République Algérienne Démocratique et Populaire
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

Université Mouloud MAMMERI de Tizi-Ouzou
Faculté de Génie Électrique et Informatique
Département Informatique

Mémoire

De fin de cycle

Vue de l'Obtention du Diplôme de MASTER en Informatique .

OPTION : système d'informatique

Thème

CONCEPTION ET RÉALISATION D'UN OUTIL D'INTERROGATION D'UN ANNUAIRE LDAP

Proposé et dirigé par :

M^r: RAMDANE

Réalisé par :

M^elle : Fayed Dyhia

M^elle :Boubeki Baya

Promotion 2013/2014

REMERCIEMENTS

Nous tenons à travers ce modeste travail, remercier tout ceux qui nous ont aidés et contribués à sa réalisation.

Nous remercions particulièrement notre promoteur Mr RAMDANE d'avoir accepté de nous encadrer, diriger et Orienter durant toute la durée de notre projet.
Nous remercions également les membres du jury d'avoir aimablement accepté de juger notre travail.

Enfin tous nos proches, amis et camarades qui nous ont toujours soutenues et encouragées au cours de la réalisation de ce mémoire.

Dédicaces

Je dédie ce modeste travail

Aux êtres qui me sont les plus
chers au monde; ma chère mère et mon père à qui
je dois mon existence et mes succès
Que Dieu le tout puissant les protège.
Mon très cher frère et sa femme
Ma très chère sœur
A toute ma famille,
A tous mes amis (es),
A ma binôme ***baya*** *et à toute sa famille.*

A tous ceux que j'aime

Dyhia

Dédicaces

Je dédie ce modeste travail :

A la mémoire de ma très chère grand mère que
Dieu la reçoit dans son paradis.

Mes très chers parents les prunelles de mes yeux pour leur soutien tant moral que financier et pour l'amour qu'ils n'arrêtent pas de m'offrir depuis ma naissance.

Mes très chères sœurs et Mes très chers frères.

Ma binôme ***Dyhia*** *et à toute sa famille.*

Tous mes amis(e).

Baya

SOMMAIRE

Sommaire

Chapitre II

Chapitre III

Chapitre IV

Liste de figure

Chapitre I

Chapitre II

Liste de figure

Chapitre III

Chapitre IV

Liste de figure

Liste des tableaux

INTRODUCTION GÉNÉRALE

Un réseau informatique est une entité complexe qui propose des services très diversifiés et renferme un grand nombre d'équipements de différents types. Devenant indispensable afin de pouvoir faire des économies et de gagner en productivité, de plus en plus d'entreprises étendent leur réseaux existants presque aussi rapidement que l'apparition des nouvelles technologies et des nouveaux produits, conséquence: au début des années 1980, la technologie des réseaux a connue une croissance phénoménale permettant aux entreprises de faire communiquer plusieurs machines entre elles afin d'assurer des échanges d'informations: le transfert de fichiers, le partage des ressources (Imprimantes et données), de la messagerie ou l'exécution de programmes à distance. Avec le développement de l'utilisation d'Internet, les entreprises ouvrent leur système d'information à leurs partenaires ou leurs fournisseurs. La protection des ressources de l'entreprise avec le contrôle des utilisateurs du système d'information en octroyant des droits d'accès est donc essentielle.

Le but principal de notre projet est de concevoir et de réaliser une application pour un administrateur réseau, cette Application permet d'accéder et de manipuler les différents objets de l'annuaire. Pour la validation de notre travail nous l'avons testé dans in environnement Activa Directory.

Notre mémoire est organisé en quatre chapitres :

Le premier chapitre, intitulé «**Généralités sur les réseaux** » traite les concepts basés sur les intranets notamment l'architecture client /server.
Le second chapitre, intitulé <<**les annuaires réseaux**>> .
Le troisième chapitre, intitulé <<**analyse et conception**>> présente le modèle de conception de notre application. Nous avons opté pour le langage UML (*Unified Modeling Langage*) comme langage de modélisation.
Le dernier chapitre, intitulé « **Réalisation** », présente l'environnement de développement de notre application, les outils utilisés, quelque code source et des interfaces de notre application.

Enfin, nous terminerons avec une conclusion générale.

CHAPITRE I :
GÉNÉRALITÉ SUE LES RÉSEAUX

Introduction

Les réseaux informatiques occupent aujourd'hui une place prépondérante dans l'évolution technologique. A leur origine, ils permettaient de relier des terminaux passifs à de gros ordinateurs, actuellement, ils permettent l'interconnexion de tous types d'ordinateurs, que ce soit de gros serveurs, des stations de travail, des ordinateurs personnels ou de simples terminaux graphiques. Ils sont nés d'un besoin d'échanger des informations de manière simple et rapide entre plusieurs machines.

1. Un réseau informatique :[4]

Est un ensemble d'ordinateurs ou terminaux qui sont reliés entre eux, capables de s'échanger des données et de partager des ressources via un réseau de transmission.

2. Classification des réseaux :[4]

2.1. Selon la topologie :

2.1.1. en bus :

Cette topologie consiste à relier chaque ordinateur a un bus par l'intermédiaire de câbles coaxiaux, et l'information envoyée par un poste est diffusée en même temps vers tous les utilisateurs mais seul le poste destinataire est censé le prendre en compte.

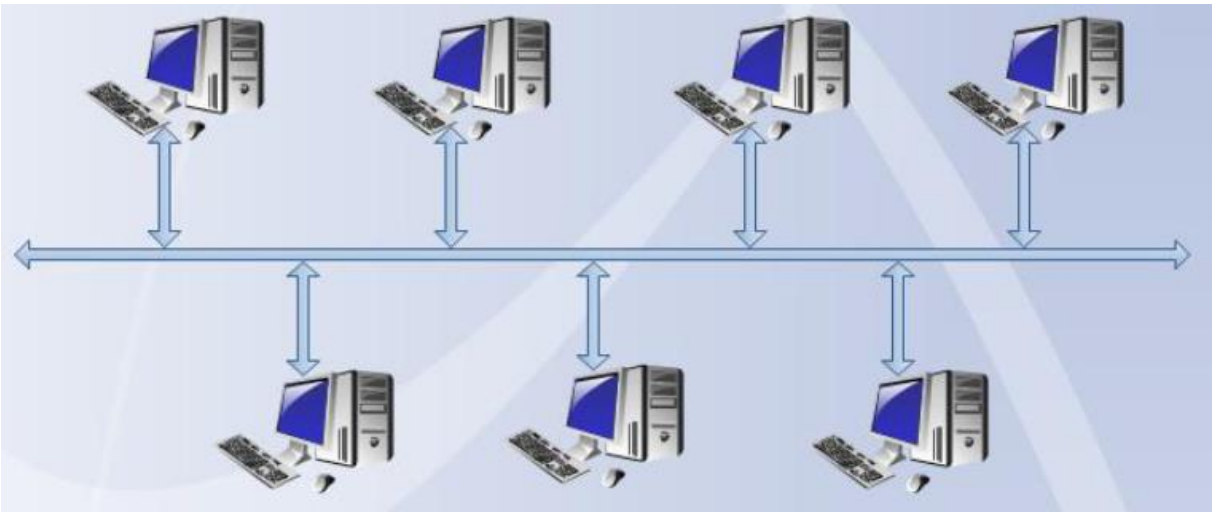


Figure 2.1.1 : Topologie en bus

2.1.2.En étoile :

Dans cette topologie chaque ordinateur est relié à un poste central (hub ou Switch), chaque message émis par un ordinateur passe par le poste central qui le renvoi à l'ordinateur destinataire.

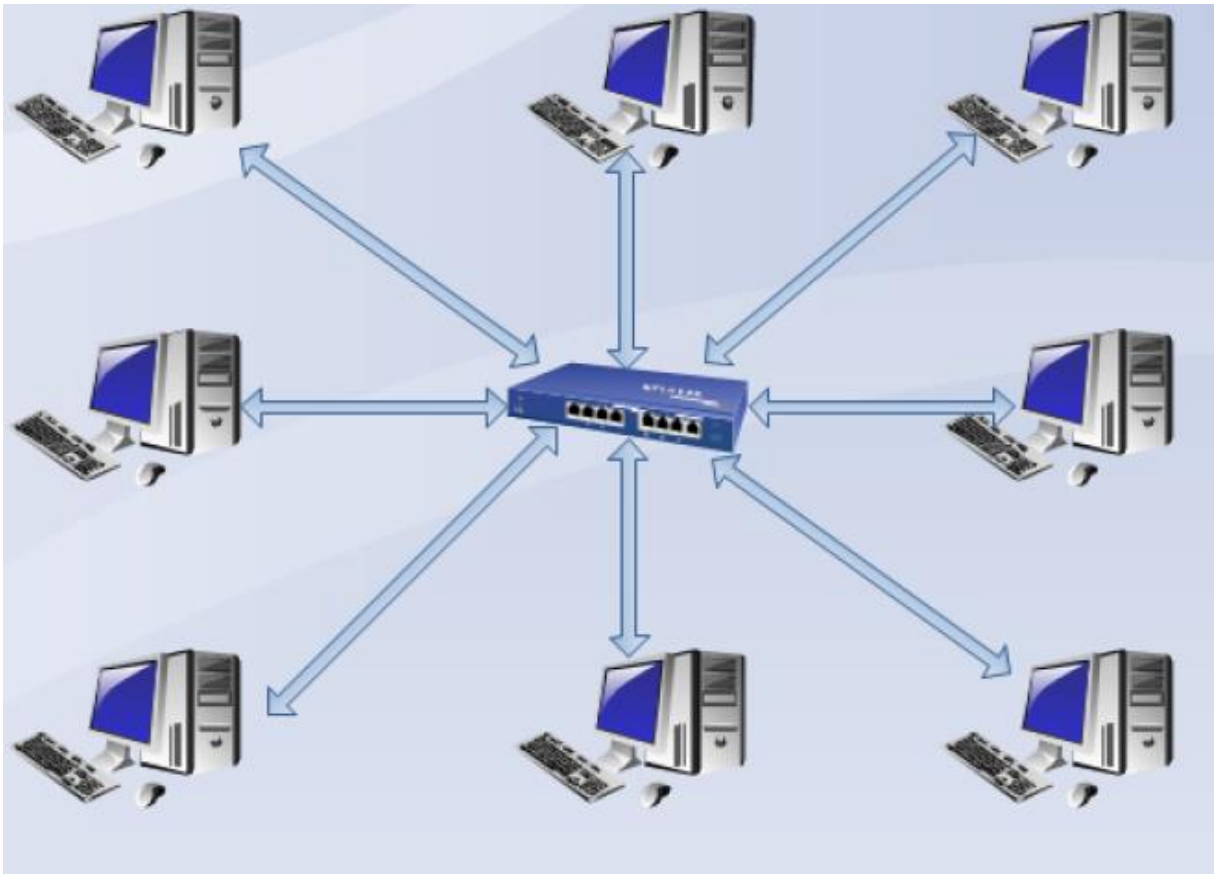


Figure 2.1.2 : Topologie en étoile

2.1.3. En anneau :

Les stations sont connectées par un câble formant un anneau (ring). Dans cette topologie le message est transmis d'une station à une autre jusqu'à destination. Chaque station recevant un message vérifie s'il lui est destiné si c'est le cas elle le garde sinon elle le transmet à la station voisine.

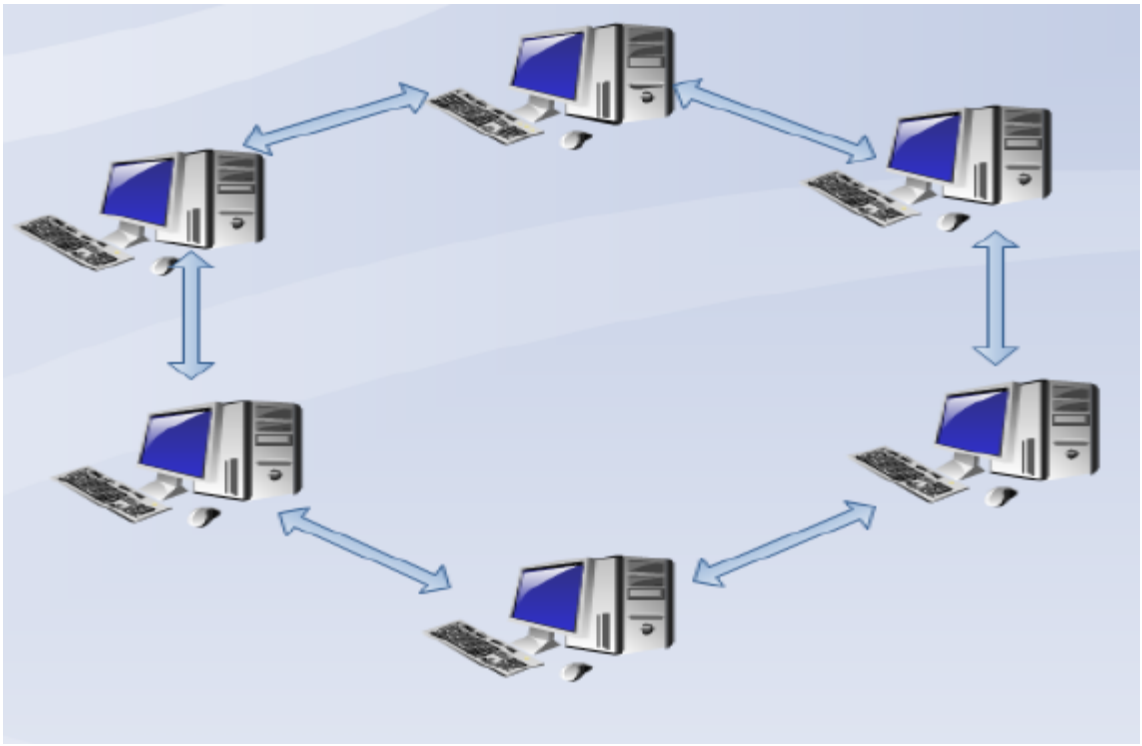


Figure 2.1.3 : Topologie en anneau

2.2. Selon la distance :

2.2.1. Réseau local (LAN: local area network) :

Est un réseau permettant de relier un ensemble d'ordinateurs appartenant à une même organisation, dans une petite aire géographique par un réseau, de quelque mètre à quelque centaine de mètres.

2.2.2. Réseau métropolitain (MAN : métropolitaine area network) :

Un réseau MAN interconnecte plusieurs LAN géographiquement proche (plusieurs kilomètres) formant un seul MAN. Ainsi un MAN permet à deux nœuds distants de communiquer comme s'ils faisaient partie d'un même réseau local.

2.2.3. Réseau étendu :

Un WAN interconnecte plusieurs LANs à travers de grandes distances géographiques (réseau de réseaux), Les WAN fonctionnent grâce à des routeurs qui permettent de choisir le trajet le plus approprié pour atteindre un nœud du réseau. Le plus connu des WAN est Internet.

2.3. Selon le type de liaison :[4]

2.3.1. Liaison filaire :

C'est un type de réseau dans lequel toutes les liaisons sont filaires (câble ou fibre optique).

2.3.2. Liaison sans fils :

C'est un type de réseau qui véhicule les informations par infrarouge ou bien par onde radio (utilisant généralement la bande de fréquence 2.4 Ghz). La transmission par onde radio est la plus répandue en raison de sa plus large couverte géographique et de son haut débit.

Un réseau local sans fils peut être implémenté comme une extension à un réseau câblé préexistant ou comme une entité indépendante.

3. Routage :[4]

Le routage est le mécanisme par lequel des chemins sont sélectionnés dans un réseau pour acheminer les données d'un expéditeur jusqu'à un ou plusieurs destinataires. Le routage est une tâche exécutée dans de nombreux réseaux, tels que le réseau téléphonique, les réseaux de données électroniques comme l'internet, et les réseaux de transports.

Un routeur est un élément intermédiaire dans un réseau informatique assurant le routage des paquets. Son rôle est de faire transiter des paquets d'une interface réseau vers une autre, selon un ensemble de règles formant la table de routage. C'est un équipement de la couche trois (03) du modèle **OSI.**

Un routeur doit être connecté à au moins deux (02) réseaux informatiques pour être utile, sinon il n'aura rien à router. L'appareil crée et/ou maintient la table de routage qui mémorise les meilleures routes vers les autres réseaux via les métriques associées à ces routes.

