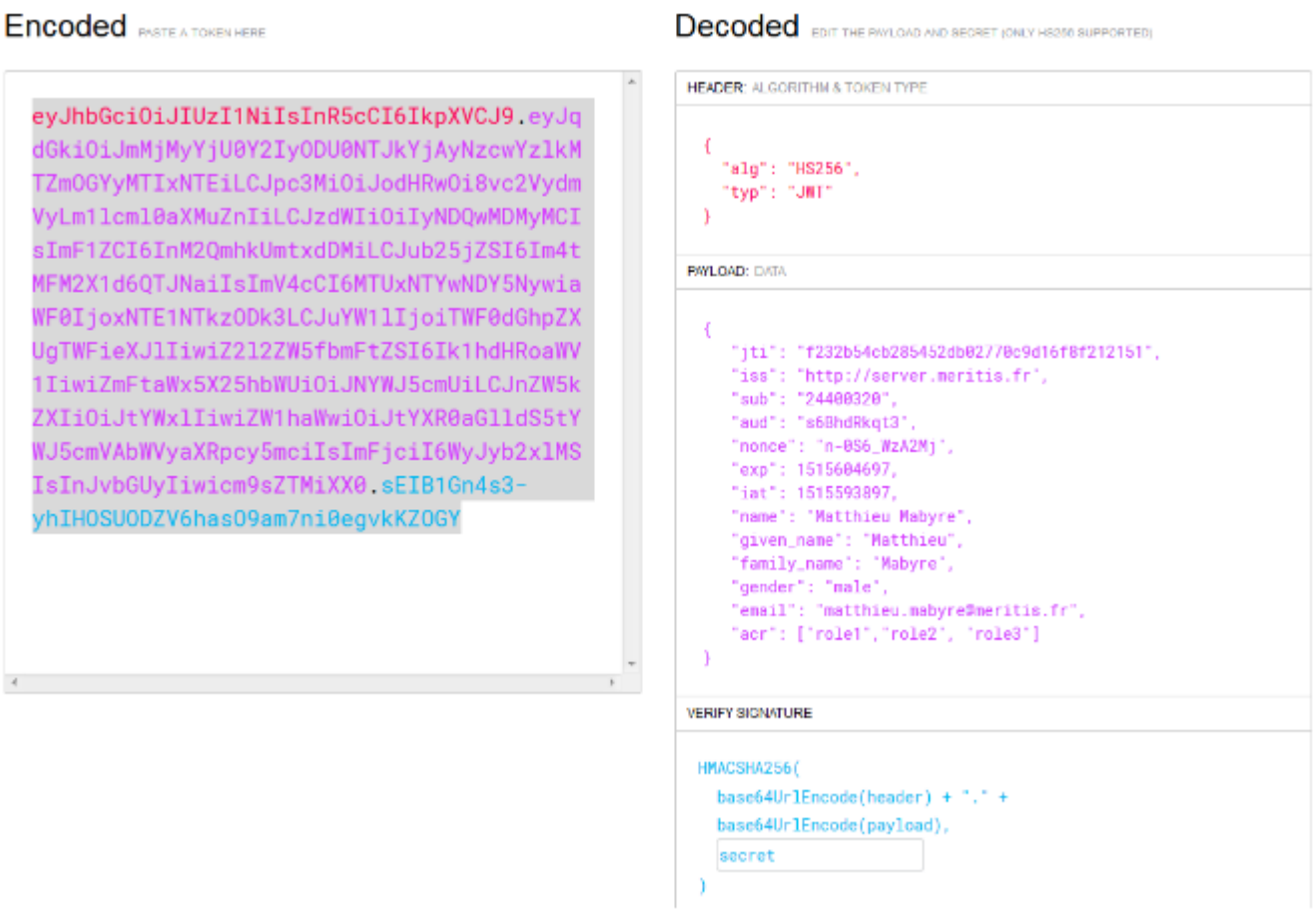


Figure 3.6 – un exemple JWS [7]

Notons que si nous avions décidé de chiffrer le jeton JWT, ce serait devenu un JWE (E pour Encrypted).

3.4.5 choix du protocole d'authentification et d'autorisation

L'on peut utiliser le protocole SAML si l'application prend en charge SAML ; utiliser OpenID Connect pour les applications mobiles. S'il faut protéger les API ou créer une passerelle API... il est préferable d'utiliser OAuth2. [6]

En plus de la mise place d'un protocole authentification et d'autorisation, il est important d'implémenter un modèle de controle d'accès pour définir les privilèges.

3.5 Modèle de contrôle d'accès

Pour bien sécuriser un système d'information il est important d'implémenter un modèle de controle d'accès interessant pour accepter ou refuser l'accès aux ressources lors du processus d'autorisation.

3.5.1 Contrôle d'accès

Le principe du « **Contrôle d'accès** » consiste à accorder ou à refuser une demande provenant d'un sujet authentifié pour effectuer des actions sur des ressources. Les sujets peuvent par exemple être des utilisateurs, des processus ou des entités informatiques qui représentent les utilisateurs dans des systèmes et agissent en leurs noms. De même les ressources peuvent être des objets, des fichiers, des imprimantes ou des tables relationnelles dans des bases de données. Enfin les actions peuvent être des opérations possibles que nous pouvons effectuer dans des systèmes d'information ou des systèmes de gestion de base de données, par exemple écrire, lire, modifier, supprimer, etc. Dans certains systèmes, un accès complet est accordé à l'utilisateur qui s'est authentifié, mais la plupart des systèmes nécessitent un contrôle plus sophistiqué et complexe.

Définition : Le **contrôle d'accès** est un mécanisme grâce auquel un système autorise ou interdit les actions demandées par des sujets (entités actives) sur des ressources (entités passives).

L'**Authentification** est le processus visant à déterminer si une personne est celle qu'elle prétend être. L'authentification se fait généralement en utilisant un nom d'utilisateur et un mot de passe, mais cela peut inclure toute autre méthode permettant d'identifier une personne, comme l'utilisation des empreintes digitales, l'analyse rétinienne, la reconnaissance vocale ou d'autres méthodes biométriques. Dans les systèmes de sécurité, l'authentification est distincte de l'**Autorisation**, qui est un processus permettant de donner aux utilisateurs un accès aux ressources du système en fonction de leur identité, mais l'authentification ne dit rien sur les droits d'accès de l'utilisateur. Ainsi, le contrôle d'accès porte sur la façon dont l'organisation est structurée. Le développement d'un système de contrôle d'accès nécessite la définition d'un règlement selon lequel l'accès doit être contrôlé. Par ailleurs leur implémentation est basée sur trois niveaux principaux : politiques, modèles et mécanismes de contrôle d'accès [21].

- **Politique de contrôle d'accès :** elle définit les règles selon lesquelles le contrôle d'accès doit être régularisé. Les politiques sont une exigence de haut niveau qui précisent d'une part, la façon dont le contrôle d'accès est structuré, et d'autre part quel utilisateur peut effectuer telles actions sur telles ressources.
- **Modèle de Contrôle d'accès :** il fournit une représentation formelle des politiques de contrôle d'accès. La formalisation permet de vérifier les propriétés de sécurité fournies par le système de contrôle d'accès.

