

**UNIVERSITÉ DE SAAD DAHLAD DE BLIDA -1-**

**Faculté des Sciences**

Département d'Informatique

## **MÉMOIRE DE MAGISTER**

Spécialité : Informatique Répartie et Mobile (IRM)

Déploiement d'une plateforme web sécurisée pour le suivi de  
la rétinopathie diabétique

Par

**FEKHAR Mohamed Amine**

Devant le jury composé de :

H.ABED	Professeur, U. de Blida-1-	Présidente
N.BOUSTIA	Maître de Conférences(A), U. de Blida-1-	Examinatrice
N.GHOUALMI	Professeur, U. d'Annaba	Examinatrice
N.BENBLIDIA	Professeur, U. de Blida-1-	Promotrice
S.OUKID	Maître de Conférences(A), U. de Blida-1-	Co-Promotrice

Blida, Janvier 2016

# RÉSUMÉ

L'émergence du Web 2.0, avec les avancées récentes dans les technologies de l'information et de la communication, et le succès des réseaux sociaux encouragent les recherches dans le domaine de la télémédecine et le développement de plateformes de santé sophistiquées pour les personnels de santé on révolutionne l'accès, l'intégration et le partage de l'information médicale. De nombreux systèmes se développent afin de répondre aux nouveaux impératifs tels que l'instantanéité (« tout de suite ») et l'ubiquité (« partout »). Ces nouveaux paradigmes de réseaux sociaux gagnent rapidement l'attention des utilisateurs des plateformes, car ils ont le potentiel d'étendre les modèles traditionnels d'exploitation.

D'autre part, le modèle du Cloud Computing permet la sous-traitance des ressources informatiques. Il crée de nouvelles possibilités à explorer leurs avantages et faciliter le partage et l'accès à distance à l'information médicale, n'importe où et n'importe quand. De plus, les plates-formes du Cloud offrent l'évolutivité et la fiabilité ce qui répond bien aux exigences des besoins de la Télémédecine.

Le but de ce travail est de développer et déployer une plateforme web sécurisée pour le suivi de la rétinopathie diabétique. Pour cela, nous effectuons une étude des technologies récentes qui seront appliquées à la Plateforme, en passant par la définition des architectures possibles, l'étude des techniques actuelles de travail collaboratif sous un paradigme de réseaux sociaux. Tout cela sera à son tour intégré dans un contexte de Cloud Computing pour soutenir la plateforme de télé-ophtalmologie et ses scénarios de travail. On a ciblé le domaine de l'imagerie médicale en prenant en compte l'interopérabilité avec les systèmes actuels. Le travail a abouti à une solution de Télé -ophtalmologie qui permet aux utilisateurs d'accélérer et améliorer les processus décisionnels dans un environnement collaboratif sécurisé basé sur le web.

**Mots clés :** Télé –ophtalmologie ; travail collaboratif ; Cloud Computing ; DICOM

# ABSTRACT

With the improvement of network devices and the web 2.0 capabilities telemedicine is evolving from a necessity for distant health centers to a must for any decent health structure. Many telemedicine frameworks help saving, retrieving and sharing medical information but fail in bringing ubiquity and responsiveness.

We think that telemedicine frameworks based on cloud technology are the main alternatives : they are highly scalable and reliable, any information stored can easily be accessed anywhere anytime and is highly available for health personnel.

Our aim is to develop and deploy a secure telemedicine solution for diabetic retinopathy .After studying different candidate designs and paradigms and inspired by collaborative communication in social network, we will propose a solution based on Cloud technology targeting medical imaging in tele-ophthalmology and helping its users to accelerate and improve decision processed in distributed and secure environments.

**Keywords:** Tele-ophtalmologie; collaborative work; Cloud Computing; DICOM

## ملخص

إن التطور والتقدم الحالي في تكنولوجيا المعلومات والاتصالات مع النجاح الباهر للشبكات الإجتماعية يفتح آفاق البحث في مجال التطبيق عن بعد ويشجع على تطوير منصات طبية جديدة ومعقدة من أجل تحسين الخدمة لمهنيي الصحة بتطوير طريقة الوصول للمعلومة الطبية ومشاركتها وادخالها. مثل ما هو الحال في عدة مجالات أخرى يتم تطوير عدة أنظمة لتتماشى مع الميزات الحديثة مثل الولوج للمعلومة في أي وقت ومن أي مكان، هذه الميزات الحديثة التي تشكل صيغة أساسية للشبكات الإجتماعية تسحتوذ سريعاً على اهتمام مستعملي المنصات الطبية لأنها لها القدرة على تطوير النماذج التقليدية المستغلة.

من جهة أخرى يسمح نموذج الحوسبة السحابية بمنح تسيير وسائل الإعلام الآلي لطرف آخر مع إحداث إمكانيات جديدة لها إيجابيات مهمة، كذلك تسمح الحوسبة السحابية بتسهيل مشاركة المعلومات الطبية والوصول إليها عن بعد في أي زمان ومن أي مكان. إضافة إلى ذلك، لدى المنصات السحابية القدرة على التطور وضمان الخدمة مما يستجيب لشروط ومتطلبات منصات التطبيق عن بعد.

إن هدف هذا البحث هو تطوير منصة وab آمنة من أجل متابعة اعتلال الشبكية السكري، من أجل تحقيق هذا الهدف سيتم القيام بدراسة للتكنولوجيات الحديثة التي سيتم تطبيقها على المنصة، كذلك تبين الهندسات المحتملة للنظام و دراسة تقنيات العمل التعاوني تحت تأثير خاصيات الشبكات الإجتماعية. كل هذه الأنظمة والتقنيات سيتم إدراجها في الحوسبة السحابية من أجل دعم منصة تطبيق العيون عن بعد و طرق عملها. لقد تركز عملنا حول مجال الصور الطبية مع الأخذ بعين الإعتبار التوافق مع الانظمة الحالية. في الأخير توصلنا بهذا العمل إلى منصة للتطبيق عن بعد تتيح لمستعمليها تسريع وتطوير طرق إتخاذ القرار في بيئة عمل تعاوني آمن في الواب.

كلمات مفتاحية: التطبيق عن بعد، العمل التعاوني، الحوسبة السحابية، DICOM

## DÉDICACES

*A mes parents, que Dieu les protège.*

*A ma très chère femme avec qui je continue de partager les joies et les chagrins.*

*A mon fils Riadh Mohamed Ali mon plus grand trésor dont le doux sourire  
et l'amour m'ont donnée l'énergie de faire ce travail*

*A tous ceux qui ont prié Dieu pour ma réussite.*

*Mohamed Amine*

# REMERCIEMENTS

Qu'il me soit permis de présenter ici mes remerciements à tout un petit monde de personnes qui ont rendu possible ce travail et qui ont contribué à son élaboration sous quelque forme que ce soit.

Je tiens tout d'abord à dire ma reconnaissance envers Madame BENBLIDIA Nadjia Professeur à l'USDB qui, malgré ses nombreuses occupations, a accepté sans réserve, de diriger ce mémoire. Elle s'y est grandement impliquée par ses directives, ses remarques et suggestions, mais aussi par ses encouragements dans les moments clés de son élaboration. Je tiens à la remercier aussi pour cette liberté qu'elle m'a permise, sans laquelle le chercheur ne saurait affirmer sa manière de penser et de procéder, sa manière d'être, bref toute sa personnalité.

Je tiens à exprimer ma reconnaissance à Mme. OUKID Saliha, Maître de conférences à l'USDB, d'avoir accepté de codiriger ce travail et pour l'aide indéniable qu'elle m'a apportée.

Je remercie également M. KACIMI-EL-HASSANI Chams EI-Dine mon ami et ancien binôme, pour sa lecture attentive du manuscrit, ses commentaires et corrections.

Je ne manquerai pas non plus de dire un grand merci aux membres du jury qui ont accepté, sans réserve aucune, d'évaluer ce mémoire à sa juste valeur, et de me faire part de leur remarques sûrement pertinentes qui, avec un peu de recul, contribueront, sans nul doute, au perfectionnement du présent travail.

Je remercie également tous mes amis et mes collègues pour leurs encouragements, leurs aides et leur support.

# TABLE DES MATIÈRES

<b>RÉSUMÉ</b>	
<b>ABSTRACT</b>	
ملخص	
<b>DÉDICACES</b>	
<b>REMERCIEMENTS</b>	
<b>LISTE DES FIGURES</b> .....	<b>9</b>
<b>LISTE DES TABLEAUX</b> .....	<b>11</b>
<b>INTRODUCTION GÉNÉRALE</b> .....	<b>12</b>
<b>CHAPITRE 1 ÉTAT DE L'ART</b> .....	<b>16</b>
1.1 L'aspect Informatique médicale .....	16
1.1.1 Télé ophtalmologie.....	16
1.1.2 La télémédecine .....	16
1.1.3 Téléradiologie .....	17
1.1.4 Système d'archivage et de transmission d'images (PACS) .....	21
1.1.5 DICOM.....	25
1.2 Cloud Computing.....	35
1.2.1 Définition.....	35
1.2.2 Caractéristiques du Cloud Computing.....	35
1.2.3 Les Modèles de Déploiement.....	39
1.2.4 La couche service dans le Cloud .....	41
1.3 Télé ophtalmologie collaborative.....	43
1.3.1 Étude des architectures possibles.....	44
1.3.2 Technologie de collaboration .....	48
1.3.3 La messagerie instantanée et présence.....	49
1.3.4 Flux XML .....	54
1.3.4.1 Stanzas XMPP .....	55
1.3.5 L'Extensibilité.....	61
<b>CHAPITRE 2 ARCHITECTURE ET APPROCHES DE LA PLATEFORME</b> .....	<b>63</b>
2.1 Les exigences du système .....	63

2.2	Spécifications fonctionnelles.....	65
2.3	Les besoins non fonctionnels .....	67
2.3.1	Accessibilité et disponibilité des informations.....	67
2.3.2	Sécurité de l'information.....	68
2.3.3	Fiabilité du service .....	68
2.3.4	L'évolutivité du service .....	68
2.4	Modélisation du système .....	69
2.4.1	Le Modèle fonctionnel.....	69
2.4.2	Modèle de données .....	79
2.4.3	Architecture.....	82
	<b>CHAPITRE 3 CONCEPTION ET IMPLEMENTATION .....</b>	<b>84</b>
3.1	DzTeleOph –SaaS- .....	84
3.1.1	Middleware Orienté Message : TeleOph Protocole de Communication Extensible – TOPCE .....	85
3.1.2	Serveur XMPP - Openfire .....	98
3.1.3	Dossier Distant Personnel - DDP .....	100
3.1.4	Organisation du cœur du Système.....	101
3.2	Passerelle du Système de Gestion d'information médicale.....	109
3.2.1	Architecture de la passerelle PACS .....	110
3.2.2	Interface XMPP.....	110
3.2.3	Interface DICOM.....	113
3.2.4	Synchroniser les dossiers .....	115
3.2.5	L'anonymisation des images.....	115
3.3	Client Web2.0 de la plateforme DzTeleOph.....	117
3.3.1	JavaScript du Client XMPP .....	119
3.3.2	Html5 et JavaScript de l'Interface de l'utilisateur .....	121
	<b>CHAPITRE 4 : EXPERIMENTATION ET RESULTATS .....</b>	<b>126</b>
	<b>CONCLUSION GÉNÉRALE ET PERSPECTIVES .....</b>	<b>135</b>
	<b>RÉFÉRENCES BIBLIOGRAPHIQUES .....</b>	<b>138</b>

## LISTE DES FIGURES

Figure 1.1 : Composants du Système PACS.....	22
Figure 1.2: Structure de la classe SOP.....	26
Figure 1.3: La hiérarchie de l'information DICOM.....	27
Figure 1.4 : La Structure d'un objet DICOM.....	29
Figure 1.5 : Elément de donné d'un objet DICOM.....	30
Figure 1.6 : Modèle SCU-SCP DICOM.....	31
Figure 1.7 : Services et les entités du réseau DICOM .....	32
Figure 1.8: C-Store DICOM.....	32
Figure 1.9: Résumé du service C-Store dans le service C-Get.....	33
Figure 1.10: C-Move avec trois entités.....	34
Figure 1.11: Exemple DICOM MWL.....	34
Figure 1.12 : Architecture 01.....	45
Figure 1.13 : Architecture 02.....	46
Figure 1.14 : Architecture 03.....	46
Figure 1.15 : Architecture 04.....	48
Figure 2.1: Cas d'utilisation du système.....	67
Figure 2.2: Infrastructure XMPP évolutive avec de multiples serveurs.....	71
Figure 2.3: Schéma plugin de stockage.....	72
Figure 2.4: Sauvegarder les données des images médicales dans un Cloud.....	73
Figure 2.5: Déploiement dans un Cloud public le stockage et le service.....	74
Figure 2.6: Organigramme du stockage des images médicales dans le Cloud.....	75
Figure 2.7: Diagramme d'un dossier partagé.....	76
Figure 2.8: Contournement du pare-feu en utilisant le serveur XMPP comme relais Cloud.....	77
Figure 2.9: Diagramme du filtre de paquets de la passerelle relais.....	78
Figure 2.10: Diagramme d'exécution d'une action à distance.....	79
Figure 2.11: Diagramme modèle de données du système de la Plateforme.....	80
Figure 2.12: modèle physique des données des fichiers partagés et la propriété des fichiers.....	81
Figure 2.13: modèle physique de données pour les règles de filtre des paquets.....	82
Figure 2.14: Schémas de l'architecture du Système.....	83
Figure 3.1: Diagramme de séquence gestion du répertoire à distance.....	86
Figure 3.2 : Diagramme du Dossier Distant Personnel.....	101

Figure 3.3: Diagramme Gestionnaire d'archive.....	103
Figure 3.4: Implémentation des plugins d'archive à distante.....	103
Figure 3.5: Diagramme de séquence de l'extension gestionnaire des paquets.....	104
Figure 3.6: Diagramme de séquence chargement des fichiers vers le stockage Cloud.....	106
Figure 3.7: L'authentification de chargement des fichiers.....	107
Figure 3.8: Diagramme de séquence de téléchargement d'un fichier.....	108
Figure 3.9: Diagramme de séquence de Téléchargement d'un fichier Partager.....	108
Figure 3.10: Diagramme composants de la passerelle PACS.....	110
Figure 3.11: diagramme de classes montre les entités qui interagissent avec la couche de XMPP. JabberSmackAPI.....	112
Figure 3.12: Diagramme de classe de la couche DICOM.....	114
Figure 3.13: Diagramme du service DICOM Query / Retrieve.....	114
Figure 3.14: Diagramme de synchronisation d'un dossier.....	115
Figure 3.15: Mise à jour des règles d'anonymisation via le mécanisme publish-subscribe.....	116
Figure 3.16: l'envoi des images avec des fonctionnalités d'anonymisation.....	117
Figure 3.17: Modèle de conception MVC de l'application Web.....	118
Figure 3.18: Prototype de l'application Web.....	119
Figure 3.19: Organisation du client JavaScript de XMPP.....	120
Figure 3.20: établissement d'une session WebRTC.....	125
Figure 4. 1: Dossier distant personnel dans le Cloud.....	127
Figure 4.2: Partage des fichiers avec des collègues.....	127
Figure 4.3: Accès aux fichiers partagé.....	128
Figure 4.4: les outils de collaboration.....	129
Figure 4.5: Synchronisation des données du Cloud avec une passerelle.....	130
Figure 4.6: Visualiseur d'images DICOM intégré.....	131
Figure 4.7: La recherche dans le PACS distant à partir de l'application web.....	132
Figure 4.8: Sélectionnez et Récupérer des images à partir du PACS distant.....	132
Figure 4.9: Partager PACS avec des collègues.....	133
Figure 4.10: L'interface graphique de la passerelle PACS.....	134
Figure 4.11: L'interface graphique de la configuration de la passerelle PACS.....	134

## LISTE DES TABLEAUX

Tableau 1.1 : Services DIMSE .....	31
Tableau 1.2 : Les types de XMPP Presence utilisés dans l'abonnement à la présence.....	58
Tableau 1.3 : Valeur d'un XMPP Presence avancée.....	59
Tableau 1.4 : Les types de Stanza Presence de XMPP avec des fonctions spécifique.....	60
Tableau 1.5 : Les types de Stanza I/Q .....	61
Tableau 2.1 : Liste des besoins fonctionnels .....	66
Tableau 3.1 : Le format de message pour la gestion des passerelles PACS distants .....	97

## INTRODUCTION GÉNÉRALE

L'informatique et la technologie de télécommunication ont le potentiel de transformer la santé comme ils ont transformé la banque, le transport, la fabrication, l'agriculture et les industries de vente au détail. L'adoption de ces technologies par le secteur de la santé peut aider la construction d'un réseau de santé plus efficace, comme conséquence de cette adoption dans l'établissement de soins de santé, la télémédecine a émergé comme un moyen permettant de fournir des services de soins de santé à distance qui permettent aux médecins, les patients, les chercheurs et les responsables de la santé publique de fournir des services de soins de santé à distance et d'accéder aux données des patients pour de nombreuses utilisations. En raison de l'évolution rapide de l'informatique et les télécommunications et la réduction de leur prix, la télémédecine est largement explorée par de nombreux chercheurs, dans le but de tirer le meilleur profit de la technologie contemporaine afin d'améliorer les services offerts par les établissements de soins de santé. Cette formidable évolution de la technologie crée des opportunités pour améliorer et même créer de nouveaux services de télémédecine. Par ailleurs, cette différence entre la télémédecine et la médecine classique va disparaître en raison de la généralisation de la télémédecine. Les applications de la technologie continueront d'évoluer rapidement et deviendront plus intégrées et ubiquitaires. Ce progrès sera fortement impacté par des innovations dans les domaines des réseaux mobiles et du haut débit. Ces changements favorisent la portabilité et l'intégration transparente de la technologie dans la vie quotidienne. Avec l'introduction des technologies au service de la médecine, la gestion de toutes les données, qui sont générées par les

établissements de santé et la façon dont l'information est échangée et transmise, est devenue plus facile et efficace. Dans le domaine de l'imagerie médicale, il est possible d'identifier certains systèmes informatiques qui ont été développés pour augmenter l'efficacité de la gestion et la transmission des données des images médicales à l'intérieur d'un établissement de soins de santé. Le système PACS a été développé pour faire face à l'énorme quantité de données, qui peut atteindre des téraoctets d'informations générées par les établissements de soins de santé. Les systèmes PACS sont largement adoptés par les institutions de santé pour mettre en œuvre un système informatique qui peut être géré efficacement, permettant la normalisation de l'acquisition de données, le stockage et la transmission [1].

Les développements déjà réalisés dans le domaine de l'informatique médicale montrent que l'application des technologies informatiques dans le domaine médical est une valeur ajoutée. Il est donc important de continuer à faire des recherches sur l'application des nouvelles technologies émergentes dans ce domaine. Ces recherches peuvent être très importantes pour améliorer les services actuels.

L'objectif de notre travail de recherche concerne les nouvelles technologies afin de créer un nouveau service de télémédecine basé sur le Cloud Computing. En outre, ce travail permettra de répondre à la question "comment l'utilisation des technologies du Cloud Computing peut être efficace et être un atout à la télémédecine" ? En profitant de ses fonctionnalités comme, par exemple, l'externalisation du calcul et du stockage, l'accès à l'information, la disponibilité, l'évolutivité, etc. Il sera proposé un système basé sur le Cloud pour fournir un service qui peut être utilisé par les médecins pour faire du télétravail à travers un environnement collaboratif inspiré des réseaux sociaux.

Le service permettra aux médecins de partager des images médicales avec des collègues ; il permettra aux médecins d'interagir les uns avec les autres dans l'examen d'une étude donnée, créant ainsi un canal qui peut être utilisé pour obtenir une deuxième opinion à un examen ou pour interagir avec un expert dans un domaine donné. En outre, le service résultat peut être utilisé pour fournir un soutien à distance aux centres de soins de santé qui ne disposent pas de médecins

spécialisés, à des fins éducatives ou intégrer des sources d'information médicale distribuée.

Pour implémenter un SaaS (software as a service) basé sur le Cloud, il est nécessaire d'analyser la façon dont les infrastructures Cloud sont déployées. Ainsi, déployer une infrastructure dans le Cloud ne signifie pas que cette infrastructure hérite des caractéristiques du Cloud, comme, par exemple, l'évolutivité. Évidemment, pour mettre en œuvre un système informatique médical, il est nécessaire de prendre soin de certaines questions liées à la sécurité et/ou confidentialité des données, parce que ces types de systèmes traitent des données sensibles et les données doivent être disponibles à tout moment. En outre, l'accès aux données à partir de l'intérieur d'un établissement de soins de santé peut apporter quelques questions à résoudre sur les pare-feu. En général, l'objectif de ce mémoire est de créer un système distribué entièrement intégré basé sur le Cloud qui aborde les problèmes soulevés et compatible avec le Web 2.0.

En conséquence, les principaux objectifs de ce travail sont les suivants:

- ❖ l'étude de la télémédecine.
- ❖ l'étude des principes fondamentaux du Cloud Computing.
- ❖ l'étude et le développement d'un système basé sur Cloud pour la télé-ophtalmologie.
- ❖ l'expérimentation et la validation du système.

Il est donc prévu de réaliser un système exploitant les nouvelles technologies, avec un cycle de vie bien défini. Il doit être fourni en tant que service dans le Cloud pour servir les médecins, permettant l'amélioration du flux de travail et le flux de données dans le télétravail.

Pour assurer une meilleure présentation du travail effectué et garantir la clarté de ce mémoire, outre cette introduction générale, ce manuscrit se compose de quatre chapitres et une conclusion générale. Chacun met en évidence une contribution particulière du travail :

- Dans le premier chapitre de ce mémoire, nous présentons un état de l'art sur les technologies qui sont actuellement des solutions d'appui pour la télémédecine.

Après une brève introduction à la télémédecine, on se concentrera sur les services de Télé radiologie, leurs domaines d'application, ainsi que la façon avec laquelle les nouvelles technologies améliorent ces services. En outre, on présente le PACS (Picture Archiving and Communication System), où est montrée son importance pour les services de télémédecine. En plus, ce chapitre présente également le paradigme Cloud Computing, décrivant certains modèles de services disponibles dans le Cloud. Enfin, sera décrit la façon avec laquelle XMPP (Extensible Messaging and Presence Protocol) peut être utile en tant que middleware orienté message pour la création d'un système de collaboration pour une Télé ophtalmologie collaborative.

- Dans le second chapitre, nous décrivons les exigences du système et les modèles y afférents. Il y a d'abord une identification de toutes les exigences fonctionnelles et non fonctionnelles définies pour le système à développer et à partir de ces exigences, une proposition d'implémentation est faite. La présentation du modèle de données avec son architecture seront également présentées.
- Dans le troisième chapitre, nous expliquons les décisions d'implémentation. Ainsi, nous présentons un système basé sur plugin qui permet à la plateforme de fonctionner comme un SaaS (Software-as-a-Service). De plus, il y a la présentation de middleware orienté messages qui soutient le système et comment a été mis en œuvre la passerelle de la plateforme avec le PACS pour permettre la création d'une interface entre le réseau XMPP et le réseau PACS. Après, on a expliqué le processus de mise en œuvre de la plateforme Télé Ophtalmologie Web2.0.
- Dans le quatrième chapitre, nous décrivons les résultats obtenus ainsi que l'implémentation du système.
- Une conclusion générale, donnée à la fin du manuscrit, résume la contribution du travail effectué. Il discute également les principaux résultats obtenus, la force et la faiblesse du système proposé et fixe les perspectives envisageables d'amélioration.

# CHAPITRE 1

## ÉTAT DE L'ART

Dans ce chapitre, nous allons présenter les technologies qui fournissent des solutions d'appui pour la télémédecine ; ces solutions permettent le stockage et l'accès aux images médicales. Suite à un aperçu rapide de la télémédecine contemporaine nous envisagerons pour notre solution une architecture de Cloud orientée services et un protocole collaboratif.

### 1.1 L'aspect Informatique médicale

#### 1.1.1 Télé ophtalmologie

Télé ophtalmologie est une branche de la télémédecine qui offre des soins de l'œil à travers un équipement médical numérique et des technologies de télécommunications [2]. La télécommunication se fait en mode différé c.-à-d. stockage-émission ou bien en temps réel permettant aux médecins de faire un diagnostic à des patients dans des régions éloignées [3], la surveillance; ainsi que la formation à distance [4].

#### 1.1.2 La télémédecine

La télémédecine, un terme inventé dans les années 1970 [5], qui signifie littéralement "la guérison à distance". Elle a pour principe l'utilisation de l'informatique et des technologies de communication afin de permettre aux médecins d'examiner, de surveiller et de traiter les patients qui sont éloignés, et n'ont pas la possibilité à se rendre à des centres de santé comme les hôpitaux ou les dispensaires.

La télémédecine permet une décision médicale collaborative. Elle joue un rôle important dans le diagnostic ou l'exclusion des maladies chez les patients au début d'admission dans les hôpitaux distants [6]. La réduction globale des coûts pour les

soins économise les examens [7], les hospitalisations et les transferts inutiles des patients [8].

La télémédecine offre une formation médicale continue pour le personnel des hôpitaux [9]. Elle permet l'amélioration de la qualité des soins et la continuité des soins, ce qui évite la mobilité des patients dans les cas de complexité ou d'urgence.

Avec l'avancement des technologies de l'information et des télécommunications, la télémédecine n'est plus une idée abstraite et le concept est de plus en plus mature et très important pour les établissements qui fournissent des soins. La télémédecine peut être considérée comme une nécessité et sa disponibilité parmi les services fournis par les établissements de santé est fondamental ; il permet aux hôpitaux offrant des services médicaux aux patients de surmonter les obstacles physiques en éliminant le besoin de déplacement des patients, ce qui permet aux hôpitaux d'offrir des services de haute qualité à distance.

### 1.1.3 Téléradiologie

Télé radiologie est une branche de la télémédecine qui a une large utilisation en médecine. Elle se base sur des images médicales au format numérique, telles que CT (Computed Tomographie), IRM (Imagerie par Résonance magnétique), etc., qui sont envoyées d'un endroit à l'autre afin d'être analysées et/ou consultées par des spécialistes. La télé radiologie traditionnelle se compose de trois éléments de base: la station qui envoie des images médicales, un canal de transmission et une station de réception.

Ce service médical est pris en charge par les technologies et les télécommunications disponibles. En fait les images sont obtenues à partir de l'équipement médical numérique avec une capacité à produire des images en format numérique. Elles peuvent être transmises à n'importe quel endroit via une connexion réseau, comme par exemple, LAN (Local Area Network) ou WAN (Wide Area Networks).

Les premières traces de la télé radiologie remontent à 1929 dans les états unis d'Amérique, lorsque la première image médicale a été transmise par deux dentistes

radiologues à l'aide d'un télégraphe. A travers un télégramme, une image dentaire rayons X était transmise à un endroit éloigné avec une qualité acceptable [10].

Actuellement la téléradiologie est largement utilisée pour la formation de nouveaux radiologues, pour des téléconsultations dans les cas où est nécessaire un deuxième avis médical, pour fournir une assistance médicale aux régions difficiles d'accès, etc.

L'intégration des équipements radiologiques provenant de divers fabricants dans un même réseau, est une activité qui provoque quelques problèmes et défis. Le besoin d'interopérabilité entre tous les équipements radiologiques est très important dans le réseau de l'hôpital, pour cela a été développé le standard DICOM. Cette norme définit comment les images médicales sont acquises, transférées et stockées.

La norme DICOM a permis l'interopérabilité entre les équipements de différents fabricants, ce qui facilite le transfert d'images médicales d'un endroit à un autre. La station émettrice, qui fait partie des systèmes de téléradiologie, est responsable de la conversion des images médicales, par exemple, aux rayons X, en format numérique et la compression des images en tenant compte de la résolution souhaitée en débit binaire. La compression de données est possible à l'aide de caractéristiques du codage au format DICOM, ce qui permet la compression avec perte ou sans perte, où le degré de fiabilité est variable et dépend de la modalité utilisée. Cette compression permet de réduire la densité ou le nombre de bits par pixel, et, par conséquent, il y a une dégradation de la résolution de l'image. Idéalement, dans la téléradiologie, il ne serait pas nécessaire de compresser les images, si elles sont obtenues par un équipement de haute résolution et avec une haute vitesse de transmission; mais cela est impossible en raison des limites technologiques. En ce sens, l'optimisation d'un paramètre implique la dégradation d'un autre (par exemple en augmentant la vitesse de transmission implique souvent une augmentation du niveau de compression et la réduction de la résolution des images)

En ce qui concerne les stations de réception, elles doivent également être reliées à un point d'accès réseau de haut débit. En outre, la qualité des écrans pour

afficher des images médicales est également une caractéristique importante pour faciliter le diagnostic. Les stations de réception sont généralement équipées de grands écrans avec une bonne résolution [2]. Une autre condition importante dans l'écran moniteur est la capacité de partage afin de permettre au radiologue de visualiser plus d'une image à la fois dans le but de comparer les différentes images. La qualité de la luminosité de l'écran est également très importante pour le médecin parce que les moniteurs avec une plus grande vivacité permettent au médecin de mieux identifier les régions d'intérêt dans l'image à analyser.

Pour l'analyse d'images médicales, il existe des logiciels spécialisés dont les caractéristiques sont un atout pour l'évaluation et la prise de décision par le médecin. La plupart des logiciels offrent un dispositif simple qui permet aux médecins de manipuler le niveau de gris de l'image, le zoom, rotation, etc. Mais, il existe des logiciels d'expertise avec des fonctions spécialisées disponibles [11], permettant au médecin de renforcer les bords, d'effectuer l'affichage, l'égalisation d'histogramme, ajouter des notes à des parties pertinentes de l'image, appliquer des filtres, etc. la valeur des fonctionnalités offertes par le logiciel, qui permettent aux médecins d'analyser des images médicales en téléradiologie dépend de l'ergonomie utilisateur et le type d'images analysées.

L'utilisation de la téléradiologie pour améliorer la qualité des services fournis par les hôpitaux est un scénario de plus en plus fréquent. Actuellement, il existe plusieurs scénarios de téléradiologie, parmi lesquels nous pouvons citer les suivants:

- L'analyse à distance

Dans ce scénario, la téléradiologie permet à un médecin d'obtenir à son domicile par un dispositif portable, les images médicales du patient qui sont transmises à partir de l'hôpital ou clinique pour l'analyser à distance. Ce service permet la consultation instantanée entre le radiologue et un médecin distant ; ce scénario conduit à l'amélioration des services fournis par les centres de santé.

- La médecine générale dans les zones rurales

Elle permet aux médecins de cliniques éloignées qui sont fournisseur de soins de santé primaires dans les zones rurales ou des zone fermées comme les prisons ou les centres de détention, d'envoyer facilement et rapidement des images médicales des patients, pour être consultées et évaluées par des experts qui sont dans les centres de santé éloignés.

- Deuxième consultation médicale

Souvent, il est nécessaire d'avoir une deuxième évaluation d'une image médicale par un autre spécialiste. Grâce à la téléradiologie, ce processus est accéléré parce que le médecin dans un centre de santé peut facilement envoyer un ensemble d'images d'une étude pour être évalué par un autre médecin spécialiste dans un domaine particulier.

Le flux de données et sa disponibilité sont très importants dans la télémédecine pour construire un système solide et utile [12]. Dans [13], un système a été créé pour résoudre le problème du flux de l'information médicale échangée, stocker et partager entre les différents hôpitaux en automatisant le processus de collecte des données vitales des patients via un réseau de capteurs connectés aux dispositifs médicaux existants. Les principaux avantages de ce système sont qu'il fournit aux utilisateurs 7 jours sur 7, les données en temps réel, collecte, élimine le travail de collecte manuelle et la possibilité d'erreurs de frappe, et facilite le processus de déploiement.

Aussi, dans [14] les auteurs ont analysé l'avantage d'utiliser le nouveau paradigme informatique Cloud pour améliorer les systèmes de soins de santé. Ils ont mis un PACS sur le Cloud qui promet l'élasticité et l'évolutivité du Cloud, l'accès universel à l'information et la haute disponibilité des données partout et en tout temps. Le système utilise le concept de «PACS-comme-un-service» qui gère les requêtes store/query/retrieve ; un indexeur d'images DICOM qui analyse les métadonnées et les stockent dans une base de données SQL Azure; et une interface utilisateur Web pour la recherche et la visualisation d'images d'archives sur la base de patients et attributs de l'image. Toutes les données stockées sur le Cloud sont chiffrées. En outre, les clés qui sont utilisées pour chiffrer les données

sont stockées sur un fournisseur digne de confiance ou en interne, de sorte que les fournisseurs du Cloud ne sont pas en mesure de décrypter les fichiers et accéder au contenu en clair.

Il y a quelques travaux qui décrivent d'autres caractéristiques comme par exemple, les interactions de l'utilisateur, et ce qu'ils peuvent faire avec les données disponibles sur le Cloud. De nombreux systèmes se sont focalisés sur l'omniprésence par l'utilisation de l'informatique mobile. Par exemple, il existe des systèmes qui peuvent être utilisés par les médecins pour gérer les dossiers de santé des patients et des images médicales en utilisant l'informatique mobile [15] [16].

Les auteurs de [16] étaient préoccupés par les réelles limitations des téléphones et tablettes mobiles comme par exemple, la puissance de calcul réduite, le stockage limité et la mémoire pour implémenter une architecture à trois niveaux. Ils ont travaillé autour de ces problèmes, visant à permettre aux utilisateurs un accès mobile aux images médicales DICOM. Les problèmes liés aux politiques de gestion de réseau à réseau de soins de santé, à celles qui permettent les protocoles tels que HTTP ou IMAP influent également sur l'architecture des systèmes de télémédecine. Cependant, les relais hébergés dans un service Cloud peuvent être une implémentation permettant la connectivité du système, en raison du fait que le relais est le responsable des communications entre dispositifs client connectés à l'internet via un VPN et un PACS de l'entreprise.

Au passé, un système crée [17] des mécanismes de collaboration Peer to Peer entre le personnel médical, l'amélioration des services de télémédecine et de la collaboration en temps réel. Le système développé a souligné l'importance de TCAO (Travail Coopératif Assisté par Ordinateur) sur Les plates-formes E-santé. Ils ont fourni des capacités de travail collaboratif synchrones grâce à des composants comme la messagerie, outil de visioconférence, espace partagé virtuel, outil traitement d'images etc. Néanmoins, ce système ne fonctionne que pour un seul établissement.

#### 1.1.4 Système d'archivage et de transmission d'images (PACS)

De nos jours, les hôpitaux ont reconnu les avantages de l'informatique et de la technologie dans la gestion de l'information médicale. Il y a trois décennies les

premiers PACS (système d'archivage et de communication) sont apparus [1] ; ils ont révolutionné la radiologie et en quelque sorte toute la médecine. En raison de la grande quantité de données générées dans le domaine de l'imagerie médicale [1] [18] par des examens médicaux comme l'échographie cardiovasculaire (US), l'angiographie (XA), la mammographie numérique, la tomographie par émission de positrons (PET), etc. il y avait la nécessité de créer une infrastructure qui définisse clairement comment sont stockées et comment on accède aux images médicales obtenues à partir du réseau de l'hôpital [1]. L'utilisation du PACS dans un réseau hospitalier facilite et accélère la façon dont les données médicales sont transmises et accessibles. Les systèmes PACS utilisent le format standard défini par la norme DICOM [1] pour stocker et transmettre des images médicales afin de soutenir l'interopérabilité entre les équipements de différents fabricants. Comme un aperçu, on peut dire que le PACS englobe les technologies utilisées pour l'acquisition, l'archivage, la distribution et la visualisation d'un ensemble d'images numériques à l'aide d'un réseau informatique pour le diagnostic et la révision dans les stations dédiées (Figure 1.1)

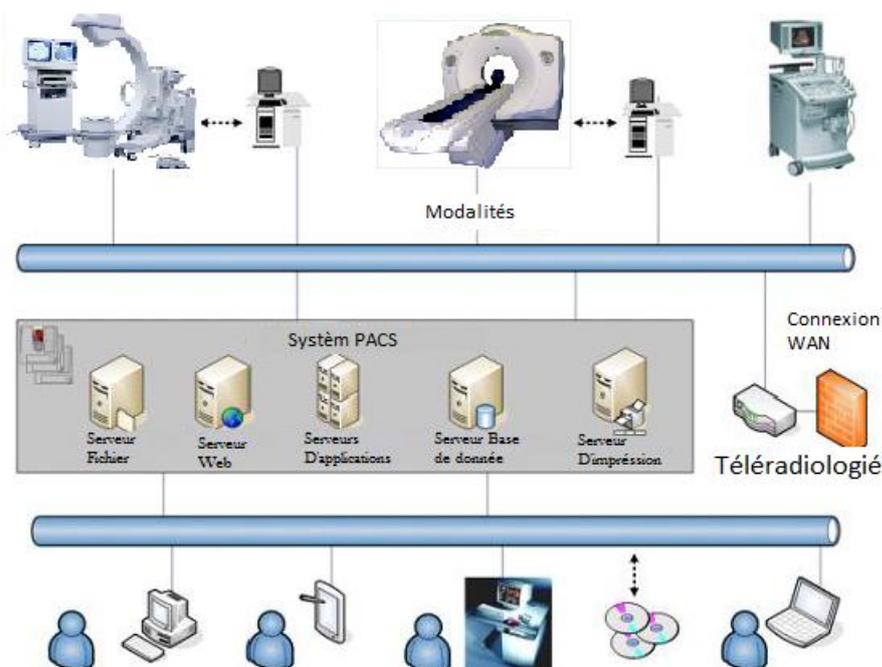


Figure 1.1 : Composants du Système PACS [1]

#### 1.1.4.1 Composants du PACS

En général, un PACS est composé de quatre éléments [1] [18]: des passerelles pour l'acquisition de l'imagerie médicale, des serveurs d'archives, des postes de travail pour la visualisation de données et des serveurs d'applications.

En plus des composants on trouve certaines applications de serveurs PACS et passerelles d'accès qui sont utilisées pour se connecter à des systèmes d'information pour d'autres institutions qui fournissent des soins de santé.

#### 1.1.4.2 Passerelle d'acquisition des images médicales

L'acquisition de l'imagerie médicale est l'une des tâches principales d'un système PACS. En raison de certains problèmes liés à l'équipement provenant de fournisseurs différents, il est nécessaire d'avoir une passerelle pour l'acquisition de données sur le système d'imagerie médicale. L'existence de la passerelle est nécessaire parce que chaque fournisseur met en place son propre protocole, conformément à la déclaration faite par la norme DICOM, et aussi parce qu'il y a un besoin de communication entre différents systèmes PACS. La passerelle constitue un point important du système où se compose la mise en forme des images médicales provenant d'équipements radiologiques dans un format standard utilisé pour communiquer dans le système PACS. Elle est compatible avec le format défini par la norme DICOM. Ces images sont transmises par la passerelle soit pour le serveur d'archives PACS ou un poste de travail pour le visionnement.

Les passerelles sont placées entre les unités responsables de l'obtention des images médicales de chaque modalité et le reste du système PACS. Elles interagissent directement avec l'appareil d'acquisition d'images qui est connecté à des modalités d'imagerie médicale.

#### 1.1.4.3 Serveur PACS

Le serveur d'archives PACS est le moteur de tout système PACS. Ce serveur se compose d'ordinateurs de haute performance. Il est composé d'une base de données où les informations sur les patients sont indexées, de l'HIS et RIS, et un système de fichiers où les fichiers sont stockés dans le court / long terme ou de façon permanente.