4. Architecture des réseaux :

Le transport des données d'une extrémité à l'autre d'un réseau nécessite un support physique ou hertzien de communication. Cependant, pour que ces données arrivent correctement à destination, il faut une architecture logicielle.

4.1. Architecture OSI :[4]

4.1.1. Définition :[4]

L'**ISO (I**nternational **S**tandardization **O**rganization**)** a normalisé sa propre architecture sous le nom d'**OSI** (**O**pen **S**ystem **I**nterconnection). L'architecture OSI est la première à avoir été définie, et ce de façon relativement parallèle à celle d'internet.

Le but de l'ISO est de créer un modèle idéal où chaque couche effectue une tâche définie et dépend des services de la couche inférieure. Chaque couche fournit ses propres services à la couche supérieure.

Figure 4.1.1. les couche du modèle OSI

4.1.2. Fonctionnalités de chaque couche :[4]

- **Couche physique (1) :** Elle transfère les bits à travers un canal de communication. Ces bits encodés peuvent être en numérique mais aussi en analogique. Cette couche transmet les bits de la couche liaison de données' à l'interface physique et vice-versa.
- **Couche liaison de données (2) :** Elle prend les données de la couche physique et fournit ses services à la couche 'réseau'. Les bits reçus sont assemblés en trames.
- **Couche réseau (3) :** Elle gère les connexions entre les nœuds du réseau et s'occupe essentiellement de l'adressage (datagrammes), du routage et du contrôle de flux.
- **Couche transport (4) :** Elle réalise le découpage des messages en paquets pour le compte de la couche « réseau » ou le réassemblage des paquets en messages pour les couches supérieures.
- **Couche session (5) :** Elle assure l'ouverture et la fermeture des sessions avec les applications, définie les règles d'organisation et de synchronisation du dialogue entre les abonnés.
- **Couche présentation (6) :** Elle gère la représentation des données
- **Couche application (7) :** Elle fournit l'ensemble des services de réseautique aux programmes d'application de l'utilisateur. Les services les plus couramment utilisés sont le contrôle de l'accès aux fichiers distants, le transfert de données et le service de messagerie.

4.2. Architecture TCP/IP :[4]

4.2.1. Historique :

- En 1969, l'agence gouvernementale d'Amérique DARPA lance un réseau expérimental intitulé **ARPANET** (**A**dvanced **R**esearch **P**rojects **A**gency **NET**work) qui est basé sur la commutation de paquets.
- En 1975, le réseau passe officiellement du stade expérimental au stade opérationnel.
- En juin 1978, Jon POSTEL définit IPv4.
- En 1981, standardisation de 'IP'.
- En 1983, les protocoles TCP/IP sont adoptés comme un standard militaire et toutes les machines sur le réseau commencent à l'utiliser.
- Depuis cette époque, un nouveau terme est apparu pour désigner cette interconnexion de réseaux, l'**internet.**
- Depuis 1990, disparition d'ARPANET.
- Depuis 1994, l'internet s'est ouvert au commerce.
- En 1995, pour faire face à sa popularité fortement croissante et aux demandes de transactions sécurisées, le protocole évolue et définit la version 6 (IPng).
- Les protocoles désignés par TCP/IP ont également envahi les réseaux locaux eux-mêmes car il est plus facile d'utiliser les mêmes protocoles en interne et en externe.

Figure 4.2.1. les couches du modèle TCP/IP

4.2.2. Caractéristiques :[4]

- C'est un protocole ouvert, les sources sont disponibles et gratuites et ont été développées indépendamment d'une architecture particulière, d'un système d'exploitation particulier.
- Ce protocole est indépendant du support physique du réseau. Cela permet à TCP/IP d'être véhiculé par des supports et des technologies aussi différents.
- Le mode d'adressage est commun à tous les utilisateurs de TCP/IP quel que soit la plateforme qu'il utilise.

- Les protocoles de haut niveau sont standardisés, ce qui permet des développements largement répandus sur tout type de machine.

4.2.3. Comparaison des architectures TCP/IP et ISO :[4]

La suite de protocoles désignée par TCP/IP ou encore « pile ARPA », est construite sur un modèle en couche moins complet que la proposition de l'ISO.
Quatre (04) couches sont suffisantes pour définir l'architecture de ce protocole.

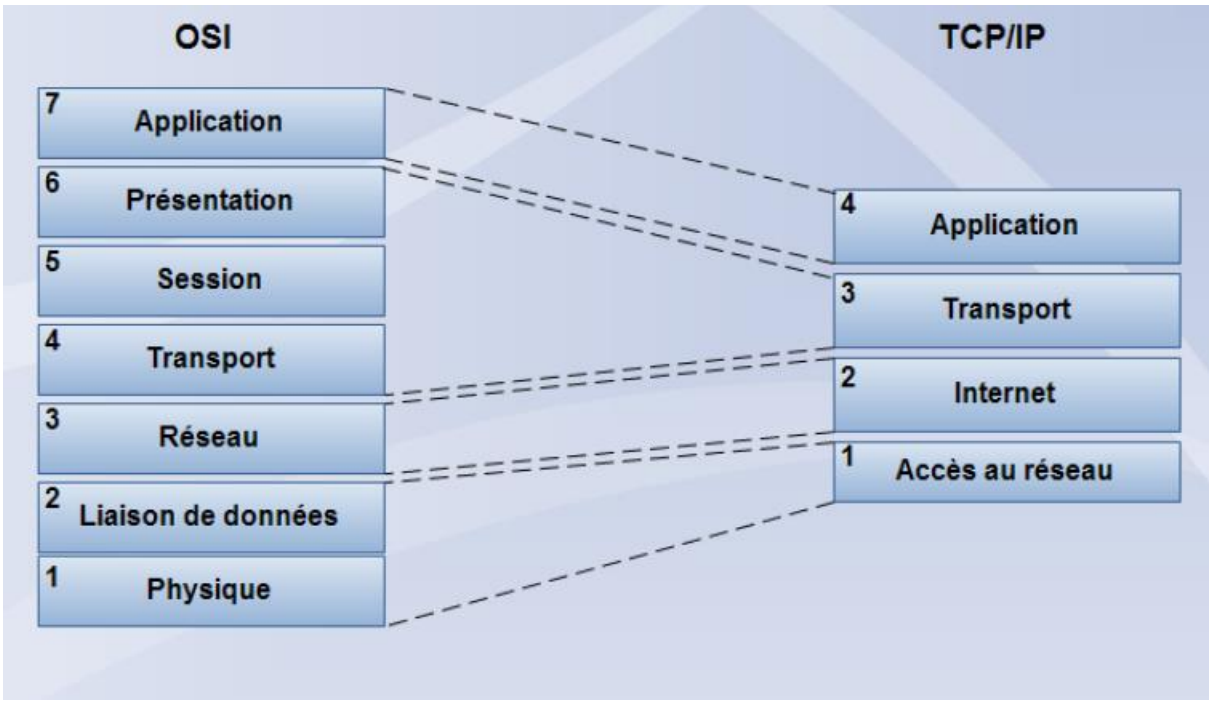


Figure 4.2.3. Comparaison des architectures TCP/IP et ISO

5. Architecture client /serveur :[4]

5.1. Les techniques de dialogue client/serveur : [4]

Le Client/serveur est avant tout un mode de dialogue entre deux (02) processus, le premier s'appelle « Client » et le second s'appelle « Serveur ». Le modèle de communication Client/serveur est orienté vers la fourniture de services par un processus serveur à un processus client. Un échange consiste en la transmission d'une requête à un processus serveur qui exécute l'opération demandée et envoi au retour la réponse.

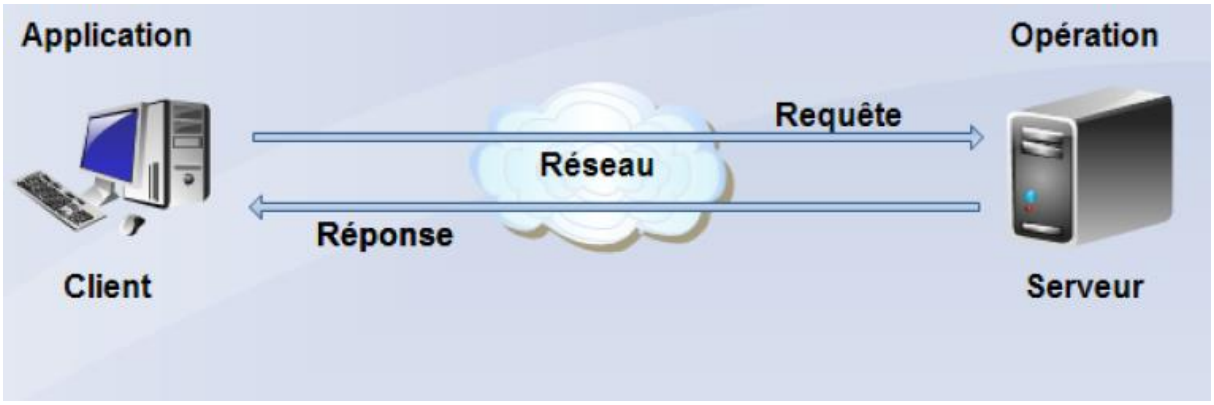


Figure 5.1. Architecture client /serveur

5.1.1. Notions de base : Les quatre (04) notions de base sont :

- **Client :** C'est une entité (processus, programme, ordinateur,…) qui demande l'exécution d'une opération à une autre entité par envoi d'un message contenant le descriptif de l'opération à exécuter et attendant la réponse à cette opération par un message au retour.
- **Serveur :** C'est une entité (processus, programme,…) qui accomplit une opération sur la demande d'un client et lui transmet la réponse.
- **Réponse :** Message transmis par un serveur à un client suite à l'exécution d'une opération contenant les paramètres de l'opération.
- **Requête :** Message transmis par un client à un serveur décrivant l'opération à exécuter pour le compte du client.

5.1.2. Le middleware:[4]

Ensemble des services logiciels construits au-dessus d'un protocole de transport afin de permettre l'échange de requêtes et des réponses associées entre client et serveur de manière transparente.

5.2. Les types d'architecture client /serveur

5.2.1. Architecture client/serveur à deux niveaux

L'architecture client/serveur repose sur un modèle d'architecture distribuée bipartite. L'interface graphique se situe sur le poste client et la base de données est localisée sur le serveur. La logique de traitement pouvant se situer sur l'une ou l'autre des parties. Dans une architecture client -serveur bipartite, les PC sont généralement connectés aux serveurs de base de données via un réseau local.

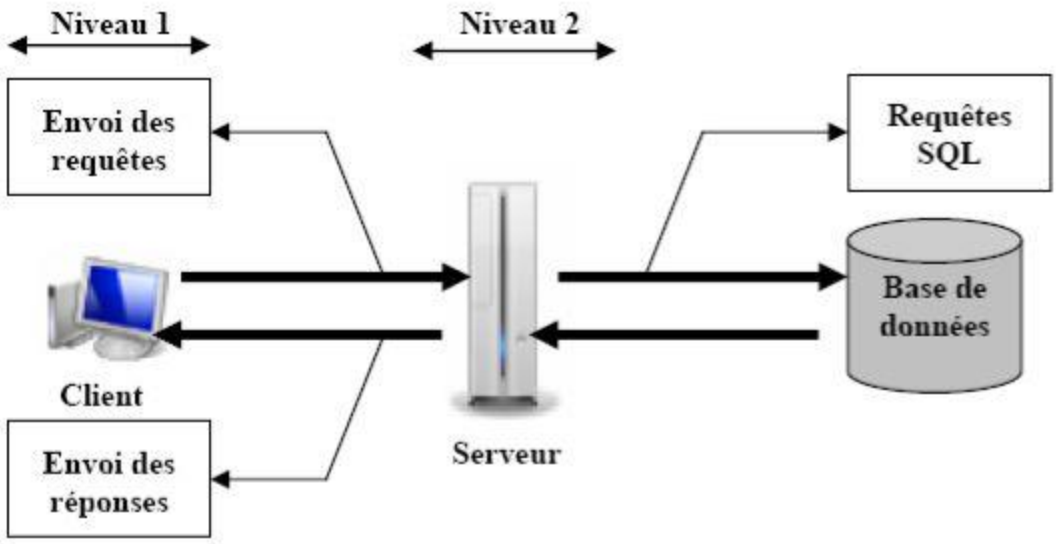


Figure 5.2.1. Architecture client/serveur à deux niveaux

5.2.2. Architecture Client/serveur à trois (03) niveaux :

L'architecture à 3 niveaux (appelée aussi architecture 3-tiers) est généralement partagée entre :

- **Un Client :** L'ordinateur demandeur de ressource

- **Un serveur d'application (Middleware) :** Il est chargé de fournir la ressource en faisant appel à un autre serveur
- **Un serveur secondaire (Serveur de base de données) :**
  Il est chargé de fournir au serveur d'application les données dont il a besoin.

Les applications de l'architecture Client/serveur 3-tiers sont plus faciles à déployer et à gérer sur le réseau, elles essayent de minimiser les échanges sur le réseau en créant des niveaux de service. Elle est recommandée pour diverses raisons dont on cite celle de l'**internet**.

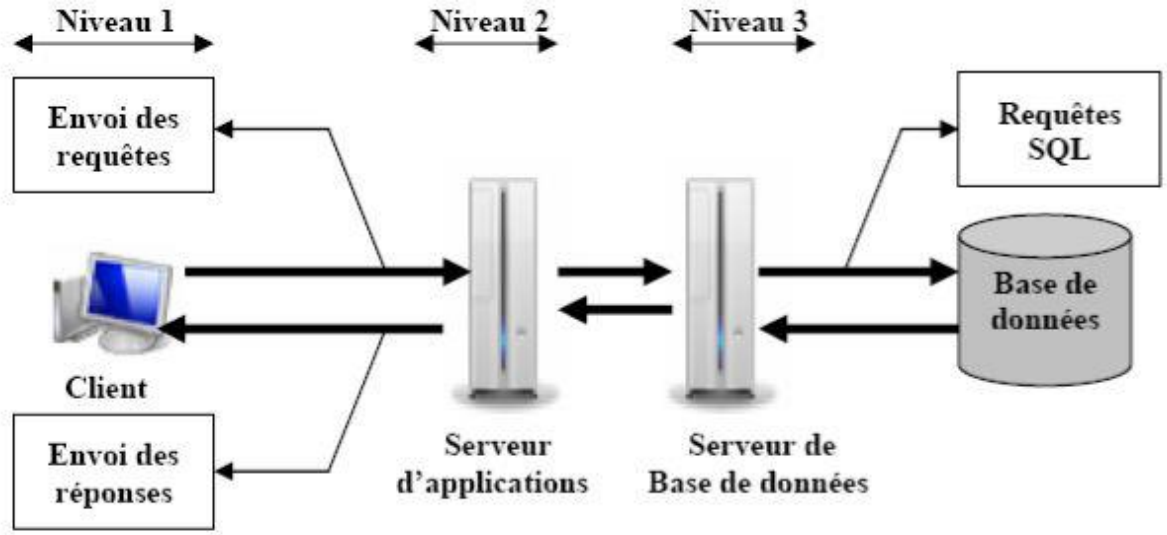


Figure 5.2.2. Architecture Client/serveur à trois (03) niveaux

5.2.3. Architecture Client/serveur à n-tiers:

Elle a été pensée pour pallier aux limitations des architectures 3-tiers et concevoir des applications puissantes et simples à maintenir. Ce type d'architecture permet de distribuer plus librement la logique applicative, ce qui facilite la répartition de la charge entre tous les niveaux.

Cette évolution des architectures 3-tiers met en œuvre une approche objet pour offrir une plus grande souplesse d'implémentation et faciliter la réutilisation des développements.