- **Mécanisme de contrôle d'accès :** il provient généralement du niveau bas de l'abstraction où il applique ces politiques de contrôle d'accès de haut niveau et traduit la demande d'un utilisateur sous forme d'une structure spécifique que le système a fournie.

Les trois niveaux sont indépendants, c'est ainsi que les politiques peuvent être analysées abstraitement sans référence au mécanisme d'application, mais seulement à travers le respect du modèle, qui à son tour, peut être pris en considération lorsque l'on montre la correction du modèle d'application.

Le *modèle de sécurité* est une spécification d'une politique de sécurité pour **la confidentialité, l'intégrité ou la disponibilité** qui n'explique pas le mécanisme particulier pour les atteindre. Il décrit uniquement les entités régies par la politique et énonce les règles qui constituent la politique. Plusieurs modèles ont été proposés pour encoder les politiques de contrôle d'accès.

3.5.2 Discretionary Access Control (DAC)

Dans Discretionary Access Control ou « **modèle de contrôle d'accès discrétionnaire** »[13] chaque objet ou ressource du système a un propriétaire (un sujet), lequel peut déterminer les privilèges d'accès à cet objet. Le sujet a un contrôle complet sur tous les objets qui lui appartiennent, il peut changer les permissions d'accès, transférer des objets authentifiés ou des accès à l'information à d'autres sujets. C'est pourquoi il est dit discrétionnaire. Les autorisations sont attribuées directement à des sujets en fonction de leur identité.

Avantage du DAC : Souplesse.

Inconvénients du DAC :

- Le DAC permet à l'utilisateur de décider des politiques du contrôle d'accès sur leurs ressources et ces politiques sont les politiques globales et par conséquent, le DAC a du mal à assurer la cohérence.
- La révocation de la permission est également complexe lorsque l'utilisateur quitte l'entreprise ou change de fonction
- Difficulté à administrer, car il est basé sur la confiance. Autrement dire, L'information peut être copiée d'un objet à un autre, de sorte que l'accès à une copie est possible même si le propriétaire initial ne donne pas accès à l'originale.
- Puisque les politiques du DAC peuvent être facilement modifiées par le propriétaire, un programme malveillant s'exécutant en son nom pourra aussi changer ces mêmes politiques, ce qui constitue une faiblesse de ce système. **Exemple « le Cheval de Troie »** qui est un programme informatique effectuant des opérations malicieuses à l'insu de l'utilisateur.

3.5.3 Mandatory access control (MAC)

Dans Mandatory Access Control ou « **modèle de contrôle d'accès obligatoire** »[8] , la politique de sécurité des systèmes d'information impose que les décisions de protection ne doivent pas être prises par le propriétaire des objets concernés, et lorsque ces décisions de protection doivent lui être imposées par le dit système. Dans ce type de contrôle d'accès les sujets ne peuvent pas intervenir dans l'attribution des droits d'accès. L'administrateur de la politique de sécurité définit l'utilisation des ressources et leur politique d'accès, ce qui ne peut être remplacé par les utilisateurs finaux et la politique décidera qui a le droit d'accéder aux programmes, fichiers etc.

Dans ce modèle, toutes les informations sont affectées à un niveau de sécurité, et chaque utilisateur est affecté à une habilitation de sécurité. Sujets et objets possèdent des habilitations et des étiquettes, respectivement, comme par exemple « confidentiel », « secret », et « très secret ». Il garantit que tous les utilisateurs n'ont accès qu'aux données pour lesquelles ils possèdent une habilitation égale ou supérieure à l'étiquette de l'objet.

Exemple : Selon la loi, les tribunaux peuvent accéder aux dossiers de conduite sans la permission des propriétaires. MAC est principalement utilisé dans un système où la priorité est basée sur confidentialité.

Avantages du MAC :

- MAC prévient l'altération non autorisée des objets
- MAC préserve la confidentialité et l'intégrité des informations, empêche certains types d'attaques comme Cheval de Troie. Donc plus sûr que le DAC

Inconvénients du MAC :

- manque de flexibilité
- la difficulté à mettre en œuvre et à programmer ce modèle.
- Ce contrôle d'accès est plus rigide que le contrôle d'accès discrétionnaire (DAC)

3.5.4 Role based access control (RBAC)

Pour fournir des droits d'accès à l'utilisateur, il est important de connaître la responsabilité de l'utilisateur assignée par l'organisation. Mais dans le DAC les droits d'utilisation des données jouent un rôle important, et la stratégie n'est meilleur ; et dans MAC, les utilisateurs doivent prendre des autorisations de sécurité et les objets ont besoin de classifications de sécurité. Role Based Access Control ou « **modèle de contrôle d'accès basé sur les rôles** » (RBAC)[12] essaie de réduire l'écart en combinant les contraintes organisationnelles forcées avec flexibilité des autorisations explicites.

Avec RBAC un **utilisateur** (un être humain, un agent autonome intelligent, etc.) peut être assigné à un ou plusieurs **rôles** (peuvent être définis comme un travail ou un titre de travail dans une organisation), et obtient ses **permissions** (l'approbation de l'accès à un ou plusieurs objets du système) à travers eux.

Un **rôle** désigne une entité intermédiaire entre les utilisateurs et les privilèges. Ces derniers ne sont plus associés, d'une façon directe aux utilisateurs mais à travers des rôles. Les deux relations **Rôle, Privilège** et **Utilisateur, Rôle** définissent les privilèges accordées à chaque utilisateur.

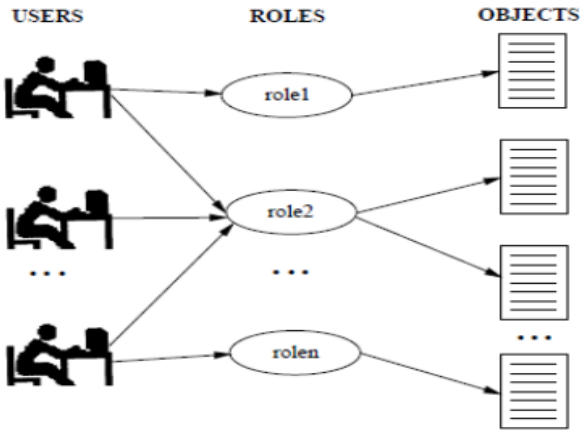


Figure 3.7 – Role-based access control [10]

Le modèle de référence RBAC et la spécification fonctionnelle sont organisés en trois composants qui sont le noyau, la hiérarchie et les contraintes.

Le noyau du RBAC

Il inclut la condition que l'affectation utilisateur-rôle et permission-rôle peut être many to many (n à n). Ainsi, le même utilisateur peut se voir assigner plusieurs rôles et un rôle peut être attribué à plusieurs utilisateurs. De même pour les permissions, une seule permission peut être assignée à plusieurs rôles et un seul rôle peut être assigné à plusieurs permissions.
Le noyau du RBAC inclut aussi le concept de session d'utilisateur qui permet l'activation et la désactivation sélective de rôles. Il permet finalement que les utilisateurs soient capables d'exercer les permissions de multiples rôles.

