

الجمهورية الجزائرية الديمقراطية الشعبية

République Algérienne Démocratique et Populaire

Ministère de l'enseignement supérieur et de la recherche scientifique

Université de 8 Mai 1945 – Guelma -

Faculté des Mathématiques, d'Informatique et des Sciences de la matière

Département d'Informatique



Mémoire de Fin d'études de Master

Filière : Informatique

Option : Informatique Académique

Thème :

**Réseau P2P pour le partage des
ressources pédagogiques en utilisant les
réseaux sémantiques.**

Encadré Par :

Dr. BRAHIMI Saïd

Présenté par :

NTIRENGANYA Delphin Juste

NTORE Yvan Axel

Juin 2014

REMERCIEMENTS

Nous tenons à remercier DIEU LE TOUT PUISSANT de nous avoir accordé la santé, le courage, ainsi que la volonté d'entamer et de terminer notre projet de fin d'étude.

Nous remercions profondément, tout d'abord, notre encadreur Dr BRAHIMI Saïd pour son excellent encadrement, sa disponibilité, son suivi permanent qui nous ont permis de mener à bien ce travail

Nous adressons nos vifs remerciements aux membres du jury d'avoir accepté d'examiner et d'évaluer notre travail

Nous remercions également tous les professeurs qui ont contribué à notre formation, qui sans leur savoir et leur compétence, nous ne serions pas à ce niveau, ainsi que tout le personnel du département informatique. Sans oublier L'Etat Algérien pour avoir été notre hôte durant notre parcours académique, ainsi que l'Etat du Burundi qui nous a octroyé une bourse en Algérie.

Que tous ceux qui, de près ou de loin ont contribué à l'élaboration de ce modeste travail, trouvent ici l'expression de nos sentiments de reconnaissance et de respect.

Dédicace

A la mémoire de mon regretté père ! Rien n'aurait pu se faire sans lui !

Le fruit de mon travail est aussi tien.

A ma mère qui a toujours, et reste, là pour moi et qui m'a donné un magnifique modèle de labeur et de persévérance. J'espère qu'elle trouvera dans mon travail toute ma reconnaissance et tout mon amour.

C'est grâce à leur soutien, leur patience et leur amour que je suis là aujourd'hui. Je leur suis très reconnaissant pour les sacrifices qu'ils ont du faire pendant mes longues années d'études et d'absence.

A mes cher frères et a ma sœur : Terry, Bertrand, Aristide et Queen

A mes tantes et mes oncles.

A chaque cousins et cousines.

A tous mes amis qui m'ont aidé de près ou de loin dans la réalisation de ce projet.

NTORE YVAN AXEL

Dédicace !

Encore merci SEIGNEUR pour ta grâce incommensurable que tu ne cesses de me manifester.

Que ce travail fasse témoignage de ma reconnaissance du soutien indéfectible que j'ai toujours joui de la part de Papa et Maman Kogoshi, et de la part de toute ma famille.

Ma pensée se tourne aussi vers tous mes amis et toutes mes amies.

Que DIEU vous prête longue vie, vous le valez bien.

Kogokogo/

Sommaire

Introduction général	1
Chapitre 1 : Partage de ressources pédagogique avec les réseaux sémantique.....	4
1. Introduction.....	5
2. Environnement numérique d'apprentissage sous TIC.....	5
2.1 Historique.....	5
2.2 Conclusion.....	6
2.3 Les réseaux "ad hoc".....	6
3. Les ressources pédagogiques.....	7
3.1. Définition.....	7
3.2. Caractéristiques d'une ressource pédagogique.....	8
4. L'ontologie.....	10
4.1. Historique.....	10
4.2. Définition de l'ontologie.....	11
4.3. Les méthodes de construction d'ontologies.....	12
4.4. Pourquoi les ontologies ?.....	13
5. Méthodes de représentation de l'ontologie.....	13
5.1 les réseaux sémantiques.....	13
5.2 Définition et Représentation.....	14
5.3 Utilisation des réseaux sémantiques.	15
6. Conclusion.....	15
Chapitre 2: Peer to Peer.....	17
2.1 Introduction.....	18
2.2 Historique.....	18
2.3 L'apparition du Peer to Peer.....	19
2.3.1 Napster.....	19
2.3.2 Limewire.....	19
2.4 Définition.....	21
2.5 Les caractéristiques du P2P.....	21
2.6 L'architecture des réseaux P2P et leurs Fonctionnements.....	24
2.6.1 L'architecture centralisée.....	24
2.6.1.1 Architecture centralisée.....	24
2.6.1.2 Fonctionnement de l'architecture centralisée.....	25
2.6.1.3 Avantages et inconvénients de l'architecture centralisée.....	25
2.6.2 L'architecture décentralisée ou distribuée.....	26
2.6.2.1 Architecture décentralisée ou distribuée.....	26
2.6.2.2 Fonctionnement de l'architecture décentralisée.....	27
2.6.2.3 Avantages et inconvénients de l'architecture décentralisée.....	28
2.6.3 L'architecture hybride.....	28
2.6.3.1 Architecture hybride.....	28
2.6.3.2 Fonctionnement de l'architecture hybride.....	29
2.6.3.3 Avantages et inconvénients	29
2.7 Conclusion.....	30

Chapitre 3. JXTA.....	31
3.1 Introduction.....	32
3.2 Historique	32
3.3 Définition.....	32
3.4 Pourquoi JXTA ?	34
3.5 Concepts de JXTA.....	34
3.5.1 Peer.....	34
3.5.1.1 Le pair simple.....	35
3.5.1.2 Le pair rendez-vous.....	35
3.5.1.3 Pair routeur.....	36
3.5.2 PeerGroup.....	36
3.5.3 Pipe.....	37
3.5.4 Message.....	38
3.5.5 Les advertisements.....	38
3.5.6 Les protocoles de JXTA.....	38
3.6 L'architecture JXTA.....	40
3.6.1 La couche Noyau.....	41
3.6.2 La couche service.....	41
3.6.3 La couche application.....	41
3.7 Conclusion.....	41
Chapitre 4 : Conception.....	43
4.1 Introduction.....	44
4.2 Objectifs.....	44
4.3 Contraintes de conception.....	44
4.4 Ontologie Informatique.....	45
4.4.1 Introduction.....	45
4.4.2 Intégration ontologie dans l'application.....	45
4.5. Plan de la conception de l'application	47
4.5.1. Création d'un pair.....	47
4.5.2 Communication des Peers.....	49
4.5.2.1 Classe Chat Input	50
4.5.2.2 classe Chat Output.....	51
4.5.2.3 classe PeerListing	52
4.5.3 Le téléchargement des fichiers.....	54
4.5.4 La classe SearchFile.....	55
4.6. Fonctionnement générale de l'application.....	57
4.7 Exécution de l'application.....	58
4.8 Conclusion.....	60

Chapitre 5 : Implémentation.....	61
5.1 Introduction.....	62
5.2 Outils de développement.....	62
5.2.1 Le langage utilisé.....	62
5.2.2 La plateforme JXTA.....	62
5.2.3 JDom.....	62
5.3 Architecture de l’application.....	63
5.3.1 Interface utilisateur.....	64
5.4 Menu de la fenêtre principale.....	65
5.4.1 Option Menu.....	65
5.4.2 Option Vue.....	65
5.4.3 Option Aide.....	66
5.5 La fenêtre secondaire de l’application.....	67
5.5.1 Onglet info de connexion.....	67
5.5.2 Onglet recherche.....	68
5.5.3 Onglet Fichiers partagés.....	68
5.5.4 Onglet téléchargement.....	69
5.5.5 Onglet chat.....	70
5.5.6 Onglet Liste des pairs.....	71
5.6. Conclusion.....	71
Conclusion générale.....	72

Listes des figures

Figure	Titre	Page
Figure 1.1	La granularité d'une ressource pédagogique	9
Figure 1.2	Cycle de vie d'une ontologie	12
Figure 1.3	Relation Sémantique	14
Figure 1.4	Réseau sémantique et relation d'inclusion.	15
Figure 2.1	Interface Limewire	20
Figure 2.2	Architecture Client/ Serveur	22
Figure 2.3	Architecture réseau Peer to Peer	22
Figure 2.4	Peer to Peer centralisé	25
Figure 2.5	Peer to Peer centralisé amélioré	26
Figure 2.6	Peer to Peer décentralisé	27
Figure 2.7	Peer to Peer Hybride	29
Figure 3.1	Réseau Virtuel JXTA	33
Figure 3.2	Pair Rendez-vous	36
Figure 3.3	Unicat & Propagates Pipes	37
Figure 3.4	L'architecture des protocoles de JXTA	40
Figure 3.5	L'architecture en couches de JXTA.	41
Figure 4.1	Diagramme de classe StartJXTA	49
Figure 4.2	Diagramme de classe ChatInput	51
Figure 4.3	Diagramme de classe ChatOutput.	52
Figure 4.4	Diagramme de classe PeerListing	53
Figure 4.5	Diagramme de classe DownloadFile	55
Figure 4.6	Diagramme de classe SearchFile	56
Figure 4.7	Diagramme de classe frmMain	58
Figure 4.8	Termes et conditions de l'application	59
Figure 4.9	Identification sur le réseau	59
Figure 4.10	Choix du dossier à partager	63
Figure 5.1	Architecture de l'application	63
Figure 5.2		64
Figure 5.3		65
Figure 5.4	Matrice réalisé avant la première étape	66
Figure 5.5	Matrice réalisé après la première étape	66
Figure 5.6	Dendrogramme	66
Figure 5.7	Fenêtre pour identification de services	66
Figure 5.8	Info de connexion	67
Figure 5.9	Onglet de recherche	68
Figure 5.10	onglet fichiers partagés	69
Figure 5.11	Onglet de téléchargement	69
Figure 5.12	Onglet de Chat	70
Figure 5.13	Onglet liste des pairs	71

Listes des figures

Introduction Générale.

Introduction Générale.

Au cours du XXème siècle l'informatique a connu deux grandes évolutions majeures. D'un côté, l'accessibilité de l'ordinateur personnel en grande masse, dans les années 1970, a permis au grand public de consulter, de manipuler et de créer du contenu numérique. D'un autre côté l'apparition du Web dans les années 1990, a permis de créer du contenu et de le partager avec les autres utilisateurs du Web. Ce mode de partage a rendu disponible une grande quantité de données sur le Web. Le flux et la nature de données échangées n'ont cessé de croître au fur des années. Pour permettre l'accès et le partage des données, de nombreux outils ont été proposés, en particulier des moteurs de recherche tels que Google, Yahoo, etc. ces derniers permettent aux internautes de formuler des requêtes, souvent à l'aide de mots-clés, et de récupérer un ensemble de documents, qu'ils peuvent ainsi échanger

Le monde pédagogique, lui aussi, cherche à tirer profit du développement des TIC (Technologies de l'Information et de la Communication). Avec l'apparition des réseaux informatique de partage, le monde pédagogique s'empare aussi de cette technologie de partage. On assiste, à la naissance de plusieurs plates-formes d'apprentissage en ligne (e-Learning). Ces systèmes informatiques hébergent les ressources pédagogiques (documents, cours, multimédia, ...), gèrent l'accès à ses ressources pédagogiques. On assiste à la naissance, surtout aux USA et en Europe, à la naissance d'« universités virtuelles » et de « campus numériques ».

Les réseaux de partage des ressources fonctionnent, dans leur globalité, avec l'architecture Client/ Serveur. Le nombre important d'utilisateurs qu'ils enregistrent, ces réseaux vont être victimes de leur succès. Face à une quantité massive de requêtes à traiter de la part d'un grand nombre d'utilisateurs. Cela a pour conséquence un ralentissement du système dans le traitement des requêtes, mais le plus redoutable de cette architecture c'est qu'une panne d'un serveur entraîne l'arrêt de tout un système.

La technologie Peer to Peer se substitue comme une alternative au système Client/ Serveur, en offrant un architecture décentralisée, sans l'obligatoire présence d'un serveur dans l'architecture. Le Peer to Peer est une technologie très jeune, conçu effectivement pour le partage des ressources, c'est un outil idéal pour concevoir une application de ressources pédagogiques. Elle offre aux plateformes d'apprentissage, une nouvelle approche de partage et d'interconnexion entre les utilisateurs de ces réseaux.

La plateforme JXTA offre des outils nécessaires pour concevoir une application Peer to Peer de partage de ressources. Elle intègre des notions qui permettent la description et la représentation des ressources. JXTA utilise XML (eXtensible Markup Language), langage qui fournit une information structurelle et sémantique relative aux ressources, pour l'écriture de ses spécifications. Ce qui permet aux développeurs de conceptualiser les ressources utilisées par l'application qu'ils veulent développer.

Notre travail a consisté à utiliser les offres de JXTA pour concevoir une application réseau Peer to Peer, en intégrant la notion de description des ressources par une ontologie, pour pouvoir effectuer une recherche sémantique des ressources disponibles. L'ontologie est une spécification explicite d'un concept, elle sert de guide sémantique à l'expression d'une requête, qu'on actionne dans la recherche de l'information. Le choix de l'ontologie c'est qu'elle peut être représentée par un réseau sémantique.

Notre application établit une communication réseau Peer to Peer entre les utilisateurs, elle permet d'effectuer une recherche sémantique sur les ressources mises en ligne par les différents peers sur le réseau, et établit une interaction entre les utilisateurs du réseau.

A travers ce travail, nous visons à atteindre les objectifs principaux suivant :

- Etudier la plateforme JXTA, et comprendre comment on peut l'utiliser pour développer une application réseau P2P pour le partage des ressources pédagogiques.
- Créer une interface graphique pour le partage des ressources semblable aux applications P2P existantes tel que LimeWire.
- Intégrer les réseaux sémantiques dans la recherche des ressources.

Notre mémoire s'articule sur 5 chapitres.

- Le 1^{er} Chapitre : présente un aperçu sur l'évolution des environnements numériques d'apprentissage, les ressources pédagogiques et les réseaux sémantiques.
- Le 2^{ème} chapitre : présente la notion du Peer to Peer, les caractéristiques, et les différentes architectures P2P et leurs fonctionnements.
- Le 3^{ème} chapitre : présente la plateforme JXTA, ses protocoles, et son architecture.
- Le 4^{ème} chapitre : présente la conception générale et détaillée de l'application.
- Le 5^{ème} chapitre : présente les outils utilisés pour le développement, et l'architecture globale détaillée de l'application.

Nous terminons ce mémoire par une conclusion générale et quelques perspectives.

*Chapitre I : Partages des ressources
pédagogiques avec les réseaux
sémantiques.*

1. Introduction.

L'enseignement traditionnel ou en présence d'un enseignant, est une transmission du savoir où l'enseignant est en face de ses apprenants. On parle d'apprentissage à distance, dans un contexte précis, où l'enseignant et l'apprenant sont dans des zones géographiquement différentes. L'apprentissage en ligne, en utilisant les réseaux d'apprentissage, est né de l'évolution des Technologies de l'Information et de la Communication (TIC). Au début des années 90, les TIC révolutionnent le monde pédagogique, avec la mise en ligne des ressources pédagogiques grâce notamment à l'Internet. L'intérêt pour ses ressources pédagogiques a cru avec les innovations numériques qui ont suivi l'apparition de l'Internet. Des systèmes qui mettent la connaissance à la disposition des utilisateurs ces nouvelles technologies d'apprentissage apparaissent. Au fur et à mesure ces systèmes sont utilisés par des établissements d'enseignement et des organisations pour la formation à distance. Ils vont constituer un soutien à l'enseignement traditionnel.

Dans la suite de ce chapitre on va parler de l'historique de l'évolution des environnements numériques d'apprentissage, des ressources pédagogiques et des réseaux sémantiques.

2. Environnement numérique d'apprentissage sous TIC.

L'évolution des TIC a beaucoup influencé les méthodes d'enseignement et d'apprentissage. Des systèmes pour l'apprentissage vont voir le jour. On assiste à la naissance de l'e-Learning au sein des universités virtuelles et des campus numériques. Le paragraphe qui suit, relate l'évolution historique de ces environnements.

2.1. Historique

Avant l'apparition des plateformes d'apprentissage, où les ressources pédagogiques sont numériques, les ressources pédagogiques étaient exclusivement physiques. Les ressources pédagogiques sont constitués de : photocopiés, livres, manuels scolaires, etc. L'apprentissage se faisait obligatoirement dans un espace non virtuel : salles de classes, amphithéâtre, ... L'évolution de ce mode d'apprentissage et le type de ressources pédagogiques s'opèrent avec l'évolution des TIC et l'apparition de l'internet, au début des années 90. Des ressources pédagogiques numériques apparaissent, les cours peuvent être dispensés sur des environnements numériques (virtuels). Ces environnements prennent le nom d'Environnement Numérique d'Apprentissage (ENA).

Les professionnels du Web, améliore les capacités de ce dernier dans la recherche et le partage des données. Le W3C (World Wide Web Consortium) met au point le "Semantic Web" ou le "Web Sémantique". Le Web Sémantique s'inspire de L'Intelligence Artificielle (IA), utilise l'ontologie pour faciliter la représentation, la description, et le partage des ressources. Ces innovations technologiques améliorent la qualité des ENA. Mais ces ENA fonctionnent tous avec la technologie Client/ Serveur, cela constitue un handicap non pas à

cause de la mauvaise qualité des ressources disponible, mais de la difficile portabilité de ces environnements.

Avec l'apparition du Peer to Peer, technologie de partage de ressources en réseau, qui ne nécessite pas un serveur pour gérer l'interaction entre les auteurs d'apprentissage. Les ENA essayent aujourd'hui d'effectuer une mutation vers ces environnements Peer to Peer. Les environnements Peer to Peer ne nécessitent pas de grandes infrastructures pour les abriter, ils fonctionnent sur les plusieurs types d'appareils numériques (Téléphone portable, Laptop, PDA,...) et grâce à la technologie Peer to Peer, ces appareils peuvent communiquer directement entre eux sans utiliser un serveur.

2.2. Conclusion

L'accès aux ressources pédagogiques numériques s'est banalisé, mais l'établissement de ces environnements numériques d'apprentissage nécessite reste toujours coûteux. En effet, ce type d'environnement nécessite des moyens considérables, ils exigent notamment l'existence d'infrastructures pour abriter les outils de travail nécessaire à ces environnements, mais aussi ils contraignent les acteurs à utiliser un certain nombre d'outils restreint pour pouvoir accéder à ces services. Quant aux applications de partage de ressources pédagogiques, qui utilisent la technologie Peer to Peer elles ne sont pas encore utilisées à grande échelle.

Mais d'autres technologies implémentant des réseaux existent. On va citer brièvement de la technologie "ad hoc" qui permet d'établir des réseaux sans nécessité d'infrastructures, et que des utilisateurs peuvent utiliser pour partager des ressources. Les réseaux "ad hoc" fonctionnent un peu comme les réseaux Peer to Peer car ils recherchent à établir une relation directe entre les entités désirant se connecter ensemble.

2.3. Les réseaux "ad hoc".

Les réseaux ad hoc sont des réseaux distribués, auto-organisés ne nécessitant pas d'infrastructure. Les entités formant un tel réseau doivent collaborer afin d'assurer le bon fonctionnement des services réseaux, tel que le routage. Dans un tel environnement, de nombreux algorithmes développés pour le monde filaire ne peuvent être adaptés de façon naïve sans entraîner une congestion importante du réseau qui va réduire son efficacité. Actuellement, la principale utilisation des réseaux sans fil se fait dans le cadre d'une architecture centralisée. En effet, dans le cas du GSM ou de Wifi, les usagers se connectent à un point d'accès central qui leur fournit l'accès au réseau. La recherche actuelle tend néanmoins à proposer les réseaux "ad hoc" comme solution, car n'utilisant pas de point d'accès et se basant sur la collaboration des entités formant le réseau. Ainsi, les réseaux "ad hoc", permet à chaque objet de servir de routeur afin de relayer les paquets pour supporter la communication d'autres objets n'étant pas à portée de communication les uns des autres.

Les réseaux "ad hoc" sont supportés par un groupe de travail de l'IETF (Internet Engineering Task Force), le groupe MANET (Mobile Ad hoc Network) [1].

3. Les ressources pédagogiques.

Les ressources pédagogiques sont diverses et multiformes, les formes classiques de ressources sont :

- Les photocopiés.
- Les livres à thèmes.
- Les ouvrages littéraires ou scientifiques.
- Les publications scientifiques ou publications des chercheurs.
- Les manuels universitaires, scolaires.

Aux exemples cités ci-dessus, les ressources pédagogiques constituent des objets d'apprentissage, une source dans laquelle les professionnels de l'enseignement étalent leurs savoirs, afin de les transmettre aux apprenants. Ils permettent par l'occasion la formation et la professionnalisation des apprenants, et la partage universel du savoir.

La révolution dans le partage s'est opérée avec le développement de l'informatique, le partage des ressources pédagogiques a franchi une autre étape. Il ne s'effectue plus seulement par le biais des livres dans les bibliothèques, ou par l'enseignement dispensé dans les salles de classes, ou par l'échange de courriers. L'informatique a fait que l'on puisse accéder aux ressources pédagogiques mises en ligne grâce à internet ou d'autres réseaux informatiques.

3.1. Définition.

Il est important de souligner que deux types de ressources pédagogiques existent : les *Ressources pédagogiques* classiques et les *Ressources pédagogiques Numériques*. Le terme *ressource pédagogique numérique* concerne les ressources pédagogiques disponible sur un système fonctionnant avec l'outil informatique et que l'accès aux ressources se fait via un réseau informatique. Ils sont l'outil moderne d'apprentissage, et constituent la plateforme d'échange des connaissances et du savoir entre les enseignants et les apprenants.

Il n'est pas aisé de donner la définition globale d'une ressource pédagogique classique ou numérique. Chercher une définition de la notion des « ressources pédagogiques » limpide et « consensuelle », n'est pas simple. En effet, cette notion est très large et vague de sens. Mais tout le monde s'accorde à dire qu'une ressource pédagogique classique ou numérique, constitue un ensemble de données d'enseignement disponible entre l'enseignant et l'apprenant dans un environnement donné.

L'IEEE (Institute of Electrical and Electronics Engineers) définit les ressources pédagogiques comme toute entité, numérique ou non, utilisées dans un processus d'enseignement, de formation, ou d'apprentissage.[2]

L'IEEE définit une ressource pédagogique numérique comme l'ensemble des services en ligne, des logiciels de gestion, d'édition et de communication (portails, logiciels outils, plateformes de formations, moteurs de recherche, applications éducatives) ainsi qu'aux œuvres numérisées (documents de référence générales, œuvres littéraires, artistiques ou éducatives, etc.) utiles à l'enseignant ou à l'apprenant dans le cadre d'un enseignement ou d'apprentissage utilisant les TIC.

Dans son livre « la Définition du concept de ressource pédagogique(ou éducative), Les dossiers de l'ingénierie éducative » du mars 2004, PUIMATTO Gérard définit ainsi les ressources pédagogiques :

« Les ressources pédagogiques constituent l'ensemble des informations, documents, logiciels, programmes, banques de données, et qui permettent de véhiculer, de transmettre ou d'appréhender des concepts et contenus d'enseignements ».

Les ressources pédagogiques constituent ainsi donc tout document, toute référence, toute source dans lequel l'enseignant peut tirer des données qui lui permettront de dispenser son métier de pédagogue, et cela s'applique pour les apprenants. De ces deux définitions, on remarque que les ressources pédagogiques ne se limitent pas à du simple matériel de lecture. Elles intègrent, tous les cas d'enseignement possible, car effectivement, ces ressources pédagogiques se présentent sous plusieurs formes, elles peuvent être sous forme documents (ou matériels) de types variés : textes, présentations audiovisuelles, didacticiels, multimédias, sites web. On observe de leur part une double nature, texte et multimédia. Elles diffèrent par leur taille et leur nature. Elles sont conçues, sélectionnées et éditées pour s'adresser spécifiquement aux apprenants et aux enseignants.

3.2. Caractéristiques d'une ressource pédagogique.

Le partage de ressources dans le cadre pédagogique n'est pas nouveau. Cette notion de ressources pédagogiques est primordiale, il est clair que les enseignants n'ont attendu l'apparition des TIC pour débiter la production et l'échange des connaissances. Mais avec l'arrivée des TIC, ainsi que l'évolution de l'arsenal techno-pédagogique les productions des ressources se sont considérablement déployées.

Mais une ressource pédagogique pour pouvoir être appelé tel quel, a des propriétés qu'elle doit respecter. En effet ayant un but d'enseignement, de formation, d'apprentissage, une ressource constitue une unité de contenu, qui est soit, produite, acquise, assemblée, modifiée grâce à des propriétés précises. Les ressources pédagogiques présentent trois propriétés à savoir :

- **La granularité** et une logique de réutilisabilité plus complexe. Agrégation de sous-unités de nature diverses dont la description et l'indexation sont souhaitables pour des raisons d'autorité et de réutilisabilité.