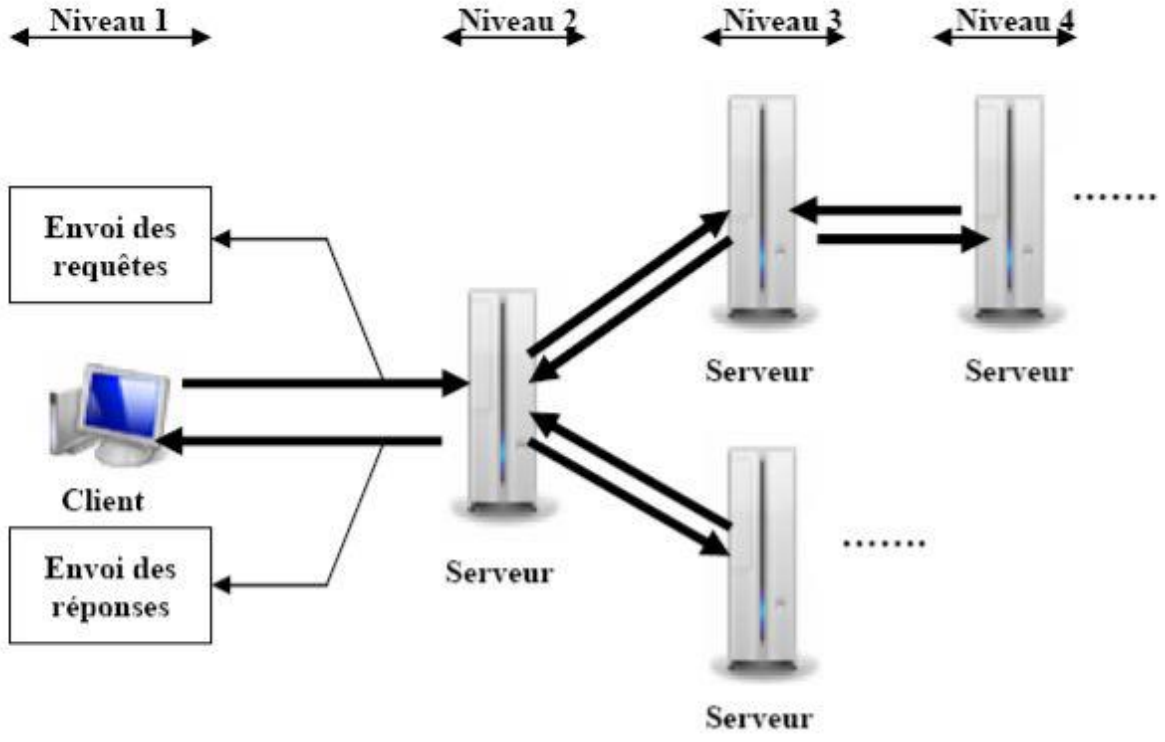


Figure 5.2.3. Architecture Client/serveur à n-tiers

5.3. Types de serveurs :

Il existe plusieurs variantes des technologies Client/serveur, comme les décrivent les paragraphes suivant :

5.3.1. Serveur de fichiers :

Dans le cas de ce serveur, le client requiert des enregistrements de fichiers en émettant des requêtes au serveur de fichiers. Ce dernier est utile pour partager des fichiers sur un réseau et ils sont indispensables pour créer des banques de documents, d'images, …etc.
L'obtention d'information nécessite de nombreux échanges de messages sur le réseau.

5.3.2. Serveur de base de données :

Dans le cas de ce serveur, le client émet des requêtes SQL sous forme de messages en direction du serveur, le résultat renvoyé au client. Le serveur utilise sa capacité de traitement pour rechercher les données demandées au lieu de transmettre tous les articles au client au lieu de le laisser en faire la sélection, donc la puissance est répartie et utilisée de façon beaucoup plus efficace.

5.3.3. Serveur de transaction :

Dans le cas de ce serveur, le client invoque des procédures distantes résidantes sur le serveur qui comporte un moteur de base de données SQL. L'échange sur le réseau consiste en un seul message requête/réponse (contrairement à l'application serveur de base de données pour laquelle le message requête/réponse est émis pour chaque instruction SQL). Pour ce type de serveur, l'application Client/serveur nécessite du code source au niveau du serveur.

5.3.4. Serveur d'application web :

Le World Wide Web est la première application Client/serveur intergalactique. Ce modèle consiste en des clients qui envoient des demandes à de très gros serveurs qui renvoient des documents (services).

5.3.5. Serveur groupware:

Le groupware s'intéresse à la gestion d'informations semi-structurées telles que le texte, l'image, le courrier, la messagerie et l'ordonnancement des tâches.

5.3.6. Serveur d'application objet :

Dans le cas de ce serveur, l'application Client/serveur est écrite sous forme d'un jeu d'objets communicants. Les objets client communiquent avec les objets serveur au moyen d'un courtier d'objets ou **ORB** (**O**bject **R**equest **B**roker) qui est un intermédiaire entre le serveur et le client, assistant ces derniers pour le choix et la consultation des banques de données. Le client invoque une méthode sur un objet distant, l'ORB localise une instance de la classe, appelle la méthode demandée et renvoie les résultats à l'objet client.

5.4. Avantages et inconvénients de l'architecture Client/serveur :

5.4.1. Avantage :

- Les serveurs sont conçus pour le partage de ressources et ne servent pas de station de travail. Il suffit de les dimensionner en fonction de la taille du réseau et le nombre de clients susceptibles de s'y connecter ;
- Les serveurs étant toujours en service (sauf en cas de panne), les ressources sont toujours disponibles pour les utilisateurs ;
- Les sauvegardes de données sont centralisées, donc beaucoup plus faciles à mettre en œuvre ;
- Un administrateur gère le fonctionnement du réseau et les utilisateurs n'ont pas à s'en préoccuper.

5.4.2. Inconvénients :

- Elle nécessite impérativement la présence d'un administrateur possédant les compétences nécessaires pour faire fonctionner le réseau ;
- Le coût est évidemment plus élevé puisqu'il faut la présence d'un ou plusieurs serveurs ;
- Si un serveur tombe en panne, ses ressources ne sont plus disponibles.

6. Internet :

6.1. Définition :[5]

Internet est un réseau international (réseau de réseau ou inter-réseau) d'ordinateurs communiquant entre eux grâce à un protocole d'échange de données standardisé TCP/IP .Chaque ordinateur de réseau possède une adresse, appelée adresse IP qui est unique dans le monde.

Les ordinateurs connectés au réseau Internet peuvent communiquer entre eux de façon transparente pour l'utilisateur, indépendamment de type d'ordinateurs utilisés, mais en utilisant des logiciels appropriés, c'est-à-dire utilisant des protocoles reconnus sur Internet.

6.2. Les protocoles et l'adressage sur Internet :[5]

6.2.1. Le protocole TCP/IP :

Ce protocole est en fait une combinaison de deux protocoles agissant chacun à un niveau différent, le protocole TCP se situe au niveau de transport et le protocole IP au niveau réseau.

- **TCP** (**T**ransport **C**ontrol **P**rotocol) : fonctionne en mode connecté, c'est-à-dire qu'une liaison est établie entre deux ordinateurs. TCP est un protocole fiable dans la mesure où il est capable de déterminer si un paquet est arrivé ou s'il est interrompu. TCP s'occupe de la transmission des paquets, vérifie l'ordre d'arrivée et l'intégrité des données.
- **IP** (**I**nternet **P**rotocol) **:** Est un protocole de la couche réseau. Il est le noeud de transmission de données au-dessous de TCP. Il s'occupe du transport de données, mais il ne contient aucun outil de contrôle de données ni de vérification de paquets, il permet d'adresser les paquets au bon endroit.

6.2.2. HTTP (HyperText Transport Protocol) :

C'est un protocole léger et rapide, utilisé pour délivrer des fichiers multimédia et hypertextes appelés plus généralement ressources, en utilisant Internet. Il gère la totalité des échanges réalisés entre le client et le serveur. Pour chaque demande d'accès à URL, une requête http est émise indépendamment par le client.

6.2.3. Le DNS (domaine Name Service) :

Le DNS est un ensemble de protocoles permettant aux utilisateurs d'un réseau TCP/IP d'accéder aux hôtes à l'aide des noms conviviaux hiérarchisés, le serveur DNS assure la transformation des noms de domaines en adresses IP et inversement, car la structure des adresses IP est complexe à manipuler pour cela on utilise le DNS, sans celui-ci les utilisateurs devrait mémoriser les adresses IP.

6.2.4. Masque de sous réseau :

Le masque de sous réseau (netmask) permet de définir le réseau dans lequel vous vous trouvez.

6.3. Les services de l'Internet :

Internet offre à ses utilisateurs une très grande variété de services. Parmi ceux-ci, trois fonctions de base simples : le courrier, le transfert de fichiers et la connexion à distance. Grace à ces trois services de base, les chercheurs ont pu exploiter les possibilités de ce nouvel environnement et l'utiliser comme un outil de travail de collaboration à travers le monde.

6.3.1. Courrier électronique (e-mail) :

Le service de messagerie électronique est l'application la plus utilisé et une des plus anciennes, permet d'échanger des messages avec des millions de personnes à travers le monde. Au départ, les messages ne contenaient que du texte pur. Suite à l'amélioration du protocole de messagerie, il est possible désormais d'inclure des pièces jointes de n'importe quel type (document bureautique, images, sons, vidéos, ..).

6.3.2. Transfert de fichiers :

Internet est utilisé comme une grande bibliothèque distribuée à travers la planète. Le transfert des données au sein d'un système d'information est dans la quasi-totalité un cas réalisé grâce au protocole **FTP** (**F**ile **T**ransfer **P**rotocol) qui permet à l'utilisateur de :

- Se connecter à tout poste de travail faisant office de serveur de documents à un instant donné et de se déconnecter à la fin du téléchargement.
- De transférer des fichiers, quel que soit la nature, entre les deux extrémités de la connexion et dans les deux sens.

6.3.3. Télécharger (Download) :

Récupérer des informations qui sont sur internet pour les stocker sur son propre ordinateur.

6.3.4. Connexion à un ordinateur éloigné :

La connexion à des ordinateurs éloignés est réalisée grâce au protocole TELNET.

6.4. Le WWW (World Wide Web):[5]

6.4.1. Définition :

Le World Wide Web ou toile d'araignée ou tout simplement le web, est une des applications majeures permises par l'internet. C'est un système qui permet de consulter, avec un navigateur, des pages mises en ligne sur des sites. Nous avons donc, d'un côté, un ensemble d'ordinateurs connectés entre eux (internet) et de l'autre, un ensemble de documents modifiables qui sont aussi connectés entre eux (web).

6.4.2. L'architecture Web : L'architecture web est du type Client/serveur

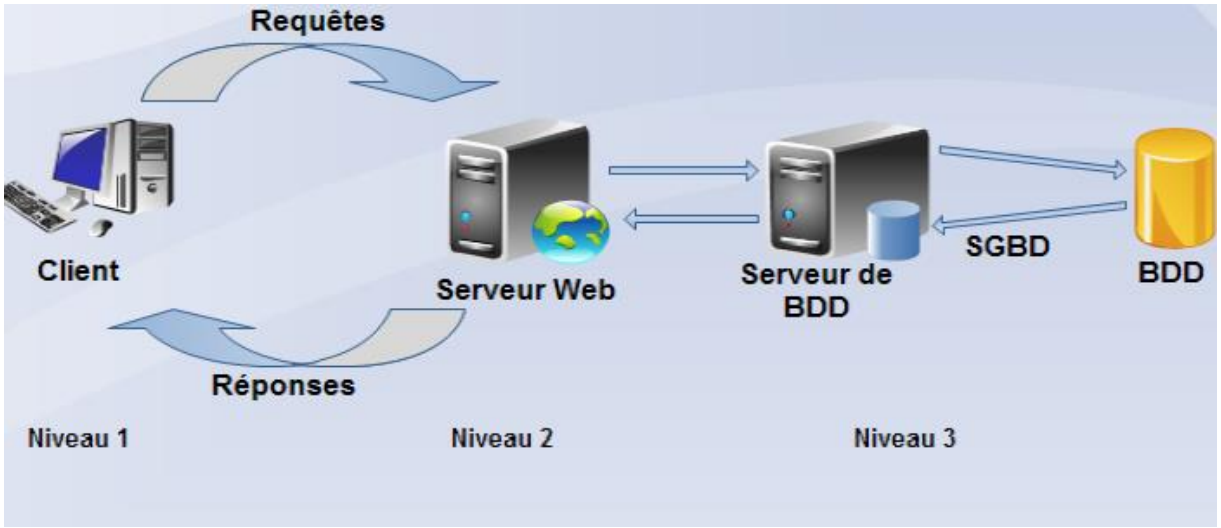


Figure 6.4.2. L'architecture Web

- **Le serveur web :** C'est un processus capable de traiter des requêtes http issues des clients web. Ce processus est présenté sur un serveur physique (machine) accessible par les ordinateurs du réseau via TCP/IP. Ce serveur contient les informations qui peuvent être mises à la disposition des utilisateurs sous forme de fichier HTML.
- **Le client web :** Appelé aussi navigateur ou browser, constitue la partie visible à l'utilisateur, c'est lui qui se charge d'afficher l'information venant du serveur web.
- **Le serveur de base de données :** Dans le cas de serveur de base de données, le client émet des requêtes sous forme de messages au serveur. Le résultat de chaque requête est envoyé au client. Le serveur utilise sa propre capacité de traitement pour rechercher les données demandées au lieu de transmettre tous les articles aux clients et de le laisser en faire la sélection.

6.4.3. Concepts du Web :[5]

6.4.3.1. L'hypertexte :

L'hypertexte est une manière d'organiser et de présenter l'information dans laquelle certains élément du texte, appelés liens, permettant de se déplacer vers d'autres zones du texte ou vers une autre page.

6.4.3.2. Hypermédia :

Extension de l'hypertexte à des données multimédias permettant d'inclure des liens entre des éléments textuels, visuels et sonores.

6.4.3.3. URL (Uniform Ressource Locator) :

Une **URL** est un format de nommage universel servant à identifier et localiser des ressources consultables sur Internet. Il s'agit d'une chaîne de caractères ASCII imprimables.

6.4.3.4. Navigateur web :

Logiciel client que l'on appelle navigateur web (browser en anglais) dont la fonction première est d'interpréter les adresses des pages web, de les afficher et d'exploiter les liens hypertextes à l'intérieur de celles-ci. Il existe un certain nombre de navigateurs web mais les plus connus restent Internet Explorer et Netscape.

6.4.3.5. Site web :

Un site web est un ensemble de pages web et d'éventuelles autres ressources du World Wide Web, hyper liées en un ensemble cohérent, c'est-à-dire conçu pour être consulté avec un navigateur Web et mis à disposition par un même auteur (organisme ou individu) dans un même but.

Conclusion :

Dans ce chapitre nous avons défini en premier lieu des généralités sur les réseaux ainsi que les concepts qui lui sont essentiels comme son architecture, ses classifications, les routeurs et les protocoles, ensuite nous avons présenté les notions de bases d'Internet.
Dans le chapitre qui suit, nous procéderons à la présentation des annuaires réseaux.

CHAPITRE II:
LES ANNUAIRES RESEAUX

Introduction :

Les particuliers et les entreprises ont de plus en plus recours aux réseaux pour accéder à des applications distribuées et à des ressources partagées (sites web, serveurs d'applications, serveurs de fichiers, etc.).

Ces applications et ces ressources doivent interagir avec des ordinateurs situés dans le même réseau local, à travers l'intranet de l'entreprise, ou plus généralement au travers de l'Internet. Cela nécessite a priori la connaissance des adresses de ces différentes machines. Or, dans la très grande majorité des cas, on n'utilise jamais les adresses réelles des machines ; on utilise des noms.

Prenons des exemples simples. L'accès à un site web se fera par l'intermédiaire d'un nom désignant le site. L'accès à une imprimante se fera également par l'intermédiaire d'un nom désignant l'imprimante. Ces informations vont être gérées dans une base de données spéciale appelée annuaire. L'annuaire va permettre de transformer le nom du site ou le nom de l'imprimante en une adresse physique permettant aux protocoles de communication d'accéder aux équipements concernés.

