Université Abou Bekr Belkaid
Tlemcen Algérie
جامعة أبي بكر بلقايد
تلمسان الجزائر

REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE
UNIVERSITE ABOU BAKR BELKAID – TLEMCEN
FACULTE DES SCIENCES
DEPARTEMENT D'INFORMATIQUE

Mémoire de fin d'études

pour l'obtention du diplôme de Master en Informatique

Option: *Réseaux et systèmes distribués (R.S.D)*

Thème

Conception et réalisation

D'un réseau social distribué

Réalisé par :

- **Abderrezzaq Khatir**
- **Imane Labbas**

Présenté le 29 Juin 2013 devant le jury composé de :

- *Mme. Amel Halfaoui* *(Présidente)*
- *Mr. Mohamed Tadlaoui* *(Encadreur)*
- *Mr. Amine Boudefla* *(Examinateur)*
- *Mr. Badr Benmammar* *(Examinateur)*

Année universitaire : 2012-2013

« La multiplication des moyens de communication au premier rang desquels les réseaux sociaux permettent aux sociétés de progresser vers une universalité des échanges constructifs tant individuels que collectifs. Néanmoins, la communication mal gérée ou manipulée se détruit elle même. La nécessité d'une gestion humaniste des échanges internet devient une impérieuse nécessité pour un équilibre global et une vision d'avenir collective. »

Pascal Monsolve

REMERCIEMENT

Il est d'usage parce que normal en fin de mémoire d'effectuer les remerciements des personnes sans lesquelles tout ce projet n'aurait pu voir le jour.

Tout d'abord nos plus sincères remerciements vont à notre encadreur Mr. Mohamed Tadlaoui sans qui le présent projet n'aurait jamais pu voir le jour. Son aide précieuse et sa patience ont été d'un secoure remarquable afin de mener ce travail à bien.

Nous tenons à remercier tout particulièrement Mr. Badr Benmammar qui nous a fait l'honneur de présider le jury qui va avoir à se prononcer sur la qualité du travail présenté. Son enseignement de haute qualité a formé puis nourrit notre réflexion.

Nous remercions également Mr. Amine Boudefla dont les remarques pertinentes ont permis l'orientation correcte du projet que nous venons de développer et de présenter par ce mémoire. Notre intérêt pour la recherche, nous la lui devons. Son acceptation à évaluer notre travail a donné une chance réelle à ce projet.

Nos remerciements également à Mme. Amel Halfaoui qui n'a eu de cesse que de nous encourager et de nous conseiller dans ce projet. Sans elle notre motivation à participer à la startup weekend d'Alger l'année dernière n'aurait pas été si grande et nous n'aurions pu gagner le 2e prix du meilleur projet. Mme. Halfaoui a accepté également d'évaluer notre projet.

Nous remercions l'ensemble de nos professeurs dont le savoir précieux ont enrichi nos parcours respectifs ainsi que toutes les personnes non citées ici mais qui ont contribué de près ou de loin à l'accomplissement de notre travail.

DEDICACES

Je dédie ce modeste travail aux personnes qui me sont chères et sans la présence desquelles il ne m'aurait pas été possible d'être présents devant vous aujourd'hui :

Tout d'abord à mes chers parents qui m'ont soutenu tout au long de mon parcours et dont tous les sacrifices justifient à eux seuls ma ténacité à achever ce projet et à avoir travaillé avec autant d'acharnement. Puissent-ils être fiers de voir leur benjamin mener à bien ses études.

Mes chers frères Boumediene et Mustapha dont la présence à mes côtés fut un soutien dans les moments de doutes et une force dans laquelle j'ai puisé.

A mes sœurs Fatiha et Souhila pour leur Amour sans lequel aucun projet n'eut pu aboutir.

A mes neveux Salim, Amira, el Hadi, Nadir et le petit Nassim qui me rappellent chaque jour que nous ne menons pas nos projets pour nous même mais également pour sa famille et au-delà de celle-ci, nous l'espérons, pour un plus grand nombre.

A mon binôme et amie Narymen pour nos échanges parfois mouvementés qui furent une stimulation de chaque instant à avancer et à réussir.

A mes amis Ilhem, Ryad, Djamel, Salah ainsi qu'à tous mes autres amis qui me font me souvenir que le travail prend aussi appui sur l'amitié comme force vivifiante.

A ma très chère amie Birgit et à son fils Stéphan qui m'ont beaucoup aidé dans la vie et qui continuent de m'aider.

A toi Pascal pour des choses qui remplissent les pages de ce mémoire et encore plus.

A toutes celles et ceux que je porte en mon cœur et que je n'ai pas cité sans qui un tel projet n'aurait pas véritablement eu de sens.

ZaQ

DEDICACES

Je dédie ce mémoire de fin d'études :

À Mon très cher père et ma très chère mère qui m'ont épaulé tout au long de mes études.

À mon frère Ryad et ma princesse Nihad.

À mon grand père, que dieu te garde.

À toute ma famille.

Un grand merci à tous mes amis, principalement ZaQ, Ilou, Oussama, Manel, Nawal, Nadi, Djamel, Kika, Dido, Mimi, Bouchra, Hafsa, Ilyass, Zineb, Mehdi, Chakib, Asma, Assia, Hakim, Khero, Reda, Otman & Zyad

Pour finir, je remercie toutes les personnes qui ont fait entrer de la couleur dans ma vie.

Narymen

RESUME

Ce travail s'inscrit dans le cadre de recherche concernant la problématique du contrôle et de l'utilisation des données au sein des réseaux sociaux centralisés.

L'objectif de ce mémoire est de faire une étude et un comparatif des architectures de quatre réseaux sociaux décentralisés que nous avons sélectionné, de proposer une nouvelle architecture distribuée et de développer un prototype qui repose sur cette architecture.

Nous proposons un nouveau modèle des réseaux sociaux distribués baptisé DisNet. Afin de valider notre modèle nous proposons un prototype nommé DisNet Prot. La problématique que nous essayons de résoudre est le stockage de données.

De plus nous apportons une solution nouvelle n'ayant pas encore été implémenté dans les réseaux sociaux actuels ni même dans les approches d'études consistant en la suppression des profils utilisateurs.

ABSTRACT

This work tries to establish a scop of research concerning the problems of control and use of the data in the centralized social networks.

The objective of this memory is to make a study and comparative architectures of four decentralized social networks which we selected, to propose a new distributed architecture and to develop a prototype which rests on this architecture.

We propose a new model of the distributed social networks baptized DisNet. In order to validate our model we propose a prototype named DisNet Prot. The problems that we try to solve are data storage.

Moreover we bring a new solution not having been implemented yet in the current social networks nor even in the approaches of studies consisting of the suppression of the user profiles.

TABLE DES MATIERES

LISTES DES FIGURES

LISTE DES TABLEAUX

INTRODUCTION

Il existe actuellement de fortes présomptions d'utilisations non autorisées des données personnelles des utilisateurs au sein des réseaux sociaux centralisés.

La récente affaire reprise par la presse internationale de cette utilisation frauduleuse par certaines agences gouvernementales américaines de premier plan incite à une prudence grandissante et à la proposition de solutions novatrices dans le champ de protection des données. Selon la chaine d'information LCI [1] concernant le système PRISM [2] : *« ... grâce à lui la National Security Agency, NSA, est au courant de tous les échanges sur Internet ; toutes les compagnies, Google, Facebook, Appel y participent. Selon le président Obama, l'Amérique n'a pas le choix ».*

Le président Obama a déclaré : *« Vous ne pouvez pas avoir 100% de sécurité et en même temps 100% de respect de la vie privée. »*

En effet ces réseaux sociaux centralisés ont construit leur accessibilité sur la gratuité d'accès, en contre partie l'utilisateur se doit de fournir des renseignements personnels conséquents que les réseaux non seulement stockent mais semblent fournir à des demandeurs extérieurs sous couvert de la mention sécurité nationale.

Mark Zuckerberg [3], le fondateur de Facebook a déclaré : *« ...Quand les gouvernements demandent des renseignements à Facebook, nous examinons chaque requête minutieusement afin de nous assurer qu'ils suivent un processus correct et en conformité avec les lois, et que seules les informations légales sont demandées.... »*

Nombreux sont les utilisateurs qui s'émeuvent de telles pratiques en violation flagrante des libertés fondamentales dont chaque individu jouit selon les lois internationales sur les libertés individuelles.

L'adhésion à un réseau social implique un contrat de confiance entre l'utilisateur de celui-ci et les acteurs du réseau social lui-même. Ce contrat de confiance devrait stipuler que l'utilisation des données personnelles ne peut se faire qu'après accord tacite de l'utilisateur ou que l'utilisation des données personnelles par le réseau social fait partie intégrante de l'accès à celui-ci.

Or, à ce jour, le contrat de confiance entre l'utilisateur et les réseaux sociaux centralisés n'est pas respecté pour plusieurs raisons :

- La première est que la majorité des utilisateurs n'est pas avertie de l'utilisation des données personnelles stockées à des fins mercantiles et commerciales ;
- Les réseaux sociaux centralisés se réfugient derrière la demande des gouvernements de transmission des données personnelles stockées afin de protéger les libertés collectives.

De nombreux cas à travers le monde sur divers réseaux sociaux centralisés ont fait apparaître ces dernières années l'incapacité de gestion des données personnelles par ces réseaux après leur divulgation. Les utilisateurs ont pu se retrouvé dans des situations personnelles délicates à la suite de l'utilisation sans accord préalable de tout ou partie des données personnelles stockées par les réseaux au premier rang desquels Facebook.

Certains pays ont pour objectif le durcissement de la législation en vue de la protection des données personnelles, il est difficile néanmoins d'équilibrer la notion de liberté individuelle ou d'entreprise avec la cadre législatif pouvant obliger les réseaux sociaux à établir un contrat de confiance avec les utilisateurs dans un cadre de visibilité totale.

Généralement l'utilisateur des réseaux sociaux centralisés est dépossédé par ceux-ci de leur capacité à gérer leur accès par exemple dans le cadre de la suppression de celui-ci. Les profils ne sont en définitives pas supprimés mais mis en sommeil, l'utilisation des données personnelles est toujours accessible par le réseau social même en l'absence de

connexion de l'utilisateur comme certaines affaires l'ont démontré ces dernières années dans divers pays.

La problématique actuelle peut également dépasser le simple cadre de l'utilisation des données personnelles ; par exemple l'ajout par certains réseaux sociaux de fonctionnalités additionnelles laisse envisager le transit par ces fonctionnalités de problématiques diverses : piratage, virus,… Or, ces ajouts mis en place par les réseaux eux même ne sont pas assumés par ces derniers qui déclinent toutes responsabilités d'utilisation par les usagers.

Nous sommes donc face à un problème majeur pouvant se résumé au droit à la vie privée face à la divulgation des données personnelles.

Si le cadre législatif semble être déficient actuellement, d'autres solutions même imparfaites permettent pourtant de proposer une alternative technique à ce qui vient d'être évoqué. Parmi ces solutions se trouve celle des réseaux sociaux décentralisés, que nous nous proposons d'étudier dans ce mémoire qui a été organisée comme suit :

- Le chapitre I présente la généralité sur les réseaux distribués.
- Le chapitre II s'organise autour d'une étude détaillée et comparative de quatre réseaux sociaux distribués.
- Le chapitre III propose une solution retenant les atouts des réseaux sociaux distribués examinés au chapitre II améliorés par des fonctionnalités novatrices.
- Une conclusion générale clôturera ce mémoire en même temps que nous aborderons quelques perspectives sur le futur des réseaux sociaux distribués.

CHAPITRE I :

LES RESEAUX DISTRIBUES

I. LE PEER TO PEER :

Le terme peer-to-peer abrégé en P2P est un modèle de réseau informatique proche du modèle client-serveur mais qui est distribué de manière à ce que les entités appelées peers jouent le double rôle client et serveur. Tous les ordinateurs récupèrent de l'information et la retransmettent, en fait ils interagissent afin d'offrir à une communauté un service de manière décentralisé comme montré dans la figure I.1.

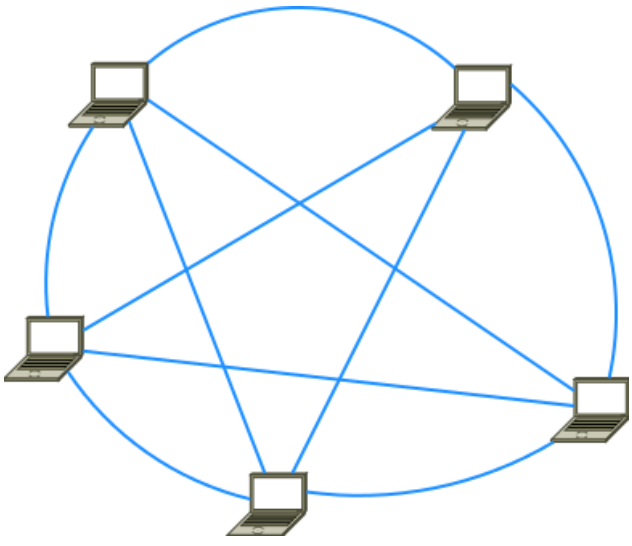


Figure I.1 : Structure du réseau P2P

Par ailleurs, contrairement aux idées admises, ce système n'est pas seulement utilisé pour le partage des fichiers tel que la musique, les vidéos ou les logiciels. De nombreuses entreprises en font un usage au quotidien et de diverses manières. Citons à titre d'exemple :

- La plate forme de calcul distribué BOINC [4] se base sur le P2P pour accéder à la puissance de calcul supplémentaire, quatre fois plus que le plus puissant des superordinateurs, dans le but de faire avancer la science, l'économie, l'art…
- Le service de stockage en Cloud d'Amazon [5] est basé sur un système de stockage de données en P2P.
- Le distributeur de musique libre Jamendo [6] transmet ses 36 000 albums à travers le réseau P2P BitTorrent [7].
- La technologie Flash [8] développée par Adobe Systems intègre maintenant un système d'échange P2P de flux vidéo.
- Le logiciel de téléphonie par Internet Skype utilise son propre réseau peer-to-peer.
- La plate forme du micro blogging Twitter utilise le P2P pour ses mises à jours.

II. LES RESEAUX SOCIAUX DISTRIBUES :

Les services de réseaux sociaux sont des plateformes dont le but est de regrouper des utilisateurs qui se connaissent et de partager de l'information (nouvelles, photos, vidéos, hyperliens, …) entre eux, selon le type de relations entretenues.

À la différence d'un service de réseau social centralisé (Facebook, MySpace ou LinkedIn), où toutes les informations sont gérées par une seule entité, les réseaux sociaux distribués (Diaspora [9], Movim [10] ou Lorea [11]) offrent un stockage et une diffusion flexible des informations. Ceci apporte une sécurité accrue, un meilleur contrôle de la vie privée et moins de contraintes quant à la liberté d'expression : l'utilisateur n'est plus seulement l'éditeur des informations, il en est aussi le diffuseur.

III. LES TABLES DE HACHAGES DISTRIBUEES :

Une table de hachage est une structure de données qui permet d'associer une clé à chaque élément. L'accès à chaque élément dans la table se fait via sa clé qui est hachée via la fonction de hachage. Le hachage est un nombre qui permet la localisation des éléments

dans un tableau. En fait c'est l'index de l'élément dans le tableau. Les tables de hachages sont nommées ainsi parce qu'elles se basent sur une fonction de hachage.

Une table de hachage distribuée abrégée en DHT quant à elle est une technologie qui permet l'identification et l'obtention d'une information dans un réseau P2P. Elle se constitue des nœuds qui sont répartis sur tout le réseau et qui agissent comme des sceaux d'informations car chacun d'eux possède une partie des données de la DHT.

La DHT fournit un algorithme de recherche efficace intitulé « lookup service » qui permet à un nœud participant de déterminer rapidement quelle machine est responsable de la délivrance d'une donnée en faisant un routage dans le réseau. En effet, la DHT grâce à ces nœuds agit pour trouver les nœuds qui contiennent les données et trouver où stocker les données.

De ce fait, les DHTs sont très utiles lorsque le nombre d'entrées est important. Elles sont utilisées notamment dans les protocoles des réseaux P2Ps tel que le protocole Chord [12], le protocole P2P CAN [13], …. En effet, les DHTs peuvent être utilisées dans plusieurs applications comme par exemple :

- Le système en réseau de forums Usenet [14]
- Le système de mailing sans serveur ePost [15]
- Le stockage de données OceanStore [16]
- Le back web anonyme FreeNet [17]

Il existe actuellement plusieurs DHTs sur le marché, certaines sont accessibles publiquement et gratuitement, d'autres sont restreintes aux personnes affiliées aux entreprises ou établissements accueillant des nœuds, nous en avons sélectionnés trois à permettre notre présentation :

III.1. KADEMLIA :

Kademlia [18] abrégée en KAD est parmi les premières tables de hachages distribuées qui fit son apparition en 2002. Elle spécifie la structure du réseau et l'échange de l'information à travers le lookup des nœuds. Ses nœuds communiquent entre eux pour former un réseau virtuel en utilisant le protocole UPD.

Chaque nœud est identifié par un nombre intitulé « node ID ». Cet identifiant est utilisé aussi par l'algorithme de Kademlia afin de localiser les fichiers hachés. En effet, le node ID fournit un mapping directe des fichiers hachés et stocke des informations sur l'endroit où obtenir les fichiers et les sources.

Afin d'explorer le réseau à la recherche d'une certaine valeur, l'algorithme KAD à besoin de connaître la clé associée à cette valeur. L'exploration s'effectuera en plusieurs étapes et dans chaque étape, l'algorithme trouvera les nœuds les plus proches de cette clé jusqu'à ce que le nœud contacté renvoi la valeur recherchée si elle existe ou une requête pour dire qu'il n'y a plus de nœud à proximité qui contient la valeur recherchée.