Dans un PACS, le serveur PACS est chargé de recevoir les images des examens médicaux et la mise à jour de la base de données du système de gestion. En plus de la fonction de collecte des images médicales, le serveur peut inclure d'autres fonctions comme, par exemple, la compression d'images, la vérification de l'intégrité des fichiers, l'extraction des informations qui décrivent les examens reçus par l'en-tête d'images DICOM, l'interfaçage d'applications de communication avec les serveurs PACS, etc.

#### 1.1.4.5 Poste Client (station de travail)

Les postes de travail incluent la communication avec le réseau du système PACS, un moniteur pour la visualisation des images et des logiciels de traitement d'images. Certains postes de travail peuvent également être équipés d'une base de données locale ou non, en fonction des besoins. Les postes de travail qui ont une base de données locale sont généralement équipés de nombreuses fonctions de traitement d'images médicales et ont besoin seulement de communiquer avec le serveur PACS de façon sporadique, car ils peuvent conserver les informations dans la base de données locale. D'autre part, les postes de travail ne sont pas équipés d'une base de données locale ; ils ont besoin que de quelques fonctions de base de traitement et sont constamment assistés par le serveur PACS.

Les radiologistes utilisent les postes de travail pour faire des diagnostics médicaux d'images de radiologie. Ils fournissent un ensemble de caractéristiques qui peuvent être utilisés pour interagir avec le serveur d'archives PACS ou même d'autres composants PACS. Ces fonctionnalités incluent la possibilité de sélectionner les images auxquelles il veut avoir accès par une requête DICOM, des outils de mesure qui peuvent l'aider dans l'interprétation des images et le diagnostic, l'accumulation de toutes les informations et les images d'un examen particulier pertinent d'un patient donné, etc.

#### 1.1.4.6 Serveur d'application

Les serveurs d'application sont connectés au serveur PACS. Ils peuvent être utilisés pour filtrer les données récupérées à partir du serveur PACS, dans le but de satisfaire à un usage particulier, ou à faire des traitements de données souhaités. Dans les PACS, il est possible de trouver différents types de serveurs d'applications.

Par exemple, il peut avoir des serveurs d'applications dont le but est de créer un service Web pour visualiser des images médicales, la création d'un dossier électronique du patient basé sur les images, créer un serveur d'applications à des fins éducatives, etc.

#### 1.1.5 DICOM

DICOM est une norme internationale [19] qui définit les formats de données, organisation de stockage et les protocoles de communication de l'imagerie médicale numérique, qui a été approuvé en Octobre 1993 par l'ACR (American College of Radiology) et la NEMA (National Electrical manufactures Association). Cette norme a vu le jour à la suite de l'apparition de nombreux équipements dont la capacité d'acquérir, de transférer et de stocker des données de l'imagerie, et la nécessité de standardiser les processus de communication de ces dispositifs sur le réseau.

L'ACR et NEMA ont élaboré un ensemble de normes, recommandations et directives qui permettent la communication de l'information entre les différents équipements d'imagerie numérique, dont est la base du développement et de l'expansion forte du PACS.

L'imagerie numérique et de communication en médecine (DICOM) [19] apparue dans le début des années 90. En 1993, a été publié une nouvelle version qui se compose de 13 pièces. Mais dans sa version actuelle, DICOM 3.0 est divisée en 18 parties qui comprennent 160 suppléments sur des aspects spécifiques d'une ou plusieurs parties de la norme [19]. Ainsi, DICOM est devenue la norme internationale qui définit les formats et les protocoles de communication d'images numériques médicales. Ce standard définit la sémantique des commandes et des données connexes, afin que les différents dispositifs puissent interagir. DICOM normalise l'ensemble des méthodes de stockage et de transmission des images médicales numériques. Ainsi, il permet la communication de renseignements médicaux entre l'équipement numérique (à partir de plusieurs fabricants différents), comme modalités d'imagerie, postes de travail, imprimantes, serveurs, etc.

### 1.1.5.1 Modèle d'information DICOM

Le modèle d'information DICOM est le modèle utilisé [19] par la norme DICOM pour représenter les informations du monde réel comme les patients, les études, les dispositifs médicaux, et ainsi de suite. Toutes les données du monde réel sont représentées dans DICOM comme des objets avec des propriétés et des attributs respectifs. Les Objets DICOM et les attributs sont normalisés selon les informations de définitions des objets DICOM (IOD). IOD sont des collections d'attributs, utilisés pour décrire chaque objet de données particulier. Par exemple un patient IOD peut être décrit par son nom, numéro de dossier médical (ID), le sexe, l'âge, et ainsi de suite.

DICOM opère suivant un modèle de rendu de services lorsque les applications DICOM fournissent des services. En outre, il existe une association entre certains types de services avec les données (IOD) qu'ils traitent. DICOM appelle ces associations paires service-objet (SOP), et les regroupe en classes SOP (Figure 1.2).

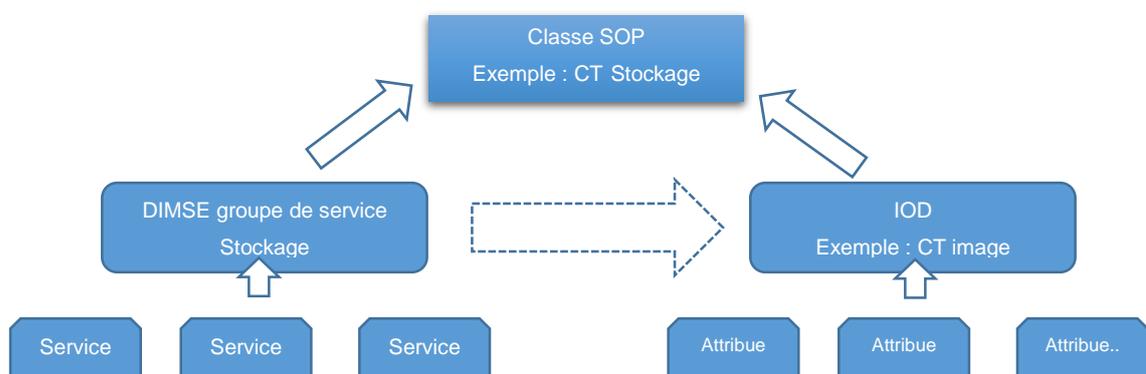


Figure 1.2: Structure de la classe SOP [19]

Toutes les commandes de DICOM et la plupart des attributs de données DICOM sont toujours liés avec le modèle d'information de quatre niveaux représentés par la hiérarchie suivante : patients, études, série, image (Figure 1.3) ; où:

1. Un patient peut avoir de multiples études.
2. Chaque étude peut inclure une ou plusieurs séries d'images.
3. Chaque série comporte une ou plusieurs images.

La hiérarchie DICOM reproduit ce qui se passe dans le monde réel. Par exemple, un patient se rend à l'hôpital et peut faire plusieurs études (par exemple, MR, CT, et échographies) ; ces études peuvent avoir de multiples séries d'images (frontal, axial, avec ou sans produit de contraste, avec des protocoles d'imagerie différentes, et ainsi de suite). Chaque série aura une ou plusieurs images qui y sont associées.

Pour identifier les patients, les études, les séries et les images, la hiérarchie de DICOM définit UID (Unique Identifier) pour chaque niveau de la hiérarchie. Le niveau du patient est identifié par "ID patient" (tous les patients doivent avoir des IDs qui les identifient de façon unique). Tout élément DICOM est identifié par une étiquette. La définition d'étiquette de DICOM sera présentée au paragraphe **Format de données DICOM (1.1.5.2)**. L'élément «Patient ID» est enregistré avec l'étiquette (010,0020), ce qui rend cette étiquette requise pour tout type d'imagerie. Le même principe vaut pour les trois autres niveaux de la hiérarchie étude-série-Image patient : au niveau de l'étude, chaque étude a ses uniques "UID d'instance", (0020,000D); au niveau de la série, chaque série a son unique "série UID d'instance" (0020,000E); et au niveau de l'image, chaque image a son propre "UID d'instance SOP» (0008,0018).

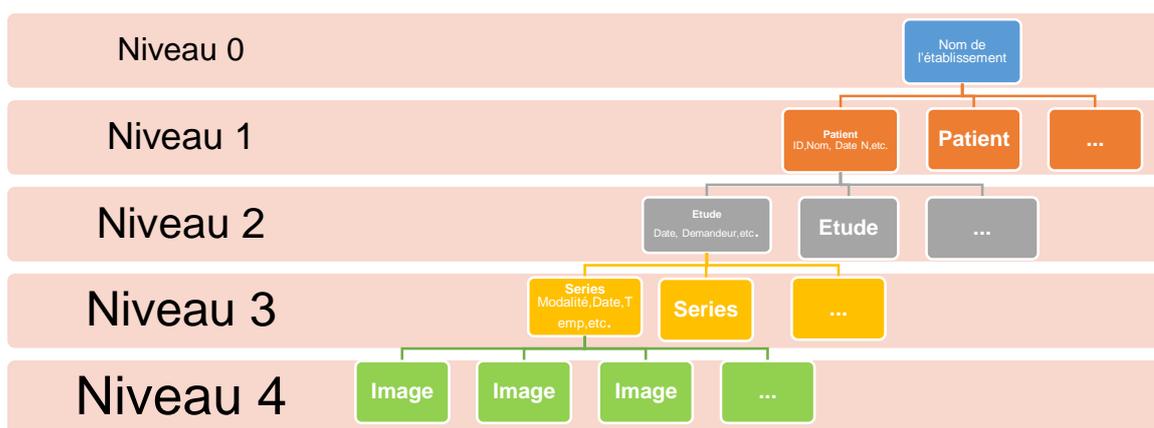


Figure 1.3: La hiérarchie de l'information DICOM [19]

### 1.1.5.2 Format des données DICOM

Les fichiers DICOM prennent en charge plusieurs types d'éléments d'information médicale, comme plusieurs modalités d'imagerie médicale, de courbes, de rapports structurés, etc. Chaque fichier DICOM a un entête qui contient des métadonnées, utilisés pour représenter le modèle d'information DICOM [1] , y compris des renseignements connexes avec les patients, le personnel de la clinique, établissement, de l'équipement et les conditions de l'examen, etc... Le modèle d'information DICOM [1] contient les données qui correspondent avec les informations du monde réel (Figure 1.4).

DICOM regroupe les informations en ensembles de données. Chaque ensemble de données d'un fichier DICOM contient des informations spécifiques ; par exemple, dans un fichier DICOM on peut trouver l'ID du patient, le nom de l'établissement, le type de modalité, les octets d'images médicales numériques. En outre, cet ensemble de données ne peut jamais être séparé par erreur.

Un objet de données DICOM est formé par un grand nombre d'attributs, et un attribut spécial contient les données de l'image en pixel. Un objet DICOM contient un seul attribut de données pixel. Cela peut correspondre à une seule image. Mais dans certaines modalités, cet attribut peut contenir plusieurs images, pour stockage de diapositives boucles ou d'autres données multi-images. En outre, les données des pixels de l'objet DICOM peuvent être compressées en utilisant une variété de normes, y compris JPEG, JPEG Lossless, JPEG 2000, MPEG3, MPEG4, etc.

Un objet d'élément de données DICOM est structuré comme suit: GROUPE (2 octets), Element (2 octets), VR (2 octets), LengthInByte (2 octets), données (longueur variable). Il peut coder des éléments de données en utilisant différents systèmes de codage. Avec un Type de la valeur de représentation (VR) d'éléments de données, où VR définit la représentation de la valeur de l'élément (qui peut être DA pour la date, TM ou temps, PN pour le nom du patient, etc.). En outre, les objets DICOM peuvent avoir des éléments de données implicites dans laquelle VR n'est pas présent [20].

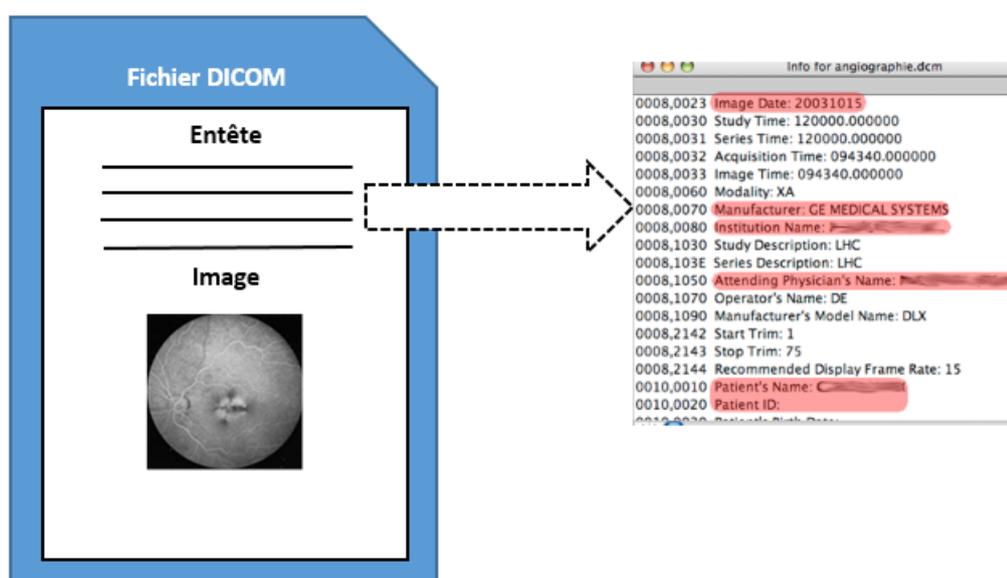


Figure 1.4: La Structure d'un objet DICOM

DICOM organise les données selon le format TLV (Tag, longueur, valeur) comme le montre la figure 1.5. Le tag dispose de 4 octets, où les 2 premiers octets sont pour le groupe et les 2 derniers octets pour l'élément, le tag "(groupe, élément)" utilisé pour identifier une rubrique donnée. Par exemple, le nom du patient a le tag 0x0010 de la valeur du groupe et 0x0010 la valeur de l'élément, donc il est représenté comme (0010,0010) tag.

Le champ de Représentation des valeurs (VR) est utilisé pour définir le type de code pour un élément donné et il dépend de l'élément de tag. Par exemple, les champs VR peuvent avoir la DA de valeur donnée pour la date, TM pour le temps, l'OB pour objet binaire, etc.

Certains éléments peuvent avoir un VR implicite car il est possible, selon une valeur d'étiquette, de consulter un dictionnaire DICOM qui définit le type d'un élément.

Le champ Longueur est utilisé pour définir la longueur en octets du champ valeur de l'élément. Cette valeur dépend de la valeur de tag. Enfin, l'élément contient le champ valeur, qui contient les données qui sont stockées dans l'élément (par exemple, les données pixels de l'image, la date de naissance du patient, le nom du patient, ID du patient, etc.).

Comme on peut le vérifier sur la figure 1.5, les fichiers DICOM sont formés par une séquence d'éléments de données ou des séquences TLV. Chaque TLV représente un attribut. En outre, il est possible de créer des tags privés et augmenter les capacités de protocole DICOM pour représenter des informations supplémentaires dans le monde réel.

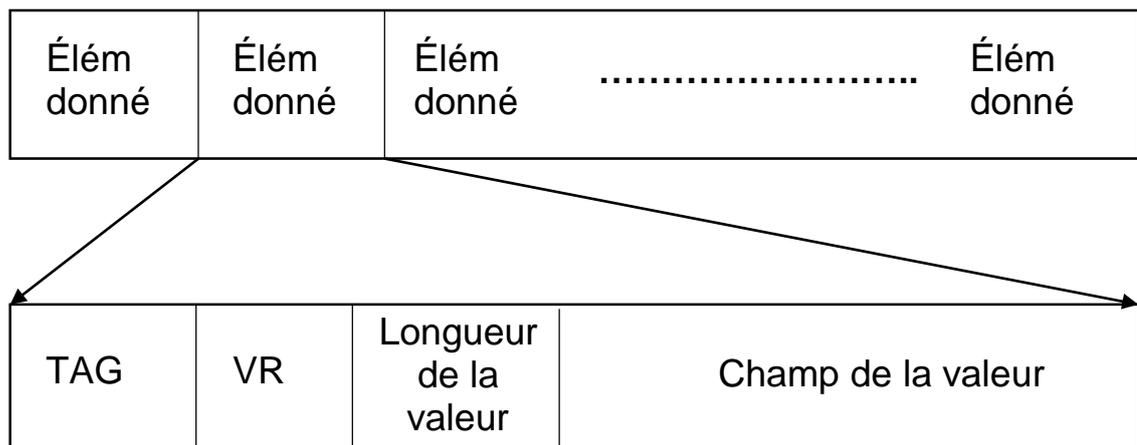


Figure 1.5 : Élément de donnée d'un objet DICOM

### 1.1.5.3 La Communications DICOM

DICOM offre de nombreux services [21] [1], dont la plupart concernent la transmission de données sur un réseau. Le protocole utilise TCP / IP pour établir une connexion fiable entre les points finaux. Comme d'autres protocoles tels que SMTP, HTTP, FTP, etc. DICOM ne fait qu'ajouter son propre langage de réseau (à la couche application) sur TCP / IP. Ce langage se compose de services de haut niveau, DICOM Message Service Elements (DIMSE).

Nom	Groupe	Type
C-STORE	DIMSE-C	opération
C-GET	DIMSE-C	opération
C-MOVE	DIMSE-C	opération
C-FIND	DIMSE-C	opération
C-ECHO	DIMSE-C	opération
N-EVENT-REPORT	DIMSE-N	notification
N-GET	DIMSE-N	opération
N-SET	DIMSE-N	opération
N-ACTION	DIMSE-N	opération
N-CREATE	DIMSE-N	opération
N-DELETE	DIMSE-N	opération

Tableau 1.1 : Services DIMSE [21]

La communication DICOM suit le modèle client / serveur [1]. Dans le réseau DICOM, on trouve le fournisseur de classes de service (SCP) et l'utilisateur du client de service (SCU). SCP ou SCU est utilisé pour désigner un type de dispositif, en fonction de son rôle dans le réseau (Figure 1.7). Une entité peut jouer le rôle de SCP ou d'application SCU permettant de faire communiquer les uns avec les autres (Figure 1.6). Par exemple, une modalité ou un poste de travail qui produit ou exploite des images, doit interagir avec l'archive PACS. Dans ce cas, la modalité ou le poste de travail est SCU et le serveur PACS appartient à SCP.

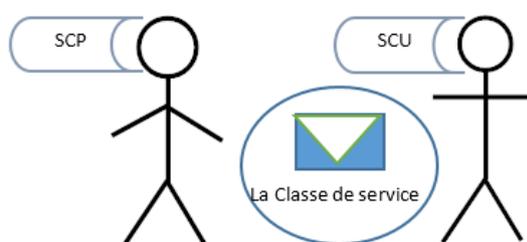


Figure 1.6: Modèle SCU-SCP DICOM [17]

Dans le réseau DICOM, chaque appareil dispose d'un Titre d'entité d'application (AETitle) qui l'identifie. AETitle est utilisé pour traiter des entités dans un réseau DICOM. Pour accéder à tous les services DICOM ou à communiquer avec tout appareil DICOM, il faut d'abord créer une association DICOM, afin de créer un canal d'échange d'informations. Dans la mise en place d'une association DICOM, des négociations de plusieurs paramètres s'effectuent, tels que le codage (ex. Little endian, big-endian), les formats de compression d'image, type d'information à transférer, et la durée de l'association. Après la négociation, les

commandes de services sont exécutées entre SCU et SCP pour effectuer l'objectif de service.

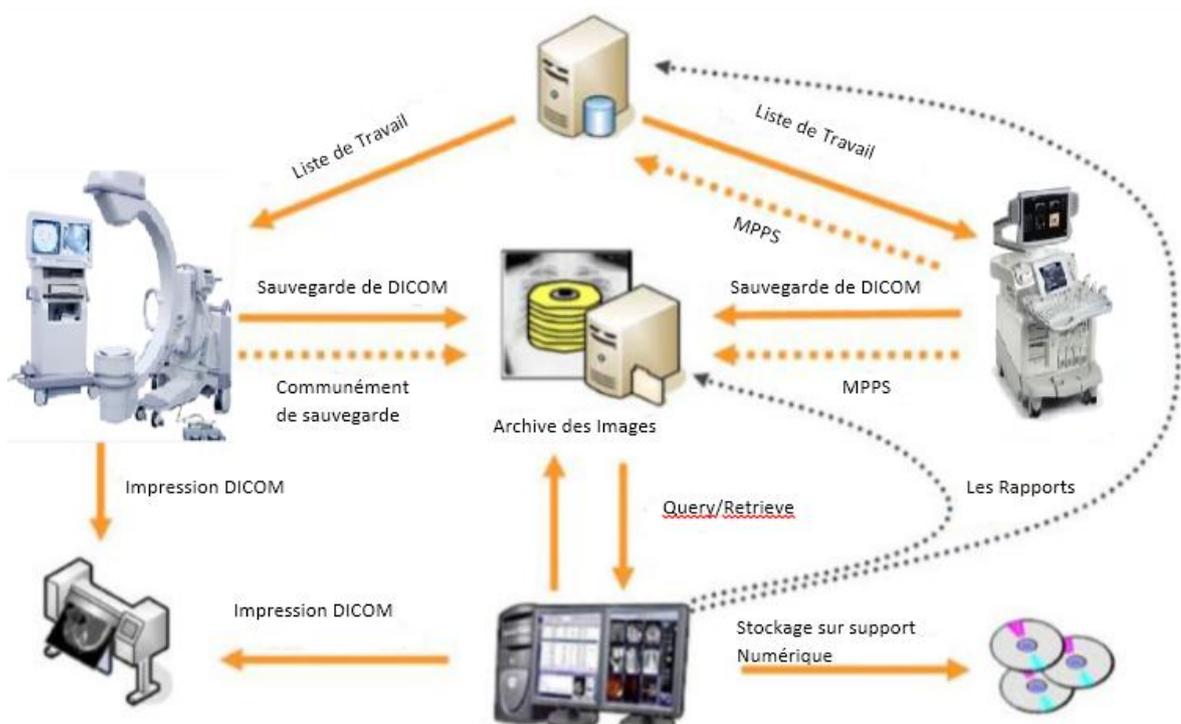


Figure 1.7: Services et les entités du réseau DICOM [17]

#### 1.1.5.4 Service Store

Le service Store de DICOM est utilisé pour envoyer des images ou autres objets (rapports structurés, des PDF encapsulés, etc.) à une archive de PACS ou poste de travail. Ce service utilise la commande C Store (stockage SOP) pour déplacer les images DICOM entre les entités, sur un réseau DICOM (Figure 1.8).

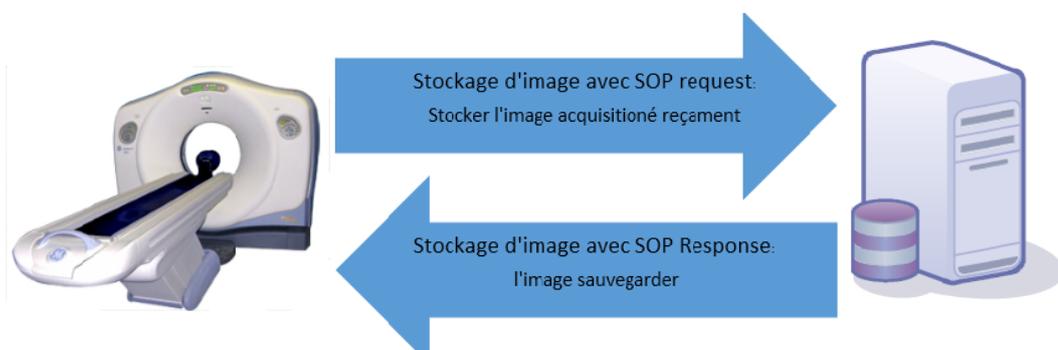


Figure 1.8: C-Store DICOM [17]

### 1.1.5.5 Service Storage Commitment

Le service « Storage Commitment DICOM » permet de confirmer que l'image a été enregistrée en permanence par un dispositif (soit sur des disques redondants ou sur des supports de sauvegarde, par exemple g sur un CD). SCU utilise la confirmation de la SCP pour s'assurer qu'il est aucun risque afin de supprimer les images localement.

### 1.1.5.6 Service Q/R (Query/Retrieve)

Cela permet à un poste de travail de trouver des listes d'images ou d'autres objets, puis les récupérer à partir de l'archive de PACS. Une recherche d'images DICOM basic est exécutée en utilisant C-Get SOP. C-Get enveloppe C-Find et C-Store dans une seule classe de service où C-Find est utilisé pour interroger l'image souhaitée, suivie par un C-Store pour récupérer ces images. Quand un C-GET est envoyé à la SCP, la SCP utilise d'abord les paramètres de recherche pour trouver les images puis appelle C-Store pour les renvoyer à la SCU (Figure 1.9).

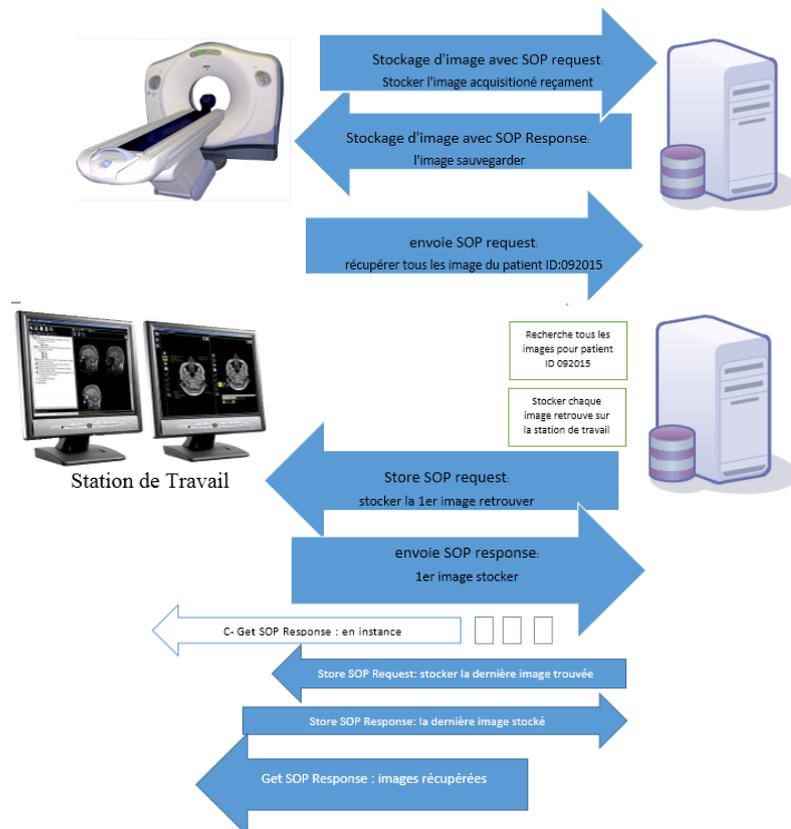


Figure 1.9: Résumé du service C-Store dans le service C-Get [17]

Par conséquent, il est possible d'utiliser un service de récupération plus avancée qui permet de déplacer des images à des tiers (Figure 1.10). Le C-get est utilisé pour renvoyer des images à la même SCU qui fait cette demande, cependant avec C-Move, il est possible d'envoyer les images à toute entité. Ainsi, C-Move a besoin de savoir où renvoyer les images. En effet, cette question n'est jamais soulevée dans C-Get; car le C-Get SCP renvoie toujours les images à l'entité d'image demandeur (C-Get SCU).

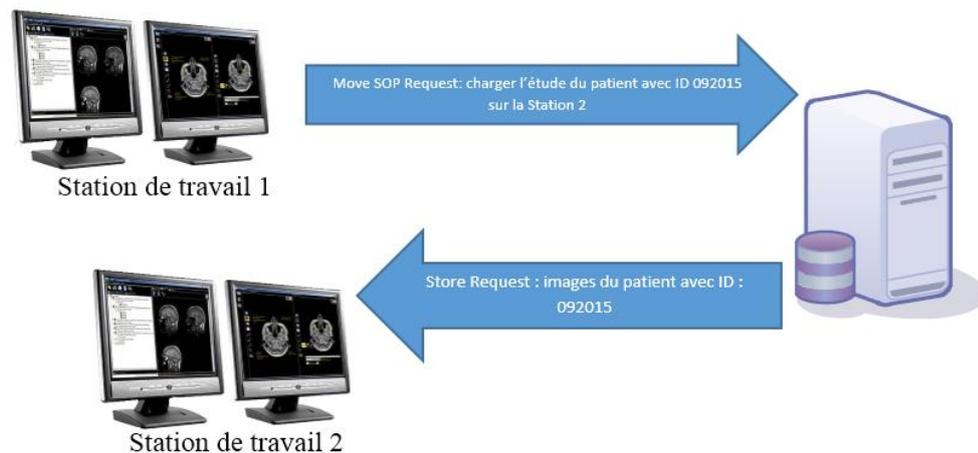


Figure 1.10: C-Move avec trois entités: la station de travail 1 demande à l'Archive d'envoyer une image pour la station de travail 2. L'archive envoie les images à la station de travail 2 avec une sous-opération C-Store [17]

#### 1.1.5.7 Service MWL (Modality Worklist)

Cela permet à un équipement d'imagerie (une modalité) d'obtenir des détails sur les patients et les examens prévus par voie électronique, en évitant le besoin de taper les informations à plusieurs reprises (et les erreurs humaines causées par la saisie) (Figure 1.11).

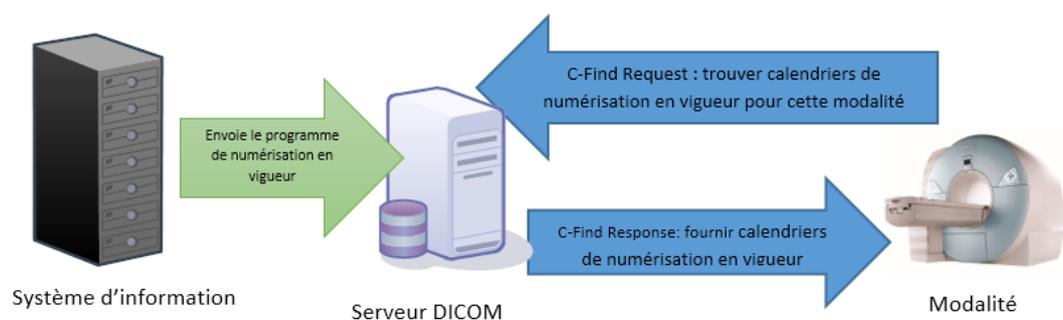


Figure 1.11: Exemple DICOM MWL: remplissage de modalités d'imagerie avec les données de base des patients [17]

#### 1.1.5.8 Service DICOM Printing

Le service d'impression DICOM est utilisé pour envoyer des images sur une imprimante DICOM, normalement à imprimer un film x-ray. Il y a un calibrage standard pour assurer la cohérence entre les différents dispositifs d'affichage, y compris une copie imprimé sur papier.

#### 1.1.5.9 Service MPPS (Modality Performed Procedure Step)

C'est un service complémentaire au service MWL, cela permet à la modalité d'envoyer un rapport au sujet d'un examen effectué, y compris des données sur les images acquises, heure de début, heure de fin, et la durée de l'étude, la dose délivrée, etc.

Il permet de donner au service de radiologie une idée plus précise sur la ressource à utiliser (station d'acquisition). Également connu sous le nom MPPS, ce service permet une modalité de mieux coordonner avec les serveurs de stockage de l'image en donnant au serveur une liste d'objets à envoyer avant ou pendant l'envoi de ces objets effectivement.

## 1.2 Cloud Computing

### 1.2.1 Définition

Le Cloud Computing est un modèle informatique qui fournit des services qui permettent une utilisation souple des serveurs virtuels. En outre, les services Cloud Computing sont le résultat d'un nouveau concept qui consiste à définir l'informatique comme un service plutôt qu'un produit [22]. Pour décrire le Cloud Computing en fonction de ses caractéristiques et de son interaction avec d'autres technologies de l'information, on peut dire que c'est un modèle informatique qui base ses fonctionnalités sur des systèmes virtuels. C'est une agrégation de ressources disponibles distribuées et partagée par des serveurs virtuels, ce qui permet une évolutivité massive et la création d'une infrastructure dynamique [22].

### 1.2.2 Caractéristiques du Cloud Computing

Il y a beaucoup de caractéristiques qui conduisent à l'utilisation du Cloud Computing par de nombreuses entreprises et fournisseurs de services basés sur le

Web. Par conséquent, les utilisateurs des services de Cloud peuvent voir Cloud Computing comme une combinaison importante des capacités [22], comme l'accès universel, le contrôle d'utilisation, les gains des prix, les plateformes standard, les services de support, etc. Ces avantages forment ensemble une combinaison unique qui améliore les services basés sur le Web.

#### 1.2.2.1 Une infrastructure hautement évolutive

L'évolutivité de l'infrastructure est la principale caractéristique de Cloud Computing. Cet aspect distingue le Cloud Computing d'autres architectures qui ont des caractéristiques similaires, comme la grille de calcul.

Du point de vue des utilisateurs finaux, l'évolutivité du Cloud Computing permet de gérer et de contrôler la quantité de calcul et stockage, dont ils ont besoin, permettant ainsi aux consommateurs de s'abstraire de la gestion technologique de l'information.

Les entreprises et fournisseurs des services basés sur le Web qui exécutent leurs propres serveurs ont généralement besoin d'élargir leur infrastructure afin d'améliorer leur puissance de calcul et leur stockage. Avec ces extensions arrivent souvent quelques problèmes liés au matériel supplémentaire à acheter ou à adapter dans les serveurs existants ; le responsable informatique va rencontrer des problèmes liés à des incompatibilités avec le système d'exploitation, les conflits dans la planification de la charge de travail, etc. Ces problèmes sont évités lorsque le Cloud Computing est utilisé, en raison de trois principes technologiques: l'allocation rapide des serveurs virtuels, un matériel normalisé et le stockage sur Cloud.

#### 1.2.2.2 Allocation rapide des serveurs virtuels

L'élasticité du Cloud Computing est l'une des principales caractéristiques qui caractérise ce modèle informatique, en permettant aux utilisateurs d'allouer le nombre et le type des machines virtuelles nécessaires pour effectuer une tâche donnée.

Dans un Cloud, tous les serveurs physiques deviennent des ressources communes, où la répartition des emplois et des serveurs virtuels fonctionnant sur

un ensemble de serveurs physiques peuvent changer rapidement entre les serveurs physiques. L'informatique élastique permet aux utilisateurs d'allouer des applications et de libérer les instances des machines virtuelles à la demande, de sorte que la répartition de la puissance de calcul soit totalement transparente pour les utilisateurs des applications.

Amazon EC2 est un exemple de service Cloud qui permet aux utilisateurs d'applications de redimensionner horizontalement des machines virtuelles si elles ont besoin des instances en plus pour exécuter une tâche. Amazon EC2 surveille l'utilisation du processeur des machines virtuelles, de sorte que si elle atteint une limite donnée d'utilisation, une nouvelle instance est créée automatiquement. Mais la vérité est que toutes les applications ne sont en mesure de prendre avantage de cette évolutivité. La demande doit être conçue en tenant compte des caractéristiques d'élasticité du Cloud. L'évolutivité de l'infrastructure ne reflète pas automatiquement l'évolutivité de l'application.

### 1.2.2.3 Un stockage persistant dans le Cloud

L'évolutivité du Cloud Computing nous permet de créer de nouvelles instances selon la nécessité d'une tâche donnée. Chaque instance alloue des ressources de calcul et de stockage temporaire sur les serveurs physiques, mais, une fois que les machines virtuelles sont libérées, toutes les données générées et stockées localement seraient perdues. Par conséquent, il est nécessaire le stockage des données persistantes sur le Cloud, soumis sous réserve des restrictions de contrôle d'accès afin que les données soit pas accessible par tous les serveurs dans le Cloud.

Il est habituel dans le Cloud de dissocier le stockage persistant de serveurs, permettant aux fournisseurs du Cloud Computing d'avoir un contrôle fin sur les ressources allouées dans le Cloud. En fait, l'évolutivité est possible grâce à la combinaison de provisionnement rapide de matériel standard (baie de stockage) et l'utilisation d'une stratégie de stockage persistant.

#### 1.2.2.4 L'accès universel

Les utilisateurs qui utilisent le Cloud Computing pour déployer des serveurs ou des services basés sur le Web ont un accès universel au service de n'importe où sur Internet. L'accès à l'information peut se faire n'importe où via un navigateur web, une application de bureau ou une application mobile, alors que les données sont stockées sur des serveurs dans le Cloud.

Inutile de dire lorsque nous parlons du Cloud privé que l'accès universel n'est pas le même que l'accès ouvert. Pour restreindre l'accès aux ressources dans le Cloud, l'autorisation et un mécanisme d'authentification sont utilisés. Même en Cloud public, ce qui concerne la nécessité de comptabilisation, un mécanisme d'identité est nécessaire pour faciliter la gestion et la facturation.

#### 1.2.2.5 Un contrôle d'utilisation et de prix

Le Cloud est en train de devenir un modèle informatique populaire, surtout parce que les consommateurs de services peuvent louer ce dont ils ont besoin au lieu de construire et d'exploiter une infrastructure entière, qui présentent des coûts énormes pour la création et gestion. Par exemple, mettre en place un centre de données n'est pas facile avec les coûts liés au matériel et les licences de logiciels. Il y a d'autres préoccupations comme la climatisation, l'électricité, la sécurité physique, les systèmes de sécurité anti-catastrophes (incendies, inondations, tremblements de terre, etc.), ...

Les avantages économiques du paradigme Cloud Computing sont des facteurs clés pour son adoption; donc les utilisateurs ont à des degrés de granularité le contrôle d'utilisation et les prix. Il est courant du problème de la quantité de calcul lors de l'achat d'un nouveau serveur pour l'infrastructure. Il pourrait y avoir le risque de i) sous-dimensionner les besoins qui ne répondent pas au niveau de service ou ii) surdimensionner les capacités nécessaires donc une dépense d'un montant d'argent inutile. Mais avec le Cloud Computing, les utilisateurs ne seront plus confrontés à ce problème, car ils peuvent affecter la puissance de calcul et le stockage selon les besoins de leurs applications. Par exemple, dans les périodes de fort demande, il est possible d'allouer des ressources de Cloud pour répondre à

la nécessité et il est possible de libérer ces ressources lorsqu'ils ne sont plus nécessaires, afin que les utilisateurs ne paient que pour ce qui est utilisé.

### 1.2.3 Les Modèles de Déploiement

Il y a 4 modèles de déploiement des infrastructures de Cloud Computing [22] : le Cloud public, le Cloud privé, le Cloud de la communauté et le Cloud hybride.

#### 1.2.3.1 Le Cloud Public

Ce modèle de déploiement est basé sur l'externalisation de calcul et de stockage à des tiers. Ces derniers sont situés dans des centres de données qui opèrent en dehors des entreprises qui les utilisent. Dans ce modèle, toutes les ressources, les processus et les données sont traitées par un fournisseur de service Cloud public. Habituellement ces services Cloud public sont fournis au public par un modèle de « payer à l'utilisation ». Ce modèle peut être confronté à des problèmes de confidentialité et de sécurité.