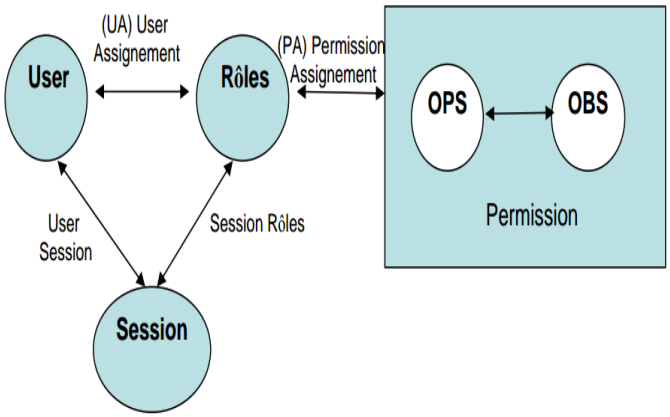


Figure 3.8 – Le noyau RBAC [16]

La hiérarchie du RBAC

La hiérarchie est mathématiquement un ordre partiel définissant la relation de supériorité entre les rôles, par laquelle les rôles supérieurs obtiennent les permissions des rôles junior, et les juniors obtiennent une adhésion des utilisateurs supérieurs.

Exemple par une représentation graphique des hiérarchies de rôles (figures suivant)

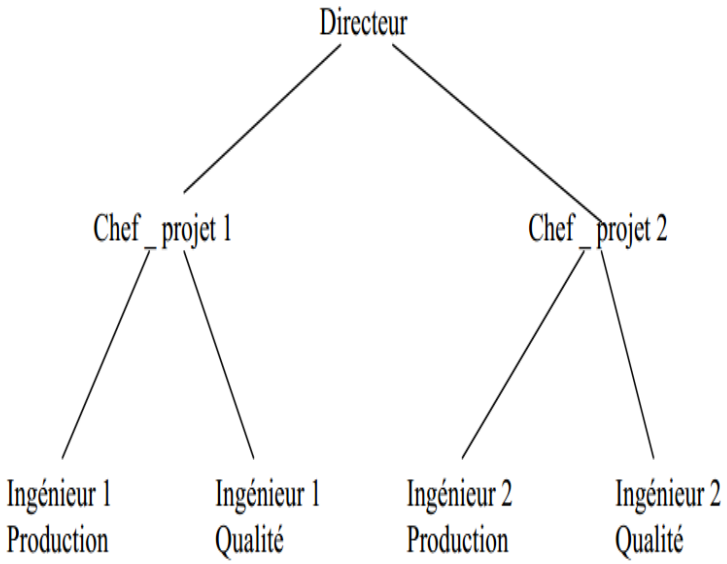


Figure 3.9 – Exemple de hiérarchie de role

Les contraintes du RBAC

Les contraintes constituent un aspect important du RBAC, et sont parfois la raison principale d'utilisation du RBAC comme moyen d'analyser le contrôle d'accès. Ces contraintes sont les contraintes prérequis, les contraintes de cardinalités, la séparation des devoirs.

- **La séparation des devoirs** vise à empêcher le conflit d'intérêt qui apparait lorsque des rôles en conflit sont associés à un même utilisateur (les rôles qui ne peuvent pas être attribués à un même utilisateur).
- **Les contraintes de prérequis** nécessitent qu'on ne puisse assigner un rôle à un utilisateur uniquement dans le cas où un rôle prérequis lui a précédemment été affecté.
- **La contrainte de cardinalité** permet de limiter le nombre de création des éléments du modèle RBAC.

Avantages du RBAC

- La capacité d'exprimer MAC et DAC
- RBAC aide les administrateurs à mieux contrôler les utilisateurs sans avoir à les associer explicitement aux ressources spécifiques ;
- Hiérarchie des rôles : De nombreuses applications ou organisations ont une hiérarchie de rôles.
- Moindre privilège : les rôles définissent le moindre privilège que l'utilisateur doit exécuter. Cela minimise les dégâts dus aux erreurs involontaires.
- Les rôles des utilisateurs sont révocables, sans nécessiter la gestion séparée des autorisations de contrôle d'accès pour chaque individu ; cela réduit les coûts administratifs et les coûts d'entretien.
- Séparation des tâches : Ce principe décrit qu'aucun utilisateur ne devrait avoir plus de droits afin qu'il puisse en abuser. Par exemple, la personne qui autorise un chèque de paie et qui peut les préparer ne devrait pas être la même personne.

Inconvénients du RBAC

- Définir les rôles dans un contexte différent est difficile et peut aboutir à une large définition du rôle. Parfois, il produit plus de rôles que d'utilisateurs.
- RBAC attribue les rôles de manière statique à son utilisateur, ce qui n'est pas le cas préféré dans un environnement dynamique. Ainsi, il est plus difficile de changer les droits d'accès de l'utilisateur sans changer le rôle de cet utilisateur. Par conséquent, le RBAC ne prend pas en charge des attributs dynamiques tels que l'heure de la journée sur lequel l'autorisation d'utilisateur est déterminée.

- Il maintient la relation entre les utilisateurs et ses rôles. Ça maintient aussi la relation entre les autorisations et les rôles. Par conséquent, pour implémenter le modèle RBAC, les rôles doivent être attribués à l'avance et il n'est pas possible de modifier les droits accès sans altérer les rôles.

3.5.5 Organization Based Access Control (OrBAC)

Le modèle OrBAC [15] a été proposé pour surmonter les limitations des modèles de contrôle d'accès existants (i.e. DAC, MAC, RBAC). Ce modèle introduit un niveau d'abstraction dans laquelle **les sujets sont abstraits en rôles, des actions sont abstraites en activités et les objets sont abstraits en vues**. Chaque politique de sécurité est définie par et pour une organisation et la spécification des politiques est paramétrée par l'organisation de sorte qu'elle est capable de gérer simultanément plusieurs politiques de sécurité associées aux différentes organisations.

Avec Or-BAC il y a donc trois entités abstraites :
L'entité **rôle** : utilisée pour structurer le lien entre sujets et organisations
Les **activités** : utilisées pour structurer le lien entre actions et organisations
Les **vues** : utilisée pour structurer le lien entre objets et organisations.

Or-BAC possède des **prédicats** pour attribuer des entités à l'organisation (rôle, activité, vue) mais aussi afin de formaliser les interactions entre les différentes entités qu'il traite. Les interactions (**Empower, Consider, Use**) existant entre les entités sont d'écrites sur le schéma suivant :

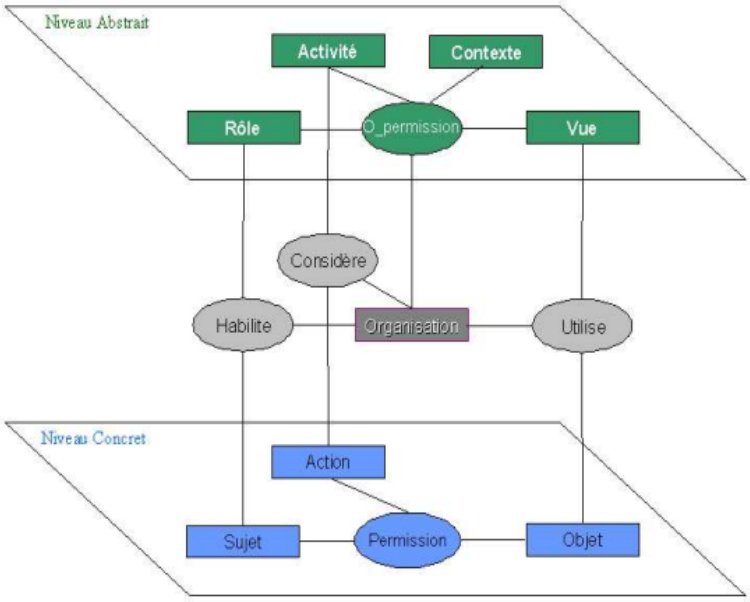


Figure 3.10 – Schéma présentant les interactions entre les entités du modèle OrBAC [9]

Explications :

- Relation **Empower**
  Une organisation habilite un sujet dans un certain rôle : **Empower(Organization, Subject, Role)**. Exemple : L'hôpital général habilite Jean à jouer le rôle de médecin : Empower(Hopital_general, Jean, Medecin).