L'un des avantages de Kademlia est l'efficacité de son algorithme qui ne contacte que $\theta\ (log(n))$ nœud pendant sa recherche sur un total de n nœud dans le système. D'autres avantages se trouvent surtout dans sa structure décentralisée qui augmente la résistance contre les attaques par dénis de services. En effet, même si un ensemble de nœuds est atteint cela n'aura qu'un effet limité sur la disponibilité du réseau. De plus le réseau est capable de reconstruire le nœud perdu.

III.2. OpenDHT :

OpenDHT [19] est une table de hachage distribuée accessible publiquement, qui est apparue en 2005 et qui appartient à Planet-lab [20], qui lui est un réseau d'ordinateurs utilisé en tant que plateforme d'essais pour de la recherche orientée réseaux et systèmes distribués.

Aucune information d'identification ou de création des comptes n'est requise pour utiliser le service. De plus le stockage de données est réparti équitablement entre tous les clients actifs. Par conséquent, « put » et « get » peuvent être émises à n'importe quel nœud de la DHT.

Ce model de service de la DHT simplifie considérablement le déploiement des applications clientes. En effet, en utilisant OpenDHT pour le lookup et le stockage des données, les clients peuvent ignorer la complexité du déploiement et le maintien de la DHT pour se concentrer sur le développement des applications réparties.

La simple interface de « put and get » est accessible via les protocoles SUN RPC [21] et XML RPC [22]. En tant que tel, le service a un accès facile à partir de pratiquement tous les langages de programmation et peut passer derrière presque tous les pare-feu.

III.3. FREEPASTRY :

FreePastry [23] est une implémentation open source du protocole P2P Pastry [24]. Ce dernier est une couverture du routage du réseau Internet pour la mise en œuvre d'une DHT. Les paires clé-valeur sont stockées sur un réseau P2P redondant de serveur connecté à Internet.

Le protocole est amorcé en lui fournissant des adresses IP des peers qui sont toujours dans le réseau. Cet amorçage permet la reconstruction et la réparation d'une table de routage dynamique.

En raison de sa nature décentralisée et redondante, il n'y a pas de point de défaillance. Ainsi, un nœud peut quitter le réseau à tout moment sans avertissement avec peu ou pas de risque de perte de données.

Le protocole est aussi capable d'utiliser une métrique de routage fournie par un programme extérieur tel que le Ping afin de déterminer les meilleurs itinéraires à stocker dans sa table de routage.

IV. LA SAUVEGARDE FRIENDSTORE :

Friendstore [25] est un mécanisme de sauvegarde coopératif des données en ligne avec les nœuds de confiance étant généralement les amis de la vie réelle, facile d'utilisation pouvant servir pour des raisons diverses comme par exemple :

- Il n'y a aucun besoin de payer pour un service de sauvegarde en ligne.
- La coopération entre les amis en ligne et les données sauvegardées pour chacun.
- Si l'utilisateur est hors ligne, les amis qui gardent une copie de ses données peuvent servir les demandeurs de ses informations.

De plus, ce mécanisme est gratuit et open source, il peut être rajouté a d'autres applications en extension comme nous allons le voir dans ce le chapitre II.

CHAPITRE II :

ETAT DE L'ART

I. INTRODUCTION :

Les réseaux sociaux distribués en ligne sont apparus afin de riposter au problème majeur des réseaux sociaux centralisés évoqué dans l'introduction à savoir la protection des données des utilisateurs. Il en existe plusieurs tel que : Diaspora, Persona [26], LotusNet [27], SuperNova [28], ….

La particularité que ces réseaux ont en commun ce compose de trois contraintes : La confidentialité, la disponibilité et l'intégrité. Certains essayent de respecter au moins deux de ces contraintes, d'autres au contraire les respectent toutes, du moins selon eux, mais est-ce vraiment le cas ? C'est ce que nous allons voir à la fin de ce chapitre. Parmi l'ensemble des réseaux disponibles, nous en avons sélectionnés quatre majeurs en fonction de leur capacité à permettre notre présentation mais voyons d'abord ce que signifient ces contraintes.

I.1. CONFIDENTIALITE :

Les informations de l'utilisateur ne doivent être vues que par l'utilisateur lui même et les personnes avec qui il les a partagé. Aucune autre partie interne ou externe ne doit pouvoir consulter le contenu de ces informations.

I.2. INTEGRITE :

Les données et l'identité de l'utilisateur doivent être protégées contre les modifications non autorisées et les falsifications. La création de faux profils est facile au sein des réseaux sociaux, donc l'authentification doit s'assurer de l'existence des personnes réelles.

I.3. DISPONIBILITE :

La disponibilité des données doit être assurée pour que les données existent en permanence, ainsi la disponibilité des profils utilisateurs doit être protégée contre les censures (La suppression autoritaire d'information) et le Hijacking [29] (détournement ou possession illégale d'une donnée).

II. DEFINITION ET CONCEPTION :

II.1. PEERSON:

PeerSon [30] est une approche peer-to-peer distribuée et couplée avec un système de chiffrement. Son infrastructure est faite justement pour supporter les caractéristiques les plus importantes des réseaux sociaux en ligne (telles que : la messagerie instantanée, le mur, le fil d'actualité,…) dans un environnement distribué. Il vise à maintenir les caractéristiques des réseaux sociaux en ligne en surmontant deux limitations : La question de confidentialité et l'exigence d'une connectivité internet pour toutes les transactions.

Pour résoudre le problème de confidentialité, un système de cryptage et de contrôle d'accès est utilisé en couplage avec une approche P2P pour remplacer l'autorité centralisée des réseaux sociaux en ligne classiques. Ces mesures empêchent les violations de la vie privée des utilisateurs par les fournisseurs de ces systèmes, les annonceurs et même les utilisateurs eux même.

Sa conception répond principalement à trois exigences : Le chiffrement des données, la décentralisation de l'architecture et l'échange direct entre les utilisateurs. En un mot, le chiffrement assure la confidentialité pour les utilisateurs, la décentralisation basée sur l'utilisation d'une infrastructure P2P permet quant à elle l'indépendance des fournisseurs des réseaux sociaux en ligne, ce qui rend plus facile l'intégration de l'échange direct des données entre les appareils des utilisateurs dans le système.

a - LE CHIFFREMENT DES DONNEES :

Le moyen de protection de la vie privée dans ce contexte est de permettre aux utilisateurs de crypter et contrôler l'accès à leurs données. L'accès à ces données se fait à travers le partage de la clé appropriée, cependant, les considérations de la sécurité incluent l'amorçage, la distribution et la révocation des clés.

b - LA DECENTRALISATION DE L'ARCHITECTURE :

Pour mieux protéger la vie privée, les données des utilisateurs sont cryptées et accessibles uniquement à ceux qui ont les clés de déchiffrement. C'est l'utilisateur lui même qui détermine avec qui partager les clés de décryptages des données qu'il publie.

Par ailleurs, il n'est pas nécessaire de faire valoir le fait que même si les données sont cryptées et centralisées, le prestataire du service central pourrait être en mesure de décrypter ces données et de les utiliser.

c - ECHANGE DIRECTE ENTRE LES UTILISATEURS :

Puisque le service du réseau social est décentralisé et n'est pas basé sur un serveur web, les utilisateurs n'ont pas besoin d'être constamment en ligne pour l'utiliser. Ils peuvent donc échanger des informations directement entre eux quand ils se connectent.

Ainsi, les utilisateurs peuvent stocker les données des autres utilisateurs et diffuser les informations à travers le réseau social physique ou retarder le téléchargement des données jusqu'à ce que quelqu'un dispose d'une connectivité en ligne.

II.2. SAFEBOOK :

Safebook [31] a pour but la protection de la vie privée en se basant sur les relations de confiance de la vie réelle. C'est une nouvelle approche des réseaux sociaux en ligne qui selon Refik Molva et .al [32] est basée sur deux principales raisons :

- *« La décentralisation via une architecture peer-to-peer, afin d'éviter le contrôle de données des utilisateurs et leur comportement sur le réseau par une seule entité tel que le fournisseur du service.*

- *L'exploitation de la confiance réelle de la vie via la gestion de confiance et de protection de la vie privée des données des utilisateurs et les communications dans le système du réseau social en ligne en exploitant les relations de confiance à partir de ce même réseau social. »*

Sa conception répond à un large éventail d'exigences de sécurités dont la pertinence est recueillie à partir d'une série d'études, on distingue : la confidentialité de bout en bout, l'authentification, le contrôle d'accès, la vie privée, l'intégrité des données et la disponibilité des données.

a - CONFIDENTIALITE DE BOUT EN BOUT :

Elle a pour but de garantir qu'aucune autre partie en dehors des deux peers communicant ne pourra avoir accès aux données échangées en rendant également impossible l'écoute. Parce que dans les systèmes peer to peer les messages échangés par les peers peuvent contenir un contenu malveillant posté par une attaque de man in the middle [33]. Il est important d'édifier un centre spécial contre ces attaques qui peuvent facilement être monter dans un tel environnement.

b - L'AUTHENTIFICATION :

Une authentification particulière des membres est requise afin d'achever le contrôle d'approche. La politique « Fine-Grained Policy [34] » du contrôle d'accès basée sur les attributs du profile et les données privées peut être utilisée pour garantir la divulgation de données en fonctions de l'intégrité du requérant.

c - LA VIE PRIVEE :

La vie privée vise l'anonymat, la non traçabilité et l'incapacité à suivre des communications d'utilisateurs aussi bien que le respect de la confidentialité des informations personnelles vis-à-vis des intrus et du fournisseur du système.

d - L'INTEGRITE DES DONNEES :

L'intégrité des données vise à empêcher la falsification des données des profils car la propriété de disponibilité de ces données représente une exigence permettant une facilité principale d'utilisation. Cela garantit l'accès aux profils à tout moment ainsi que la

délivrance des messages à tout utilisateur dans les mêmes conditions en empêchant des attaques de déni de service [35] et les attaques Sybil [36].

Bien que Safebook respecte ce qui a été mentionné ci-dessus, la faisabilité d'une approche décentralisée en terme de disponibilité des données ainsi que la réactivité du système restent une question ouverte.

II.3. DECENT :

Decent [37] est un réseau social en ligne décentralisé, qui emploie une DHT pour stocker et récupérer des objets de données créés par leurs propriétaires. Chaque objet est crypté pour fournir la confidentialité. L'avantage principal de cette architecture est sa modularité, c'est-à-dire, les politiques d'accès, les mécanismes cryptographiques et la DHT sont trois composants séparés, interagissant l'un avec l'autre par des interfaces bien définies.

Pour la mise en œuvre du prototype, la conception modulaire fournit la capacité d'utiliser n'importe quel type de DHT et n'importe quel type de plan cryptographique. C'est une conception concernant les réseaux sociaux décentralisés mettant l'accent sur la sécurité et la vie privée.

Les utilisateurs de Decent utilisent un mécanisme cryptographique efficace pour la confidentialité, combinant des plans cryptographiques traditionnels et avancés pour l'intégrité et la disponibilité. La simulation et les expériences avec le prototype préliminaire de Decent montrent que le respect de la vie privée a été amélioré.

L'architecture Decent fournit la flexibilité pour la direction de données dans une conception orientée objet. Elle utilise un plan cryptographique approprié et avancé qui soutient une révocation d'accès efficace et des politiques « Fine-Grained Policy » sur chaque pièce de données.

Les autres architectures se concentrent seulement sur un ou deux des trois contraintes des réseaux sociaux distribués citées dans l'introduction. La nouveauté de cette architecture se trouve dans l'intégration d'existants primitifs qui sont adaptés pour améliorer la sécurité et la vie privée des réseaux sociaux en ligne. Ces existants primitifs sont entre autre la politique d'accès, le mécanisme de chiffrement et les algorithmes utilisés dans les DHTs.

II.4. CACHET :

Cachet [38] est une approche d'un réseau social distribué, basée sur Decent, qui donne une sécurité optimale ainsi que le respect de la vie privée. En particulier cachet protège la confidentialité, l'intégrité et la disponibilité des données des utilisateurs, aussi bien que l'intimité de leurs relations via le réseau.

Cachet utilise un réseau distribué de nœuds pour le stockage des données et assure la disponibilité de ces derniers. Mais ces nœuds ne sont pas dignes de confiance, donc, le niveau de cryptographie a été augmenté par rapport à Decent pour protéger les données. Le contenu des données des utilisateurs est stocké dans un groupe de nœuds distribués basé sur la DHT FreePastry.

Une cryptographie adaptée est utilisée pour le stockage des données dans les nœuds pour une authentique mise à jour des requêtes. Un cryptage basé sur les attributs des objets échangés dans le réseau est utilisé pour diminuer le temps de cryptage et de décryptage qui était important dans l'architecture de Decent.

Cachet est en réalité une évolution de Decent. Le résultat obtenu montre l'importance d'utiliser le réseau de Cachet, qui réduit la latence de la visualisation d'une page d'actualité de 100s en moins de 10s. Cette architecture démontre que la combinaison de plusieurs systèmes distribués et les techniques cryptographiques peuvent être utilisées pour avoir une alternative de protection de vie privée convaincante par rapport aux réseaux sociaux centralisés.

III. ARCHITECTURE DES RESEAUX SOCIAUX DISTRIBUES :

Nous avons étudié soigneusement les architectures et les composants de ces quatre réseaux sociaux, et nous les avons simplifié afin de les présenter comme suit :

III.1. ARCHITECTURE DE PEERSON :

Sonja et .al [39] ont définit une approche P2P distribuée couplée avec un système de chiffrement et une extension de l'approche décentralisée par l'échange directe des données entre les utilisateurs.

Pour valider cette approche, Doris Schiöberg [40] a conçu une architecture 2 tiers

avec des protocoles qui recréent les caractéristiques fondamentales des réseaux sociaux classiques de manière décentralisée.

a - 1ER TIERS : (LE LOOK UP SERVICE)

Il s'agit d'un système de consultation des DHT, qui stocke les métadonnées telles que : Les adresses IP, les informations sur les fichiers, des informations sur les utilisateurs, … Ces métadonnées sont nécessaires afin de pouvoir trouver les utilisateurs, leurs statuts et les données qu'ils stockent.

PeerSon utilise OpenDHT, donc si l'utilisateur est hors ligne, alors la DHT pourra stocker les données qu'elle reçoit pour une durée limitée. Plus de détails peuvent être retrouvés sur OpenDHT dans le chapitre I.

b - 2e tiers : (LES PEERS)

Ce tiers est constitué des peers et contient les données des utilisateurs. Chaque utilisateur qui se connecte au réseau contactera d'abord le lookup service pour le prévenir de sa connexion. Un utilisateur qui souhaite communiquer avec un autre de manière synchrone ou asynchrone interroge d'abord le lookup service pour obtenir toutes les informations nécessaires. Les peers se connectent directement entre eux après cela et interrompent leur lien de conversation directement après l'échange des messages asynchrones.

Une représentation simplifiée de l'architecture de PeerSon est montrée dans la figure II.1.

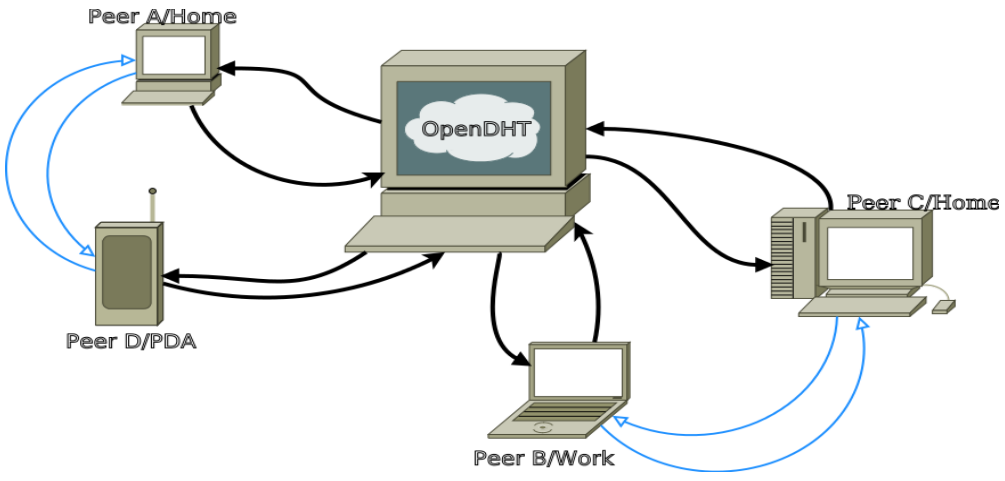


Figure II.1 : Architecture de PeerSon.

III.2. ARCHITECTURE DE SAFEBOOK :

Leucio Antonio et .al [41], ont proposé une nouvelle approche pour s'attaquer aux problèmes de sécurité et de confidentialité liés aux réseaux sociaux centralisés. Le prototype [42] de cette approche est écrit en python et peut être exécuté sur de multiples systèmes d'exploitation. Il est composé de quatre gestionnaires différents :

- Le gestionnaire de communication qui s'occupe de l'envoi et de la réception des paquets sur le réseau.
- Le gestionnaire de S2S qui s'appuie principalement sur la couche de la DHT.
- Le gestionnaire de la Matryoshka qui se base sur la couche du réseau social.
- Le gestionnaire de l'utilisateur qui implémente l'interface utilisateur.

Afin d'assurer la confidentialité des utilisateurs face à d'éventuelles violations de la vie privée par le fournisseur, l'approche proposée adopte une architecture 3tiers. Cette approche est décentralisée avec un mapping directe entre les trois couches qui constituent le réseau social, en s'appuyant sur la coopération entre un certain nombre de parties indépendantes qui sont également des utilisateurs de l'application de ce même réseau.