De nombreux outils d'annuaires ont donc vu le jour au fil des années, offrant des services divers et variés ; certains ont périclité, d'autres sont devenus immédiatement des standards incontournables, tel DNS (Domain Name System). Depuis quelques années maintenant, est apparu un nouveau standard, lui-même en passe de devenir absolument indispensable connu sous le sigle LDAP (Lightweight Directory Access Protocol). Ce standard ne remplacera pas DNS, ce n'est pas sa vocation, mais il permet d'unifier certains besoins tels que ceux d'annuaires de type pages blanches, d'annuaires de type NIS (Network Information Service ; « yellow pages »), d'authentification, etc.

1. Définition des annuaires réseaux [10]

Un annuaire est une liste, un répertoire mis à jour régulièrement qui regroupe des informations (nom, adresse, coordonnées, etc.) sur les membres d'une association, d'une entreprise, d'un établissement d'enseignement ou d'un organisme professionnel, ou sur les abonnés à un service.

En informatique : [10]

Dans le monde de l'informatique, un annuaire est un système de stockage de données, dérivé des bases de données hiérarchisées, permettant en particulier de conserver les données pérennes, c'est-à-dire les données n'étant que peu mises à jour (historiquement, sur une base annuelle, d'où le nom), comme les coordonnées des personnes, des partenaires, des clients et des fournisseurs d'une entreprise, les adresses électroniques. La recherche peut se faire selon de multiples critères et les données peuvent être utilisées par des logiciels clients comme des applications serveurs (serveur de messagerie : Postfix, Sendmail, etc…). En outre, certaines versions de service d'annuaires savent gérer les droits sur les données mais aussi les services proposés sur les machines clientes par une authentification grâce à un couple login / mot de passe. Aujourd'hui ces données sont centralisées sur une ou plusieurs machines et les données sont transférées entre les machines via des protocoles réseaux.

2. La différence entre un annuaire et une base de données : [11]

Un annuaire est un type de base de données spécifique, c'est-à-dire qu'il s'agit d'une sorte de base de données ayant des caractéristiques particulières :

- un annuaire est prévu pour être plus sollicité en lecture qu'en écriture. Cela signifie qu'un annuaire est conçu pour être plus souvent consulté que mis à jour.
- les données sont stockées de manière hiérarchique dans l'annuaire, tandis que les bases de données dites "relationnelles" stockent les enregistrements de façon tabulaire.
- les annuaires doivent être compacts et reposer sur un protocole réseau léger
- Un annuaire doit comporter des mécanismes permettant de rechercher facilement une information et d'organiser les résultats
- les annuaires doivent pouvoir être répartis. Cela signifie qu'un serveur d'annuaire doit comporter des mécanismes permettant de coopérer, c'est-à-dire d'étendre la recherche sur des serveurs tiers si jamais aucun enregistrement n'est trouvé
- Un annuaire doit être capable de gérer l'authentification des utilisateurs ainsi que les droits de ceux-ci pour la consultation ou la modification de données

Ainsi, un annuaire est généralement une application se basant sur une base de données afin d'y stocker des enregistrements, mais surtout un ensemble de services permettant de retrouver facilement les enregistrements à l'aide de requêtes simples. Une base de données par contre n'est pas forcément un annuaire...

3. Caractéristique :

- un index qui facilite la communication entre des entités
- une organisation hiérarchique optimisée pour un accès rapide à de nombreuses informations en petits volumes
- des entités et objets sous la forme de personnes, de communautés, de ressources ou d'équipements
- des accès aux bases de données mises à jour par les utilisateurs des applications informatiques.

4. Avantage des annuaires :

- Rapidité pour accéder aux informations,
- Les mécanismes de sécurité pouvant être mis en œuvre,
- La centralisation des informations
- Les possibilités de redondance de l'information.

5. Les types des annuaires [22]

La forme des annuaires électronique a fortement évolué depuis leur apparition au début de l'ère informatique. Citons-en quelques-uns :

- Unix : /etc/passwd (années 70 – 80). Ce type d'annuaire, local à une machine, permet de gérer les différents utilisateurs pouvant être autorisés à se connecter sur cette machine.
- NIS (« yellow pages »; Network Information Service). Annuaire dont les données sont réparties sur l'ensemble des machines composant le réseau de l'entreprise. Une machine au moins doit jouer le rôle de serveur.
- DNS (Domain Name System; [rfc1034] [rfc1035]). Cet annuaire complètement réparti sur l'ensemble du réseau permet principalement la résolution de noms de machines en adresses réseau.
- whois (1985). Annuaire autrefois géré par l'Internic. Il stocke un certain nombre d'informations sur le réseau lui-même (adresses des sites, des entreprises, noms de domaines...).
- X.500 (1988, 1993, 1997). Annuaire global de type pages blanches et pages jaunes.
- LDAP (1993). Une version allégée des annuaires X.500.

6. Annuaires LDAP :

6.1. Historique :[11]

En 1988, l'Union Internationale des Communications (UIT) met au point les annuaires X.500. Le but de cette opération est d'uniformiser l'accès aux services, de centraliser les ressources et de les protéger. Le protocole utilisé pour y accéder est le protocole DAP (Directory Access Protocol).

Malheureusement, le protocole DAP s'avère difficile à mettre en œuvre et ne fonctionne pas sur les réseaux TCP/IP. En 1993, l'Université du Michigan réfléchit donc à un moyen de pallier ces deux problèmes :

Elle met en place le protocole LDAP (Lightweight Directory Access Protocol), au départ simple "connecteur" TCP/IP avec des annuaires X.500.

En 1995, LDAP devient un protocole natif et utilisable indépendamment de X.500.

LDAP est donc une évolution de la norme X.500. Sa version actuelle est la version 3 (RFC 2251), elle propose les évolutions suivantes par rapport à la version 2 :

- Le support des communications chiffrées via SSL/TLS
- L'authentification via SASL
- Le support des Referrals (une branche pointe vers un autre annuaire)
- Le support d'Unicode (internationalisation)
- La capacité d'étendre le protocole
- Le support des schémas dans l'annuaire

6.2. Quelque annuaire LDAP :[12]

Voici une liste des principaux annuaires LDAP existant sur le marché :

- OpenLDAP: http://www.openldap.org
- Apache Directory Server: http://directory.apache.org

- Sun (One/Java) Directory Server : http://www.sun.com
- Active Directory : http://www.microsoft.com

...

6.3. Objectif : [11]

- fournir aux utilisateurs des informations fiables, facilement accessibles
- permettre aux utilisateurs de mettre à jour eux-mêmes leurs informations personnelles
- rendre les informations accessibles de façon contrôlée
- éviter la redondance d'informations : un seul annuaire pour l'ensemble des services
- faciliter la gestion (administration) des postes de travail, des équipements réseau

7. Le protocole LDAP :

7.1. Définition :

LDAP "Lightweight Directory Access Protocol" est un protocole *simplifié* dérivé de DAP permettant un accès à un annuaire en mode client/serveur à l'aide des protocoles TCP/IP.

7.2. Fonctionnement du protocole LDAP : [11]

Pour son fonctionnement, LDAP met en place 2 méthodes de communication pour 2 fonctionnalités différentes. Une communication de type client/serveur pour permettre au client d'accéder aux informations contenues sur le serveur. Une communication de type serveur/serveur pour permettre au serveur de dupliquer ou synchroniser ses informations sur d'autres serveurs.

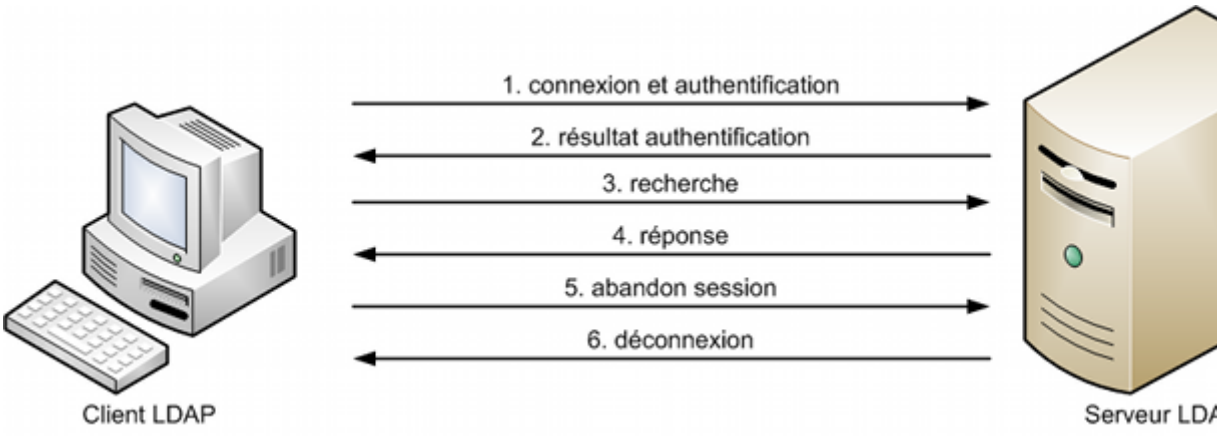


Figure 7.2. Exemple de communication client/serveur

Les échanges avec le protocole LDAP se font au format ASCII comme pour HTTP ou SMTP. En plus des opérations présentées sur l'exemple de communication client/serveur ci-dessus, les opérations de base définies par le protocole LDAP sont :

- interrogation: search, compare

- mise à jour: add, delete, modify
- connexion: bind, unbind, abandon

Etant donné que ces échanges sont réalisés au format ASCII, des mécanismes d'authentification et de chiffrement sont mis en place pour sécuriser le service.

Le protocole LDAP met en jeu 5 modèles qui définissent son fonctionnement à différents niveaux. Ces 5 modèles sont :

- un modèle d'information: pour définir le type de données de l'annuaire
- un modèle de nommage: pour indiquer comment les données sont organisées
- un modèle fonctionnel: pour indiquer comment accéder aux données
- un modèle de sécurité: pour indiquer comment protéger l'accès aux données
- un modèle de duplication: pour indiquer comment répartir les données entre serveurs

7.3. Explication des modèle LDAP :

7.3.1. Le modèle d'information :[12]

LDAP permet de gérer des données. Ces données utilisent un modèle particulier pour être stockées. Dans ce modèle, l'élément de base est appelé "Entry".

- **Le Directory Information Tree :**

Les données LDAP sont structurées dans une arborescence hiérarchique qu'on peut comparer au système de fichier Unix. Chaque nœud de l'arbre correspond à une **entrée** de l'annuaire ou directory service entry (DSE) et au sommet de cette arbre, appelé Directory Information Tree (DIT), se trouve la racine ou suffixe. Ce modèle est en fait repris de X.500, mais contrairement à ce dernier, conçu pour rendre un service d'annuaire mondial (ce que le DNS fait par exemple pour les noms de machines de l'Internet), l'espace de nommage d'un annuaire LDAP n'est pas inscrit dans un contexte global.

Les entrées correspondent à des objets abstraits ou issus du monde réel, comme une personne, une imprimante, ou des paramètres de configuration. Elles contiennent un certain nombre de champs appelés attributs dans lesquelles sont stockées des valeurs. Chaque serveur possède une entrée spéciale, appelée root directory spécific entry (rootDSE) qui contient la description de l'arbre et de son contenu.

Exemple :

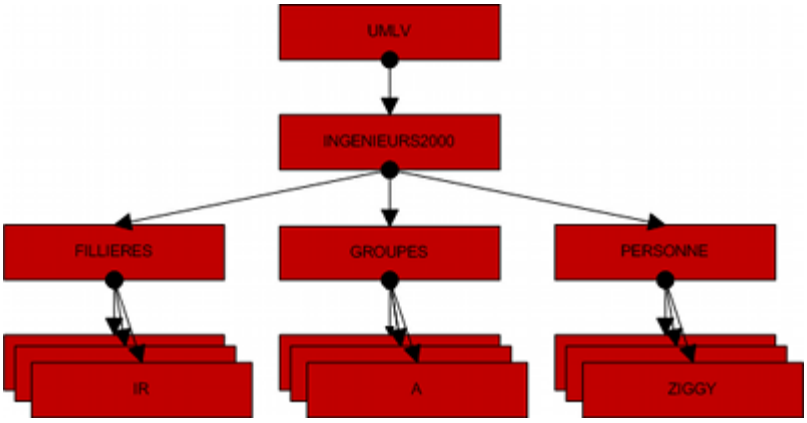


Figure 7.3.1.1.Exemple de Directory Information Tree (DIT)

- **Attribut :**

Un attribut est caractérisé par:

- ✓ un nom
- ✓ un type
- ✓ une méthode de comparaison
- ✓ un « Object Identifier » (IOD)
- ✓ une valeur

On distingue habituellement deux types d'attributs:

- Les **attributs utilisateurs** (**user attributes**) sont les attributs caractérisant l'objet manipulé par les utilisateurs de l'annuaire (nom, prénom, ...)
- Les **attributs opérationnels** (**system attributes**) sont des attributs auxquels seul le serveur peut accéder afin de manipuler les données de l'annuaire (dates de modification, ...)

Remarque: un attribut peut être possédé par plusieurs classes !

Exemple :

Une entré de type "Fournisseur" peut avoir le même attribut "cn" (common name) qu'une entrée de type "Client".

Voici une liste des attributs classiques que l'on retrouve sur les entrées d'un service LDAP:

attribu	description
cn	« common name » ou nom commum
o	« organization name » ou nom de l'organisation
gn	« given name » ou le surnom
l	« locality name » ou nom de la localité
st	« state name » ou nom de l'état
ou	« organisational unit » ou unité d'organisation
dc	« domain component » ou nom de domaine

Figure 7.3.1.2. Les attributs classiques de LDAP

D'une manière générale, tous les types d'entrées (Client, Fournisseur, ...) et leur attributs (cn, ou, ...) sont définis dans un schéma.

- **Classe Object** :

Une classe d'objet permet de décrire une entité (une personne par exemple) par une liste d'attribut. Elle est définie par :

- un nom
- un OID
- des attributs obligatoires
- des attributs optionnels
- un type (structurel, auxiliaire ou abstraite)

Le type d'une classe d'objets est lié à la nature des attributs qu'elle utilise :

- Une classe *structurelle* correspond à la description d'objets courants de l'annuaire : les personnes, les groupes, les unités organisationnelles... Une entrée appartient toujours à une classe de type structurelle.
- Une classe *auxiliaire* permet de rajouter des informations complémentaires à des objets structurels.
- Une classe *abstraite* désigne des objets basiques du schéma (top, alias...)
-

Le tableau donne un exemple de classes d'objet :

Nom	Supérieur	Type	Attribut Obligatoire	Attribut facultatif	Description
TOP	Aucun	ABSTRACT	Aucun	Aucun	Classe parente de toutes les classes
Person	TOP	STRUCTURAL	Sn,cn	TelephoneNumber, description	Classe de base modélisant une personne

group	TOP	STRUCTURAL	cn	description	Groupe d'utilisateur

Tableau 7.3.1.1.Exemple de classe d'objet

- **Schéma :**

Le schéma définit l'ensemble des types d'entrées par le service LDAP. Chaque entrée de l'annuaire fait obligatoirement référence à une classe d'objet du schéma. Les types d'entrées sont organisés de manière hiérarchique.

Exemple :

Le sommet de cette organisation hiérarchique est toujours occupé par le type "Top". Et cette organisation met en place un système d'héritage où chaque type hérite des attributs de son type parent.

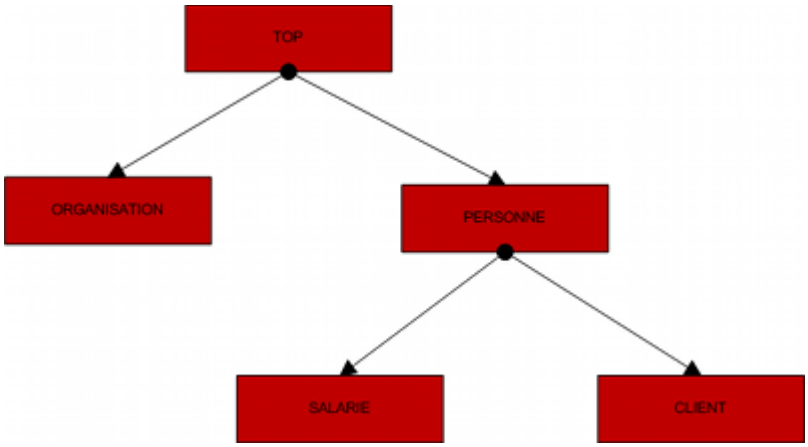


Figure 7.3.1.3.exemple d'organisation hiérarchique

Sur l'exemple ci-dessus, les types "Client" et "Salarié" héritent des attributs du type "Personne" qui lui-même héritent des attributs du type "Top".