L'utilisateur du Cloud public voit les ressources dans le Cloud comme une ressource infinie, qu'il peut allouer selon ses besoins et payer seulement ce qu'il utilise. Les fournisseurs de Cloud fournissent habituellement des services Web ou des applications Web pour accéder à l'infrastructure via Internet. Les Cloud public sont fournis par des sociétés géantes comme Amazon, Hewlett-Packard, IBM, Google, Microsoft, Rackspace, Salesforce, etc.

#### 1.2.3.2 Le Cloud Privé

Les Clouds privés sont gérés par une entreprise ou un fournisseur de Cloud Computing. Les services fournis sont consommés intérieurement par une seule entreprise et ne sont pas accessibles au public. Le Cloud privé utilise la même technologie que le Cloud public ; il est souvent utilisé lorsque la politique de l'infrastructure repose sur la sécurité, et d'autres risques qui sont des facteurs déterminants. Il permet également à une entreprise individuelle de maximiser l'utilisation de ses ressources informatiques. Par conséquent, l'utilisation des Cloud privés permet aux utilisateurs de mieux gérer les politiques et le contrôle d'accès, et de définir ses propres machines virtuelles utilisée dans le Cloud.

L'utilisation du Cloud privé à usage interne peut être plus efficace et réactive pour les besoins d'une entreprise par rapport au modèle traditionnel d'exploitation informatiques, car le Cloud Computing prend avantage de certaines caractéristiques importantes pour améliorer l'allocation des ressources qui ont été discutées au début. Cependant, par rapport au Cloud public, il nécessite une énorme quantité de dépenses en capital du matériel et des licences de logiciels. L'utilisation du Cloud privé apparaît avec une certaine critique, comme les serveurs traditionnels de l'infrastructure maintenus par une société, où une équipe de professionnels de l'informatique qui doit être disponible pour gérer l'infrastructure. Quoi qu'il en soit, le Cloud privé ne se débarrasse pas de la question de la planification et de la capacité, et l'expansion de son infrastructure nécessiterait des dépenses de capitaux et de temps.

#### 1.2.3.3 Le Cloud de la communauté

Le Cloud de la communauté est celui où le Cloud a été organisé pour servir une fonction ou un objectif commun. Il peut être pour une organisation ou pour plusieurs organisations, mais ils partagent des préoccupations communes telles que leur mission, les politiques, la sécurité, les besoins de conformité réglementaire, et ainsi de suite. Un nuage de la communauté peut être géré par l'organisation (s) de constituant ou par un tiers.

#### 1.2.3.4 Le Cloud Hybride

Les Cloud hybrides sont une composition des Cloud privé et public. C'est le cas d'une entreprise qui possède un Cloud privé, elle peut utiliser les ressources de Cloud publics pour étendre la capacité de son Cloud. Les Cloud hybrides sont couramment utilisés lorsque l'entreprise a besoin de faire face à quelques questions de sécurité, de confidentialité et de contrôle d'accès. Il existe quelques moyens pour mettre en œuvre un Cloud hybride.

Ce modèle de déploiement pourrait utiliser deux Cloud gérés séparément comme une plateforme de services, où certaines politiques sont définies pour désigner les tâches qui peuvent être exécutées dans le Cloud public, et celles qui peuvent être exécutées dans le Cloud privé. Les consommateurs peuvent choisir librement le service qu'ils vont utiliser, en tenant compte de certaines variables

importantes comme, par exemple, le coût d'utilisation du Cloud public dans un modèle hybride peut être fournir plus de capacité moins cher que le Cloud privé.

Une autre façon de mettre en œuvre un Cloud hybride est de créer un Cloud où la gestion des composantes privées et publiques est présentée dans une seule plateforme de gestion de service. En utilisant cette approche, le Cloud privé et Cloud public sont toujours deux services indépendants, qui sont gérés à l'aide d'un point de gestion unique.

Dans certains Cloud hybride, il y a une mise en œuvre dans le Cloud public d'un réseau privé virtuel (VPN), afin de traiter une partie de ce Cloud public comme une extension du Cloud privé.

Certains fournisseurs offrent des solutions qui peuvent être utilisées pour permettre le déploiement Cloud hybride [22]. Par exemple, on peut trouver des offres comme Amazon Virtual Private Cloud, Enterprise Cloud Services-Virtual Private Cloud pour HP, et Cisco Private Cloud solutions. Ces solutions utilisent des tunnels IPsec VPN pour se connecter à l'infrastructure Cloud public.

#### 1.2.4 La couche service dans le Cloud

Les services du Cloud Computing peuvent être divisés en trois couches [22] ; chaque mise en œuvre d'un modèle de service différent sera présentée dans cette section. Ces modèles sont très importants pour définir un modèle architectural pour les solutions basées sur le Cloud.

Les couches des services du Cloud Computing sont organisées de haut vers le bas : chaque couche peut être composée à partir des services de la couche inférieure.

##### 1.2.4.1 La couche Logiciel comme un Service (SaaS)

C'est la couche la plus haute dans le modèle proposé. Software-as-a-service se caractérise par le haut niveau d'abstraction, la prestation des services déployés et configurés pour être utilisée par les utilisateurs finaux. Dans cette couche, les utilisateurs n'ont aucun contrôle sur la couche inférieure de l'infrastructure du Cloud,

donc elle représente juste un point d'accès pour les utilisateurs finaux afin d'accéder à un service donné comme, par exemple, des portails ou des outils de visualisation.

Contrairement aux logiciels traditionnels, SaaS n'a pas besoin d'une installation de logiciel côté client, et toutes les données et le flux de l'entreprise sont tenus dans l'infrastructure Cloud. Une des caractéristiques importantes du SaaS est que les logiciels fournis comme un service dans un nuage ont un accès universel à travers Internet avec une haute disponibilité. Les applications SaaS ont généralement une interface Web accessible via un Services Web, Web2.0 ou Web3.0 [22].

De nos jours SaaS est massivement utilisé, un de ses principaux exemples d'utilisation est le service de messagerie électronique, qui est très utilisé sur Internet, avec une haute disponibilité. L'un de ces services de messagerie est Gmail qui est toujours disponible, et peut être consulté sur Internet à l'aide d'un navigateur Web, une tablette ou un smartphone. En plus des services de courrier électronique, nous pouvons trouver d'autres logiciels fournis comme un service tel que, par exemple, Dropbox, Google Apps, les applications de réseaux sociaux (par exemple Facebook, LinkedIn), jeux en ligne, etc.

#### 1.2.4.2 La couche Platform comme un Service (PaaS)

La couche Platform-as-a-Service permet aux consommateurs d'avoir une abstraction sur l'application par rapport aux limitations traditionnelles du matériel. Elle permet aux développeurs de se concentrer sur le développement d'applications et de ne pas se soucier des systèmes d'exploitation, l'évolution de l'infrastructure, l'équilibrage de charge et la tâche de l'administration du système. PaaS permet aux utilisateurs de développer des applications à déployer sur le Cloud des fournisseurs d'infrastructures. Ces applications sont développées selon les langues de programmation et les API définies par le fournisseur du Cloud. Le PaaS offre aux utilisateurs un contrôle limité sur la couche inférieure de l'infrastructure Cloud. Ils peuvent déployer et configurer les applications créées à l'aide de l'environnement de programmation du vendeur. Le processus de mise en développement et le déploiement d'une application Cloud devient plus facile tout en permettant au programmeur de se concentrer sur d'autres questions importantes.

Un exemple bien connu du PaaS est le Google App Engine, qui permet aux développeurs de déployer des applications en utilisant Python et API Java. Windows Azure est une PaaS de Microsoft qui propose pour les applications différents types d'environnements d'exécution et des services de stockage.

#### 1.2.4.3 La couche Infrastructure comme un Service (IaaS)

La couche Infrastructure comme un Service est la couche Cloud Computing avec le peu de ressources virtuelles. Le calcul, le stockage et le réseau, sont offerts sur demande. IaaS permet aux utilisateurs d'instancier les serveurs virtuels qui peuvent exécuter plusieurs choix de systèmes d'exploitation et de logiciels. Le Cloud Infrastructure-as-a-Service offre l'élasticité et une évolutivité instantanée, de sorte qu'il est possible d'augmenter dynamiquement la puissance de calcul des applications à travers l'instanciation rapide de nouvelles machines virtuelles; permettant aux utilisateurs d'augmenter la puissance de calcul selon le besoin.

La couche IaaS est la couche inférieure des couches de services du Cloud Computing sur laquelle se base toutes les autres couches services. Les ressources sont disponibles dans un système virtuel, où les utilisateurs ont accès administratif complet à leurs machines virtuelles. En outre, certains fournisseurs de Cloud permettent aux utilisateurs de choisir l'image de machine virtuelle qu'ils veulent utiliser à partir d'une variété d'images de machine virtuelle. Cette dernière peut être basée sur une plate-forme Windows ou une plate-forme Linux. L'exemple le plus connu est Amazon EC2.

### 1.3 Télé ophtalmologie collaborative

Ce travail vise à utiliser le paradigme des réseaux sociaux et les technologies d'information dans le milieu médical, où la connexion entre les utilisateurs est très importante. Poussé par la nécessité d'un environnement de travail collaboratif fortement connecté, où les utilisateurs peuvent interagir les uns avec les autres afin d'améliorer le télétravail.

De nombreux systèmes de télémédecine ont été proposés au fil des années, impliquant divers domaines de recherche. Szostek, K. et A. Piórkowski ont développé une application web multiutilisateurs basé sur OpenGL pour visualiser

les données médicales en 3 dimensions [23]. Le travail de Lee, K.-T., et al [24] est un autre exemple, où une architecture est présentée pour un système de télémédecine basé sur l'internet ; le modèle client-serveur est utilisé comme une architecture de processus dans le système. L'application des clients est mise en œuvre via l'Applet Java. La visualisation des images DICOM a été traitée dans de nombreux travaux de recherche. Laird, S. P. et al. ont proposé deux architectures pour leur système [25]. La première utilise CORBA (Common Object Request Broker Architecture) à travers le middleware pour fournir la transparence de localisation des images et un mécanisme cohérent pour accéder à l'ensemble des bases de données (RIMS et serveurs d'images). La seconde utilise des connexions directes aux bases de données locales disponibles.

La synthèse de tous ces travaux va nous amener à proposer un système multiplateforme pour la visualisation des images médicales et la détection automatique des lésions rétinienne en temps réel sur le web. Ce système est basé sur le serveur jabber et HTML5, qui intègre la dernière technologie du Web. Il peut être utilisé pour collecter, stocker et partager toutes les données d'un patient, échanger des diagnostics sur la rétinopathie diabétique entre les ophtalmologistes des différents services de la santé en temps réel sur le Web.

Dans la section suivante, nous décrivons quatre architectures différentes pouvant être utilisées pour la visualisation des images médicales et la détection automatique des lésions rétinienne dans un navigateur Web.

### 1.3.1 Modèle des architectures possibles

Avec la technologie actuelle, il est possible de visualiser des images médicales sans recourir à du matériel spécialisé offrant une interactivité élevée client-serveur. Ceci peut être réalisé en utilisant le serveur comme un centre de traitement pour effectuer le traitement d'image hors-ligne et un navigateur Web pour afficher le résultat final. Dans ce qui suit, nous allons présenter quatre types possibles d'architectures logicielles pour visualiser des images médicales et détecter automatiquement des lésions rétinienne sur un navigateur Web.

## A. Architecture 01

L'architecture 1 (Figure. 1.12) comprend deux modules. Le module de visualisation d'images DICOM effectue toutes les opérations pour visualiser et manipuler une image DICOM ; le module de traitement d'images effectue toutes les opérations pour détecter automatiquement les lésions rétiniennes. Dans l'idéal, les deux modules devraient être mis en œuvre sur un serveur PACS pour des performances optimales. Ces modules doivent avoir accès au PACS, généralement situés dans le même emplacement physique du serveur HTTP, dans le côté serveur. Une fois la détection automatiquement des lésions rétiniennes terminée, les annotations de la nouvelle image sont envoyées à un port de réseau en utilisant une connexion de socket. Les modules de visualisation, traitement d'image et le serveur HTTP doivent être constamment connectés au port.

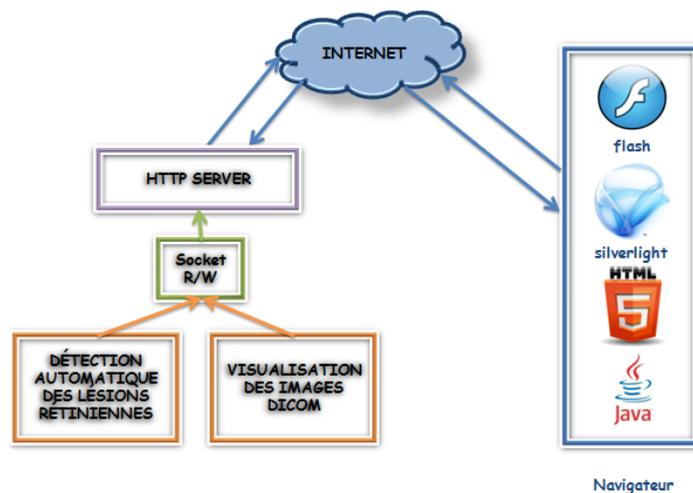


Figure. 1.12 Architecture 01

## B. Architecture 02

L'architecture de la figure 1.13 est plus simple que la précédente. Le serveur reçoit seulement les requêtes HTTP du client et y répond ; ensuite, le navigateur doit télécharger l'image et faire la détection automatique des lésions rétiniennes.

Les données sont stockées sur le serveur pour que le client puisse y accéder directement en utilisant une URI ou adresse URL. Une autre option très utilisée correspond à l'envoi par le serveur d'une version à faible résolution des données.

Cette version peut varier en fonction des méthodes de compression, la taille du modèle original et la bande passante disponible.

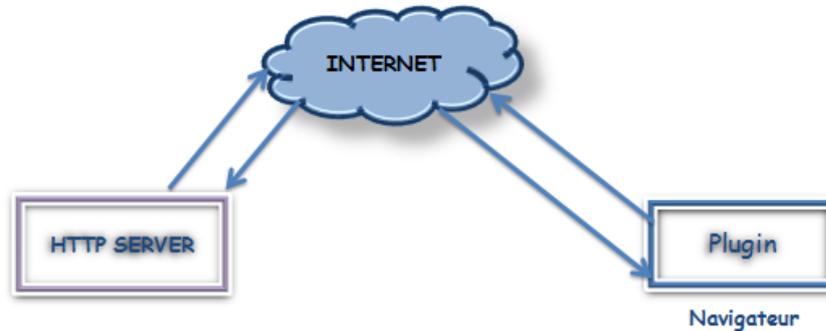


Figure. 1.13 Architecture 02

Du côté client, il doit y avoir une technologie capable de visualiser et traiter les images et l'afficher dans le navigateur. Une option serait d'utiliser les plugins offrant ainsi la possibilité de visualiser et de traiter les images. Cependant, il faut avoir des plugins pour chaque navigateur. Comme les plugins effectuent les traitements en utilisant le processeur et la carte graphique du client, alors le temps requis va dépendre toujours de la qualité du matériel du client.

### C. Architecture 03

Dans l'architecture 03, les modules de visualisation et de traitement sont intégrés dans le serveur HTTP. De cette façon, le traitement et la communication associés aux deux modules sont complètement intégrés dans le serveur. Ceci signifie qu'ils sont tous les deux en cours d'exécution dans le même langage de programmation ou en train d'utiliser des bibliothèques partagées. Ils sont aussi totalement dépendants les uns des autres.

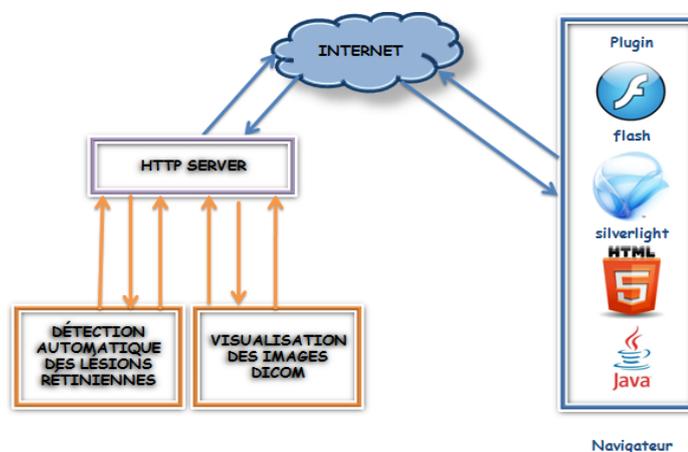


Figure. 1.14 Architecture 03

Par ailleurs, le langage de programmation doit être soigneusement sélectionné en considérant l'efficacité du pipeline processeur. Les langages tels que JavaScript, C + +, Java, Ruby, sont de bons candidats pour contenir les modules de visualisation de traitement et aussi travailler comme un serveur HTTP. Dans l'idéal, le serveur doit être un ordinateur de haute performance capable de se connecter à l'Internet à haute vitesse.

Ainsi qu'avec l'architecture n° 1, une technologie RIA (Application Internet Riche) est nécessaire sur le côté client. De même, le client peut exiger un plugin pour visualiser les images envoyées par le serveur. L'objectif principal est d'assurer le transfert de données en temps réel. Dans cette architecture, la totalité du traitement de l'image est exécutée sur le serveur. Du côté client, la procédure est plus simple du point de vue de la programmation, puisque le fait d'envoyer l'image à partir du serveur aux clients suffit pour afficher l'image dans le navigateur.

Les plugins effectuent les traitements en utilisant le processeur et la carte graphique du client, donc le temps dépend toujours de la qualité du matériel du client.

#### D. Architecture 04

L'idée principale de cette architecture consiste à faire le traitement d'images en utilisant une grille informatique à haute performance (Figure. 1.15). Ainsi, le processus de détection automatique des lésions rétiniennes est parallélisé afin de distribuer la charge de calcul dans des unités de calcul (CU) surtout dans le cas de la recherche où il faut faire le traitement de milliers d'images. Les unités de calcul doivent assurer une faible latence pour obtenir une meilleure performance. Chaque unité de calcul doit traiter une partie de l'ensemble des images.

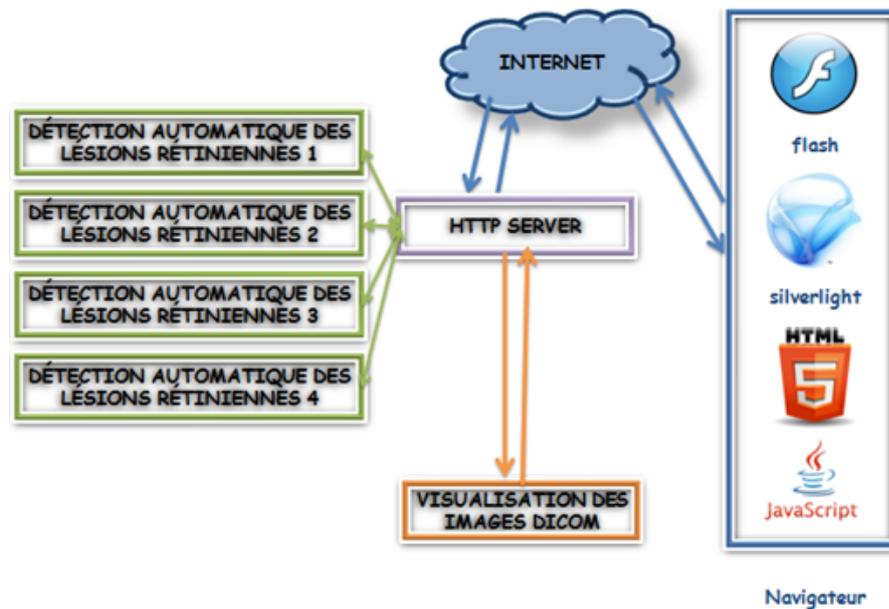


Figure. 1.15 Architecture 04

Par ailleurs, les UC doivent être gérées et contrôlées par le serveur HTTP ou un processus intégré. Le serveur doit s'occuper de la coordination entre tous les ensembles des images traitées correctement. Ces tâches peuvent générer une surcharge sur le serveur. Tous les ensembles d'images sont ensuite envoyés au navigateur, qui doit afficher les images finales.

Cette architecture propose une approche où le parallélisme de données est pertinent ; la cohérence des données et de la concurrence pourront être exploitées.

### 1.3.2 Technologie de collaboration

Le terme technologie de collaboration est utilisé pour décrire un ensemble d'outils flexibles intégrés qui prennent en charge trois grandes formes de collaboration-communication, la coordination et le partage de l'information [26].

Il y a plusieurs formes de la technologie de collaboration parmi eux :

- Des outils informatiques (logiciels, progiciels) ;
- Une base de connaissances structurée comprenant des guides ou méthodes de travail en groupe, pour améliorer la communication, la production, la coordination ;(Wiki)
- Un service de messagerie (ex. : messagerie rapide Peer to Peer) ;

- Un système de partage de ressources et de fichiers (Client/serveur et Peer to Peer) ;
- Des outils de type forum, pages de discussions, « chat » multi-utilisateurs, etc. ;
- Annuaire des profils des utilisateurs ;
- Des blogs, par projet ou par thématique ;
- Un système de vote ou de notation (rating) sur les articles ;
- Un calendrier ;
- Un système d'archive collective, et de pages personnelles ;
- Un index ou outil listant les tâches faites et à faire ;
- Des outils complémentaires (Audioconférence, conférence téléphonique, élaboration partenariale d'arbres heuristiques, etc.)

La messagerie instantanée est l'une des formes les plus importantes de la technologie de collaboration, Il est donc impératif de fouiller dans les concepts liés à la messagerie instantanée pour être en mesure de bien réaliser cette recherche.

### 1.3.3 La messagerie instantanée et présence

L'utilisation de la messagerie instantanée est récemment devenue populaire non seulement en privé, mais aussi dans le secteur d'activité. Lorsque les gens communiquent grâce à des systèmes de messagerie instantanée, un grand catalyseur de service doit être inclus: *Présence*. Sans connaître l'état actuel de l'autre extrémité (c'est-à-dire, ses informations de présence), le message instantané serait rien de plus qu'un Email ordinaire. Sachant que l'autre partie, en fait en ligne, permet des dialogues à base de texte et d'autres services instantanés.

Dans cette partie, nous présentons des techniques pour un espace de travail collaboratif dans un service de messagerie instantané. Les planches partagées fournissent divers outils de dessin pour les groupes de chat et des services de messagerie.

Nous avons opté pour le protocole de messagerie instantanée XMPP. Le protocole XMPP permet la création d'un système évolutif élevé. XMPP et SIMPLE (Session Initiation Protocol for Instant Messaging and Presence Leveraging Extensions) sont les deux protocoles de messagerie instantanée et de présence dominant, qui ont été en compétition au fil des années. Protocole SIMPLE est

apparu à la suite de la nécessité d'un soutien de messagerie instantanée et de présence dans SIP (Session Initiation Protocol). Des simples extensions définissent des méthodes de signalisation SIP pour gérer le transport des données et la présence. Dans cette bataille entre ces deux protocoles de messageries instantanées, XMPP est celui qui gagne du terrain dans le Web [27]. En effet, comme XMPP est basé sur XML il a bénéficié de tous les avantages de ce dernier, donc il peut être étendu avec de nouvelles fonctionnalités. XMPP a gagné du terrain en particulier dans le domaine des réseaux sociaux [27] et des applications de messagerie instantanée [27].

La principale raison qui a conduit à l'utilisation de XMPP est parce qu'il a tout pour devenir l'avenir des services de Cloud Computing. Il est un protocole riche qui permet la communication bidirectionnelle, avec des fonctionnalités important comme la fonctionnalité PUB-SUB, aussi il est basé sur XML ce qui tournent dans un protocole facilement extensible / flexible qui permet la création de systèmes personnalisés. Par exemple, il est couramment utilisé pour implémenter un middleware pour le système en temps réel [28] [29], profitant de XMPP pour les fonctions de communication en temps réel.

#### 1.3.3.1 Définition

XMPP est un protocole de messagerie instantanée standard ouvert, qui est défini par les RFC 6120, 6121, 6122 (qui remplacent depuis 2011 les 3920 et 3921), 3922 et 3923 du XMPP [27]. D'après l'analyse de RFC 6120, c'est possible de comprendre, comment est défini le protocole et comment il peut être une valeur ajoutée pour la messagerie en quasi-temps-réel.

XMPP est très variés, par conséquent, il est non seulement utilisé dans la messagerie instantanée, mais aussi dans de nombreuses applications qui mettent l'accent sur la voix sur IP, la collaboration en temps réel, réseaux sociaux, microblogging, middleware léger, Cloud Computing, etc. [27]. En raison que XMPP utilise XML pour l'échange de données entre le client et le serveur, il hérite l'extensibilité de XML. Par conséquence, cette souplesse / extensibilité est introduit sur de nombreux extensions de protocole qui sont disponibles pour le XMPP.

### 1.3.3.2 Architecture du XMPP

Le protocole XMPP n'est pas implémenté pour toute architecture réseau spécifique, mais il est couramment utilisé dans une architecture client-serveur, où les clients accèdent au serveur en utilisant le protocole XMPP avec une connexion TCP. En outre, il est possible de faire connecter des serveurs via une connexion TCP. Le XMPP définit trois principales entités qui fonctionnent dans un réseau XMPP, qui sont le serveur XMPP, le client et les composants.

- Serveur

Les serveurs est une entité de réseau XMPP qui fournit une voie d'échange de messages entre les clients. Le serveur permet l'échange d'informations avec d'autres serveurs XMPP via un protocole serveur ou avec les clients par le biais d'un protocole client-serveur. En particulier les serveurs sont le système de circulation pour tout réseau XMPP, par conséquent, ils ont pour mission de faire le routage des stanzas XMPP d'un utilisateur à un autre dans le même domaine ou d'un utilisateur local à un utilisateur sur un serveur distant.

L'un des atouts de XMPP vient de sa décentralisation, permis d'autoriser toute personne à utiliser un serveur XMPP. Il y a quelques serveurs qui prend en charge partiellement ou totalement la définition XMPP, il est disponible pour les systèmes Windows, Mac OS X ou Linux, par exemple, ejabberd, Openfire et Tigase [27], qui sont les serveurs les plus populaires de l'open source XMPP.

- Client

Les clients sont les entités qui se connectent avec le serveur en utilisant le protocole client-serveur. Habituellement ces entités sont des humains qui utilisent la messagerie instantanée traditionnelle (IM), mais ils peuvent être également des clients automatisé fonctionne comme des robots qui exécutent des services de détails. De plus pour accéder au réseau XMPP les clients doivent être authentifiés par un serveur XMPP. En outre, chaque client, après l'accès au serveur de son domaine, peut gérer certaines fonctionnalités telles que sa liste de contact, groupe, le contenu privé, etc. Ces éléments sont maintenus par le serveur, qui est

également responsable de la gestion de certains aspects de la session client comme, par exemple, l'échange de la présence, gestion des contacts, etc.

- Composants

Les composants sont utilisés pour gérer les services sous forme de modules d'extension pour le serveur XMPP. Ils sont la troisième classe d'entités qui peuvent se connecter à un serveur XMPP, outre les clients et les autres serveurs XMPP. Les composants, comme conversation à plusieurs utilisateurs, se connectent à un serveur et obtiennent un sous-domaine spécifique du serveur qui leur est assignée (par exemple, `groupchat.act.univ-blida.dz`). Lorsque le serveur XMPP reçoit une stanza adressée à un JID (Jabber Identifier) dans ce domaine, il fait directement le routage de la stanza au composant via la connexion des composants avec le serveur. La configuration d'un flux entre le composant et un serveur se fait en utilisant un protocole de Handshake simple, défini dans le protocole Jabber Component.

- Adressage

Dans le réseau XMPP, chaque entité est adressée par un ou plusieurs identifiants de Jabber, ou JIDs.

Les JIDs sont très similaires avec les adresses e-mail, par exemple, une entité peut être adressée à l'aide `entite@univ-blida.dz`. Pour envoyer n'importe quel données à une entité dans un réseau XMPP, la JID doit être utilisée pour identifier quel entité va recevoir le XML ainsi l'expéditeur doit être identifié par son JID.

- Domaines

XMPP comme d'autres services dans l'Internet utilisent le système de nom de Domaine (DNS) pour fournir la structure sous-jacente de l'adressage, Le domaine est présent dans l'adresse de toute entité au lieu d'utiliser le protocole Internet (IP).

Généralement, il est plus facile pour l'utilisateur de mémoriser qu'il y a un service disponible sur `univ-blida.dz` plutôt que ce rappeler de l'adresse IP du serveur. De plus, ce format utilise l'infrastructure complète du DNS comme espace d'adressage. Parce que XMPP utilise le DNS pour l'adresse du serveur dans un

réseau différent, il devient plus facile de gérer les serveurs en utilisant des adresses IP pour identifier chaque serveur.

Habituellement, chaque JID se compose de trois parties qui sont : une partie locale, le nom de domaine, et de la ressource. Cependant un JID qui se compose seulement par le nom de domaine est également un JID valide, et représente l'adresse du serveur XMPP sur ce domaine.

La combinaison d'un (utilisateur) partie locale et un domaine est appelé un "JID nu", et il est utilisé pour identifier un compte particulier dans un serveur exemple `Amine@univ-blida.dz`.

Un JID qui comprend une ressource est appelé un "JID complet", et il est utilisé pour identifier une connexion du client particulier (i.e., une connexion spécifique pour le compte associé " JID nu") exemple `user1@univ-blida.dz/labo`.

- L'utilisateur

L'identification de l'utilisateur dans un réseau de XMPP est faite en utilisant la partie locale de l'adresse, qui est séparée du reste de la JID par le caractère @, et par le domaine. Quand un utilisateur enregistre sur un serveur XMPP qui est dans un domaine spécifique, par exemple `univ-blida.dz`, il peut choisir le nom qui sera utilisé pour l'identifier dans le domaine du serveur. Par exemple, l'utilisateur peut choisir le nom de `user1` et il sera identifié comme `user1` dans le domaine `univ-blida.dz`, représentée par le JID `user1@univ-blida.dz`. Dans certains cas, l'adresse de l'utilisateur dans un réseau XMPP est attribuée automatiquement par intégration d'une adresse e-mail existante d'un service comme l'adresse de l'utilisateur dans le réseau. Un exemple de l'affectation automatique est arrivé sur GTalk, où les utilisateurs qui ont un compte Google mail peuvent accéder au réseau XMPP de Gtalk en utilisant leur adresse de courriel de Google, sans avoir d'enregistrer sur le service Gtalk.

- Resource

Un client peut avoir plusieurs connexions actives sur un réseau XMPP, donc chaque connexion identifiée est la ressource. De toute évidence, car un utilisateur peut avoir plusieurs connexions actives, la ressource sera utilisée pour router les messages XML vers la bonne connexion. Par exemple, l'utilisateur `User1` peut avoir un client qui est actif à son domicile et un autre client qui se trouve sur son lieu de

travail, donc cette connexion peut être identifiée comme ***user1@univ-blida.dz/maison*** et ***user1@univ-blida.dz/travaille*** respectivement. La ressource permet le routage des messages vers une connexion spécifique, par exemple, si un utilisateur veut communiquer avec ***User1*** que lorsque ***User1*** connecte avec le client qui est au ***user1@univ-blida.dz/maison***, le message est envoyé à cette ressource. Évidemment, s'il y a beaucoup de connexions associées à un utilisateur, chaque connexion a son propre état de présence, avec différents états de disponibilité, les capacités, etc.

#### 1.3.4 Flux XML

XMPP définit une technologie de flux XML. En outre, la diffusion de XML, entre les extrémités, ne se produit à travers un canal de communication crypté, en utilisant Transport Layer Security (TLS). En outre, avant de commencer à diffuser XML sur le flux créé, le client doit être identifié en utilisant un mécanisme qui se fait via l'authentification SIMPLE et couches de sécurité (SASL). Après que le client négocie le flux avec le serveur et établit la connexion, il peut échanger trois types spécial de XML, appelées XML Stanzas, à travers le flux. Ces Stanzas sont **<message >**, **<presence >** et **<iq >**.

Le flux XMPP est fondamentalement un document XML qui est construit de façon incrémentielle par le client et le serveur. L'élément racine du document est **<Stream: Stream>** et les éléments se composent des Stanzas routables qui sont échangés entre le client et le serveur.

Contrairement à la plupart des technologies Internet traditionnelles, comme le courrier électronique ou le Web, le protocole XMPP ouvre une connexion TCP long vie qui est utilisée par le client et le serveur pour échanger des Stanzas XML dans un flux XMPP. Les technologies traditionnelles comme le courrier électronique et le Web, utilisent une connexion transactionnelle, où le client fait une demande, attend la réponse du serveur et, après la recevoir, la connexion est fermée. En conséquence, le lien transactionnel ne convient pas pour les applications en temps réel, car il n'y a pas une connexion directe et longue entre le client et le serveur qui peut être utilisé par le serveur pour envoyer l'information vers le client. En revanche, le XMPP crée une connexion "Toujours ouvert" qui est utilisée par le client pour envoyer plusieurs demande sans être bloquer tout en attendant que le serveur

réponde, et le serveur répondra dès qu'il a une réponse qui satisfait la demande en utilisant la connexion TCP longue vie créer pour le flux des Stanzas XML. XMPP crée de nouveaux défis pour les développeurs qui sont généralement familiarisés avec un concept de connexion différente. Sur l'utilisation de XMPP, les développeurs doivent être en mesure de voir la connexion entre le client et le serveur comme un canal qui est utilisé pour faire circuler l'information asynchrone, sous forme de flux de Stanza XML.

#### 1.3.4.1 Stanzas XMPP

Comme il a été mentionné précédemment, le flux XML sur XMPP est basé sur trois types de Stanza XML: **<message >**, **<presence >** et **<iq >**. Pour chacune de ces primitives de communication, il existe des attributs qui définissent la sémantique de la primitive. En outre, la Stanza peut avoir un ou plusieurs sous éléments, qui définissent sa charge utile.

Il y a certains attributs qui sont communs à toutes les Stanzas XML. Ils sont utilisés pour déterminer comment la Stanza doit être acheminé et quel type de Stanza est en cours de traitement. Les attributs suivants ont la même signification pour l'ensemble des Stanzas XML:

- from

Il identifie le client qui envoie la Stanza.

- to

Il identifie le client de la destination de cette Stanza.

- type

Il peut prendre plusieurs valeurs, en fonction de la Stanza. Il est utilisé pour déterminer le type spécifique de Stanza: **<message >**, **<presence >** ou **<iq >**.

- id

Il est utilisé pour créer un moyen pour déterminer les réponses. Cet attribut est requis pour **<iq >** Stanzas, parce que habituellement **<iq >** demande de réponse, et cette identification peut identifier quel réponse d'une requête donnée. Cette identification est rendue possible grâce à tous les Stanzas de réponse ayant le même identifiant que la Stanza demande, qu'ils répondent.

### **La Stanza Message**

La Stanza **<message >** est utilisée pour envoyer des informations d'un endroit à un autre, et généralement elle utilise une méthode push qui n'a pas besoin d'accusé de réception. La Stanza **<message >** est basé sur un concept de tirer et oublier, il n'y a donc pas de haut-fiabilité. Après avoir envoyé un message, l'expéditeur n'a pas la confirmation que le message a été livré ou quand il a été reçu. Mais, si elle est nécessaire de recevoir l'accusé de réception, le protocole d'application peut être étendu en utilisant, par exemple, des messages Receipts définis dans XEP-0184 [30].

L'application de stanza de message est généralement associée à la messagerie instantanée, mais la portée de cette utilisation peut être vastes, il y a une application de ce type de stanza dans le transport de tout type d'information structurée. Par exemple, il peut être utilisé pour faire le chat en groupe, envoyer des notifications, alertes, pour transporter les instructions de dessin, de communiquer un état de jeu et de nouveaux mouvements de jeu, etc.

La stanza message peut être divisée en cinq types [27], qui se distinguent par le type attribué:

- normal

Les Messages de type normal sont plus semblables aux e-mails, car ils sont des messages unique qui peuvent avoir une réponse ou non.

- chat

Messages de type **chat** sont échangées en "session" temps réel entre deux entités, comme une conversation de messagerie instantanée entre deux amis.

- groupchat

Messages de type **GroupChat** sont échangés dans un environnement de salon de discussion multi-utilisateur, similaire à Internet Relay Chat (IRC)

- headline

Les Messages de type headline sont utilisés pour envoyer des alertes et des notifications, avec aucune prévision de réponse (un client qui reçoit Messages de type headline ne doit pas permettre à un utilisateur de répondre).

- error

Si une erreur se produit dans le cas d'un message envoyé précédemment, l'entité qui détecte le problème va retourner un message de type **error**. Par exemple, si une entité essaie d'envoyer un message à une autre entité qui existe ou non à un domaine que le serveur ne connaît pas, le serveur génère un message de réponse de type **error**.

## La Stanza Presence

La Stanzas Presence est utilisés pour faire des annonces de disponibilité sur le réseau XMPP. L'annonce de disponibilité est un catalyseur pour la communication et la collaboration dans l'Internet, car il permet aux gens de savoir si d'autres sont en ligne et disponible pour communiquer. Contrairement au service de messagerie, les utilisateurs ne sont pas en mesure de savoir si le destinataire vérifie les mails, avec la messagerie instantanée et les notifications de présence, les utilisateurs sont en mesure de réaliser que le destinataire est disponible avant d'envoyer le message.

Cependant, dans un réseau XMPP, il est nécessaire d'être autorisé de visualiser un état de présence de l'entité. Cette autorisation est recueillie par un handshake entre les entités, elle s'appelle la souscription de la présence. Pour faire

l'a souscription il y a certains types de Presence spécialisés, représentés dans le tableau 1.2

Type	Description
Subscribe	L'expéditeur souhaite souscrire à la présence du destinataire.
Subscribed	L'expéditeur a permis au destinataire de recevoir sa présence.
Unsubscribe	L'expéditeur se désabonne de la présence d'une autre entité.
Unsubscribed	La demande de souscription a été refusée ou une souscription précédemment accordée a été annulée.

Tableau 1.2: Les types de XMPP Presence utilisés dans l'abonnement à la présence.

```
<presence from='bob@univ-blida.dz/université ' to='alice@univ-blida.dz ' type='subscribe ' />
<presence from='alice@univ-blida.dz/Enligne ' to='bob@univ-blida.dz/université ' type='subscribed ' />
<presence from='alice@univ-blida.dz/Enligne ' to='bob@univ-blida.dz' type='subscribe ' />
<presence from='bob@univ-blida.dz/bureau' to=' alice@univ-blida.dz/Enligne ' type=' subscribed ' />
```

Le flux XML suivant, représente un exemple de la souscription de la présence réciproque entre Bob et Alice.

La présence XMPP de L'abonnement de est une méthode spécialisée de notification de souscription, dans lequel l'entité abonnés recevront des mises à jour des informations de présence quand l'entité est en ligne, modifie le statut à «en réunion» ou «disponible pour discussion" ou bien déconnecté.

La XMPP Presence de base peut notifier si un utilisateur est en ligne ou hors ligne. Mais Le noyau XMPP fournit des présences prolongées qui introduisent d'autres Etats comme "Absent", "Ne pas déranger" et "absence prolongée". Cette Presence avancée est définie en utilisant un **<show >** comme une charge utile de

Stanza Presence. Le **<show >** peut prendre les valeurs représentée dans le tableau suivant.

Valeur	Description
away	L'entité ou la ressource est temporairement absent.
chat	L'entité ou la ressource est intéressé au chat.
dnd	L'entité ou la ressource est occupée
xa	L'entité ou la ressource est absent pour une longue période

Tableau 1.3: Les valeurs d'un XMPP Presence avancée

La Stanza Presence du XMPP peut être enrichie par des messages d'état personnalisés comme, par exemple «*Je rédige ma thèse, ne me dérange pas maintenant*».