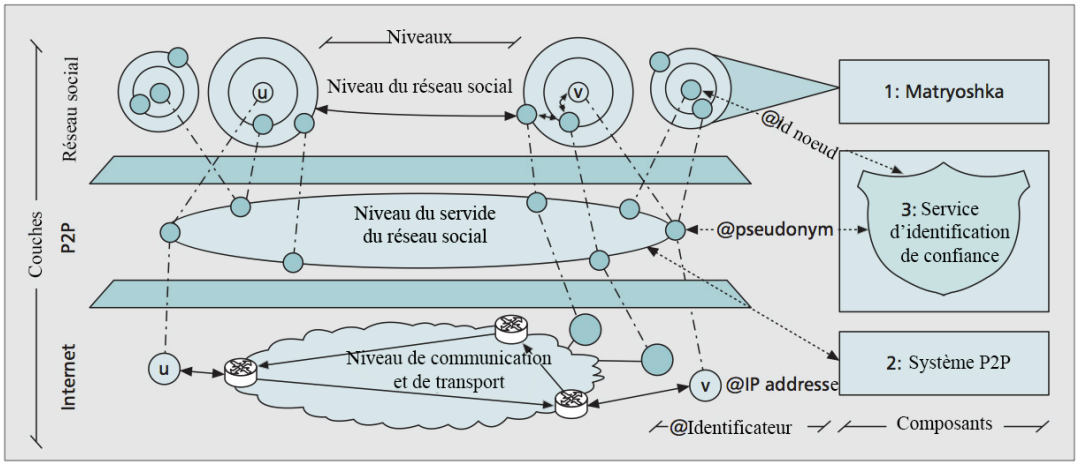


Figure II.2 : Couches de Safebook (à droite) et principaux composants (à gauche)

Les couches et principaux composants de Safebook sont représentés dans la figure II.2 [43] et détaillés comme suit :

- La couche du réseau social centrée sur l'utilisateur implémente le niveau du réseau social qui est la représentation numérique des membres et leurs relations dans les réseaux sociaux en ligne.

- Le service de management offert par le fournisseur du réseau social est implémenté dans la couche du support P2P afin de gérer l'infrastructure de l'application.
- Internet qui, elle, représente la couche de communication et de transport.

Dans Safebook, chaque partie est représentée par un nœud considéré comme un nœud hôte dans la couche Internet, un nœud peer dans la couche P2P et un membre dans la couche du réseau social. Ces nœuds forment deux types de couche :

a - MATRYOSHKA :

Un ensemble de Matryoshka est une structure concentrique dans la couche du réseau social, elle fournit le stockage des données et la confidentialité des communications autour de chaque nœud.

Les Matryoshkas sont construites autour de chaque nœud de membre afin de fournir un stockage de données fiable, une possibilité de récupération de données et un brouillage des communications.

Chaque Matryoshka, comme représentée dans la figure II.3 [43], protège le centre du nœud qui est appelé noyau, ce qui, dans la couche du réseau social est adressé par son identificateur de nœud. Ce noyau est encerclé par des nœuds. Ces nœuds sont reliés par des chemins radiaux sur lesquels les messages peuvent être relayés de manière récursive à partir de la couche externe vers le noyau et vice versa. Tous les chemins sont basés sur des relations de confiance apparentées au réseau social. Chaque saut connecte une paire de nœud appartenant à des utilisateurs liés par une relation de confiance dans la vie réelle.

Les nœuds qui sont les plus interne et, ceux les plus externe ont un rôle particulier. Le cercle le plus interne est composé des contacts directes du noyau, chacun de ses contacts stocke chez lui les données de ce noyau sous forme cryptée. Ces nœuds sont donc appelés miroir. Quand aux nœuds du cercle le plus externe, ils sont considérés comme une passerelle pour toutes les demandes adressées à ce noyau. Ils sont appelés des points d'entrées.

Toute demande vers un noyau, lui est adressée en utilisant son identificateur, qui peut être retrouvé dans la DHT. Cependant la communication en temps réel est transmise

par le noyau lui même, par contre les communications en mode hors ligne peuvent être desservies par un de ces miroirs.

Le nombre de miroirs et de points d'entrée dans chaque voie est fixé. Par contre, le nombre de nœud entre eux peut être variable ce qui conduit à des chemins de longueur variable dans la même Matryoshka.

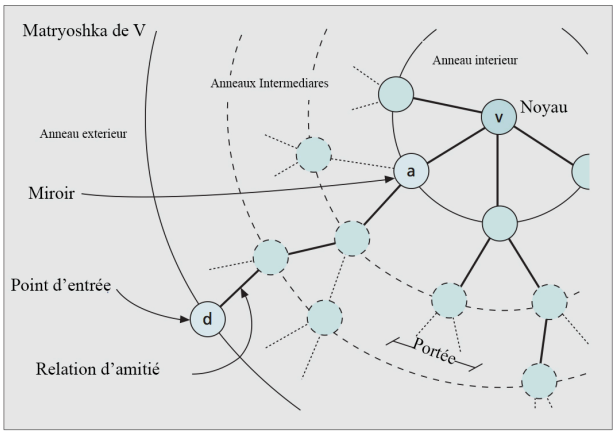


Figure II.3 : Structure d'une Matryoshka

b - LE SYSTEME P2P:

Afin de fournir un service de localisation pour trouver les points d'entrés vers une Matryoshka d'un utilisateur, les nœuds créent un support P2P. Actuellement ce support ressemble à KADMELIA cité dans le chapitre I ; ainsi les pseudonymes sont utilisés comme identifiants pour la DHT. Les clés de recherche qui sont enregistrées sont la propriété des membres participants ainsi que leurs identifiants de nœuds.

Cependant la communication à travers la couche P2P ne repose pas sur des liens de confiance contrairement à la voie à travers une Matryoshka. Toutefois, l'utilisation des pseudonymes permet la protection des membres contre la violation des droits de la vie privée fondée sur l'identification et le traçage des nœuds via des liens P2P non fiables.

c - SERVICE D'INDENTIFICATION DE CONFIANCE :

Ce service garantit que chaque utilisateur de Safebook obtient au plus un identifiant unique dans chaque catégorie d'identifiants. Il est basé sur une procédure d'authentification nommée out-of-band [44]. Ce service d'identification de confiance accorde à chaque utilisateur une paire unique d'identifiants de nœud et de pseudonyme.

Cette paire est une clé résultante du calcul d'une fonction de hachage sur l'ensemble des propriétés qui l'identifient de façon unique dans la vie réelle telles que : nom, prénom, date de naissance, lieu de naissance, …

Il est à noter que ce service d'identification de confiance est utilisé comme une troisième partie qui est centralisée et semble s'opposer au but de la décentralisation. Cependant, il ne constitue aucune menace à la vie privée car il ne peut pas tracer les utilisateurs ou leurs messages échangés.

III.3. ARCHITECTURE DE DECENT ET CACHET :

Decent comme montré dans la figure II.4 est basé sur une architecture 2 tiers où les données sont stockées comme des objets dans une DHT qui utilise un « object ID » comme clé en plus d'un standard « push and get operation » ; la DHT supporte aussi une opération additionnelle au stockage dans les nœuds avec la WritePolicy [45] sur les objets.

La disponibilité des données est assurée par leur réplication. Le lookup service utilise le protocole de la WritePolicy pour empêcher les utilisateurs malveillants de modifier ou supprimer les données disponibles dans la DHT parce qu'ils n'ont pas la bonne signature. Celle-ci est générée par le hachage de la clé publique de la WritePolicy et le cryptage des données. Elle fera partie de l'objet métadonnée également crypté. Donc le nœud de stockage refusera de réécrire l'objet stocké sauf si la nouvelle donnée est signée proprement par la clé WritePolicy.

La clé publique de la WritePolicy est aléatoire pour chaque objet afin d'empêcher la liaison d'un objet à son propriétaire.

Cependant, vu que Cachet est une évolution de Decent alors il se base bien évidemment sur son architecture mais en rajoutant un tiers, là ou dans Decent il n'y avait que les peers et les DHT, Cachet a rajouté des DHT pilotes comme montré dans la figure II.4. Ces DHTs contiennent l'annuaire globale du réseau, parce qu'elles ont connaissance de contacts appartenant à quelle DHT. Cela facilitera la recherche et optimisera le temps.

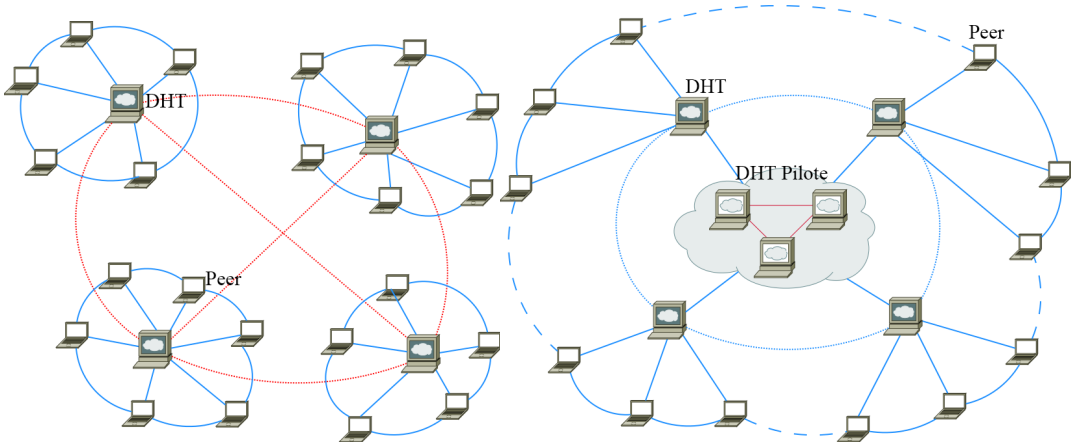


Figure II.4: Architecture de Decent à gauche et architecture de Cachet à droite

IV. PROTOTYPES DES RESEAUX SOCIAUX DISTRIBUES :

IV.1. PROTOCOLE DE PEERSON :

Cette section décrit comment l'architecture de PeerSon est implémentée et à quoi ressemblent les différents protocoles. Le détail de toute la syntaxe du protocole peut être retrouvé dans[14].

a - IDENTIFIANT GLOBALE UNIQUE (GUID) :

Le système du réseau social distribué doit résister à des attaques importantes comme les attaques Sybil et les dénis de services tout en s'assurant que les identités des utilisateurs sont uniques.

Cette résistance peut être facilement deployable dans un environnement centralisé, hormis le fait que dans un environnement décentralisé le déploiement est très difficile. PeerSon assume qu'aujourd'hui chaque utilisateur possède une adresse email qui est unique et donc utilisable comme identifiant. Mais pour empêcher les nœuds malicieux dans les DHT de collecter les adresses emails, un hash de chaque émail est calculé et utilisé comme identifiant unique.

b - LA PROCEDURE DE CONNEXION :

Se connecter au réseau signifie qu'un certain utilisateur annonce qu'il est actuellement en ligne avec les métadonnées nécessaires pour le joindre, ainsi qu'une liste des fichiers que ce peer stocke.

Cette annonce est envoyée à l'OpenDHT qui à son tour prévient les autres utilisateurs. C'est ce qui permet aux utilisateurs de suivre le statut de leurs amis.

c - LA RECUPERATION D'UN FICHIER:

Les messages sur PeerSon sont traités comme des fichiers. La découverte de ces fichiers est traitée par des requêtes envoyées à la DHT pour trouver quel peer contient le fichier recherché. Le résultat renvoi le nom du fichier si celui-ci existe, le numéro de la (les) version (s) existante(s) ainsi que le nom du peer contenant ce fichier. C'est l'utilisateur demandant qui décidera quelle version il souhaite obtenir et depuis quel peer.

d - LES MESSAGES ASYNCHRONES :

PeerSon permet les messages asynchrones tel que le mur de Facebook ou la messagerie instantanée même lorsque l'utilisateur est déconnecté. La réception en mode hors ligne est gérée par le stockage des messages au sein de la DHT jusqu'à ce que le récepteur se reconnecte.

IV.2. PROTOTYPE DE SAFEBOOK :

Safebook a implémenté différentes opérations des réseaux sociaux en ligne telles que : la création des comptes, la publication des données, la récupération des données, les requêtes de demande de contact et d'acceptation ainsi que la gestion des messages.

Lors de l'ajout d'un nouveau contact, une confiance particulière lui est attribuée de la part de celui qui l'ajoute. Ce niveau de confiance ne peut être connu que par l'utilisateur lui même et non pas par ses contacts. Donc ses contacts ne peuvent savoir dans quel cercle de confiance ils se trouvent. C'est ce niveau de confiance qui permet de déterminer qui de ces contacts va être utilisé comme miroir pour répliquer ses données.

Pour garantir la confidentialité dans Safebook, les données peuvent être privées, protégées ou publiques.

- Privées : les données ne sont pas publiées
- Protégées : les données sont publiées mais cryptées.
- Publiques : les données sont publiées.

Toutes les données publiées sont répliquées dans les miroirs de l'utilisateur ; cependant, dans Safebook si un utilisateur et tous ses miroirs sont hors ligne alors son contenu sera inaccessible.

IV.3. PROTOTYPE DE DECENT ET CACHET :

Decent et Cachet ont implémenté différentes parties dans leurs prototypes propre à eux et qui font leurs points forts tel que l' « ABE Policy » [46] et la « Cryptography Policy » [47], où ils sont passés de quelques centaines de seconde de cryptage et décryptage chez Decent à quelques dizaine de seconde seulement chez Cachet.

Pour leur prototype, Decent et Cachet ont développé un mur et une page d'actualité qui pour un utilisateur est une collection des derniers statuts mise à jour par chacun de ses contacts (amis) pour avoir une page d'actualité pour chaque utilisateur ;

L'ABE format contient la politique nécessaire pour le chiffrement des données, les utilisateurs peuvent savoir s'ils seront aptes à décrypter l'objets ou non et s'ils seront autorisés à lire les informations ou pas donc ils ne perdront pas de temps à décrypter des informations s'ils n'ont pas le droit de les lire.

Dans l'architecture de Decent, le décryptage des données nécessitait beaucoup de temps sans aucune performance de « caching » pour générer la page d'actualité. Par ailleurs, l'utilisateur n'avait pas besoin de lire toutes les actualités de ses contacts, une sélection devenait donc nécessaire, elle fut introduite dans l'architecture de Cachet.

De plus, Cachet adopte le « social caching », car étant donné l'utilisation de la décentralisation et la cryptographie dans l'architecture précédente qui est Decent, télécharger et reconstruire le mur d'un contact social ou la page d'actualité est un long processus exigeant les étapes suivantes :

- Décrypter des objets de mise à jour, qui sont ABEncrypted suivant la ABE Policy;
- Rapporter des métadonnées comme une mise à jour ;
- Indexer la clé de décryptage symétrique dans la DHT;
- Accéder à des petits objets multiples situés dans différents nœuds de stockage utilisant la DHT fournie dans l'étape précédente ;
- Décrypter des objets de mise à jour avec leurs clés symétriques correspondantes.

a - RECHERCHE D'UN CONTACT SOCIAL :

Pour permettre aux utilisateurs de rechercher leurs contacts, Cachet propose un service centralisé d'administration qui maintient le mapping entre les contacts et leurs racines (profile). Pour permettre à ce service de connaître les relations des utilisateurs, cette architecture utilise :

- Un canal de communication anonyme ;
- Afin de cacher les noms des utilisateurs dans l'annuaire de la DHT, il est nécessaire d'augmenter le niveau du protocole de sécurité concernant la vie privée.

Il est d'usage de croire que cachet garantit le respect de la vie privée et surpasse tous les autres systèmes existant dans ce domaine mais, beaucoup de changements afin de l'améliorer sont encore à mettre en place tels que:

- l'algorithme de social caching transmet des informations sur les identités des utilisateurs.
- le mur d'actualité révèle si un utilisateur est en ligne ou hors ligne.

b - SUPPRESSION ET ANNULATION :

Quand un utilisateur efface un objet ou modifie la politique d'accès à celui-ci, ces changements sont affectés immédiatement aux données. Le protocole de modification n'est pas spécifié dans ce travail, l'utilisateur peut donc envoyer une requête de révocation pour effacer des objets sur Cachet.

c - PROTOCOLE DE PRESENCE :

En générale dans un réseau centralisé le serveur garde la trace de la présence de ses utilisateurs. Cependant, dans Cachet une approche distribuée est appliquée quand chaque nœud enregistre sa présence dans la DHT afin que la présence des utilisateurs dans le réseau soit effective.

La présence de cet utilisateur en ligne a la même structure que les objets et est cryptée avec ABEncryption pour empêcher la DHT de lire le contenu de cet objet. Cette structure est définie par une adresse IP et un numéro de port. Ce qui permet à l'utilisateur d'effectuer des modifications à son profil en ajoutant ou supprimant des informations

quand il le désire. La DHT met à jours ces informations qui sont similaires au remplissage de la page d'actualité.

V. STOCKAGE DES DONNEES :

Pour le prototype actuel de PeerSon, la meilleure façon pour stocker les données est de le faire chez leurs propriétaires ainsi que sur l'espace de stockage de leurs amis en suivant l'algorithme Friendstore [48]. Dans ce cas là, ces données ne sont disponibles que lorsque le propriétaire et/ou ses amis sont en ligne. Ceux qui veulent les consulter doivent avoir les permissions nécessaires.

Quant à Safebook, les données sont stockées dans les machines appelées miroirs, c'est l'utilisateur qui attribue un certain niveau de confiance à chacun de ses contacts et c'est ce niveau de confiance qui va permettre de choisir les contacts chez qui il stockera ses données.