- Relation **Consider**
  Une organisation considère qu'une action entre dans la réalisation d'une certaine activité : **Consider(Organization, Action, Activity)**. Exemple : Hopital_general considère qu'on peut effectuer une lecture dans une consultation, Empower(Hopital_general, read, Consultation).

- Relation **Use**
  Une organisation utilise un objet dans une certaine vue : **Use(Organization, Object, View)**. Exemple : Hopital_general utilise le fichier25.pdf dans Dossier_medical User(Hopital_general, fiche25.pdf, Dossier_medical).

- Notion de **contexte** : l'entité Contexte qui permet aux organisations de spécifier des autorisations de rôles pour effectuer des activités sur les vues dans une circonstance concrète, ce qui n'est pas réalisable dans RBAC. Les entités sujets, objets et actions sont définies par la relation Define(org, s, o, act, c) cela signifie que dans l'organisation org, le contexte c est vrai entre le sujet s, objet o et actions act.

- La relation **o_permission(org, r, a, v, c)** signifie que l'organisation org accorde au rôle r la permission de réaliser l'activité a sur la vue v dans le contexte c.

- la relation **permission(s, act, o)** a été introduite pour modéliser la permission concrète. Cela signifie que le sujet s est autorisé à effectuer l'action act sur l'objet o.

Avantages du OrBAC :

- Considération explicite de l'aspect « organisation »

- OrBAC peut améliorer la gestion de la politique de sécurité et réduire sa complexité grâce aux abstractions des entités

- possibilité de nuancer les autorisations qui n'est pas offertes par DAC, MAC, RBAC.

- Permet d'exprimer des permissions et des interdictions

Inconvénients du OrBAC

- il n'est pas adapté aux systèmes distribués et interopérables

- la règle de sécurité sous la forme Permission(org, r, v, a, c) ne permet pas de représenter des règles qui impliquent plusieurs organisations, par exemple dans le cas de la coopération entre plusieurs entreprises, un utilisateur appartenant à l'entreprise A souhaitant accéder aux ressources appartenant à ses partenaires (entreprises B, C, etc.). Par conséquence, OrBAC n'est pas très adapté pour modéliser un environnement dynamique et décentralisé.

- Difficile à implémenter.

3.5.6 Attributes Based Access control (ABAC)

C'est un **modèle de contrôle d'accès basé sur des attributs**. Dans ABAC, les autorisations d'accès aux objets ne sont pas directement données au sujet. Il utilise les attributs de sujet, d'objet et d'environnement pour fournir des autorisations :

- Attributs du sujet : qui sont associés avec un sujet (utilisateur, application, processus) qui en définissent l'identité et les caractéristiques. **Exemple** : Numéro d'identification, nom, profession, âge, ...

- Attributs d'objet : qui sont associés avec une ressource (webService, Fonction Système, donnée)

- Attributs d'environnement : qui décrivent l'environnement opérationnel, technique ou fonctionnel dans lequel le service est accédé. **Exemple** : la date, l'heure, niveau de menace, ...

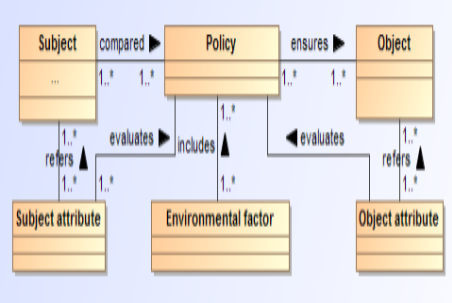


Figure 3.11 – MetaModel du ABAC [11]

Lorsque le sujet demande l'accès à l'objet, ses attributs sont évalués par la politique de contrôle d'accès ainsi que les conditions environnementales. Ensuite, on s'assure que le sujet peut accéder à l'objet demandé.
La définition des attributs et leur combinaison logique permet la définition souple de contraintes liées à l'autorisation sans avoir à définir des rôles complexes comme RBAC.
Exemple : Définition du contrôle d'accès à un film en fonction de l'âge.

Avantages du ABAC :

- ABAC résout le problème d'attribution de rôle d'utilisateur qui est présenté dans RBAC ; au lieu de se concentrer sur les rôles, il se concentre sur les attributs d'un utilisateur pour attribuer les droits d'accès.
- ABAC offre une plus grande flexibilité dans un environnement distribué partageable et dynamique où le nombre d'utilisateurs est très haut. Par conséquent, ABAC est un modèle très flexible pour le but administratif et est ainsi mieux que RBAC.
- Il fournit un stockage central pour les attributs des utilisateurs, il augmente l'interopérabilité et partage entre plusieurs fournisseurs de services pour décider des droits d'accès.

Inconvénients du ABAC :

- L'hétérogénéité de la complexité des informations utilisateur est augmentée, donc pour résoudre ce problème, il faut que la base de données centrale ait tous les attributs dans le même format.
- Si plusieurs organisations décident d'un ensemble commun d'attributs normalisés, il y aura un problème de faible expressivité dans la représentation des sujets et des objets. Ainsi, ABAC ne sera plus flexible.

3.6 Conclusion

Un modèle de contrôle d'accès nous permet d'exercer un contrôle ultraprécis sur l'accès des utilisateurs aux ressources. Au regard des avantages et inconvénients des différents modèle d'accès étudiés dans ce chapitre, nous constatons que **RBAC** est **idéal** pour des environnements multi-domaines lorsque les politiques sont exprimées en utilisant les hiérarchies de rôles et les contraintes ; de plus, RBAC peut être facilement incorporé dans les technologies courantes. Après le processus d'authentification avec des protocoles, il est important d'utiliser le modèle d'accès dans le processus d'autorisation d'une application (en l'occurence le chatbot) à accéder aux ressources des API d'un système d'informtion. Nous présenterons dans le chapitre suivant, l'analyse, la conception et la réalisation de notre projet.

Chapitre 4

ANALYSE, CONCEPTION ET REALISATION

4.1 Introduction

Après la présentation des concepts liés à notre projet, nous allons effectuer l'analyse et la conception de notre chatbot sécurisé d'entreprise. Pour cela, nous effectuerons premièrement le choix du cycle de vie de développement logiciel qui nous convient, deuxièmement, nous procéderons à l'analyse des besoins, troisièmement nous ferons la conception proprement dite du chatbot et enfin nous l'implémenterons.