Pour actualiser l'état de présence, l'entité doit envoyer une Stanza **<presence >** contenant des informations sur son nouveau statut. Le XML suivant, montre un exemple d'une Stanza Presence

```
<presence from='bob@univ-blida.dz/mobile '>
  <show>xa</show>
  <status> à la bibliothèque </ status>
</ presence>
```

.A côté de ces types de Stanzas Presence qui sont utilisés pour faire la souscription ou le désabonnement de la présence de l'utilisateur il y a d'autres types de Stanzas Presences avec certaines fonctions spécifiques dans le protocole XMPP (tableau 1.4):

Type	Description
unavailable	Signale que l'entité est plus disponible pour la communication.
probe	Une demande de présence actuelle d'une entité.
error	Une erreur est survenue dans le traitement ou la livraison d'une Stanza Presence envoyé précédemment.

Tableau 1.4: Les types de Stanza Presence de XMPP avec des fonctions spécifique.

### La stanza IQ

La Stanza Info/Query, ou généralement connu comme le **IQ**, est utilisés pour créer une interaction requête-réponse et implémenter un simple flux de travail, similaires aux méthodes de HTTP, GET, POST et PUT. Cette Stanza permet à la fois d'obtenir et mettre des informations d'un destinataire ou d'un serveur. Chaque Stanza IQ envoyé doit recevoir une réponse avec le même identifiant.

Contrairement à la Stanza `</ message>` La stanza `<iq >` ne peut contenir qu'un seul élément comme sa charge utile. L'élément de la donnée utile définit la demande qui doit être traitée ou l'action qui doit être exécuté par le destinataire. Finalement La Stanza `<iq >` existe en quatre types différenciés par le type d'attribut de la Stanza. Ces quatre types de Stanza IQ peuvent être divisés en deux groupes (tableau 1.5)

	Type	Description
Demande	get	Identifie l'intention d'obtenir des informations du destinataire.
	set	Identifie l'intention de mettre des informations pour le destinataire.
Réponse	result	il est utilisé pour répondre au succès de l'action et effectué par un IQ-set ou IQ-get.
	error	Ce type est utilisé pour signaler une erreur générée par l'exécution d'une action associée à un <b>IQ set</b> ou <b>IQ get</b> .

Tableau 1.5:Les types de Stanza I/Q

### 1.3.5 L'Extensibilité

Une stanza XML peut contenir n'importe quel nombre de **child elements**, y compris les messages body formaté en XHTML, des pointeurs vers des URL, RSS, les données SOAP pour le web services ou XML-RPC, les emplacements géographiques, et un grand nombre d'autres charges utiles. (Le "X" dans XML et XMPP signifie "extensible", donc les types de charge utile sont limitées que par le développeur). Des dizaines d'extensions de la couche Stanza du noyau XMPP sont développées par la communauté des développeurs de XMPP, ces extensions sont publiées sur le site officiel par *XMPP Standards Foundation*. donc il est possible de définir des extensions privées pour des fonctions personnalisées.

XMPP bénéficie d'espaces de noms XML comme une charge utile des Stanza, donc l'application est adaptée en utilisant le nom de l'élément et son espace de noms. Un exemple de l'extension XMPP est le stockage XML privée définie par XEP-0049 [31]. Cette extension est utilisée pour stocker des données XML sur un serveur privé XMPP. Il est identifié par l'espace de noms **jabber: iq: privée** et la charge utile de child element **query** (`<xmlns query = 'jabber: iq: privée'>`). Ainsi, pour créer une extension à une Stanza XMPP, les développeurs doivent définir un

nom de l'élément, par exemple **<telemed >** et un espace de noms à la portée de cette extension comme, par exemple, ***custom: iq: telemed.***

# CHAPITRE 2

## ARCHITECTURE ET APPROCHES DE LA PLATEFORME

L'objectif principal de ce mémoire est de créer un système qui permet aux médecins de partager des images médicales et d'améliorer la télémédecine et le télétravail en explorant une nouvelle approche pour les environnements de coopération sur le Cloud. L'optimisation du partage de l'information permet aux médecins d'obtenir un deuxième avis de ceux qui sont spécialisés.

Dans ce chapitre, il sera présenté les exigences qui ont été considérés pour mettre en œuvre un SaaS qui mène à l'amélioration des flux de travail et les flux de données de télémédecine.

### 2.1 Les exigences du système

Au début, ont été identifiées quelques caractéristiques qui pourraient améliorer les processus de télétravail.

Ces caractéristiques ont été établies afin de faciliter le partage des images médicales entre les médecins, en tenant compte des besoins des utilisateurs et comment un système basé sur le Cloud pourrait être utile pour enrichir la productivité des utilisateurs en améliorant le flux de travail dans le télétravail.

Dans la télémédecine, il y a une nécessité de lier les gens ainsi que les machines. A partir de cette nécessité, le concept de réseaux sociaux a émergé pour connecter les collègues. La création d'une plateforme qui prend en charge les réseaux sociaux peut être une base importante pour la création d'une communauté virtuelle en médecine. Ainsi, ce travail avait également pour but d'intégrer le concept des réseaux sociaux, dans un service de télémédecine, qui peuvent être utilisés par des radiologues. Par conséquent, les utilisateurs peuvent profiter du système

proposé pour leur permettre d'interagir les uns avec les autres d'une manière synchrone ou asynchrone, créant ainsi un environnement favorable au travail coopératif.

Le travail coopératif est basé sur le partage des images médicales entre collègues, permettant aux utilisateurs d'obtenir un deuxième avis sur un examen donné, partager des cas cliniques intéressants, mettre en œuvre des rapports de service, etc. Grâce à ce service, à la suite d'un examen, un médecin peut annoter une image médicale donnée, y compris le texte, le son et dessiner, et d'interagir avec ses collègues à travers l'utilisation des réseaux sociaux et des fonctionnalités telles que, par exemple, la messagerie instantané.

Le télétravail nécessite généralement l'accès à l'information qui est stockée dans des serveurs distants à l'intérieur des établissements de soins de santé. Un système qui vise à créer des services pour le télétravail, doit le prendre en compte. Ainsi, le système proposé a été défini de telle façon qu'il permet à des utilisateurs l'accès à distance à des dossiers. Cette fonction a été définie comme une condition importante pour le système. Il permettra aux utilisateurs d'accéder à des images médicales stockées dans un PACS à distance, aussi il va permettre à un utilisateur de partager un dossier ou plusieurs avec d'autres collègues, afin d'améliorer l'environnement du réseau social et de la coopération.

Évidemment, il faut mettre un mécanisme pour contrôler l'accès sur le contenu lorsqu'un utilisateur partage ses dossiers PACS avec d'autres collègues. L'accès à ces dossiers PACS doit être contrôlé par une liste d'accès où les utilisateurs non autorisés ne peuvent pas accéder à l'information. Ainsi, un utilisateur peut avoir plusieurs dossiers PACS disponibles dans le système; par conséquent, il doit gérer plusieurs listes d'accès, comme il doit gérer qui peut accéder à un serveur PACS donné. Outre le contrôle de la liste d'accès, il devrait être possible de définir si l'information est partagée en anonymat ou non.

La création d'un service basé sur le Cloud pour la télémédecine devient un atout important si elle permet aux utilisateurs d'accéder à leurs informations de n'importe où et à tout moment. Cela peut permettre le stockage d'informations dans le Cloud. Donc, notre système proposé a été conçu pour permettre aux utilisateurs

de stocker des images médicales dans le Cloud. La solution permet aux utilisateurs de simplement stocker leurs fichiers dans leurs archives personnelles, il est possible de partager des images médicales avec des amis et améliorer le flux de travail de la télémédecine.

Construire le service dans le Cloud permet aux fichiers d'être disponible via le Web pour les utilisateurs partout où ils sont, dans leur maison, au bureau, etc. Suivant le concept qui a conduit à la création de certains services comme Dropbox, SkyDrive et Google Drive, le service proposé permet également aux utilisateurs de synchroniser des fichiers de leur archivage à distance avec un poste de travail donné.

Avec toutes les informations recueillies à partir des images médicales référencées dans le PACS et les fichiers personnels stockés dans le Cloud, le système permet aux utilisateurs d'accéder et de manipuler ces informations, simplement en utilisant un navigateur Web, en fournissant des mécanismes pour visualiser des images médicales dans une application Web et joindre les renseignements au sujet de l'examen d'une étude donnée.

## 2.2 Spécifications fonctionnelles

À partir des exigences présentées dans la dernière section, ont été tirées les spécifications fonctionnelles qui sont définies dans le tableau 2.1.

Spécifications	Description
<b>PASSERELLE PACS</b>	
Configurer l'accès Web au serveur PACS	La passerelle devrait agir comme une interface qui doit permettre aux clients web d'interagir avec l'infrastructure PACS.
Activer les requêtes sur le serveur PACS	La passerelle doit recevoir les messages des clients Web et de les transformer en DICOM C-Find, permettant aux clients de recherche sur le serveur PAC

Activer la récupération des données à partir du serveur PACS	Il devrait être possible à un client Web de demander à la passerelle de récupérer des études données à partir du serveur PACS et de les stocker dans un dossier sur le Cloud.
<b>APPLICATION WEB</b>	
Envoyer des fichiers vers le Cloud	L'utilisateur sélectionne une liste des fichiers sur l'ordinateur local et envoient ces fichiers à son dossier personnel dans le Cloud
Parcourir Le répertoire Personnel	Le système doit permettre à l'utilisateur de naviguer dans les dossiers qu'il a dans son répertoire personnel.
Parcourir le répertoire partagé	Il doit être possible à un utilisateur de parcourir les fichiers partagés par d'autres collègues.
Supprimer un fichier du répertoire	Les utilisateurs doivent être en mesure de supprimer les études à partir de leur répertoire personnel
Synchroniser des fichiers avec le poste de travail	Il doit être possible aux utilisateurs de synchroniser son dossier dans le Cloud avec un poste de travail donné.
Partager un dossier avec des amis	Les utilisateurs doivent être capables de partager tous les fichiers d'un dossier donné qui représente une institution, patient, étude ou série avec un ami.
Voir et manipuler des images médicales	Le système devrait permettre aux utilisateurs de visualiser et de manipuler des images médicales en utilisant certains outils comme, par exemple, le zoom, contraste, encadrer les zones, etc.
Visualisez la présence des utilisateurs	Il devrait être possible en utilisant le service de connaître l'état de présence d'un ami donné.
Discuter avec des collègues	Le système doit permettre à l'utilisateur de communiquer avec ses collègues.
Visualisez son statut de PACS	L'utilisateur doit être en mesure de voir si passerelle PACS soient en marche ou en arrêt
Partager l'accès au PACS avec des amis	Il devrait être possible de donner l'accès au PACS à un collègue.
Visualisez l'état des accès partagé	Les utilisateurs devraient voir si les accès au PACS, qui ont été partagées par des collègues, sont actifs ou non.

Tableau 2.1: Liste des besoins fonctionnels.

Les principaux cas d'utilisation du système sont représentés dans la figure 2.1, où nous pouvons voir comment l'utilisateur peut interagir avec l'application Web et le PACS.

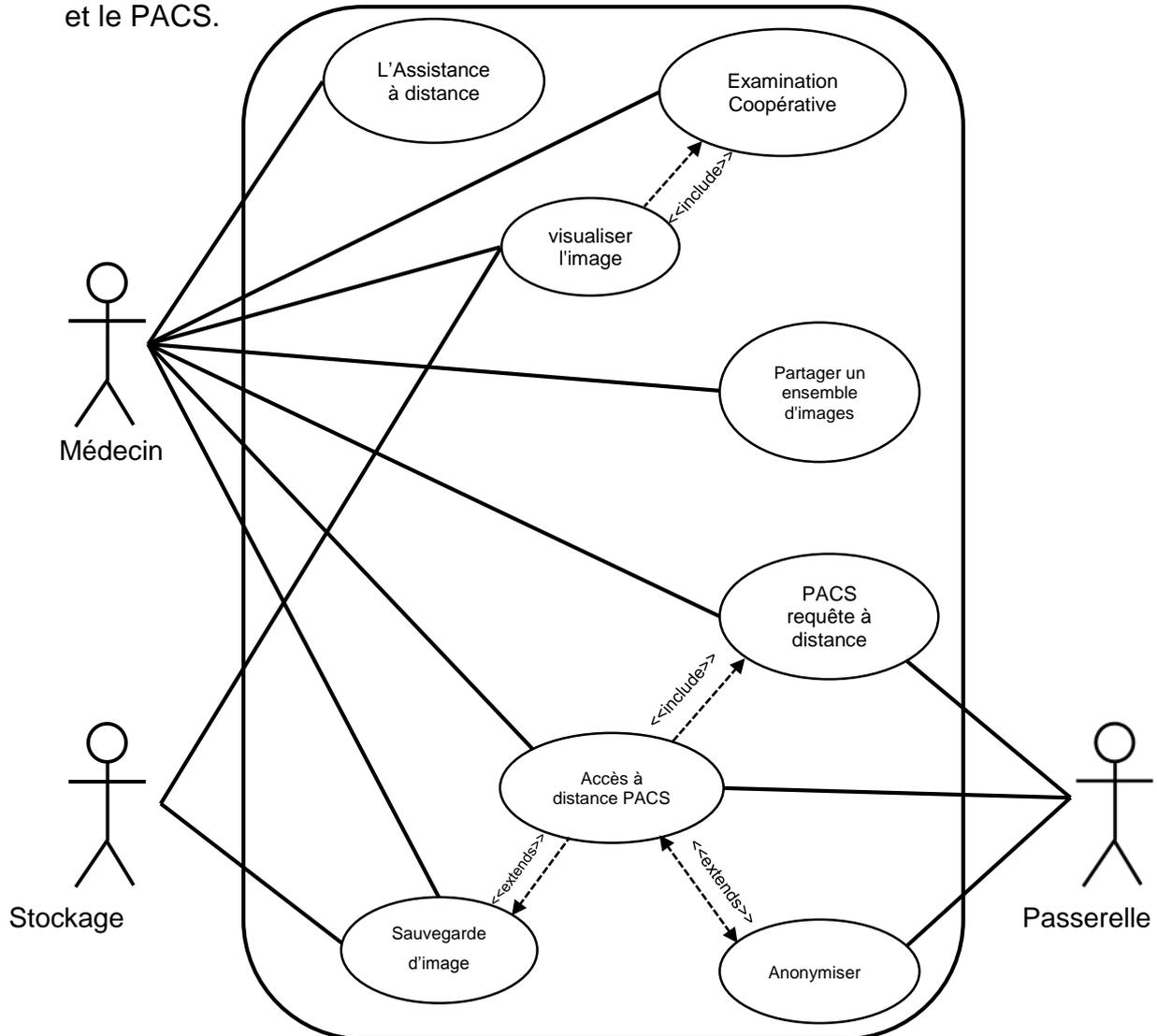


Figure 2.1: Cas d'utilisation du système

## 2.3 Les besoins non fonctionnels

### 2.3.1 Accessibilité et disponibilité des informations

Dans les exigences du système, on a démontré que chaque fois que le médecin tente d'y accéder, le service devrait être accessible partout avec une haute disponibilité. Cette caractéristique est très importante dans les scénarios de télémédecine où le flux de l'information ne peut être interrompu. Ainsi, la disponibilité est un facteur important pour créer un système efficace, qui prend en

charge un service essentiel qui devrait être toujours disponible pour répondre aux besoins du système de la santé.

### 2.3.2 Sécurité de l'information

En outre, l'information qui circule dans les services de télémédecine est très sensible, de sorte que le système doit être mis en œuvre en prenant en compte cette question liée à la sécurité, notamment l'authentification des utilisateurs, l'intégrité des données et la confidentialité.

### 2.3.3 Fiabilité du service

Aussi, quand un système manipule des données médicales, il doit être développé avec quelques préoccupations concernant la fiabilité des données, des mécanismes doivent donc être créés pour empêcher les données d'être perdues ou volées, en raison de leur nature.

### 2.3.4 L'évolutivité du service

En outre, le système doit être conçu en tenant compte de son extensibilité, afin d'être facile d'améliorer ses capacités de calcul et de stockage pour gérer les exigences élevées qui paraîtront à l'avenir. Le système devrait fournir une architecture qui permet l'extension horizontale, avec mise en place de nouveaux serveurs pour gérer l'équilibrage de charge et d'autres stockages.

Les quatre exigences présentées (accès aux données et la disponibilité, la sécurité, la fiabilité et l'évolutivité) peuvent être atteintes en prenant avantage du concept Cloud Computing. Quand un système est développé sur la base du Cloud Computing, il hérite directement des caractéristiques du modèle de Cloud Computing, comme l'accès au système n'importe où et à tout moment, et aussi la fiabilité. Ces caractéristiques sont généralement définies dans les termes de la SLA du contrat (service level agreement). Le plus grand soin doit être pris avec la sécurité de l'information. Quoi qu'il en soit, un système basé sur le Cloud ne peut jamais améliorer la sécurité des données, donc il faut chiffrer les informations qui sont stockées dans le Cloud.

## 2.4 Modélisation du système

Dans la section suivante le système sera conceptualisé selon les exigences identifiées dans les sections précédentes. Tout d'abord, il sera présenté le modèle fonctionnel du système proposé, y compris tous les blocs fonctionnels, et l'interaction entre eux. Après, dans la section Modèle de données, il sera présenté un modèle de données de haut niveau du système, qui souligne les données nécessaires créées par des processus. Enfin, il sera présenté comme résultat l'architecture du système qui intègre tous les composants nécessaires à la construction du système souhaité.

### 2.4.1 Le Modèle fonctionnel

Le modèle fonctionnel du système a été créé à partir de l'identification de tous les composants clés d'une analyse descendante des exigences. On a réalisé une décomposition fonctionnelle, qui divise le système d'une manière telle que chaque bloc identifié soit auto descriptif. Par conséquent, il était possible d'identifier plusieurs processus généraux qui composent le système, provenant à partir des exigences du système. Les processus clés de ce système sont:

Le stockage de l'image médicale dans le Cloud; l'accès à ces fichiers pour examen; le partage de ces images avec des collègues; l'accès à des fichiers du PACS à distance; le partage du PACS à distance avec des collègues; permettre une collaboration active.

#### 2.4.1.1 Le système middleware orienté Message

La modélisation du système a été fortement influencée par les principaux processus métier, donc au stade de la modélisation, il était nécessaire de prendre en compte certaines caractéristiques liées à chaque processus pour une meilleure définition d'une approche basé sur l'implémentation de ces caractéristiques et la façon dont ils sont directement ou indirectement influencés les uns avec les autres.

Les exigences du système nécessitent que le système fournisse des fonctionnalités pour permettre la collaboration entre les utilisateurs et, en outre, permettre aussi l'accès à l'information à partir de l'infrastructure PACS à l'intérieur d'un établissement de soins de santé. Toutefois, cet accès doit être fait par une

passerelle, l'utilisateur doit être conscient de son état actuel (actif ou inactif). Donc, à l'étape de modélisation, il était nécessaire de définir que le système fournisse des mécanismes pour permettre de passer le pare-feu et notifier la présence de la passerelle. De plus, associé à la nécessité de permettre à l'utilisateur de discuter avec des autres, notification de la présence des collègues, et de créer un moyen souple pour accéder au serveur distant PACS, conduire à modéliser un système basé sur un middleware orienté message. En conséquence, il a été choisi d'utiliser *Extensible Messaging and Presence Protocol* (XMPP), car il possède une forme comme un gant dans le service proposé basée sur le Cloud.

Le XMPP permet de mettre en œuvre certaines fonctionnalités comme, par exemple, le chat et la notification de présence. En outre, il est possible de contourner le problème lié avec les passerelles et les clients situés derrière un pare-feu, parce que le système va utiliser un serveur XMPP dans le Cloud, qui agira comme un relais pour les messages échangé entre les clients XMPP. De façon plus général, il y a d'autres caractéristiques qui ont conduit à l'utilisation de XMPP, surtout il est basée sur des normes ouvertes pour mettre en œuvre une technologie de messagerie et de présence, aussi il est largement utilisé et soutenu par des entreprises comme Google, Facebook, Apple, IBM, etc... par ailleurs, XMPP est décentralisée, afin que toute personne peut gérer son propre serveur XMPP et il est possible de se connecter avec d'autres serveurs XMPP (Figure 2.2). Par conséquent, XMPP est un moyen pour créer un système hautement évolutif. Par exemple, l'accès à Gtalk de Google est dimensionné à des millions d'utilisateurs simultanés sur un seul service.

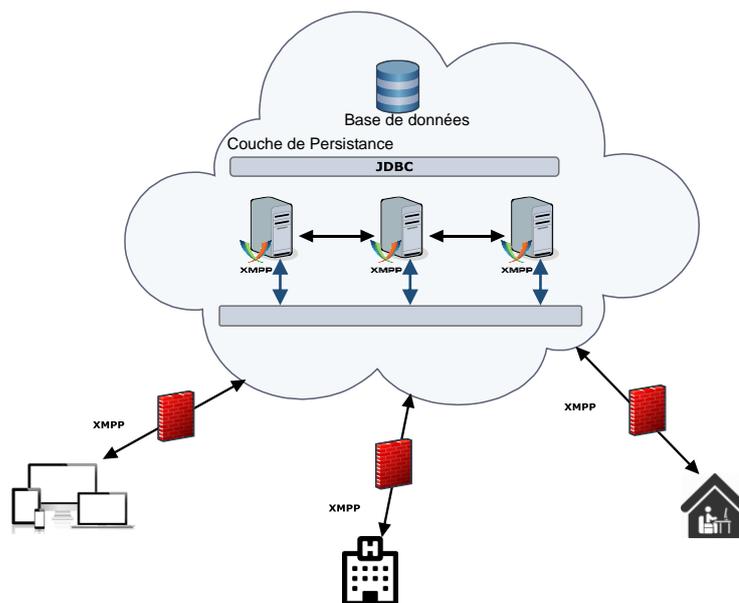


Figure 2.2: Infrastructure XMPP évolutive avec de multiples serveurs. La communication de serveur à serveur dans un réseau XMPP permet d'équilibrer la charge entre les serveurs.

L'architecture du réseau XMPP est hautement évolutive, car il est possible de déployer plusieurs serveurs XMPP et les faire communiquer entre eux en utilisant un protocole de serveur à serveur. L'utilisation de plus d'un serveur XMPP dans le réseau XMPP est un atout lorsque le réseau de la charge d'un seul serveur devient dans certaine manière insupportable. Ainsi, dans le réseau XMPP où il y a plusieurs serveurs pour gérer les connexions des utilisateurs, il est possible d'utiliser un équilibreur de charge pour répartir la charge équitablement par chaque serveur.

Dans les préoccupations de sécurité, XMPP fournit un protocole de sécurité robuste en utilisant SASL (Simple Authentication and Security Layer) et TLS (Transport Layer Security). La sécurité dans le protocole XMPP est une préoccupation qui a été pris en compte depuis le début, et a été défini dans les spécifications de base du XMPP.

XMPP est basé sur XML, en conséquence c'est un protocole facilement extensible / flexible, qui permis la création ses services personnalisés du Cloud Computing. Par exemple, chess [32] est un service qui intègre un jeu d'échecs dans un environnement de réseau social, profitant de la fonction de communication en temps quasi réel de XMPP, qui convient parfaitement dans un jeu qui est souvent rapide comme les échecs.

La flexibilité de XMPP est très importante pour notre système, car elle nous permet de mettre en œuvre un protocole personnalisé ([section 3.1.1](#)) pour interagir avec le PACS.

#### 2.4.1.2 Dissocier le cœur de l'infrastructure et le stockage Cloud

Modélisation d'un système pour qu'il soit déployé dans le Cloud est une étape très importante de sa conception. En raison du grand nombre de fournisseurs de Cloud qui apparaissent de nos jours, il est très important de concevoir un système en épousant une approche basée sur un fournisseur de Cloud qui supporte le découplage.

Donc, à ce stade, la conception du système doit être conceptualisée tenant compte de l'évolution technologique rapide et l'apparition de nouveaux fournisseurs de Cloud offrant un meilleur service à moindre coût.

Sans aucun doute, dans la modélisation de notre système le découplage Cloud est considéré comme modèle, car notre système est conçu pour être déployé chez tous les fournisseurs de Cloud. En outre, le découplage sépare l'infrastructure de base du Cloud, qui est déployé dans un IaaS, de l'infrastructure de stockage en Cloud. Ce découplage de calcul et de stockage permet au système d'être mis en œuvre dans plusieurs configurations qui cherchent le meilleur capital et le minimum de dépenses opérationnelles. Le découplage est archivé à l'aide de modèles de conception d'interface bien définis, où il est possible de définir une interface utilisée pour brancher l'infrastructure de base à l'infrastructure de stockage (Figure 2.3). En découplant l'infrastructure de calcul du noyau para port au stockage, il devient possible d'utiliser des fournisseurs différents pour le stockage et le calcul.

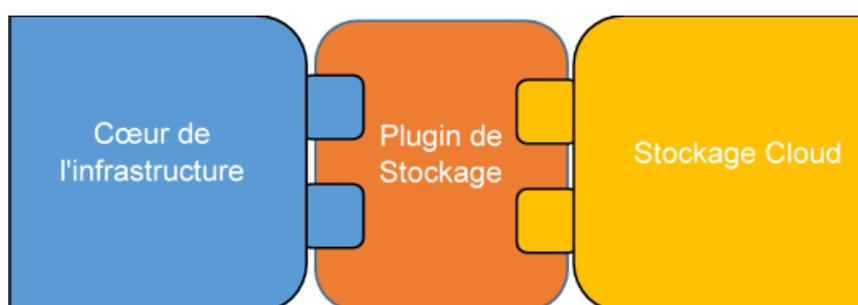


Figure 2.3: Schéma plugin de stockage.

Le cœur de l'infrastructure nécessite un composant de stockage qui implémente une interface donnée. Le composant de stockage peut être mis en œuvre pour utiliser n'importe quel fournisseur de stockage Cloud.

En théorie, si par l'audit d'un service de stockage Cloud public a été déterminé que les niveaux de sécurité de l'information, la vie privée et la fiabilité ne répond pas aux exigences du système, il est possible de déployer le système sur un Cloud hybride en utilisant des fournisseurs Cloud public pour déployer l'infrastructure de base et une infrastructure Cloud privé pour répondre aux exigences de la sécurité de données du système comme il est représenté sur la figure 2.4.

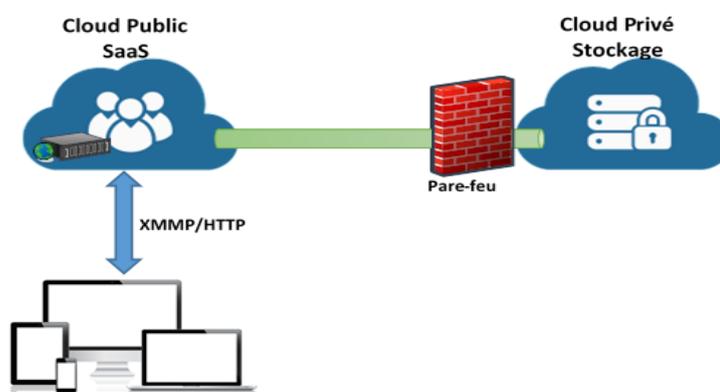


Figure 2.4: Sauvegarder les données des images médicales dans un Cloud privé. Toutefois, le service est disponible comme un SaaS dans un Cloud public.

Cependant, pour prouver ce concept dans ce travail, une infrastructure Cloud publique a été utilisée où les deux l'infrastructure de base et l'infrastructure de stockage sont déployés, tel que représenté dans la figure 2.5.

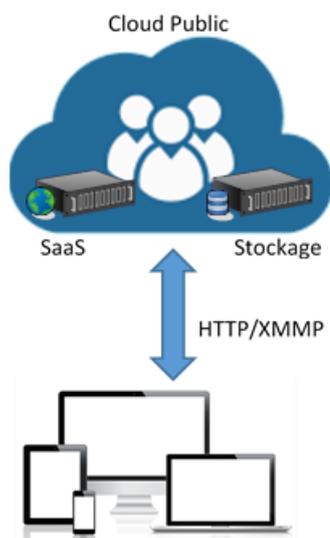


Figure 2.5: Déploiement dans un Cloud public le stockage et le service

#### 2.4.1.3 Résumé du répertoire privé sur le Cloud

Le système fonctionne comme Google Drive, Dropbox et SkyDrive, il propose un répertoire privé dans le Cloud où les utilisateurs peuvent stocker leurs données et y accéder à partir de n'importe où. En conséquence, il est important de modéliser les processus métier de base qui consiste à définir la façon avec laquelle les utilisateurs stockent les données dans le Cloud et comment ils peuvent manipuler ces données. Lors de la modélisation de processus on considère que chaque utilisateur dispose d'un répertoire privé dans le service fourni, donc tous les fichiers qu'il stocke dans le Cloud sont lié automatiquement avec leur répertoire privé, qui lui seul peut accéder à moins qu'il partage une étude spécifique avec quelques amis.

La création d'une application Web qui permet à l'utilisateur d'interagir avec le système est une exigence; cette application Web doit mettre en œuvre une partie de l'abstraction utilisée dans une application de bureau pour améliorer l'expérience utilisateur. A savoir, quand un utilisateur stocke des images médicales dans le

Cloud, cette action peut être associée à l'événement déclenché par l'utilisateur quand il fait glisser et dépose une liste de fichiers dans le navigateur.

La manière dont le système gère l'action déclenché dans un répertoire privé spécifique est représentée par le schéma de la figure 2.6. En outre, dans le but d'atteindre une meilleure performance et une réduction de charge de traitement dans le serveur, des prétraitements et une modélisation soient prises en considération. Par conséquent, il est nécessaire de créer un fichier JPEG de haute qualité (qui est utilisé dans la visualisation) et un fichier JPEG de qualité inférieure (utilisé dans l'application web comme des vignettes de visualisation) et lorsque le système exécute une requête donnée qui doit accéder à une image, il n'est pas nécessaire de faire la conversion au chargement. En outre, pour améliorer les performances dans le domaine de recherche des images médical, des informations extraites à partir de l'en-tête de l'image DICOM sont indexés dans une base de données SQL, permettant aux utilisateurs de soumettre des requêtes pour sélectionner une étude ou d'une série donnée.

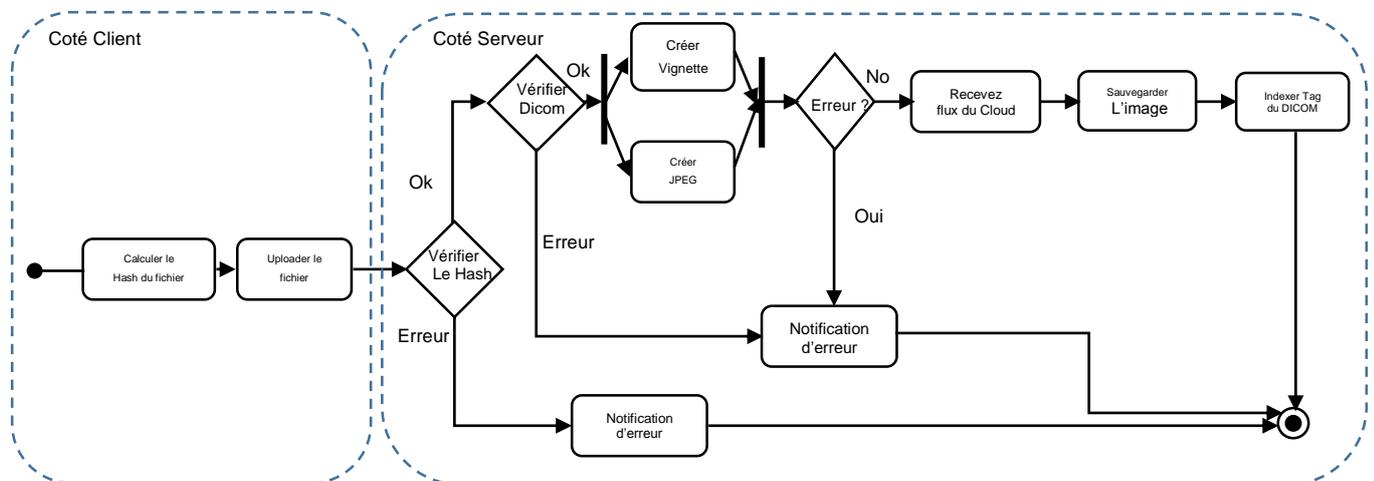


Figure 2.6: Organigramme du stockage des images médicales dans le Cloud.

L'en-tête des images DICOM permet au système d'obtenir des informations de l'institution, patient, l'étude et la série d'une image donnée. Ces informations identifient un arbre hiérarchique structuré, qui représente les informations du monde réel (Figure 1.3). Donc, le système conserve ces informations indexées, visant à permettre à l'utilisateur d'identifier facilement la relation entre les institutions, les patients, les études, les séries et les images médicales et, lui permettant d'exécuter des requêtes pour récupérer les données souhaitées. Si un utilisateur veut partager

un objet donné qui représente une série, une étude ou un établissement, avec un collègue donné, il doit effectuer une action (Figure 2.7) qui indexe ce partage dans la base de données du système, et donc, son collègue devient en mesure de visualiser et de faire un examen sur les images appartenant à cet objet.

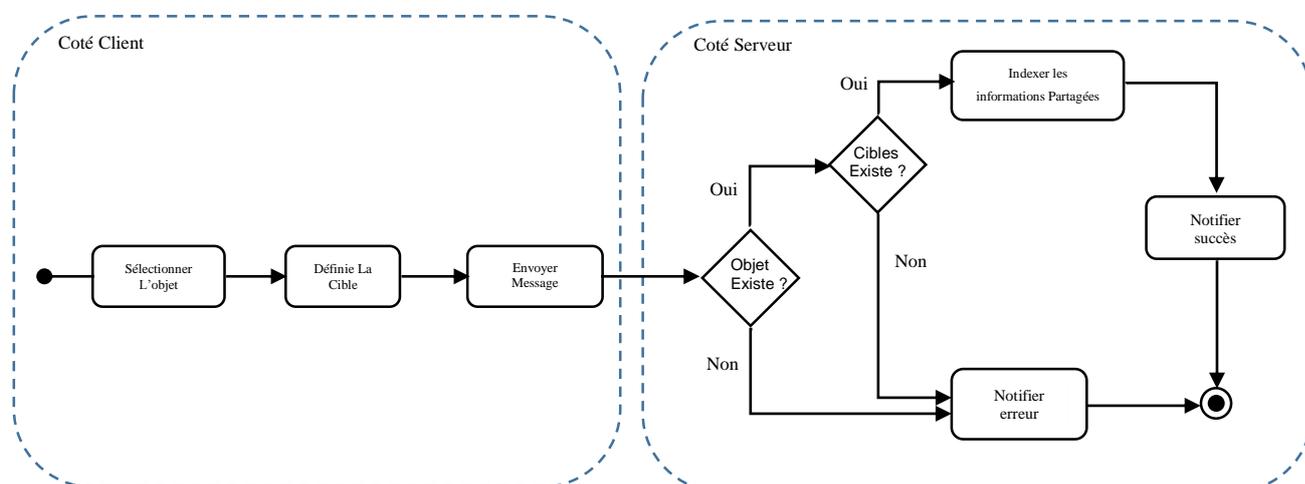


Figure 2.7: Diagramme d'un dossier partagé. L'objet peut être une institution, étude ou une série donnée, et la cible de l'action représente un utilisateur qui est enregistré dans le système.

#### 2.4.1.4 Accès à distance aux PACS

Les exigences du système définissent que ce dernier doit être modélisé afin de rendre possible aux utilisateurs de récupérer série / études à partir d'un serveur de PACS et de stocker ces images médicales dans le répertoire des utilisateurs dans le Cloud.

L'accès à distance à un serveur PACS est une exigence fonctionnelle qui est modélisé à l'aide d'une passerelle pour établir la communication entre le système et le serveur d'archives PACS. Cette fonction est un atout pour le système désiré. Par conséquent, en permettant aux utilisateurs d'accéder à distance aux serveurs d'archivage PACS, les utilisateurs deviennent capables de rechercher et de récupérer des informations à partir des répertoires distants qui sont à l'intérieur du système d'information de l'institution de soins de santé.

De toute évidence, on peut dire que le système est modélisé dans le but d'atteindre l'informatique ubiquitaire (omniprésente) désiré, qui est en train de devenir une caractéristique importante dans les systèmes d'information contemporains. En utilisant le paradigme de l'informatique ubiquitaire, la plateforme

fournit un système d'information intégré dans les activités quotidiennes des utilisateurs. Ainsi, la création d'un système d'informatique omniprésent où l'information est disponible partout, que ce soit à la maison, au travail ou sur déplacement.

Pour contourner les problèmes liés aux pare-feu qui peuvent bloquer l'accès d'un peu partout aux PACS ( qui ne sont pas à l'intérieur de l'institution), un relais compatible Web 2.0 basée dans le Cloud a été modélisé, en utilisant un middleware orienté message pour connecter tous les composants du système (Figure 2.8).

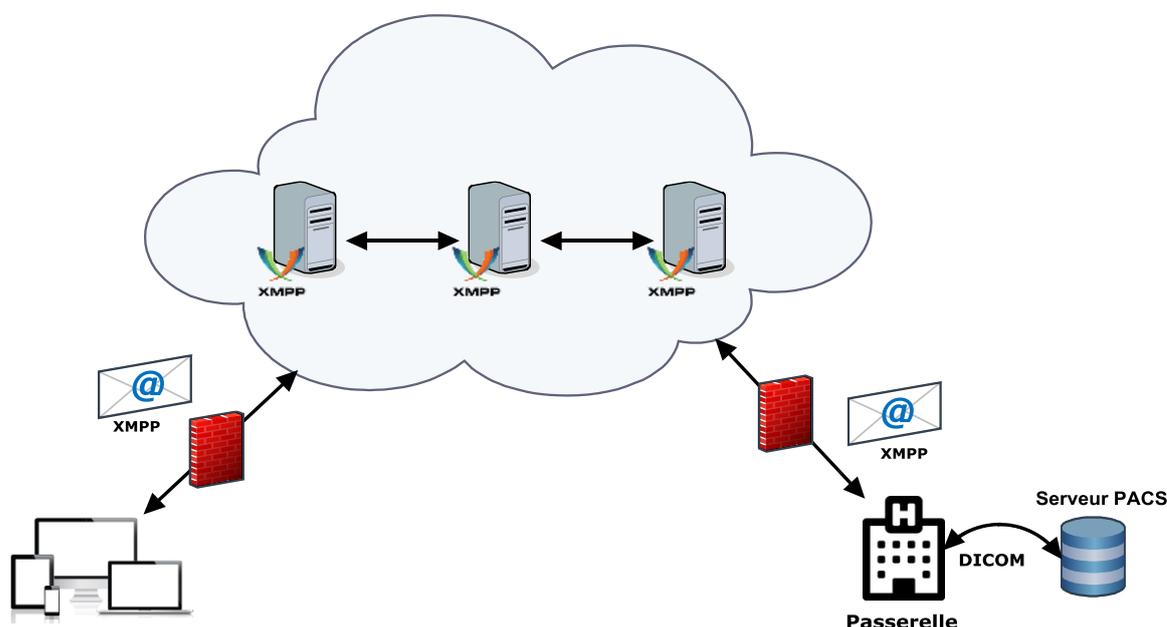


Figure 2.8: Contournement du pare-feu en utilisant le serveur XMPP comme relais Cloud. Le protocole XMPP est utilisé comme un middleware orienté message pour permettre une communication entre toutes les entités du système.