Cependant, Decent et Cachet utilisent un autre système de stockage, ils stockent les données dans des DHTs distribuées afin d'assurer une disponibilité permanente tout en augmentant le niveau de cryptage de données pour assurer l'intégrité et la confidentialité.

VI. DIAGRAMME DE SEQUENCE

Après avoir fait une analyse de l'architecture et du prototype de chacun des réseaux sélectionnés, nous avons réalisé des diagrammes simplifiés pour une meilleure compréhension des comportements de ces réseaux sociaux. Nous avons traité dans chacun d'eux quatre cas : création d'un nouveau profil, connexion au réseau, gestion des messages et l'ajout d'un nouveau contact.

VI.1. DIAGRAMME DE PEERSON :

Dans l'architecture de PeerSon il y a différentes parties :

Tout d'abord, PeerSon est réalisé de telle manière que nous puissions accéder au réseau social depuis différents appareils et divers endroits. Afin de savoir sur quel appareil l'utilisateur est disponible, l'équipe de PeerSon a défini trois modes de connexion, le

mode en ligne, le mode actif et le mode hors ligne. Par ailleurs il existe un point spécifique dans cette architecture :la possibilité d'accéder au réseau même en mode hors ligne.

- Le mode en ligne signifie que l'ordinateur de cet utilisateur se trouve connecté au réseau sans que l'utilisateur ne soit nécessairement présent.
- Le mode actif, définit que l'utilisateur est connecté et disponible sur le réseau.
- Le mode hors ligne est utilisé pour prévenir la DHT que cet utilisateur a quitté le réseau pour qu'à son tour elle puisse prévenir les autres utilisateurs.

a - CREATION D'UN NOUVEAU PROFIL :

Quand un nouvel utilisateur voudra s'inscrire sur le réseau PeerSon, il devra télécharger d'abord une application qui lui permettra l'utilisation de PeerSon et le stockage des fichiers.

Lors de la première utilisation, le nouvel utilisateur créera un profil comme montré dans la figure II.5, la DHT utilisera le hashage de son email pour générer un identifiant globale unique (GUID). L'utilisateur doit alors confirmer son GUID, et au final la DHT ajoutera alors les métadonnées nécessaires pour sa localisation et son statut. Il est à noter aussi que dans PeerSon l'utilisateur peut se connecter depuis plusieurs endroits, c'est la combinaison GUID/statut qui permet justement ce genre de connexion.

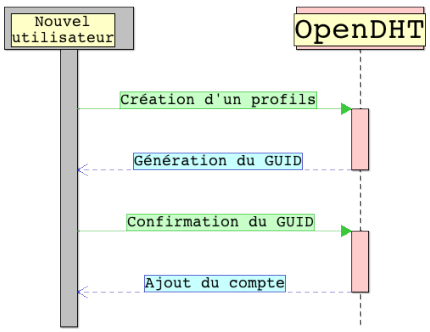


Figure II.5 : PeerSon – Inscription

b - CONNEXION AU RESEAU :

Quand un utilisateur rejoint le réseau, comme montré dans la figure II.6 il envoi une requête à la DHT afin de la prévenir qu'il vient de rejoindre ledit réseau. La DHT lui renvoie alors le statut des autres appareils depuis lesquels il a l'habitude de se connecter et si l'un d'entre eux est en mode actif, alors le mode en ligne se mettra sur cet appareil et le

mode actif apparaîtra sur l'appareil avec lequel il vient de se connecter. Si aucun appareil n'est en mode actif, alors ce dernier recevra ce mode par défaut.

Nous dénommons cette procédure par le mot login, ces trois étapes ne changent pas pour les autres utilisateurs lorsqu'ils se connectent au réseau.

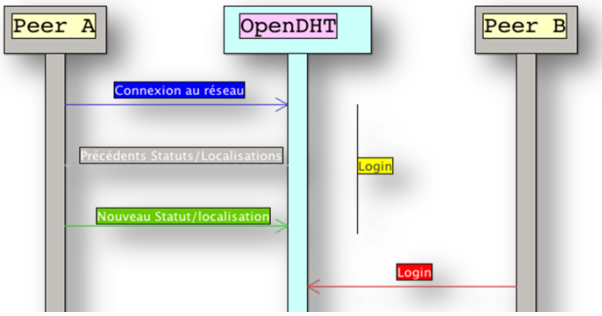


Figure II.6 : PeerSon – Connexion au réseau

c - GESTION DES MESSAGES :

PeerSon permet l'utilisation du réseau en mode en ligne et hors ligne, de plus en l'absence de l'utilisateur, son contenu peut être présent en fonction de ses amis. Cependant, avant qu'un utilisateur communique avec un autre il doit le localiser.

- **Localisation et statut du peer :**

Quand un peer veut communiquer avec un autre peer alors il doit demander à la DHT de localiser le peer désiré ainsi que son statut comme montré dans la figure II.7. La localisation permet de savoir sur combien d'appareils ce peer à l'habitude de se connecter. Le statut permet de définir tout d'abord si l'utilisateur est en ligne ou hors ligne. S'il est en ligne, le statut permet de vérifier le mode actif sur un appareil ou simplement le mode en ligne.

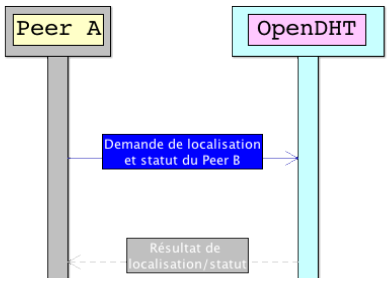


Figure II.7 : PeerSon – Localisation et statut du peer

- **Mode synchrone et asynchrone en ligne :**

Après le résultat de la localisation et le statut, si le peer recherché (dans notre cas c'est le peer B) est en ligne, alors que se soit pour la communication en mode synchrone (messagerie instantanée) ou asynchrone (publication sur le mur) la communication s'effectuera directement entre les peers (expéditeur et destinataire) comme montré dans la figure II.8. Cependant, il est à noter qu'une déconnexion immédiate se fera après l'échange des messages asynchrones.

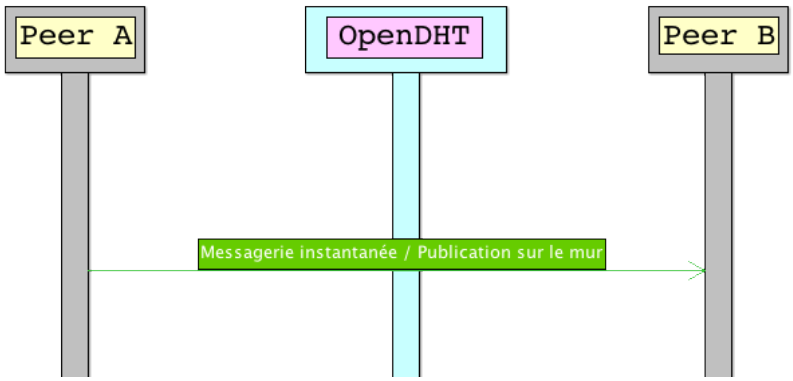


Figure II.8 : PeerSon – Mode synchrone et asynchrone en ligne

- **Mode asynchrone hors ligne :**

Quand un peer A annonce qu'il voudrait consulter le mur d'un peer B, la DHT vérifie s'il est en ligne ou hors ligne ; s'il est en ligne, il passe par l'étape que nous venons de voir à la figure II.8. Mais s'il est hors ligne comme dans la figure II.9, alors la DHT vérifiera d'abord si les amis du peer B qui stockent ses données sont en ligne, si oui, elle donnera l'adresse IP des peers contenant les données de B ainsi que la version disponible. Le peer A peut alors consulter le mur du peer B ou même effectuer des publications, commentaires, ….

Lors de la connexion ultérieure du peer B au réseau, il effectuera d'abord la requête login ensuite il demandera une actualisation de son profil. La DHT annoncera à ses amis qu'il s'est connecté et ceux qui ont la dernière version de son statut procèderont à son actualisation.

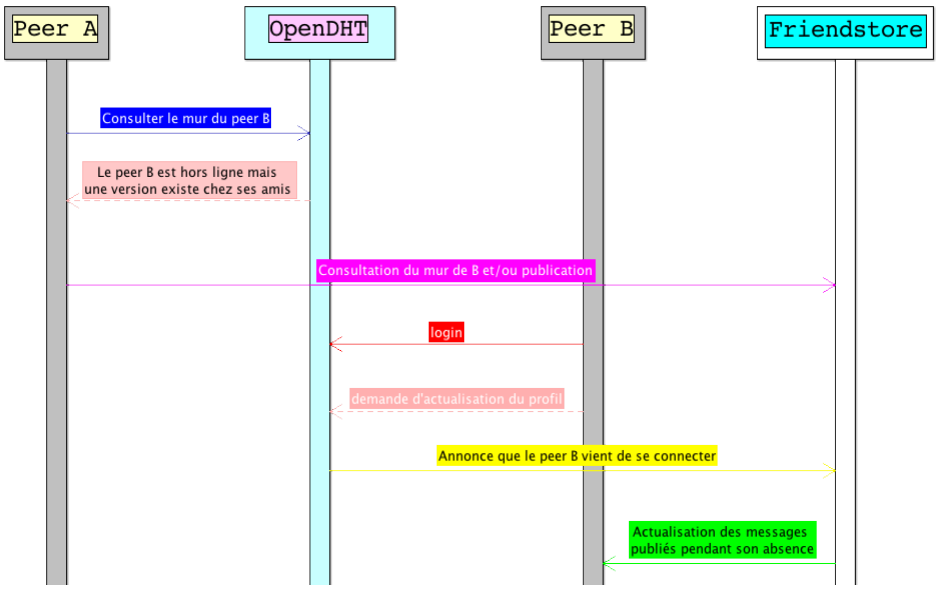


Figure II.9 : PeerSon – Mode asynchrone hors ligne

- **Mode synchrone hors ligne :**

Quand un peer A désire envoyer un message instantané au peer B et détecte que celui-ci n'est pas connecté, il peut lui laisser un message qui sera stocké dans la DHT comme montré dans la figure II.10, mais l'OpenDHT ayant quelques limites le message sera stocké pendant sept jours et ne devra pas dépasser 800 caractères. Toutefois quand le message est stocké sur la DHT, si le peer destinataire se connecte au réseau en utilisant la procédure login, puis demande une actualisation de son profil, sans avoir dépassé les sept jours requis, la DHT lui enverra une mise à jour de ce qu'il aura pu rater ainsi que les messages qui ont été stockés chez elle pendant son absence. Après cela le message sera effacé de la DHT et sera stocké chez son destinataire.

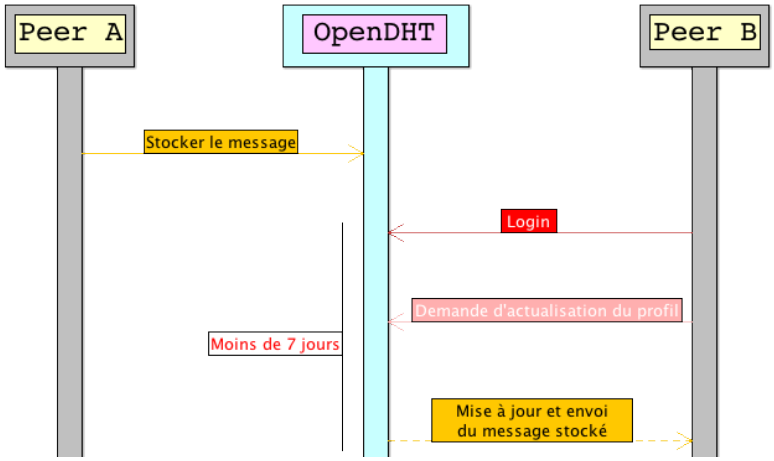


Figure II.10 : PeerSon – Mode synchrone hors ligne

d - AJOUT D'UN NOUVEAU CONTACT :

Dans PeerSon quand un utilisateur A veut ajouter un utilisateur B à sa liste d'ami, il enverra une requête à la DHT pour rechercher d'abord si ce dernier existe et si tel est le cas de le localiser comme montré dans la figure II.11. Une fois que le peer A aura reçu l'adresse du peer B, il lui envoi une demande d'ajout dans la DHT et c'est elle qui s'occupera de lui envoyer la requête de l'ajout lorsqu'il sera enligne, s'il accepte la demande alors la DHT mettra à jour leurs listes d'amis respectives. Il ne leur reste plus que d'échanger leurs clés de décryptage respectives et commencer à communiquer.

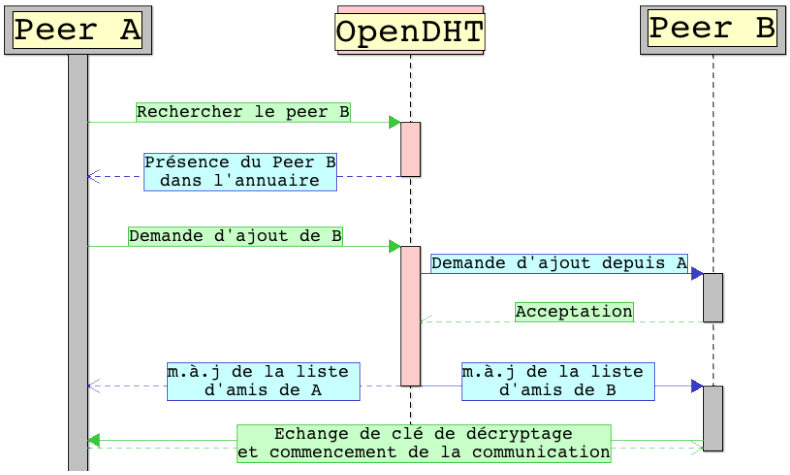


Figure II.11 : PeerSon – ajout d'un contact

VI.2. DIAGRAMME DE SAFEBOOK :

Après avoir fait une analyse de l'architecture et du prototype de Safebook, nous proposons des diagrammes simplifiés pour une meilleure compréhension du comportement de ce réseau social.

Tout d'abord nous allons voir les étapes de création d'un nouveau profil, ensuite nous verrons comment un utilisateur peut se connecter au réseau, nous aborderons par la suite la communication synchrone et asynchrone entre deux utilisateurs en mode en ligne et en mode hors ligne.

a - CREATION D'UN NOUVEAU PROFIL :

Pour rejoindre le réseau Safebook, un nouveau membre doit être invité par un de ses amis de la vie réelle déjà membre enregistré dans le réseau. Le compte de ce nouveau membre

est donc créé lors de deux étapes : création de l'identité et création de la Matryoshka. Nous avons simplifié ces étapes de création et nous vous les présentons sous forme de diagramme de séquence comme suit :

- **Création de l'identité :**

Nous avons supposé que l'utilisateur A est déjà enregistré sur le réseau Safebook. Cet utilisateur va alors envoyer une invitation par email à un de ses amis (amis, famille, …) de la vie réelle. Si cette personne décide de rejoindre le réseau, elle se connecte sur ce dernier et crée un profil. Le profil est créé dans la machine de l'utilisateur et référencé dans la DHT, cette dernière ajoutera les métadonnées du nouveau profil et préviendra le service d'identification de confiance (S.I.C). Le S.I.C va donc demander une fourniture de preuve pour s'assurer qu'il s'agit bien de cette personne en fournissant par exemple des informations telles que le nom, le prénom, la date et le lieu de naissance.

Cette preuve est présente sous forme de processus out-of-band et consiste à rajouter une pièce d'identité, un passeport ou encore une réunion en face à face entre le nouvel utilisateur et la représentation du S.I.C. Cette étape est primordiale car elle permet d'éviter les attaques d'usurpation d'identité et les attaques Sybil.

Après la vérification de la certitude des informations, le S.I.C calcule l'identifiant du nœud de ce membre qui sera unique dans le réseau. Une fois que l'utilisateur aura reçu son identifiant il pourra alors rejoindre le système P2P en utilisant le contact qu'il l'a invité comme un nœud d'amorçage et peut donc commencer la création de sa Matryoshka. La figure II.12 montre les étapes citées ci-dessus en un diagramme de séquence :

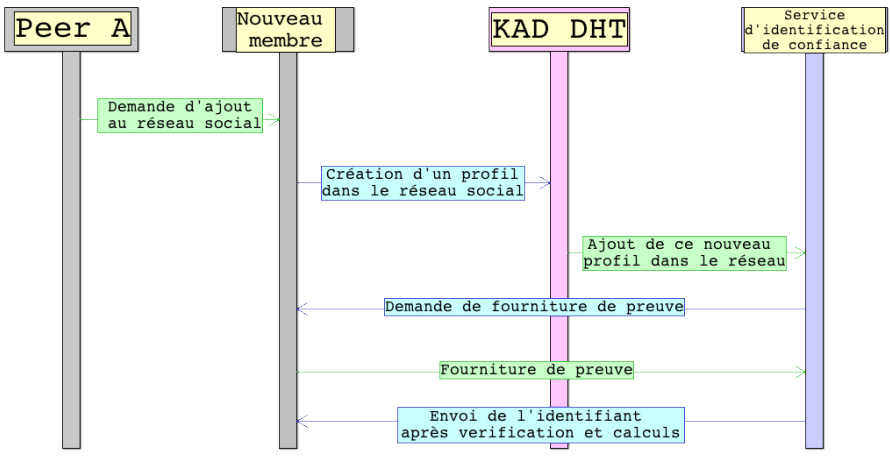


Figure II.12: Rejoindre Safebook - Création de l'identité

- **Création de la Matryoshka :**

Le nouveau membre a seulement son invitant comme contact pour commencer. Il lui envoi une requête pour créer le chemin qui contient les clés de la recherche qu'il souhaite enregistrer dans la DHT, le temps de vie de cette requête (TTL) et le nombre de membres à qui son invitant doit transmettre la demande. Ce nombre sera appelé facteur de portée par la suite.