4.2 Choix du cycle de vie de developpement logiciel

Le cycle de vie d'un logiciel en anglais software lifecycle désigne toutes les étapes du développement d'un logiciel de sa conception à sa disparition [18]. L'objectif d'un tel découpage est de permettre de définir des jalons intermédiaires permettant la validation de la conformité du logiciel avec les besoins exprimés. Le respect d'un cycle de vie permet de limiter les couts quant à la détection tardive d'erreurs. Le cycle de vie permet donc de détecter les erreurs plus tôt et de maitriser ainsi la qualité du logiciel, les délais de sa réalisation et les couts associés .

Le cycle de vie du logiciel comprend généralement les activités suivantes [18] :

- Définition des objectifs, consistant à définir la finalité du projet et son inscription dans une stratégie globale.
- Analyse des besoins et faisabilité, c'est-à-dire l'expression, le recueil et la formalisation des besoins du demandeur (le client) et de l'ensemble des contraintes.
- Conception générale : Il s'agit de l'élaboration des spécifications de l'architecture générale du logiciel.

- Conception détaillée, consistant à définir précisément chaque sous-ensemble du logiciel.
- Codage (Implémentation ou programmation), soit la traduction dans un langage de programmation des fonctionnalités définies lors de phases de conception.
- Tests unitaires, permettant de vérifier individuellement que chaque sous- ensemble du logiciel est implémenté conformément aux spécifications.
- Intégration, dont l'objectif est de s'assurer de l'interfaçage des différents éléments (modules) du logiciel. Elle fait l'objet de tests d'intégration consignés dans un document.
- Qualification c'est-à-dire la vérification de la conformité du logiciel aux spécifications initiales.
- Qualification c'est-à-dire la vérification de la conformité du logiciel aux spécifications initiales.
- Documentation, visant à produire les informations nécessaires pour l'utilisation du logiciel et pour des développements ultérieurs.
- Mise en production
- Maintenance, comprenant toutes les actions correctives (maintenance corrective) et évolutives (maintenance évolutive) sur le logiciel.

Il existe plusieurs modèles de cycle de vie à savoir le cycle en cascade, le cycle en V, le cycle en spirale, le cycle semi-itératif [1] mais pour notre projet, nous avons décidé de suivre le modèle du cycle en V.

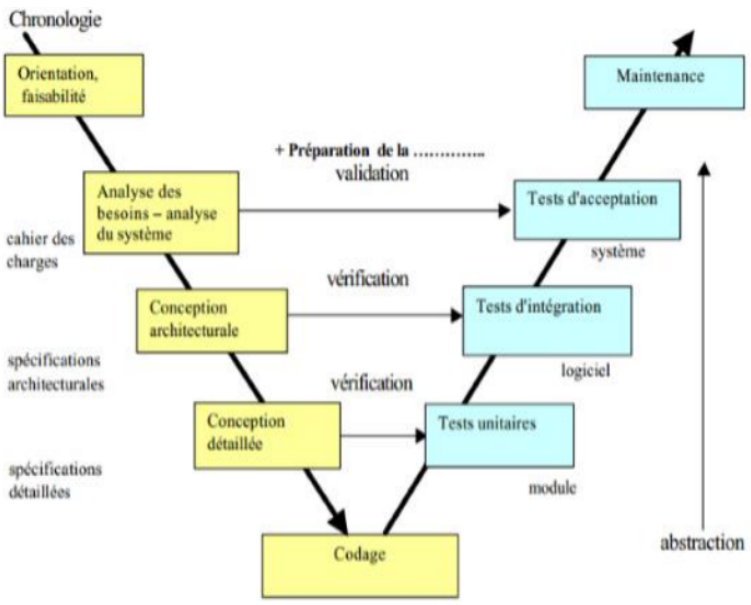


Figure 4.1 – Modèle de cycle de vie en V [1]

Nous avons choisi le cycle de vie en V parce qu'il tient d'avantage compte de la réalité et le processus de développement n'est pas réduit à un enchaînement de tâches séquentielles. Ce modèle montre que c'est en phase d'analyse des besoins que l'on se préoccupe des tests d'acceptation, c'est en phase de conception architecturale que l'on se préoccupe des tests d'intégration, c'est en phase de conception détaillée que l'on prépare les tests unitaires [1]. Le modèle de cycle de vie en V permet donc d'anticiper sur les phases ultérieures de développement du produit, d'éviter d'effectuer des choix et d'être bloquer après par la suite.

4.3 Orientation et faisabilité

Le projet intitulé « conception et mise en place d'un chatbot sécurisé d'entreprise » est né du fait que l'entreprise WORLD VOICE GROUP utilise beaucoup ses ressources humaines et perd beaucoup de temps pour gérer les préoccupations des clients.

La mise en place d'un tel système donc très bénéfique pour l'entreprise en ce sens qu'il permettra d'automatiser considérablement une bonne partie du service client.
La mise en place de cet outil permettra donc à WORLD VOICE GROUP d'avoir un robot qui :

- est disposé à converser avec les clients dans le but de donner une réponse à leurs préoccupations
- accède aux systèmes d'information de l'entreprise en cas de necessité pour donner une reponse fiable aux clients, avec **prise en compte de la sécurité**. La sécurité

permettra d'etre sûr que seul le robot poura accéder aux systèmes d'informations de l'entreprise.

4.4 Analyse des besoins

Ici nous recueillons les besoins du demandeur (le client) et l'ensemble des contraintes liés au système à mettre sur pied.

4.4.1 besoins fonctionnels

Ce sont les besoins spécifiant un comportement d'entrée / sortie du système (chatbot) :

- Lancer une conversation avec un client : Le chatbot lance la conversation automatiquement lorsqu'un client se connecte sur le site (Par exemple en disant « bonjour, je m'appelle Bob et vous ? »)
- Engager une conversation : Le chatbot engage la discussion en expliquant ce qu'il sait faire et ce qu'il peut apporter comme information.
- Afficher des réponses rapide en forme de bouton, des listes, sur lesquels le client peut cliquer, afin de guider la conversation et de s'assurer que le client reste dans le flux de la conversation
- Comprendre le problème du client : Une fois que le client est bien guidé dans la conversation, le chatbot peut facilement comprendre le problème du client.
- Répondre au client : Si le chatbot a la réponse dans sa base de connaissance statique, il peut tranquillement répondre au client.
- Extraire les informations du SI pour répondre au client : Pour mieux répondre à certaines préoccupations, il est nécessaire que le chatbot accède aux SI de l'entreprise via des APIs pour récupérer des données. Pour cela, le chatbot doit envoyer une requête HTTPS, ayant dans le corps de la requete, l'identifiant du robot, l'identifiant de l'utilisateur, le canal de communication, etc. L'API doit lui retourner les informations suivantes : l'identifiant de l'utilisateur, l'identifiant de l'interaction et le message (comportant des données du SI) qui sera affiché au client.
- Appeler un opérateur pour continuer la conversation avec le client sur l'interface chatbot : En cas d'incapacité de réponse ou de dialogue, le chatBot peut bloquer la conversation et demander à un opérateur de continuer.
- Demander l'adresse email du client pour envoyer à un conseiller client.
- Demander un numéro de téléphone whatsapp du client pour permettre à un opérateur de continuer la conversation avec le support whatsapp : Le contexte dans lequel se situe la conversation peut amener le chatbot à demander le numéro whatsapp du

client pour que le conseiller client continue la conversation à partir de whatsapp Business.