En raison de l'utilisation d'un middleware orienté message pour connecter les composants de chaque système, il devient possible d'envoyer un message donné à une passerelle qui est à l'intérieur d'un établissement de soins de santé. Ce middleware utilise le protocole XMPP, et il permet d'échanger des messages entre les composants. En profitant de la flexibilité de XMPP, il était possible de créer des fonctionnalités personnalisées, qui suivent le paradigme d'appel de procédure à distance, où un composant envoie un message par le relais Cloud pour un composant donné, et ces messages contiennent des informations sur l'action qu'il faut être exécuté dans la cible. Notez que le système peut gérer plus d'une passerelle de chaque client; par conséquent, il est conçu de telle sorte qu'il permet

au client une discrimination entre eux, en utilisant un identificateur de ressource XMPP [30].

De plus, dans la modélisation de système, ont été examinés les conditions d'accès et les questions de sécurité. Ainsi, il a été créé une sorte de filtre de paquets dans le système de relais Cloud. Ce filtre de paquet a pour but de contrôler l'accès à tous les composants de la passerelle PACS. Ainsi, la passerelle qui permet l'accès à un serveur de PACS est contrôlée par une liste d'accès, qui est maintenu sur une base de données SQL dans le Cloud. Ainsi, seuls les utilisateurs autorisés à contacter un serveur PACS donné peut exécuter des actions dans la passerelle cible. Le processus de travail de filtrage de paquets est défini dans la figure 2.9. En bref, il agit comme un pare-feu de couche 7 qui interprète tous les messages XMPP qui passent à travers le relais Cloud serveur XMPP et autorise ou bloque le message à accéder en conséquence des règles centrales. La conception du système permet la gestion des règles de pare-feu de chaque passerelle PACS seulement par le propriétaire de la ressource. Ainsi, le propriétaire d'une passerelle de PACS peut définir qui peut accéder au dépôt par la gestion de sa liste d'accès.

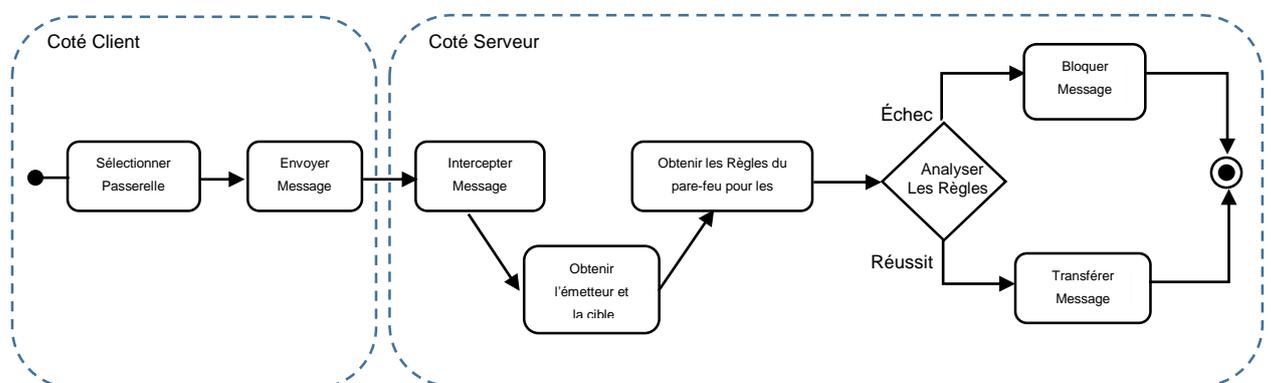


Figure 2.9: Diagramme du filtre de paquets de la passerelle relais Cloud.

Le schéma représente un intercepteur de paquets qui agit quand un message est reçu. Le filtre de paquets analyse la cible et la source de ce paquet et le message envoyé est transmis à la cible si les règles le permettent sinon il est bloqué.

Le schéma suivant, représenté dans la figure 2.10, montre comment le système fournit un moyen pour permettre la communication avec les composants de la passerelle PACS, en créant ainsi un environnement intégré, avec un accès de partout à l'information à l'intérieur des établissements de soins de santé. La passerelle joue un rôle important dans le système en permettant la traduction des messages de protocole XMPP au protocole DICOM. En conséquence, il devient possible d'accéder à toute entité dans un PACS à partir du réseau XMPP et par conséquent de l'ensemble d'Internet, parce que le système fournira un SaaS qui permet aux utilisateurs d'accéder au réseau XMPP via une application Web.

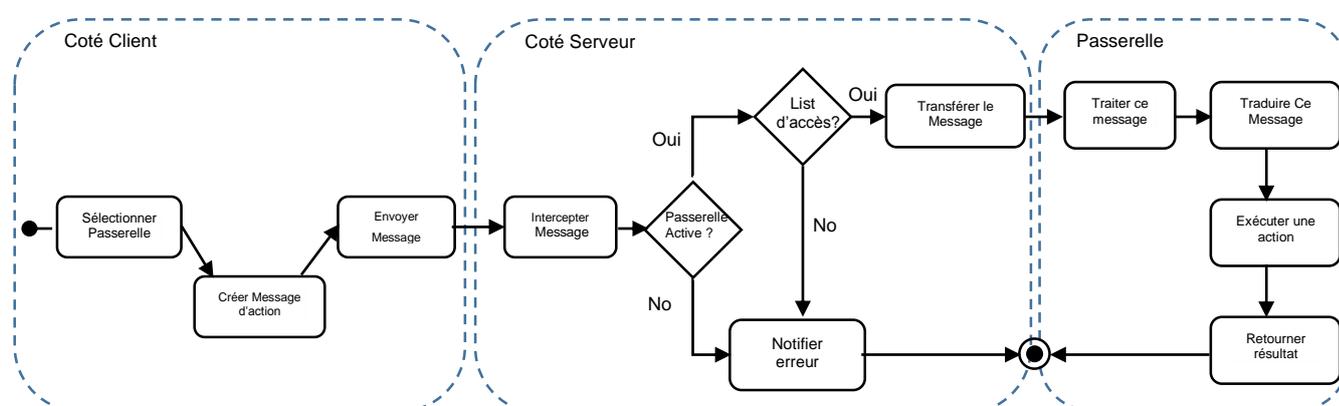


Figure 2.10: Diagramme d'exécution d'une action à distance

Ce diagramme modélise l'interaction entre les entités XMPP et le serveur PACS via une passerelle. Cette interaction est basée sur appel de procédure distante. Un message est envoyé à la passerelle cible qui a pour rôle d'agir comme une interface entre le réseau XMPP et l'infrastructure PACS cible.

#### 2.4.2 Modèle de données

La plateforme est conçue comme un système avec des ressources distribuées qui sont branchés au cœur de l'infrastructure et fournis en mode SaaS. Pour rendre facile la gestion du système, il a été décidé de découpler le cœur de l'infrastructure de l'entreprise de sources de données. Le système est composé de plusieurs sources de données (figure 2.11):

- Système informations base de données SQL

Base de données SQL pour stocker des informations sur le fichier, les partages et les règles du filtre des paquets.

- Stockage dans le Cloud

Il représente le stockage des images médicales et il est mis en œuvre en utilisant un service de stockage Cloud autonome.

- Serveurs PACS

Il peut y avoir certains serveurs PACS plug-in sur le système de la plateforme pour permettre aux utilisateurs d'accéder aux référentiels de l'institution et partager les ressources avec des amis.

Le système de la plateforme dispose d'un design complètement distribuée, dans lequel les ressources sont souvent propagées par l'Internet. Dans le système, le stockage dans le Cloud et SaaS peuvent cohabiter dans le même fournisseur ou elles peuvent être déployées sur différents fournisseurs de Cloud. Il peut y avoir plusieurs serveurs PACS disponibles à travers le système et comme indiqué dans les sections précédentes, l'accès à toutes les données détenues par un serveur PACS se fait par une passerelle, qui sert d'interface entre l'infrastructure de PACS et la plate-forme.

En outre, il y avait la nécessité d'utiliser une base de données SQL pour stocker des informations sur les activités de base comme, par exemple, des informations indexés sur les propriétaires des fichiers, partage des fichiers et les règles de pare-feu des relais Cloud pour chaque passerelle PACS.

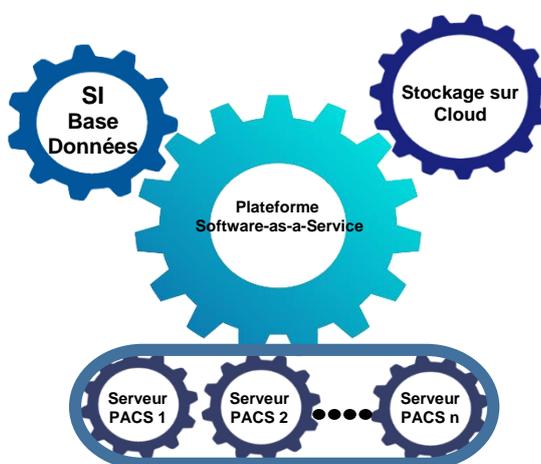


Figure 2.11: Diagramme modèle de données du système de la Plateforme

Comme on peut le voir dans le diagramme, le système de la plateforme crée un environnement parfaitement intégré, dans laquelle les utilisateurs peuvent accéder aux données stockés sur le Cloud et à partir des serveurs PACS distants.

La base de données d'activité est une base de données SQL. Cette base de données est gérée directement par le cœur de l'infrastructure de métier qui fournit les informations pour cette base de données. Le diagramme physique qui représente la manière dont les données sont organisées dans cette base de données est représenté dans la figure 2.12 et figure 2.13. La figure 2.12 représente l'organisation des données dans la base de données pour le partage de fichier et ça propriété et la Figure 2.13 montre comment le système stocke l'information concernant les règles de filtres des paquets pour un filtrage des messages dans le relais Cloud.

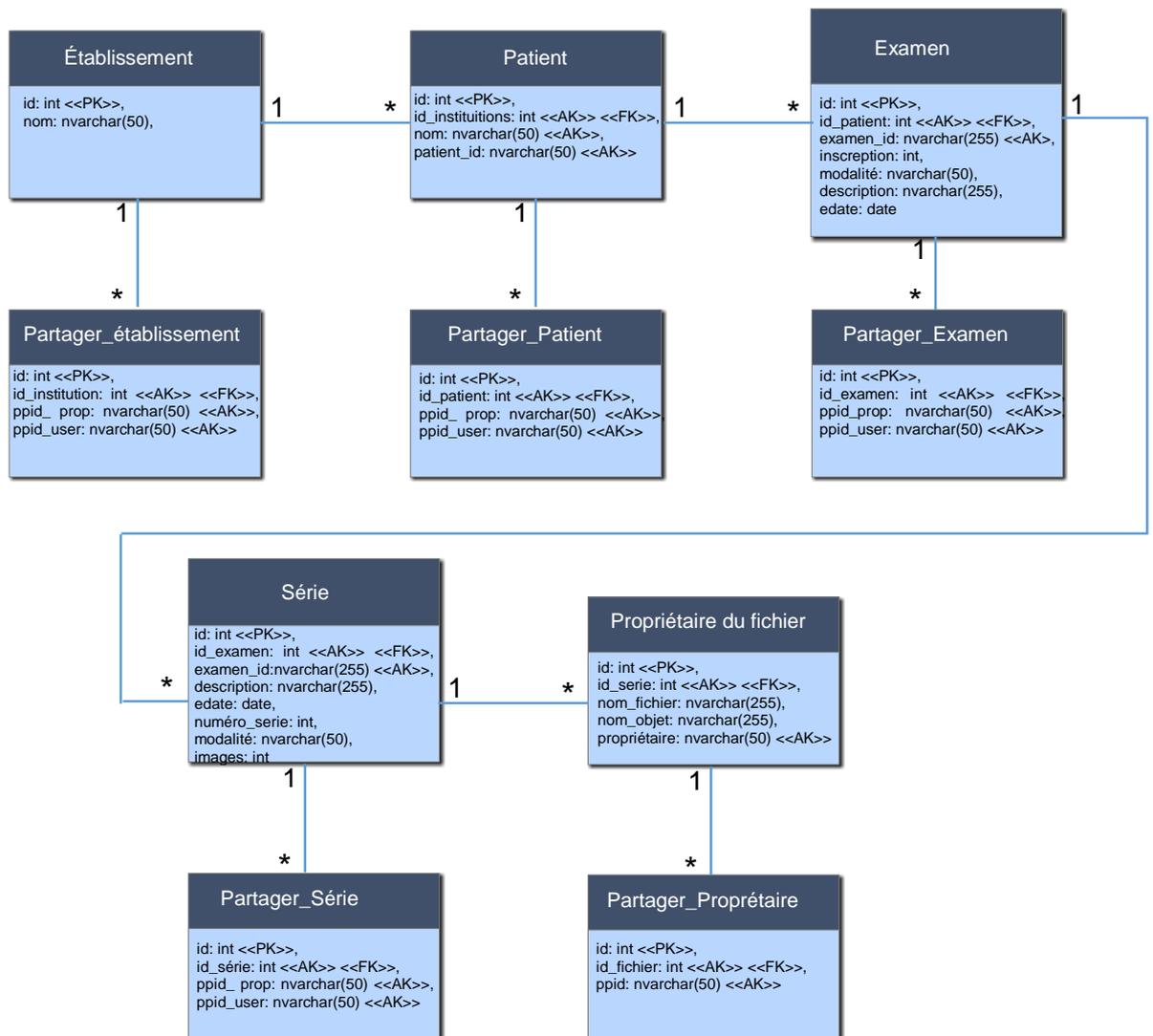


Figure 2.12: Modèle physique des données des fichiers partagés et la propriété des fichiers.

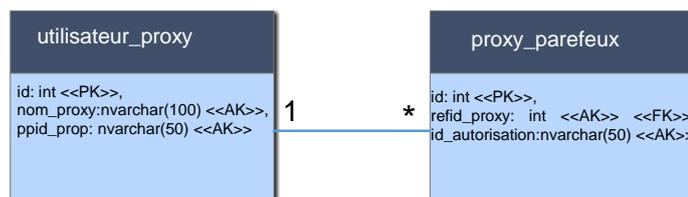


Figure 2.13: Modèle physique de données pour les règles de filtre des paquets.

### 2.4.3 Architecture

Comme résultat de l'étape de modélisation, une architecture (Figure 2.14) a été conçue pour répondre parfaitement aux exigences du système. Cette architecture prend en compte quatre grandes composantes, qui sont le serveur XMPP qui est le composant principal, l'application Web pour l'accès au SaaS, le stockage et la passerelle PACS.

Le serveur XMPP est le noyau du système. Par conséquent, il est responsable de tout le cœur de l'infrastructure. Le serveur fonctionne dans une infrastructure d'un fournisseur de Cloud. En outre, le serveur fournit un SaaS, qui fournit un middleware orienté message XMPP pour permettre l'accès à travers l'Internet à l'ensemble du système.

Une application Web a été créée Pour accéder aux SaaS fournis par l'infrastructure, développé selon le paradigme Web 2.0. Cette application agit comme un client XMPP dans le réseau XMPP. Ainsi, en utilisant un middleware orienté message, les utilisateurs peuvent interagir avec des collègues et gérer leurs ressources, qui peuvent être des fichiers stockés sur dossiers personnels ou des données à partir d'un serveur PACS qu'ils possèdent.

L'accès aux serveurs PACS est effectué en utilisant des passerelles. Ces passerelles sont installées à l'intérieur d'un établissement de santé, sur un poste de travail qui peut se connecter au serveur PACS cible. Ces passerelles sont considérées comme une composante au sein du réseau XMPP. Ainsi, il utilise le middleware pour interagir avec le serveur cœur de l'infrastructure et l'application Web.

Enfin, il y a le composant de stockage qui est utilisé par le système pour stocker les images médicales dans le Cloud. Cet élément est un composant abstrait. En effet, pour être en mesure de mettre en œuvre le système en utilisant un service de stockage Cloud, il était nécessaire de créer une abstraction dans la définition de l'élément de stockage. Cette abstraction conduit à la création d'une interface bien définie qui peut être implémenté pour utiliser n'importe quel fournisseur de stockage Cloud qui correspond au mieux avec les besoins du système. Par conséquent, en utilisant une interface bien définie et un système basé sur les plug-ins, il est possible d'ajouter n'importe quel nouveau composant qui implémente l'interface de stockage de notre plateforme. Selon cette approche, le système découple l'infrastructure de calcul de l'infrastructure de stockage.

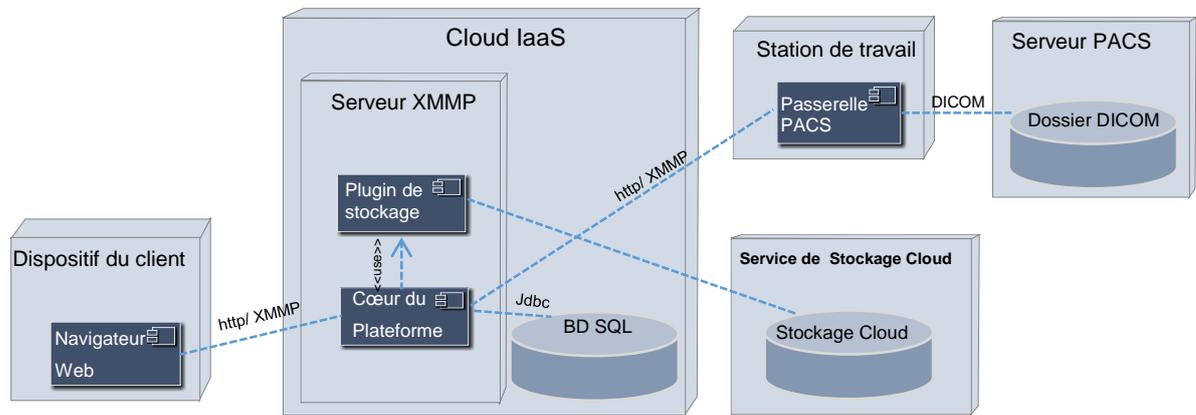


Figure 2.14: Schéma de l'architecture du Système

## CHAPITRE 3

# CONCEPTION ET IMPLEMENTATION

Ce chapitre explore les bases de la mise en œuvre du système, les problèmes rencontrés liés à la construction et les solutions techniques qui ont conduit à un système efficace. Les choix effectués pendant l'implémentation du système sont très importants pour arriver à un succès dans le résultat final. Certaines options ont été faites pour résoudre les problèmes techniques spécifiques comme, par exemple, permettre l'accès au système même derrière un pare-feu très restrictif qui ne permet pas la communication par le protocole HTTP via le port 80, etc.

### 3.1 DzTeleOph –SaaS-

Se concentrer sur l'implémentation de l'architecture souhaitée, on peut voir que la couche persistante de stockage est maintenue dans le Cloud, la mise en œuvre de cette composante est une étape critique pour la réalisation du système.

Dans la mise en œuvre du système de Télé ophtalmologie, il est très important d'éviter toute dépendance associée à tout système d'exploitation ou de tout fournisseur de Cloud. Par conséquent, son noyau doit être capable de fonctionner sur n'importe quelle plate-forme. Ainsi, il n'y a aucun doute que le langage de programmation Java est un bon choix pour développer le système. De plus, avec l'utilisation de Java pour le développement du système, il devient possible de déployer le noyau dans un IaaS de tous les fournisseurs de Cloud, même s'ils fournissent un IaaS avec système d'exploitation basé sur Unix ou Windows.

Dans la phase de modélisation, il a été défini qu'il faut utiliser un Middleware Orienté Message pour connecter tous les composants. De plus, nous avons choisi

le standard ouvert XMPP pour soutenir ce middleware orienté message. Ainsi, il est nécessaire d'utiliser un serveur XMPP dans le noyau de notre système. En outre, il n'est pas nécessaire de développer un serveur XMPP à partir du zéro, parce qu'il y a beaucoup de serveurs XMPP open source déjà implémentés [33]. Ainsi, il est nécessaire de choisir celui qui correspond bien aux exigences du système. Le choix s'est porté sur le serveur Openfire [34], qui est développé avec le langage de programmation Java et permet aux programmeurs d'étendre ses capacités avec de nouveaux plugins, même il est bien soutenu par la communauté.

### 3.1.1 Middleware Orienté Message : TeleOph Protocole de Communication Extensible –TOPCE

Le système utilise un middleware orienté message, afin qu'il soit possible de définir des messages personnalisés pour gérer des actions spécifiques. Profitant des avantages de l'extensibilité de Middleware Orienté Message de XMPP, il a été créé un groupe de messages qui définit un protocole personnalisé, nommé TeleOph Protocole de Communication Extensible (TOPCE). Ce groupe de messages personnalisés représente des extensions du protocole XMPP, qui étend les *stanzas* XML de **<iq \>** et **<message \>** avec des charges contenant des *espaces de nom* et *éléments* spécifiques.

Pour gérer ces extensions personnalisées, les entités du réseau XMPP doivent comprendre la signification de ces nouvelles extensions afin d'être en mesure d'agir en conséquence. Les entités doivent mettre en œuvre certains Handler **<iq \>** et **<message \>** pour soutenir ces extensions spécifiques et déclencher des actions selon, les charges utiles de ces extensions *stanza*.

Les fonctionnalités introduites avec les extensions de TOPCE peuvent être divisées en quatre groupes :

- Gérer les règles de filtrage de paquets
- Gérer l'anonymisation de proxy
- Gérer l'interaction à distance avec les passerelles du PACS
- Gérer à distance le dossier personnel

### 3.1.1.1 Gérer le dossier distant personnel

Cette fonctionnalité permet aux clients de contrôler leur dossier distant. Cette gestion se fait à l'aide d'une *stanza* XMPP `<iq \ >` personnalisée, dans laquelle il est prévu comme sa charge utile une action à exécuter et des paramètres. Pour identifier cette extension de XMPP, une `<iq \ >` personnalisé a été créé, qui commence avec le nom de la balise de l'élément **DDP** et utilise l'espace de noms personnalisé **custom:iq:ddp**. Ainsi, lorsque le serveur reçoit une `<iq \ >` qui appartient à cette extension, il déclenche le gestionnaire responsable à cette extension personnalisée de XMPP et le gestionnaire de IQ exécute l'action, en prenant en considération certains paramètres fournis en tant qu'élément enfant de la charge utile et renvoie une réponse en fonction de l'action et le résultat de sa exécution. Le flux d'exécution de l'action est représenté sur la figure 3.1.

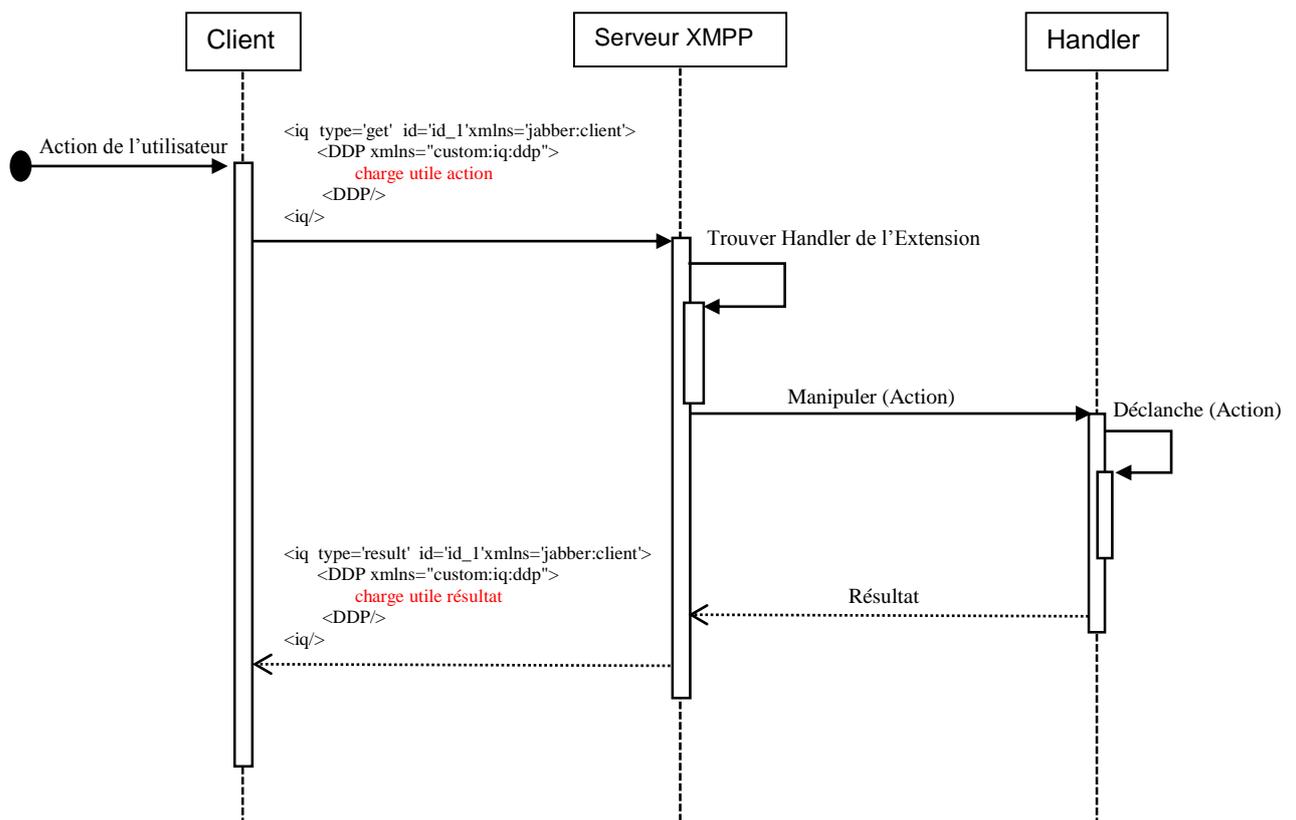


Figure 3.1: Diagramme de séquence gestion du répertoire à distance.

Les actions définies qui peuvent être effectuées sont:

- Recherche

Cette action permet de rechercher des études dans un dossier donné. Dans les paramètres de l'action peuvent être fournis le nom de l'établissement, le nom du patient, ID du patient, ID étude, ID série d'instance ou le nom du fichier.

De plus, il faut fournir dans toutes les actions de recherche un paramètre **level** qui identifie dans quel niveau doit être effectuée la recherche. Par exemple, si un utilisateur recherche chaque image médicale d'un établissement donné, il doit fournir le **level** = "IN" et le nom de l'institution, mais s'il veut rechercher toutes les images médicales d'une série d'un patient donné, il doit fournir le **level** = "SE", le nom de l'institution, le nom du patient, ID du patient, ID de l'étude et ID de la série (listing 3.1).

Listing 3.1:La charge utile de l'action Recherche

---

```
<recherche in='CHU.Benboulaid.Blida' np='User1' idp='A1520' et='TSE'  
se='512' level='SE' />
```

---

Comme résultat d'une action **recherche**, le serveur renvoie un **IQ**-résultat avec une charge utile (par exemple listing 3.2) contenant la liste des fichiers qui répondent à la recherche exécutée en fonction des paramètres fournis.

Listing 3.2: La charge utile du résultat de Recherche

---

```

<fichiers>
  < prop_ fichier ='utilisateur1 @teleoph.univ-blida.dz'
    nomf ='UCL300/PNEUMATIX/NOID/1.2.840.1137...' />
  < prop_ fichier ='utilisateur1@teleoph.univ-blida.dz'
    nomf ='UCL300/PNEUMATIX/NOID/1.2.840.1137...' />
  < prop_ fichier ='admin@teleoph.univ-blida.dz'
    nomf ='UCL300/PNEUMATIX/NOID/1.2.840.1137...' />
  . . .
  < prop_ fichier ='admin@teleoph.univ-blida.dz '
    nomf='UCL300/PNEUMATIX/NOID/1.2.840.1137...' />
</fichiers>

```

---

- Liste des fichiers partagés

Cette action (Figure 4.3) renvoie la liste de tous les fichiers actuellement partagés avec l'utilisateur qui exécute cette action. Le résultat est une liste de **<fichiers>** avec les attributs **nomf** et **prop** qui représentent respectivement, le nom de ce fichier et son propriétaire (par exemple, listing 3.2).

Listing 3.3 : la charge utile de l'action 'Liste des fichiers partagés'

---

```

<listfichiersPartager />

```

---

- Liste des Fichiers de l'utilisateur

Cette action (listing 3.4) renvoie la liste de tous les fichiers qui appartiennent à l'utilisateur qui exécute cette action (listing 3.5).

Listing 3.4 La charge utile de l'action 'Liste des fichiers de l'utilisateur'

---

```

<listeMesFichiers/>

```

---

## Listing 3.5 La charge utile du résultat

---

```

<fichiers>
  < prop_ fichier ='admin@teleoph.univ-blida.dz'
    nomf ='UCL300/PNEUMATIX/NOID/1.2.840.1137...'/>
  ...
  < prop_ fichier ='admin@teleoph.univ-blida.dz '
    nomf ='UCL300/PNEUMATIX/NOID/1.2.840.1137...'/>
</fichiers >

```

---

- Liste de partage d'un objet

Un utilisateur peut partager un objet avec quelques collègues. Cet objet peut être un dossier qui représente une institution, un patient, une étude ou une série à chaque objet il y a une liste associée qui identifie qui peut y accéder. Cette liste est la liste de partage. Quand un utilisateur exécute cette action une liste est retournée avec tous les utilisateurs qui appartiennent à la liste de partage d'un objet (Listing 3.7).

L'identification de l'objet doit être fournie pour exécuter l'action. Cette identification est faite par le niveau de l'objet, qui peut être *IN*, *NP*, *ET* ou *SE*, respectivement, une institution, un patient, une étude ou un dossier des séries. De plus le **level**, il faut fournir aussi le nom de l'institution, le nom et le ID du patient, ID de l'étude et ID de la série, selon le **level** sélectionné.

Listing 3.6 : charge utile de l'action *partagerAvec*


---

```

<partagerAvec level='SE' in='CHU.Benboulaid.blida' np='Amine' idp= '7DfDKDK'
  et=' 1.2.840.113745.101000.1008000.38446.6272.7138759 '
  se=' 1.3.12.2.1107.5.2.13.20561.30000005042216091690600002608 ' />

```

---

Listing 3.7: charge utile du résultat *partager Avec*


---

```
<utilisateurs>
  < utilisateur nom=' amine @teleoph.univ-blida.dz ' />
  < utilisateur nom=' admin @teleoph.univ-blida.dz ' />
</utilisateurs>
```

---

- Supprimer un fichier

Cette action (Listing 3.8) permet à l'utilisateur de supprimer un objet de son répertoire. L'identification d'objet est réalisée de la même manière que décrit ci-dessus. Si un utilisateur veut supprimer un objet donné de son répertoire, il doit identifier cet objet dans les paramètres de l'action de suppression.

Il n'y a aucun résultat spécifique. Si le serveur répond avec un **IQ résultat** l'action a été exécutée avec succès. Si le serveur répond avec un **IQ erreur** la suppression n'a pas eu lieu.

Listing 3.8 : charge utile de l'action *supprimer*


---

```
<supprimer level='SE' in='CHU.Benboulaid.blida' np='Amine' idp='7DfDKDK'
et=' 1.2.840.113745.101000.1008000.38446.6272.7138759 '
se=' 1.3.12.2.1107.5.2.13.20561.30000005042216091690600002608 ' />
```

---

- Configurer les autorisations de stockage

Le stockage de données est réalisé hors bande. En d'autres termes, le système n'utilise pas XMPP pour stocker des images médicales dans le Cloud parce que, il n'est pas conçu pour gérer les données binaires, Ainsi, pour contrôler le stockage des données hors bande dans un dossier d'un utilisateur de Cloud, il est nécessaire de demander l'autorisation de stockage avant de stocker les données dans le dossier. L'action est refusée en cas de tentative de stockage des données sans permission.

Un utilisateur peut seulement définir les autorisations de stockage de son dossier, mais il peut obtenir la permission de stockage dans les dossiers des autres utilisateurs.

La permission de l'action stockage (Listing 3.9) doit fournir un paramètre de niveau **level** qui peut être défini comme **étude** ou **série**, pour identifier les études ou séries qui peuvent être stockées dans le dossier. De plus, il y a le paramètre **PermisUtilisateur** qui identifie l'utilisateur qui peut stocker les études ou séries.

Cette action ne nécessite pas de résultats.

#### Listing 3.9 : charge utile de l'action

---

```
<SavgPermission PermisUtilisateur = 'amine@teleoph.univ-blida.dz' level=' étude '
ide=' 1.2.840.113745.101000.1008000.38446.6272.7138759 ,
1.2.840.113745.101000.1008000.38446.6272.7138760 ,
... ,
1.2.840.113745.101000.1008000.38446.6272.7138769 ' />
```

---

- Partager objet

L'action partager objet (Listing 3.10) permet à l'utilisateur de partager un objet avec d'autres utilisateurs. Pour effectuer cette action l'utilisateur doit identifier l'objet à partager comme décrit ci-dessus. En outre, il doit fournir une liste d'utilisateurs, en utilisant le paramètre **partagerAvec** afin de les ajouter à la liste de partage de l'objet.

Listing 3.10 : charge utile de l'action Partager objet

---

```

<Partage>
  <item partagerAvec='amine@teleoph.univ-blida.dz' level='SE'
    in='CHU.Benboulaid.blida' np='Amine' idp='7DfDKDK'
    et='1.2.840.113745.101000.1008000.38446.6272.7138759'
    se='1.3.12.2.1107.5.2.13.20561.30000005042216091690600002608' />
  ...
  <item partagerAvec='admin@teleoph.univ-blida.dz' level='SE'
    in='CHU.Benboulaid.blida' np='Amine' idp='7DfDKDK'
    et='1.2.840.113745.101000.1008000.38446.6272.7138759'
    se='1.3.12.2.1107.5.2.13.20561.30000005042216091690600002618' />
< Partage />

```

---

- Supprimer un partage d'objet

Comme l'action partager objet, cette action a besoin des paramètres pour identifier l'objet cible et aussi la liste des utilisateurs à supprimer de la liste de partage des objets. La charge utile de cette action est presque équivalent à la charge associée à l'action partager objet (Listing 3.10), mais au lieu de commencer avec le nom de l'élément **< Partage >** ça commence avec le nom de l'élément **< NonPartage >**.

### 3.1.1.2 Gérer les règles de filtrage des paquets

Comme l'extension définie dans la section précédente, la gestion des règles de filtrage de paquets repose également sur **<iq \>** personnalisé, identifié par l'élément **PARFEU** et l'espace de nom **custom: iq: parfeu**. Cette **<iq \>** personnalisée est utilisée pour déclencher des actions dans le but de permettre à un utilisateur de gérer l'accès à son serveur PACS via une passerelle.

Les actions possibles déclenchées avec cette extension personnalisée sont les suivants :

- Obtenez l'ACL d'une passerelle

Cette action peut être invoquée par l'envoi d'un `<iq \>` avec une charge utile pour identifier la passerelle cible (Listing 3.11). Une copie de sa liste d'accès sera retournée, où sont présentés les utilisateurs qui peuvent accéder à la passerelle.

Listing 3.11 : charge utile de l'action Obtenez ACL

---

```
< ObtenezACL proxy='PACS1' />
```

---

- Refuser l'accès

Cette action permet à un utilisateur de refuser l'accès à une passerelle pour un groupe d'utilisateurs. Comme représenté dans le Listing 3.12, une liste des règles doit être fournie ; dans cette liste la passerelle du PACS est identifiée par l'attribut **proxy** et l'utilisateur bloqué par l'attribut **Blockuser**.

Listing 3.12 : charge utile de l'action Refuser l'accès

---

```
< RefuserAccess/>
  <regle proxy='PACS1' Blockuser=' amine @teleoph.univ-blida.dz' />
  <regle proxy='PACS2 ' Blockuser=' amine @teleoph.univ-blida.dz' />
  <regle proxy='PACS 1' Blockuser=' admin @teleoph.univ-blida.dz' />
<RefuserAccess/>
```

---

- Autoriser l'accès

Pour permettre à une liste d'utilisateurs d'accéder à une passerelle donnée, un message doit être envoyé avec les passerelles et les utilisateurs ayant la permission d'accès à ces passerelles.

Listing 3.13: charge utile de l'action autoriser l'accès.

---

```
<AutoriserAcces />  
    <regle proxy='PACS1 ' Autoriseruser=' amine @teleoph.univ-blida.dz' />  
    <regle proxy='PACS2 ' Autoriseruser=' amine @teleoph.univ-blida.dz' />  
    <regle proxy='PACS1 ' Autoriseruser=' admin @teleoph.univ-blida.dz' />  
< /AutoriserAcces />
```

---

### 3.1.1.3 Gérer l'anonymisation de proxy

Certaines préoccupations liées à la confidentialité des données médicales créent d'habitude quelques restrictions de gestion. Par conséquent, le système devrait créer certaines fonctionnalités pour éviter des situations indésirables et permettre aux utilisateurs d'utiliser le service fourni par ce système sans se soucier d'avoir violé une règle de confidentialité. Une façon trouvée pour s'attaquer à ce problème est d'introduire un outil d'anonymat qui permet aux utilisateurs de définir qu'une passerelle doit rendre anonyme les fichiers qu'il envoie au Cloud. De plus, dans le système de la plateforme, il y a des contraintes implicites qui sont imposées lorsqu'un utilisateur partage des études ; ces contraintes définissent une limitation d'accès pour ces images médicales. Cette limitation consiste à ne pas permettre aux utilisateurs de partager avec d'autres collègues des fichiers qui ont été partagés avec eux. D'ailleurs, le système ne fournit pas les outils de téléchargement direct des images partagées.

L'extension de stockage XML privé de XMPP [31] est utilisée pour indiquer les passerelles qu'ils doivent rendre anonymes, tous les fichiers qu'ils envoient à un utilisateur donné, cette extension est une extension standard de XMPP identifiée par **<query>** et l'espace de noms **'jabber:iq:private'**.

- Définir les règles d'anonymisation

Cette action permet aux utilisateurs d'indiquer à une passerelle donnée, si elle doit ou non anonymiser les fichiers exploités par un utilisateur tiers. Il utilise l'extension stockage XML pour envoyer un IQ-set avec un fichier XML contenant cette information.

Pour stocker des informations sur l'anonymisation pour un proxy donné, on utilise une charge utile permettant d'identifier la passerelle, dans lequel le nom d'élément est le nom de la passerelle et l'élément enfant est délimité par l'espace de nom ' **custom:proxy:anon**' (Listing 3.14).

Listing 3.14 : messages échangés

---

*CLIENT:*

```
<iq type=' set ' id=' 7594:saveXML ' xmlns=' jabber:client '>
  <query xmlns=' jabber:iq:private '>
    <PACS1 xmlns=' custom:proxy:anon '>
      < utilisateurs >
        < utilisateur nom=' user0@teleoph.univ-blida.dz' anon=' true ' />
        < utilisateur nom=' user1@teleoph.univ-blida.dz' anon=' false ' />
        < utilisateur nom=' user2@teleoph.univ-blida.dz ' anon=' false ' />
        < utilisateur nom=' user3@teleoph.univ-blida.dz' anon=' true ' />
      </ utilisateurs >
    </PACS1>
  </query>
</ iq>
```

*SERVER:*

```
<iq type="result "
to=' user4@teleoph.univ-blida.dz /webclient '
id="7594:saveXML"
xmlns='jabber:client ' />
```

---

- Liste des Règles d'anonymisation

Cette action permet à une passerelle d'obtenir des informations sur les règles d'anonymisation définies. Pour obtenir ces règles d'anonymisation spécifiques, la passerelle doit envoyer un **IQ-get** avec une charge utile où le nom de l'élément, le nom de la passerelle et l'élément enfant sont délimités par l'espace de nom ' **custom:proxy:anon**' (Listing 3.15).