La figure ci-dessous montre les différents composants que peut avoir une ressource pédagogique en son sein.[3]

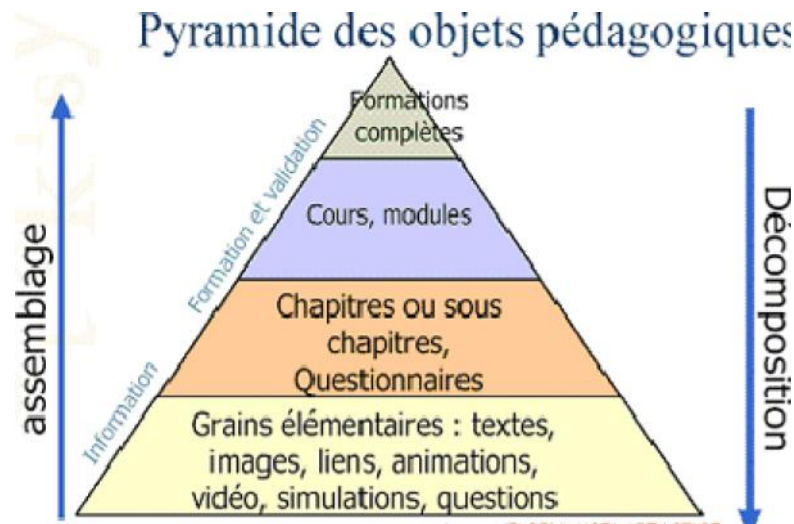


Figure 1.1 La granularité d'une ressource pédagogique

Cette figure illustre ce que peut effectivement être une ressource partagée, c'est-à-dire un ensemble de textes, d'animations, d'illustrations, de vidéos, de quizz,... Ainsi cette propriété permet, quand la ressource est partagée, de faire gagner du temps dans sa recherche. Ainsi la ressource peut être rapidement retrouvée au niveau le plus élémentaire. Aussi cette propriété permet d'élaborer des parcours personnalisés.

- **Leur niveau de dépendance et d'interopérabilité** avec d'autres ressources et systèmes. La ressource est rarement autonome mais liée à un projet et à une activité pédagogique mettant en jeu trois acteurs : l'apprenant, l'enseignant, l'auteur.
- **L'indexation** de la ressource demande des compétences qui ne sont pas strictement documentaires mais qui touchent à la définition d'un projet et au contexte d'utilisation

L'indexation a pour rôle de rendre visibles les ressources pédagogiques. Et des schémas de description pour les ressources ont été développés. Le LOM (Standard for Learning Object Metadata), est l'outil d'indexation des ressources Pédagogiques. Il répond à la nécessité de fournir un schéma de description adapté aux ressources à but pédagogique. L'utilisation de LOM permet d'échanger uniquement les métadonnées et facilite l'échange, le partage et la réutilisation de ces ressources pédagogiques. [3]

Toute entité (unité de contenu significative), numérique ou non, utilisée dans un processus d'enseignement, de formation ou d'apprentissage, constitue donc une ressource pédagogique. Cette *unité de contenu* peut être produite, acquise, assemblée, modifiée et réutilisée, grâce à un ensemble de spécifications communes, afin de construire des unités d'apprentissage plus ou moins complexes comme un module, une leçon, une évaluation, un cours (notion d'agrégation). Ces assemblages peuvent constituer à leur tour de nouvelles ressources pédagogiques, reconfigurables et utilisables à d'autres fins que celles auxquelles elles ont été initialement prévues.

L'idée maîtresse est que la ressource soit suffisamment modulaire pour être **réutilisée** dans un contexte autre que celui de sa création.

En *grosso modo*, les ressources pédagogiques correspondent à tout ce qui concerne l'enseignement et les apprentissages dans leur globalité. Elles se réfèrent en particulier au contenu pédagogique. Les matériaux sont donc créés, sélectionnés et édités spécifiquement pour les apprenants et les enseignants. Elles peuvent contenir à la fois du texte et à la fois des multimédias. Elles ne sont pas nécessairement limitées à du simple matériel de lecture. Leur taille et leur nature sont variées et peuvent se présenter sous différentes formes. Elles peuvent jouer un rôle différent selon les besoins des acteurs et sont autonomes. En définitive, les ressources pédagogiques constituent l'ensemble des informations et des matériaux qui soutiennent l'enseignement et permettent les apprentissages.

L'importance des ressources pédagogiques est fragrante, comme nous l'avons déjà cité elles constituent la pièce maîtresse dans ce scénario d'apprentissage. Mais les questions de savoir : Comment une ressource pédagogique créée va être éditée ? Comment reconnaître son contenu ? ... lorsque celle-là est partagée dans un réseau informatique, du fait qu'on sait dores et déjà l'immensité des données qu'on peut trouver sur les réseaux informatiques.

Plusieurs techniques existent pour pouvoir décrire les ressources afin de les rendre identifiables, afin de pouvoir le repérer lorsqu'on effectue recherche. Dans le cas de notre projet, nous utilisons un réseau P2P donc il est nécessaire de pouvoir intégrer une approche qui permet aux utilisateurs de pouvoir trouver et partager des données facilement et efficacement. On a choisi l'approche ontologique et les réseaux sémantiques.

4. L'ontologie.

4.1. Historique

Historiquement l'Ontologie est un mot utilisé en philosophie. L'ontologie est une branche fondamentale de la Métaphysique qui s'intéresse à la notion d'existence, aux catégories fondamentales de l'existant et étudie les propriétés les plus générales de l'être. L'Ontologie a donc un rapport direct avec notre conception de la réalité. Le terme ontologie vient du mot grec "*ôn*" ou "*ontom*" qui signifie "être" et du mot grec "*logos*" qui signifie discours ou étude. L'ontologie est la science ou l'étude de l'être.

De la philosophie à l'informatique, les ontologies apparaissent au début des années 90 dans la communauté de l'ingénierie des connaissances. McCarthy a été le premier dans le milieu de l'intelligence artificielle à s'intéresser aux ontologies de la philosophie, afin de construire des théories logiques de systèmes d'intelligence artificielle. Il affirme que pour pouvoir construire des systèmes intelligents fondés sur la logique on devait construire une ontologie du monde afin d'énumérer tout ce qui existe.

4.2. Définition de l'ontologie.

ARISTOTE a défini l'Ontologie comme la science de l'Être. Définition que l'on retrouve dans le Petit Robert avec « Ontologie » : la partie de la métaphysique qui s'intéresse à l'Être en tant qu'Être ». Le dictionnaire Larousse, lui, définit l'ontologie comme « l'étude de l'Être en tant qu'être, de l'être en soi ». Dans ces définitions, les objets ne sont pas pris dans un sens informatique mais comme objets du monde réel que le système modélise.

Du point de vue informatique et plus particulièrement ingénierie des connaissances, beaucoup de définitions ont été proposées, mais la définition la plus communément admise est celle de (*Gruber, 1993*) : « **une ontologie est une spécification explicite d'une conceptualisation d'un domaine** ». Le terme "conceptualisation" réfère à un modèle abstrait d'un certain phénomène de la réalité et qui permet d'identifier les concepts pertinents de ce phénomène. Le terme "explicite" signifie que le type des concepts utilisés ainsi que les contraintes sur leur emploi, sont réellement définies d'une manière claire et précise [4].

La conceptualisation permet d'identifier par un processus d'abstraction les concepts essentiels référencés par les termes du domaine et la spécification rend explicite le sens associé à ces concepts en leur associant une définition. Ces définitions peuvent être formelles ou non. Une ontologie est une structure permettant de représenter des connaissances en définissant des concepts et en liant les concepts par des relations [SS04, Pow03]. Bien qu'elle nécessite d'annoter ou d'indexer sémantiquement les documents et les requêtes, cette approche a l'avantage de considérer le sens des concepts plutôt que seulement leurs représentations, et les relations entre les différents concepts. Les ontologies permettent de raisonner sur les concepts qui les composent. Par exemple, il est possible de développer des méthodes d'expansion de requêtes : les requêtes sont enrichies par des concepts proches de ceux considérés initialement. [5]

Un fait remarquable sur la définition de l'ontologie du point de vue informatique, ce qu'elle fait apparaître la notion de « conceptualisation ». Une conceptualisation, se définit comme, une vue abstraite et simplifiée du monde que l'on souhaite représenter dans un certain but. Cette définition prouve donc qu'une ontologie, dans un domaine, ne peut pas englober toute la réalité de ce domaine du fait que sa perception dépend du point de vue de celui qui la crée. Aussi il faut noter la nécessité de l'indexation sémantique des documents ou ressources, si ils sont définis par une ontologie.

4.3. Les méthodes de construction d'ontologies.

Construire des ontologies est un processus complexe, dont la complexité augmente si les développeurs doivent implémenter directement l'ontologie dans un langage formel, sans avoir à leur disposition un outil d'aide à la construction. Pour faciliter cette tâche, les premiers environnements de construction d'ontologies, tel que OWL (Ontology Web Language), ont été créés au milieu des années 90. Ces derniers fournissent des outils qui aident des utilisateurs à effectuer certaines des tâches principales du processus de développement d'ontologie, telles que la conceptualisation, l'implémentation, le contrôle de cohérence, et la documentation. Ces dernières années, le nombre d'outils pour le développement d'ontologies a considérablement augmenté et a été diversifié. La majorité des outils existants concernent la création des ontologies, une fois le consensus réalisé. Les méthodes de construction d'une ontologie sont décrites dans la figure ci-dessous, elle montre le cycle de vie d'une ontologie.[6]

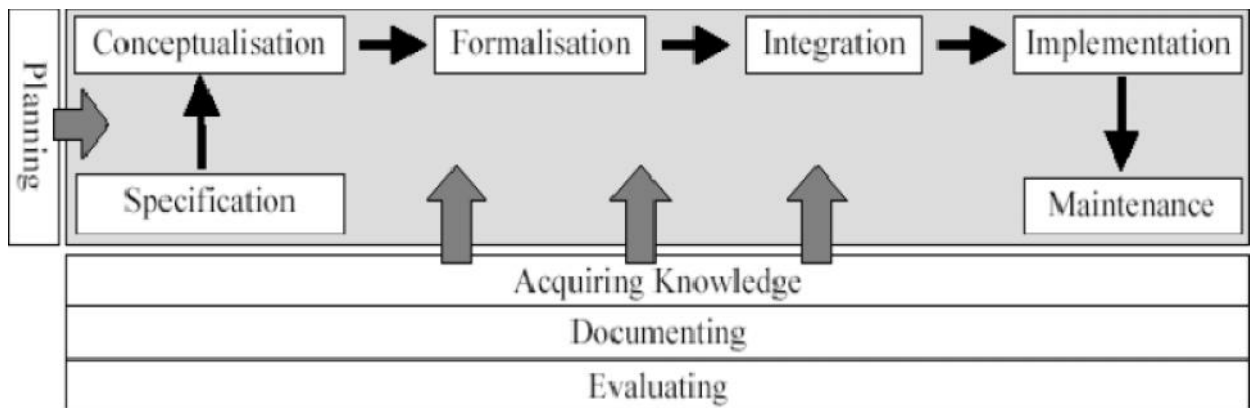


Figure 1.1 Cycle de vie d'une ontologie [Fernandez et al].

Gruber a proposé un certain nombre de principes à respecter pour construire une ontologie :

Clarté : Les ambiguïtés doivent être réduites. Quand une définition peut être axiomatisée, elle doit l'être. Dans tous les cas, des définitions en langage naturel doivent être fournies.

Cohérence : Une ontologie doit être cohérente. Les axiomes doivent être consistants. La cohérence des définitions en langage naturel doit être vérifiée autant que faire se peut.

Extensibilité : L'ontologie doit être construite de telle manière que l'on puisse l'étendre facilement, sans remettre en cause ce qui a déjà été fait.

Biais d'encodage minimal : L'ontologie doit être conceptualisée indépendamment de tout langage d'implémentation. Le but est de permettre le partage des connaissances (de l'ontologie) entre différentes applications utilisant des langages de représentation différents.

Engagement ontologique minimal : Une ontologie doit faire un minimum d'hypothèses sur le monde : elle doit contenir un vocabulaire partagé mais ne doit pas être une base de connaissances comportant des connaissances supplémentaires sur le monde à modéliser.

4.4. Pourquoi les ontologies ?

Cette question est pertinente, car tout individu prenant lecture de notre sujet de mémoire légitimement pourrait la soulever. Trouver les ressources souhaitées lors d'une recherche, peu importe le domaine, n'est pas chose facile surtout lorsqu'on utilise les moteurs classiques (google, yahoo,...). Notre sujet portant sur le partage de ressources pédagogiques sur un réseau Peer to Peer la recherche s'annoncerait plus ce que compliquée du fait que, contrairement aux réseaux Client/ Serveur, les réseaux Peer to Peer ne disposant pas de serveurs chaque client étant en même temps client et serveur, devra gérer ces recherches et ses ressources à partager .

L'introduction de l'ontologie dans l'implémentation de notre projet, a pour but d'essayer, de conceptualiser les ressources pour qu'elles aient un objectif pédagogique. Ainsi l'introduction de l'ontologie chez les utilisateurs du réseau, sera d'essayer de réduire, d'éliminer la confusion entre les demandeurs et les détenteurs de ressources. Ainsi l'ontologie va offrir aux utilisateurs du réseau, un certain nombre d'informations se trouvant au sein du système, afin de les aider dans leurs recherches, en utilisant par exemple des mots-clés. La conceptualisation du système de partage permettra ainsi aux réseaux sémantiques de représenter les ressources disponibles en établissant les relations existant entre elles.

5. Méthodes de représentation de l'ontologie.

5.1. les réseaux sémantiques.

Diverses disciplines comme la psychologie cognitive se sont intéressées au problème de la représentation des connaissances, on essaie de décrypter les mystères du fonctionnement mental de l'être humain et les fonctions actives lors d'apprentissage. Comment l'être humain acquiert, retiens et utilise ses connaissances sont sujets fondamentaux de la psychologie cognitive.

Pour la représentation des connaissances, les recherches ont abouti à un modèle qui montre comment l'information pourrait être représentée en mémoire et comment on pourrait accéder à ses informations. Les réseaux sémantiques sont donc des outils qui simulent la représentation de la mémoire humaine. Un réseau sémantique représente ainsi l'ensemble des connaissances qu'un individu se construit pour un domaine spécifique.

Les réseaux de part leurs capacités de représentation des connaissances permettent ainsi d'insister sur le concept, le contenu sémantique des documents en les reliant entre eux suivant les concepts et le domaine. Ils pourront ainsi établir les relations entre différents concepts.

5.1. Définition et représentation.

Conçus à l'origine en linguistique pour devenir ensuite un langage pour la représentation de concepts très divers, un réseau sémantique est une structure informatique très utilisée en IA.

D'une manière générale un « **Réseau Sémantique** » est défini comme une structure de graphes, acyclique, dont la fonction est l'encodage des connaissances taxonomiques concernant des objets ainsi que leurs propriétés. Il constitue une classe de structure des données facilement représentables sur l'ordinateur. Une **Réseau Sémantique** constitue une représentation des connaissances en mémoire, qui fait ressortir l'ensemble des interconnexions entre les composantes significatives.

Dans un réseau sémantique la réalité est décrite sous forme de graphes (réseaux) composés de nœuds qui représentent les concepts reliés par des arcs, exprimant la relation entre les concepts. La figure ci-dessous montre une relation avec un arc étiqueté entre deux concepts A et B

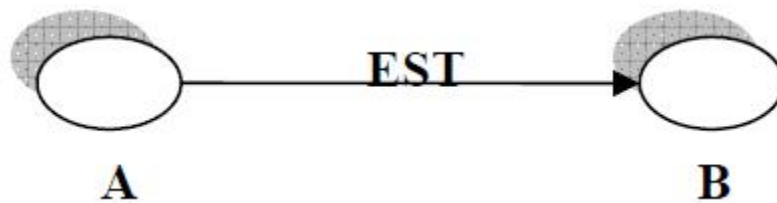


Figure 1.2 Relation Sémantique.

5.2. Représentation.

Dans les **Réseaux Sémantiques** les éléments de connaissance sont représentés par des nœuds et sont reliés par des liens associatifs ou sémantiques. Les concepts à modéliser, comme cités dans la définition, sont représentés par des nœuds, et les relations entre les concepts par des arcs étiquetés interconnectant les nœuds associés à ces concepts. Chaque concept est représenté par des attributs propres aux nœuds et par des relations qui l'associent aux autres nœuds. L'idée principale c'est de pouvoir décrire effectivement la réalité sous formes de graphes. Les nœuds et les arcs sont en général étiquetés. Aux nœuds, on associe les objets (concepts, événement, situation), aux arcs les relations entre les objets, d'où la désignation de " structure objet - relation". Les liens les plus spécifiques de ce type de réseau sont les liens " **sorte - de** " exprimant la relation d'inclusion des classes, et les liens " **est - un** " qui représente la relation d'appartenance d'un élément à une classe lorsqu'elle n'est pas exprimée dans le lien " **sorte - de** ". La figure 1.2 illustre cette relation d'inclusion représentée par " **sorte - de** ".

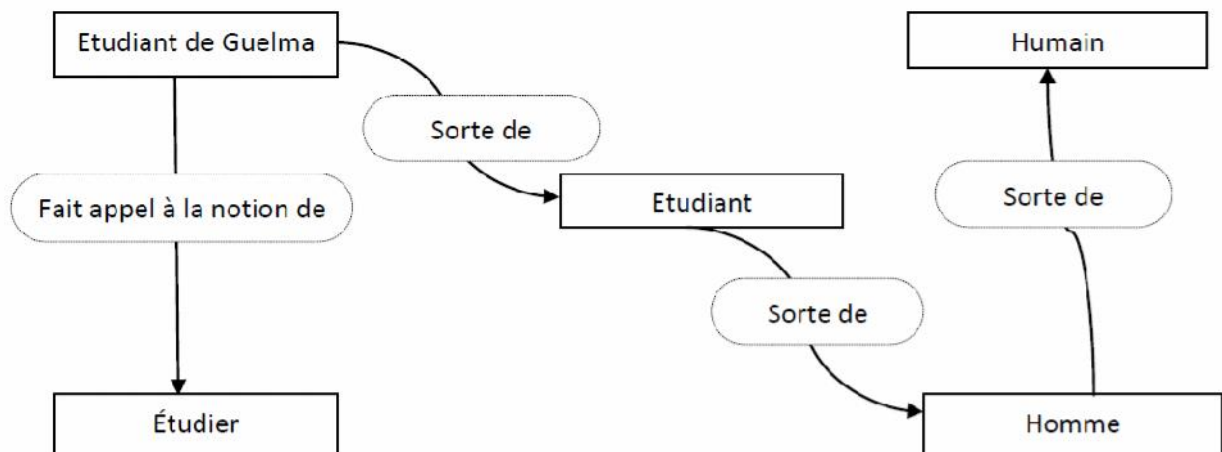


Figure 1.3 Réseau sémantique et relation d'inclusion.

Ainsi pour déterminer l'appartenance d'un objet représenté un nœud A, à un ensemble d'un autre objet représenté par un nœud B, il suffit de suivre tous les arcs allant de A vers B.

5.3. Utilisation des réseaux sémantiques.

De part leur capacité à représenter les connaissances, les réseaux sémantiques ont été beaucoup utilisés dans le domaine de l'IA. Cette capacité de représentation de connaissances offre un axe organisationnel des ressources et des connaissances dont on dispose dans un domaine. La possibilité d'organiser les ressources en une structure de graphes, établissant les liens de ces dernières sous forme de concepts, fait que les réseaux sémantiques permettent l'extraction d'informations. Ces informations sont tirées dans le graphe constitué, où il suffit de défiler à partir de n'importe quel nœud.

6. Conclusion.

Les traits essentiels et importants qu'on peut soulever à propos des réseaux sémantiques, et de leur usage peuvent se résumer en ce qui suit :

- Les connaissances représentées sont facilement accessibles du fait que les réseaux sémantiques offrent le chemin de ces dernières.
- Un puissant sens de déduction des faits et vérités des propriétés d'héritage et d'inférence offertes par les réseaux sémantiques.
- La facilité d'être représentés, parce qu'ils utilisent des graphes, et de cet effet ils offrent une facilité de compréhension pour un débutant dans un domaine représenté.
- Formalisme graphique, permettant une bonne compréhension, qui est intéressant à un premier stade de formalisation de la connaissance.

Pour une application de partage de ressources, les réseaux sémantiques offrent une possibilité organisationnelle des ressources disponibles. C'est un outil qui avec, l'ontologie de part sa capacité à conceptualiser les domaines, permet un partage objectif et non ambiguë des données que l'on souhaite partager, ou rechercher.

Le chapitre aborde la notion du Peer to Peer donne un aperçu de cette technologie, en donnant certains détails en quoi il est innovant, et les capacités qu'elle possède pour le partage des ressources.

*Chapitre II : Réseaux
Peer To Peer.*

1. Introduction.

L'internet constitue un réseau de millions d'ordinateurs connectés à travers le monde, et depuis l'apparition de ce dernier, l'échange des données est la principale activité qui se déroule sur la toile. Avec la croissance exponentielle, du nombre d'utilisateurs, observée ces dernières années, on ne cesse de chercher de nouvelles technologies pour pouvoir satisfaire la demande des internautes. L'internet a un indéniablement un rôle toujours croissant dans l'économie et dans la société, et l'informatique réseau ne cesse de chercher des innovations pour exploiter les ressources immense qu'un réseau de cet ampleur regorge.

Fin de l'an 1998, apparait **Napster** un logiciel qui va révolutionner le système de partage sur la toile, le monde de l'informatique est en effervescence, le Peer to Peer est réinventé. Mal comprise, mal identifiée et surtout mal considérée, aujourd'hui l'idée a beaucoup mûrie, qu'on qualifie le Peer to Peer d'un modèle qui a changé radicalement certaines **approches de l'informatique en réseau**.

Le terme Peer to Peer que l'on peut traduire en français par pair à pair est un système permettant d'échanger des données avec plusieurs utilisateurs. Le Peer to Peer représente aujourd'hui la plus grande source de téléchargement dans le monde et concerne l'ensemble de la société par son ampleur. Peer to Peer (P2P) désigne pour certains un type d'interconnexion, pour d'autres une technologie, mais tous s'accordent à dire que le P2P permet un partage décentralisé de données, diminue le cout et le temps dans le système de partage entre les utilisateurs.

Dans ce chapitre on va évoquer l'historique, définir, donner les caractéristiques techniques et le fonctionnement du P2P.

2. Historique.

L'informatique s'est d'abord développée autour du concept de l'ordinateur central tout puissant, au début les machines communiquaient entre elles qu'au travers de la machine centrale. Le concept a évolué pour donner place au concept client/ serveur où les utilisateurs sont attachés au serveur par un logiciel propriétaire.

Dans les années 60, du fait de la concurrence les grandes entreprises font en sorte que leurs logiciels ne soient pas compatibles avec ceux des autres. Il est donc quasiment impossible de partager des ressources et des données entre ordinateurs construits par des sociétés différentes. Cela est contraire à la logique de progression du savoir scientifique par le partage des ressources et des informations entre les chercheurs universitaires. L'Arpanet permet de mettre en réseau différentes machines omettant l'origine du produit. [7]

Arpanet est remplacé par Internet au début des années 90, l'Internet a grandi et il a fallu ajouter la centralisation pour supporter la montée en charge. Des PC manquant de ressources et de bande passante devinrent clients de réseaux c'est-à-dire qu'ils ne pouvaient par être des

fournisseurs actifs du réseau. Ces facteurs renforcèrent la nécessité de l'architecture **client / serveur**.

3. L'apparition du Peer to Peer.

Le Peer to Peer, connu avant sous l'appellation de hôte à hôte, était le premier à établir une communication d'égal à égal entre deux Peer. Ce nouveau concept permet à chaque d'agir tout comme un client, qu'un serveur. Garantissant un bon contrôle de l'information, le client / serveur s'est montré capable de répondre parfaitement aux attentes des acteurs de l'Internet. Cependant, ce modèle s'éloigne de la philosophie égalitaire qui avait donné naissance à Arpanet. Les deux paragraphes ci-dessous montrent les exemples typiques d'applications P2P de partage de ressources : Napster et LimeWire.

3.1. Napster.

Shawn Fanning, un étudiant américain passionné d'informatique alors âgé de 19 ans vient bouleverser le monde bien établi du client / serveur. Il décide de quitter l'université et se lance dans l'écriture d'un logiciel pour permettre l'échange de fichiers musicaux. La raison d'être de ce logiciel repose sur le constat suivant : rechercher des MP3 sur les moteurs de recherche habituelle conduit à une perte de temps énorme, et les réponses sont souvent inappropriées.