L'invitant sélectionne un intervalle de portée des sauts suivants et leur transmet ce message d'enregistrement. Ce processus est répété de manière récursive jusqu'à ce que le TTL expire. A son expiration, le dernier nœud qui aura reçu le message d'enregistrement l'enregistre dans la DHT en utilisant son identifiant et son adresse à lui et commence à agir comme étant son premier point d'entrée.

La DHT renvoi alors à ce nouvel utilisateur la confirmation de la création de sa Matryoshka ainsi que l'adresse de son premier point d'entrée. Cependant parce que le facteur de portée peut être différent d'un chemin à l'autre, plusieurs points d'entrées peuvent être définis et ne pas appartenir à la même couche.

La figure II.13 montre les étapes de création de la Matryoshka avec un seul point d'entrée. Il s'agit de la même procédure pour les autres points d'entrées vers le même utilisateur.

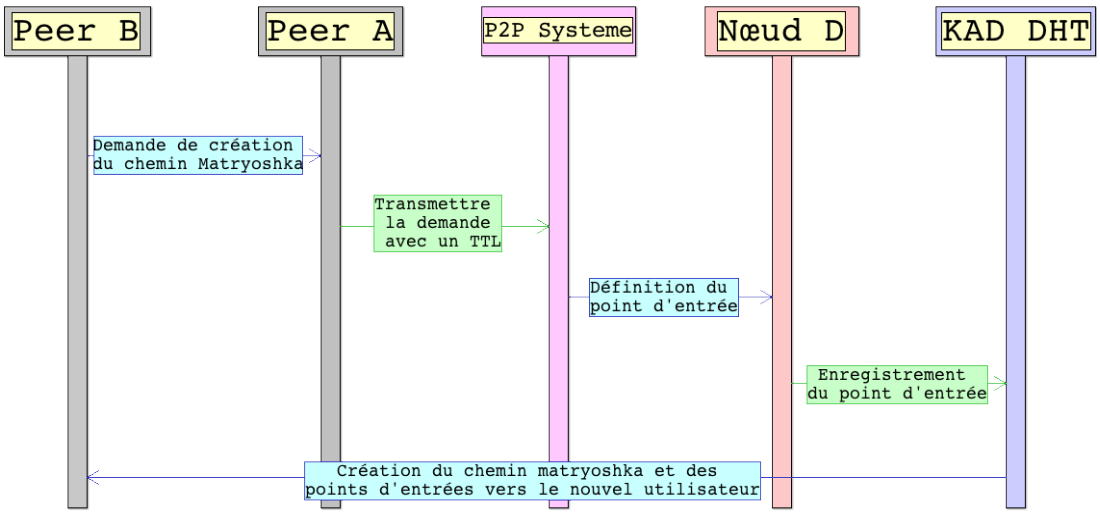


Figure II.13 : Rejoindre Safebook - Création de la Matryoshka

b - SE CONNECTER AU RESEAU :

Quand un utilisateur se connecte au réseau, il envoie une requête de connexion à la DHT, cette dernière diffuse l'information que le peer A s'est connecté sur le réseau P2P, ses amis qui se trouvent en ligne à ce moment vont être informés de sa connexion et si ses miroirs sont en ligne alors ils synchroniseront avec lui les données asynchrones publiées en son absence comme montré dans la figure II.14.

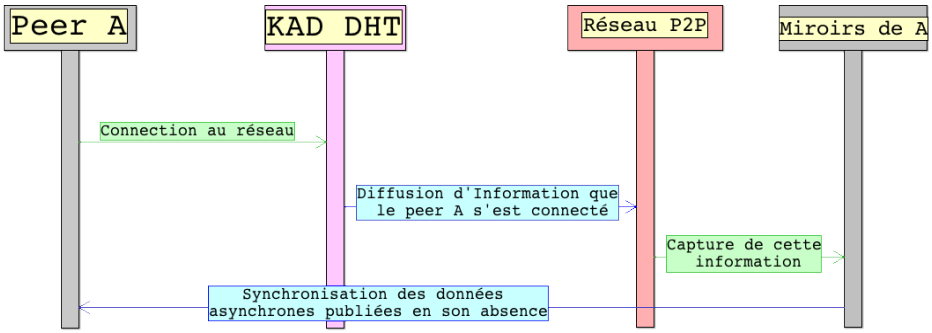


Figure II.14 : Safebook - se connecter au réseau

c - GESTION DES MESSAGES

En dehors de ses amis appelés miroirs, dans Safebook, chaque utilisateur qui veut consulter les données d'un autre utilisateur envoie d'abord une requête récursive dans le système P2P.

En fonction de la structure de la DHT, le nœud responsable de la clé de recherche répond avec une liste des points d'entrées qui constituent le cercle le plus externe vers cet utilisateur. Dans ce cas, l'utilisateur qui veut consulter les données d'un autre, pourra demander à un des points d'entrées de cet utilisateur de lui transmettre la requête à travers la Matryoshka.

Nous avons distingué quatre modes de délivrance de cette requête :

- **Mode asynchrone en ligne :**

Dans ce mode, la requête est envoyée dans le système P2P à travers la Matryoshka en saut par saut jusqu'à ce qu'elle arrive chez l'utilisateur ciblé. Ce dernier chiffre ses données avec la clé publique du demandeur et les renvoie à travers le chemin inverse comme montré dans la figure II.15.

Le protocole de Safebook utilise la récursivité pour cacher la source des demandes, et le cryptage pour s'assurer que seul l'émetteur de la requête est légitime à regarder ses données. Bien évidemment ces données sont aussi cryptées et signées par leur propriétaire et seul les membres qui auront une clé de décryptage pourront en voir le contenu. Les attaquants dans ce cas n'auront aucun moyen d'identifier une source de demande car, il n'est pas possible de faire la distinction entre les requêtes générées et transmises.

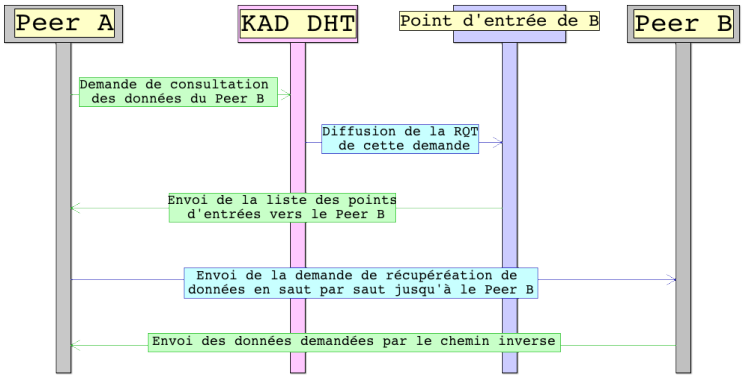


Figure II.15 : Safebook – mode asynchrone en ligne

Une étude de faisabilité préliminaire réalisée avec une approche précédente moins performante[7] a montré que la récupération de données fonctionne bien même si les messages sont acheminés le long de plusieurs sauts dans la Matryoshka.

- **Mode asynchrone hors ligne :**

Dans ce mode la requête est aussi envoyée dans le système P2P à travers la Matryoshka en saut par saut jusqu'à ce qu'elle arrive chez les amis miroirs de l'utilisateur demandé. Si un de ces miroirs est en ligne alors il chiffre les données demandées avec la clé publique du demandeur et les renvoie à travers le chemin inverse comme montré dans la figure II.16.

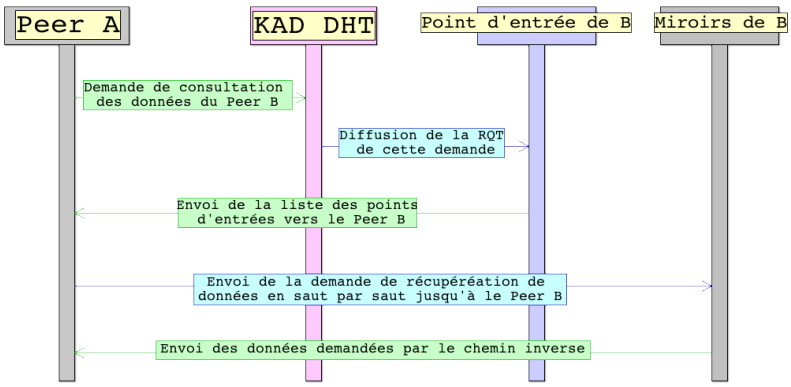


Figure II.16 : Safebook – mode asynchrone hors ligne

- **Mode synchrone en ligne :**

La messagerie instantanée est transmise et manipulée par le noyau et son interlocuteur uniquement. Comme la montre la figure II.17.

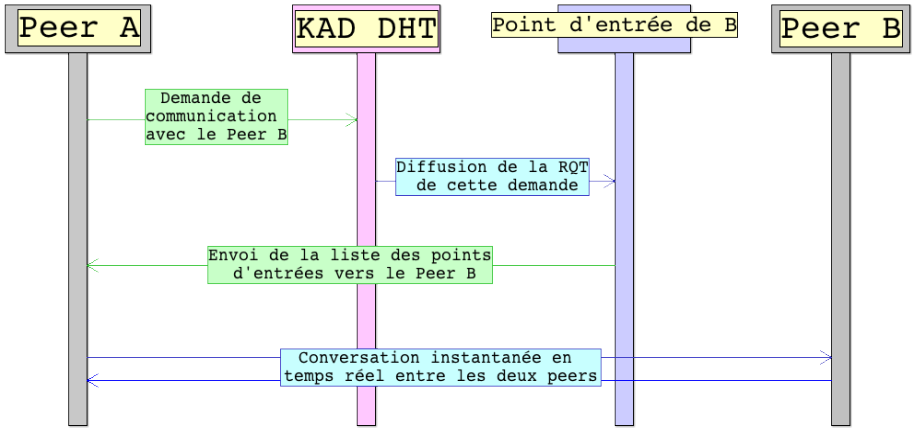


Figure II.17 : Safebook – mode synchrone en ligne

- **Mode synchrone hors ligne :**

Dans Safebook la messagerie instantanée n'est pas gérée dans le mode hors ligne, et si un utilisateur envoie ce genre de message à un autre utilisateur alors que ce dernier est déconnecté, un message d'erreur lui est envoyé par la DHT comme montré dans la figure II.18.

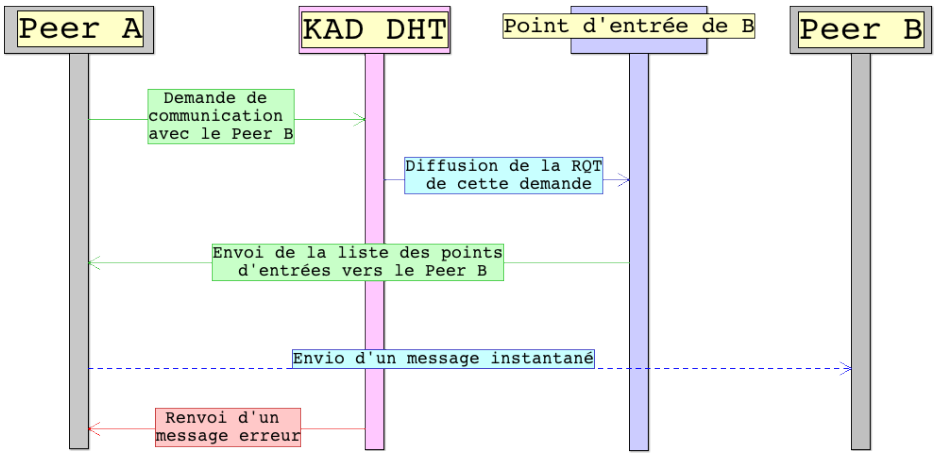


Figure II.18 : Safebook – mode synchrone hors ligne

Nous avons aussi constaté lors de notre étude que tout au long des échanges entre les utilisateurs, les données qui transitent sont toujours cryptées même si elles sont publiées sans être cryptées. Aussi il est à noter que l'utilisateur qui reçoit une requête de

récupération de ses données peut choisir entre la signer, la republier s'il s'agit d'une publication sur son mur ou l'ignorer.

Ainsi, dans Safebook, certains utilisateurs ont des privilèges appropriés et peuvent donc accéder et mettre à jour les profils des autres membres. Cependant, l'entité et l'authentification des données sont fournies par des signatures communes et des programmes de cryptage.

d - AJOUT D'UN NOUVEAU CONTACT :

Un membre A enregistré sur le réseau social qui veut ajouter un autre membre C aussi enregistré sur le réseau social dans sa liste de contact, envoie un message de demande de contact suivant les mêmes étapes que dans le cas de récupération de données.

Nous avons supposé que l'utilisateur C a accepté l'utilisateur A comme un nouveau contact. Le peer C associe avec le peer A un certain niveau de confiance qui n'est connu que par C lui même et personne d'autre. Il envoie au peer A une clé lui permettant de décrypter les données publiées et cryptées par C. La figure II.19 montre ces étapes en diagramme de séquence.

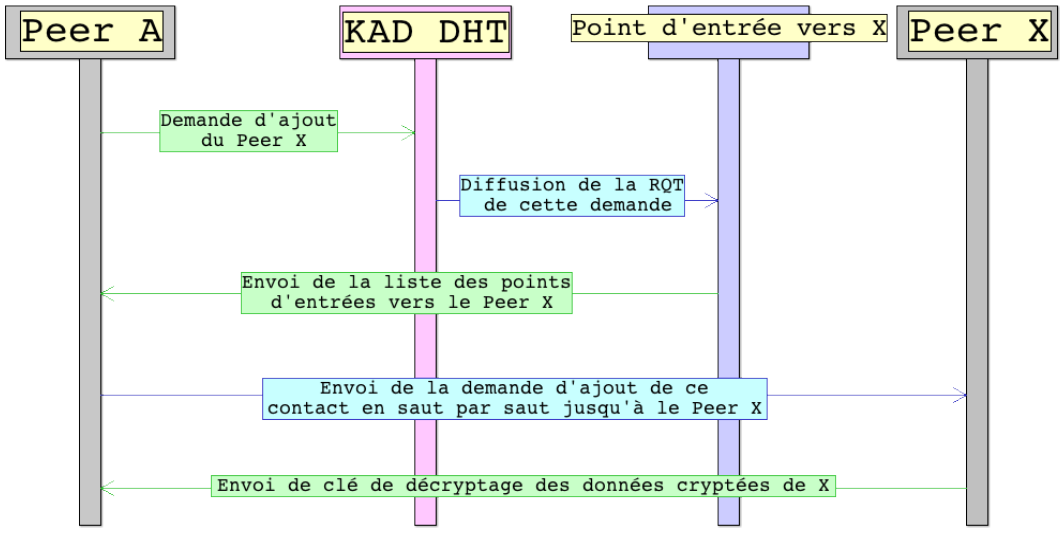


Figure II.19: Safebook – demande d'ajout d'un contact

VI.3. DIAGRAMME DE DECENT ET CACHET :

Parce que Cachet est une évolution de Decent, nous avons analysé le comportement de chaque réseau et nous vous les présentons sous forme de diagramme de séquence qui

comporte différente partie : inscription au réseau, connexion au réseau, gestion des messages et l'ajout d'un nouveau contact.

a - INSCRIPTION DANS LE RESEAU :

Dans Cachet comme dans Decent, la procédure d'inscription est la même comme le montre la figure II.20. Un nouvel utilisateur sur le site du réseau, créera un nouveau profil, la DHT lui demandera alors de définir son ABE Policy, quand c'est fait, la DHT ajoutera son profil sur le réseau et pourra donc commencer à l'utiliser.

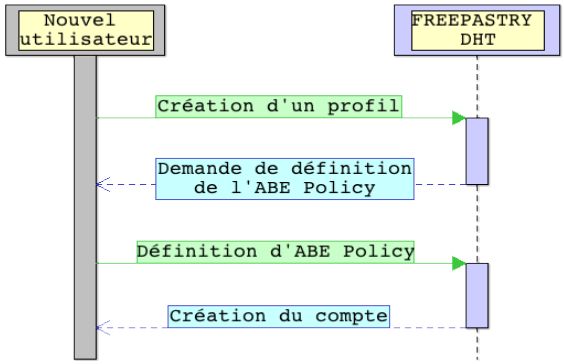


Figure II.20: Decent / Cachet – Inscription

b - SE CONNECTER AU RESEAU :

Quand un utilisateur A se connecte dans Decent comme montrée à la figure II.21, la DHT ou il stocke ses données propage l'information qu'il s'est connecté aux DHT voisines, toutes les DHTs font pareil jusqu'à ce que tout le réseau soit au courant, les DHTs qui contiennent l'annuaire de ses amis les notifieront. De son côté, l'utilisateur sera notifié des éventuelles publications le concernant effectuées en son absence.

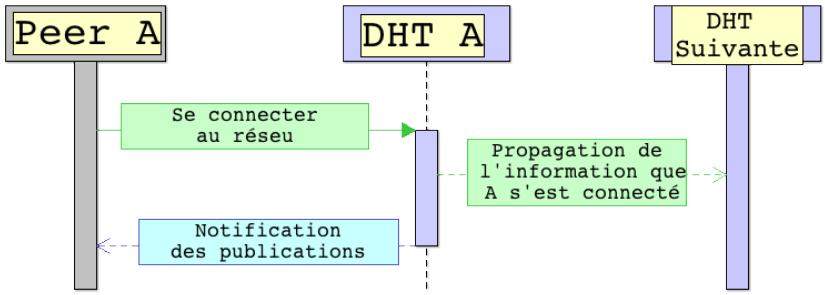


Figure II.21: Decent – se connecter

Toutefois, le principe de connexion sur Cachet diffère un peu de celui de Decent, en effet, quand un utilisateur se connecte comme montré dans la figure II.22, sa DHT notifiera seulement la DHT pilote la plus proche et c'est cette dernière qui s'occupera de propager l'information vers tout le réseau. Le reste du principe est le même que celui de Decent.