- **Définition de schémas LDAP :**

Il existe plusieurs formats pour décrire un schéma LDAP :

- slapd.conf : fichier de configuration utilisé par U-M slapd, OpenLDAP et Netscape Directory
- ASN.1 : utilisé dans les documents décrivant les standards LDAP et X500
- LDAPv3 : La version 3 du protocole LDAP introduit l'obligation pour un serveur de publier son schéma via LDAP, pour permettre aux applications clientes d'en connaître le contenu. Le schéma est localisé par l'attribut opérationnel subschemaSubentry de l'entrée rootDSE. La valeur de cet attribut est une liste de DNs qui pointent vers des entrées, dont la classe d'objet est subschema, dans lesquelles sont stockées les descriptions des objets et des attributs.

Le tableau suivant montre les différences de syntaxe pour l'attribut cn et l'objet person

	Sldap.conf	ASN.1	LDAPv3
attribut	attribut cn commonName 2.5.4.3 cis	ub-common- name INTEGER ::= 64 commonName ATTRIBUTE WITH ATTRIBUTE - SYNTAX caseIgnoreStringSynta x (SIZE (1..ub - common-name))::= {attributeType 3}	Attributetypes: (2.5.4.3 NAME 'cn' DESC 'commonName Standard' Attribute' SYNTAX 1.3.5.1.4.1.1466.115.121.1.15)
objet	objectclass person oid 2.5.6.6 superior top requires sn, cn allows description, seeAlso, telephoneNum ber, userPassword	person OBJECT-CLASS ::= { SUBCLASS OF top MUST CONTAIN { commonName, surname} MAY CONTAIN { description, seeAlso, telephoneNumber, userPassword} ::= {objectClass 6}	Objectclass: (2.5.6.6 NAME 'person' DESC 'standard person' Object Class' SUP 'top' MUST (objectclass $ sn $ cn) MAY (description $ seealso $ telephonenumber $ userpassword))

Tableau 7.3.1. Format de description du schema

- **Les OIDs :**

Les objets et leurs attributs sont normalisés par le RFC 2256 de sorte à assurer l'interopérabilité entre les logiciels. Ils sont issus du schéma de X.500, plus des ajouts du standard LDAP ou d'autres consortiums industriels. Ils sont tous référencés par un Object identifier (OID) unique dont la liste est tenue à jour par l'Internet Assigned Numbers Authority (IANA).

Il est possible de modifier le schéma en rajoutant des attributs à un objet (déconseillé) ou en créant un nouvel objet (mieux) et d'obtenir un OID pour cet objet auprès de l'IANA (encore mieux).

Un OID est une séquence de nombres entiers séparés par des points. Les OID sont alloués de manière hiérarchique de telle manière que seule l'autorité qui a délégation sur la hiérarchie "1.2.3" peut définir la signification de l'objet "1.2.3.4".

Par exemple :

2.5 - fait référence au service X500
2.5.4 - est la définition des types d'attributs
2.5.6 - est la définition des classes d'objets
1.3.6.1 - the Internet OID
1.3.6.1.4.1 - IANA-assigned company OIDs, used for private MIBs
1.3.6.1.4.1.4203 – OpenLDAP

7.3.2. Le modèle de nommage :

Une fois le modèle d'information définit, il faut pouvoir définir la manière dont sont référencées les différentes informations gérer par le service LDAP. C'est le rôle du modèle de nommage. Il définit comment sont organisées les entrées de l'annuaire et comment elles sont référencée.

Exemple :

Cette organisation est représentée par le Directory Information Tree (DIT). C'est une classification comparable au système de fichier UNIX.

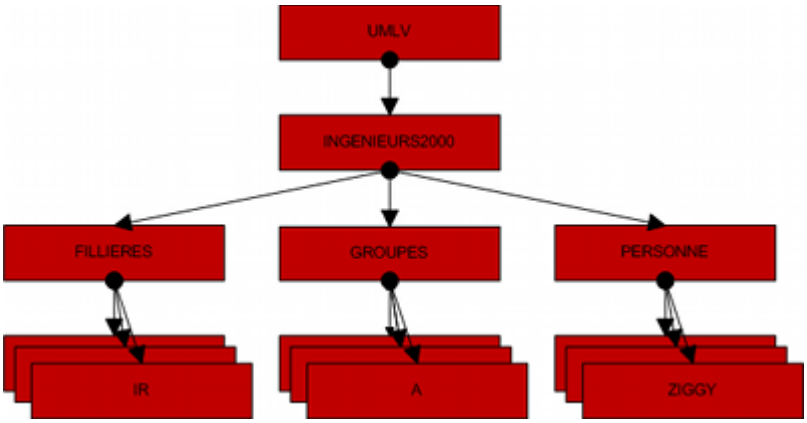


Figure7.3.2.1. exemple de directory information tree(DIT)

Chaque nœud du DIT correspond à une entrée de l'annuaire. Au sommet se trouve l'entrée "Suffix" ou "Root Entry". Cette dernière correspond à l'espace de nommage géré par le serveur LDAP. Il faut savoir qu'un serveur LDAP peut gérer plusieurs arbres donc plusieurs "Root Entry".

Exemple :

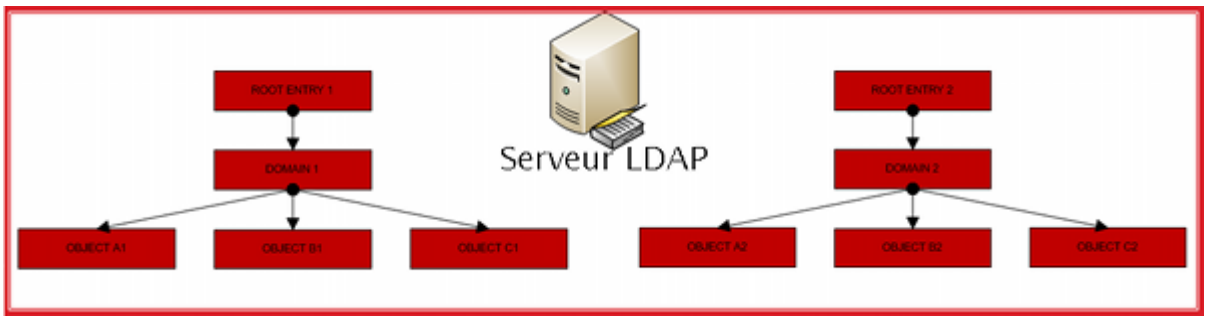


Figure 7.3.2.2. Schéma d'un serveur LDAP avec 2 "Root Entry"

Pour référencer de manière unique une entrée contenue dans le DIT, on utilise un "Distinguish Name" (DN). Ce dernier est équivalent à un path d'un fichier UNIX. Chaque élément qui compose le DN est appelé "Relative Distinguish Name" (RDN).

Un DN est constitué d'un ensemble d'attribut et de leurs valeurs provenant de chacune des entrées parentes mises bout à bout.

Voici un exemple de DN :

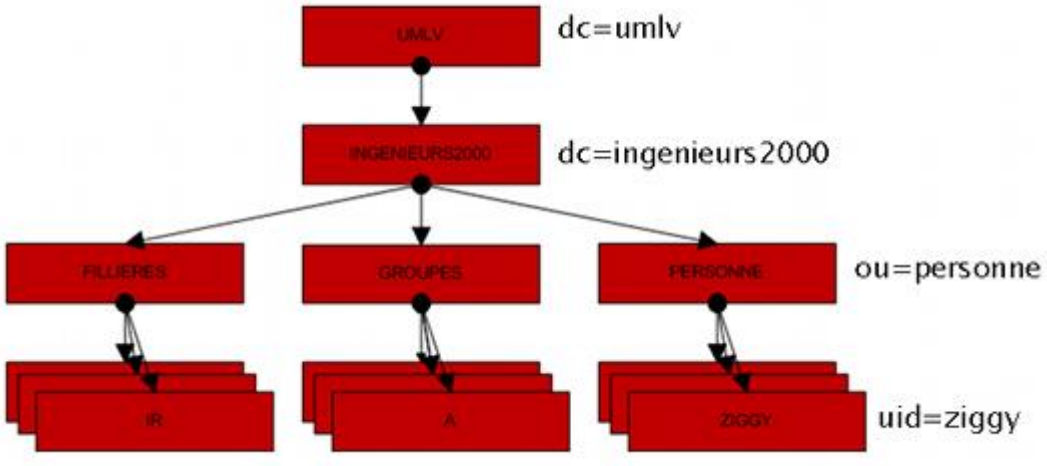


Figure 7.3.2.3.exemple de DN

DN de l'entrée ziggy = [uid=ziggy, ou=personne, dc=ingenieurs2000, dc = umlv]
On doit s'assurer que 2 entrées du DIT n'aient pas le même DN

Remarque: il est important de s'assurer que 2 entrées d'un même DIT n'aient pas le même DN. Pour cela, il faut s'assurer que la sélection des attributs composant le DN donne un résultat unique.

7.3.3. Le modèle de fonctionnement :

Une fois les données stockées et référencées, il faut permettre d'utiliser ces données. Pour cela, LDAP définit un modèle de fonctionnement. Ainsi, ce modèle définit les opérations possibles sur les données. Elles sont résumées dans le tableau suivant. Elles permettent d'accéder au serveur ou de modifier la structure de l'arbre et/ou les entrées de l'annuaire. Elles jouent un rôle analogue aux commandes de manipulation de fichiers d'Unix (cp, mv, rm...).

Opération LDAP	**Description**
Search	recherche dans l'annuaire d'objets à partir de critères
Compare	comparaison du contenu de deux objets
Add	ajout d'une entrée ajout d'une entrée
Modify	modification du contenu d'une entrée
Delete	suppression d'un objet
Rename (Modify DN)	modification du DN d'une entrée
Bind	connexion au serveur
Unbind	Déconnexion
Abandon	abandon d'une opération en cours
Extended	opérations étendues (v3)

Tableau 7.3.3.1. Opération de base

Les commandes search et compare se font sous la forme d'une requête composée de 8 paramètres :

Paramètre	**Description**
base object	l'endroit de l'arbre où doit commencer la recherche
Scope	La profondeur de la recherche
derefAliases	Si on suit les liens ou pas
size limit	Nombre de réponses limite
time limit	Temps maxi alloué pour la recherche
AttrOnly	Renvoie ou pas la valeur des attributs en plus de leur type
search filter	Le filtre de recherche
list of attributes	La liste des attributs que l'on souhaite connaître

Tableau 7.3.3.2. Paramètres d'une requête

Le scope définit la profondeur de la recherche dans le **DIT**. Il peut prendre différentes valeurs selon la portée de la recherche souhaitée :

- **base** : recherche dans le niveau courant
- **one-level** : recherche uniquement dans le niveau inférieur au niveau courant
- **subtree**: recherche dans tout le sous arbre à partir du niveau courant

7.3.4. Le modèle de sécurité

Le modèle de sécurité permet de protéger l'accès aux données de l'annuaire. La sécurité se fait à plusieurs niveaux.

7.3.4.1. L'authentification :

LDAP est un protocole avec connexion : l'ouverture de session (bind) s'accompagne d'une identification et, éventuellement, d'un mot de passe (optionnel en V3). LDAP propose plusieurs choix d'authentification :

- Anonymous authentification : correspond à un accès au serveur sans authentification, qui permet uniquement de consulter les données accessibles en lecture pour tous.
- Root DN Authentification : Utilisateur privilégié. Il a accès en modification à toutes les données.
- Mot de passe + SSL : La session entre le serveur et le client est chiffrée et le mot de passe ne transite plus en clair
- Simple Authentification and Security Layer : permet de faire des mécanismes d'authentification plus élaborés à base de clés
- Certificats sur SSL : Echange de certificats (clefs publiques/privées)

7.3.4.2. Les ACLs :

Les ACLs (Access Control Lists) interviennent après la notion de binding. Il sera possible de donner des droits de lecture, d'écriture (ou d'autres droits divers) sur des branches particulières de l'annuaire au compte connecté. Ceci permet de gérer finement les droits d'accès aux données.

7.3.4.3. Le chiffrement des communications (SSL/TLS) :

Le chiffrement des communications, via SSL (Secure Socket Layer, ou TLS - Transport Layer Security) est également une méthode de protection de l'information. Il est possible, avec la plupart des annuaires existants, de chiffrer le canal de communication entre l'application cliente et l'annuaire. Ceci permet de garantir (un minimum) la confidentialité des données et d'éviter qu'un tiers n'écoute les communications sur le réseau.

7.3.4.4. SASL :

SASL (Simple Authentication and Security Layer) est un mécanisme qui permet d'ajouter des méthodes d'authentification à des protocoles orientés connexion tels que LDAP ou IMAP. Il est défini dans la RFC 2222 ; l'implémentation la plus couramment utilisée est Cyrus-Sasl.

SASL donne la possibilité au client et au serveur de sélectionner quelle sera la méthode d'authentification utilisée. Ces méthodes sont extensibles via des plugins. Il permet également de mettre en place une couche de connexion sécurisée telle que SSL/TLS (sans rapport direct avec le chiffrement indépendant des connexions que nous avons cité ci-dessus).

7.3.5. Le modèle de duplication

Le protocole LDAP offrent des facilités pour dupliquer ou synchroniser les données entres plusieurs serveurs LDAP. Pour réaliser cela, il définit un modèle de duplication. Ce dernier définit comment échanger les informations d'un serveur à l'autre.

L'intérêt de dupliquer un serveur est de pallier une panne de l'un des serveurs, ou d'une coupure réseaux. Mais aussi pour répartir la charge du service et garantir une qualité de service.

Cependant ce modèle n'est pas encore standardisé. IETF est en préparation du protocole LDUP (Lightweight Directory Update Protocol) pour résoudre ce problème.

7.4. Déployer un service LDAP:[13]

Déployer un service d'annuaire LDAP nécessite en premier lieu une réflexion sur la nature des données que l'on y met, sur la manière dont on les récupère, sur l'utilisation que l'on compte en faire et sur la façon de gérer le tout. Dans notre cas, il s'agit de règle de politique. La mise en place d'un annuaire LDAP met donc en jeu plusieurs phases de conception que l'on va passer en revue.

7.4.1. Déterminer les besoins en service d'annuaire et ses applications :

Déployer un système d'annuaire se fait généralement sous la contrainte de la mise en place ou du remplacement d'une application. C'est alors que se pose la question d'élargir le service à d'autres types d'applications, la première venant à l'esprit étant un annuaire de régler. Cette phase consiste donc à prévoir toutes les applications possibles, actuelles ou futures, d'un annuaire LDAP.

7.4.2. Déterminer quelles données sont nécessaires :

Il s'agit d'inventorier la liste exhaustive des données que l'on souhaite inclure dans le système d'information et de déterminer ensuite par quelle source les obtenir et les maintenir à jour. Des aspects comme le format, la taille des données, leur confidentialité, leur pertinence, leur source (statique, dynamique...), leur pérennité, les personnes susceptibles de les fournir, de les maintenir et d'y accéder doivent être prises en compte lors de cette phase. De ce point de vue, c'est la plus délicate à franchir car elle implique d'autres intervenants. Il faut également se faire une idée précise sur la manière dont les données vont être maintenues à jour : synchronisation avec un SGBD, intervention manuelle, scripts automatiques...

7.4.3. Choisir son schéma :

Dans cette phase, on désigne le schéma, il s'agit de choisir, en fonction des données que l'on a retenues, quelles sont les classes d'objets et les types d'attributs qui s'en rapprochent le plus pour construire son annuaire LDAP.