Après quelques mois de travail acharné, une première version du logiciel est disponible. Fanning décide de tester une première version le 1er juin 1999 et appelle son logiciel Napster (son pseudo sur Internet). Napster est un logiciel de partage de fichiers, il facilite la localisation, et l'échange de fichiers qu'ils soient de type image, audio, ou vidéos. Il fut le précurseur de tous les logiciels de partage tel, Bearshare, Morpheus, Kazaa, LimeWire. [7]

Le logiciel qui ne devait être testé que par quelques-uns de ses amis remporte un succès des plus rapides. Il conquiert notamment les universités. Shawn Fanning se retrouve propulsé à la tête d'une start-up pleine d'avenir. Lui qui déclare à propos de Napster qu'il n'avait "aucune envie d'en faire un business" voit les utilisateurs arriver en masse.

En septembre 2000, Napster atteint un nombre de téléchargement record. 1,39 milliard serait le nombre de chansons échangées par ses utilisateurs.

Cependant Napster fut victime de son succès, les étudiants l'utilisent tellement qu'ils saturent les bandes-passantes. De là tombent les premières sanctions de la part des universités, à cela s'ajoutent les plaintes des maisons de disque qui dénoncent le non respect du droit de l'auteur.

II. 2.2 Limewire.

C'est une autre application de partage de fichiers populaire, Limewire utilise un réseau P2P décentralisé, et permet le partage de toutes sortes de fichiers entre les clients. Il est doté

d'une interface graphique agréable à utiliser, et d'un système de notation à étoiles qui indique la qualité du transfert. La figure ci-dessous représente l'interface de LimeWire. [8]

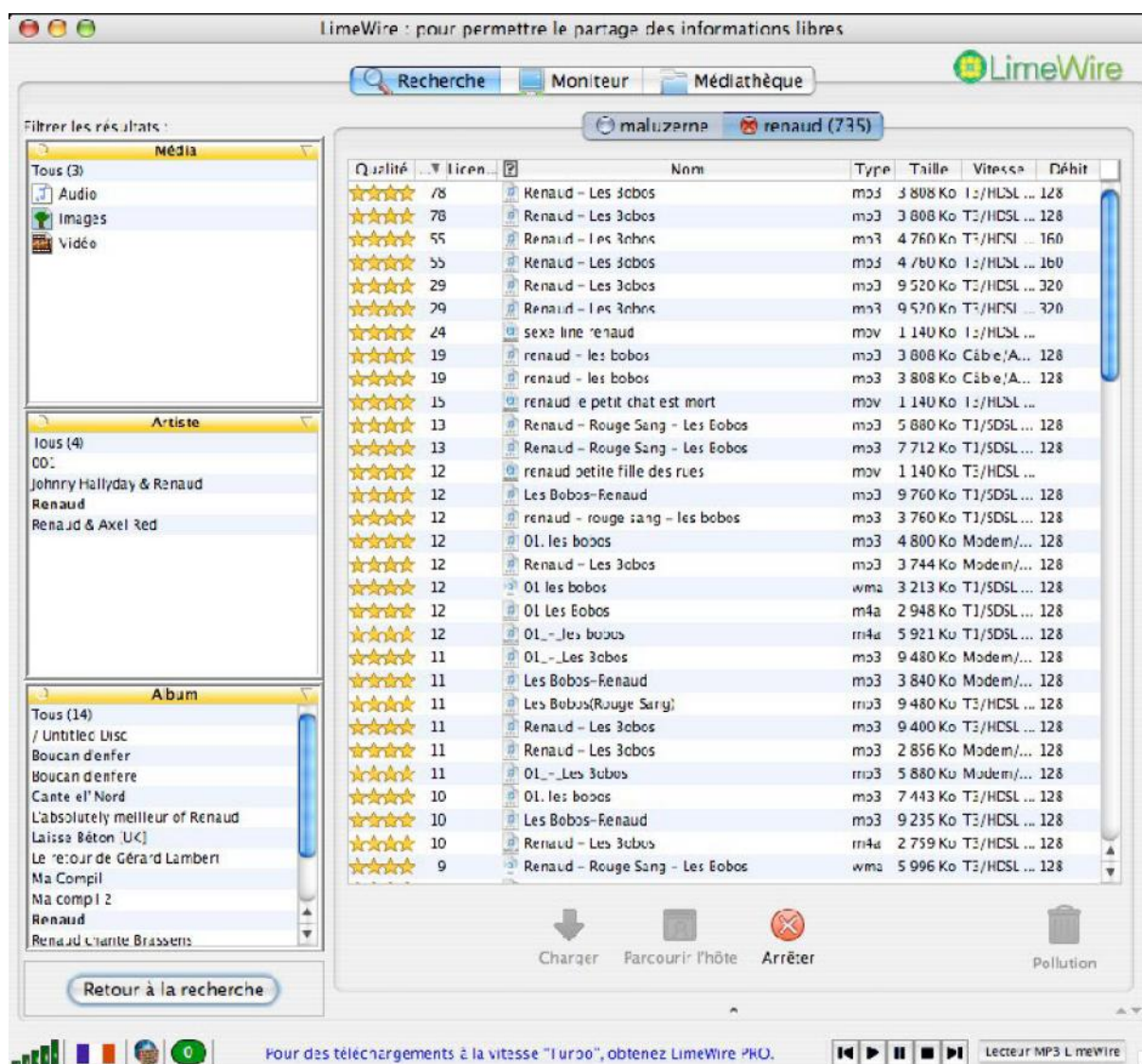


Figure 2.1 Interface LimeWire.

L'essor de l'informatique en réseau et l'explosion des utilisateurs ainsi que de leurs besoins a fait que la technologie Client/ Serveur ne suffise plus, de là est apparu le P2P. La partie qui suit va essayer de définir le P2P dans toute sa complexité à être défini.

4. Définition.

C'est quoi le Peer to Peer ? Avant de définir ce qu'est un réseau P2P, on va d'abord expliciter la notion du Peer.

Un Peer, c'est une application qui peut individuellement envoyer, recevoir, ou acheminer des messages à d'autres Peer.

Un réseau Peer to Peer Le terme p2p (Peer-to-Peer, que l'on peut traduire par « pair-à-pair » ou « égal-à égal ») tel qu'il est défini dans l'encyclopédie Wikipédia désigne « des protocoles et des réseaux informatiques dont les éléments (les nœuds) ne jouent pas exclusivement les rôles de clients ou de serveurs mais fonctionnent des deux façons, en étant à la fois clients et serveurs des autres nœuds de ces réseaux, contrairement aux systèmes de type client-serveur, au sens habituel du terme ». Les échanges peuvent se faire au niveau des données mais aussi, au niveau de la puissance de calcul et de l'espace de stockage. Il existe plusieurs types de réseaux pair-à-pair qui diffèrent selon qu'ils soient plus ou moins décentralisés. [8]

C'est une technologie qui permet à n'importe quel Peer de communiquer afin de répondre aux besoins d'autres Peer. Un Peer dans un réseau P2P agit et en client, et en serveur. Elle offre multiples avantages : tout d'abord, l'information est disponible directement depuis sa source. Le problème de stockage est résolu car les données sont sur les disques durs des utilisateurs du réseau et le traitement des données se fait localement. Ainsi à part les données du Web conservées dans un serveur, le P2P permet théoriquement d'accéder aux fichiers, et aux données de tout ordinateur connecté à internet.

L'un des enjeux majeurs dans les réseaux pair-à-pair est la recherche de ressources. En L'accès aux données par les Peer du réseau détermine les caractéristiques et la nature du réseau, et de son fonctionnement. Comme les ressources ont une connectivité instable ou des adresses IP variables, elles fonctionnent de manière autonome, indépendamment de systèmes centraux comme les DNS. Ce qui a rendu Napster et des systèmes similaires populaires, c'est le fait de tirer partie des ressources qui étaient auparavant inutilisées en tolérant une connectivité aléatoire. La possibilité à un Peer de se connecter de manière intermittente, ou une autonomie dans le partage, tels sont l'un des caractéristiques d'un réseau P2P.

5. Les caractéristiques du P2P.

L'une des principales caractéristiques des réseaux P2P, c'est d'être différent du système Client/ Serveur sur le plan architectural. L'architecture Client/ Serveur est un réseau divisé en deux, où les clients sont connectés indépendamment à un serveur, qui leur fournit individuellement des services suivant la requête de chacun. La figure 2.2 montre les caractéristiques de l'architecture traditionnelle Client/ Serveur.

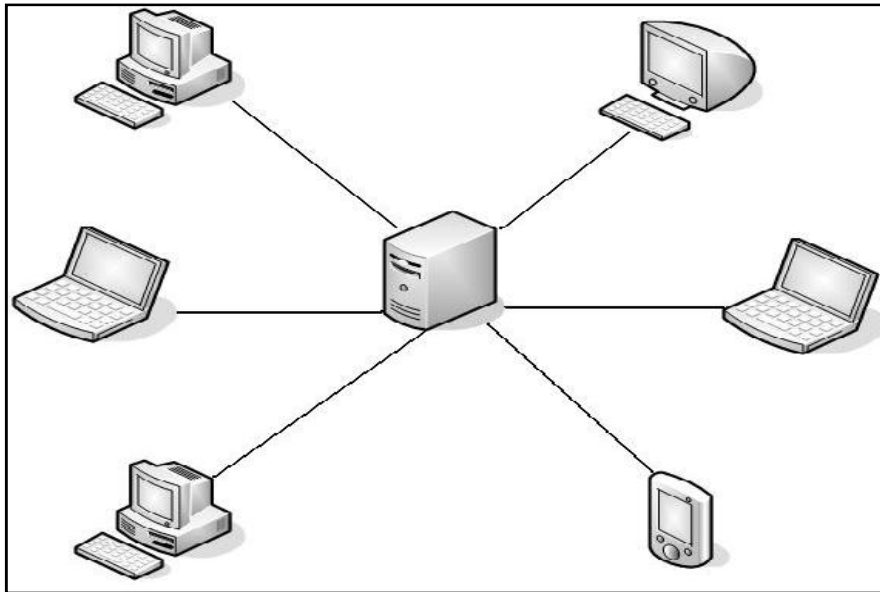


Figure 2.2 Architecture Client/ Serveur.

L'indépendance vis-à-vis d'une architecture centrale et des machines, est donc la principale caractéristique d'un réseau pair-à-pair. En plus, tout type de matériels peut être intégré au réseau (ordinateurs, PDA, téléphones...). De plus, la mise en œuvre est instantanée. Les utilisateurs peuvent se connecter directement entre eux, sans avoir besoin d'un serveur central (dans la majorité des cas pour les réseaux pair-à-pair) de manière à pouvoir échanger librement les fichiers qu'ils souhaitent partager, ou même fournir de la puissance de calcul. La figure 2.3 montre l'architecture générale d'un réseau Peer to Peer.

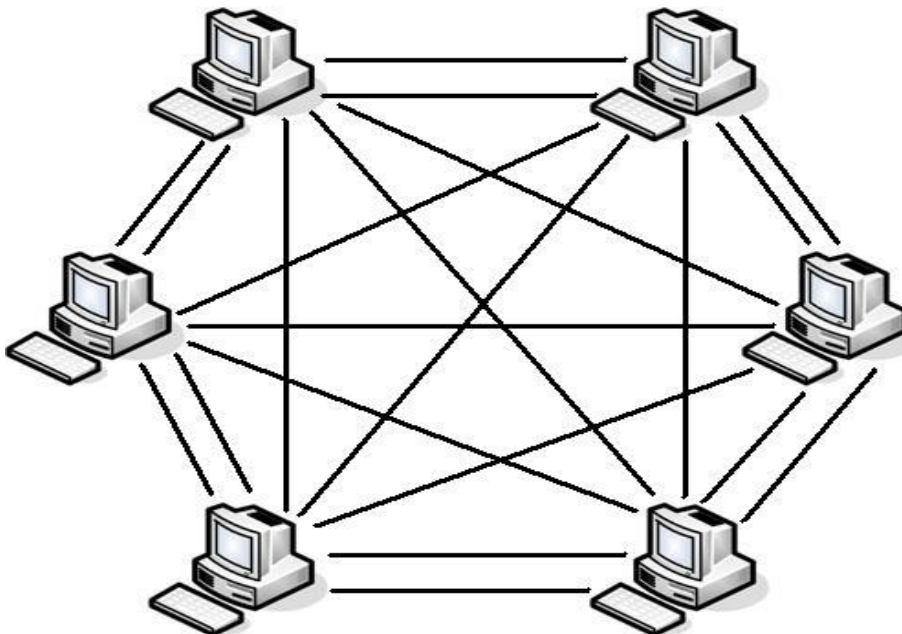


Figure 2.3 Architecture réseau Peer to Peer.

Les réseaux P2P observent quatre autres caractéristiques [4] :

- **distribution des données** : les données mises à disposition par les utilisateurs ne sont pas centralisées dans un serveur : elles sont réparties sur l'ensemble des pairs du système,
- **tolérance aux pannes** : le fait qu'un pair soit défaillant ne remet pas en cause le fonctionnement global du système,
- **passage à l'échelle** : le nombre de pairs dans le système et la quantité d'information qu'ils partagent n'empêche pas le système de fonctionner,
- **dynamisme** : les pairs sont libres de rejoindre ou de quitter le système à tout moment,
- **autonomie** : les pairs sont libres dans le choix des ressources qu'ils souhaitent partager, et de la manière dont ils veulent les partager.

La notion d'autonomie est fondamentale dans les systèmes P2P. En effet c'est l'autonomie laissée aux pairs qui leur permet de contrôler l'information qu'ils possèdent et qui leur permet de rejoindre ou de quitter librement le système. Dans la section suivante nous présentons les différentes architectures de P2P.

Ainsi ces caractéristiques permettent à répondre à deux questions qui permettent de vérifier si, effectivement, on est en présence d'un réseau P2P. Ces deux sont les suivantes :

- Est-ce que le système permet à chaque pair de se connecter de manière intermittente avec des adresses IP variables ?
- Est-ce que le système donne à chaque pair une autonomie significative ?

Si la réponse est oui à ces 2 questions, le système est P2P. Notons que le système client/serveur apporte une réponse négative aux deux questions. Les deux questions, auxquelles les réseaux P2P apportent des réponses positives, reflètent l'une des forces de ces réseaux à savoir ; l'anonymat des utilisateurs, ils n'ont pas besoin de délivrer toute leur identité, ainsi que la volatilité des pairs, chaque pair ayant le droit de rester sur un réseau ou de s'en déconnecter sans préavis. [7]

Une autre manière de distinguer un système P2P est de raisonner en termes de "propriété". Il faut remplacer la question "Est-ce que le système donne à chaque pair une autonomie significative ?" Par la question "Qui possède les ressources qui font tourner le système ?". Dans un système comme Yahoo!, l'essentiel des ressources est possédé par Yahoo! tandis que dans un système comme Napster, l'essentiel des ressources est possédé par les utilisateurs de Napster.

Dans les vrais systèmes P2P, on ne se contente pas de décentraliser les fonctions mais aussi les coûts et les charges d'administration. Avec un calcul conservateur, 100 millions de PC connectés à Internet avec une puissance de 2 GHz et 500 Mo de disques représentent une puissance de deux cents milliards de Mhz et cinquante mille téraoctets de stockage. C'est de ces ressources dont il faut tirer parti.

Une caractéristique des réseaux P2P est que la qualité et la quantité des données disponibles augmentent à mesure que le nombre d'utilisateurs augmente. La valeur du réseau augmente donc avec sa popularité.

Enfin, la pertinence d'un système P2P réside dans sa capacité à localiser les ressources efficacement quelle que soit la taille du réseau. Ainsi, ces systèmes doivent reposer sur des méthodes efficaces de découverte des ressources désirées.

Le système P2P tire sa puissance dans sa capacité à tirer profit de toutes les ressources présentes dans toutes les machines participatives dans le réseau, et l'avantage est de plus grand que l'efficacité du système augmente avec un nombre élevé de participant dans le réseau. Il existe plusieurs architectures qui permettent d'établir cette connexion décentralisée des machines pour le partage des ressources. Dans le paragraphe qui suit on va détailler le fonctionnement d'un système P2P dans toutes ses architectures qui existent

6. L'architecture des réseaux P2P et leurs Fonctionnements.

6.1. L'architecture centralisée.

6.1.1. Architecture centralisée.

Le réseau P2P le plus connu du grand public est Napster sans doute, et il utilisait une architecture centralisée.

Dans cette architecture, un client (un logiciel utilisé par les membres) se connecte à un ou plusieurs serveurs qui gèrent les partages, la recherche, l'insertion d'informations, bien que celles-ci transitent directement d'un utilisateur à l'autre. Toutes ces informations sont collectées à l'aide d'un système d'indexation appelé la « table de hachage distribuée » qui permet en théorie d'éviter la multiplication de fichiers inutiles. En effet, les fichiers peuvent être identiques sans pour autant posséder le même nom et inversement. On associe donc un identifiant unique (« hash ») à chaque pair, chaque mot clé et chaque fichier en fonction de son contenu et non de son nom. Ce qui permet par exemple de récupérer le bon fichier même si le nom est incorrect. [8]

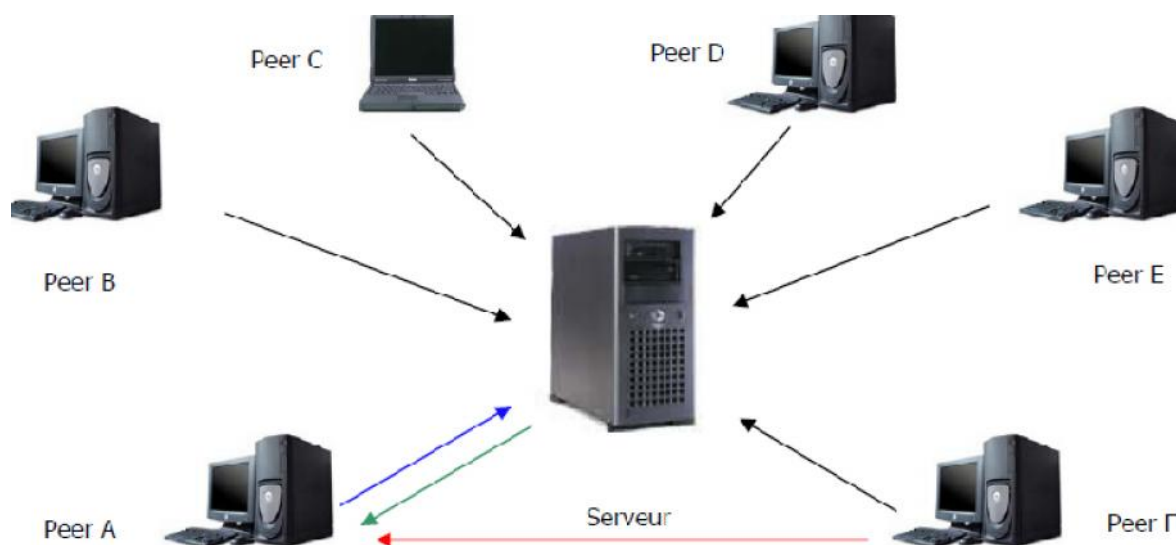


Figure 2.4 Peer to Peer centralisé

6.1.2. Fonctionnement de l'architecture centralisée.

Dans cette architecture centralisée un serveur se charge d'établir une relation directe entre tous les utilisateurs connectés. L'intérêt de cette technique c'est de pouvoir indexer au niveau du serveur tous les répertoires et de pouvoir nommer comme partagés tous les fichiers mis en ligne par les utilisateurs. La mise à jour des fichiers s'opère en temps réel, dès qu'un utilisateur se connecte ou quitte le réseau.

Tout comme dans un moteur de recherche classique, le client lance une requête en inscrivant le mot-clé. Il obtient la liste des utilisateurs connectés au service correspondant au terme de sa recherche, et dont les fichiers correspondent au terme recherché. Dès lors il suffit de cliquer sur un des liens pour entrer en connexion directe sur la machine correspondante.

6.1.3. Avantages et inconvénients de l'architecture centralisée.

Ce type d'architecture centralisée permet d'avoir une vue globale et très complète du réseau, à condition que le nombre d'utilisateurs soit limité pour ne pas se retrouver submergé par une liste interminable d'informations, elle se montre efficace et rapide dans le cas d'une recherche ciblée. En recevant la requête le serveur vous achemine vers le ou les utilisateurs susceptibles d'héberger le fichier recherché avant de vous connecter directement au PC concerné.

Cependant malheureusement le réseau centralisé est très vulnérable par ce qu'il suffit d'arrêter le serveur pour arrêter tout le réseau. Aussi le fait de passer par une architecture centralisée, où il faut s'enregistrer ne garantit pas l'anonymat. Le service connaît l'adresse IP de l'utilisateur, et pourrait établir un profil de l'utilisateur suivant ses recherches sur le réseau.

Mais pour palier à cette vulnérabilité du réseau centralisé, pour le rendre plus robuste et améliorer la qualité de connexion avec le serveur, un autre type de réseau centralisé a été mis au point. On ne travaille plus avec un seul serveur central. Le serveur central est remplacé par

un anneau de serveur. Ceci permet d'éviter la chute de réseau si un serveur tombe en panne ; mais aussi cela permet d'augmenter les performances du réseau en rapidité, parce que qui dit plusieurs serveurs, dit meilleure répartition des données et aussi cela permet de désengorger la bande passante.

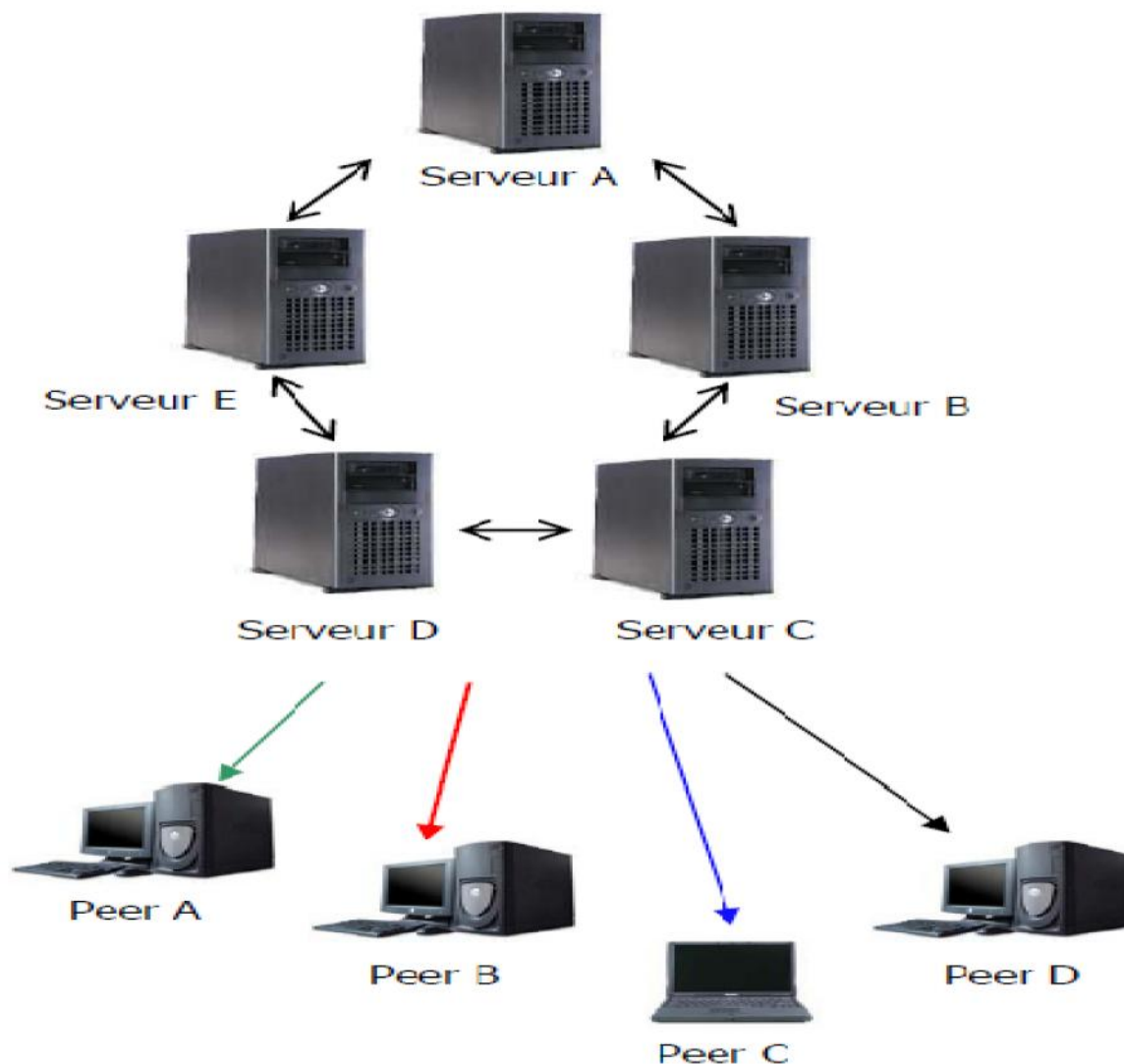


Figure 2.5 Peer to Peer Centralisé amélioré.