## Listing 3.15 : messages échangés

---

```

CLIENT:
<iq type='get' id='7597:loadXML' xmlns='jabber:client'>
  <query xmlns='jabber:iq:private' >
    <PACS1 xmlns='custom:proxy:anon' />
  </query>
</iq>
SERVER:
<iq xmlns='jabber:client' to='user4@teleoph.univ-blida.dz/webclient'
  type='result' id='7597:loadXML' >
  <query xmlns='jabber:iq:private' >
    <PACS1 xmlns='custom:proxy:anon' >
      <utilisateurs >
        <utilisateur anon='true' nom='user0@teleoph.univ-blida.dz' />
        <utilisateur anon='false' nom='user1@teleoph.univ-blida.dz' />
        <utilisateur anon='false' nom='user2@teleoph.univ-blida.dz' />
        <utilisateur anon='true' nom='user3@teleoph.univ-blida.dz' />
      </utilisateurs >
    </PACS1>
  </query>
</iq>

```

---

#### 3.1.1.4 Gérer l'interaction à distance avec les passerelles PACS

Contrairement aux extensions référencées ci-dessus qui représentent des nouvelles fonctionnalités dans la communication client-serveur, cette extension ajoute des nouvelles fonctionnalités dans la communication client-client. Cette extension est chargée d'introduire de nouveaux messages dans la couche protocole, ces messages peuvent être utilisés par les utilisateurs pour gérer et à interagir avec leurs passerelles PACS distants.

Cette extension est basée sur la *stanza* XML `<message\>`. L'extension est identifiée par un élément enfant `<XMPPTeleOph\>` et l'espace de nom `http://teleoph.univ-blida.dz`.

De plus, toutes les charges utiles des extensions contiennent les mêmes paramètres (tableau 3.1) :

Paramètre	Description
<b>Id</b>	Il identifie le message. Il doit être unique dans une session.
<b>Type</b>	Il identifie le type de message. Il peut prendre les valeurs suivantes :  <b>-QUERY</b> pour identifier un message requête.  <b>-QR</b> pour indiquer que ce message doit être transformé en command DICOM <b>Query/Retrieve</b> .  <b>-SYNCH</b> il est utilisé pour demander à une passerelle de synchroniser un dossier donné à partir du Cloud avec une station de travail dans la passerelle où elle est installée (seulement le propriétaire de la passerelle peut exécuter cette action).
<b>Status</b>	Il est utilisé pour identifier la réussite ou l'échec d'une demande donnée. Ce paramètre prend la valeur <b>OK</b> pour identifier une réponse qui a été exécutée avec succès et prend la valeur <b>ERREUR</b> pour identifier une défaillance dans l'exécution d'une action demandée.
<b>Data</b>	Ce paramètre peut contenir n'importe quel type de texte. Il est utilisé pour le transport des paramètres supplémentaires pour une demande ou des résultats dans un message de réponse.

Tableau 3.1 Le format de message pour la gestion des passerelles PACS distants

Cette extension ajoute des nouvelles fonctionnalités au Middleware Orienté Message qui permettent aux utilisateurs d'exécuter les actions suivantes dans leur passerelle PACS :

- Lancer des requêtes

Cette action est utilisée pour envoyer une requête à une passerelle PACS donnée, quand le paramètre **Data** envoie un texte supplémentaire formater qui contient les paramètres à utiliser pour créer la commande DICOM **C-FIND**.

- Récupérer des images médicales à partir du serveur PACS

Cette action permet à l'utilisateur de récupérer des images médicales à partir d'un serveur d'archivage PACS, via la passerelle du PACS et de les stocker dans le dossier personnel dans le Cloud. Il utilise également le paramètre **Data** pour envoyer les paramètres utilisés dans la construction des commandes DICOM **C-FIND** et **C-MOVE**.

- Synchroniser les données

Cette action est utilisée pour récupérer des images médicales d'un dossier donné dans le Cloud et le synchroniser avec un dossier dans la station de travail, où la passerelle est installée. Les informations qui identifient le dossier à synchroniser avec cette station de travail sont envoyées dans le paramètre **Data**.

### 3.1.2 Serveur XMPP - Openfire

Pour gérer le middleware orienté message, notre système utilise le serveur XMPP Openfire pour sa facilité d'installation et sa simplicité dans l'administration, toutefois il propose une performance et une sécurité solides. Par ailleurs, une des caractéristiques les plus importantes du serveur Openfire et qui mène à l'utiliser comme le cœur de notre système est sa bonne conception, qui permet aux développeurs d'étendre facilement ses fonctionnalités à l'aide des plugins.

#### 3.1.2.1 Système de plugin

Le serveur Openfire fournit une interface bien définie qui permet aux développeurs d'implémenter d'une manière simple de nouveaux plugins. Ainsi, pour mettre en œuvre toute la couche cœur de notre système, il est nécessaire de développer un plugin qui s'exécute dans le serveur Openfire.

Pour créer un plugin sur Openfire, les développeurs ont besoin d'implémenter une classe d'initialiseur de plugin qui implémente *org.jivesoftware.openfire.container.Plugin*. Cette classe est chargée de mettre en œuvre les méthodes *initializePlugin* et *destroyPlugin*, qui font respectivement, le code d'initialisation du plugin et la suppression du plugin en *destroy* (Listing 3.16).

Listing 3.16 : Class qui initialise un plugin

---

```

package TeleOph.openfireplugin ;
import org.jivesoftware.openfire.container.Plugin ;
public class TeleOphOpenfirePlugin implements Plugin {

    public void initializePlugin (PluginManager manager, File pluginDirectory ) {
                                                // code d'initialisation
                                                }

public void destroyPlugin ( ) {
                                                // code destruction
                                                }

```

---

De plus, afin de charger le plugin, il est nécessaire d'identifier la classe, entre autres dans le fichier `.jar` qui est responsable de son chargement. C'est pourquoi, les développeurs doivent placer dans les racines de `.jar` un fichier nommé `plugin.xml`. Le chargeur des plugins d'Openfire lit ce fichier et identifie la classe nécessaire pour charger le plugin et exécuter la méthode `initializePlugin` (Listing 3.17) de cette classe.

Listing 3.17 : Main class plugin

---

```

<plugin>
  <!-- Main plugin class -->
  <class>TeleOph.openfireplugin.TeleOphOpenfirePlugin
  </class>

  <!-- Plugin meta-data -->
  <name>TeleOphOpenfirePlugin</name>
  <description>Interact with Cloud providers</description>
  <author>F.a</author> <version>1.0</version>
  <minServerVersion>3.3.0</minServerVersion>
  <licenseType>gpl</licenseType>
</plugin>

```

---

### 3.1.3 Dossier Distant Personnel - DDP

Aujourd'hui, les Dossiers Cloud sont très utilisés dans les activités quotidiennes des utilisateurs. Actuellement, les utilisateurs utilisent des services comme Sky Drive, Dropbox, Google Drive, etc., pour stocker leurs fichiers dans le Cloud et aussi pour pouvoir accéder à ces fichiers depuis n'importe où.

En utilisant le même concept introduit par certains fournisseurs de stockage de fichiers dans le Cloud, notre système fournit aussi un mécanisme qui permet de stocker des images médicales dans le Cloud avec des fonctionnalités de partage. D'ailleurs, les utilisateurs ont la notion de dossier personnel, qu'ils peuvent utiliser pour stocker des fichiers afin qu'ils soient accessibles n'importe où en utilisant des postes de travail ou des appareils mobiles. Le dossier est géré par le plugin de stockage qui est connecté avec le plugin du noyau Openfire afin de fournir un flux d'entrée / sortie pour écrire ou lire des fichiers à partir de n'importe quel fournisseur de Cloud.

Dans le système de la plateforme télé-ophtalmologie, chaque utilisateur possède un répertoire, qui peut contenir plusieurs dossiers c.-à-d. études. L'utilisateur peut partager ses dossiers avec d'autres utilisateurs. Ces derniers deviennent capables de visualiser ou gérer ces dossiers, bien sûr, selon la permission partagée. Chaque dossier peut contenir plusieurs images médicales, stockées dans un seul service de stockage Cloud.

La relation entre les utilisateurs et les dossiers est représentée dans la figure 3.2

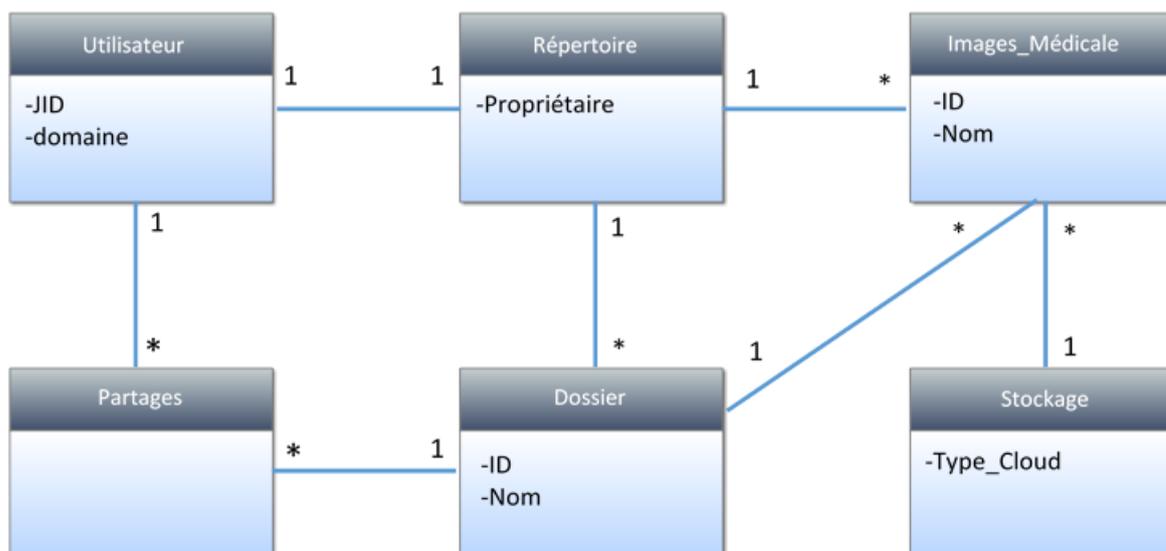


Figure 3.2 : Diagramme du Dossier Distant Personnel montre la relation entre les utilisateurs et les dossiers dans le Cloud

#### 3.1.4 Organisation du cœur du Système

Le cœur du système est un élément important qui est déployé comme un plugin du serveur Openfire. Ce composant apporte toute la sémantique pour le système. Il permet au serveur d'interpréter les fonctionnalités personnalisées créées sur la base du protocole XMPP, nommé TOPCE ([section 3.1.1](#)) et dispose également la tâche de chargement du plugin responsable de la gestion de stockage.

Toute la logique du système se pose sur le middleware orienté message du XMPP. Ainsi le cœur du système implémente quelques gestionnaires pour fonctionner avec le middleware, afin de soutenir le protocole de communication et permettre au système de fonctionner correctement.

Le cœur du système a trois composantes principales :

- Stockage

C'est un composant de stockage qui est responsable de la gestion d'accès à un service de stockage Cloud du système. Ce composant de stockage est utilisé pour stocker, dans un fournisseur de Cloud, toutes les images médicales détenues par ce système.

- Enregistreur de gestionnaire

L'enregistreur de gestionnaire est le composant qui enregistre certains gestionnaires pour faire face à chacune de nouvelles extensions de TOPCE, construit sur la base du protocole XMPP.

- Servlet chargement/téléchargement

Ce composant est utilisé pour permettre le chargement et le téléchargement hors bande des images médicales.

#### 3.1.4.1 Stockage

Le composant de stockage est aussi un plugin de serveur Openfire. Cependant, ce plugin dispose d'un parent spécifique, qui est le plugin du cœur de système.

Ce plugin est le composant qui permet au noyau du système d'être découplé de tout fournisseur de stockage Cloud. Ce découplage est possible parce que le noyau du système définit une interface (Listing 3.18) qui devrait être implémenté par un plugin de stockage afin d'être fixé sur le cœur du système, permettant ainsi au cœur du système d'accéder à un stockage Cloud spécifique.

Listing 3.18: interface de Stockage

---

```
public interface ArchiveStream {
    public void writeFile ( String owner , DicomTreeInfo dicomTree ,String id ,
                        byte [ ] in , boolean append ) throws IOException ;
    public void writeFile ( String owner , DicomTreeInfo dicomTree , String id ,
                        InputStream in ) throws IOException ;
    public InputStream readFile ( String owner , String id ) throws IOException ;
    public boolean removeFile ( String owner , String id ) throws IOException ;>
    public boolean removeFile ( String owner , DicomInfo info ) throws IOException ;
    public boolean exists ( String owner , String id ) throws IOException ;
    public List<String> listDirectory ( String owner , DicomFileFilter filter )throws IOException ;
}
```

---

Afin d'être disponible, le plugin de stockage doit être ajouté à un gestionnaire d'archive (figure 3.3) dans le cœur du système. Ce gestionnaire identifié par son nom de classe permet au système d'accéder à un plugin spécifique de stockage. Avec un peu de flexibilité pour le choix d'un plugin, il est possible de connecter le

cœur du système avec plusieurs plugins de stockage spécifique. Il y a un plugin par défaut qui est le premier plugin connecté avec le cœur du système.

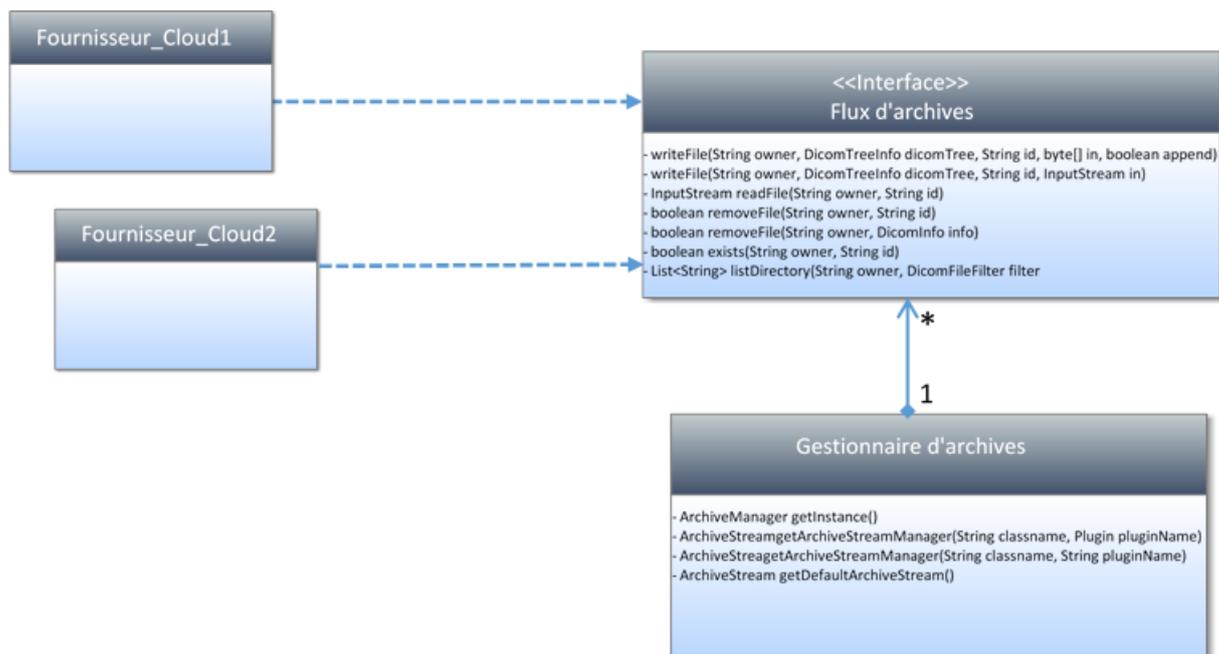


Figure 3.3: Diagramme Gestionnaire d'archive

Pour stocker les images médicales numériques dans un fournisseur de Cloud, un plugin doit être créé pour utiliser le stockage Cloud de ce fournisseur (figure 3.4). Pour des raisons de sécurité et de confidentialité pour certains fournisseurs Cloud, comme par exemple Amazon S3, le plugin de l'archivage à distance doit crypter les fichiers avant de les envoyer à l'infrastructure du fournisseur de Cloud et les décrypter lorsqu'un utilisateur veut l'exploiter.

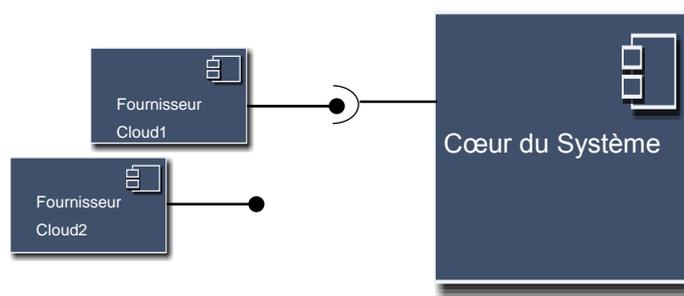


Figure 3.4: Implémentation des plugins d'archive à distante

### 3.1.4.2 Enregistreur de gestionnaire

Le cœur du système met en œuvre un enregistreur de gestionnaire qui instancie certains gestionnaires pour traiter les extensions TOPCE de XMPP. Le groupe d'extensions TOPCE définit deux extensions client-serveur, qui sont l'extension de gestion d'archive distant personnel et l'extension de gestion des règles du filtre de paquets. Ainsi, pour faire fonctionner chacune de ces extensions, il a été implémenté certains gestionnaires spécifiques. Chaque gestionnaire est en mesure de comprendre l'extension respective et agit en fonction du message reçu.

De plus, ces deux extensions client-à-serveur qui sont manipulées par le Plugin du noyau Openfire, sont basées sur les *stanzas* IQ du XMPP. Ainsi, pour mettre en œuvre les gestionnaires à faire face à certaines IQ spécifiques, nous avons créé quelques classes qui étendent la classe **IQHandler** du SDK du serveur Openfire et mettent en œuvre la méthode *handleIQ* (paquet IQ). Dans l'implémentation de la méthode *handleIQ* (paquet IQ) le gestionnaire analyse le paquet reçu et déclenche une méthode pour exécuter une action, en fonction du contenu du paquet reçu (figure 3.5).

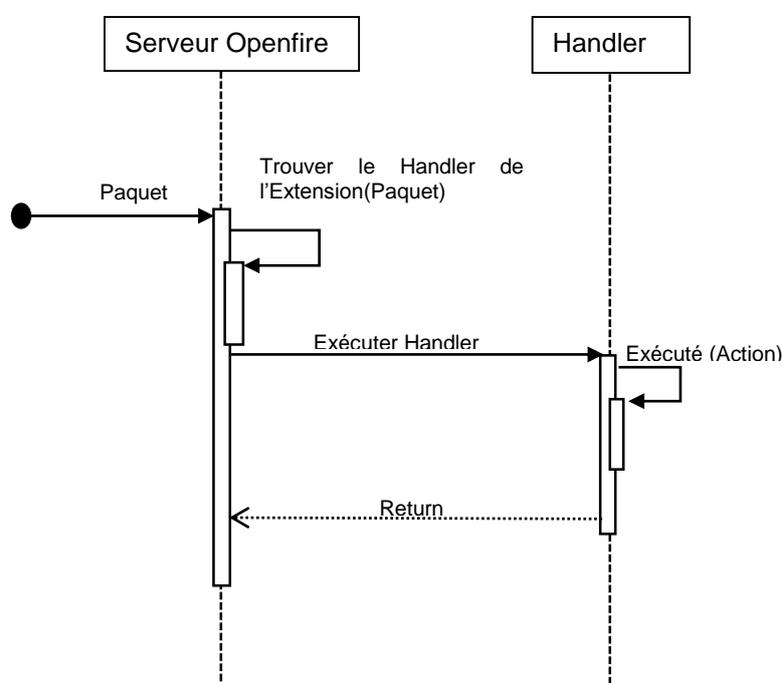


Figure 3.5: Diagramme de séquence de l'extension gestionnaire (handler) des paquets

### 3.1.4.3 Servlet chargement/téléchargement

Le middleware utilisé dans notre système révèle une certaine limitation du traitement de données binaires, car pour gérer le transfert de données binaires dans la bande le Middleware Orienté Message du XMPP est relativement inefficace. Cette limitation est due au fait que la connexion XMPP est codée comme un flux XML de longue durée de vie, donc pour transférer des données binaires, il est nécessaire de les coder en utilisant Base64 avant qu'il puisse être transmis dans la bande. L'utilisation de Base64 [35] n'est pas efficace pour transférer de grandes quantités de données, car le nombre d'octets de sortie par rapport au nombre des octets d'entrée est d'environ 33% en plus.

Pour contourner cette limitation du XMPP dans le transfert des données binaires, il existe quelques méthodes qui consistent à envoyer les données directement d'une entité à une autre via une connexion sans intermédiaires et faire la signalisation dans la bande avec XMPP. Par exemple, en utilisant SOCKS5 spécifié dans SOCKS5 Bytestreams [XEP-0065] [36] ou en utilisant Jingle RTP sessions [XEP-0167] [37] pour établir une connexion par voix et vidéo. Cependant, dans le système proposé les données des images médicales numériques sont envoyées au serveur via HTTP POST, aussi il y a une signalisation dans la bande en utilisant le middleware du XMPP pour informer le serveur qu'un client est en train d'envoyer une image donnée pour le stocker dans le Cloud.

Le système fournit une servlet pour permettre le chargement des images médicales numériques vers le Cloud. Évidemment, pour uploader un fichier vers le dossier personnel d'un utilisateur donné dans le Cloud il y a quelques problèmes de sécurité qui doivent être prises en compte. Dans notre système, chaque utilisateur a son dossier personnel dans le Cloud, donc lui seul peut uploader des fichiers à son dossier ou donner une autorisation à un ami sur la même passerelle PACS du système pour uploader des fichiers à son dossier.

Les listes des études ou séries autorisées pour un ami sont fournies par un mécanisme de signalisation dans la bande. Cette fonctionnalité utilise des messages TOPCE ([section 3.1.1](#)) afin de fournir les informations souhaitées au noyau du système.

Quand un utilisateur autorisé du système (accès partagé) fait une action Query/Retrieve sur des images médicales de la passerelle PACS du système, il est nécessaire d'exécuter quelques étapes avant que les fichiers peuvent être stockés dans le dossier personnel de l'utilisateur. Ces étapes sont représentées dans le schéma suivant (figure 3.6) :

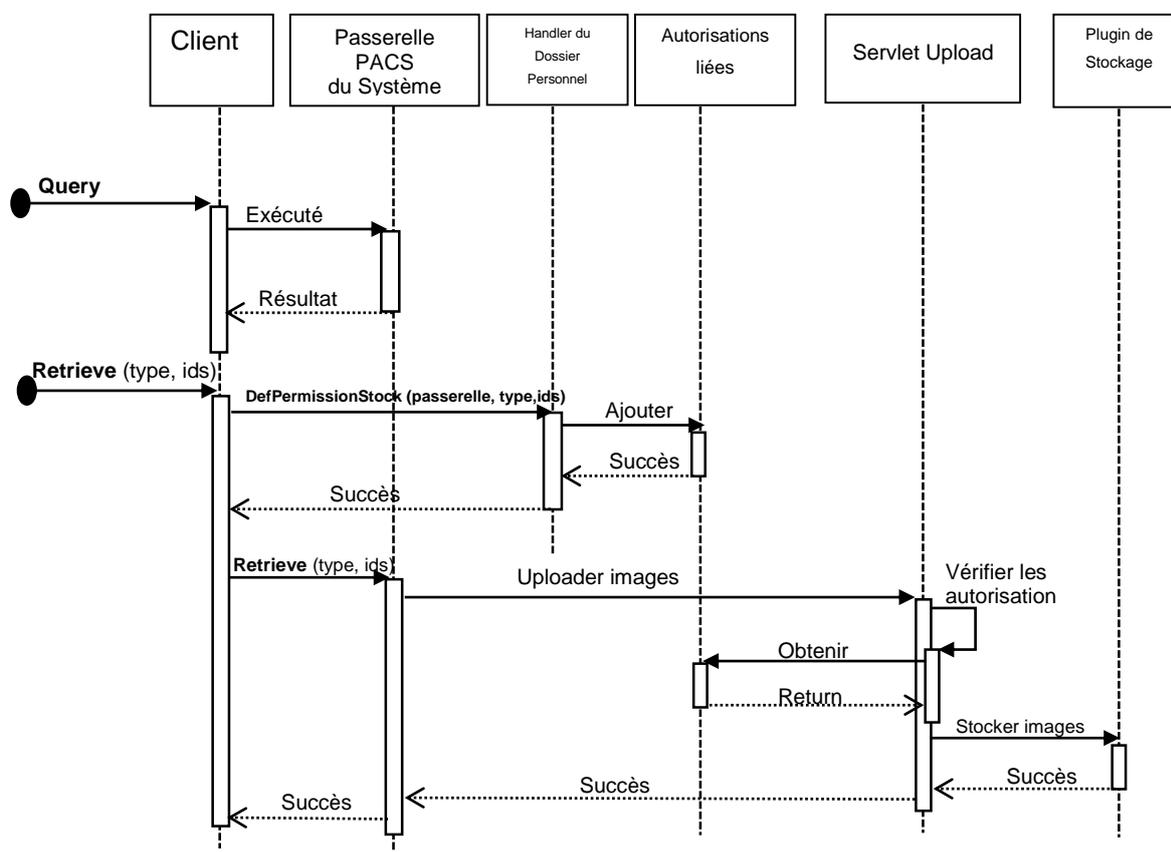


Figure 3.6: Diagramme de séquence chargement des fichiers vers le stockage Cloud

Tel que mentionné avant, le chargement se fait en utilisant un HTTP POST hors bande. Ainsi, dans la méthode POST, le client doit fournir des informations sur le dossier personnel de l'utilisateur où doit être conservé le fichier. Pour vérifier l'authenticité de ce POST, le JID de l'utilisateur qui envoie le fichier est utilisé avec un hachage (Figure 3.7).

Pour authentifier les fichiers que les utilisateurs envoient vers le Cloud, le système utilise la fonction SHA1 HMAC [38]. Le noyau du système utilise le HMAC pour vérifier si le fichier a été chargé par un utilisateur avec des droits d'écriture dans le dossier voulu. Avant de faire le chargement, le client doit calculer un SHA1

HMAC des données en utilisant le mot de passe de l'utilisateur. De ce fait, le système est en mesure de vérifier l'authenticité HTTP POST parce que le mot de passe est partagé entre le client et le serveur [39].

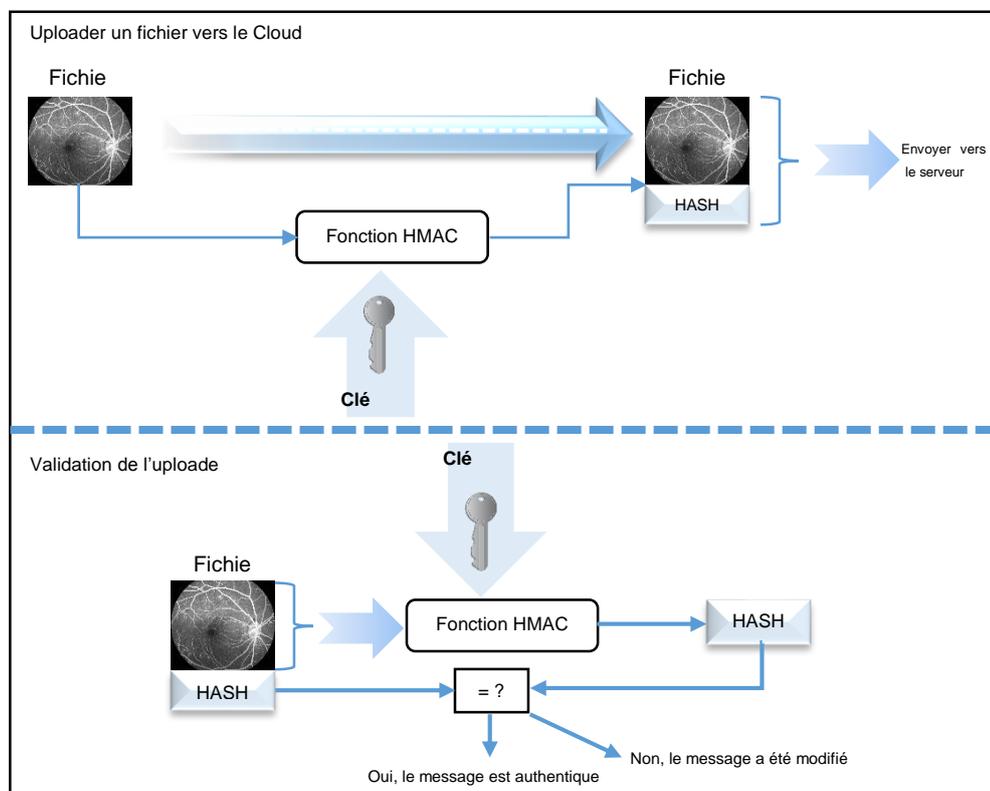


Figure 3.7: L'authentification de chargement des fichiers

De plus, pour télécharger un fichier depuis un dossier personnel, le système exige aussi que l'authentification de la méthode HTTP GET. L'authentification est faite en envoyant dans le paramètre **hmac** dans la chaîne de requête le SHA1 HMAC du nom du fichier à télécharger, qui est créé avec l'utilisation du mot de passe de l'utilisateur.

Par ailleurs, pour télécharger un fichier, il est nécessaire d'envoyer les paramètres **jid** et **owner** (propriétaire). Le paramètre **jid** identifie l'utilisateur qui exécute cette action, le paramètre **hmac** est calculé en utilisant son mot de passe. On utilise le paramètre **owner** pour choisir le propriétaire du dossier où l'image doit être téléchargée. Si le paramètre **owner** est égal au paramètre **jid**, alors c'est une demande de téléchargement d'un fichier envoyé par le propriétaire du dossier (Figure 3.8). Sinon, il y a un accès à un dossier partagé. Il est donc nécessaire de

vérifier si l'utilisateur a la permission d'accéder au dossier avant de lui permettre de télécharger des images (Figure 3.9).

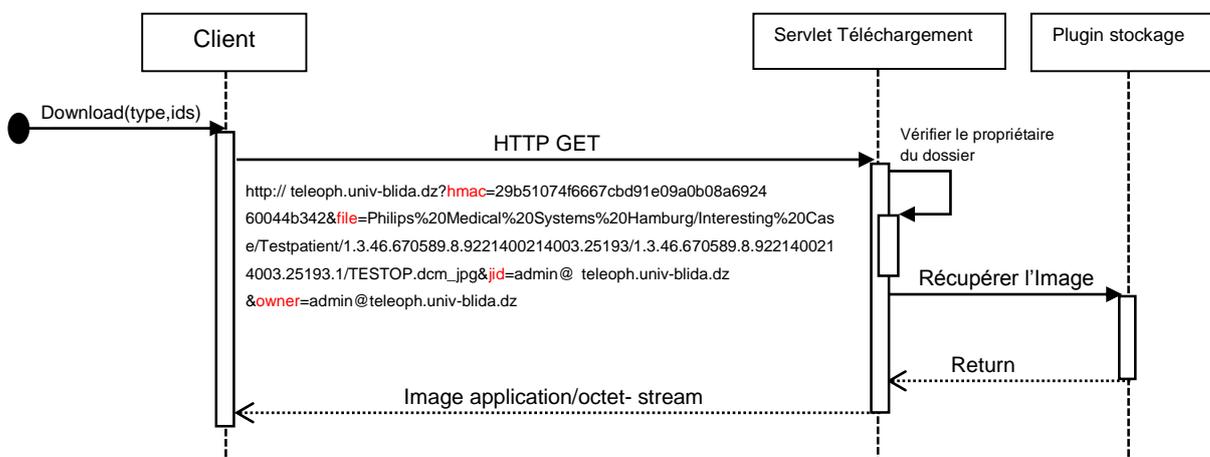


Figure 3.8: Diagramme de séquence de téléchargement d'un fichier  
Il représente le flux de travail, quand un client télécharge un fichier à partir de son dossier

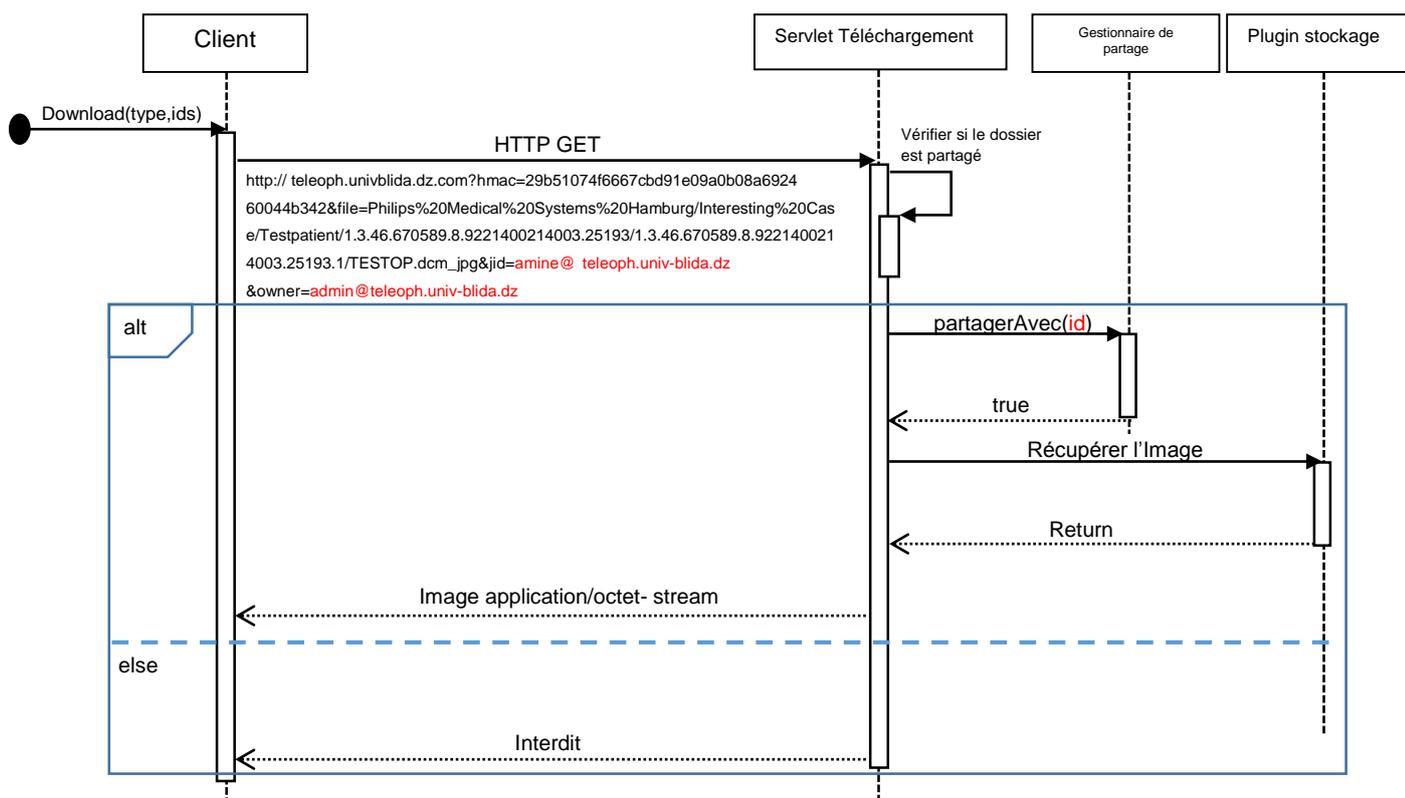


Figure 3.9: Diagramme de séquence de Téléchargement d'un fichier  
Partager représente le flux de travail, quand un client télécharge un fichier à partir d'un dossier partagé.

De plus, le canal de communication est sécurisé entre les clients et la servlet upload/téléchargement. Pour télécharger un fichier à partir d'un dossier ou uploader un fichier vers un dossier, le client doit établir une connexion HTTPS avec le serveur pour pouvoir accéder à la servlet. Par conséquent, l'amélioration du canal de communication permet d'éviter l'accès aux dossiers par des utilisateurs non autorisés et se protéger des attaques d'espionnage, par exemple l'attaque homme du milieu (MIM).

### 3.2 Passerelle du Système de Gestion d'information médicale

La passerelle est un élément très important dans l'architecture du système, car il est le pont qui permet la communication entre le réseau XMPP et le PACS. Le PSGIM (Passerelle du Système de Gestion d'information médicale) est composé de deux parties, où l'un agit comme un client XMPP et crée une interface qui permet à la passerelle de se connecter avec le réseau XMPP, et l'autre partie met en œuvre certaines caractéristiques qui permettent de communiquer avec le PACS en utilisant le protocole DICOM.

Pour communiquer avec un serveur d'archives PACS, la PSGIM utilise la bibliothèque DICOM dcm4che2, qui offre toutes les fonctionnalités pour extraire des informations à partir d'un serveur PACS via le protocole DICOM.

Sur un autre côté, la PSGIM utilise la bibliothèque du API Smack pour créer un client XMPP. En agissant comme un client XMPP, il est nécessaire d'implémenter quelques gestionnaires pour manipuler certains `<iq \>` spécifiques pour être capable de comprendre les caractéristiques de TOPCE.

La façon dont laquelle PSGIM permet aux clients XMPP Cloud d'interagir avec le PACS est basée sur un mécanisme de traduction de message. Son fonctionnement est basé sur un auditeur de paquet XMPP (packet listener), qui traite chaque paquet reçu par la passerelle. Après la réception d'un paquet, la passerelle analyse le genre d'action que ce message veut invoquer et selon l'action et les paramètres fournis, la passerelle le traduit en commande DICOM.

De plus, la passerelle permet aux utilisateurs de synchroniser leur dossier distant avec la station de travail, où la passerelle est en cours d'exécution. Les

utilisateurs doivent envoyer un `<iq \>` spécifiques, tel que défini dans le TOPCE, ce `<iq \>` spécifiques indiquent de façon explicite le dossier qui devrait être synchronisé. Après la réception de ce `<iq \>`, la passerelle procède le téléchargement de tous les fichiers qui appartiennent au dossier et les stocke dans un dossier spécifique de la station de travail.

### 3.2.1 Architecture de la passerelle PACS

Le rôle de ce composant est d'agir comme une interface entre le réseau XMPP dans le Cloud et le réseau DICOM. Par conséquent, l'implémentation du système est construite avec deux composantes principales (Figure 3.10) qui fournissent une interface pour chaque réseau :

- DCM4CHE

Ce composant fournit une API avec des outils utilisés dans la création de l'application DICOM, qui est capable d'interagir avec le réseau PACS.

- SMACK

Cette bibliothèque fournit une API qui peut être utilisée pour mettre en œuvre un client XMPP afin d'interagir avec le réseau XMPP.