La plupart du temps, les schémas standards, issus de X.500 et de LDAP conviennent aux besoins de modélisation. De plus, en fonction du logiciel que l'on choisira, des objets supplémentaires seront fournis. Il reste, au final, la possibilité de créer ses propres objets, spécifiques à ses besoins. En règle générale, il faut éviter de modifier le schéma existant car l'on risque de rendre son annuaire inutilisable par les applications clients ou les autres serveurs.

Il est préférable de créer une sous classe d'une classe d'objet existante et exploiter le mécanisme d'héritage d'attributs des classes objets.

Par exemple Supposant que l'on doit fournir un service vidéo Svideo. Cette opération doit être mise en application comme politique : Fournissez le service Svideo visuel pour les utilisateurs autorisés entre les points autorisés, mais seulement à des heures convenues.

On peut créer la classe d'objet ApprovedUsers fille de VideoServices dans laquelle on définira les attributs nécessaires à ses besoins :

```
objectclass ApprovedUsers
 superior VideoServices
 requires
        sn,
        cn
  allows
    user1,
    user2,
    user3,
    user4,
```

Dans tous les cas, il faut prévoir de documenter son schéma pour en faciliter la maintenance et l'évolution. Il faut proscrire également la désactivation de l'option de schema checking implantée dans la plupart des serveurs, qui permet de vérifier que les attributs saisis sont bien conformes au schéma que l'on a choisi.

7.4.4. Concevoir son espace (modèle) de nommage :

Cette étape consiste à définir comment les entrées de l'annuaire vont être organisées, nommées et accédées. L'objectif est de faciliter leur consultation et leur mise à jour mais aussi de prévoir leur duplication, leur répartition entre plusieurs serveurs ou leur gestion par plusieurs personnes. En fonction de ces priorités, on privilégiera tel ou tel espace de nommage.

Les paramètres qu'il faut prendre en compte lors de cette étude sont les suivants :

- Le nombre d'entrées prévu et son évolution ?
- La nature (type d'objet) des entrées actuelles et futures ?
- Va-t-il mieux centraliser les données ou les distribuer ?

- Seront-elles administrées de manière centrale ou faudra-t-il déléguer une partie de la gestion
- La duplication est-elle prévue ?
- Quelles applications utiliseront l'annuaire et imposent-elles des contraintes particulières ?
- Quel attribut utiliser pour nommer les entrées et comment garantir son unicité ?

Durant cette phase, nous allons choisir le modèle d'organisation des données, leur mode de désignation et le suffixe de notre organisation.

7.4.4.1. Le Directory Tree :[14]

Le modèle de nommage structuré en arbre hiérarchique de LDAP est repris du standard X.500. Ce dernier a été conçu dans l'optique d'un service global.

Le modèle LDAP, lui, n'impose pas une racine universelle du DIT car il renonce à être un service d'annuaire mondial et se limite à une petite communauté. Dans ce cadre, le modèle LDAP peut être plat (figure7.4.4.1.1), branché par type d'objet (figure 7.4.4.1.2).

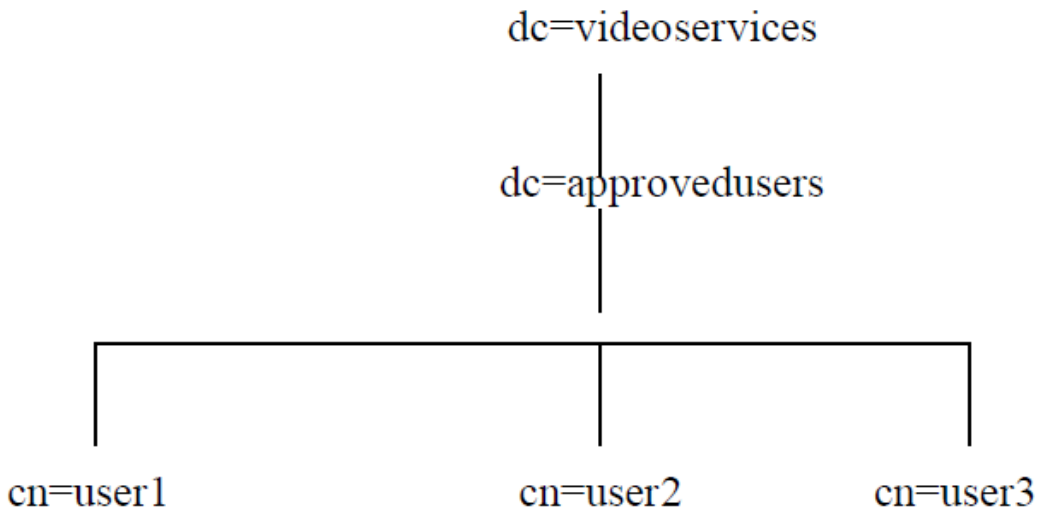


Figure 7.4.4.1.1.Espace de nommage plat

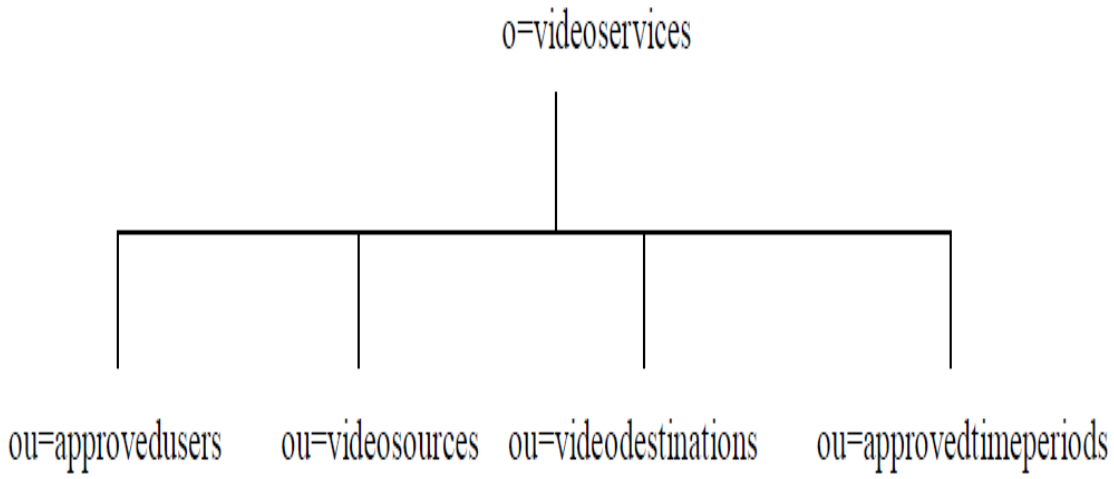


Figure 7.4.4.1.2. Espaces de nommage branché par type d'objet

En règle générale, en termes de performance, il vaut mieux avoir un arbre le plus plat possible. Par contre, en termes de facilité d'administration, il vaut mieux introduire du branchage par type d'objet ou par organisation. Cela facilite les mises à jour de données ou la mise en place de règles d'accès spécifiques à une partie de l'annuaire ou la répartition de la gestion de l'arbre entre plusieurs serveurs. Si votre organisation change souvent ou bien que le personnel est très mobile, le branchage par organisation est alors à proscrire.

7.4.4.2. Nommage des entrées :

La deuxième étape consiste à choisir l'attribut utilisé pour le DN de l'entrée, dans la partie RDN. Ce choix peut dépendre des applications clientes, mais il doit se faire surtout de manière à garantir l'unicité de la valeur utilisée.

7.4.4.3. Choix du suffixe :

La dernière étape consiste à choisir le suffixe. C'est en quelque sorte l'identifiant de l'annuaire. Son choix est important car, même si la base n'a qu'une vocation interne, elle peut à terme devenir en partie un maillon d'un annuaire global ou d'un système d'information. Le suffixe peut à l'extrême être une chaîne vide, mais dans l'optique d'une diffusion de son annuaire, on choisira, en général, un suffixe unique au monde. Pour cela, il est recommandé d'utiliser comme suffixe d'annuaire, le nom de domaine DNS de l'organisation.

7.4.5. Définir la topologie de son service :

Dans cette phase, il faut réfléchir sur la manière dont le service d'annuaire LDAP va être rendu en termes de performance, de fiabilité et de facilité de gestion. Il faut prendre en compte :

- Les applications qui vont utiliser l'annuaire et le nombre d'utilisateurs.
- Les capacités du logiciel serveur qui va être choisi.
- La topologie de son réseau.
- Le design de son espace de nommage.

La conception de la topologie du service d'annuaire LDAP est étroitement liée à celle de l'espace de nommage. Il est donc possible de devoir revenir sur l'une ou l'autre durant la phase de conception ou même celle d'exploitation, dans le cas d'un changement d'organisation interne ou de marque de logiciel serveur.

La question principale de cette phase est de déterminer si la base et sa gestion seront centralisées sur un seul serveur ou si elles devront être éclatées sur plusieurs serveurs. La deuxième étude porte sur le nombre de serveurs redondants à déployer et leur emplacement sur le réseau physique.

7.4.6. Le partitionnement :

Il consiste à séparer les données de l'annuaire sur plusieurs serveurs. Il peut être imposé par le volume d'entrées à gérer, leur gestion répartie sur plusieurs sites, les types d'accès au réseau physique ou le mode d'organisation de la société. Séparer les données ne veut pas dire forcément les dissocier : les standards LDAP et X500 définissent des moyens de les relier (recoller). Ce sont les services referral et replication.

7.4.6.1. Le service referral :

Les méthodes permettant de créer des liens entre des partitions d'annuaires sont appelées les knowledge references. Les knowledge references sont des mécanismes qui permettent de relier virtuellement des arbres entre eux, en indiquant à quel point de branchement d'un arbre vient se raccrocher un autre DIT (immediate Superior knowledge référence) et inversement quels sont les arbres qui viennent se raccrocher à tel ou tel point de branchement du sien (subordinate reference).

Ces liaisons permettent à un serveur de faire suivre les requêtes des utilisateurs lorsque l'objet recherché n'appartient pas à l'arbre qu'il gère. La résolution de nom est le mécanisme par lequel un serveur détermine quel objet de sa base est désigné par le DN qu'un client lui fournit. Si le DN est bien dans son contexte de nommage, il exécute la requête du client (search, modify, bind...), sinon il renvoie un signal " object not found ". Si l'objet n'est pas dans son espace de nommage, le serveur utilise alors ses liens knowledge reference soit pour faire suivre la requête vers le serveur qui peut fournir l'objet, soit pour indiquer au client lequel contacter.

La figure suivante montre le cas de figure de deux serveurs gérant chacun deux suffixes, dont un des deux est un lien vers l'autre serveur.

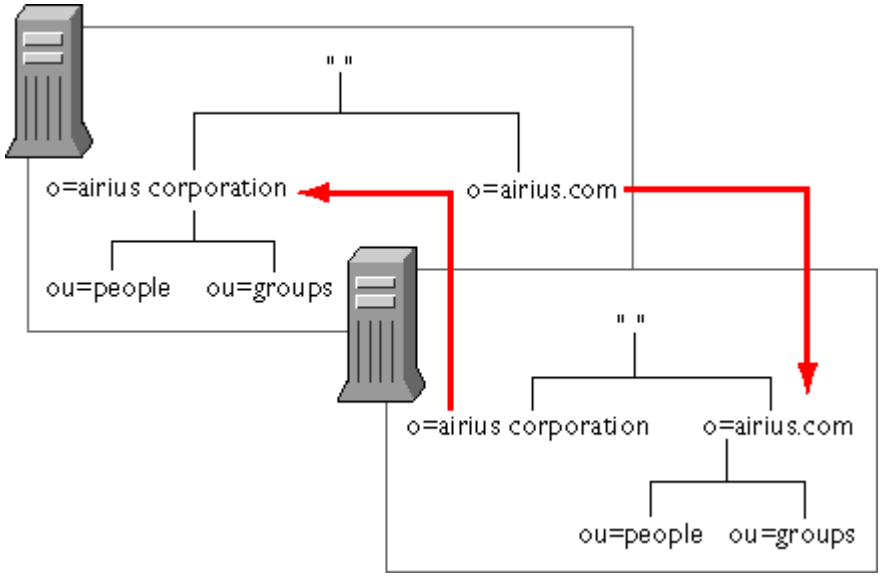


Figure 7.4.6.1. Les referrals

Les serveurs LDAP utilisent deux méthodes pour faire suivre les requêtes le long de ces liens:

1. Le Referral est une information que retourne au client le serveur LDAP, lorsque l'entrée recherchée n'appartient pas à son arborescence, lui indiquant vers quel serveur il doit reformuler sa requête. Il utilise pour cela les URLs LDAP. Le mécanisme de referral est standardisé dans le protocole LDAPv3.
2. Le chaînage (chaining) est un mécanisme où c'est le serveur qui se charge de contacter un autre serveur pour le compte du client et lui retourne la réponse. Le chaînage n'est pas un standard du protocole LDAP, il est plutôt utilisé dans les logiciels X.500.

7.4.6.2. La réplication service :

La duplication consiste à recopier le contenu de tout ou partie de son arbre sur un autre serveur son but :

- ✓ rapprocher le service du réseau physique des clients (performances),
- ✓ répartir la charge sur plusieurs serveurs (load balancing),
- ✓ assurer une redondance en cas de panne (disponibilité),
- ✓ gérer localement des entrées et les diffuser dans l'organisation (partitionnement).

La réplication service est le moyen d'assurer un service d'annuaire fiable, hautement disponible, et performant.

7.4.6.2.1. Différents modes de duplication : [16]

- **multi-serveurs :**

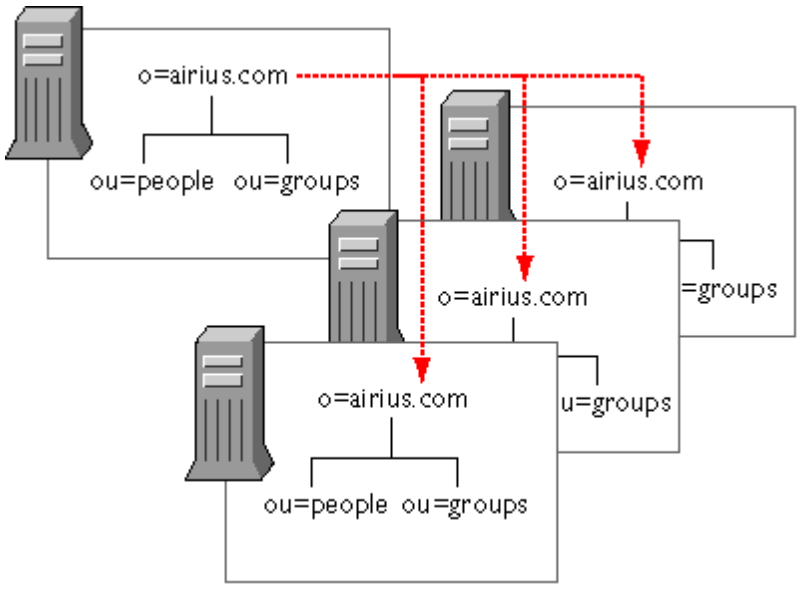


Figure 7.4.6.2.1.1 duplication multi-serveurs

Le supplier (read-write) duplique sur un ou plusieurs consumers (read-only).

- **En cascade :**

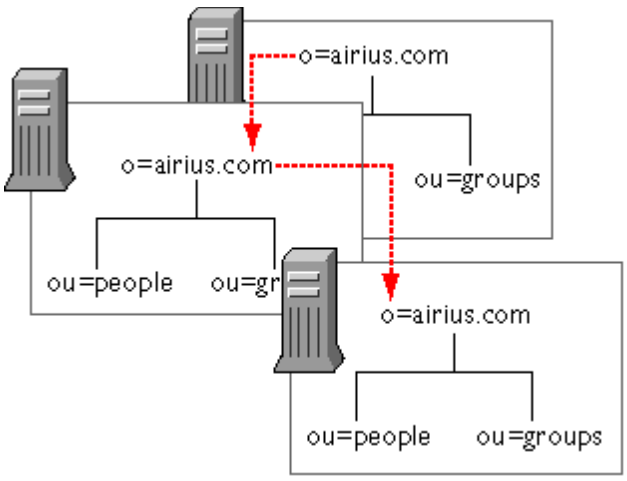


Figure 7.4.6.2.1.2.duplication en cascade

Le supplier duplique sur un consumer qui lui-même duplique sur un autre. Cas où les liaisons réseau entre sites sont de qualité variable.