Même si l'architecture centralisée permet la connexion directe entre les utilisateurs, il reste vulnérable, et ne diffère en grand-chose avec le système Client/ Serveur. On a cherché à décentraliser le réseau, pour arriver à une réelle relation de pair à pair. L'architecture distribuée ou décentralisée.

6.2. L'architecture décentralisée ou distribuée.

6.2.1. Architecture décentralisée ou distribuée.

Les systèmes décentralisés ou distribués sont conçus de telle sorte qu'aucune donnée ne soit centralisée. Chaque pair stocke ses propres données, ainsi que les index

correspondants. La figure 2.6 présente un système P2P distribué dans lequel huit pairs sont interconnectés. Pour maintenir la connexion avec d'autres pairs, chaque pair maintient une vue locale du système, contenant les identifiants de certains autres pairs. L'identifiant d'un pair permet de le contacter, et peut être simplement défini à partir de son adresse IP et d'un numéro de port. Dans l'exemple de la figure 2.2, les connexions sont bidirectionnelles (ce qui le rend beaucoup moins vulnérable). Le réseau distribué pour sa part n'utilise aucun serveur, chaque utilisateur est à la fois client et serveur, et tous se découvrent et se connectent dynamiquement entre eux en relayant les requêtes d'ordinateur à ordinateur. Si cette architecture a l'avantage d'être robuste, elle exige en revanche un trafic important car la recherche d'un fichier prendra plus de temps. Chaque requête est adressée à chacun des utilisateurs connectés. De leur côté faisant de même, ce qui peut engendrer un long moment pour avoir une réponse à une requête si des milliers d'utilisateurs sont connectés.

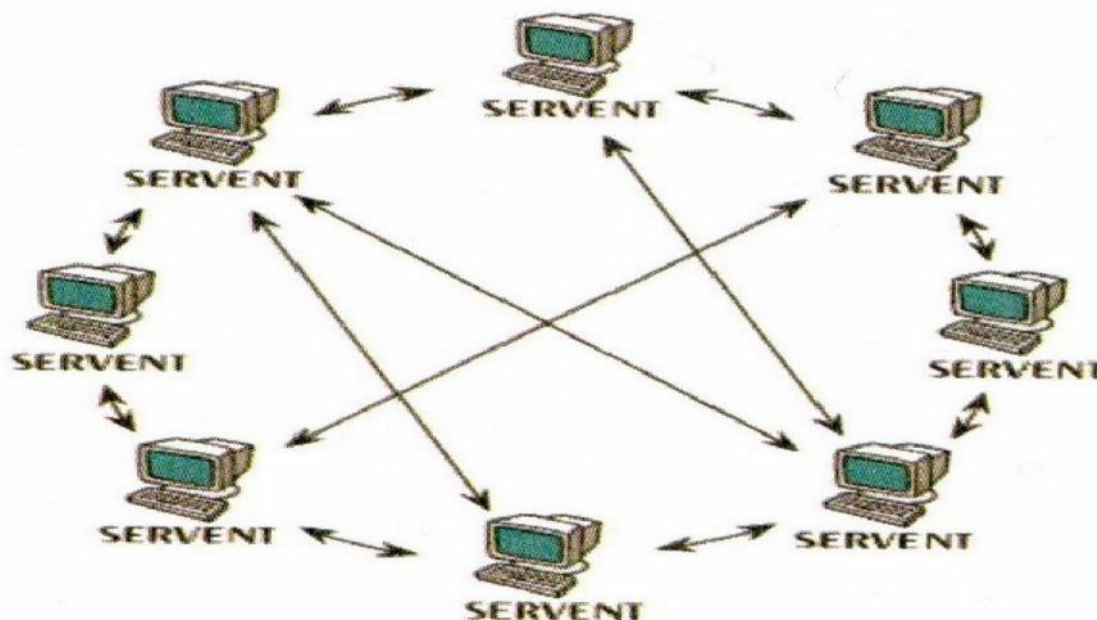


Figure 2.6 Peer to Peer Décentralisé.

6.2.2. Fonctionnement de l'architecture décentralisée.

Plusieurs méthodes existent pour palier à ce problème d'augmentation exponentielle du trafic lors de la recherche des ressources. L'Horizon est la limitation du nombre d'utilisateurs (pairs) visibles par un autre pair. Ainsi les requêtes sont d'abord envoyées au pair de l'Horizon, ce n'est seulement qu'en cas d'échecs que la requête est transmise à d'autres pairs.

Le « time out » est un délai limite avant abandon d'une requête en cas d'échec d'une réponse. Il est utilisé pour réduire sensiblement le trafic et pour éviter tout goulet d'étranglement.

L'absence du serveur oblige chaque pair à jouer le rôle du serveur et donc une partie de hachage. Le fait est que dans le cas de disparition de plusieurs pairs une partie de l'annuaire viendrait à disparaître, avec les renseignements sur les pairs et toutes les ressources partagées.

Pour éviter une telle situation plusieurs pairs sont simultanément chargés de la responsabilité de la table de hachage.

Pour tirer le maximum des architectures centralisée et décentralisée, ce qui permet de diminuer le nombre de connexions sur un seul serveur, et ainsi améliorer les performances du réseau on a regroupé les deux architectures pour former une architecture hybride. [8]

6.2.3. Avantages et inconvénients de l'architecture décentralisée.

Le principal désavantage des systèmes non structurés est que la recherche de données est complexe. Pour trouver et accéder à une donnée (ou à un ensemble de données), un pair doit émettre une requête et la transmettre aux autres pairs. Cela a comme conséquence de générer un nombre de messages important et donc d'encombrer le réseau.

Les avantages de cette architecture sont nombreux. Tout d'abord elle est tolérante aux pannes et n'est pas sensible aux attaques. En effet la défaillance d'un pair entraîne la perte des données qu'il stockait mais ne remet pas en cause le fonctionnement du système dans sa globalité. Ensuite, le fait que les pairs stockent leurs index permet de garantir leur autonomie, en particulier dans la manière de construire l'index. Enfin la décentralisation totale des données permet de passer à l'échelle. Il n'y a pas de risque de surcharge dans la mesure où les pairs sont "égaux".

6.3. L'architecture hybride.

6.3.1. Architecture hybride.

L'architecture hybride ou le réseau en super nœud est complexe à mettre en œuvre, elle combine à la fois l'architecture centralisée et décentralisée. Il s'appuie sur un ensemble de serveurs (l'ordinateur d'un utilisateur peut devenir un serveur) gérant un groupe d'utilisateurs suivant l'architecture centralisée. Puis, chaque serveur est ensuite connecté à d'autres serveurs suivant l'architecture distribuée. De cette façon, si un fichier recherché par un utilisateur n'est pas indexé par le serveur auquel il est rattaché, celui-ci transmet alors la requête à un autre serveur. Le modèle hybride permet de bénéficier d'une meilleure bande passante en réduisant le trafic de requêtes.

D'autre part le réseau de serveurs utilise un mécanisme issu des réseaux décentralisés pour tenir à jour un annuaire client et un index des fichiers à partir des informations provenant des autres serveurs. Un serveur peut donc proposer à n'importe quel client toutes les informations contenues sur le réseau. Le réseau n'est plus pollué par les trames de diffusions. Mais la contrepartie est que l'anonymat n'est plus garanti.

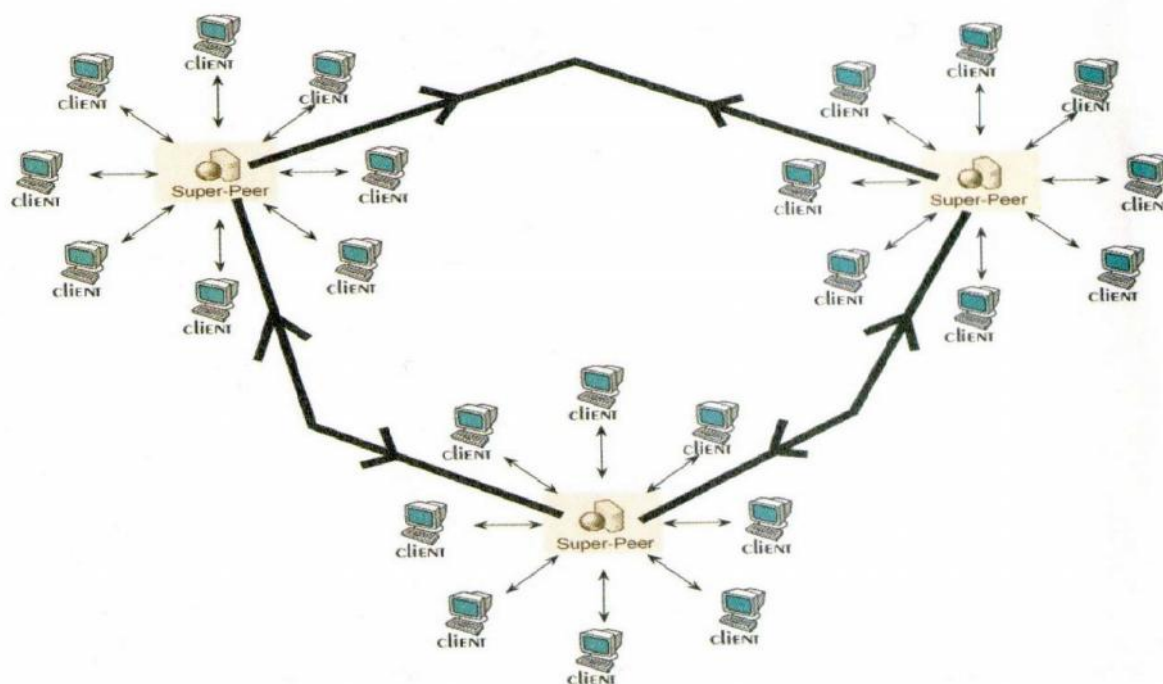


Figure 2.7 Peer to Peer Hybride.

6.3.2. Fonctionnement de l'architecture hybride.

En combinant à la fois les réseaux distribués et centralisés, le réseau hybride utilise les internautes de deux manières différentes. Outre les utilisateurs lambda, ceux qui possèdent une bande passante plus importante sont utilisés comme des serveurs centraux qualifiés de « super-Peer ». Ils sont reliés entre eux de manière décentralisée.

Gnutella est le protocole le plus connu de cette architecture, il est basé sur le principe d'une architecture de réseau décentralisée. Comme nous l'avons vu dans la description de l'architecture décentralisée, cette dernière permet d'interroger tous les pairs pour lancer une recherche, et tous les pairs procèdent de la même façon. La notion de Time out a été introduite pour éviter l'engorgement inutile du réseau. Gnutella introduit dans cette architecture hybride, les « ultraPeers » (ou supernœuds). Ça consiste à cibler les machines les plus performantes du réseau, ce sont ces dernières qui deviennent les « supernœuds », disposant d'une bonne connexion internet. Elles sont chargées de transmettre les requêtes aux autres « ultraPeers » et pairs. Cela permet d'effectuer une recherche sur plusieurs pairs à l'aide d'une seule requête, les machines les moins performantes se contentent de répondre seulement aux requêtes. Limewire est l'exemple d'un logiciel utilisant ce protocole.

6.3.3. Avantages et inconvénients :

Les avantages du modèle hybride, c'est qu'il a pour but d'utiliser les avantages des deux autres types (centralisé et décentralisé). En effet sa structure permet de diminuer le nombre de connexions sur chaque serveur, et ainsi d'éviter les problèmes de bandes passantes.

D'autre part le réseau de serveurs utilise un mécanisme issu des réseaux décentralisés pour tenir à jour un annuaire client et un index des fichiers à partir des informations provenant des autres serveurs. Un serveur peut donc proposer à n'importe quel client toutes les informations contenues sur le réseau. Le réseau n'est plus pollué par les trames de diffusion.

Mais l'inconvénient de ce modèle est que l'anonymat n'est plus assuré. Il existe tout de même des risques de passer à côté d'une réponse contrairement aux techniques d'inondations qui vont effectuer leurs recherches dans tout le réseau.

7. Conclusion.

Dès son apparition, le Peer to Peer a été adopté par les internautes. Les réseaux Peer to Peer sont un phénomène important avec plusieurs facettes, structurés sous plusieurs architectures. La première génération a été centralisée avec Napster. Ensuite d'autres modèles totalement décentralisés, ont été proposés.

Bien que cette technologie traîne la réputation de plateforme de piratage de fichiers, Nous avons délibérément occulté cet aspect du Peer to Peer pour nous concentrer uniquement sur les aspects techniques et pratiques. Le Peer to Peer comme on va le voir dans le chapitre qui suit, chapitre sur le JXTA, est un concept prometteur. Les domaines d'application sont nombreux (partage des connaissances, travail collaboratif), car il permet de mutualiser les ressources et de supporter un grand nombre d'utilisateurs. Il constitue un atout raisonnable pour notre projet.

Chapitre III : JXTA

1. Introduction.

Concevoir un réseau où toutes les machines peuvent garantir le même service, un réseau configuré de telle sorte que la présence d'un serveur ne s'avère indispensable, un réseau où les pairs agissent en tant que client et en même temps que serveur. Le modèle P2P répond à tous ces besoins, il permet un partage extraordinaire de données, une présence de ressources qui croît avec le nombre d'utilisateurs, sans contrainte les utilisateurs peuvent échanger les ressources directement entre eux. C'est cette possibilité d'échanger sans passer par obligatoirement par l'intermédiaire d'un serveur et la disponibilité énorme des données, qui a séduit les utilisateurs du modèle P2P. Fort de leurs succès les applications P2P sont en grande expansion, mais ces dernières n'arrivent pas à communiquer entre elles,

Différentes protocoles, différentes architectures, différentes implémentations. Cela décrit parfaitement de nombreuses applications P2P. Les développeurs utilisaient diverses méthodologies et approches pour créer des applications P2P. Mais contrairement aux standards qui uniformisent l'architecture client/ serveur, étaient absent chez les développeurs d'applications P2P. C'est dans ce contexte, en vue de combler ce déficit, que Sun a développé JXTA.

Dans ce chapitre on va parler de JXTA, voir les ressources nécessaires qu'il faut pour pouvoir implémenter, gérer, et faire fonctionner la plateforme JXTA. Voir les protocoles qui permettent de créer cette interopérabilité des applications P2P, afin de leurs permettre de communiquer entre elles sans tenir compte du langage ou de l'environnement de programmation.

2. Historique.

JXTA est un projet Open Source présenté pour la première fois en février 2001, et lancé par Sun Microsystems en avril 2001. Le but de JXTA est de pouvoir interconnecter n'importe quel système sur n'importe quel réseau. Le P2P permet de créer une sorte de réseau au-dessus des autres. JXTA c'est plus de 17000 développeurs en 2004, avec plus d'une centaine de projets qui ont adopté cette nouvelle technologie. Sun cherche à l'imposer comme standard du P2P. Le projet JXTA fut le premier à généraliser les fonctionnalités P2P, et à concevoir la technologie principale qui permet aujourd'hui de dépasser les limites établies des applications P2P. Il se concentre sur l'établissement des mécanismes de base et laisse aux développeurs le choix de la méthode de développement. [10]

3. Définition.

JXTA est un middleware pour des applications en Peer-to-Peer développé au sein du modèle Apache par Sun Microsystems. C'est une suite de six protocoles Open Source

développée par Sun, qui permettent le partage, la communication et la collaboration entre les peers. L'origine du nom JXTA provient du verbe « juxtaposer », c'est à dire le fait de mettre deux entités côte à côte. Il offre la possibilité à chaque composant d'un réseau de communiquer, de collaborer et de partager des ressources. Les peers JXTA créent un réseau virtuel au dessus du réseau physique, cachant ainsi la complexité de celui-ci. Dans un réseau virtuel JXTA, chaque peer peut interagir avec tous autres, sans se soucier de son emplacement, du type de composant ou de l'environnement d'exécution et même si il est situé derrière un firewall ou qu'il utilise une autre couche de transport réseau. La technologie JXTA fonctionne sur n'importe quel composant, qu'il s'agisse d'un PC ou d'un PDA. [9]

L'idée de juxtaposition vient du fait que l'équipe de développement croit que les applications Peer-to-Peer vont coexister avec leurs ancêtres : les applications client-serveur. Les systèmes P2P ne vont pas remplacer les systèmes client-serveur, selon le type d'application et le problème posé, les développeurs choisiront l'une ou l'autre des architectures. [11]

La figure ci-dessous donner un aperçu d'une représentation d'un réseau P2P avec JXTA. Avec une indépendance à n'importe quel langage de programmation, de tout plateforme réseau et tout système d'exploitation, JXTA est basé sur des standards tels que TCP/IP, HTTP et XML. JXTA permet d'établir des réseaux P2P, avec des mécanismes simples, petits et flexibles.

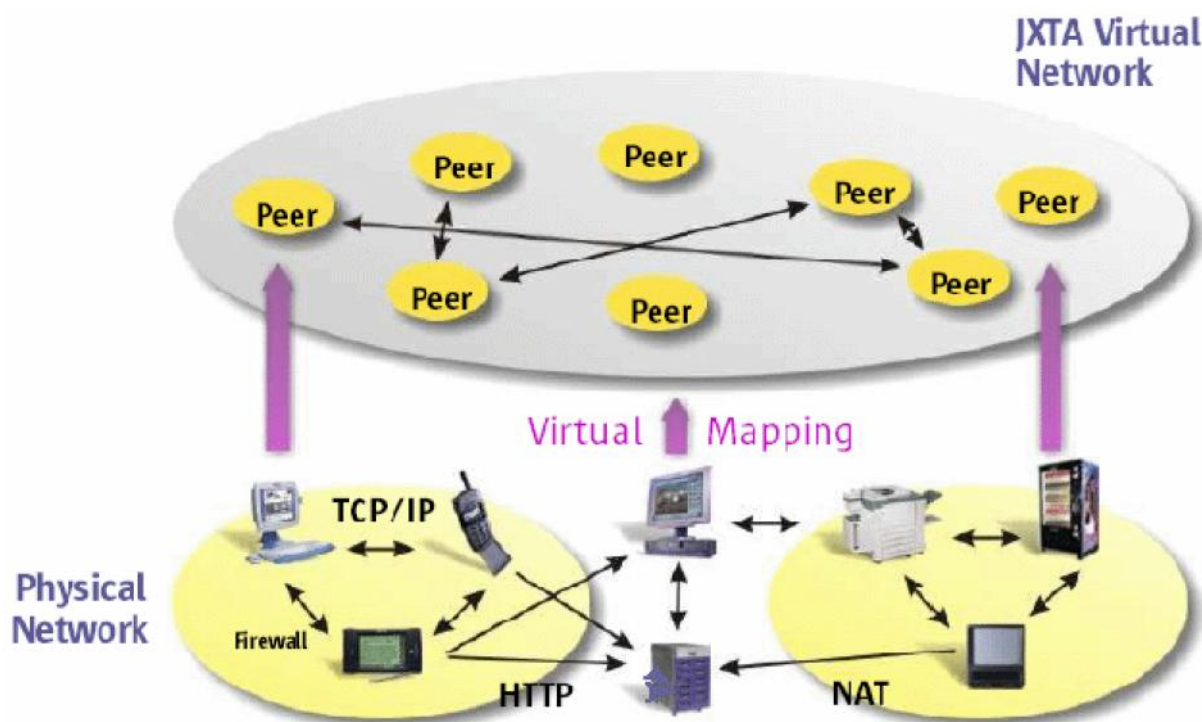


Figure 3.1 Réseau virtuel JXTA.

4. Pourquoi JXTA ?

JXTA se veut être une plateforme offrant des fonctionnalités assez standard pour permettre aux développeurs, d'horizons distincts, de pouvoir créer des applications peer-to-peer interopérables. JXTA a pour objectif d'offrir une infrastructure de base sur laquelle d'autres applications P2P peuvent être implémentées. Pour le cas de notre projet il s'agit d'un prototype d'une application partage de ressources pédagogiques en utilisant les ressources pédagogiques. Et pour atteindre ces objectifs JXTA s'est doté de six protocoles. Les objectifs de JXTA peuvent se résumer en quatre points suivants : [10]:

- **Interopérabilité** : entre les différents systèmes P2P, entre les applications de différents types, entre le P2P et le web service.
- **Indépendance** : vis-à-vis des applications, des langages (Perl, Python, Ruby,...), des systèmes d'exploitation et des protocoles réseaux (Wifi, TCP, http).
- **Ubiquité** : peut aller sur n'importe quel type d'appareil avec un processeur.
- **Sécurité** : la sécurité est prise en compte dès le noyau de JXTA. L'objectif clair de JXTA est de s'intégrer dans les gouvernements, Entreprises, Universités.

Ces objectifs ont été décortiqués en six protocoles, qui forment les services de base de la plateforme JXTA. L'implémentation de ses protocoles se fait dans un niveau plus bas et peut être réalisé en n'importe quel langage, respectant ainsi l'indépendance de la plateforme vis à vis des systèmes et des langages de programmation. Dans la suite des paragraphes, nous définissons les concepts clés de JXTA. Nous présentons ensuite, le modèle conceptuel et les principaux protocoles qui le forment, ainsi que l'architecture générale de la plateforme.

5. Concepts de JXTA.

On a vu dans le paragraphe précédent que l'objectif de JXTA était d'offrir une plateforme pour la création des applications Peer-to-Peer. En mettant en lumière les concepts du JXTA, on a choisi tout d'abord à définir en premier lieu la notion de « peer » ou « pair » dans JXTA. Par la suite on va donner une définition encore selon JXTA d'un « peerGroup », « Pipe », « Message » et « Annonce ». Le paragraphe se termine avec les six protocoles de JXTA qui permettent l'implémentation d'une application P2P. [12]

5.1. Peer.

Un pair est un nœud dans le réseau de JXTA. Il constitue toute entité matérielle ou logicielle implémentant les protocoles nécessaires de JXTA permettant son adressage et identification sur le réseau. Il est essentiel de rappeler la nécessité de faire la distinction entre nœud et ordinateur, car un pair peut être aussi un simple téléphone ou un PDA contenant un programme implémentant les protocoles de base de JXTA. Les peers sont automatiquement configurés pour découvrir d'autres nœuds sur le réseau et pour former des relations appelées peer groups. Les peers n'ont aucune obligation quant à la durée de leur connexion sur le

réseau. Un peer ne peut donc pas être garanti que le peer lui offrant un service restera online jusqu'à la fin de celui-ci.

Nous distinguons trois types de pairs dans la plateforme de JXTA : les pairs simples, les pairs rendez-vous et les pairs routeurs.

5.1.1. Le pair simple.

C'est un pair conçu pour servir un seul utilisateur, lui permettant d'utiliser des services offerts par d'autres pairs ou de proposer et publier ses propres services. Dans les environnements actuels et par soucis de sécurité, ces pairs sont confrontés à plusieurs types de protections qui constituent des contraintes pour le pair de deux points de vue.

Si le pair simple se trouve isolé de l'extérieur par un Firewall, par exemple, il ne sera pas visible de l'extérieur. Cela pose un problème du fait que la diffusion et l'utilisation des services qu'il offre au monde extérieur, ne seront pas garanties. Le problème de visibilité et accessibilité des pairs, que ce soit de ou vers l'extérieur, est un défi majeur pour les applications ouvertes, comme le cas de notre projet qui est basé sur le partage dans l'architecture P2P.

Pour résoudre ce problème d'accessibilité aux services offerts par les pairs dans un environnement comme le P2P, JXTA propose deux autres types de pairs : le pair rendez-vous, et le pair routeur.

5.1.2. Le pair rendez-vous.

Ce pair se distingue des pairs simples car il offre un service particulier. Le pair rendez-vous permet la découverte des pairs se trouvant sur sa responsabilité et les ressources qu'ils offrent. Il reçoit des requêtes provenant de certains pairs, et il se charge de renvoyer des réponses en fonction des informations qu'il possède sur les pairs qu'il connaît. Les différents pairs rendez-vous d'un même réseau peuvent collaborer en s'échangeant des informations, et en diffusant les requêtes auxquels ils ne peuvent pas répondre. Deux cas de figures existent pour la localisation d'un peer rendez-vous par rapport à un réseau local qu'il représente. Primo, le cas classique, c'est que le pair se trouve à l'extérieur du réseau, donc visible pas les autres pairs. Secundo, le pair se trouve à l'intérieur du réseau et dans ce cas il doit, soit avoir les privilèges nécessaires pour qu'il puisse outrepasser les protections du réseau, soit utiliser un pair spécial : le pair routeur. La figure ci-dessous représente un pair rendez-vous.