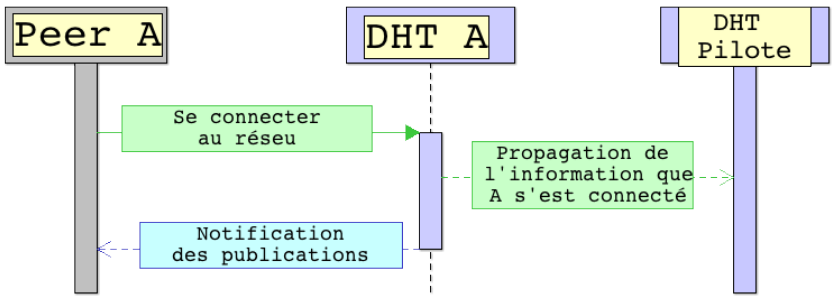


Figure II.22 : Cachet – se connecter

c - GESTION DES MESSAGES :

Decent et Cachet dans leurs travaux actuels n'ont pas encore géré la messagerie instantanée, leur travail consiste seulement aux échanges sur les murs et les pages d'actualités. Par ailleurs, le problème de disponibilité ne se pose plus car les données sont stockées dans les DHT et elles seront donc toujours disponibles.

D'une part, la consultation du mur d'un utilisateur dans Decent comme le montre la figure II.23 s'effectuera par une requête envoyée à la DHT où sont stockées les données du demandeur, cette DHT enverra une requête de recherche de la DHT qui contient le profil du mur demandé aux DHTs voisines. La requête de recherche est propagée dans le réseau d'une DHT à une autre jusqu'à ce que la DHT qui contient l'utilisateur recherché soit trouvée, une liaison directe entre ces deux DHTs s'établira.

S'il s'agit d'une consultation sur le mur d'un utilisateur B par exemple, la DHT B recherchera l'information demandé, si elle existe, elle la décryptera, vérifiera ensuite si le demandeur dispose de l'autorisation nécessaire pour voir le contenu en utilisant la clé de décryptage envoyée avec la demande. Si tel est le cas, elle donnera son accord à la DHT du peer qui a demandé la consultation, il pourra donc aller consulter directement dans la DHT B. S'il effectue un commentaire alors la DHT B notifiera l'utilisateur B du nouveau commentaire.

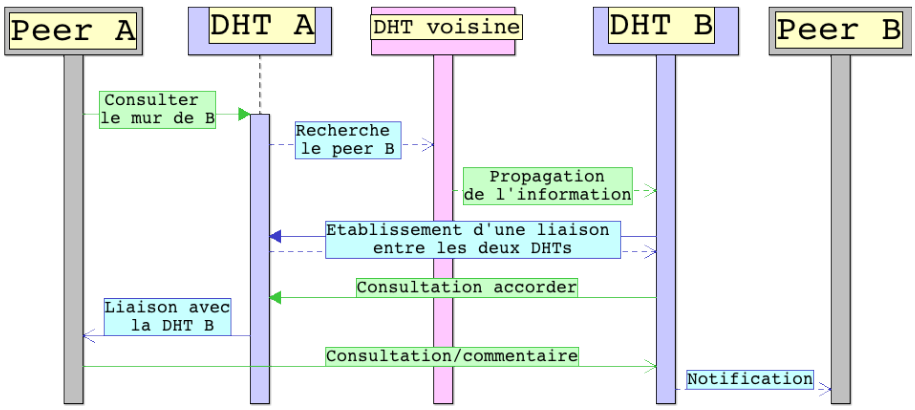


Figure II.23 : Decent – gestion des messages

D'une autre part, la consultation du mur d'un utilisateur dans Cachet a été améliorée, du coup, lorsqu'un utilisateur A veut consulter le contenu d'un utilisateur B, il demandera à sa DHT de trouver ou est stocké le contenu de B comme montré dans la figure II.24. La DHT A enverra cette requête de recherche à la DHT pilote qui contient dans son annuaire la liste des DHTs et les contacts de chaque DHT et fera une liaison directe entre la DHT A et la DHT B.

Une fois la liaison faite, la DHT A demandera à la DHT B l'autorisation de consultation pour son client en envoyant sa clé de décryptage. La DHT B vérifiera alors la clé de décryptage et si c'est la bonne alors elle décryptera les données et laissera l'utilisateur A consulter le mur de l'utilisateur B. S'il effectue un commentaire alors B en sera notifié.

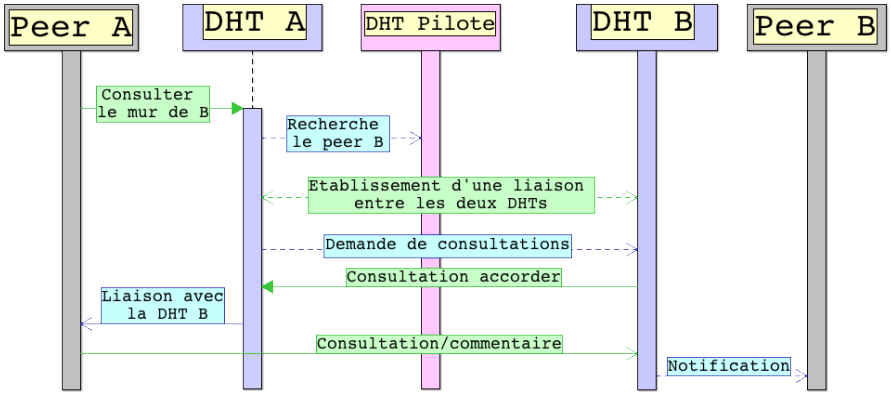


Figure II.24: Cachet – gestion des messages

d - L'AJOUT D'UN CONTACT :

Pour l'ajout d'un nouveau contact dans Decent, un utilisateur A demandera à sa DHT de rechercher ce contact, sa DHT propagera cette demande dans le réseau jusqu'à ce que cet

utilisateur soit retrouvé comme le montre la figure II.25, la DHT qui contient ce profil dans son annuaire informera la DHT du demandeur de la présence de cet utilisateur. Le peer A peut décider d'ajouter cet utilisateur, une demande d'ajout lui sera envoyée depuis la DHT A vers sa DHT qui le notifiera de cette demande quand il sera présent sur le site. S'il accepte la demande d'ajout, alors les DHTs mettront à jour leurs listes d'amis respectives dans leurs annuaires et les peers échangeront leurs clés de décryptages respectives. Il ne leur reste plus que de commencer à communiquer.

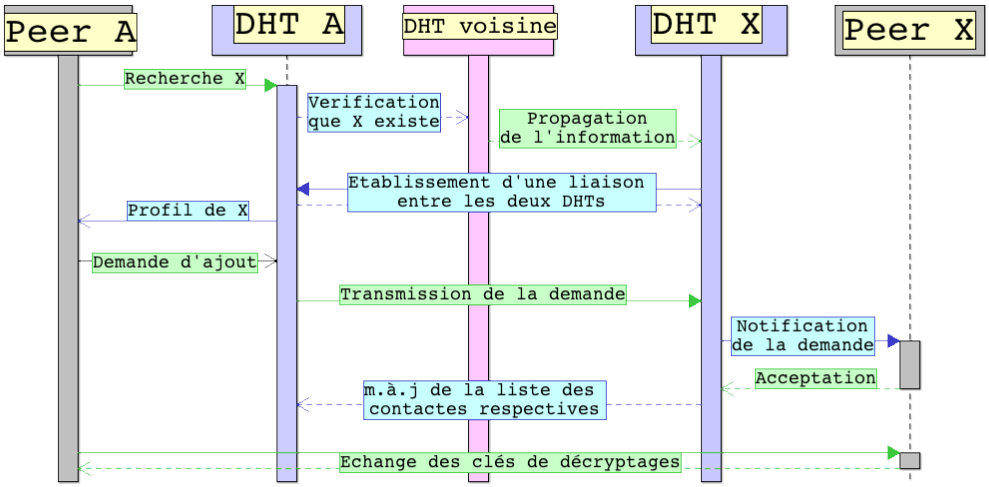


Figure II.25 : Decent – ajout d'un contact

En outre, l'ajout d'un nouveau contact dans Cachet a été amélioré comme le montre la figure II.26. En effet quand un utilisateur A veut rechercher un utilisateur X il enverra une requête de recherche à sa DHT, à son tour la DHT A transmettra cette requête à la DHT pilote qui pourra faire le lien directement et vérifiera si cet utilisateur existe, si tel est le cas elle établira une liaison entre les deux DHTs. Le peer A peut donc choisir à ajouter le peer X, les étapes qui restent sont les mêmes que pour Decent.

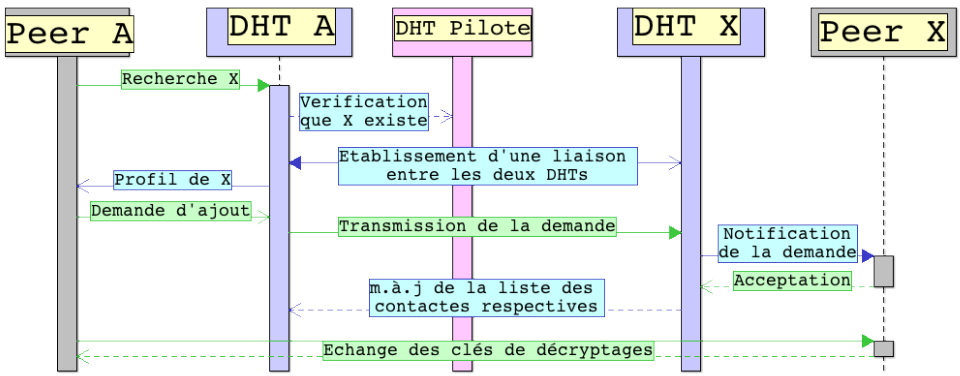


Figure II.26: Cachet – Ajout d'un contact

VII. SYNTHESE :

Lors de l'analyse de PeerSon nous avons constaté que, en cas de déconnection soudaine, la DHT ne pourra plus savoir quel est le mode de connexion de cet utilisateur. De plus s'inscrire sur un réseau social en ligne aujourd'hui est possible, se désinscrire n'est pas possible et cela quel que soit le type de réseau social en ligne. Seule la désactivation des profils est permise. Les réseaux sociaux distribués ont justement pour but principal la protection des données des utilisateurs, mais jusqu'à présent la suppression des données des utilisateurs qui se désinscrivent n'a pas encore été gérée.

L'une des plus grandes limitations de l'opendht est le fait qu'elle ne peut stocker les informations chez elle que pendant une durée limitée et que les utilisateurs doivent se connecter régulièrement. PeerSon quant à lui, permet les messages asynchrones même en mode hors ligne, en stockant sur la DHT les messages envoyés jusqu'à ce que le receveur se connecte, mais bien sur à condition que ce dernier se connecte avant une semaine, car sinon le message qui lui aura été envoyé sera tout simplement effacé de la DHT. L'expéditeur n'aura donc aucun accusé de réception du message envoyé. Ainsi le récepteur sera porté disparu, son GUID sera effacé de la DHT et ne pourra plus être retrouvé. Seules ses données peuvent l'être si elles ont été stockées chez ses amis.

Quant à Safebook le premier manque que nous avons constaté est le fait de ne pas pouvoir se connecter au réseau que depuis un seul terminal, Il s'agit du premier terminal avec qui l'utilisateur s'est enregistré au réseau la première fois.

La messagerie instantanée quant à elle n'est gérée qu'en mode en ligne, donc si un utilisateur veut laisser un message à un autre utilisateur déconnecté, il ne pourra pas le transmettre car les messages synchrones ne sont enregistrés que sur la machine de l'utilisateur et non pas sur les autres machines miroirs.

Nous avons aussi remarqué que pour contacter un utilisateur de Safebook, en dehors de ses amis miroirs, tout autre utilisateur devra passer impérativement par l'un des points d'entrée vers un utilisateur A par exemple pour le contacter. Sachant que ces points d'entrée sont aussi des utilisateurs enregistrés dans le réseau social, alors si à un moment donné, tous ces utilisateurs/points d'entrée sont déconnectés ,l'utilisateur A ne pourra pas être retrouvé et donc ne pourra pas être contacté.

Si un utilisateur et tous ses amis miroirs sont hors ligne, alors son profil ne sera pas disponible et donc Safebook n'assure pas une disponibilité totale.

L'approche de Decent et de Cachet quant à elle assure une disponibilité totale par le stockage des données dans des DHT, mais nous revenons à notre problème de départ, le regroupement des données chez un seul fournisseur et même si c'est données sont cryptées et distribuées dans des DHTs, il ne faut pas oublier que ces DHTs ne forment en réalité qu'une seule entité.

Par ailleurs, Decent et Cachet ont misé beaucoup plus sur le cryptage des données, ils l'ont certes levé pour gagner en sécurité mais ils ont beaucoup perdu en terme de temps de chiffrement et déchiffrement, ce qui empêche malheureusement l'utilisateur d'être en mesure d'utiliser le système beaucoup plus rapidement.

Le tableau ci-dessous représente un comparatif entre les quatre réseaux sociaux étudiés. Les critères de comparaison que nous avons sélectionné sont les types d'architectures, le type du client, les DHTs et leurs services, le mode de connexion et le mode de gestion de la messagerie. Ces critères nous ont permis de proposer notre modèle détaillé dans le Chapitre III.

Réseaux sociaux / *Caractéristiques*	***PeerSon***	***Safebook***	***Decent***	***Cachet***
Architecture	2 tiers	3 tiers	2 tiers	3 tiers
Type du client	Client lourd		Client léger	
DHT	OpenDHT	Kademlia	FreePastry	
Capacité de stockage DHT	Moyenne	Petite	Grande	
Service DHT	- Lookup service - Routage - Stockage pendant une durée de 7 jours	- Lookup service - Routage	- Lookup service - Routage - Stockage des données	
Mode en ligne	Oui	Oui	Oui	
Mode hors ligne	Oui si Friendstore en ligne	Oui si miroir en ligne	Oui toujours	
Messagerie instantanée	Oui : en ligne et hors ligne	Oui : en ligne seulement	Non : la messagerie instantanée n'est pas encore gérée	
Mode connexion	Avec ou sans Internet	Connexion Internet requise		

CHAPITRE III

PROPOSITION DISNET

I. OBJECTIFS DE DISNET

Le but principal de notre mémoire est la gestion du stockage des données dans les réseaux sociaux distribuées. Pour cela nous avons pris le meilleur de chaque approche vue dans la Chapitre II soit PeerSon, Safebook, Decent et Cachet pour proposer un nouveau modèle de réseaux sociaux distribués que nous avons baptisé DisNet.

Notre modèle consiste non seulement à apporter des solutions à quelques problèmes rencontrés par ces approches mais aussi à un problème qui n'a pas encore été ajouté à aucun réseau social et n'a pas figuré non plus dans les approches étudiées dans le chapitre II, soit la suppression d'un compte utilisateur.

Afin de valider notre modèle architectural DisNet, nous avons développé un prototype intitulé DisNet Prot que nous allons présenter dans ce chapitre. DisNet Prot a pour but de montrer quelques étapes du fonctionnement du réseau DisNet.

II. ARCHITECTURE DE DISNET :

Notre proposition adopte la solution du stockage distribué chez les utilisateurs en se basant sur l'algorithme de Friendstore qui est adopté aussi par PeerSon. Nous avons trouvé que cet algorithme est abouti et répond au besoin de notre proposition.

Par ailleurs, nous allons assurer une disponibilité optimale des données mais pas totale. Car nous ne voulions pas tomber dans le problème de regroupement des données chez un seul fournisseur comme le cas de Decent et Cachet.

L'intégrité des données quant à elle est assurée par le protocole d'authentification out-of-band inspiré de Safebook et qui permet de protéger les données ainsi que l'identité de l'utilisateur contre les modifications non autorisées et les falsifications.

Au final, la protection de la vie privée de l'utilisateur et la confidentialité de ses informations sont assurées par un système de chiffrement afin de ne permettre la consultation de ces données que par l'utilisateur lui même et les contacts avec qui il les partage.

En outre, notre proposition se base sur une architecture 2 tiers montrée dans la figure III.1, semblable à celle de PeerSon.

II.1. LE 1ER TIERS :

Il s'agit du service de consultation des DHTs ou sont stockées les métadonnées nécessaires pour trouver les utilisateurs, leurs statuts ainsi que les données qu'ils stockent. La DHT que nous avons décidé d'utiliser est FreePastry, donc, la même que celle utilisée par Decent et Cachet. Cette DHT permet de contourner le problème de stockage de données pendant seulement sept jours rencontré dans OpenDHT qui est la DHT utilisée par PeerSon.

De plus, FreePastry utilise plus de 100 nœuds répartis dans le monde contrairement à OpenDHT qui n'utilise que des nœuds répartis au sein du laboratoire Planet-lab aux usa.

II.2. LE 2E TIERS :

Ce sont les machines des utilisateurs. Les utilisateurs qui souhaitent rejoindre le réseau DisNet devront télécharger une application que nous allons mettre dans notre site pour pouvoir l'utiliser. Grâce à cette application ils pourront se connecter au réseau mais aussi

stocker leurs données et les données de leurs amis sélectionnés par le protocole Friendstore. Grâce à ce protocole, notre application permettra de sécuriser les données en les cryptant mais aussi permettra la propagation et la réplication des données stockées sur au moins trois machines. Des détails supplémentaires pourront être retrouvés au chapitre I.