- **Partie en arbre :**

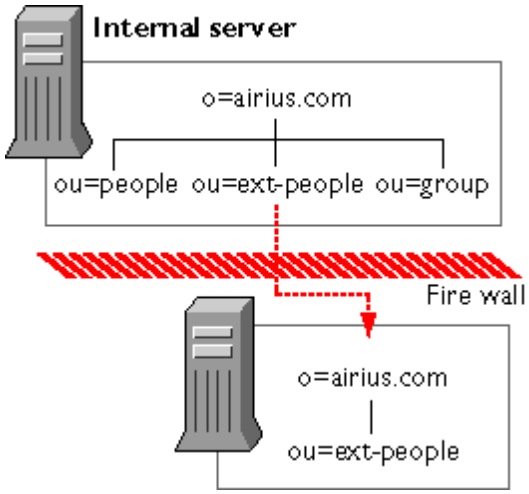


Figure 7.4.6.2.1.3. Duplication partie en arbre

Le supplier coupé de l'extérieur ne duplique qu'une branche publique de l'arbre sur un consumer accessible depuis l'internet.

- **Croisées :**

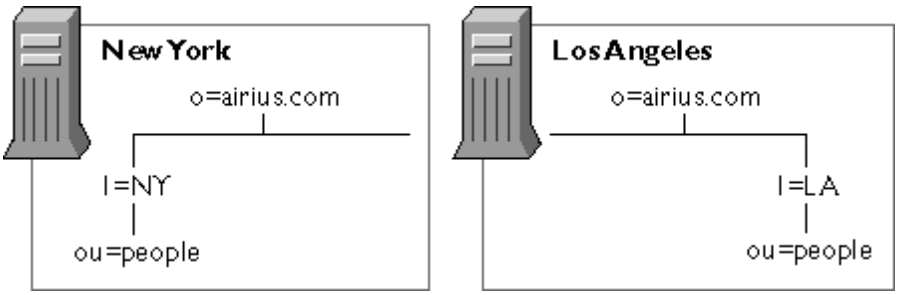


Figure 7.4.6.2.1.4.1.duplication croisées

La société Airius a deux agences à NY et LA qui gèrent chacune leur branche du serveur d'annuaire. Duplication est mise en œuvre pour ramener les branches distantes localement (performance) et assurer une redondance de tout l'arbre en local (disponibilité).

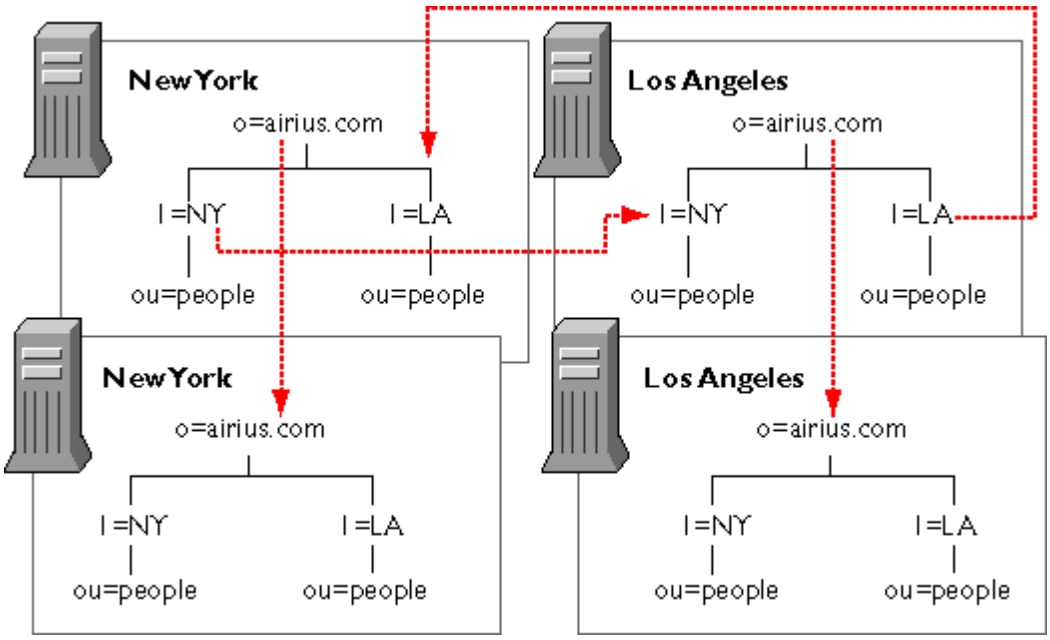


Figure 7.4.6.2.1.4.2.duplication croisées

Les branches sont dupliquées réciproquement sur chaque site. De plus, l'arbre entier est dupliqué en local.

7.5. Architecture LDAP:[17]

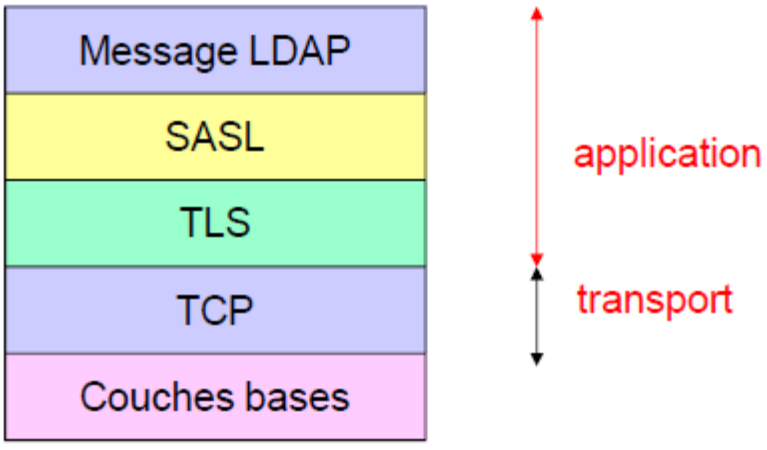


Figure 7.5.Architecture LDAP

7.5.1. Les principaux messages :

Le protocole LDAP fonctionne sur un modèle Requête-Réponse(s) : tant qu'un client ne sollicite pas le serveur, il ne lui transmet pas de réponse. En revanche, depuis l'ajout de l'extension IntermediateResponse à LDAP, une requête peut correspondre à plusieurs réponses. En revanche, il existe dans LDAP des évènements asynchrones, appelés "alertes". Une alerte est souvent attachée à un évènement, par exemple la fin du support de TLS lors d'une session protégée par StartTLS. Ces alertes peuvent émaner du client comme du serveur, indifféremment.

Ainsi, voici la description des principaux messages de requête et de réponses utilisés dans le protocole LDAP :

Message	Signification
bindRequest	Demande la connexion (authentifiée ou anonyme) à un annuaire
bindResponse	Réponse à la demande d'authentification
unbindRequest	Demande de déconnexion/fin de session
searchRequest	Demande à effectuer une recherche en fonction d'un filtre donné
searchResEntry	Réponse à une recherche, contenant une entrée LDAP
searchResDone	Dernier message indiquant la fin des réponses à une recherche
StartTLS Request	Demande de création d'une connexion chiffrée par une couche TLS émanant du client.
StartTLS Response	Réponse de la demande de création d'une connexion par couche TLS, continue par un handshake
TLS closure alert	Message envoyé pour demander/acquitter la fin d'une session protégée par une couche TLS
addRequest	Demande d'ajout d'une entrée dans l'annuaire
modifyRequest	Demande de modification d'une entrée de l'annuaire
modifyDNRequest	Demande la modification d'un Distinguished Name de l'annuaire (cf. section modèles de données)

Tableau 7.5.1.Les principaux messages

7.5.2. Séquençage pour une connexion classique / SSL :

Pour une connexion classique (non chiffrée) ou chiffrée par l'utilisation d'une couche SSL/TLS, les communications se font sur le modèle suivant :

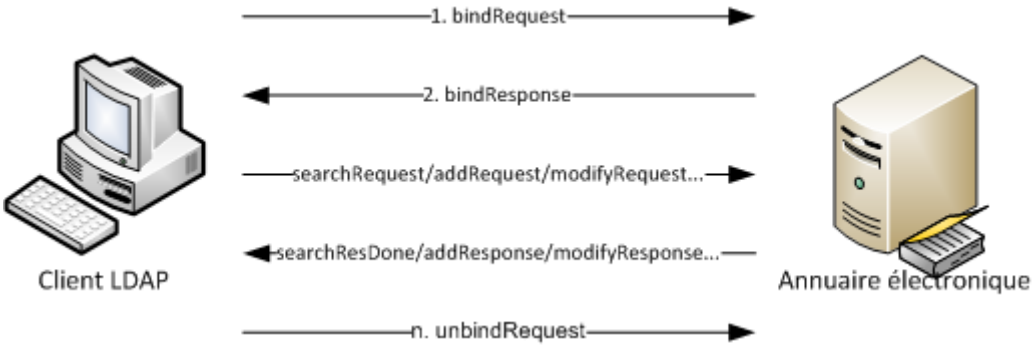


Figure 7.5.2.Séquençage pour une connexion classique / SSL

Les phases 1 et 2 représentent la connexion, avec ou sans authentification de l'utilisateur, au service d'annuaire. Une fois la connexion établie, la session est conservée jusqu'à la rupture de la connexion, ou jusqu'à ce qu'un message unbindRequest le demande explicitement. L'authentification se fait, dans LDAPv3, en utilisant SASL, ce qui permet de supporter de nombreux mécanismes d'authentification (Plaintext, NTLM, Anonymous, Digest-MD5...).

Pendant la durée d'une session, tous les messages sont pris en compte en fonction des autorisations d'accès accordées à l'utilisateur authentifié ou à la connexion anonyme.

Remarque : la connexion SSL est implicite, et le handshake a lieu avant l'échange de paquets au format LDAP. C'est pourquoi ces informations ne sont pas représentées sur la figure ci-dessus. Ceci ne change en aucune façon le séquençage des messages LDAP.

7.5.3. Séquençage pour une connexion TLS :

Dans ce modèle, la connexion chiffrée en TLS est explicite, et la demande de sécurisation peut se faire à [presque] tout moment. De même, il est possible de retourner en mode non chiffré à presque tout moment à l'aide de l'alerte TLS closure alert.

Cependant, pour des raisons de sécurité, il est fortement conseillé d'envoyer le message StartTLS avant l'opération bindRequest. Autrement, les informations liées au processus d'authentification seront envoyées en clair, ce qui exposerait ces informations à des attaques de type wiretapping (écoute du réseau).

Le modèle recommandé d'utilisation de StartTLS est donc le suivant :

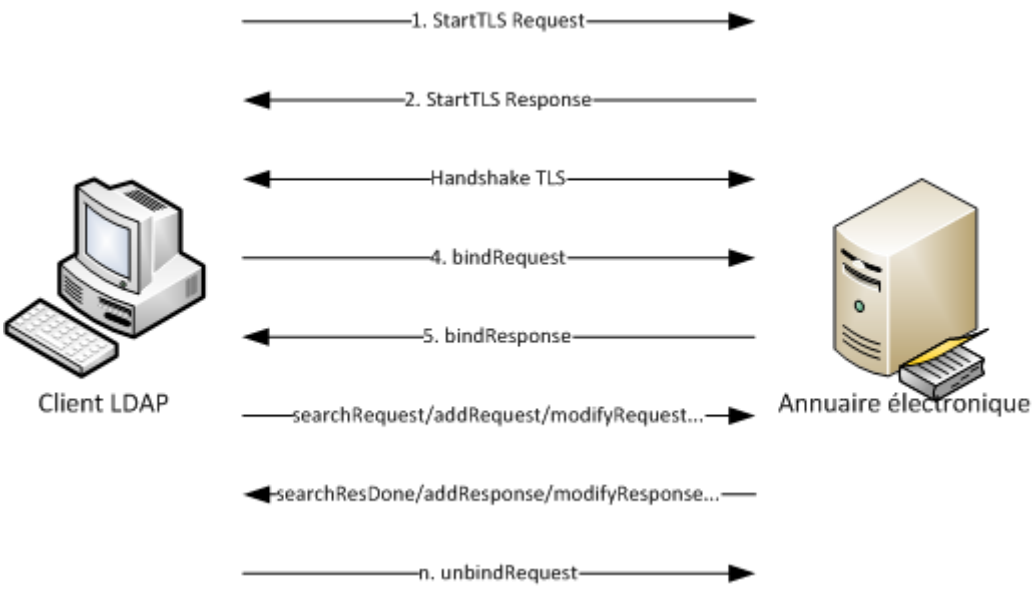


Figure 7.5.3.Séquençage pour une connexion TLS

L'opération StartTLS peut avoir lieu à n'importe quel moment excepté :

- Si TLS est déjà utilisé sur la session,
- Si une négociation SASL à étapes multiples est en cours,
- Et si des réponses sont en cours d'envoi au client, suite à une requête de sa part.
-

7.6. Les différents types d'architecture :

Un serveur LDAP peut constituer un SPOF (*Single Point of Failure*) dans l'architecture logicielle d'une entreprise, dans la mesure où la plupart des applications s'appuient sur l'annuaire LDAP pour l'authentification et la récupération d'informations concernant les utilisateurs et le matériel.

7.6.1. Maître-esclave(s) (ou Single Master Replication) :[19]

Dans ce modèle, le maître unique est le seul annuaire sur lequel les modifications peuvent être opérées. Les esclaves ne permettent que la lecture d'informations, par exemple pour l'authentification des utilisateurs, ou pour les applications ne stockant pas d'information dans l'annuaire.

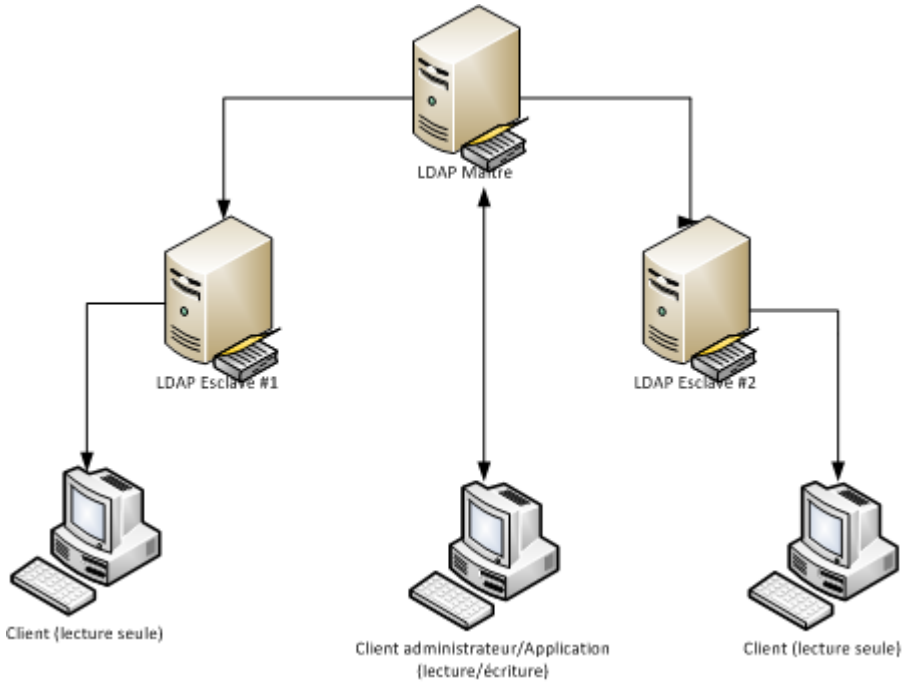


Figure 7.6.1. Maître-esclave(s) (ou Single Master Replication)

7.6.2. Multi-maîtres (ou Multiple Master Replication) :[19]

Dans ce modèle, plusieurs maîtres cohabitent ensemble sur le réseau. Des modifications peuvent être réalisées sur tous les annuaires du réseau, et les mises à jour sont donc bidirectionnelles.