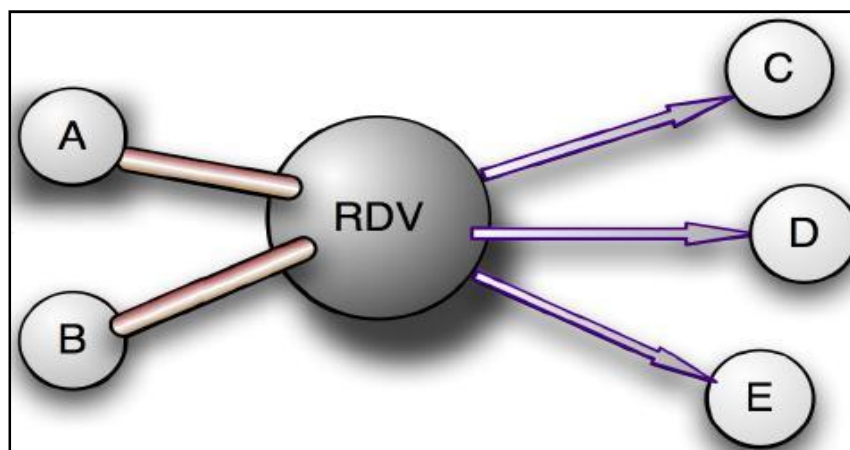


Figure 3.2 Pair Rendez-vous

5.1.3. Pair routeur.

Un pair de type routeur permet à des pairs, séparés du réseau externe par mécanismes de protection, un Firewall par exemple, de communiquer avec les autres pairs. Un pair qui veut communiquer avec un autre pair isolé doit, en premier lieu, déterminer quel pair routeur il doit utiliser. Ensuite, c'est au routeur de trouver l'adresse IP dynamique relative au pair destinataire. Notons enfin qu'un simple pair peut jouer le rôle de rendez-vous ou de routeur ou les deux simultanément. Il suffit d'implémenter les services en question pour devenir un pair rendez-vous ou un routeur.

5.2. PeerGroup.

Un peerGroup est formé d'une collection de pairs qui se sont mis d'accord pour le partage d'un certain nombre de services (échange de fichiers, résolution de problèmes...). A chaque groupe est attribué un identifiant unique. Former un peerGroup nécessite deux étapes au minimum. La première est la création d'un processus d'inscription pour les pairs qui seront membres du groupe. Un mécanisme de protection de type (nom, mot de passe) peut être utilisé pour réaliser cette étape, sinon le groupe est ouvert à tout membre désirant le rejoindre. Si un pair crée un groupe et lui rattache des mécanismes de sécurité restreignant son accès, le groupe est alors considéré comme privé. La deuxième condition nécessaire est que tous les pairs puissent communiquer selon le même type de messages et se basent sur les mêmes protocoles de communication.

Un pair peut appartenir à plusieurs groupes en même temps, et tous les pairs dès leurs créations appartiennent automatiquement à un groupe par défaut dit « World Peer Group ». Ce groupe est l'ancêtre de tous les autres groupes qui peuvent être créés. Un deuxième groupe par défaut existe en JXTA il s'agit du « Net Peer Group ». Ces deux groupes de base doivent offrir des services essentiels pour le fonctionnement des autres pairs sur le réseau.

Chaque peer group possède un certains nombres de services de base qui ont été définis par une spécification. Il s'agit de :

- **Discovery Service** : permet de rechercher du contenu dans un groupe.

- **Membership service** : permet la création d'un groupe sécurisé.
- **Access Service** : permet de valider l'accès d'un peer à un groupe.
- **Pipe Service** : permet de créer et utiliser des tubes de communications.
- **Resolver Service** : permet les requêtes et les réponses pour les services de peers.
- **Monitoring Service** : permet aux peers de surveiller d'autres peers et groupes

L'importance des groupes de pairs, c'est qu'ils constituent un moyen de rassembler des pairs partageant un intérêt commun. Une meilleure organisation est d'une part établie, et d'autre part cette association facilite la communication et la découverte. Les politiques d'accès réduisent les risques, en matière de sécurité, pour les pairs d'un même groupe.

5.3. Pipe.

Dans JXTA les nœuds communiquent entre eux par des pipes. Ce concept de tubes de communication découle des systèmes UNIX. L'information entre dans une des extrémités du tube et sort de l'autre côté. Grâce aux pipes, des messages peuvent être envoyés entre les différents peers sans se soucier de l'infrastructure sous-jacente. Les peers ne doivent pas s'inquiéter à propos de la topologie du réseau ou de la localisation d'un autre peer pour pouvoir communiquer. Les pipes utilisent la notion d'endpoint pour désigner les points d'entrée et de sortie d'un canal de communication.

La plateforme JXTA propose deux types de « Pipes » : « Unicast Pipe » et « Broadcast Pipe » :

- **Unicast Pipe** : unidirectionnel, non sécurisé et non fiable. Les pipes unicast connectent un peer à un autre pour une communication dans un sens unique.
- **Propagating Pipe** : tube de propagation, non sécurisé et non fiable. Les pipes unicast Le pipe de propagation un tube de sortie à plusieurs entrées de tubes.

La figure ci-dessous montre un exemple des deux types de pipe.

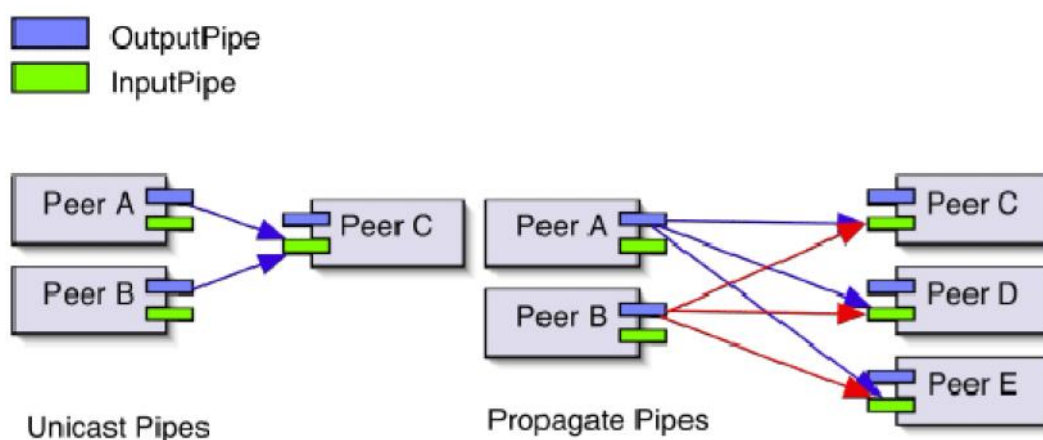


Figure 1.3 Unicast & Propagates Pipes.

Le paragraphe suivant explique ce qu'est un message pour JXTA, et donne les formes possibles que peut avoir un message.

5.4. Message.

JXTA permet l'envoi des messages entre les pairs à travers les pipes. Les messages sont envoyés sous deux formats **XML** ou **Binaire**. Les messages échangés entre « Pairs » doivent être sous un format bien précis, pour permettre aux destinataires de trouver facilement l'information recherchée, contenue dans le message. Contrairement à JXTA tous les autres systèmes P2P utilisent des formats de message propre à eux (Napster, Limewire,...), et c'est pour cette raison que les « Peer » de ces applications n'arrivent pas à communiquer entre eux. L'utilisation de la représentation XML offre plus de flexibilité, et ce dernier est facilement adaptable en fonction de l'environnement. JXTA propose un format de messages précis que toutes les applications basées sur sa plateforme doivent utiliser. Ainsi les développeurs sans avoir à définir des mécanismes des pairs de deux applications différentes peuvent facilement communiquer.

JXTA offre comme service la découverte des ressources. Chaque pair dans le système peut avoir des informations concernant les autres pairs, les peerGroups, et ainsi que tous les services qu'ils proposent. Ce mécanisme est assuré le concept d'**Advertissement**, ou publicité.

5.5. Les advertisements.

Un « advertisement » est un document XML qui décrit un message JXTA, un pair, un peerGroup, ou un service. Tous les protocoles de la plateforme JXTA utilisent les « advertisement » pour s'échanger des informations. Les « advertisements » sont utilisés pour l'échange des informations sur les ressources disponibles sur le réseau JXTA.

Les pairs utilisent le pair Rendez-vous pour découvrir les advertisements provenant des autres pairs du réseau. Le pair Rendez-vous conserve les advertisements et se charge de la recherche.

5.6. Les protocoles de JXTA.

Les protocoles de JXTA sont utilisés par les pairs pour interagir, et pour communiquer entre eux. Les pairs utilisent les protocoles pour publier et rechercher leurs ressources à travers le réseau. Les protocoles sont indépendants entre eux, mais malgré cette indépendance, ils travaillent ensemble pour faciliter la découverte la découverte, l'organisation, le monitoring, et la communication entre pairs. Les protocoles de JXTA constituent un concept de base pour la conception des applications P2P.

Nous présentons dans la suite brièvement les six protocoles :

Peer Discovery Protocol (PDP).

Le Peer Discovery Protocol (PDP) permet au pair de découvrir d'autres pairs. Ils peuvent être des pairs Advertisements, les peerGroups, des services ou des pipes. Ce protocole permet aussi à un pair de se faire découvrir. En effet, chaque pair l'utilise pour se présenter et ainsi faciliter aux autres de connaître sa nature et les services qu'il propose. Le PDP est un mécanisme de recherche utilisé par les pairs pour localiser les informations dans le réseau.

Peer Information Protocol (PIP).

Le Peer Information Protocol (PIP) permet à un pair d'avoir des informations sur l'état des autres pairs. Un pair peut savoir si un autre pair est toujours vivant ou non et recevoir plus d'informations sur lui. En effet le PIP est utilisé par les pairs pour envoyer des pings et cela leur permet effectivement d'avoir des informations sur l'état des autres pairs.

Peer Resolver Protocol (PRP).

Le Peer Resolver Protocol (PRP) permet à un pair d'envoyer des requêtes à n'importe quel nombre d'autres pairs et d'en recevoir autant de réponses que de requêtes remises. Les requêtes d'un pair à un autre sont envoyées dans un message sous format XML, pour pouvoir être comprises. Les protocoles PDP et PIP utilisent le PRP grâce à sa capacité à générer des messages. Le premier (le PIP) l'utilise pour formuler des requêtes questionnant sur l'état d'un pair distant, alors que le second (le PDP) est utilisé pour découvrir les ressources offertes par d'autres pairs.

Pipe Binding Protocol (PBP).

Le PBP permet de créer un chemin de communication entre les différents pairs. Ce protocole permet à un pair, via un canal de communication virtuel (le Pipe), de communiquer avec un ou plusieurs autres pairs du réseau. Un Peer peut ainsi créer un nouveau pipe, se connecter à une pipe existant, ou ne pas se connecter du tout. Le PBP est utilisé pour gérer tous ces cas.

End Point Routing Protocol (EPRP).

L'End Point Routing Protocol est utilisé pour trouver une route entre deux pairs. Le protocole propose un ensemble de fonctionnalités pour permettre le routage des messages d'un pair source vers un pair destination. Il montre aux requêtes des Peer le chemin à emprunter pour envoyer des messages. Il aide un message d'un pair d'arriver à destination en indiquant le chemin direct ou indirect aux messages d'un pair à un autre. Si la topologie du réseau change ou cesse de fonctionner, l'EPRP est utilisé pour déterminer un éventuel chemin à emprunter.

Rendez-vous Protocol (RVP).

Le RVP est utilisé pour propager les messages dans le réseau. Un pair peut jouer le rôle de Rendez-vous et offrir ainsi le mécanisme de propagation à un groupe de pairs qui doivent se connecter à lui pour pouvoir propager et recevoir des messages.

Théoriquement un pair n'est obligé d'implémenter les six protocoles haut-cités pou pouvoir être fonctionnels. Néanmoins s'il veut être adressé dans le réseau et donc reconnu en tant que pair, il est obligé d'implémenter au moins les protocoles Peer Resolver Protocol et le Endpoint Routing Protocol. Les pairs implémentent différent protocoles selon les besoins de chacun. Ainsi par exemple si un pair désire les ressources disponibles et faire connaître les siennes, il sera obligé d'implémenter le PDP.

La figure ci-dessous montre l'organisation des protocoles JXTA par ordre d'interdépendance.

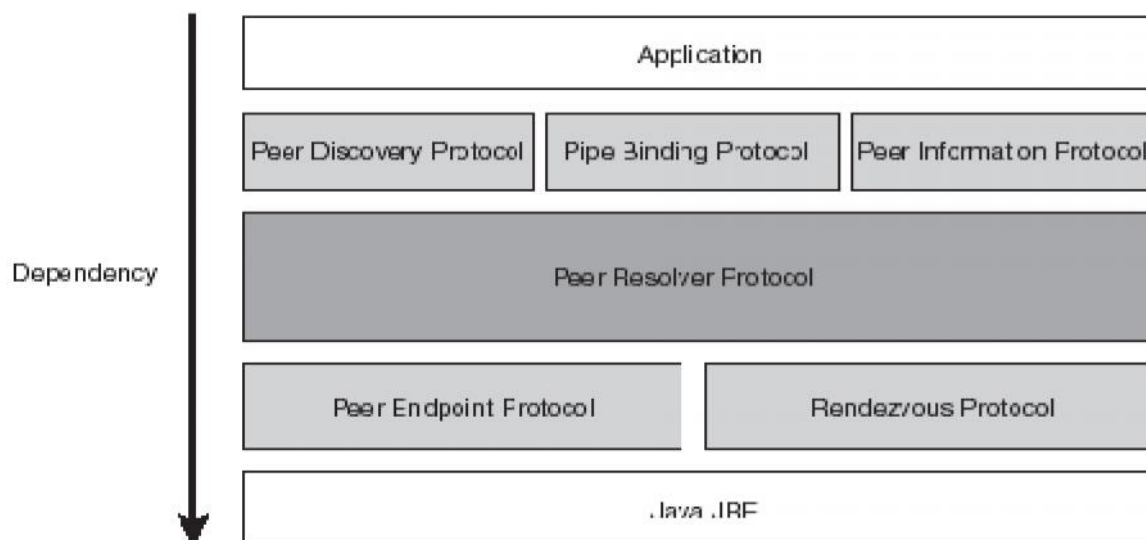


Figure 3.4 L'architecture des protocoles de la plateforme JXTA.

Nous remarquons que les protocoles Peer Discovery, Peer Binding et Peer Information Protocol sont au même niveau et se situent au dessus des autres. Nous notons aussi que le protocole Peer Resolver est à un niveau intermédiaire entre les deux protocoles de haut niveau qui sont : Peer End Point et Rendez-vous, et les deux de bas niveau.

Avec ces protocoles les développeurs peuvent ainsi s'amuser à créer des leurs applications P2P, qui vont pouvoir tourner d'une façon décentralisée. Mais pou permettre à ces protocoles de travailler ensemble et de former un système complet, JXTA a mis en place une architecture basée sur un modèle en couche. Le paragraphe suivant présente l'architecture de la plateforme JXTA, qui implémente les protocoles conceptuels de JXTA.

6. L'architecture JXTA.

L'architecture de JXTA est une architecture en trois couches, chaque couche apporte ainsi ses services, permettant aux couches supérieures de les utiliser. Cette architecture permet l'utilisation, de façon harmonieuse, des différents protocoles pour bâtir un système complet, facilitant la réalisation d'application P2P. Cette architecture est formée de trois couches. La première couche est le "Core Layer" : la couche noyau. La deuxième couche est le "Service Layer" : la couche des services et enfin le "Application Layer" : la couche application. La figure ci-dessous représente l'architecture en couches de JXTA.

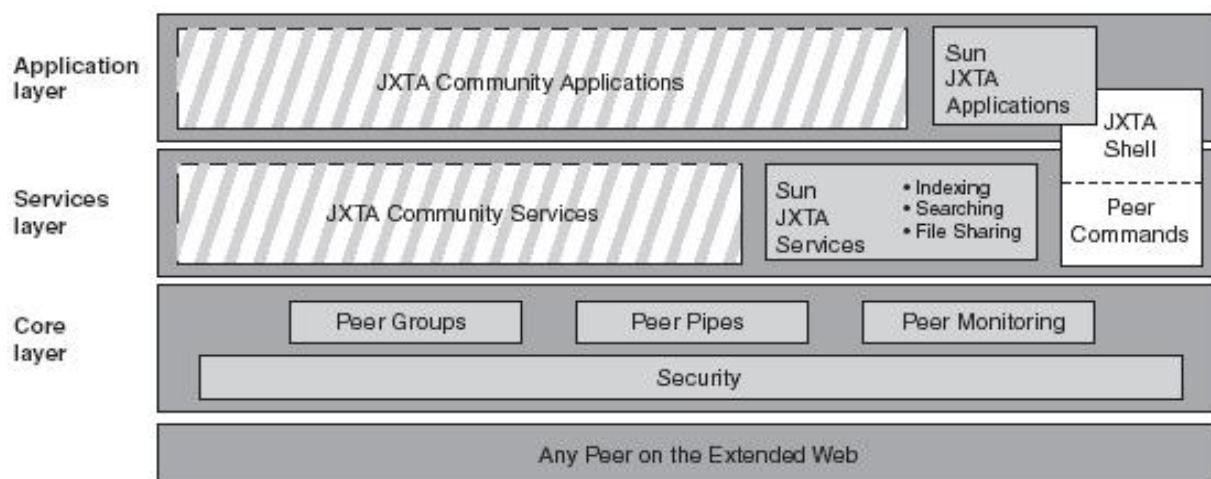


Figure 3.5 L'architecture en couches de JXTA

6.1. La couche Noyau.

C'est dans le "core layer" ou la "couche noyau" que se situe le code implémentant les différents protocoles. Nous remarquons sur la figure 5, que cette couche contient les mécanismes permettant la gestion des groupes de pairs, des "pipes" qui sont les canaux permettant la communication entre pairs, ainsi que peut avoir des pairs sur d'autres. La sécurité est aussi traitée à ce bas niveau de l'architecture pour maximiser les chances de succès des politiques de sécurité.

6.2. La couche service.

Le "service layer" ou "couche service" offre différents services qui utilisent les protocoles de la couche inférieure pour accomplir une tâche. Il existe deux types de services, les essentiels ou non. Les services essentiels sont donc ceux indispensables au bon fonctionnement du réseau et les autres non. Cette couche offre ainsi un ensemble de services aux utilisateurs afin de leur faciliter la tâche de la création d'applications en P2P.

6.3. La couche application.

L'"application layer" ou "la couche application" est la couche où viennent se placer les différentes applications, profitant des services de la couche précédente. Ces applications regroupent les peers et leurs offrent différentes fonctionnalités. Cette couche est réservée aux développeurs d'application P2P, c'est leur champ d'action et de création. C'est à ce niveau que les programmeurs mettront en œuvre leurs applications P2P en se basant sur les deux couches de bas niveau : le Noyau et les Services.

7. Conclusion.

Dans ce chapitre, nous avons étudié JXTA, un modèle conceptuel basé sur un ensemble de protocoles, et implémenté par une architecture de trois couches. Cette étude nous a permis, d'

une part, de présenter les différents principes et concepts de la plateforme JXTA. La majorité des applications P2P utilisent ces différents principes et concepts. C'est à base de ces derniers qu'on a pu implémenter notre application de partage de ressource en utilisant la plateforme JXTA. Le choix de JXTA pour un développer une application réseau P2P pour le partage, après avoir vu les solutions qu'il offre, devient justifié. Dans le chapitre qui suit on va parler de comment on a pu concevoir notre application, citer les protocoles et le raisonnement choisis pour sa conception.

Chapitre IV : Conception.

1. Introduction.

La fin du vingtième et le début du vingt unième fut marqué par l'apparition des applications P2P. Le logiciel Napster fut le premier a lancé le partage de fichiers à grande échelle sur l'internet en utilisant la technologie P2P. Très vite d'autres logiciels apparurent en se basant sur le même modèle P2P. Aujourd'hui la technologie P2P a fait du chemin plusieurs domaines du monde informatique, poussés par les avantages multiples qu'elle offre, font recours à cette technologie. Le monde éducatif n'est pas resté insensible à cette technologie, sujet même de notre mémoire qui porte sur le partage des ressources pédagogiques. Notre application a été conçue avec la plateforme JXTA, afin de créer un réseau P2P de partage sémantique de ressources pédagogiques. Au cours de chapitre, on va détailler comment, en suivant les concepts et les protocoles de JXTA, on a pu réaliser une application de partage de ressources pédagogiques.

2. Objectif.

L'objectif principal de notre projet est de réaliser une application réseau P2P pour le partage sémantique des ressources pédagogiques avec la plateforme JXTA. L'outil développé utilise le langage JAVA, intègre les protocoles de JXTA et de JDOM, pour la création des fichiers XML, et crée une interface de partage des ressources et de communication entre les utilisateurs de l'application sur un réseau P2P.

3. Contraintes de conception.

Concevoir une application dite P2P exige de tenir compte de certaines contraintes, dues aux exigences de ce type d'application. La plateforme JXTA permet de concevoir d'application réseau P2P.

La contrainte primaire pour concevoir une application P2P est d'être capable de créer un pair. Le pair créé, on doit lui attribuer des propriétés pour qu'il puisse être identifié et identifier par les autres pairs sur le réseau. Cette opération s'effectue en utilisant les protocoles offerts par JXTA.

Tous les pairs dans notre système doivent être de capable se trouver et d'établir une communication entre eux afin de pouvoir partager leurs ressources. Ici il s'agit du "Peer Discovery Protocol, protocole qu'offre JXTA qui permet au pairs non seulement de publier un advertisement pour qu'il puisse être trouvé sur le réseau , publier la liste des fichiers qu'il partage, déclarer le peerGourp au quel il appartient mais aussi faire savoir au autres pairs sur le réseau les services qu'il offre.

L'accès au réseau respecte plusieurs approches comme : un accès sécurisé au réseau ou un accès libre au réseau. Un pair qui utilise l'accès sécurisé dans le réseau, est identifié par un seul nom d'utilisateur et un mot de passe. Cela permet d'identifier facilement les propriétaires

des ressources qui ont été partagées sur le réseau, dans le cas d'une correspondance entre une recherche et un advertisement sur le réseau.

Une fois les pairs sur les réseaux, ils cherchent d'autres, et commencent à propager des requêtes sur le réseau ou à répondre aux requêtes disponibles sur le réseau.

Les pairs soumettent leurs requêtes en donnant un mot clé, et en cas de résultat approprié, la ressource s'affiche avec les détails sur l'identité du propriétaire, adresse JXTA de ce dernier, adresse JXTA du fichier. Ainsi chaque pair effectue un téléchargement d'une ressource en connaissance de ses propriétés.

Une autre contrainte, non moindre, que doit respecter notre application est qu'elle doit effectuer une recherche sémantique sur les fichiers partagés sur le réseau. Pour ce faire voyons d'abord les concepts informatiques qui nous ont permis de définir un moteur de recherche sémantique.

4. Ontologie Informatique

4.1. Introduction.

La rapidité de l'évolution de la masse d'information dans tous les domaines a généré un besoin d'organisation et de structuration des contenus. En philosophie, l'ontologie est l'étude de l'être en tant qu'être, c'est-à-dire l'étude des propriétés générales de ce qui existe.

Par analogie, le terme est repris en informatique et en science de l'information, où une ontologie est l'ensemble structuré des termes et concepts représentant le sens d'un champ d'informations, que ce soit par les métadonnées d'un espace de noms, ou les éléments d'un domaine de connaissances. L'ontologie constitue en soi un modèle de données représentatif d'un ensemble de concepts dans un domaine, ainsi que des relations entre ces concepts. Elle est employée pour raisonner à propos des objets du domaine concerné.

Les ontologies servent à la représentation des données échangées dans un domaine particulier afin de faciliter la communication interne au système informatique et externe entre les différents acteurs du domaine. Leur utilisation peut varier de la représentation des données à la recherche d'informations.

4.2. Intégration ontologie dans l'application.

L'intégration de l'ontologie dans notre application a été la structuration de concepts d'appartenance des fichiers partagés par un utilisateur. Pour ce faire on a utilisé les fichiers XML pour y représenter ces concepts avec JDOM.

JDOM est une API open source java dont le but est de représenter et de manipuler un document XML de manière intuitive pour un développeur java sans requérir une connaissance pointue en XML.

La finalité de la classe generateurXML de notre application a été de créer le fichier XML suivant :

```
<?xml version="1.0" encoding="UTF-8"?>
<RessourcePedagogiqueS>
  <categorie>
    informatique
    <soucategorie>
      programmation
      <soucategorie>
        linux
        <ressource>
          <titre>linux_tutoriel</titre>
          <auteur>www.siteduzero.com</auteur>
          <resume>Linux est un syst me d'exploitation tr s riche, vous allez le voir. On peut y
trouver de nombreux logiciels diff rents et il existe des centaines de fa ons diff rentes
de l'installer.Pour simplifier la vie des utilisateurs et pour leur permettre de faire un choix, on
a cr   diff rentes distributions de Linux. C'est un concept qui n'existe pas vraiment
sous Windows.*</resume>
        </ressource>
      </soucategorie>
    </soucategorie>
    java
    <ressource>
      <titre>apprenez_a_programmer_en_java</titre>
      <auteur>www.siteduzero.com</auteur>
      <resume />
    </ressource>
  </soucategorie>
</soucategorie>
<soucategorie>
  bureautique
  <ressource>
    <titre>excel</titre>
    <auteur />
    <resume />
  </ressource>
</soucategorie>
</categorie>
<categorie>
  telecom
  <soucategorie>
    frequence
    <ressource>
      <titre>testapi</titre>
      <auteur />
      <resume />
    </ressource>
  </soucategorie>
</categorie>
```

```
<categorie>biologie</categorie>  
</RessourcePdagogiqueS>
```

Où on voit comment a été catégorisé les ressources d'un pair du réseau : les ressources pédagogiques partage sur le réseau ont été mis dans des catégories et sous catégories. Par exemple la ressource "linux_tutoriel.pdf" est dans la catégorie informatique, sous catégorie "programmation", sou catégorie "linux".