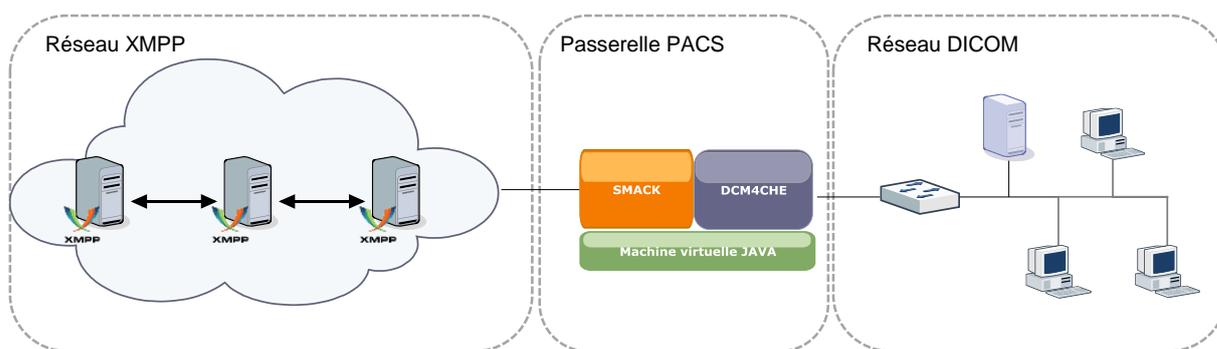


Figure 3.10: Diagramme composants de la passerelle PACS

### 3.2.2 Interface XMPP

La passerelle dispose une interface Web qui utilise le protocole HTTP pour créer un flux XMPP. Mais, le protocole HTTP ne fournit pas de flux bidirectionnels.

Par conséquent, nous avons utilisé le protocole BOSH défini par «*XEP-0124: Bidirectional-streams Over Synchronous HTTP*» [40] et «*XEP-0206: XMPP Over BOSH*» [41]. Le protocole BOSH utilise la méthode HTTP POST et le *long pooling* pour renforcer la diffusion des stanzas XMPP. Ce protocole et le client XMPP ont été implémentés en utilisant l'API SMACK, qui devait être renforcé avec quelques fonctionnalités pour permettre l'utilisation du protocole BOSH dans la connexion entre les clients et le serveur. D'ailleurs, pour permettre à la passerelle PACS d'interpréter et de réagir quand elle reçoit une stanza XMPP spécifique, nous avons mis en œuvre certains messages d'audite (messages listeners) et fournisseurs de paquets (packet providers). Le diagramme de classe suivant (Figure 3.11) représente les entités qui permettent à la passerelle de communiquer avec le réseau XMPP.

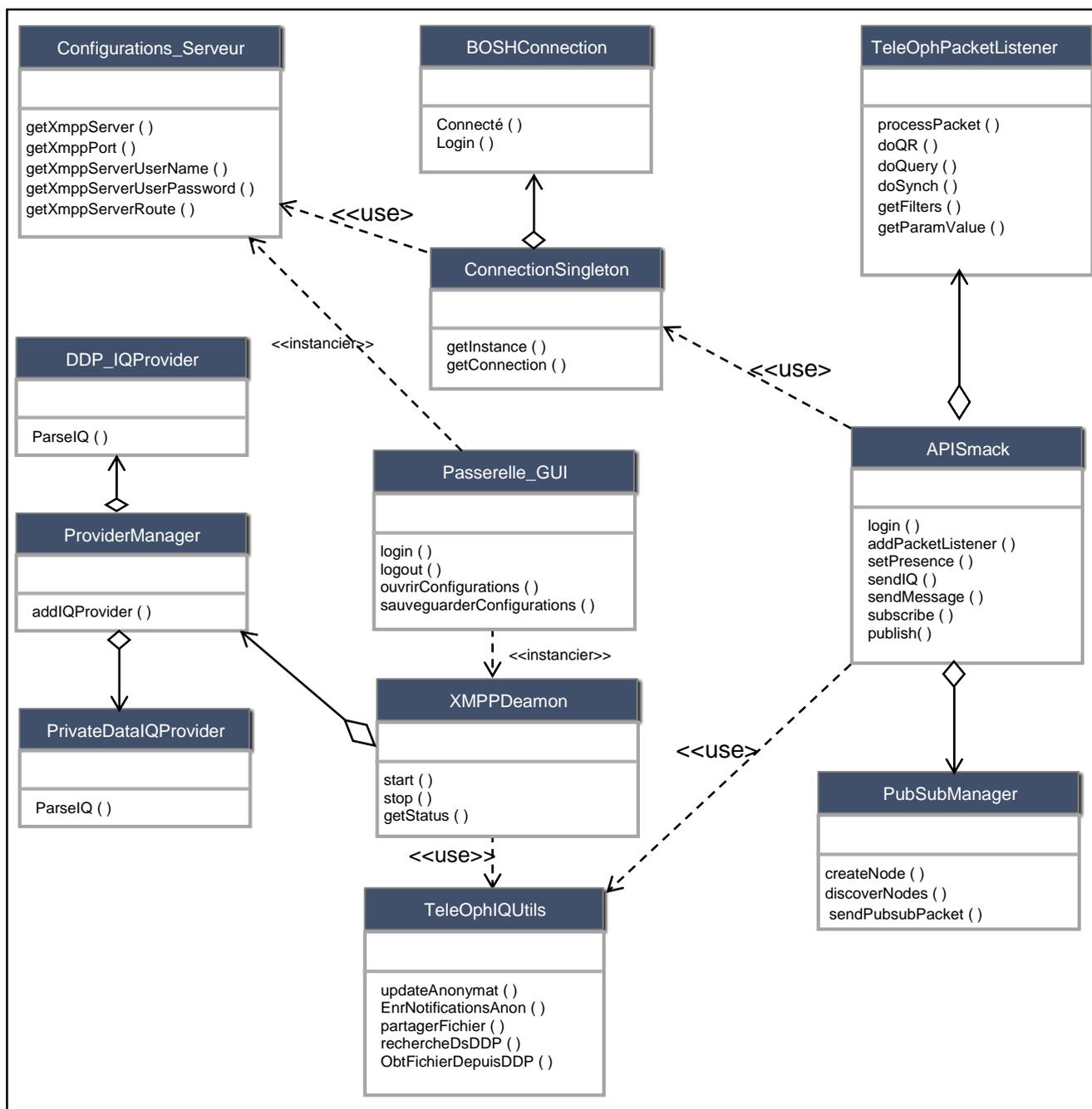


Figure 3.11: Diagramme de classes montrant les entités qui interagissent avec la couche de XMPP. JabberSmackAPI

JabberSmackAPI est une entité du client XMPP qui implémente plusieurs fonctionnalités utiles comme, par exemple, se connecter, envoyer des messages XMPP, envoyer IQ XMPP, publier une notification, etc. De plus, il utilise la fonction `addTeleOphPacketListener` pour enregistrer un `TeleOphPacketListener`, qui va intercepter tous les messages de TOPCE reçus par la passerelle.

### 3.2.3 Interface DICOM

Pour l'interface de l'infrastructure PACS, une bibliothèque DCM4CHE a été utilisée. Cette bibliothèque a permis la mise en œuvre de la couche de communication DICOM entre la passerelle PACS du système et un serveur d'archivage PACS. Il a été mis en place certains services DICOM, qui permettent à la passerelle demander (C-FIND) et récupérer (C-Move) des images médicales à partir des archives PACS (Figure 3.12). Pour améliorer le mécanisme de multithreading dans la tâche effectuée par la passerelle, une couche a été mise en œuvre avant d'appeler les services DICOM. Cette couche est responsable de la gestion de deux ports du pool, l'un pour Query et un autre pour Query / Retrieve. La partie la plus compliquée dans les tâches multithreading est de gérer plusieurs Query / Retrieve en même temps puisque la passerelle doit identifier chaque demande Query / Retrieve (C-Move). Cela est possible, en utilisant le paramètre DICOM MoveOriginatorMessageID [42]. Donc, lorsque le service récepteur DICOM reçoit un fichier, il est en mesure de savoir quels messages XMPP ont demandé le transfert du fichier vers le Cloud et par conséquent l'utilisateur qui a fait la demande et où ce fichier doit être stocké. Néanmoins, le stockage est effectué seulement si le propriétaire du dossier cible a accordé l'autorisation nécessaire pour l'utilisateur qui demande le stockage (Figure 3.13).



### 3.2.4 Synchroniser les dossiers

Le système fournit aussi des mécanismes pour permettre aux utilisateurs de synchroniser des dossiers du répertoire Cloud avec un poste de travail local. La fonction de synchronisation permet la synchronisation sélective. Ainsi, les utilisateurs peuvent synchroniser simplement un dossier donné qui contient une étude ou une série. L'action de synchronisation (Figure 3.14) est déclenchée par l'application web, qui envoie des messages personnalisés ([section 3.1.1.4](#)) contenant des informations sur chaque dossier en cours de synchronisation.

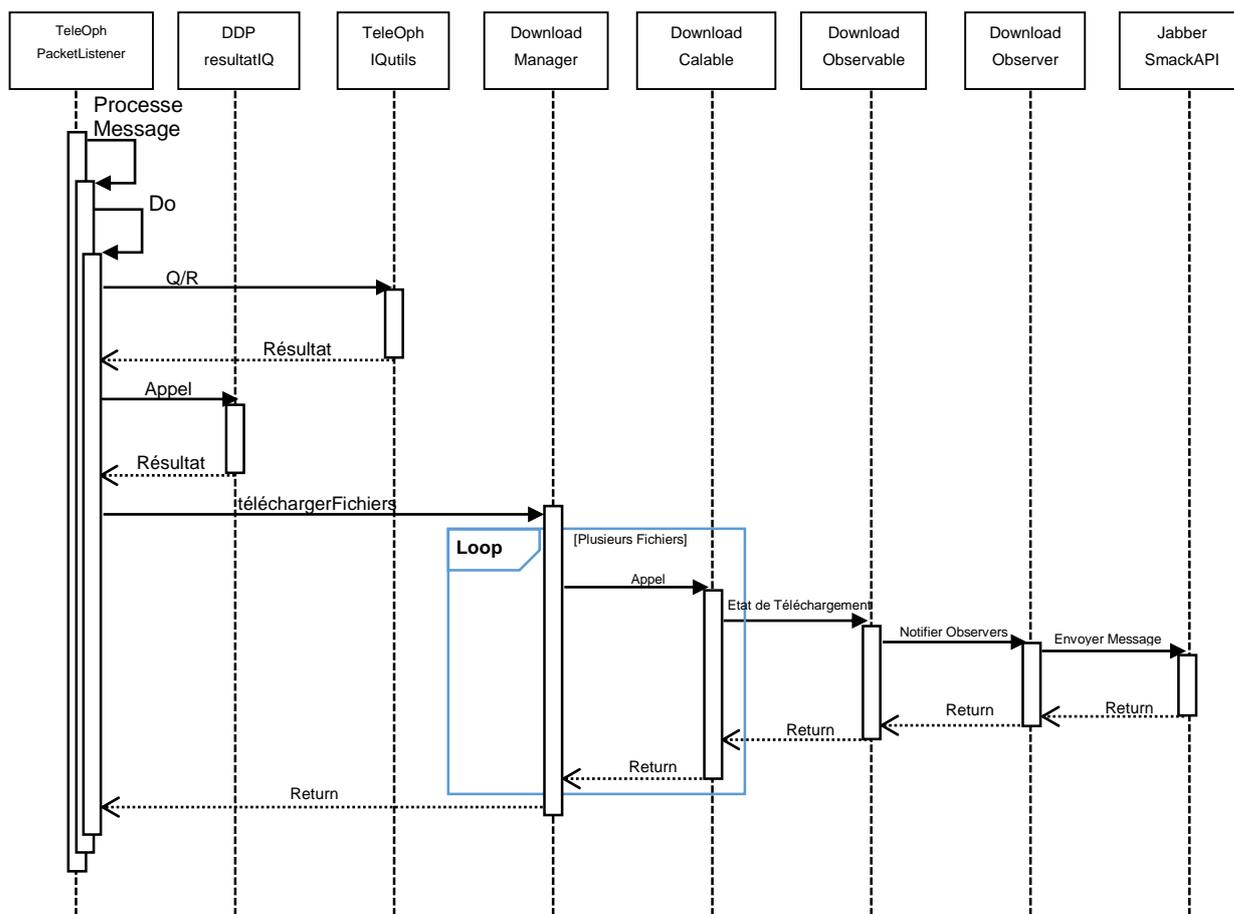


Figure 3.14: Diagramme de séquence synchronisation d'un dossier

### 3.2.5 L'anonymisation des images

Les utilisateurs sont capables de gérer l'accès à leurs passerelles PACS à travers une application Web fournie en mode SaaS. L'application Web exploite le Middleware Orienté Message pour communiquer avec les passerelles PACS. Ainsi, il y a des messages spécifiques qui peuvent être utilisés pour partager ces accès et définir les règles d'anonymisation. Les règles d'anonymisation sont stockées dans

le serveur en utilisant l'extension de stockage *Private XML*. Pour notifier les passerelles à propos de changement des règles, un modèle *publish-subscribe* est utilisé, où les passerelles souscrivent un canal utilisé par le propriétaire des ressources pour publier les règles d'anonymisation. Après changement des règles, la passerelle est notifiée et met à jour les nouvelles règles (Figure 3.15).

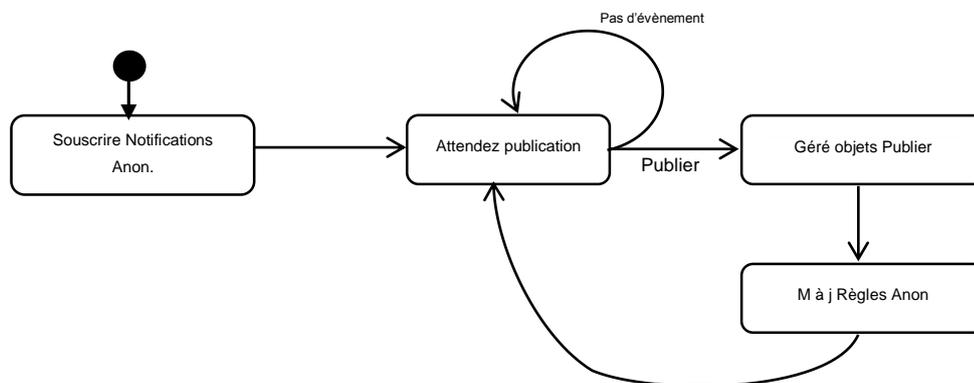


Figure 3.15: Mise à jour des règles d'anonymisation via le mécanisme publish-subscribe

Avant d'uploader une image vers le Cloud une vérification des règles d'anonymisation est appliquée. Les règles sont utilisées pour vérifier si un fichier doit être envoyé anonymisé pour un utilisateur donné. Par défaut, les fichiers sont envoyés anonymisés, mais le propriétaire de la passerelle peut gérer les règles d'anonymisation pour chaque partage.

Lorsque la passerelle reçoit un message pour exécuter Query / Retrieve, elle démarre le récepteur DICOM (s'il n'est pas démarré encore) pour commencer à recevoir des images DICOM est après cela, elle exécute les commandes DICOM **C-Move**. Dès qu'elle reçoit une image, elle vérifie les règles d'anonymisation afin de savoir si elle devrait (ou non) anonymiser l'image avant de la transférer vers le Cloud. La figure 3.16 montre comment récupérer les images médicales avec des fonctionnalités d'anonymisation.

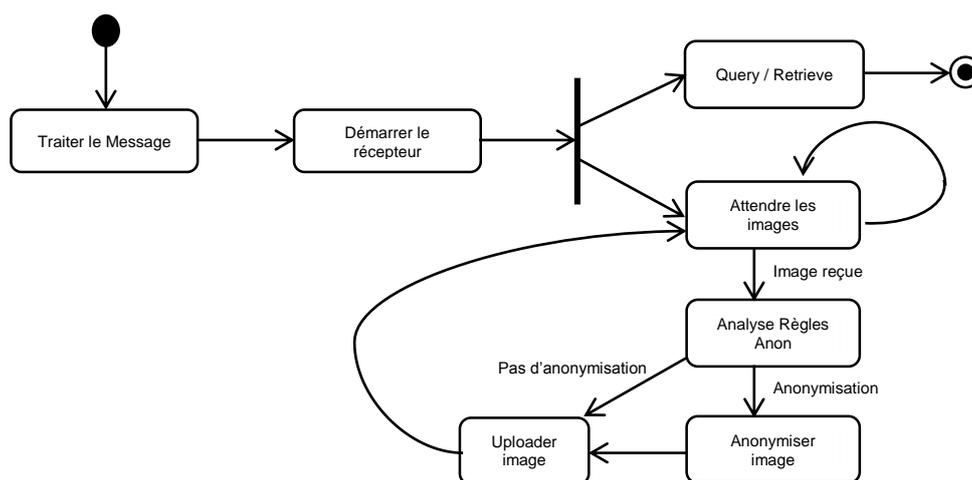


Figure 3.16: l'envoi des images avec des fonctionnalités d'anonymisation

### 3.3 Client Web2.0 de la plateforme DzTeleOph

Actuellement, les applications Web sont très améliorées par des langages de programmation des interfaces qui sont disponibles et prises en charge par les navigateurs web. Les grandes améliorations des API Web proviennent de la norme HTML5, ce qui permet aux programmeurs de créer des applications Web très complexes en utilisant des API comme, par exemple, la géolocalisation, la base de données *SQL Web*, *File API*, *Drag and Drop*, *Offline application Web*, etc.

Les applications Web deviennent très sophistiquées et certaines de ces applications remplacent les applications de bureau. Les applications Web 2.0 sont couramment utilisées pour exploiter les services disponibles dans le cloud, permettant ainsi aux utilisateurs d'accéder au service de n'importe où en utilisant simplement un navigateur web. Donc, ils ne doivent pas installer une application de bureau particulière pour utiliser le service.

Certainement, la création d'une application Web est une valeur ajoutée au SaaS fourni par le système DzTeleOph. Cette application web permet aux utilisateurs d'accéder à toutes les fonctionnalités de DzTeleOph, par le biais d'un navigateur Web. De plus, certains nouveaux API de HTML5 améliorent l'expérience de l'utilisateur et la qualité de l'application en général.

Le DzTeleOph client Web 2.0 a été entièrement développé en utilisant le langage de programmation JavaScript et les API HTML5. De plus, il a été mis en

œuvre en utilisant le modèle de conception MVC [43] (Figure 3.17). Comme nous pouvons le voir dans la figure 3.17, le fichier app.js est le modèle principal de l'application DzTeleOph Web2.0. Ce modèle utilise certains contrôleurs comme, par exemple html5uploader.js, chat.js, filebrowser.js, etc. pour contrôler les composants de l'interface utilisateur (Figure 3.17). Ces contrôleurs ont des fonctions qui sont déclenchées par l'utilisateur lors d'une interaction avec l'interface utilisateur. Des événements comme, glisser et déposer un fichier sur le composant uploader de l'interface utilisateur, déclenchent la fonction chargement dans le contrôleur html5uploader.js. L'action de chargement peut envoyer plusieurs fichiers vers le serveur, donc le contrôleur doit mettre à jour l'état du composant interface utilisateur avec le progrès de chargement.

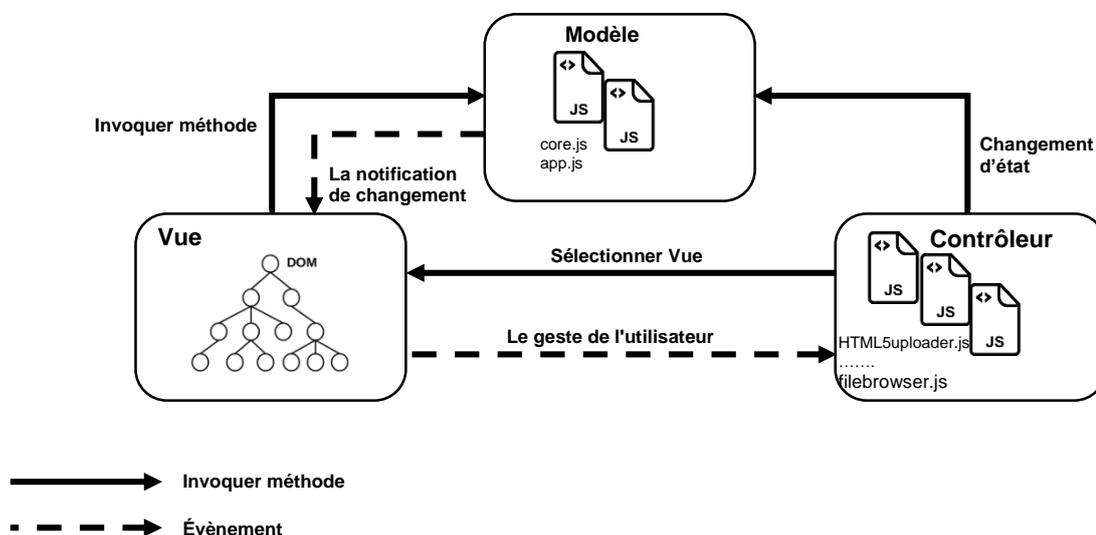


Figure 3.17: Modèle de conception MVC de l'application Web

La structure de l'interface utilisateur est simple (Figure 3.18) ; elle permet aux utilisateurs d'avoir une interaction intuitive avec l'application Web. La plupart des fonctions sont directement accessibles via la page principale. Cette page permet aux utilisateurs d'identifier facilement des fonctionnalités telles que, par exemple, le chat, l'accès aux PACS éloignés, l'accès aux PACS partagés, l'accès aux dossiers Cloud, etc. De plus, la page principale possède une section dynamique qui est utilisée pour rendre le contenu dynamique, selon le contexte et la fonction sélectionnée.

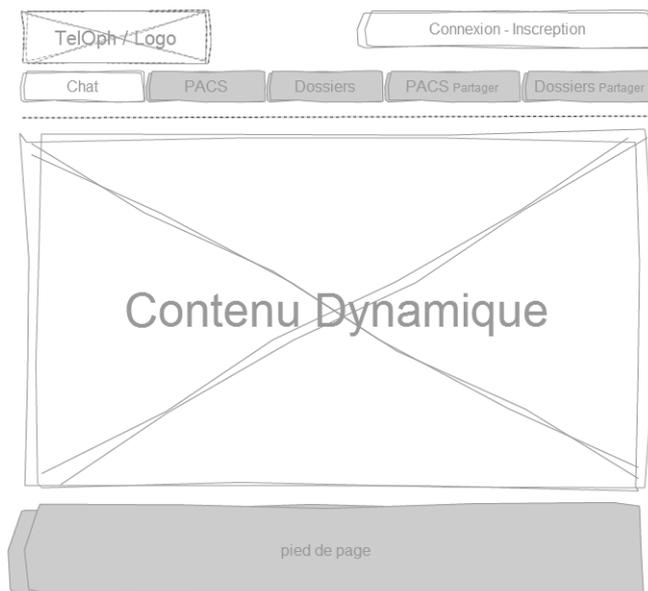


Figure 3.18: Prototype de l'application Web

### 3.3.1 JavaScript du Client XMPP

Le système DzTeleOph utilise un Middleware Orienté Message XMPP, donc l'application Web doit agir comme un client XMPP pour interagir avec d'autres composants sur le réseau XMPP.

DzTeleOphWeb client 2.0 utilise la bibliothèque Strophe.js [44] pour implémenter le client Web de XMPP. Cette bibliothèque permet à l'application Web de parler XMPP mais, contrairement à d'autres bibliothèques disponibles, elle a été conçue pour soutenir des applications qui utilisent le chat et aussi implémenter les jeux en temps réel, des systèmes de notification, moteurs de recherche, services de Cloud Computing, etc.

Comme n'importe quelle application basée sur le Web, la connexion de strophe.js au serveur distant est basée sur le protocole HTTP et Websockets. Généralement, la bibliothèque Strophe.js utilise un flux de données bidirectionnelles via HTTP synchrone (BOSH) pour se connecter avec le serveur XMPP. BOSH est une technologie de communication bidirectionnelle via le protocole HTTP, qui utilise la méthode *long pooling* [40] [41]. Outre les implémentations basées sur BOSH, il est possible de trouver une autre mise en œuvre en utilisant Websockets. L'utilisation possible de Strophe.js à la base du Websocket était retirée car actuellement, il est impossible d'utiliser sur le même port le Websockets et une application sur serveur. Ainsi, cette limitation ne nous permet pas d'exécuter le

serveur Web (pour servir l'application Web) et le Websocket sur le même port, c.-à-d. le port 80. D'ailleurs, si les Websockets utilisent un autre port, le service peut rencontrer des problèmes avec les pare feux. Les problèmes qui affectent le fonctionnement de Websocket n'affectent pas la mise en œuvre de BOSH parce qu'il est basé sur la méthode HTTP POST pur, donc il est possible d'utiliser BOSH et le serveur Web sur le même port.

Le schéma suivant (Figure 3.19) représente l'organisation du client JavaScript de XMPP qui a été mis en œuvre pour être utilisée par le client Web2.0 de DzTeleOph.

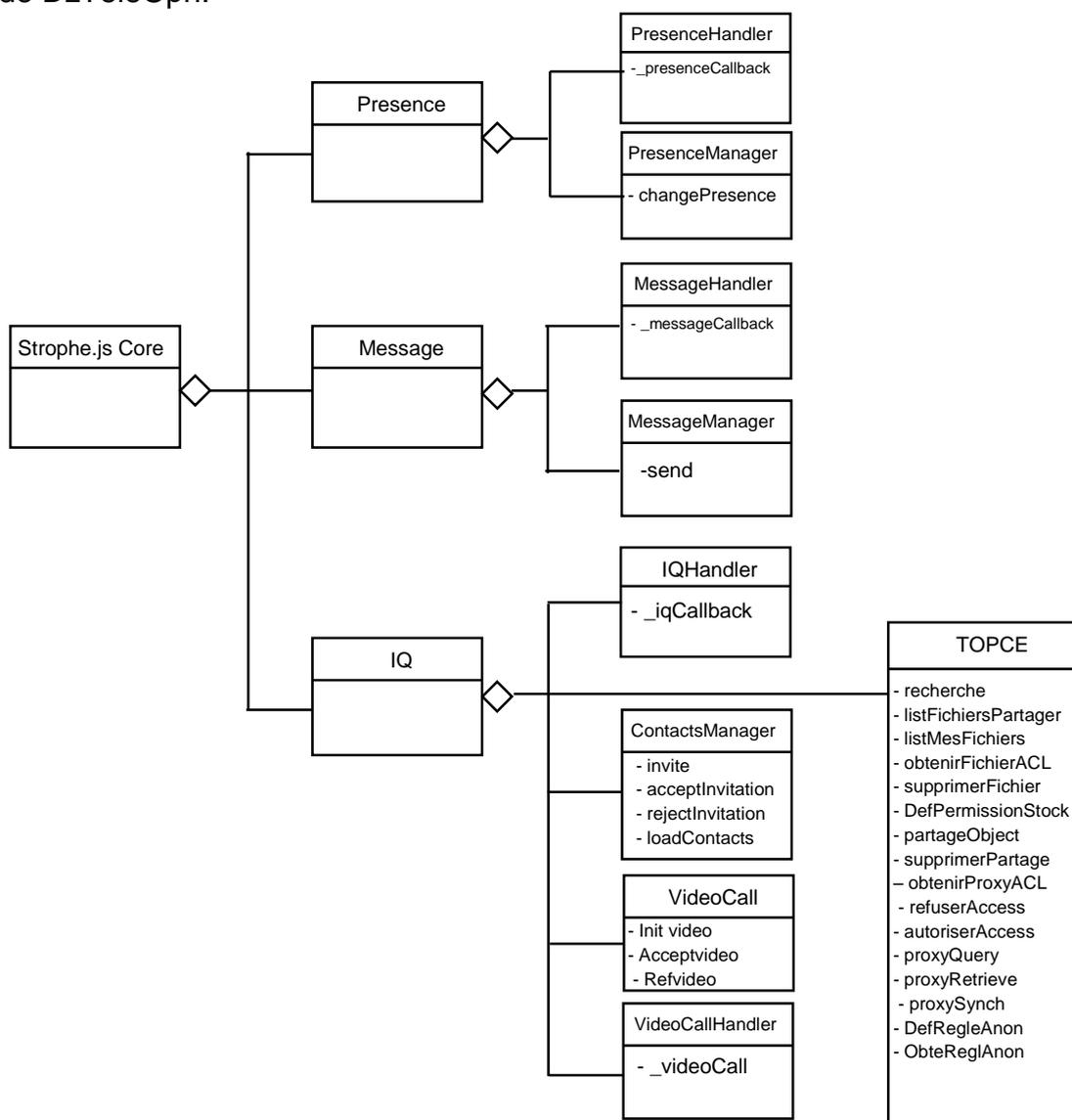


Figure 3.19: Organisation de client JavaScript de XMPP

La bibliothèque Strophe.js permet à l'application Web de comprendre les messages par défaut de XMPP, qui permettent de chatter, afficher la présence des

utilisateurs, gestion des contacts. Donc, il était nécessaire d'implémenter certaines fonctionnalités dans cette bibliothèque afin de permettre au client Web2.0 de DzTeleOph d'interpréter les nouvelles IQ introduites par le TOPCE (décrit dans la [section 3.1.1](#)), et des fonctionnalités pour faire la signalisation dans les appels vidéo.

Pour montrer la présence des utilisateurs, le système implémente un gestionnaire de présence qui est appelé à chaque fois que le noyau Strophe.js reçoit un **<presence \>** à partir d'un contact. Ce gestionnaire interagit avec l'interface utilisateur afin d'informer l'utilisateur sur une modification de présence du contact. De plus, afin de recevoir un **<presence \>** d'un contact, l'utilisateur doit souscrire à la présence du contact. Cette souscription est faite en utilisant le gestionnaire de contacts.

### 3.3.2 Html5 et JavaScript de l'Interface de l'utilisateur

#### 3.2.2.1 Glissez et déposez (Drag And Drop)

Une action naturelle que les utilisateurs font généralement sur leur ordinateur est de faire glisser et déposer des fichiers d'un endroit à l'autre. Par conséquent, cette action est disponible dans le DzTeleOph client Web2.0, où les utilisateurs peuvent glisser et déposer des fichiers à partir de leurs ordinateurs à leur dépôt personnel dans le Cloud. Actuellement, HTML5 définit un API directs pour gérer glisser-déposer.

L'API Drag & Drop de HTML5 définit certains événements natifs pour gérer les interactions des utilisateurs :

- DragStart  
Appelé lorsque l'élément "déplaçable" est en cours de commencer le premier glissement.
- DragEnd  
Appelé à la fin d'un événement glissé, réussie ou annulé.
- Dragenter  
Appelé sur un conteneur de drop cible lors du déplacement de l'élément sur la cible.

- Dragleave  
Appelé en continu lorsque l'élément est sur la cible.
- Drop  
Appelé lorsque l'événement Drag est fini par la chute de l'élément sur le récipient cible.

Dans le client Web, le JavaScript API Drag & Drop est utilisé pour indiquer au navigateur où un fichier peut être déposé. Si un fichier est déposé dans un conteneur qui permet l'action de chute (drop), un événement est déclenché et un gestionnaire est appelé pour prendre soin de cet événement.

Néanmoins, pour accéder aux octets du fichier il est nécessaire d'utiliser l'API *FileAPI* de HTML5. Cette API permet l'accès aux octets d'un fichier en utilisant la classe *FileReader*. Le *FileReader*, offre plusieurs façons de lire les données des fichiers, telles que:

- *readAsArrayBuffer*  
Crée un *ArrayBuffer* contenant les données du fichier.
- *readAsBinaryString*  
Crée une chaîne qui contient les données binaires brutes à partir du fichier.
- *readAsDataURL*  
Crée une URL représentant les données du fichier.
- *readAsText*  
Crée une chaîne de caractères qui contient le contenu du fichier.

Le Drag & Drop de HTML5 et le FileAPI ont été utilisés pour mettre en œuvre la fonctionnalité qui permet aux utilisateurs de faire glisser des fichiers à partir du disque dur de l'ordinateur local dans un conteneur spécifique dans l'application Web. Elle consiste à prendre le contenu des fichiers et permet à l'application Web d'envoyer ces fichiers vers le Cloud.

#### 3.3.2.2 Partage des fichiers

Dans la télémédecine, le partage à distance des informations est un facteur clé pour fournir une assistance à d'autres collègues ou pour obtenir l'aide dans une consultation. De plus, si l'information est facilement disponible partout, par exemple,

en utilisant simplement un navigateur Web, les utilisateurs sont plus susceptibles à coopérer avec les autres.

Pendant ce temps, le système est disponible sur internet comme un service sur Internet (SaaS), qui fournit des fonctionnalités qui permettent aux utilisateurs de partager des dossiers, qui sont dans le Cloud, avec leurs amis. Le TOPCE est utilisé pour gérer l'accès de chaque dossier que l'utilisateur possède dans son répertoire distant.

Pour assurer que seul le propriétaire d'un dossier est en mesure de le partager, l'action est limitée à la portée d'une connexion d'un utilisateur de XMPP, donc si un client Web envoie une action de partage au serveur, le serveur tente de l'exécuter en prenant en compte l'authentification de l'utilisateur qui a envoyé la stanza XML.

### 3.3.2.3 La messagerie intégrée

Pour améliorer la collaboration entre collègues, certaines fonctionnalités comme la messagerie et la notification de présence apparaissent très cruciales pour améliorer la collaboration. Dans le télétravail, les utilisateurs doivent être tout le temps en contact, donc une fonctionnalité de messagerie est très importante pour permettre aux utilisateurs de communiquer les uns avec les autres, afin d'obtenir des informations de certains patients, ou de tout autre type de travail collaboratif. De plus, l'échange de présence permet aux utilisateurs d'être informés de la disponibilité des collègues et donc être en mesure de savoir quand un collègue est prêt à discuter avec eux.

Pour créer un environnement collaboratif, le client Web utilise le serveur XMPP pour gérer la liste des contacts de l'utilisateur. La liste des contacts est gérée en utilisant des stanza XML spécifiques de XMPP qui sont envoyées au serveur pour ajouter ou supprimer des amis à partir des contacts. De plus, il faut ajouter ou supprimer des amis de la liste de l'utilisateur, pour être en mesure de recevoir une notification de présence d'un ami. L'utilisateur doit souscrire à la présence de ses amis et attendre leur accord pour avoir l'autorisation d'accéder à leur présence. La liste et l'abonnement de présence sont faits en utilisant certains stanza `<iq \>` spécifiques.

La fonction de chat est basée sur l'échange de stanza `<message \>` de XMPP entre les utilisateurs. L'application Web est capable de traiter ces `<message \>` en utilisant un gestionnaire de messages qui interceptent toutes les stanza `<message \>` reçu par le client Web. Le message est introduit dans le fil de discussion respectif et le noyau du client Web déclenche une notification afin de mettre à jour l'interface utilisateur.

Après avoir été abonné, l'utilisateur reçoit des notifications à propos de la modification de présence du collègue. Pour pouvoir notifier dans l'interface utilisateur, la modification de présence des amis, un gestionnaire fait le traitement de tous stanza `<presence \>` reçu par le client Web.

#### 3.3.2.4 Vidéo conférence basée sur WebRTC

Le système utilise l'API WebRTC de HTML5 [45] pour établir une connexion entre deux navigateurs. Cette connexion est utilisée pour transporter le flux vidéo. WebRTC est un protocole qui permet aux navigateurs Web de soutenir la communication en temps réel (RTC) via des API simples de Javascript. WebRTC permet d'enrichir, d'augmenter la qualité des applications RTC développées pour les navigateurs web à travers des API Javascript simples et de HTML5.

Comme résultat de l'utilisation du WebRTC pour implémenter la fonction de vidéo-conférence, il a été possible d'enrichir l'application DzTeleOph Web2.0 avec des capacités de vidéo conférence sans la nécessité d'installation d'un plugin dans le navigateur Web. Donc, les navigateurs qui sont compatibles avec les API HTML5 et support le WebRTC, permettent aux utilisateurs de profiter de cette fonctionnalité.

Pour établir une connexion de WebRTC, il est nécessaire d'échanger des messages afin de faire la négociation médias. Pour faire cette négociation, WebRTC utilise le SDP (Session Description Protocol) [46]. Cependant, les messages SDP doivent être échangés entre les navigateurs. Donc, l'application DzTeleOph Web2.0 utilise XMPP pour faire la signalisation. Une extension à la bibliothèque Strophe.js a été implémentée pour gérer la couche de signalisation dans l'établissement d'une connexion WebRTC (Figure 3.20). Cette extension est l'entité VideoCall représentée dans la figure 3.19. Donc, cette entité permet d'initialiser, d'accepter ou de refuser les appels vidéo.

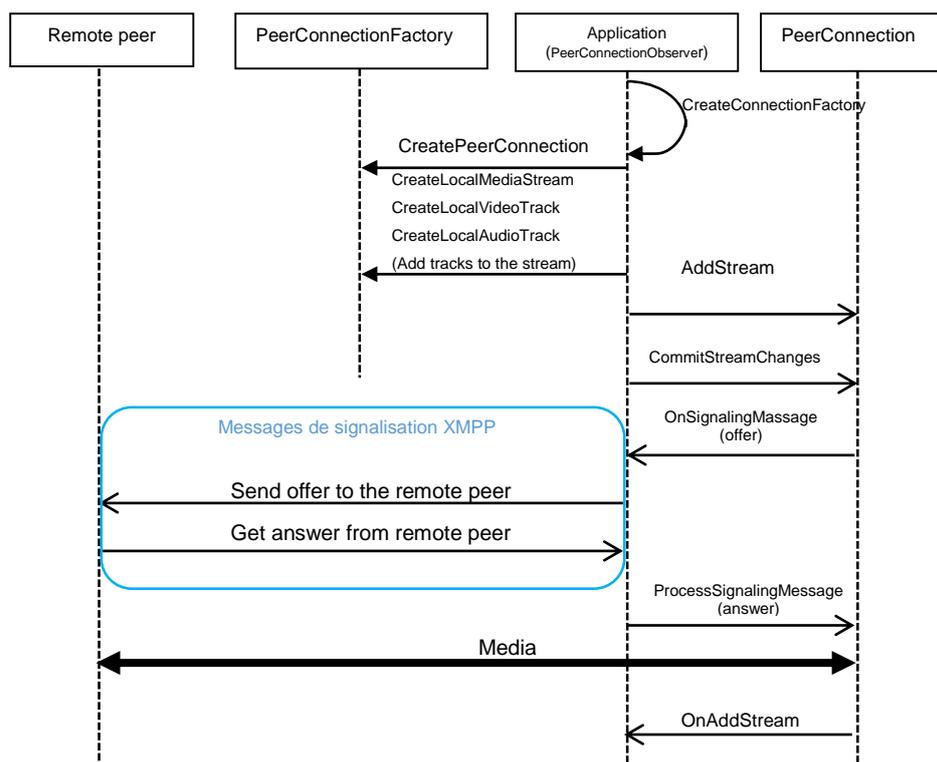


Figure 3.20: Etablissement d'une session WebRTC

### 3.3.2.5 Visualiseur d'images intégré

DzTeleOph fournit un visualiseur d'images intégré qui peut être utilisé dans le SaaS par des médecins pour analyser des images médicales. Le visualiseur fournit aussi des outils pour l'aide à la décision.