- Demander un numéro de téléphone du client pour émettre un appel à partir du Call Center : Si le problème du client semble être urgent, le chatbot récupère le numéro du client pour envoyer à un conseiller approprié. Ce dernier va continuer par appel téléphonique.

4.4.2 Besoins non fonctionnels

Il s'agit des besoins qui caractérisent le système. Ce sont des besoins en matière de performance, de type de matériel ou le type de conception. Dans notre cas on doit avoir :

- Un chatBot qui fonctionne 7jours/7 et 24h/24 sur différentes plateformes : Facebook Messenger, sites web de WORLD VOICE GROUP, Skype, Telegram, Slack, Line Messenger. Ceci permet d'être à tout moment à la disposition du maximum possible de clients qui aimeraient converser avec le support clients.
- ChatBot doit gérer le contexte et mémoriser des informations afin de ne pas les redemander.
- Mettre en devance d'autres liens de contact comme appel téléphonique, site web.
- La communication entre le chatbot et le SI doit être sécurisée. Pour cela, à chaque fois que le chatbot tente d'accéder aux données du SI, en envoyant une requete HTTPS, il doit etre identifié, authentifié et autorisé.

4.5 Budgetisation

Ici nous ressortons le budget d'un tel projet (tableau suivant).

Ressources	Noms	Quantité	Prix unitaire	Total
Ressources humaines	Ingénieur en génie logiciel	1	1 100 000 FCFA	1 100 000 FCFA
	Ingénieur en cryptographie et sécurité informatique	1	1 100 000 FCFA	1 100 000 FCFA
Ressources Matérielles	Serveur	1	550 000 FCFA	550 000 FCFA
	Imprévus		800 000 FCFA	800 000 FCFA
TOTAL				**3 550 000 FCFA**

4.6 Conception architecturale

Nous élaborons les spécifications de l'architecture générale de notre système. Notre système doit fonctionner en ligne avec architecture suivant.

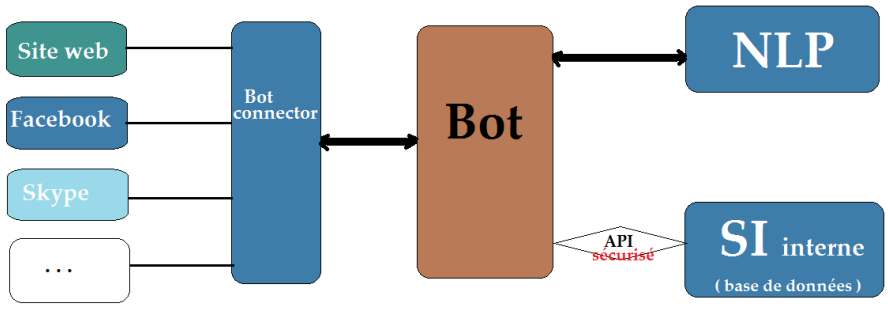


Figure 4.2 – Architecture du système

Nous avons dans cette architecture, le **bot (robot)** qui est chargé de converser avec les clients sur plusieurs **plate-formes (sites web, Facebook Messenger, Skype, ...)** par le biais du **BotConnector**. Ce bot se sert du **NLP (Natural Language Processing)** pour comprendre l'intention du client, c'est le moteur de traitement automatique de langue. Une fois que le bot a compris l'intention du client, il peut lui repondre en utilisant sa base de connaissance statique ou en accédant au **système d'information (SI interne) de l'entreprise** via des **API sécurisées**.

4.7 Conception detaillée

Pour effectuer la conception détaillée de notre système, nous utilisons le langage de modélisation UML (Unified Modeling Language).

4.7.1 présentation du langage UML

UML est un langage de modélisation qui permet d'exprimer et d'élaborer des modèles objets indépendamment du langage de programmation. L'UML a été conçu pour servir de support à une analyse basée sur le concept objet. Il se définit comme un langage de modélisation graphique et textuel destiné à décrire des besoins, à spécifier et documenter des systèmes, à esquisser des architectures logicielles, à concevoir des solutions et communiquer des points de vue. UML utilise des diagrammes pour modéliser un système. Il ne s'agit pas d'une simple notation graphique car les concepts transmis par un diagramme ont une sémantique.

4.7.2 Modélisation avec le langage UML

Pour la modélisation de notre système avec le langage UML, nous allons utiliser la plateforme en ligne **" lucidchart.com "**.

Le chatbot sera fait pour traiter plusieurs cas de problèmes précis des clients. Voici la description générale des scénarios de conversationnels entre **les acteurs (le client et le bot)** :

scénario nominal :

1. le client se connecte sur la plateforme de conversation
2. le bot engage la conversation
3. le client entre dans la conversation
4. le bot cherche l'intention du client en le guidant dans le flux de la conversation.
5. le bot comprend son intention
6. le bot répond directement ou accède aux données du SI via l'API avec un jeton d'accès (access token en anglais)
7. l'API sécurisé vérifie le jeton d'accès avant de donner l'autorisation d'accès aux données.
8. le bot recupère les données et donne une réponse valable au client
9. le client est satisfait.

scénario alternatif

1. le bot ne comprend pas la préoccupation du client, et recupère son adresse email pour alerter un conseiller client de l'entreprise.
2. le bot comprend la préoccupation du client, mais incapable de répondre, il récupère son numéro de téléphone pour alerter un conseiller approprié.
3. le jeton d'accès que le bot utilise n'est pas valide, le système le renvoie une réponse d'erreur

Diagramme de séquence :

Le diagramme de séquence montre les interactions entre les objets ; il montre en particulier les objets participant à l'interaction par leurs lignes de vie et les messages qu'ils s'échangent. Les diagrammes de séquences permettent de décrire comment les éléments du système interagissent entre eux et avec les acteurs. Les objets d'un système interagissent en s'échangeant des messages. Les acteurs interagissent avec le système au moyen d'interfaces homme-machine.

- **Diagramme de séquence pour "alerter un conseiller"**

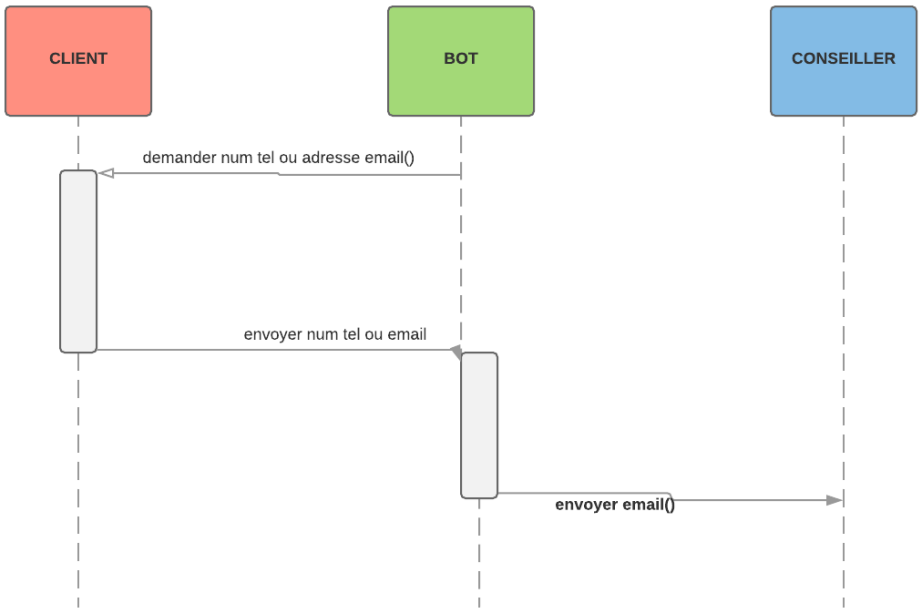


Figure 4.3 – Diagramme de séquence pour "alerter un conseiller"