Cette catégorisation des ressources nous permettra par exemple, si l'utilisateur entre comme mot clé de la recherche "informatique" de renvoyer tout les ressources contenues dans toutes les catégories associé la catégorie "informatique", qui sont la programmation, la bureautique et les sous catégories sous les sous catégories cité s'il y en a, et il en sera de même si l'utilisateur entre un mot-clé qui est contenu dans le nom d'une sous catégorie.

Si le mot clé n'est ni contenu dans le nom d'une catégorie ni dans celui d'une sous catégorie, alors l'application cherchera dans les titres de toutes les ressources partage sur le réseau. Une correspondance avec le mot clé entré.

5. Plan de la conception de l'application.

Pour arriver à concevoir l'application, on a dû suivre plusieurs étapes. La conception de l'application a nécessité de tenir compte des contraintes ci-haut citées. Les étapes franchies pour arriver au but final sont les suivantes :

5.1. Création d'un pair.

La création d'un pair se fait au travers de la classe StartJXTA. Cette classe va successivement : créer le pair, le connecter au réseau par défaut de JXTA qu'est NetPeerGroup, récupérer les services de ce groupe tel que le Discovery Service :

```
try{  
    // Connexion au groupe par défaut  
    netPeerGroup = PeerGroupFactory.newNetPeerGroup();  
}catch(PeerGroupException e){  
    //si il ne peut se connecter au groupe par défaut alors il va se deconnecter  
    System.out.println("[-] erreur:" + e.getMessage());
```

```
e.printStackTrace();  
  
System.exit(-1);  
  
}  
  
//recuperation des services  
  
myDiscoveryService = netPeerGroup.getDiscoveryService();
```

Après la connexion au groupe par défaut, le pair alors tentera alors de trouver le 8maiGroup dans le NetPeerGroup. Après un certain nombre d'essai, si le 8maiGroup n'est pas trouvé, il va alors créer et propager l'avis de recherche du 8maiGroup pour qu'à la prochaine fois qu'un pair se connectera au groupe global il puisse trouver le 8mai group s'il a envi de s'y connecter.

Voici le code qui permet au pair de faire ca :

```
//ID du groupe qui sera utilisé pour créer le groupe  
  
private final String stringID = "jxta:uuid-4E0742B0E54F4D0ABAC6809BB82A311E02";  
  
//recherche dans le cache local de l'existence du 08maiGroup  
  
adv =  
myDiscoveryService.getLocalAdvertisements(DiscoveryService.GROUP,"Name","08maiGroup");  
  
//recherché de pair connecté au 08maiGroup  
  
myDiscoveryService.getRemoteAdvertisements(null,DiscoveryService.GROUP,"Name","08maiGroup",1);  
  
//creation du 8maiGroup  
  
08maiGroup = createGroup();  
  
//connection au groupe  
  
joinToGroup(08maiGroup);  
  
  
  
//partie essential pour la création du groupe  
  
ModuleImplAdVERTISEMENT myMIA =  
netPeerGroup.getAllPurposePeerGroupImplAdVERTISEMENT();  
  
myNewGroup = netPeerGroup.newGroup(getGID(),  
myMIA,
```

```
"08maiGroup ",
"08mai P2P File Sharing Application");
```

Après l'exécution de la classe "StartJXTA Class", il y a une méthode dans cette classe qui renvoie les informations sur elle pour toutes les classes qui voudraient l'utiliser.



Figure 4.1 Diagramme de la classe StartJXTA.

5.2. Communication des Peers.

Une fois que les peers ont été créés, et qu'ils ont rejoint le "08maiGroup"; ils peuvent commencer à établir une communication entre eux. Ainsi un peer est capable d'envoyer un message à un autre. On a utilisé deux classes pour cet effet :

- Chat Input class : cette classe permet la réception des messages.
- Chat Output class : cette classe permet l'envoi des messages.

Les deux classes utilisent un fichier XML appelé : "08maiPipe.adv". La présence de ce fichier est obligatoire, sans ce fichier l'application ne s'exécute pas. Ce fichier contient les informations telles que : le type et le nom des Pipes, Pipe UUID. Ces informations permettent d'envoyer des messages sur les différentes Pipes. Sans la présence du fichier XML les peers ne peuvent pas communiquer entre eux, ce qui insinue que l'application ne marcherait pas.

```
<?xml version="1.0" ?>
  <!DOCTYPE jxta:PipeAdvertisement (View Source for full doctype...)>
- <jxta:PipeAdvertisement xmlns:jxta="http://jxta.org">
  <Id>urn:jxta:uuid-
59616261646162614E5047205032503337165B7AB98642E6A6918591D053591704</Id>
  <Type>JxtaUnicast</Type>
  <Name>8maiPipeAdv</Name>
</jxta:PipeAdvertisement>
```

Le pipe avertissement.

5.2.1. Chat Input Class.

Cette classe une fois appelée par un peer, il cherche Pipe Service du "08maiGroup" pour pouvoir créer une "Input pipe". Cette classe, pour créer une nouvelle "Input pipe", utilise l'information présente dans le fichier "08maiPipe.adv". Une fois l'"Input pipe" créée, l'interface "Pipe Event Handler" est implémentée. Elle permet de détecter l'arrivée des messages. Les messages entre les Input et les Output pipes sont sous le format XML, et contiennent les données comme : le nom du peer émetteur, l'UUID du peer émetteur. La partie importante du code de cette classe est donnée ci-dessous.

```
//Obtaining Pipe Services from 08maiGroup
myPeerID = 08maiGroup.getPeerID().toString();

//Reading "08maiPipe.adv" file
FileInputStream is = new FileInputStream("08maiPipe.adv");

//Creating new Pipe using information from advertisement file
pipeAdv = (PipeAdvertisement)AdvertisementFactory.newAdvertisement(
    MimeMediaType.XMLUTF8,is);

//Starting the Pipe Event Handler interface
public void pipeMsgEvent(PipeMsgEvent ev)
{
    log.append("[+]Message Received...\n");
    Message myMessage = null;
    try{
        myMessage = ev.getMessage();
        if(myMessage == null){
```

```

return;
}
} catch (Exception e) {
System.out.println("[-]Exception!");
e.printStackTrace();
} //XML file that received, and its elements
MessageElement me = myMessage.getMessageElement("peerName");
MessageElement me2 = myMessage.getMessageElement("peerID");
MessageElement me3 = myMessage.getMessageElement("chatMessage");
MessageElement me4 = myMessage.getMessageElement("Time");
}

```

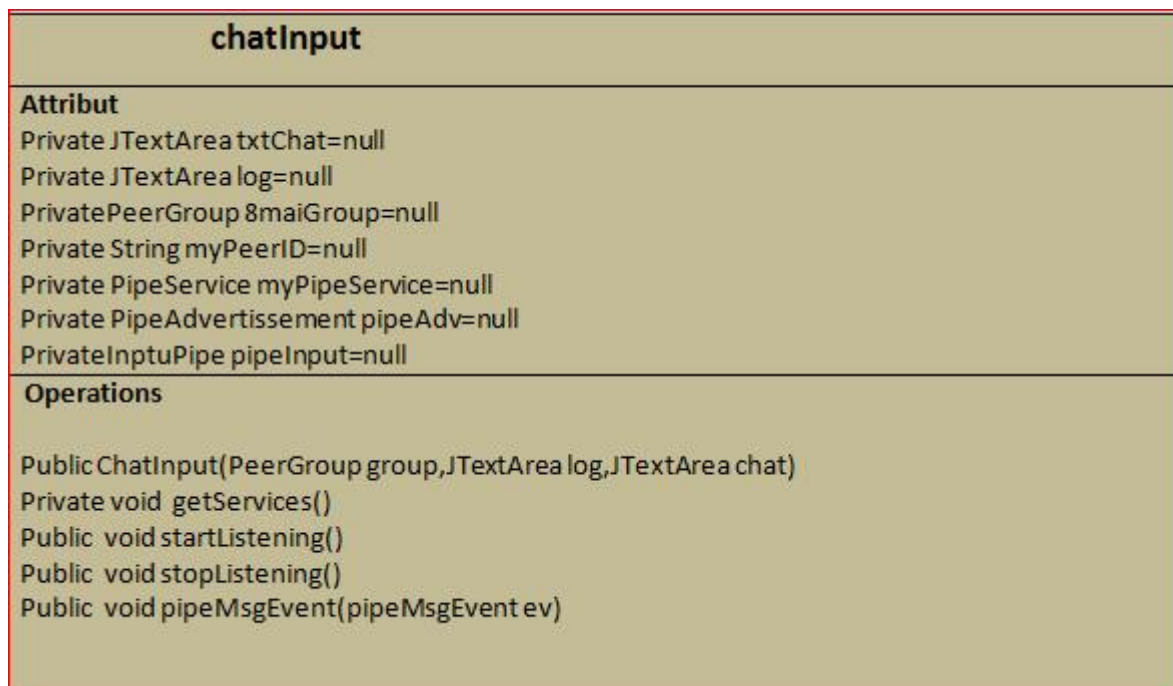


Figure 4.2 Diagramme de la classe ChatInput.

5.2.2. Chat Output class.

Cette fonctionne comme la Chat Input class. Elle utilise les mêmes procédures comme la Chat Input class, à part qu'au lieu de créer une "Input pipe", crée une "Output pipe". Après avoir créé l'"Output pipe", elle lui ajoute un message XML contenant : le nom du peer, l'ID du peer, le contenu du message, et l'heure de la création du message.

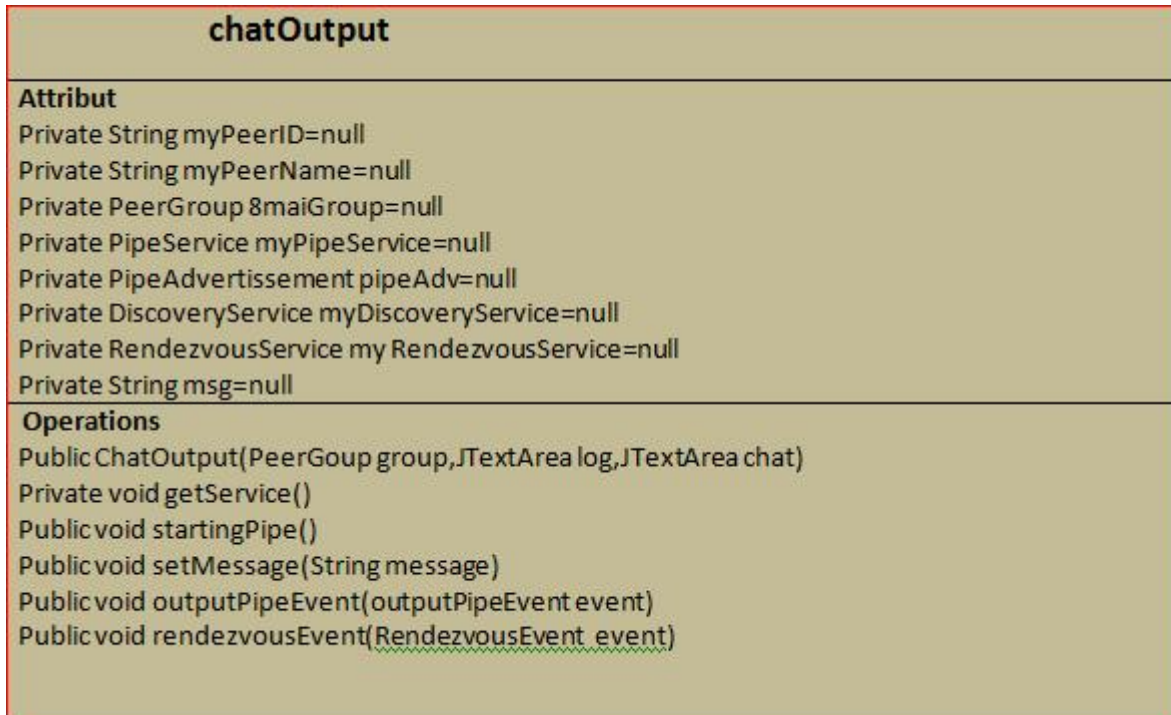


Figure 4.3 Diagramme de la classe ChatOutput.

5.2.3. Peer Listing class.

L'objet de cette classe, c'est d'établir une liste de noms des peers actifs dans le "08maiGroup". Cette classe utilise le "Discovery service" du groupe pour créer l'objet "Discovery Event Handler". Cet objet est responsable de la mise à jour des peers sur le groupe, dès qu'un advertisement d'un peer est détecté sur le groupe "08maiGroup". Une partie essentiel du code de la classe Peer Listing est donné ci-dessous.

```
//obtention des services de discovery
myDiscoveryService = 08maiGroup.getDiscoveryService();

//creation d'un objet d'ecoute
public void discoveryEvent(DiscoveryEvent event)
{
    DiscoveryResponseMsg res = event.getResponse();
    String name = "unknown";
    boolean isInList = false;
    PeerAdvertisement peerAdv = res.getPeerAdvertisement();
    if(peerAdv != null){
```

```

name = peerAdv.getName();
}
PeerAdvertisement myAdv = null;
Enumeration en = res.getAdvertisements();
Vector peerList = new Vector();
    while(en.hasMoreElements()){
        myAdv = (PeerAdvertisement) en.nextElement();
        peerList.addElement(myAdv.getName());
    }
//actualisation de la liste des peers
updatePeerList(peerList);
}

```

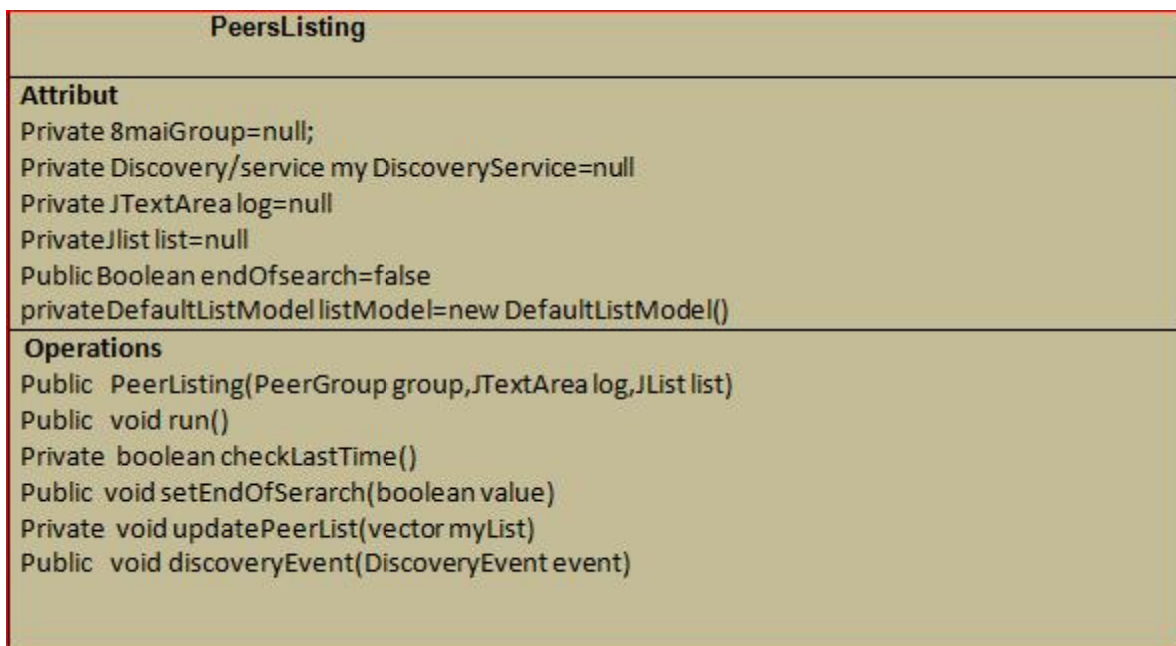


Figure 4.4 Diagramme de la classe PeerListing.

5.3. Le téléchargement des fichiers.

Le téléchargement des fichiers s'effectue après que la recherche d'un utilisateur ait abouti. C'est une phase importante dans l'application, car elle permet à l'utilisateur d'obtenir la ressource qu'il a recherchée. Cette partie de l'application est gérée par "DownloadFile class".

La classe " DownloadFile " est une sous-classe (hérite) de la classe "GetContentRequest" dans la librairie "CMS ". Après avoir lancé le processus de recherche, l'utilisateur attend de recevoir le fichier d'avertissement, et quand il l'obtient la classe se charge du téléchargement. L'exécution de cette classe débute le téléchargement du fichier choisi. L'application affiche le taux du transfert du téléchargement pour informer l'utilisateur de l'état d'avancement et du succès du transfert ou de son échec. Une partie essentielle du code de la classe " DownloadFile " se trouve ci-dessous.

```
//cette classe herite de la classe GetContentRequest
class GetRemoteFile extends GetContentRequest
{
public GetRemoteFile(PeerGroup group, ContentAdvertisement contentAdv, File destination)
{//appel de la super classe et initialisation
super(group, contentAdv, destination);
this.progressBar = progress;}
//notification sur l'avancement du telechargement
public void notifyUpdate(int percentage)
{progressBar.setValue(percentage); }
//notification de fin de téléchargement
public void notifyDone()
{
log.append("[+]telechargement termine.\n");}
```

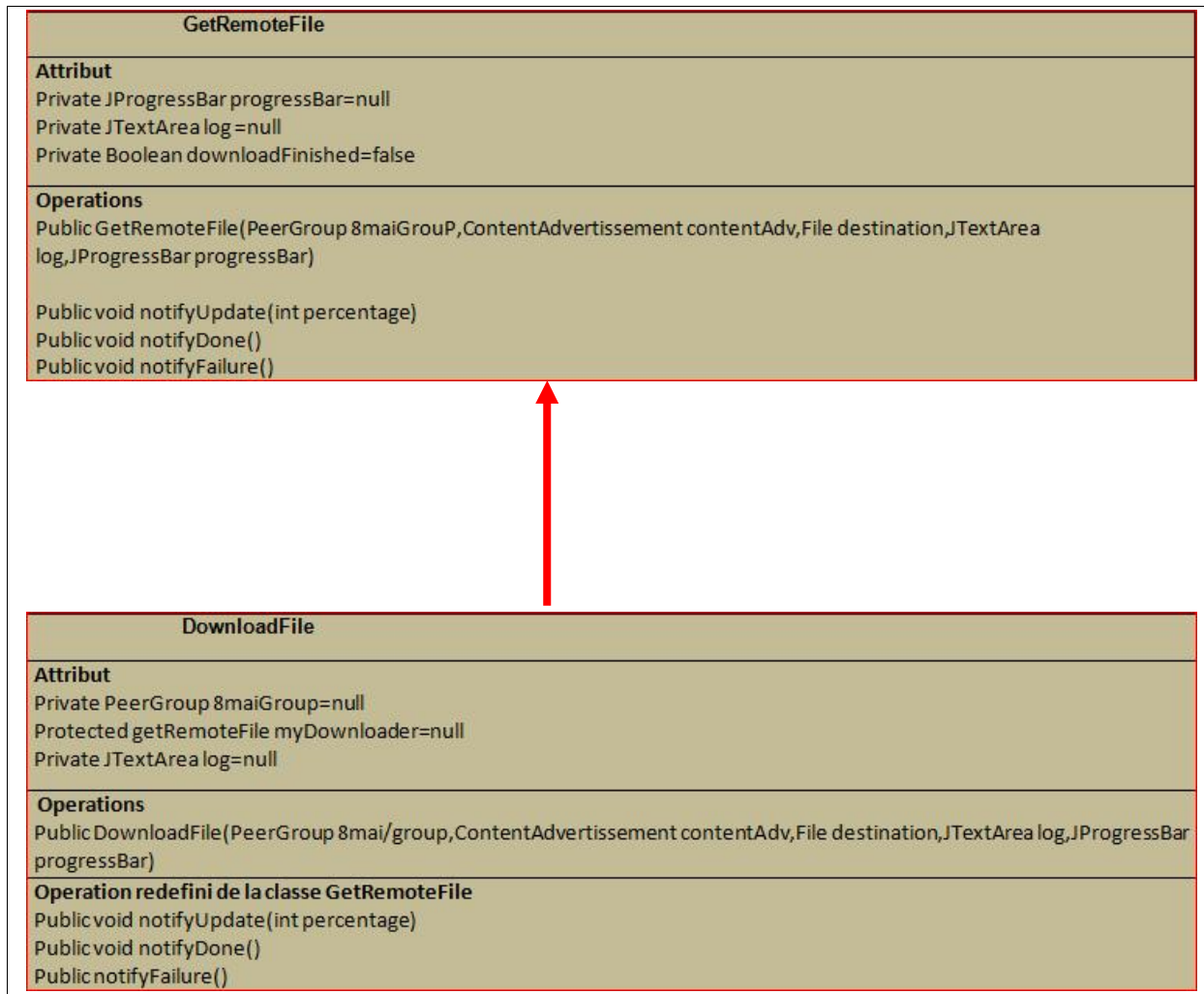


Figure 4.5 Diagramme de la classe DownloadFile.

5.4. Classe SearchFile.

Le but de cette classe est de rechercher parmi les ressources partagées sur le réseau les ressources correspondant au mot clé entré par l'utilisateur. Comme mentionné plus haut, la classe SearchFile va d'abord effectuer une requête XML qui lui permettra d'avoir une liste de titre qu'il va ensuite pouvoir entrer en paramètre à la classe ListeRequestor de JXTA pour pouvoir récupérer les avis relatifs aux titres des fichiers trouvés avec la requête XML et ainsi donner la possibilité à la classe DownloadFile de les télécharger. La partie du code ci-dessous montre cette opération :

```

public void run() //cause this thread to execute as long as needed to find
{
    // the Contents
    while(true) {
  
```

```

if(running == false){      break;      }

requeteXML(searchKey,titreTrouve,searchKey);

//for(int i=0; i < titreTrouve.size();i++){

    requestor=newListRequestor(SaEeDGroup," titreTrouve.get(i);",log, table);

requestor.activateRequest();

try{

Thread.sleep(8*1000); //le temps maximum pour la recherche d'un document

} catch(InterruptedException ie)      {

    stopThread();      }

//}

}    log.append("[-]Recherche terminé.\n");

}

```

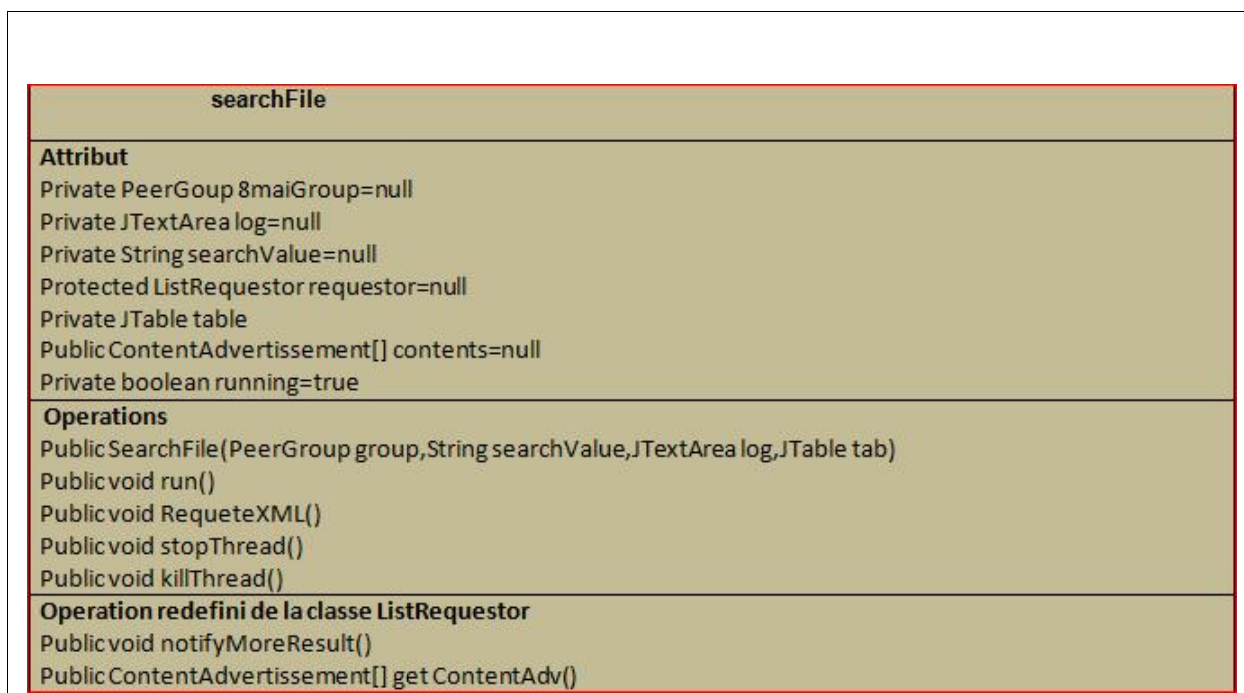


Figure 4.6 Diagramme de la classe SearchFile.