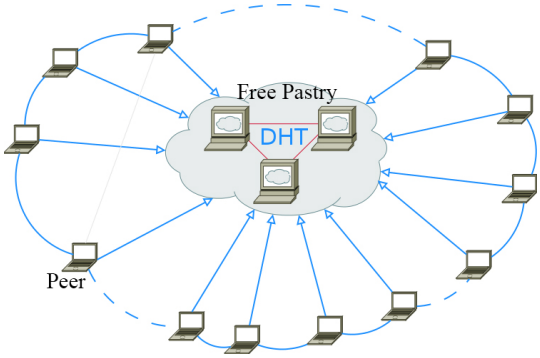


Figure III.1 : Architecture de DisNet

III. PROTOTYPE DE DISNET (DISNET PROT)

Dans cette section nous allons présenter les différentes parties que nous avons implémentées dans DisNet Prot ainsi que les améliorations abordées dans nos objectifs.

III.1. REJOINDRE DISNET :

Lors de la première utilisation de DisNet, le nouvel utilisateur devra s'inscrire. Pour une meilleure convivialité au sein du réseau, nous avons décidé de rajouter quelques contraintes pour prouver qu'il s'agit d'une personne réelle et qu'elle n'a pas usurpé l'identité de quelqu'un d'autre.

Lorsqu'un nouvel utilisateur aura rempli les champs de renseignements demandés tels que le nom, prénom, adresse email, date et lieu de naissance, et aura cliqué sur le bouton « créer un compte » la DHT lui demandera alors d'identifier le parrain. S'il dispose d'un parrain enregistré dans le réseau comme montré dans la figure III.2, il l'identifiera. La DHT vérifiera alors si ce parrain connaît l'utilisateur qui vient de demander son parrainage et s'il souhaite le parrainer. Si ce dernier confirme son parrainage, la DHT créera alors les métadonnées nécessaires pour ce nouvel utilisateur et l'ajoutera à l'annuaire des utilisateurs enregistrés. L'utilisateur pourra donc commencer à utiliser le réseau.

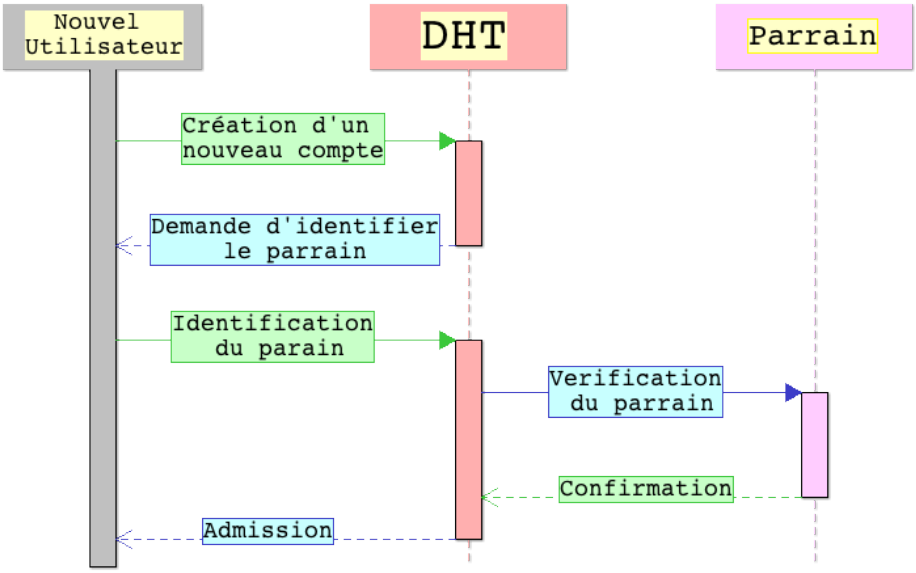


Figure III.2 : Rejoindre DisNet – Présence du parrain

Cependant, si lors de l'inscription l'utilisateur n'a pas de parrain dans le réseau comme dans la figure III.3, la DHT lui propose un défit de « challenge/réponse » définit pas le protocole d'authentification out-of-band, s'il le réussit alors la DHT validera son profil.

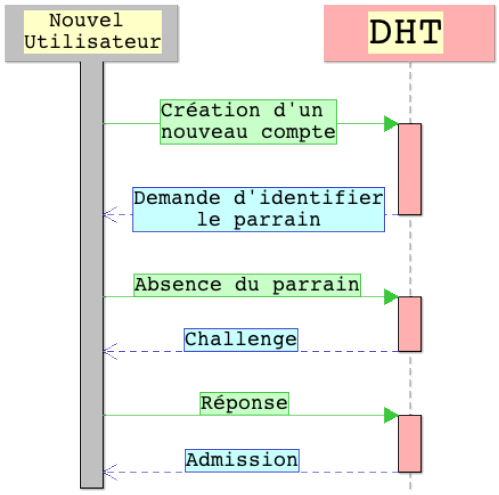


Figure III.3 : Rejoindre DisNet – Challenge/réponse

III.2. SE CONNECTER A DISNET :

Quand un utilisateur se connecte à DisNet comme dans la figure III.4, la DHT enverra une requête d'actualisation à ses amis qui stockent ses données pour les inviter à synchroniser

avec lui les messages asynchrones publiés en son absence. De son coté, l'utilisateur recevra une requête pour voir les amis qui sont en ligne à ce moment.

Par ailleurs la DHT effectuera une autre mise à jour et diffusera sur le réseau que cet utilisateur vient de se connecter, Ses amis qui se trouvent en ligne à ce moment recevront une notification les prévenant de sa connexion.

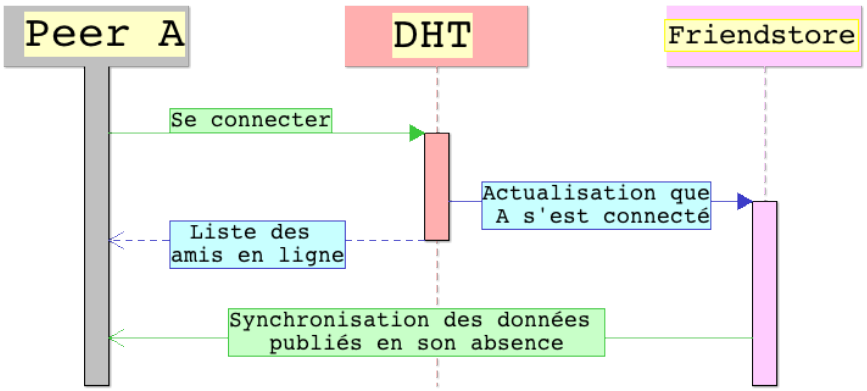


Figure III.4 : DisNet – se connecter

III.3. GESTION DES MESSAGES :

Si un utilisateur A veut communiquer avec un autre utilisateur B, il envoie d'abord une requête à la DHT pour lui demander l'adresse IP du peer B, si ce dernier est en ligne et que le peer A est habilité à le contacter, alors la DHT établira un lien de communication directe entre eux comme montré dans la figure III.5, les peers peuvent donc communiquer par des messages instantanés, une publication. un commentaire sur le mur. …

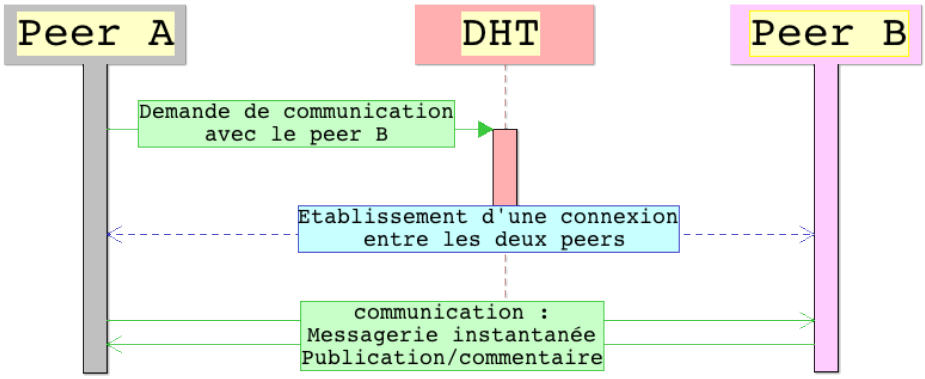


Figure III.5 : DisNet – Communication en ligne

Toutefois, si le peer A veut envoyer un message instantané au peer B et découvre que ce dernier est hors ligne comme dans la figure III.6, le message qu'il enverra sera

stocké dans la DHT jusqu'à ce que le peer B se reconnecte au réseau. Le message stocké lui sera retransmis, effacé de la DHT et le peer A sera acquitté de la transmission de son message.

Nous avons préféré un stockage des messages instantanés envoyés en mode hors ligne dans la DHT pour que l'information reste fraîche. Il devient évident que dès que le peer B se reconnectera à DisNet et même en l'absence de ses Friendstores, il recevra le message qui lui a été envoyé de la part du peer A en son absence directement depuis sa DHT.

Ainsi nous avons rajouter une requête qui sera envoyée au peer A si jamais le message qui a été envoyé n'a pas encore été lu en moins d'une semaine ; comme cela l'information restera fraiche et A pourra donc choisir entre garder le message dans la DHT ou l'effacer s'il n'est plus d'actualité.

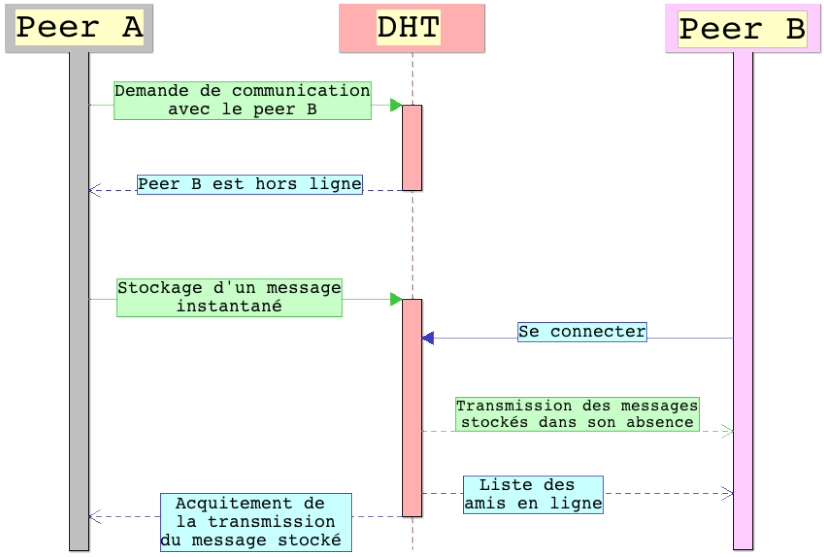


Figure III.6 : DisNet – Message instantané hors ligne

Par ailleurs, si le peer A veut consulter le mur du peer B alors que ce dernier est hors ligne, il demandera à la DHT si le contenu de B est tout de même disponible, la DHT vérifiera d'abord si A est habilité à consulter les données de B et si un des Friendstores de B est en ligne, si tel est le cas alors elle établira un canal de communication comme montré dans la figure III.7. A pourra donc consulter le profil de B, publier un statut ou un commentaire sur le mur de B via la Friendstore. Ce dernier recevra une synchronisation de la mise à jour de son mur dès la prochaine connexion en même temps que ses amis Friendstore.

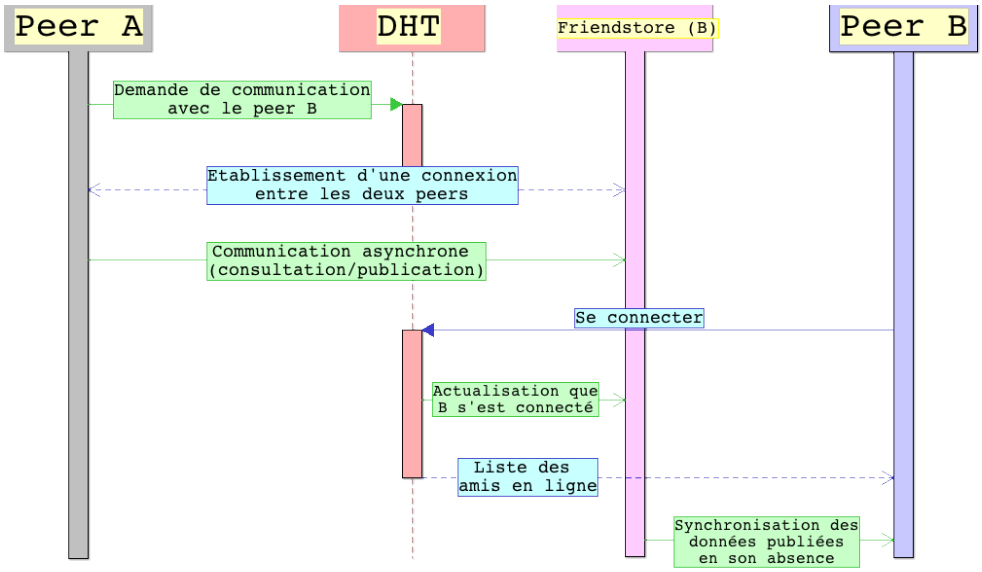


Figure III.7 : DisNet – communication asynchrone – Friendstore

III.4. AJOUT D'UN CONTACT :

Quand un utilisateur A voudra ajouter un utilisateur X à sa liste d'amis, il demandera à la DHT de rechercher dans son annuaire si cet utilisateur existe, si tel est le cas, la DHT enverra une demande d'ajout au peer X. Dans le cas d'acceptation comme montré dans la figure III.8, la DHT effectuera une mise à jour des amis de A dans son annuaire en ajoutant X et une autre mise à jour des amis de X en ajoutant A. Les nouveaux amis échangeront ensuite leurs clés de décryptage respectives pour décrypter les données qu'ils cryptent selon l'ABE Policy adopté aussi par Decent et Cachet.

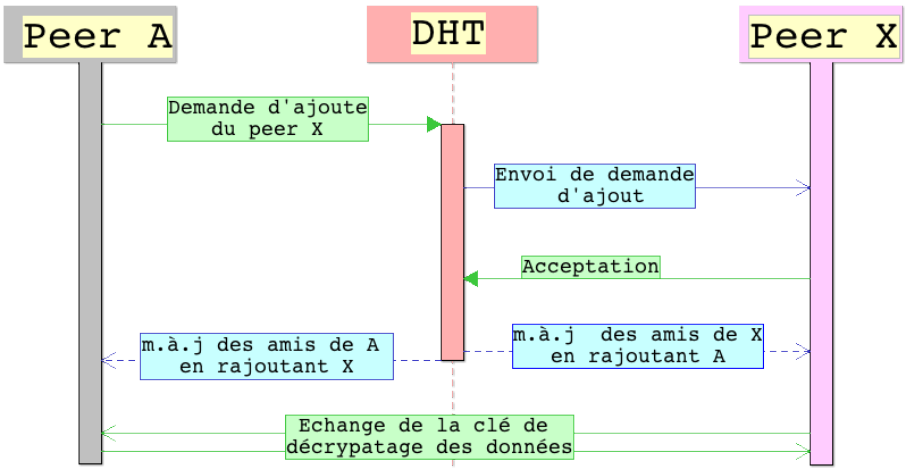


Figure III.8 : DisNet – ajout d'un contact

III.5. SUPPRESSION D'UN COMPTE ET DESTRUCTION DES DONNEES

Comme dernier cas, nous avons rajouté une nouvelle solution qui n'a été traitée par aucun réseau social ni aucune des approches vues dans le chapitre II jusqu'à présent et qui consistent à la suppression d'un compte utilisateur et la destruction de ses données avec possibilité de récupération de son profil.

Quand un utilisateur veut supprimer son compte définitivement de DisNet comme montré dans la figure III.9, il enverra une requête de suppression du compte à la DHT. Après confirmation de la requête pour s'assurer qu'il ne s'agit pas d'une erreur de sa part, la DHT s'occupera tout d'abord par propager la requête de destruction de ses données.

Des la réception de la requête par les machines des Friendstore de cet utilisateur, les données commenceront à se détruire jusqu'à ce qu'il ne reste aucune trace de cet utilisateur. Quand tout cela est fait, la DHT enlèvera de son annuaire les relations que cet utilisateur avait, les métadonnées qui étaient nécessaires à sa connexion, lui enverra un fichier nommer « DisNet Resuc » et rompra le contact avec lui. Les données ce cet utilisateur resteront stockées dans sa machine.

Par ailleurs, si cet utilisateur veut s'inscrire de nouveau dans le réseau et récupérer son profil, il utilisera le fichier DisNet Resuc pour cela. Ce fichier permet à la DHT de rétablir le contact avec lui car il contient les métadonnées autrefois nécessaires à sa connexion. L'utilisateur pourra également rétablir le lien avec ses amis.