- **Diagramme de séquence pour "accéder au SI"**

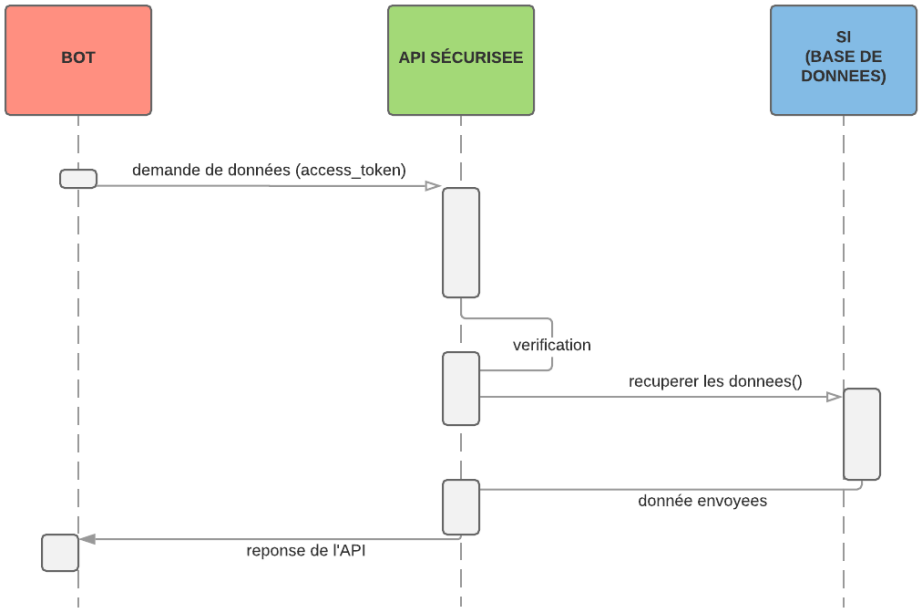


Figure 4.4 – Diagramme de séquence pour ”acceder au SI”

- **Diagramme de séquence générale pour ”conversation entre le client et le chatbot”**

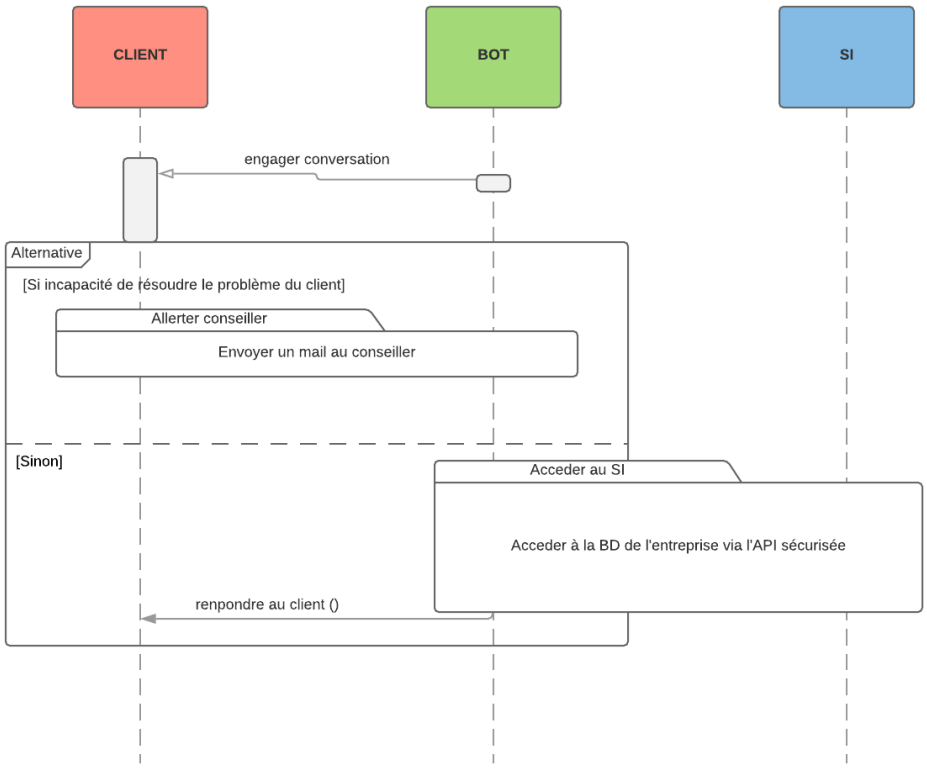


Figure 4.5 – Diagramme de séquence général pour ”conversation entre le client et le chatbot”

Diagramme de déploiement :

Le diagramme de déploiement vise à représenter les choix technologiques et matériels finaux sur lesquels seront positionnés les composants. Avec ce diagramme, on aborde donc la topologie du matériel.

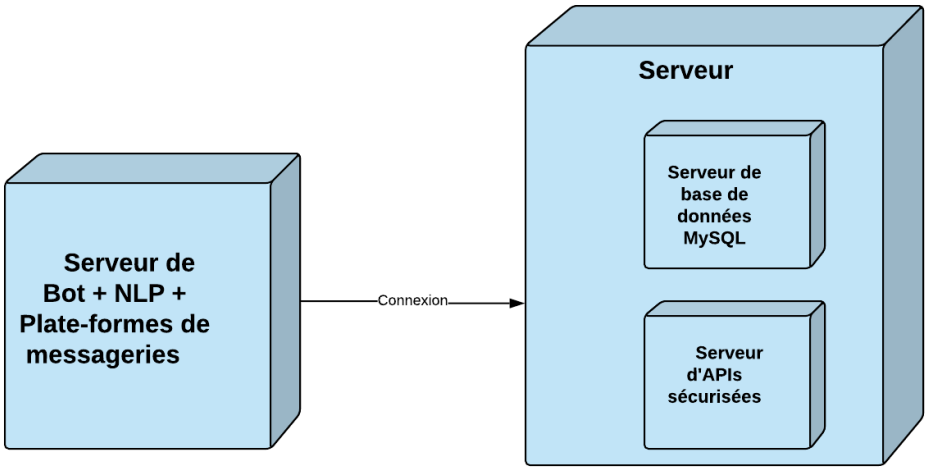


Figure 4.6 – Diagramme de déploiement

4.8 Implémentation

Il s'agit de l'implémentation ou programmation dans un langage de programmation des fonctionnalités définies lors de phases de conception. Nous utiliserons la technologie **Snatchbot** pour developper le bot et le framework **Spring-security** pour implémenter l'API sécurisée de l'entreprise.