6. Fonctionnement générale de l'application.

La "FrmMain class" est la classe principale qui gère l'ensemble des classes de l'application. Elle utilise une variété de composants standards JAVA comme : JButton, JTable, JList,... pour la construction de l'interface de l'application. Cette classe joue le rôle de parent des autres classes. Elle permet l'initialisation de tous les objets et les connecte au NetPeerGroup. Les classes "Chat Input class" et "Chat Output class", sont initialisées par la "FrmMain class" pour commencer le chat service qui permet l'échange des messages entre les peers.

Elle initialise la classe "PeerListing ", pour permettre aux peers de se connecter au "08maiGroup", ce qui permet de pouvoir lancer le "Sharing Service" qui permet l'échange des fichiers entre les peers. Le code ci-dessous est une partie de la classe "FrmMain class" qui initialise toutes les classes ci-haut citées.

```
private void MenuItemConnectActionPerformed(java.awt.event.ActionEvent evt) {//GEN-FIRST:event_MenuItemConnectActionPerformed

// initialization de tout les objets

    MenuItemConnect.setEnabled(false) ;

//Starting Chat Services, which includes chat input and Output using the same pipe.

    chatIn = new ChatInput(connection.get08maiGroup(), this.txtLog, this.txtChatArea);
chatIn.startListening();

    chatOut = new ChatOutput(connection.get08maiGroup(), this.txtLog, this.txtChatArea);
chatOut.start();

//initialization de la classe peerListing

    peersList = new PeersListing(connection.get08maiGroup(), this.txtLog, this.ListOfPeers);
//Start Sharing Services

    File myPath = new File(Path);

    launchSharing = new 08maiSharing(connection.get08maiGroup(), this.txtLog, myPath);

    //debut du thread de partage

    launchSharing.start();
```



Figure 4.7 Diagramme de la classe frmMain.

7. Exécution de l'application.

Cette section présente brièvement l'exécution de l'application, et explique les étapes. Mentionnons que, pour que l'application puisse s'exécuter il faut que les bibliothèques de JXTA soit présent dans le répertoire à partir duquel l'application s'exécute ainsi que le fichier XML 8maiPipeAdv.

Au début de l'exécution l'utilisateur est prié d'accepter les termes et les conditions d'utilisation de l'application sinon, s'il refuse l'application se ferme.

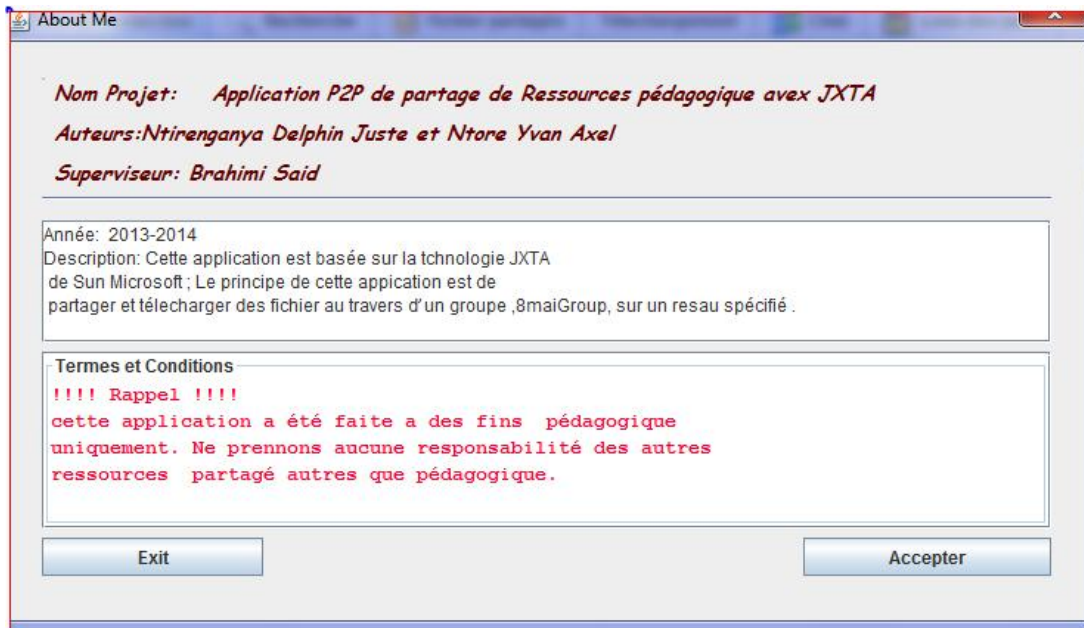
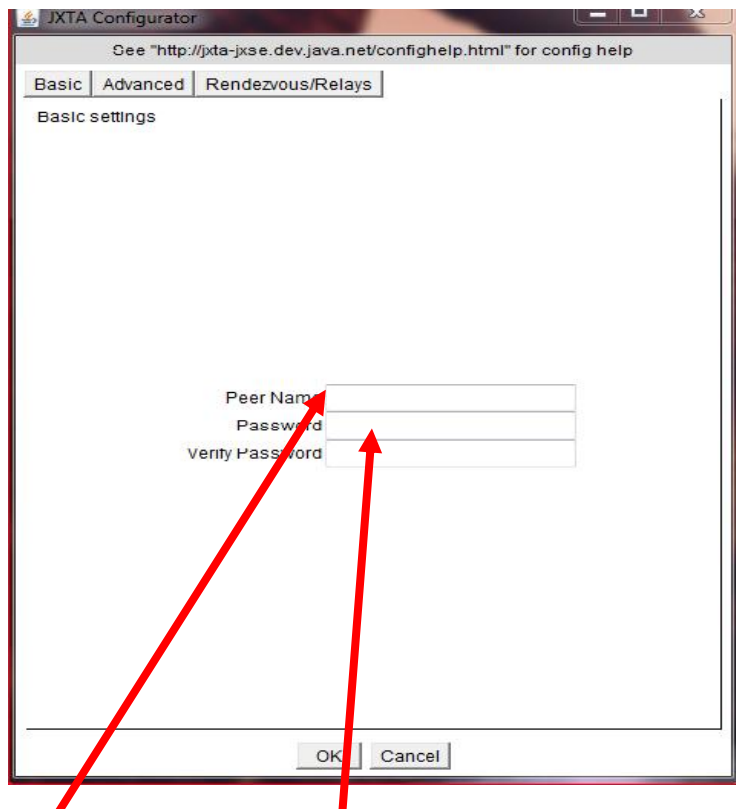


Figure 4.8 Termes et conditions de l'application.

Après l'acceptation des termes et des conditions de l'utilisation de l'application, si c'est la première fois que l'utilisateur utilise l'application, il se voit demandé de fournir un nom d'utilisateur et un mot de passe et de préciser le dossier qu'il veut partager sur le réseau et ainsi un fichier qui s'appelle config.ini sera créé et stockera toute ces informations.



Nom d'utilisateur

mot de 4.9 passe

Figure 4.9 Identification sur le réseau.

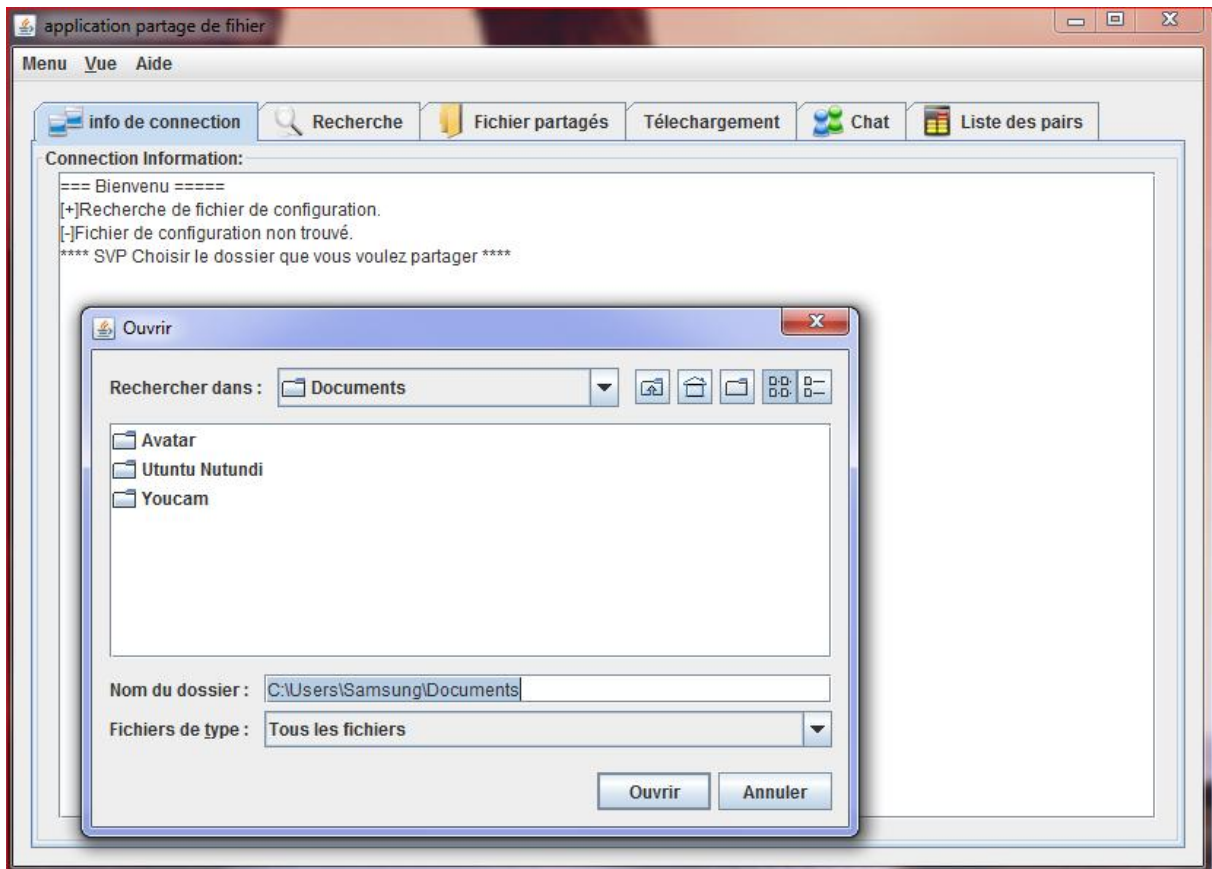


Figure 4.10 choix du dossier à partager.

Une fois toutes ces étapes franchis, l'application se connectera au réseau global par default, récupérera les services de ce dernier, se connectera au groupe 8maigroup et sera alors capable de partager des fichiers et d'en télécharger.

8. Conclusion.

Ce chapitre a décrit l'étude conceptuelle de notre application. On a donné les différents outils qui nous ont permis dans la conception, ainsi que les contraintes à relever. Aussi, nous avons détaillé les classes qui ont à concevoir l'application, en utilisant les diagrammes de classes.

Le prochain chapitre va présenter les techniques utilisées pour implémenter l'application conçue dans ce chapitre.

Chapitre V : Implémentation

1. Introduction.

Après la présentation de l'architecture de notre application dans le chapitre 4, nous allons dans ce chapitre aborder la phase de sa réalisation, on va décrire les différentes étapes empruntées, et les différents aspects techniques de notre projet. Le chapitre débute par une petite présentation de l'environnement logiciel, sur lequel l'application a été réalisée. Il se termine sur une présentation du fonctionnement de l'application et de son interface graphique.

2. Outils de développement.

Notre application a été réalisée avec les outils suivants :

2.1. Le langage utilisé.

Nous avons utilisé le langage JAVA pour implémenter notre application. C'est un langage orienté objet conçu par James Gosling en 1994 chez SUN. Il est portable, ne dépend pas d'une plateforme donnée, ce qui permet de l'utiliser que ce soit sur Windows, sur Macintosh, ou sur d'autres systèmes d'exploitation sans aucune modification. JAVA est donc un langage multiplateforme, permettant aux développeurs d'écrire des codes qu'ils peuvent exécuter dans tous les environnements. Le langage JAVA possède une riche bibliothèque de classes comprenant des fonctions diverses.

2.2. La plateforme JXTA.

JXTA est un projet Open Source lancé par Sun_Microsystems en avril 2001. Le but de JXTA est de pouvoir interconnecter n'importe quel système sur n'importe quel réseau. JXTA est constitué par un ensemble de six protocoles ouverts basés sur le XML et une API pour Java afin de fournir une plateforme générique pour le P2P. Ces protocoles ne dépendent ni d'aucun langage, ni d'aucune plateforme, mais ils protocoles sont imprésentables sur n'importe quel langage, ainsi permettant à des concepteurs de logiciel d'avoir les outils nécessaires pour la création d'application P2P innovatrice. En 2004, JXTA regroupait une communauté de développeurs dépassant 30.000 individus, travaillant sur une centaine de projet.

2.3. JDom.

DOM est l'acronyme de *Document Object Model*. C'est une spécification du W3C (World Wide Web) pour proposer une API qui permet de modéliser, de parcourir et de manipuler un document XML. Le principal rôle de DOM est de fournir une représentation mémoire d'un document XML sous la forme d'un arbre d'objets et d'en permettre la manipulation (parcours, recherche et mise à jour). JDOM est une API open source java dont le but est de représenter et de manipuler un document XML de manière intuitive pour un développeur java sans requérir une connaissance pointu en XML. JDOM permet donc de construire des documents, de naviguer dans leur structure, d'ajouter, de modifier, ou de supprimer leur contenu.

3. Architecture de l'application.

L'architecture de notre application est représentée le diagramme de la figure 5.1 ci-dessous :

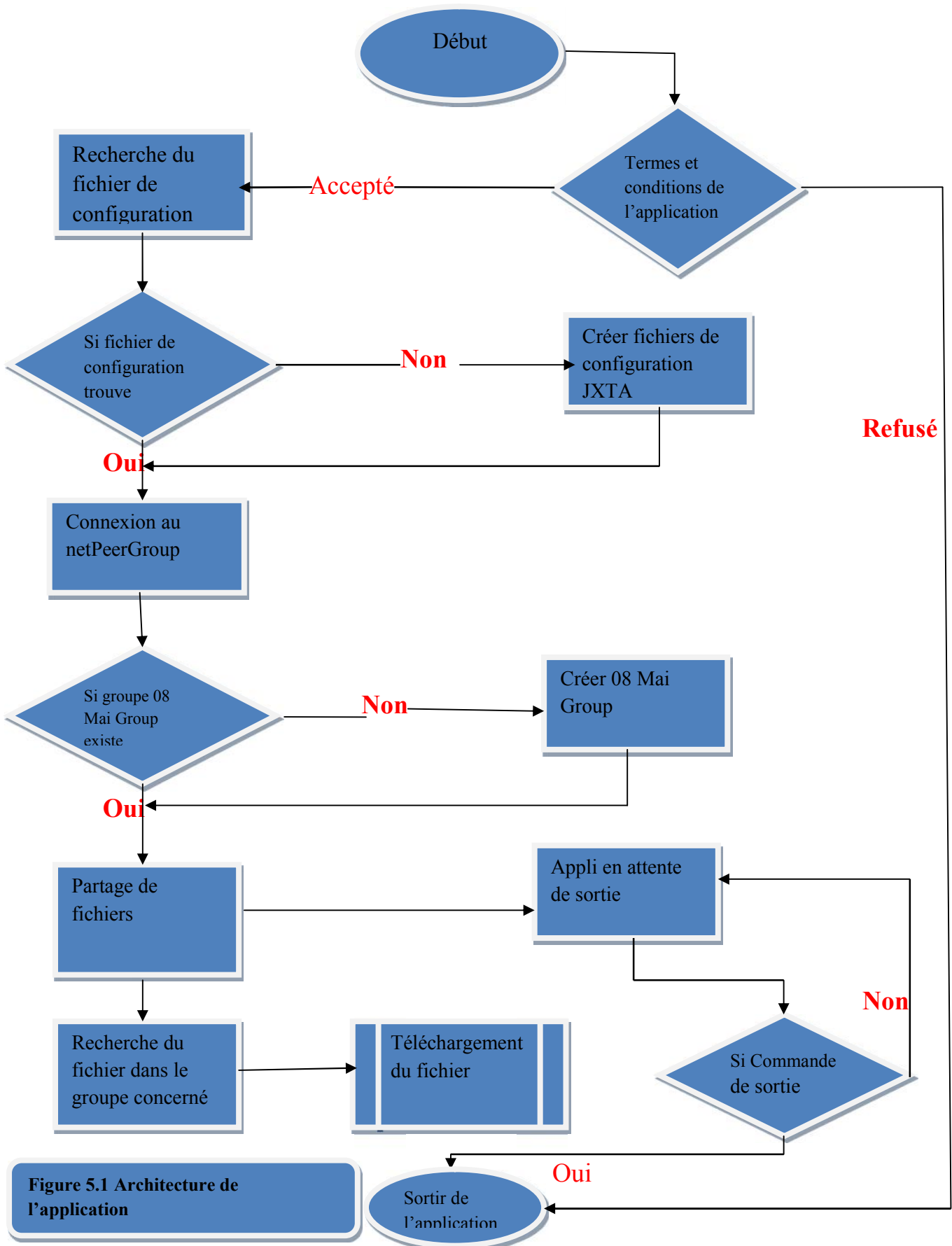


Figure 5.1 Architecture de l'application

3.1. Interface utilisateur

L'interface utilisateur affiche toutes les fonctionnalités. Il a été conçu de façon à rendre l'utilisation de l'application plus facile et plus intuitive. Il englobe toutes les étapes représentées par la figure 5.1. LA suite du paragraphe décrit les fonctions de chaque composante de l'interface utilisateur.

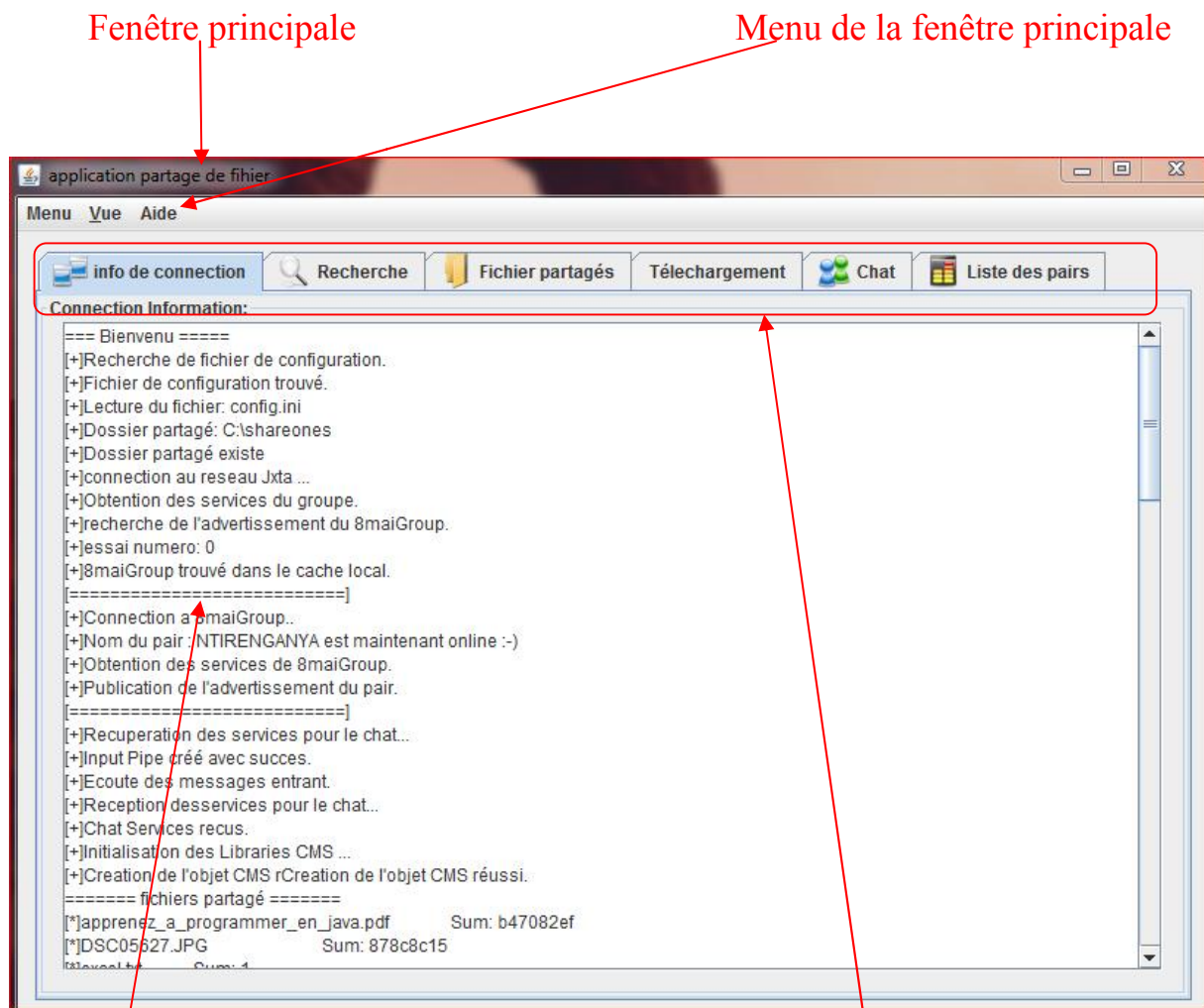


Figure 5.2 Interface utilisateur.

Information sur les étapes que suit l'application

Fenêtre secondaire représentant chacune des fonctionnalités de l'application

4. Menu de la fenêtre principale

Cette composante de l'application offre le choix entre 3 éléments qui chacun représente une fonctionnalité différente :

- 1) Menu
- 2) Vues
- 3) Aide

4.1. Option Menu.

Avec l'option **Menu** l'utilisateur a le choix de se déconnecter ou se reconnecter du groupe de partage de ressource pédagogique. Cette option donne le choix à l'utilisateur de se déconnecter du réseau n'importe quand il le souhaite. La figure 5.3 ci-dessous montre les éléments de l'option Menu.

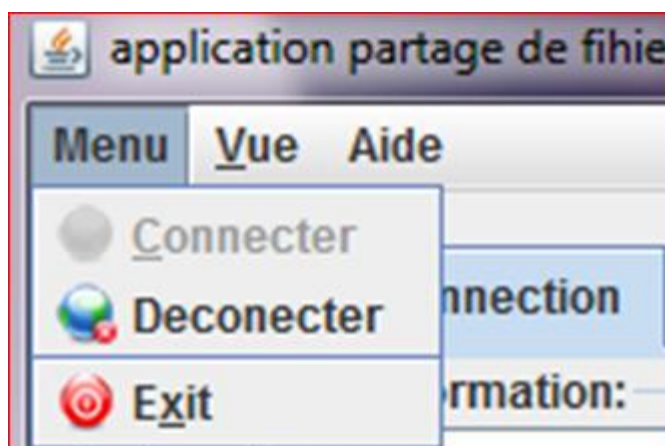


Figure 5.3 Option Menu.

4.2. Option Vue.

L'option **Vue** offre à l'utilisateur de choisir entre 3 apparences différentes de l'interface. L'utilisateur peut choisir entre : Window looking style, Swing looking style et autres. Les figures qui suivent, montrent un aperçu des apparences proposes par l'option Vue.

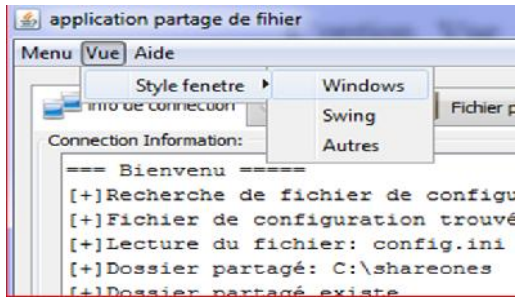


Figure 5.4 Apparence Window.