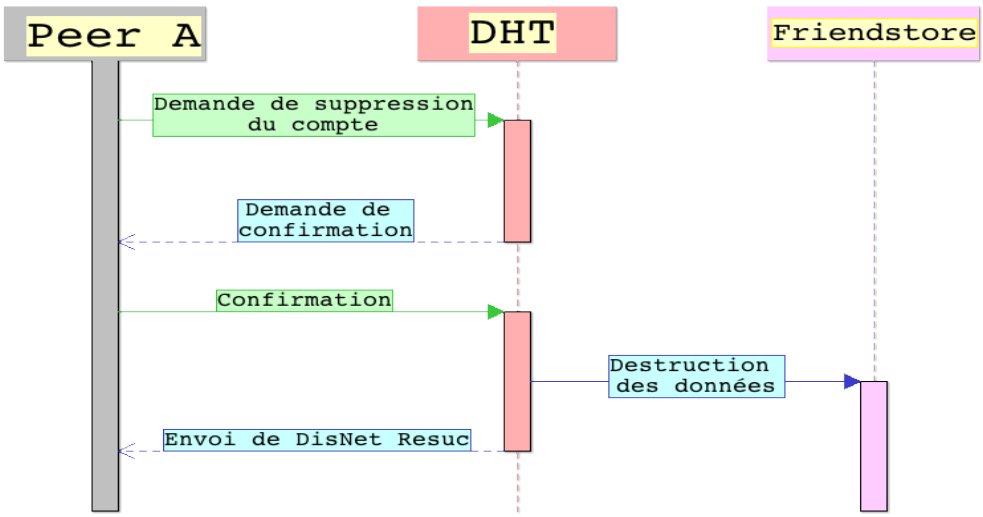
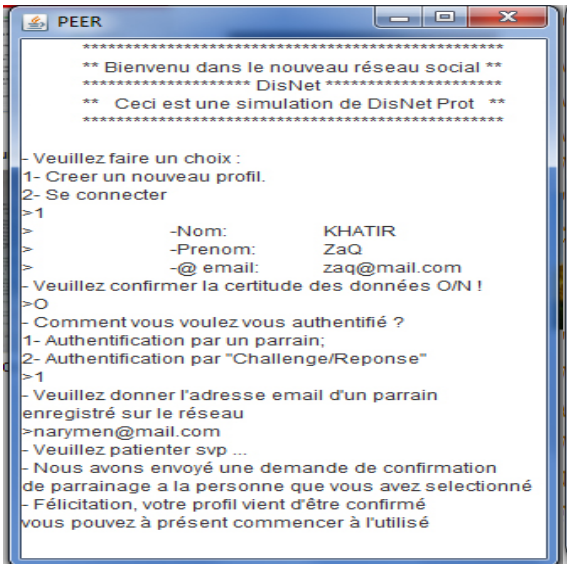


Figure III.9 : DisNet – suppression d'un compte utilisateur

IV. SIMULATION

Afin de valider notre travail nous avons effectué une simulation en Java pour montrer les différentes interactions entre les utilisateurs et la DHT ou entre les utilisateurs eux même.

Nous avons implémenté la plus part des étapes de DisNet Prot, lorsqu'un utilisateur se connecte comme le montre la figure III.10, l'application lui propose alors de choisir entre se connecter s'il dispose d'un profil ou d'en créer un s'il n'en à pas encore. Quant à la figure III.11, elle montre le fonctionnement de la DHT si l'utilisateur choisit de créer un nouveau profil. En effet lorsqu'il aura choisi de créer un nouveau profil, il devra alors remplir ses coordonnées. Ces coordonnées seront stockées dans l'annuaire provisoire de la DHT jusqu'à ce qu'il s'authentifie. Dans le cas de ces deux figures, l'utilisateur a choisi une authentification par un parrain, donc après confirmation du parrainage la DHT enregistrera le profil dans l'annuaire définitif et créera les métadonnées qui seront nécessaires aux prochaines connexions de cet utilisateur.



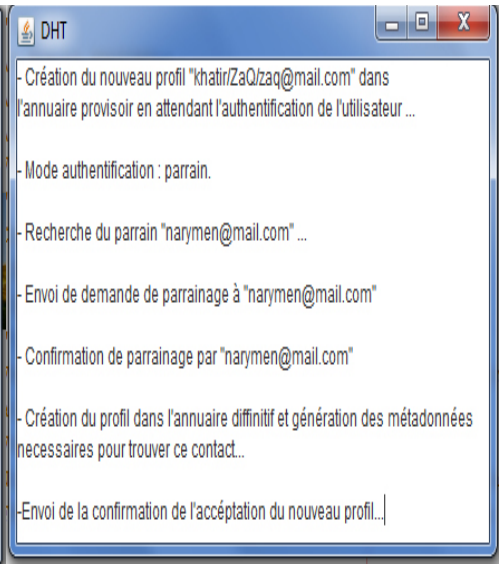


Figure III.10 : DisNet Prot – créer un profil – peer **Figure III.11 : DisNet Prot – créer un profil – DHT**

La connexion d'un utilisateur et l'envoi d'un message sont montrés dans la figure III.12 car lorsque l'utilisateur aura choisi de se connecter, il devra fournir son adresse email et son mot de passe, la DHT qui est montrée dans la figure III.13 synchronisera avec lui les messages envoyés en son absence, préviendra Friendstore de sa connexion et actualisera sa liste d'ami en ligne, s'il choisit d'envoyer un message à un autre utilisateur

qui est en ligne alors une liaison s'effectuera entre les deux peers et pourra donc communiquer.

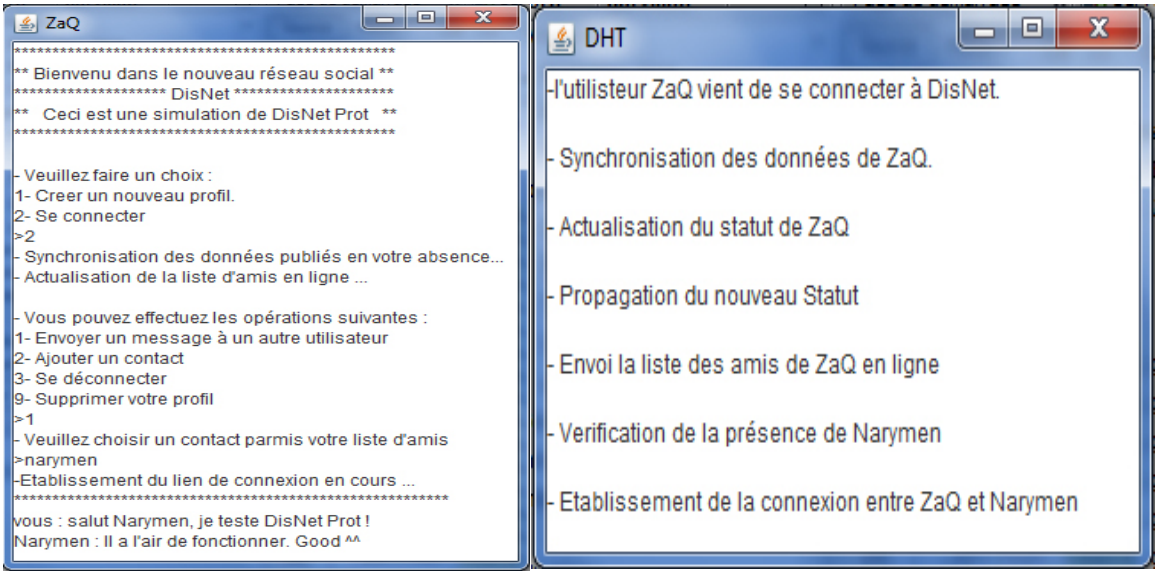


Figure III.12 : DisNet – envoi message – peer

Figure III.13 : DisNet – envoi de message DHT

Quand un utilisateur souhaite donc supprimer son profil comme montrer dans la figure III.14, l'application lui demande d'abord de vérifier s'il ne s'agit pas d'une erreur, ensuite après confirmation, la DHT comme montrée dans la figure III.15, procède à la destruction de ses données chez ses Friendstore, elle génère ensuite le fichier DisNet Resuc pour lui permettre de récupérer son profil s'il le souhaite, elle le lui envoi et rompe le contact avec lui. L'utilisateur aura ses données chez lui et pourra donc récupérer son profil avec le fichier DisNet Resuc, qui lui permettra aussi de récupérer tous ses contacts.

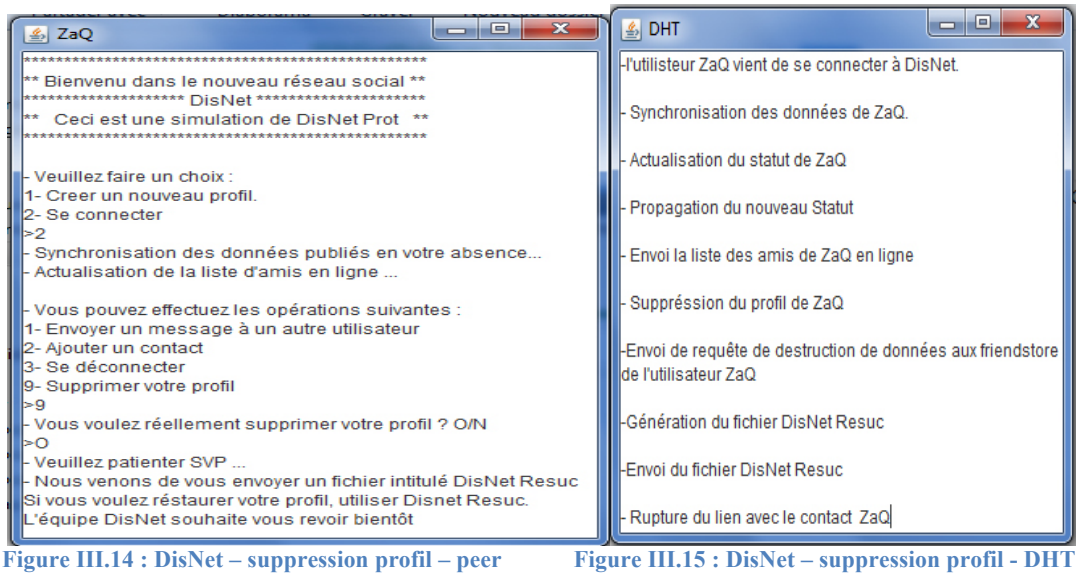


Figure III.14 : DisNet – suppression profil – peer

Figure III.15 : DisNet – suppression profil - DHT

CONCLUSION ET PERSPECTIVES

Le service des réseaux sociaux aujourd'hui est l'un des services prédominant du web. Ces réseaux visent un large éventail d'utilisateurs de toutes tranches d'âges et de sexes et cela pour diverses raisons tant personnelles que professionnelles.

L'avantage qu'offre ces réseaux est leur facilité d'utilisation pour publier des informations personnelles et communiquer avec aisance, même par des utilisateurs aux compétences techniques limitées.

La motivation principale pour un membre à joindre un tel réseau est la facilité de créer un profil, d'utiliser les différentes applications offertes par le service ainsi que la possibilité de partager facilement des informations avec des contacts sélectionnés ou le public et cela à des fins personnelles ou professionnelles.

Les données que l'utilisateur publie sont en fait stockées dans les serveurs du fournisseur du réseau social utilisé afin de permettre une disponibilité totale du contenu stocké. Cependant, l'utilisation frauduleuse des données personnelles commises par les différents réseaux sociaux centralisés à des fins commerciales et mercantiles ou sous couvert de la demande de certains gouvernements nécessite une réponse innovante et forte. C'est dans cette perspective que le présent mémoire a été élaboré.

La mise en sécurité des données personnelles et le respect de la vie privée sont deux priorités essentielles sur lesquelles peut se construire une réponse pertinente comme

le cas de PeerSon, Safebook, Decent et Cachet, quatre approches de réseaux sociaux distribués que nous avons étudié en détail.

Cette étude nous a permis d'étudier minutieusement leurs architectures et de comprendre le prototype de chaque approche, ensuite nous avons simplifié leur fonctionnement en proposant des diagrammes de séquences montrant chaque étape du fonctionnement de ces approches. Ce qui nous a permis de découvrir les manques existant dans ces approches pour proposer un nouveau model de réseau social distribué que nous avons baptisé DisNet.

Notre model a repris les points forts des approches évoquées ensuite nous avons apporté plusieurs amélioration concernant les manques découverts dans la synthèse. Ces améliorations touchent entre autre la gestion du stockage des données, le mode d'enregistrement, l'assurance de la disponibilité des données même en mode hors ligne. Nous avons proposé une solution nouvelle à un problème n'ayant pas été traité par aucun réseau social actuel et ne figurant pas non plus dans les approches étudiées à savoir : la suppression d'un compte utilisateur avec destruction de ses données tout en s'assurant que l'utilisateur pourra récupérer son profil avec toutes ses données et ses relations comme il les avait autrefois s'il le souhaite et cela d'une manière innovante.

Pour valider notre model, nous avons proposé une simulation de notre prototype que nous avons baptisé DisNet Prot. Cette simulation nous permet d'appréhender les divers problèmes auxquels nous pourrions être confrontés dans l'implémentation d'un produit fini.

Les perspectives de ce travail consiste à finaliser le prototype DisNet Prot et à l'améliorer, parmi les améliorations possibles citons :

- Essayer de trouver une solution pour assurer la disponibilité des données des utilisateurs même en l'absence de ces derniers tout en restant dans la décentralisation.
- Décentraliser le service de lookup qui pour le moment est considéré comme un service centralisé, afin de respecter une architecture 100% décentralisée.
- Automatiser le service d'authentification de challenge/réponse qui pour le moment est considéré comme un service centralisé collectant les données des utilisateurs.

REFERENCES BIBLIOGRAPHIQUES

[1] http://lci.tf1.fr/chaine-lci/

[2] http://www.washingtonpost.com/wp-srv/special/politics/prism-collection-documents/

[3] https://www.facebook.com/zuck

[4] http://boinc.berkeley.edu/

[5] Giuseppe DeCandia and .al, Dynamo: Amazon's Highly Available Key-value Store, SOSP '07 Proceedings of twenty-first ACM SIGOPS symposium on Operating systems principles, Pages 205-220, October 2007

[6] http://www.jamendo.com/fr/

[7] http://www.bittorrent.com/

[8] http://www.beet.tv/2010/05/adobes-big-peertopeer-plans-.html

[9] https://joindiaspora.com/

[10] http://movim.eu/

[11] https://lorea.org/

[12] Robert Morris and .al, Chord: A scalable peer-to-peer look-up protocol for internet applications, Saarland University, Department of Computer Science, November 2003.

[13] Sylvia Ratnasamy and .al, A Scalable Content-Addressable Network, Saarland University, Department of Computer Science, 2002.

[14] http://fr.usenet.nl/

[15] http://www.epostmail.org/

[16] http://oceanstore.cs.berkeley.edu/

[17] https://freenetproject.org/

[18] http://kademlia.scs.cs.nyu.edu/

[19] http://opendht.org/

[20] http://www.planet-lab.org/

[21] http://www.iss.net/security_center/advice/Services/SunRPC/default.htm

[22] http://xml-rpc.net/

[23] http://www.freepastry.org/

[24] http://research.microsoft.com/en-us/um/people/antr/pastry/

[25] http://friendstore.news.cs.nyu.edu/

[26] Randy Baden and .al, Persona: An Online Social Network with User-Defined Privacy, conference on Data communication, Proceedings of the ACM SIGCOMM ,2009.

[27] Luca Maria Aiello and Giancarlo Ruffo, LotusNet: tunable privacy for distributed online social network services, Preprint submitted to Computer Communications, December 19, 2010.

[28] Rajesh Sharma and Anwitaman Datta, SuperNova: Super-peers Based Architecture for Decentralized Online Social Networks, Fourth International Conference on Communication Systems and Networks (COMSNETS), 2012.

[29] Cashion, Protocol for mitigating the risk of hijacking social networking sites, 7th International Conference on Collaborative Computing: Networking, Applications and Worksharing (CollaborateCom), 2011.

[30] www.peerson.net

[31] www.safebook.us

[32] Refik Molva and .al, Safebook: Feasibility of Transitive Cooperation for Privacy on a Decentralized Social Network, World of Wireless, Mobile and Multimedia Networks & Workshops,IEEE, 2009.

[33] Alberto Ornaghi and Marco Valleri, Man in the middle attacks demos, Blackhat Conference USA, 2003.

[34] Yanchao Zhang, A Fine-Grained Reputation System for Reliable Service Selection in Peer-to-Peer Networks, Transactions on Parallel and Distributed Systems, IEEE, August 2007.

[35] Vadivu, G.S, Protecting peer-to-peer networks from the denial of service attacks, 3rd International Conference on Electronics Computer Technology (ICECT), 2011.

[36] Xu Xiang, Defeating against sybil-attacks in peer-to-peer networks, IEEE 26 International Parallel and Distributed Processing Symposium Workshops & PhD Forum (IPDPSW), 2012.

[37] Nikita Borisov and .al, DECENT: A Decentralized Architecture for Enforcing Privacy in Online Social Networks, IEEE International Conference on Pervasive Computing and Communications Workshops (PERCOM Workshops), 2012.

[38] Nikita Borisov and .al, Cachet: A Decentralized Architecture for Privacy Preserving Social Networking with Caching, CoNEXT '12 Proceedings of the 8th international conference on Emerging networking experiments and technologies, Pages 337-348, 2012

[39] Sonja Buchegger and .al, PeerSoN: P2P Social Networking — Early Experiences and Insights, SNS '09 Proceedings of the Second ACM EuroSys Workshop on Social Network Systems, 2009.

[40] Doris Schiöberg, A Peer-to-peer Infrastructure for Social Networks, Diplomarbeit in Informatik, Technische Universität Berlin Fakultät IV, Dezember 2008.

[41] Leucio Antonio et .al, Safebook: A Privacy-Preserving Online Social Network Leveraging on Real-Life Trust, CONSUMER COMMUNICATIONS AND NETWORKING, IEEE Communications Magazine, December 2009.

[42] http://www.safebook.us/prototype.html

[43] Refik Molva and .al, Safebook: A Distributed Privacy Preserving Online Social Network, IEEE International Symposium on World of Wireless, Mobile and Multimedia Networks (WoWMoM), 2011.

[44] http://www.authentify.com/solutions/out-of-band-authentication.html

[45] Goldfarb, Writing policies and procedures manuals, IEEE Transactions on Professional Communication, 2013.

[46] Jinguang Han, Privacy-Preserving Decentralized Key-Policy Attribute-Based Encryption, IEEE Transactions on Parallel and Distributed Systems, 2012.

[47] Hoffman, Cryptography policy, Communications of the ACM, Pages 109-117, 1994.

[48] Dinh Nguyen and .al, Friendstore: cooperative online backup using trusted nodes, Proceedings of the 1st Workshop on Social Network Systems Pages 37-42, 2008.