4.8.1 Environnement de développement

Environnement matériel :

Nous avons utilisé côté matériel les éléments suivants :

- Un PC de marque HP avec pour caractéristiques ;
- Système d'exploitation : Windows 7 ;
- Type du système : Système d'exploitation 32 bits ;
- Processeur : Intel(R) Core(TM) i5 CPU @ 2.20GHz 2.20GHz ;
- RAM : 2,00 Go ;
- Disque Dur : 160 Go.

Logiciel

En ce qui concerne les logiciels, nous avons utilisé :

- **Xampp**
  Xampp est une plateforme de développement Web, permettant de faire fonctionner localement (sans se connecter à un serveur externe) des scripts PHP. Ce n'est pas en soi un logiciel, mais un environnement comprenant deux serveurs (un serveur web Apache et un serveur de bases de données MySQL), un interpréteur de script (PHP), ainsi qu'un logiciel d'administration SQL phpMyAdmin. Il permet donc d'installer en une seule fois tout le nécessaire au développement local PHP mais ce qui nous intéresse beaucoup plus c'est la base de données MySQL.

- **Eclipse**
  Eclipse est un environnement de développement intégré (EDI) placé en open source. il permet la prise en charge du langage java. Eclipse est l'editeur que l'entreprise WORLDVOICE GROUP utilise pour dévélopper des applications en J2EE (Java Entreprise Environment). Eclipse est disponible sous Windows, Linux, MAC OS.

- **Sublime Text 3**
  C'est un éditeur de texte qui propose beaucoup de éléments pour faciliter la tache au programmeur en php, html, javascript, css et bien d'autre. Nous l'utiliserons pour developper l'API sécurisé en PHP qui servira de **demo**. Par la suite, l'entreprise utilisera la meme logique pour developper en J2EE cette API sécurisée.

4.8.2 développement du bot

Il existe plusieurs technologie de mise en oeuvre d'un chatbot. Notre bot est développée en utilisant la technologie **SNATCHBOT**, elle est important pour nous dans la mesure où elle nous permet de couvrir tous les modules que nous avons présenté dans notre conception architectural. Elle dispose d'un moteur de traitement automatique (NLP) de langue basée sur l'algorithme de Bernoulli-Naive-Bayes, un botconnector qui permet d'intégrer plusieurs plate-formes de messageries instantanée (site web, facebook, skype, Slack, ...), d'une interface permettant de developper la connexion au système d'information de l'entreprise. Avec cette technologie, le développement du bot se fait en ligne dans la plate-forme **snatchbot.me** .

4.8.3 développement de l'API sécurisée

Pour ce qui est de la sécurisation des APIs dans notre cas, nous fonctionnerons avec les protocoles OAuth2 et HTTPS. Car le but ici est de nous assurer que seul le robot du chatbot poura acceder à l'API. Le système d'information de WORLD VOICE GROUP tient déjà compte du modèle d'accès RBAC. L'API sécurisée sera developpé en J2EE avec le framework **Spring-security**. Ce dernier est une technologie permettant d'intégrer presque tous les éléments de sécurité d'une application Web J2EE basée sur le framework **Spring**. Donc elle permet d'implémenter le protocole OAuth2 et le modèle de controle d'accès basé sur les rôles (RBAC).

Nous avons d'abord contruit un mini système d'information et developpé une API sécurisé avec le PHP, dans l'optique de produire une **Démonstration** (vu qu'un simple projet fait en PHP et MySQL peut trouver un hebergeur gratuit). Par la suite, l'entreprise utilisera la même logique pour developper ces APIs sécurisées en J2EE.

4.9 Résultat

4.10 Conclusion

CONCLUSION GENERALE ET PERSPECTIVES

Bibliographie

[1] 03 Juin 2019. https ://www.commentcamarche.net/cycle de vie d'un logiciel.

[2] 21 Avril 2019. https ://www.coe.int/fr/web/artificial-intelligence/what-is-ai.

[3] 23 Avril 2019. http ://www.artificiel.net/machine-learning-definition.

[4] Chiffres des réseaux sociaux au cameroun, 13 Avril 2019. https ://histoiresdecm.com/2018/01/08/chiffres-des-reseaux-sociaux-au-cameroun-en-2018/.

[5] modèle d'architecture d'un chatbot, 27 Avril 2019. https ://www.suricats-consulting.com/lab4us-chatbot/.

[6] Oauth vs. saml vs. openid connect, 03 Avril 2019. https ://www.gluu.org/resources/documents/articles/oauth-vs-saml-vs-openid-connect/.

[7] protocole saml, 29 Avril 2019. https ://www.orange-business.com/fr/blogs/securite/webtech/le-saml-le-petit-protocole-qui-monte-qui-monte.

[8] M. Benantar. *Mandatory-Access-Control Model*. Springer US, Boston, MA, 2006.

[9] Coma Céline. *Administration d'une politique de contrôle d'accès*. ENST Bretagne, Janvier 2005.

[10] Bokefode Jayant. D., Ubale Swapnaja A, Apte Sulabha S., and Modani Dattatray G. Analysis of dac mac rbac access control based models for security. *International Journal of Computer Applications (0975 – 8887)*, pages 8–11, October 2014.

[11] Sivalingapandi Darwin. *Comparison and Alignment of Access Control Models*. PhD thesis, UNIVERSITY OF TARTU, 2017.

[12] F. David and K. Richard. Role-based access controls. *Proceedings of 15th NIST-NCSC National Computer Security Conference*, page 563, 1992.

[13] R. P. Gallagher. *A guide to understanding discretionary access control intrusted systems*. 1987.

[14] Hugo HOUYEZ. Qu'est-ce qu'une api rest ou restful ?, 10 Avril 2019. https ://www.supinfo.com/articles/single/5642-qu-est-ce-qu-une-api-rest-restful.

[15] Baida R. Balbiani P. Benferhat S. Cuppens F. Deswarte Y. Miege A. Saurel C. Kalam, A. and G. Trouessin. *Organization based access control*. 2003.

[16] N. E. KHELIFA. *Integration du modèle de controle d'accès RBAC dans les diagrammes UML*. PhD thesis, Université d'Oran.

[17] M. MABYRE. Authentification et habilitations avec openid connect, oauth 2 et jwt, 23 Avril 2019. https ://meritis.fr/techno-archi/openid-connect/.

[18] Robert Ogor. *Modelisation avec UML*. ENST Bretagne, Mai 2003.

[19] Jean-François Pillou. sécurité informatique, 05 Avril 2019. https ://www.commentcamarche.net/contents/1033-introduction-a-la-securite-informatique.

[20] Romain Mainnevret Quentin Peltier, Valentin Bigarnet. *Analyse des plateformes de création de chatbots*. Le Lab Lawbydesign, 2018.

[21] P. Samarati and S. de Vimercati. *Accesscontrol :Policies,models,and mechanisms*. InFocardi, R.and Gorrieri, R., editors, Foundations of Security Analysis and Design, volume 2171 of Lecture Notes in Computer Science, pages 137–196. Springer Berlin Heidelberg, 2001.

[20] Worachet UTTHA. (26 septembre 2016). Étude des politiques de sécurité pour les applications distribuées. thèse de doctorat en informatique. pages 21-28. Université d'AIX-MARSEILLE.

[21] documentation snatchbot. https ://support.snatchbot.me/docs vu 03 mai 2019

[22] Spring Security 4 Tutorial. http ://websystique.com/spring-security-tutorial/ vu 29 mars 2019

ANNEXES