Dans la télémédecine, le visualiseur d'images est généralement très important pour enrichir les fonctionnalités du système. C'est un élément essentiel pour presque toutes les applications de télémédecine [47], [48] car il fournit des fonctionnalités très utiles pour les médecins comme, par exemple, les outils d'annotation des images médicales, traitement de l'image basée sur l'amélioration du contraste, transformation de couleur, rotation, zoom, etc.

Le visualiseur d'images créé, utilise des bibliothèques JavaScript comme, par exemple, raphael.js [49], [50] jquery.js et d'autres bibliothèques de dessin pour implémenter la couche de visualisation. Ces outils permettent aux utilisateurs de dessiner, manipuler le contraste, appliquer des filtres, mesurer la densité de pixels et d'autres.

## **CHAPITRE 4**

# **EXPERIMENTATION ET RESULTATS**

Les objectifs définis lors de la modélisation du système ont été atteints avec succès et pour illustration, un système avec un grand potentiel a été créé et qui rend possible la télémédecine dans un réseau collaboratif. Les résultats obtenus montrent le potentiel de l'utilisation du Cloud Computing pour les services de télémédecine. Le déploiement en Software-as-a-Service pour la télémédecine est parfaitement adapté à ce type de domaine. Pour implémenter un système de haute performance pour le télétravail, plusieurs aspects ont été pris en compte ; comme par exemple, la disponibilité et l'accès aux données, le travail collaboratif, la visualisation d'images médicales, l'accès aux PACS à distance, l'informatique ubiquitaire, etc. Dans la télémédecine, la disponibilité des données est très importante. Pour profiter des caractéristiques des services Cloud, un dossier personnel est mis en œuvre dans le Cloud (Figure 4. 1) où les utilisateurs peuvent stocker des images médicales numériques.

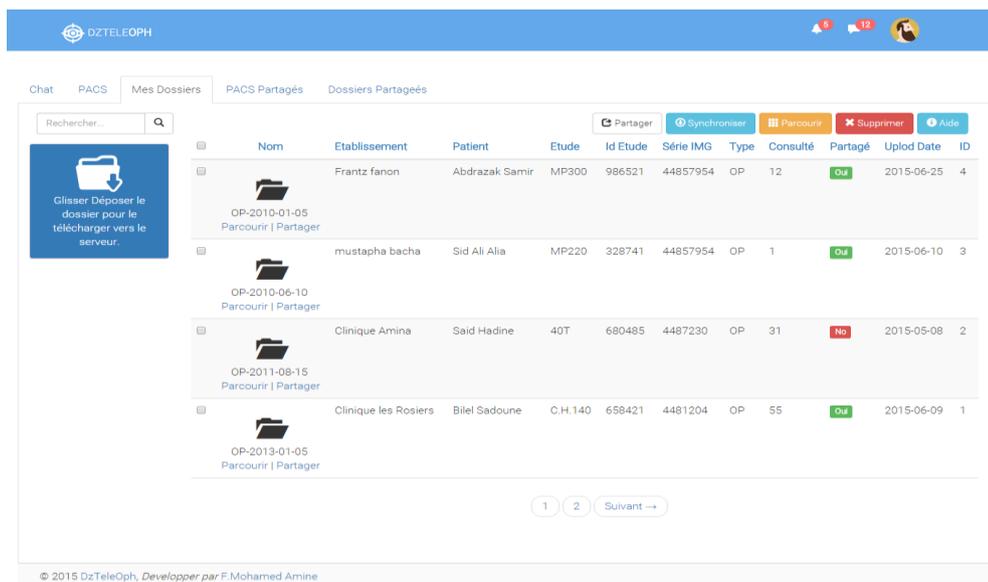


Figure 4. 1: Dossier distant dans le Cloud

Aussi, en utilisant cette fonctionnalité de dossier personnel distant, il est possible d'exploiter les services de télémédecine avec le renforcement de la collaboration entre collègues en permettant aux utilisateurs de partager leurs données avec leurs collègues (Figure 4.2). Cette fonctionnalité améliore les capacités du système, en raison du fait qu'elle crée une souplesse appliquée au passage des flux de données en télémédecine. Ainsi, si un médecin doit envoyer un examen pour être analysé par un collègue, il lui suffit seulement de partager cet examen avec son collègue qui peut ensuite y accéder à partir de n'importe quel endroit et à tout moment (Figure 4.3).

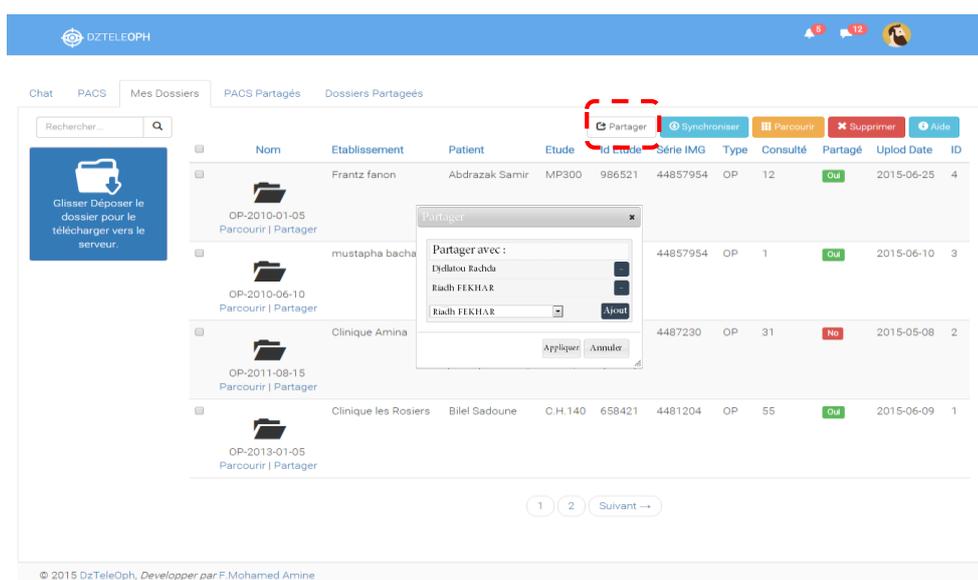


Figure 4.2: Partage des fichiers avec des collègues

The screenshot shows the DZTELEOPH web application interface. At the top, there is a navigation bar with the logo and the text 'DZTELEOPH'. Below the navigation bar, there are several menu items: 'Chat', 'PACS', 'Mes Dossiers', 'PACS Partagés', and 'Dossiers Partagés'. The 'Dossiers Partagés' menu item is highlighted with a red dashed box. Below the menu items, there is a search bar with the text 'Rechercher...'. Below the search bar, there is a table with the following columns: 'Norm', 'Etablissement', 'Patient', 'Etude', 'Id Etude', 'Série IMG', 'Type', 'Consulté', 'Partagé Par', 'Uplod Date', and 'ID'. The table contains four rows of data. The 'Partagé Par' column is highlighted with a red dashed box. Below the table, there are two buttons: '1' and '2', and a 'Suivant ->' button. At the bottom of the page, there is a copyright notice: '© 2015 DzTeleOph, Developper par F.Mohamed Amine'.

Norm	Etablissement	Patient	Etude	Id Etude	Série IMG	Type	Consulté	Partagé Par	Uplod Date	ID
OP-2013-08-05 Parcourir   Partager	UCLA Head 3T	Abdrzak Samir	UCLA MP300	986521	44857954	OP	12	Samir KADIK	2015-06-25	4
OP-2014-01-08 Parcourir   Partager	Cardio Fascular	Sid Ali Alla	UCLA MP300	328741	44857954	CT	1	Sid Ahmed ZEMIRLINE	2015-06-25	3
OP-2014-03-04 Parcourir   Partager	Clinique Amina	Said Hadine	40T	680485	4487230	OP	31	Imen DJELLATOU	2015-06-25	2
OP-2014-04-09 Parcourir   Partager	Clinique les Rosiers	Bilal Sadoune	C.H.140	658421	4481204	OP	55	Mohamed FEKHAR	2015-06-25	1

Figure 4.3: Accès aux fichiers partagés

De plus, ce système se focalise sur l'amélioration de la coopération et la collaboration dans le télétravail, avec quelques fonctionnalités implémentées qui suit le paradigme des réseaux sociaux. Ce paradigme améliore les fonctions de coopération et de collaboration. Les utilisateurs peuvent ajouter des contacts comme des amis et de suite peuvent partager des données avec eux, interagir avec eux dans les analyses des examens et faire des vidéo-conférences (Figure 4.4). Une autre caractéristique importante pour la collaboration que ce système offre est d'être informé sur l'état de présence de chaque utilisateur. Cette fonctionnalité est très importante car elle permet aux médecins de savoir quand ils peuvent communiquer avec les autres et quand leurs collègues sont en mesure de répondre, ce qui est impossible dans les systèmes classiques comme la messagerie.

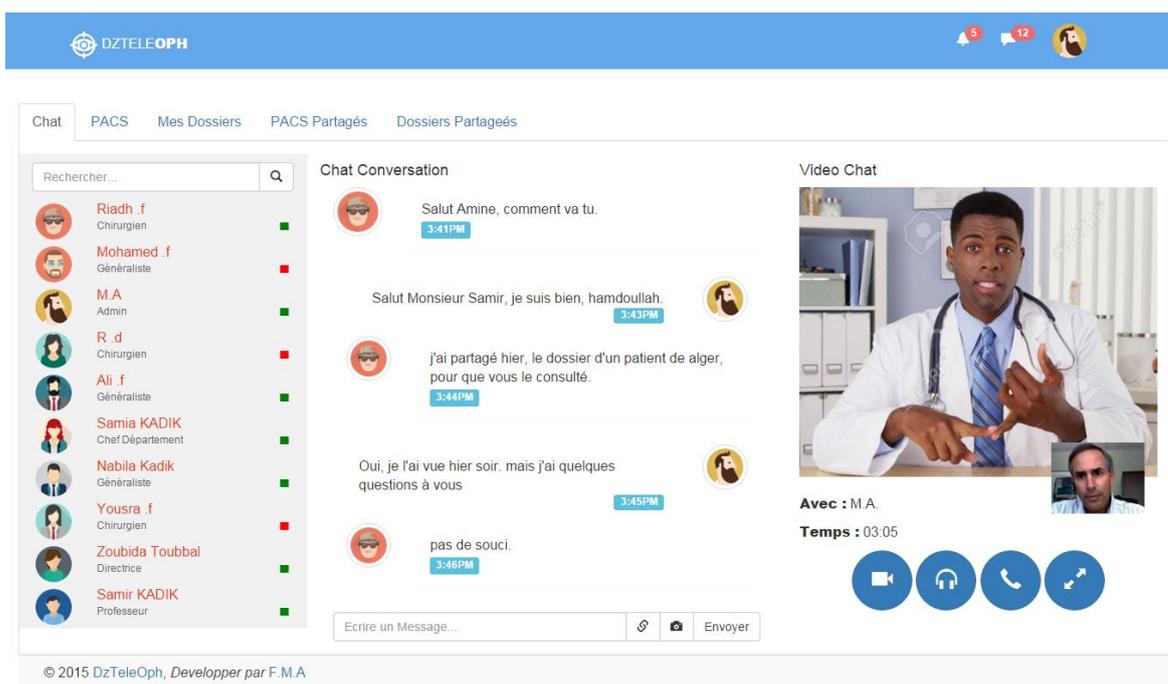


Figure 4.4: Les outils de collaboration

Certains utilisateurs travaillent généralement sur plusieurs postes de travail (par exemple, à la maison, bureau, ordinateur portable, etc.) et quelquefois ils ont besoin de synchroniser les données entre les différents postes de travail. Donc, pour augmenter les capacités du système, une fonctionnalité a été implémentée qui permet aux utilisateurs de faire la synchronisation sélective des données en sélectionnant un dossier donné, à partir du répertoire Cloud, pour être synchronisé avec un poste de travail (Figure 4.5).

The screenshot shows the DzTeleOph web application interface. At the top, there is a blue header with the logo and navigation tabs: Chat, PACS, Mes Dossiers, PACS Partagés, and Dossiers Partagés. A search bar is located on the left. The main content area displays a table of medical cases. The 'Synchroniser' button in the top right of the table is highlighted with a red dashed box. A blue box on the left contains instructions: 'Glisser Déposer le dossier pour le télécharger vers le serveur.' The table has columns for Nom, Etablissement, Patient, Etude, Id Etude, Série IMG, Type, Consulté, Partagé, Upload Date, and ID. The first row is highlighted with a red dashed box.

Nom	Etablissement	Patient	Etude	Id Etude	Série IMG	Type	Consulté	Partagé	Upload Date	ID
OP-2010-01-05 Parcourir   Partager	Frantz fanon	Abdrzak Samir	MP300	986521	44857954	OP	12	Oui	2015-06-25	4
OP-2010-06-10 Parcourir   Partager	mustapha bacha	Sid Ali Alia	MP220	328741	44857954	OP	1	Oui	2015-06-10	3
OP-2011-08-15 Parcourir   Partager	Clinique Amina	Said Hadine	40T	680485	4487230	OP	31	No	2015-05-08	2
OP-2013-01-05 Parcourir   Partager	Clinique les Rosiers	Bilel Sadoune	C.H.140	658421	4481204	OP	55	Oui	2015-06-09	1

© 2015 DzTeleOph, Développeur par F.Mohamed Amine

Figure 4.5: Synchronisation des données du Cloud avec une passerelle

La fonction de synchronisation permet aux utilisateurs de synchroniser des fichiers dans le Cloud avec un poste de travail. Cela permet de réduire le temps de latence et permet aux utilisateurs d'ouvrir des images médicales avec des applications spécialisées. Cependant, le système DzTeleOph fournit un visualiseur des images DICOM intégrée, implémenté avec HTML5 et JavaScript, il rend possible la visualisation des images médicales DICOM directement à partir du Cloud, sans avoir besoin de les télécharger (Figure 4.6).



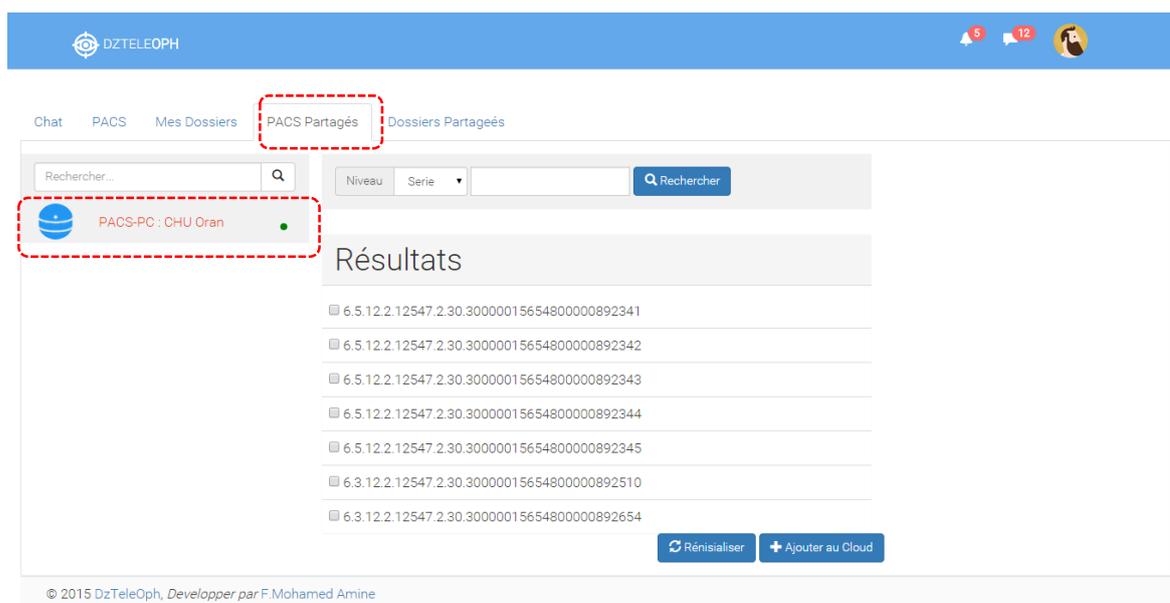


Figure 4.7: La recherche dans le PACS distant à partir de l'application web

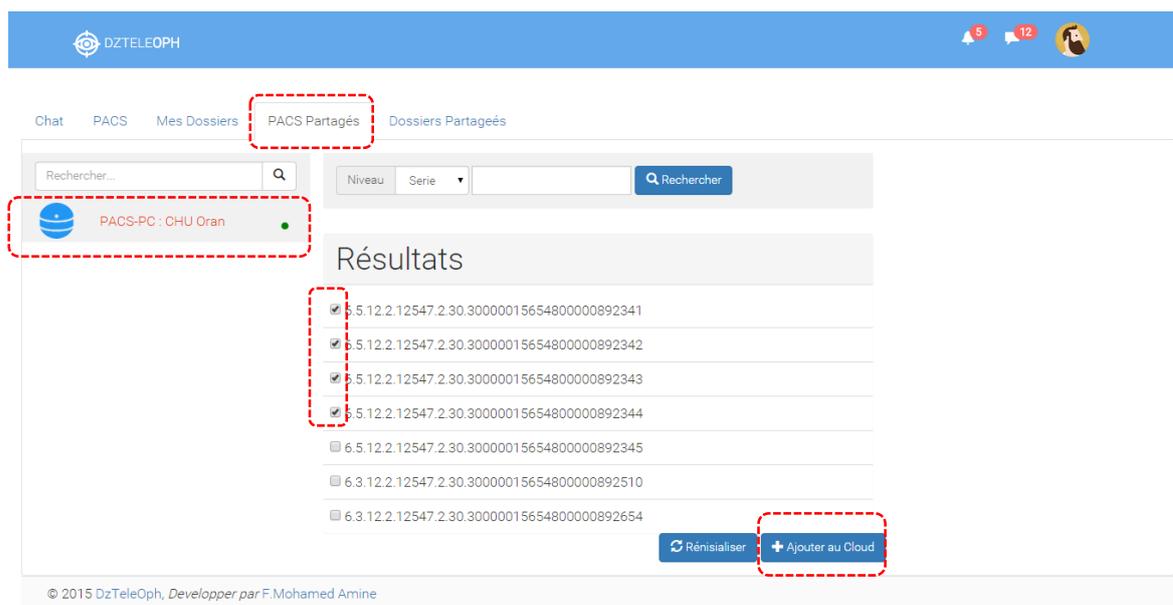


Figure 4.8: Sélection et Récupération des images à partir du PACS distant

De plus, pour favoriser la collaboration et l'environnement coopératif dans l'application web, les utilisateurs peuvent aussi partager leurs PACS avec des collègues pour faire de la sous-traitance, par exemple, mettre en œuvre des rapports. Pour des raisons de confidentialité, le propriétaire du PACS peut définir la manière avec laquelle les données sont accédées en appliquant le principe de l'anonymisation. Le cas échéant, lorsque ses collègues récupèrent les examens à partir de ce PACS, les informations sensibles leurs sont anonymes (Figure 4.9).

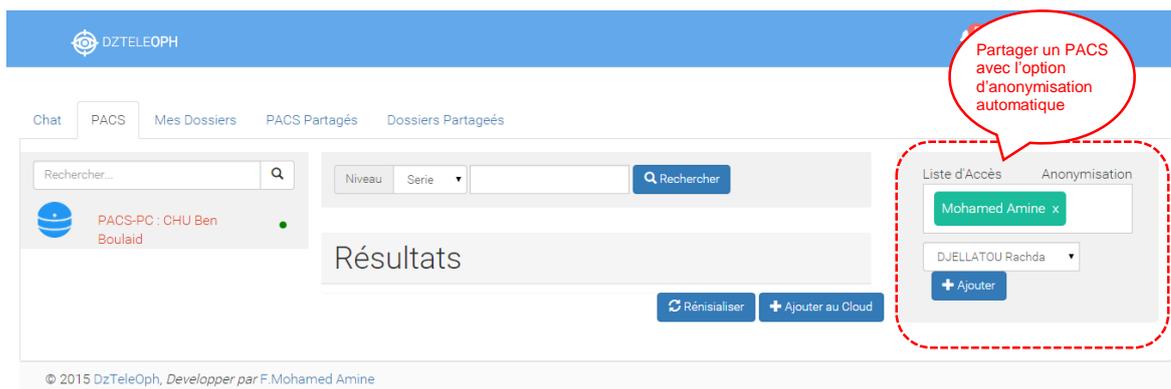


Figure 4.9: Partager PACS avec des collègues

L'interface graphique de la passerelle permet à l'administrateur de visualiser les utilisateurs actifs dans cette passerelle. Aussi, elle donne la possibilité de mettre certaines configurations de la passerelle (Figure 4.11). Par exemple, il est possible de définir l'adresse et le port du serveur PACS, l'adresse et le port du serveur XMPP, le dossier local où sont stockés les fichiers synchronisés à partir du Cloud, etc. Ces configurations sont stockées dans un fichier de configuration qui est chargé au lancement de la passerelle. Après la configuration de la passerelle et l'insertion des informations d'identification pour authentifier (Figure 4.10) la passerelle dans le réseau XMPP, elle devient opérationnelle et prête à recevoir et répondre à tous les messages.

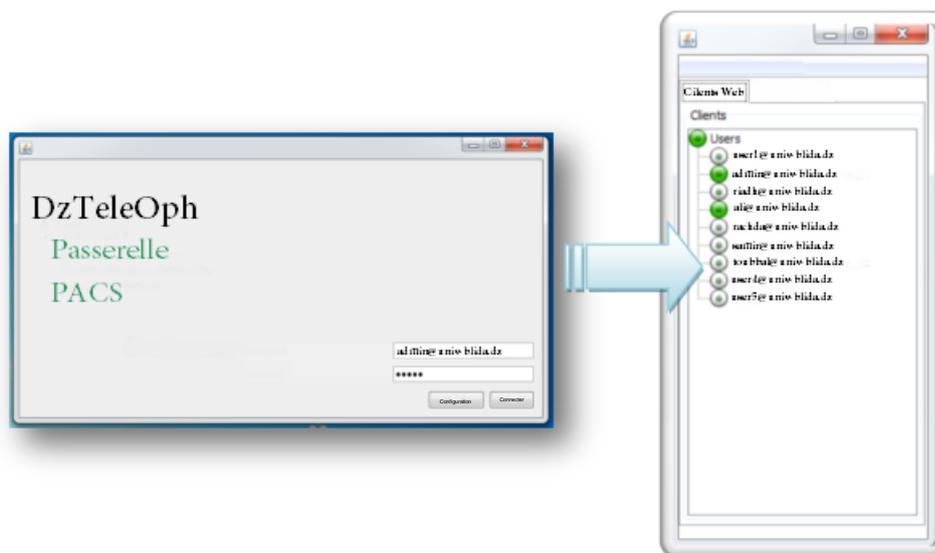


Figure 4.10: L'interface graphique de la passerelle PACS

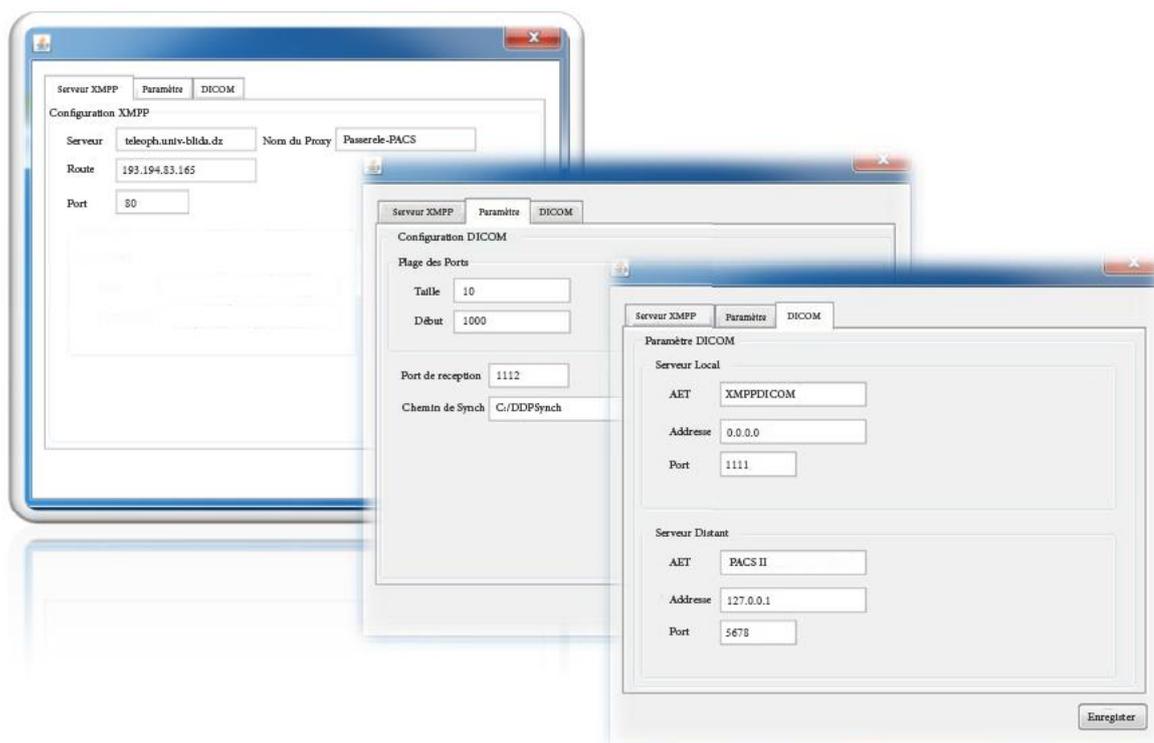


Figure 4.11: L'interface graphique de la configuration de la passerelle PACS

## CONCLUSION GÉNÉRALE ET PERSPECTIVES

L'objectif principal est d'explorer à quel point les nouvelles technologies, en particulier le Cloud Computing, peuvent améliorer le flux de travail et le flux de données en télémédecine en général et en télé ophtalmologie en particulier. De plus, pour améliorer l'assistance dans la télé ophtalmologie, nous avons étudié comment un système de collaboration peut améliorer la manière dont les médecins interagissent les uns avec les autres afin de partager des connaissances dans l'analyse des images médicales. En gros, l'objectif de ce mémoire de magister est de créer un environnement intégré où les médecins pourraient partager des données médicales et travailler en collaboration afin d'améliorer le télétravail. D'ailleurs, cet environnement complètement intégré profite des caractéristiques du Cloud pour fournir un SaaS, auxquels les utilisateurs peuvent accéder à l'aide d'une connexion Internet à partir de n'importe quel endroit et à tout moment.

Pour implémenter un système qui atteint ces objectifs, une infrastructure a été créée, qui intègre plusieurs sources de données distribuées dans un système unique avec un middleware orienté message pour connecter les sources de données avec le noyau du système. De plus, dans le déploiement du système, on a utilisé un IaaS dans lequel a été installé un serveur XMPP qui échange avec le Middleware Orienté Message, une base de données SQL (pour indexer les informations des utilisateurs et les sources de données) et un service de stockage en Cloud pour stocker des images médicales.

Sans aucun doute, la mise en œuvre de ce système, en utilisant le Cloud Computing, est cruciale pour profiter de la coopération et la collaboration dans la télé ophtalmologie. Ainsi, un serveur XMPP a été choisi pour créer un environnement coopératif où les utilisateurs peuvent interagir entre eux par chat et vidéo conférence. De plus, le système de collaboration offre un mécanisme de notification de présence aux utilisateurs afin de connaître la disponibilité de leurs collègues.

Le système final permet aux utilisateurs de partager des images médicales avec leurs collègues avec une simplicité similaire à celle des réseaux sociaux. Pour activer cette fonction, il était important d'utiliser le stockage Cloud pour archiver ces images médicales, afin d'être disponibles à tout moment et partout. De plus, le système utilise une approche basée sur les plugins pour faire le découplage de l'infrastructure de calcul à partir de l'infrastructure de stockage. Ce découplage est implémenté par la création d'un plugin avec une interface spécifique, qui a le rôle de permettre la communication entre l'infrastructure Cloud de calcul et l'infrastructure de stockage Cloud.

Afin d'améliorer la qualité de service et la collaboration entre les médecins dans la plateforme, le système permet aux utilisateurs d'accéder et de partager leur serveur d'archivage PACS avec d'autres collègues, en utilisant une passerelle pour permettre la communication entre ce système et le serveur PACS à l'intérieur d'un établissement de santé.

Le système présente un nouveau concept en utilisant les nouvelles technologies. L'adoption des systèmes basés sur le Cloud pour gérer des services de télémédecine n'est pas très utilisée dans les environnements médicaux algériens, occidentaux et même mondiaux. C'est dû aux préoccupations au sujet de la protection des images médicales et de protection de la vie privée. Mais, avec l'évolution des services Cloud et l'amélioration de leurs modèles de sécurité et de confidentialité, il y aura une adoption de plus en plus grande de ce concept pour implémenter des systèmes informatiques médicaux.

Certaines fonctionnalités pourraient être explorées davantage afin de profiter du système Cloud développé. Par exemple, des outils pourraient être développés

comme le visionnement coopératif des images en étendant quelques fonctionnalités supplémentaires pour améliorer l'aide à la décision dans l'analyse des images. Une des fonctionnalités correspondrait à un Visualiseur d'images interactif en temps réel avec plusieurs utilisateurs manipulant la même image.

De plus, des contraintes de sécurité supplémentaires pourraient être prises en compte dans la poursuite des travaux. Afin de renforcer le contrôle de partage des PACS. Le système pourrait implémenter des fonctionnalités permettant à des propriétaires du PACS de limiter certaines requêtes à certains types d'utilisateurs ou à des périodes temporelles définies

Le système développé se présente sous la forme d'un noyau bien défini pour les autres applications, construit sur un middleware orienté message. Il pourrait être étendu avec des composants capables de communiquer avec le middleware. Par exemple, il est possible d'implémenter des passerelles communicantes avec d'autres types de source d'informations pour des données médicales comme, par exemple, HL7.

## RÉFÉRENCES BIBLIOGRAPHIQUES

- 1 H. K. Huang, *PACS and Imaging Informatics: Basic Principles and Applications*. John Wiley & Sons, Inc., 2009.
- 2 G. & C. L. Yogesan K, Kumar S, *Teleophthalmology*. Springer Berlin Heidelberg, 2006.
- 3 Y. Wu, Z. Wei, H. Yao, and Z. Zhao, "TeleOph: a secure real-time teleophthalmology system," ... *Biomed. IEEE ...*, vol. 14, no. 5, pp. 1259–66, Sep. 2010.
- 4 K. Yogesan, L. Goldschmidt, and J. Cuadros, Eds., *Digital Teleretinal Screening*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012.
- 5 W. H. Organization and others, "Global observatory for eHealth series-- volume 2 telemedicine--opportunities and developments in Member States. 2010," 2013.
- 6 A. Taddei, A. Gori, E. Rocca, T. Carducci, G. Piccini, G. Augiero, P. Festa, N. Assanta, G. Ricci, and B. Murzi, "Telemedicine Network for Early Diagnosis and Care of Heart Malformations," vol. 42, Y.-T. Zhang, Ed. Cham: Springer International Publishing, 2014, pp. 268–271.
- 7 S. Jones and R. T. Edwards, "Diabetic retinopathy screening: a systematic review of the economic evidence.," *Diabet. Med.*, vol. 27, no. 3, pp. 249–56, Mar. 2010.
- 8 C. L. Webb, C. L. Waugh, J. Grigsby, D. Busenbark, K. Berdusis, D. J. Sahn, and C. a Sable, "Impact of telemedicine on hospital transport, length of stay, and medical outcomes in infants with suspected heart disease: a multicenter study.," *J. Am. Soc. Echocardiogr.*, vol. 26, no. 9, pp. 1090–8, Sep. 2013.

- 9 A. de F. dos Santos, H. J. Alves, J. T. Nogueira, R. M. Torres, and M. do C. B. Melo, "Telehealth distance education course in Latin America: analysis of an experience involving 15 countries.," *Telemed. J. E. Health.*, vol. 20, no. 8, pp. 736–41, Aug. 2014.
- 10 J. Reponen, "Teleradiology: Changing Radiological Service Processes from Local to Regional, International and Mobile Environment," 2010.
- 11 "I Do Imaging - Programs." [Online]. Available: <http://www.idoimaging.com/programs>. [Accessed: 21-Nov-2014].
- 12 E. Krupinski, "Teleradiology: current perspectives," *Reports Med. Imaging*, p. 5, Jan. 2014.
- 13 C. O. Rolim, F. L. Koch, C. B. Westphall, J. Werner, A. Fracalossi, and G. S. Salvador, "A Cloud Computing Solution for Patient's Data Collection in Health Care Institutions," *2010 Second Int. Conf. eHealth, Telemedicine, Soc. Med.*, no. ii, pp. 95–99, Feb. 2010.
- 14 C. Teng, J. Mitchell, C. Walker, A. Swan, C. Davila, D. Howard, and T. Needham, "A medical image archive solution in the cloud," in *2010 IEEE International Conference on Software Engineering and Service Sciences*, 2010, pp. 431–434.
- 15 G. Mathew and Z. Obradovic, "Improving Computational Efficiency for Personalized Medical Applications in Mobile Cloud Computing Environment," in *2013 IEEE International Conference on Healthcare Informatics*, 2013, pp. 535–540.
- 16 R. Parameswari and N. Prabakaran, "An Android Enabled Mobile Cloud Framework for Development of Electronic Healthcare Monitoring System using VPN Connection," *Int. J.*, vol. 1, no. 7, pp. 198–202, 2013.
- 17 I. Maglogiannis, C. Delakouridis, and L. Kazatzopoulos, "Enabling Collaborative Medical Diagnosis Over the Internet via Peer-to-Peer Distribution of Electronic Health Records," *J. Med. Syst.*, vol. 30, no. 2, pp. 107–116, Apr. 2006.
- 18 P. I. Informatics, *Practical Imaging Informatics*. Springer New York, 2010.
- 19 S. G. L. George C. Kagadis, *Informatics in Medical Imaging*. CRC Press, 2011.
- 20 N.E.M.A, *Digital Imaging and Communications in Medicine ( DICOM ) Part 5 : Data Structures and Encoding*. 2004.
- 21 R. K. T. Alex A.T. Bui, *Medical Imaging Informatics*. Boston, MA: Springer US, 2010.

- 22 K. Chandrasekaran, *Essentials of Cloud Computing*. Taylor & Francis, 2014.
- 23 K. Szostek and A. Piórkowski, "OpenGL in Multi-User Web-Based Applications," in *Innovations in Computing Sciences and Software Engineering*, Dordrecht: Springer Netherlands, 2010, pp. 379–383.
- 24 K.-T. Lee, H.-J. Lim, S.-H. Shin, S.-W. Kim, M.-H. Jang, Y.-H. Ahn, and Y.-D. Yoon, "An Internet-Based Telemedicine System," *IJCSNS*, vol. 7, no. 1, p. 51, 2007.
- 25 S. P. Laird, J. S. K. Wong, W. J. Schaller, B. J. Erickson, and P. C. de Groen, "Design and implementation of an Internet-based medical image viewing system," *J. Syst. Softw.*, vol. 66, no. 2, pp. 167–181, 2003.
- 26 J. F. Nunamaker, N. C. Romano, and R. O. Briggs, *Collaboration Systems: Concept, Value, and Use*. 2013.
- 27 P. Saint-Andre, K. Smith, and R. Tronçon, *XMPP: the definitive guide*. O'Reilly Media, 2009.
- 28 M. Abousharkh and H. Mouftah, "XMPP-enabled SOA-driven middleware for remote patient monitoring system," in *Information Technology and e-Services (ICITeS), 2012 International Conference on*, 2012, pp. 1–5.
- 29 A. Hornsby, "XMPP message-based MVC architecture for event-driven real-time interactive applications," in *Consumer Electronics (ICCE), 2011 IEEE International Conference on*, 2011, pp. 617–618.
- 30 "Extensible Messaging and Presence Protocol (XMPP): Core." [Online]. Available: <http://xmpp.org/rfcs/rfc3920.html>. [Accessed: 25-Dec-2014].
- 31 P. Saint-Andre and R. Davis, "Private XML Storage," 01-Mar-2004. [Online]. Available: <http://xmpp.org/extensions/xep-0049.html>. [Accessed: 25-Mar-2015].
- 32 "Play Chess Online - Free Chess Games at Chess.com." [Online]. Available: <http://www.chess.com/>. [Accessed: 07-Sep-2015].
- 33 "Servers – The XMPP Standards Foundation." [Online]. Available: <http://xmpp.org/xmpp-software/servers/>. [Accessed: 07-Jan-2015].
- 34 "Ignite Realtime: Openfire Server." [Online]. Available: <http://www.igniterealtime.org/projects/openfire/>. [Accessed: 03-Feb-2015].
- 35 S. J. <simon@josefsson.org>, "The Base16, Base32, and Base64 Data Encodings."
- 36 D. Smith, M. Miller, P. Saint-Andre, and J. Karneges, "SOCKS5 Bytestreams." XMPP Standards Foundation, 20-Apr-2011.

- 37 S. Ludwig, P. Saint-André, S. Egan, R. McQueen, and D. Cionoiu, “XEP-0167: Jingle RTP Sessions,” 2009. [Online]. Available: <http://xmpp.org/extensions/xep-0167.html>. [Accessed: 20-Mar-2015].
- 38 F. Breu, S. Guggenbichler, and J. Wollmann, “HMAC: Keyed-Hashing for Message Authentication,” *Vasa*, 2008. [Online]. Available: <http://tools.ietf.org/html/rfc2104.html>. [Accessed: 10-Mar-2015].
- 39 B. Sullivan and V. Liu, *Web Application Security, A Beginner’s Guide*. McGraw-hill, 2011.
- 40 I. Paterson, D. Smith, P. Saint-Andre, J. Moffitt, L. Stout, and W. Tilanus, “Bidirectional-streams Over Synchronous HTTP (BOSH).” XMPP Standards Foundation, 09-Apr-2014.
- 41 I. Paterson, P. Saint-Andre, L. Stout, and W. Tilanus, “XMPP Over BOSH.” XMPP Standards Foundation, 09-Apr-2014.
- 42 “DICOM PS3.7 2015c - Message Exchange.” [Online]. Available: [http://medical.nema.org/medical/dicom/current/output/html/part07.html#sect\\_9.1.1.1.7](http://medical.nema.org/medical/dicom/current/output/html/part07.html#sect_9.1.1.1.7). [Accessed: 01-May-2015].
- 43 a Osmani, *Learning JavaScript Design Patterns*. O’Reilly Media, Incorporated, 2012.
- 44 “Strophe.js - An XMPP library for JavaScript.” [Online]. Available: <http://strophe.im/strophejs/>. [Accessed: 17-May-2015].
- 45 “WebRTC.” [Online]. Available: <http://www.webrtc.org/>. [Accessed: 27-May-2015].
- 46 “Session Description Protocol,” 2011. [Online]. Available: <http://www.ietf.org/rfc/rfc2327.txt>. [Accessed: 27-May-2015].
- 47 A. Parikh and N. Mehta, “PACS on mobile devices,” *Proc. SPIE*, vol. 9418. p. 94180F, 2015.
- 48 A. Sharma and J. Shetty, “Advanced Medical Image Viewer for X-Ray Images,” *IJITR*, pp. 203–208, 2015.
- 49 “Raphaël—JavaScript Library.” [Online]. Available: <http://raphaeljs.com/>. [Accessed: 10-Jun-2015].
- 50 “jQuery.” [Online]. Available: <https://jquery.com/>. [Accessed: 20-Jun-2015].