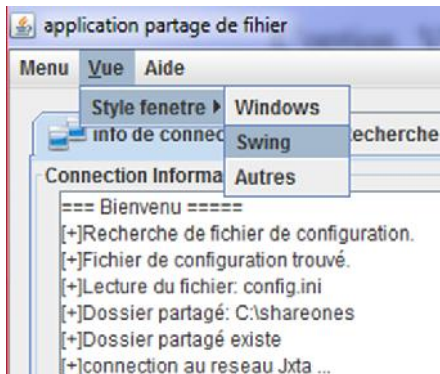


Figure 5.5 Apparence Swing.

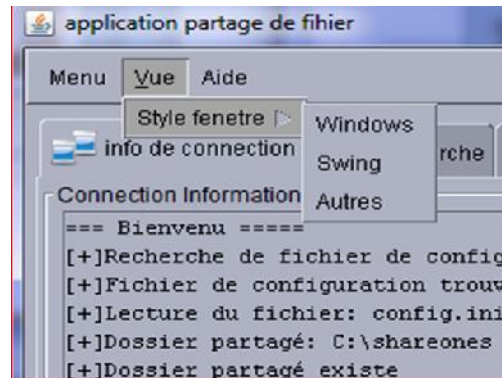


Figure 5.6 Apparence Autres.

4.3. Option Aide.

Le menu **Aide** permet à l'utilisateur de voir une brève description de l'application ainsi que les termes et les conditions d'utilisation de l'application. C'est la même boîte de dialogue aussi qui est affichée au début de l'exécution de l'application pour s'assurer que l'utilisateur ait accepté les termes et les conditions d'utilisation. La figure 5.6 montre la brève description de l'application.

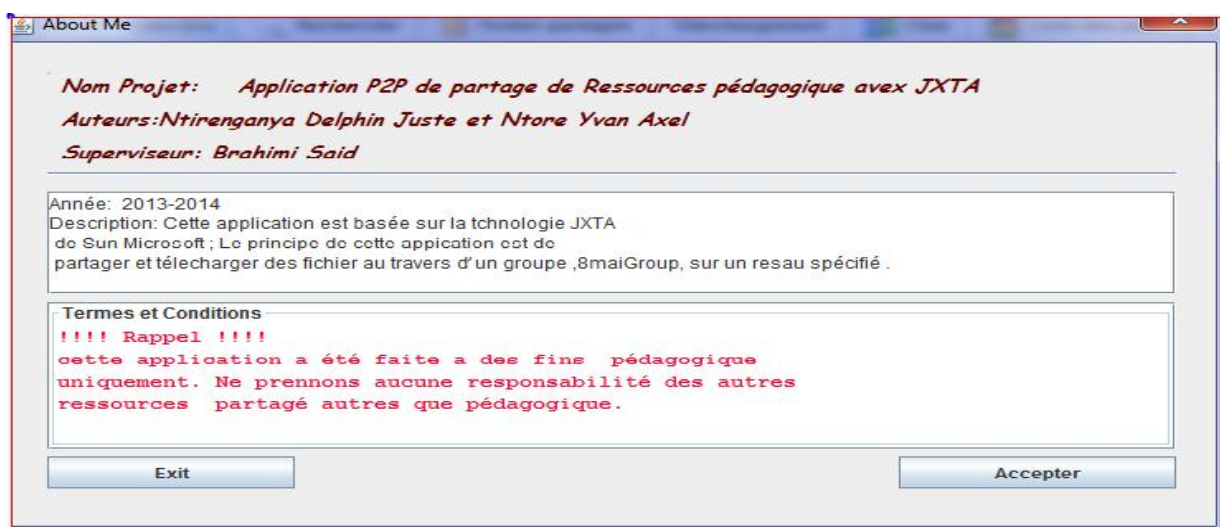


Figure 5.7 Termes et conditions de l'utilisation de l'application.

5. La fenêtre secondaire de l'application.

La fenêtre secondaire de l'application comprend plusieurs onglets dont chacun est responsable d'une fonctionnalité différente de l'application.

5.1. Onglet info de connexion.

Cet onglet est responsable de l'affichage des détails de connexions ainsi que toutes les informations relatives aux différentes classes utilisé par l'application. Cet onglet permet entre autre d'avoir les détails sur les étapes du fonctionnement de l'application. En cas d'erreurs, il donne les informations sur la partie où se sont produite les erreurs, et le pourquoi de l'erreur. Un exemple d'informations affichées dans cet onglet, est donné ici : Nom du pair, le statut du pair dans le réseau JXTA, dossier partagé, fichiers partagé mais aussi le statut de toutes les classes utilisé dans l'application. La figure 5.7 montre l'onglet info de connexion.

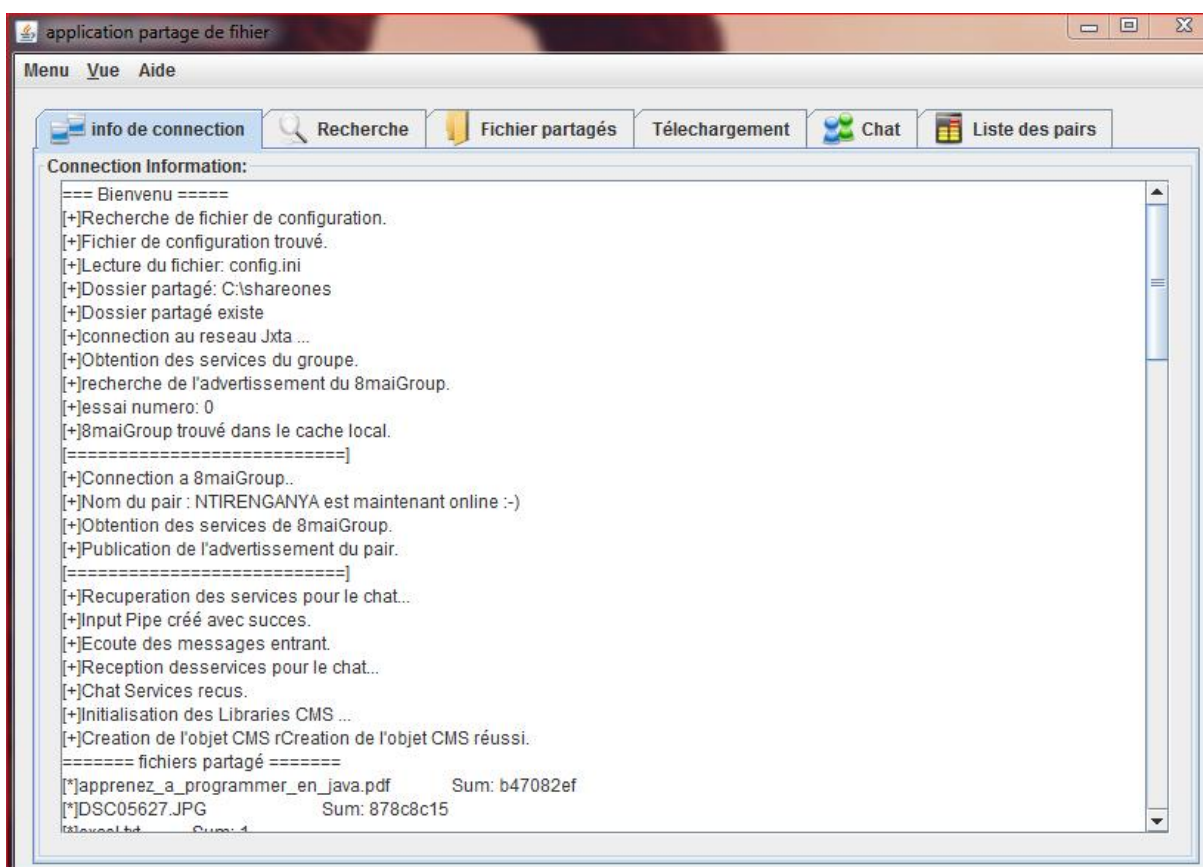


Figure 5.8 Info de connexion.

5.2. Onglet recherche.

Cet onglet est responsable de la recherche de documents dans le groupe. Le contenu de la recherche sera propagé chez tous les membres du groupe. Si une, ou des, correspondance est effectuée, les différents documents relatifs à la recherche seront proposés à l'utilisateur pour téléchargement. La figure 5.8 illustre l'onglet de recherche.

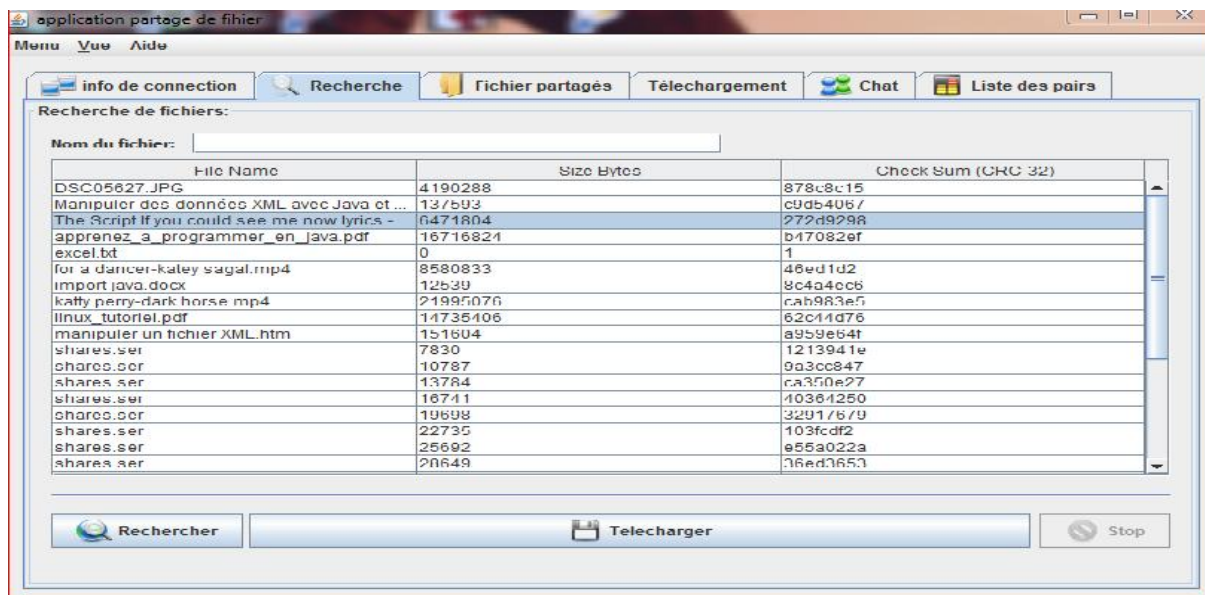


Figure 5.9 onglet de recherche.

5.3. Onglet Fichiers partagés.

Cet onglet est fait pour permettre à l'utilisateur de voir les ressources que lui-même a partagées sur le réseau. Il montre également les informations sur les fichiers dans le dossier partagé tel que : type du fichier, nom du fichier, taille du fichier en byte et le checksums du fichier. La figure 5.9 illustre l'onglet des fichiers partagés.

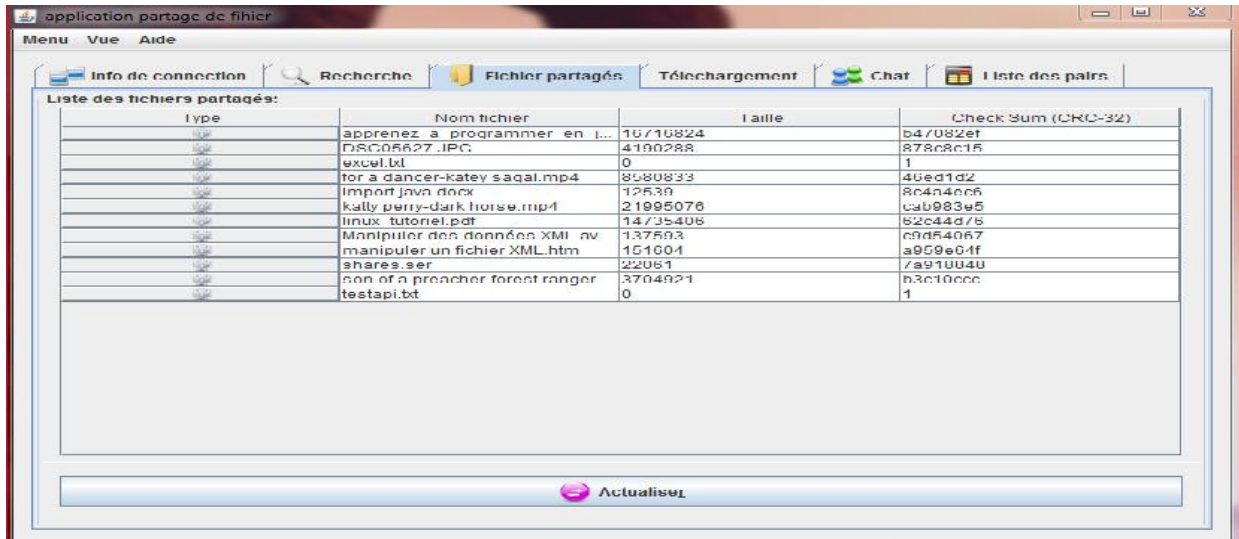


Figure 5.10 Onglet fichiers partagés

5.4. Onglet téléchargement.

Cet onglet présente à l'utilisateur toute les informations concernant le téléchargement en cours et lui donne d'autres informations supplémentaires tel que : Nom du fichier en téléchargement, statut du téléchargement, etc. L'utilisateur a également la possibilité d'interrompre un téléchargement ou bien de vérifier qu'un fichier qu'il vient de télécharger, n'est pas corrompu. La figure 5.10 illustre l'onglet de téléchargement.

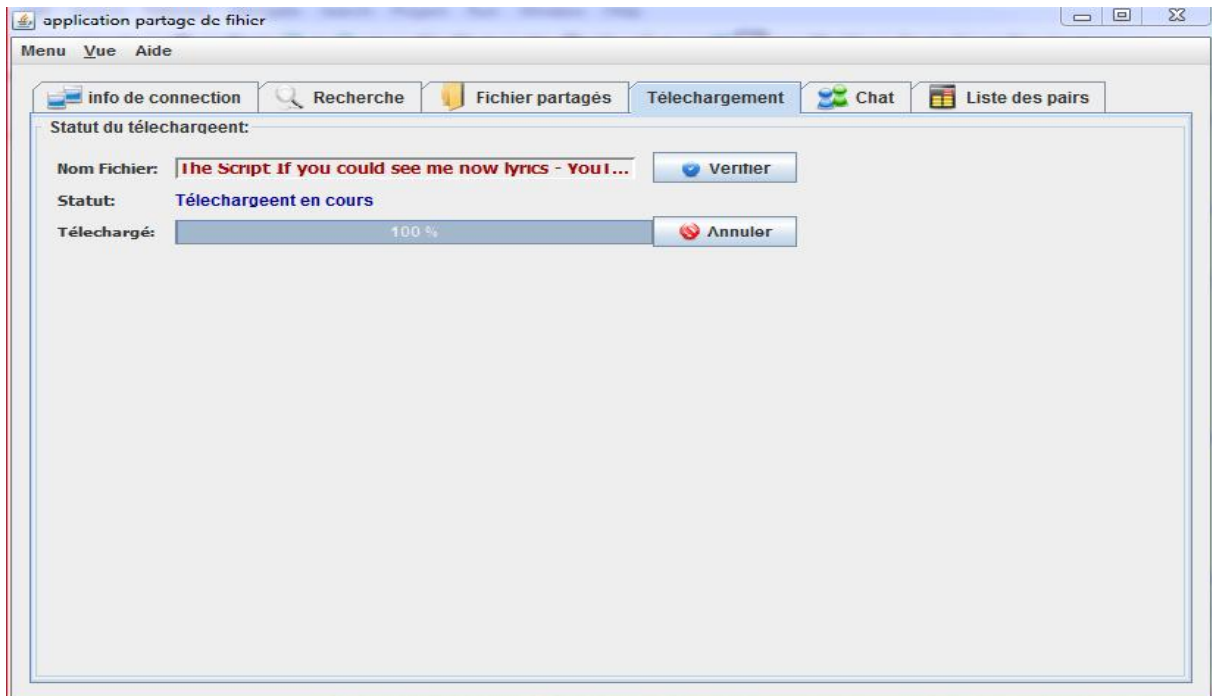


Figure 5.11 Onglet de téléchargement.

5.5. Onglet chat.

L'application ne permet pas seulement le partage des fichiers entre les utilisateurs, l'onglet Chat propose un espace de messageries instantanés, pour permettre aux utilisateurs de converser en direct en plus de s'échanger des ressources. L'utilisateur choisit dans la liste des pairs la personne avec qui il a envie d'échanger les messages, puis clique sur le bouton "Envoyer" pour envoyer son message.



Figure 5.12 Onglet de Chat.

5.6. Onglet Liste des pairs.

La découverte des autres pairs est un aspect très important dans la technologie de partage de ressources JXTA. Cet onglet permet de chercher, trouver et afficher les autres pairs qui sont présent dans le groupe. Pour chercher un pair il suffit à l'utilisateur de cliquer sur le bouton "Trouver les pairs", et il lui suffit de cliquer sur la bouton "Stop" pour arrêter la recherche. La figure 5.12 illustre l'onglet d'affichage des pairs connectés sur le réseau.

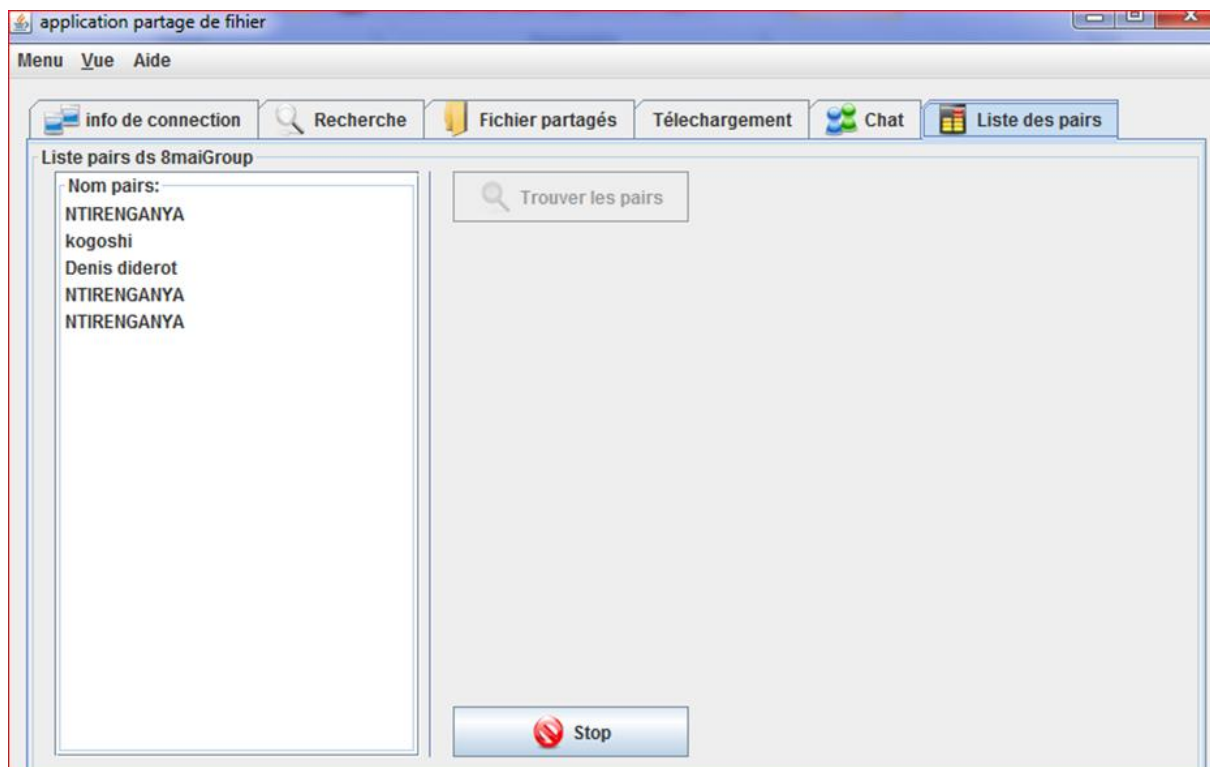


Figure 5.13 onglet liste des pairs.

6. conclusion

Dans ce chapitre, nous avons présenté les outils utilisés pour développer notre application de partage de ressources en utilisant les réseaux sémantiques sur un réseau Peer to Peer avec JXTA, à savoir Eclipse, JXTA, JDom.

Les détails de l'implémentation sont illustrés par des imprimés écran, pour qu'à chaque étape de 'utilisation de l'application, l'utilisateur ait une idée de ce que l'application peut lui offrir comme service.

*Conclusion Générale et
perspectives.*

Conclusion générale

Avec ce projet on a exposé sur la technologie Peer to Peer en relation avec le partage de ressource sémantique. Le but principal de ce projet était d'implémenter une application réseau Peer to Peer décentralisé en partageant sémantiquement les ressources pédagogique. Les outils de développement utilisés, sont les technologies Java, JXTA et JDOM. Le projet a abouti sur l'établissement d'un moteur de recherche sémantique qui s'appui sur la catégorisation des ressources pédagogiques disponibles, en soulignant la nature de leur contenu suivant un concept prédéfini.

Les problèmes rencontrés lors de la réalisation de l'application, étaient surtout dû à l'outil JXTA. Plusieurs versions de la plateforme font des bugs et ne marche qu'avec certaines versions précises de systèmes d'exploitation. On doit souligner aussi un manque évident de documentation sur cette récente technologie. Mais notre appréciation personnelle, confirme que JXTA est une plateforme qui offre beaucoup d'avantage pour la programmation des applications Peer to Peer.

Comme perspective, le système qu'on a implémenté fait la recherche sémantique de fichiers en se basant sur un fichier XML prédéfini, qui renferme une description statique de toutes les ressources partagées sur le réseau. Dans le futur il serait intéressant de voir, cette application intégrer un générateur du fichier XML qui serait dynamique. Le fichier catégoriserait périodiquement les ressources en tenant compte des requêtes des utilisateurs et ainsi il étendrait la sémantique des relations. Ce qui permettrait l'ajout des nouveaux concepts tirés des recherches et des nouvelles ressources provenant des nouveaux utilisateurs au fur et à mesure que le réseau s'agrandit.

Bibliographie.

- [1] : Michaël HAUSPIE : Contributions à l'étude des questionnaires de services distribués dans les réseaux ad hoc. Pour obtenir le grade de Docteur en Informatique. Université des Sciences et Technologies de Lille, Numéro d'Ordre : 3506, Année : 2005.
- [2] : NTIRENGANYA et RUSHOZA : Partage des ressources pédagogiques en utilisant les réseaux sémantiques. Diplôme de licence en Informatique. Université 08 Mai 1945 de Guelma, Année : 2011.
- [3] : Anne GUEGAN : Valorisation des ressources pédagogiques en ligne au SCD de l'Université de Limoges : le cas des Universités Numériques Thématiques. Formation initiale des bibliothécaires d'État, Année : 2010.
- [4] : Mohamed GHARZOULI : Composition des Web Services Sémantiques dans les systèmes Peer-to-Peer. **Thèse** Pour obtenir le diplôme de Doctorat en sciences en Informatique. Université Mentouri Constantine Faculté des Sciences de l'Ingénieur Département d'Informatique. Année : 2011.
- [5] : Thomas CERQUEUS : Contributions au problème d'hétérogénéité sémantique dans les systèmes pair-à-pair : application à la recherche d'information. Thèse de Doctorat, Discipline : Informatique, Spécialité : Informatique. UNIVERSITÉ DE NANTES FACULTÉ DES SCIENCES ET DES TECHNIQUES, ÉCOLE DOCTORALE SCIENCES ET TECHNOLOGIES DE L'INFORMATION ET MATHÉMATIQUES. Année : 2012.
- [6] : Abdel Kader Keita et al : Un outil d'aide à la construction d'ontologies pré-consensuelles : le projet Towntology. LIRIS Laboratoire d'Informatique en Image et Systèmes d'information UMR5205 CNRS INSA, Campus de la Doua, Bâtiment Blaise Pascal (501), 20avenue Albert Einstein 69621 VILLEURBANNE CEDEX.
- [7] : Nathalie BUDAN et al : Nouvelles Technologies Réseau Les réseaux peer-to-peer Fonctionnement, exemples, limites. Année : 2003.
- [8] : Dimitri Caron et Dominique Poure : L'empire du « Peer to Peer ». Université Paris Dauphine. Année : 2007.
- [9] : Anthony Heukmes : www.2dconcept.com. Année : 2009.
- [10] : jxta.free.fr.
- [11] : Saber Mansour : Un modèle de gestion distribuée de groupes ouverts et dynamiques d'agents mobiles. Thèse présentée à l'Université de Pau et des Pays de l'Adour pour obtenir le diplôme de DOCTORAT. Année : 2007.
- [12] : Joseph D. Gradecki : Mastering JXTA, Building Java Peer to Peer Applications. Publisher: John Wiley & Sons, ISBN: 0471250848.