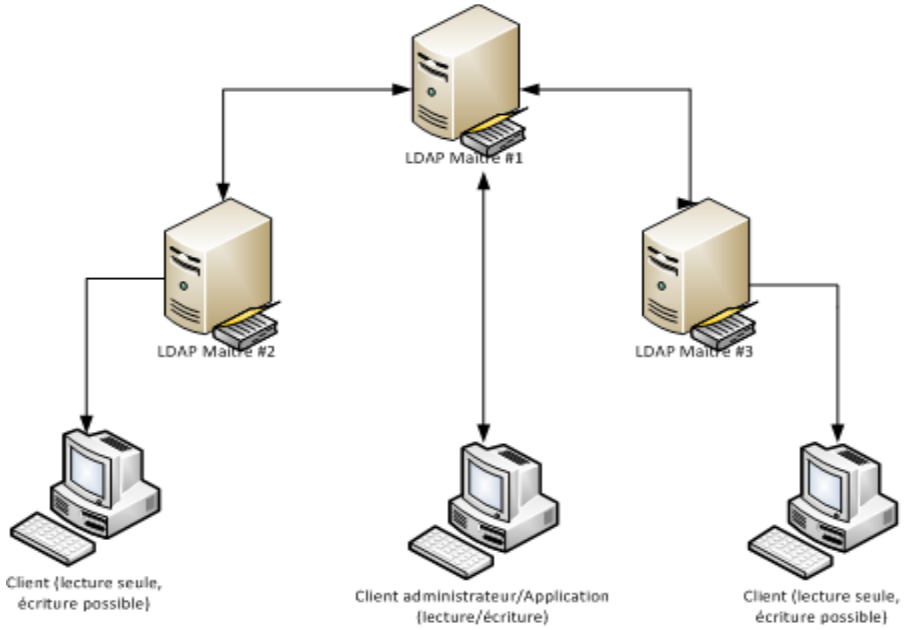


Figure 7.6.2.Multi-maîtres (ou Multiple Master Replication)

7.6.3. Architecture mixte :

Dans ce modèle, on fait cohabiter plusieurs maîtres avec un modèle maître esclave. Dans ce cas, les modifications sont bidirectionnelles entre les maîtres LDAP, et unidirectionnelle entre les maîtres et les esclaves.

Dans le cas de configuration, on peut alors prévoir des annuaires maîtres modifiables par les serveurs d'applications et les administrateurs, et des serveurs esclaves dédiés aux processus de login, et de consultation d'informations issues de l'annuaire

7.7. Les principale APIs pour LDAP :[09]

LDAP a été normalisé pour la première fois il y a 15 ans . Aussi, de nombreuses APIs clientes pour LDAP ont été développées, pour tous les langages de développement existant et possédant des capacités de connexion réseau, parmi lesquels :

- Java (intégré dans JNDI, Novell Java LDAP classes)
- C (Mozilla LDAP C SDK)
- C++ (OpenLDAP C++ API)
- PERL (Perl-LDAP classes Net::LDAP)
- Python (LDAP Client API for Python)
- Ruby (Ruby/LDAP)
- .NET (C#, VB.NET, J#, ...) (intégré via les classes System.DirectoryServices)
- ...

Pour chaque langage, il existe une variété de bibliothèques permettant l'utilisation de LDAP. Les informations des sections suivantes concernent les principaux patterns rencontrés dans ces librairies.

7.8. Format des URLs dans LDAP :[09]

Pour définir plus simplement une référence entre plusieurs annuaires LDAP, ou pour indiquer un chemin LDAP dans lequel on souhaite rechercher un objet, une syntaxe par URL a été définie par l'IETF.

Comme pour toutes URLs, les URLs ldap sont de la forme :

Protocole://nom_ou_ip [: port]/ [ressource] avec pour protocole :

- ldap pour LDAP et LDAP TLS (port 389 si non précisé)
- ldaps pour LDAP sur SSL/TLS (port 636 si non précisé)

Et comme indication de ressource le Distinguished Name (DN) soit de l'annuaire auquel on souhaite se connecter, soit de l'entrée à laquelle on souhaite faire référence (pour les références inter-annuaires).

Voici quelques exemples d'URL LDAP, valides ou fictives :

- ldap://ldap.mit.edu:389/dc=mit,dc=edu (annuaire LDAP du MIT)
- ldap://ldap.mit.edu/ou=employees,dc=mit,dc=edu (groupe des salariés de l'annuaire LDAP du MIT)
- ldap://ldap.mit.edu/cn=Patricia A White+employeeNumber=f3ba36cea9368a352f692ca87573b3d4,ou=employees,dc=mit,dc=edu (Entrée LDAP représentant Patricia A. White, salariée du MIT)
- ldaps://192.168.0.3/ou=groups,dc=fictivecorp,dc=local (URL fictive d'un annuaire local auquel on accède en SSL

7.9. Format de recherche :

Les filtres de recherche LDAP sont définis par une syntaxe assez peu usuelle, sur le modèle de la notation polonaise inversée.

Les principes de base de l'écriture d'une requête sont les suivants :

- La requête doit être entourée par des parenthèses (...)
- Chaque composante de la comparaison doit être entourée par des parenthèses (xxx op.yyy)
- La partie précédant l'opérateur de comparaison correspond au nom d'un attribut LDAP
- La partie droite correspond à la représentation textuelle de la valeur à comparer
- Pour échapper un caractère, il faut utiliser l'anti-slash "\" suivi de deux digits représentant le code hexadécimal dans la table ASCII 8bits du caractère à représenter
- Pour les compositions (et/ou/non), l'opérateur se place avant les éléments à comparer (op_cmp (...) (...) (...)...)

Les opérateurs de comparaison les plus courants sont les suivants :

Opérateur	Signification
=	Opérateur d'égalité (peut être case insensitive en fonction du comparateur défini dans le schéma)
>=	Supérieur ou égal à...
<=	Inférieur ou égal à...
~=	Approximativement égal à (par exemple, par application de la méthode SOUNDEX)

Tableau 7.9.1. Les opérateurs de comparaison

Les opérateurs booléens les plus courants sont les suivants :

Opérateur booléen	Signification
&	ET/AND
|	OU/OR
!	NON/NOT

Tableau 7.9.2.Les opérateurs booléens

De plus, il est possible d'utiliser le caractère "*" pour matérialiser 0 à n caractères différents constituant une chaine (ou partie de chaîne) recherchée.

Voici quelques exemples de filtres de recherche :

Filtre cherchant toutes les entrées où l'attribut ou vaut n'importe quelle chaine se finissant par users

(ou=*users)

Filtre cherchant toutes les unités organisationnelles dont l'attribut ou se termine par users

(& (objectClass=organizationalUnit)(ou=users*))

Filtre cherchant toutes les unités organisationnelles et les personnes dont l'attribut cn existe

(& (cn=*)(|(objectClass=organizationalUnit)(objectClass=inetOrgPerson)))

Remarque : utilisé sans précision, attribut=* signifie que l'attribut "a une valeur". Si l'attribut est optionnel (m-may), la recherche ne retournera que les entrées pour lesquelles l'attribut a une valeur définie.

8. Exemple d'annuaire (active directory) :

8.1. Définition :[18]

Active directory est un annuaire au sens informatique et technique chargé de répertorier tout ce qui touche au réseau comme le nom des utilisateurs, des imprimantes, des serveurs, des dossiers partagés, etc. L'utilisateur peut ainsi trouver facilement des ressources partagées, et les administrateurs peuvent contrôler leurs utilisations grâce à des fonctionnalités de distribution, de duplication, de partitionnement et de sécurisation des accès aux ressources répertoriées.

Il est possible d'interroger l'annuaire pour obtenir une liste des objets possédant des attributs, en formulant par exemple une requête du type : " Trouver toutes les imprimantes couleur de l'étage 2 ".

Active Directory supporte les protocoles suivants :

- **TCP/IP** : c'est le protocole de transport réseau.
- **DNS** : l'espace de noms Active Directory s'appuie sur ce service
- **DHCP** : Ce protocole va permettre de distribuer les adresses IP et de configurer les clients dans DNS.

- **SNTP** (Single Network Time Protocol) est le protocole de distribution et de synchronisation de l'heure. Il est impératif que toutes les machines du domaine Windows disposent de la même heure afin de synchroniser leurs actions.
- **LDAP** (Lightweight Directory Access Protocol) : ce protocole permet de gérer l'annuaire d'Active Directory et d'opérer des recherches dans sa base de données.[20]

8.2. Historique :[18]

Active Directory est le résultat de l'évolution de la base de données de comptes du domaine (principaux de sécurité) SAM (Security Account Manager) et une implémentation de LDAP, protocole d'accès à l'annuaire éponyme, lui-même dérivé de X.500. SAM est une base de données sommaire et plane (sans notion de hiérarchie). La technologie de stockage est fondée sur le stockage du registre Windows, la base SAM constituant à elle seule une ruche, ce qui physiquement correspond à un fichier portant le nom SAM, tout comme les fichiers system et software.

Active Directory revoit complètement le stockage des informations de sécurité du domaine, de la structure de la base jusqu'au niveau sémantique.

Tout d'abord, le moteur de base de données retenu est ESENT, également connu sous le nom de Jet Blue, pour lever l'ambiguïté avec les bases de données Microsoft Access utilisant le moteur Jet Red. Le prédécesseur d'ESE98 n'était autre qu'ESE97, le moteur de base de données utilisé pour l'annuaire Exchange 5.5. La différence principale entre ESENT et ESE98 est la taille des pages utilisées et la taille des journaux de transaction.

Au niveau sémantique, Active Directory est un annuaire LDAP, tout comme l'annuaire d'Exchange 5.5. Exchange 5.5 n'est pas pour autant le seul antécédent technologique à Active Directory, citons également l'annuaire Novell NDS, qui a constitué également un saut technologique en comparaison avec le précédent système Bindery.

Active Directory peut donc être considéré comme la réponse technologique aux technologies d'annuaire Novell, les deux systèmes étant dérivés de X.500.

8.3. Caractéristique :[18]

- Active Directory permet de recenser toutes les informations concernant le réseau, que ce soient les utilisateurs, les machines ou les applications.
- Active Directory constitue ainsi le noyau central de toute l'architecture réseau et a vocation à permettre à un utilisateur de retrouver et d'accéder à n'importe quelle ressource identifiée par ce service.
- Active Directory est donc un outil destiné aux utilisateurs mais dans la mesure où il permet une représentation globale de l'ensemble des ressources et des droits associés il constitue également un outil d'administration et de gestion du réseau.

- Il fournit à ce titre des outils permettant de gérer la répartition de l'annuaire sur le réseau, sa duplication, la sécurisation et le partitionnement de l'annuaire de l'entreprise.
- La structure d'Active Directory lui permet de gérer de façon centralisée des réseaux pouvant aller de quelques ordinateurs à des réseaux d'entreprises répartis sur de multiples sites.

8.4. Principe de fonctionnement :

Active Directory permet de représenter et de stocker les éléments constitutifs du réseau (les ressources informatiques mais également les utilisateurs) sous formes d'objets, c'est-à-dire un ensemble d'attributs représentant un élément concret. Les objets sont organisés hiérarchiquement selon un schéma (lui-même stocké dans l'annuaire) définissant les attributs et l'organisation des objets. Le service d'annuaire Active Directory permet de mettre ces informations à disposition des utilisateurs, des administrateurs et des applications selon les droits d'accès qui leur sont accordés.

8.5. Structure d'Active Directory :[20]

Les objets d'Active Directory (Utilisateurs, Groupes, Ordinateurs, etc.) correspondent à des classes, c'est-à-dire des catégories d'objets possédant les mêmes attributs. Ainsi un objet est une « instanciation » d'une classe d'objet, c'est-à-dire un ensemble d'attributs avec des valeurs particulières.

Lorsqu'un objet contient d'autres objets, on le qualifie de conteneur. Les conteneurs permettent de regrouper les objets dans une optique d'organisation. A l'inverse si l'objet est au plus bas niveau de la hiérarchie, il est qualifié de feuille.

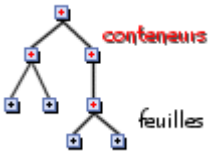


Figure 8.5.1.Exemple d'arbre

La hiérarchie composée de l'ensemble des conteneurs (noeuds) et des feuilles est appelée arbre. La notion d'arbre est étroitement liée à la notion de domaine, permettant de circonscrire des ressources informatiques dans un même périmètre de sécurité. Un domaine est ainsi constitué d'un ensemble défini d'éléments et possède une politique de sécurité (contrôles d'accès) qui lui est propre. Deux domaines (ou plus) possédant le même schéma peuvent établir entre eux des relations d'approbation (relations de confiance) bidirectionnelles et transitives basées sur le protocole Kerberos. L'ensemble des domaines

reliés entre eux hiérarchiquement par des relations d'approbation constituent un arbre de domaines (appelé arbre). Le domaine situé au sommet de la hiérarchie est appelé « domaine racine » et les domaines situés en dessous sont des sous-domaines. Les domaines d'un même arbre partagent nécessairement le même espace de nom.

Exemple :[20]

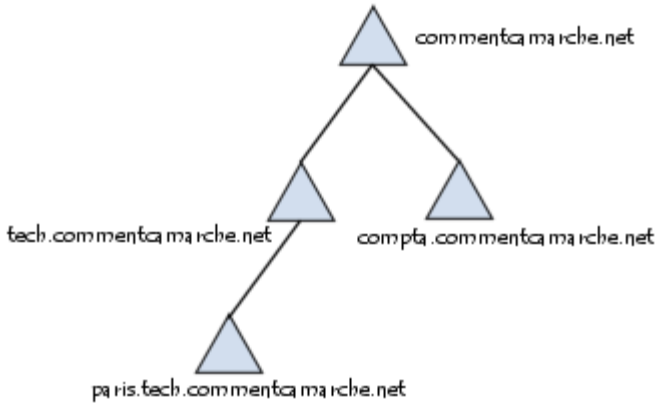


Figure8.5.1.Exemple de relation d'approbation

Ainsi les domaines commentcamarche.net, tech.commentcamarche.net, compta.commentcamarche.net et support.commentcamarche.net font partie du même espace de nom et constituent un arbre lorsqu'ils sont liés par des relations d'approbation. Le domaine commentcamarche.net représente ici le domaine racine. On appelle « forêt » le regroupement (par relations d'approbation) de plusieurs arbres possédant le même schéma mais ne possédant pas nécessairement le même espace de nom, afin par exemple de joindre les annuaires de deux entreprises.

8.6. Avantages d'Active Directory :[21]

- Intégration avec DNS
- Flexibilité des requêtes
- Capacités d'extension
- Administration par stratégie
- Adaptabilité
- Réplication d'informations
- Sécurité des informations
- Interopérabilité.

Conclusion :

Il est de plus en plus courant de trouver dans la littérature, comme dans des pages web, de nombreuses références à LDAP. Et il semble bien évident que ce phénomène s'accroisse de plus en plus. En effet, il est de plus en plus difficile d'imaginer un monde sans annuaires partagés et en particulier sans annuaires basés sur LDAP. Ce protocole semble s'être imposé naturellement comme le standard du domaine. Les usages de LDAP couvrent de nombreux aspects de la vie de l'entreprise, et ce, quelle que soit sa taille. LDAP est en effet utilisé pour la diffusion d'annuaires de type « pages blanches », commence à remplacer de plus en plus souvent les serveurs NIS, permet de jouer le rôle de serveur de distribution de certificats, permet l'authentification des utilisateurs, etc.

Dans ce chapitre nous avons présenté les grandes généralités liées à l'annuaire, nous avons présenté les concepts nécessaires pour une bonne connaissance de l'annuaire LDAP et nous avons fait un aperçue sur active directory.

CHAPITRE III : ANALYSE ET CONCEPTION

Introduction :

Cette phase de développement de notre application repose sur la modélisation des éléments pertinents de notre application mais aussi leurs relations avec l'extérieur (acteur).Cette modélisation consiste à introduire un ensemble de concepts proposés par le langage UML.
La figure suivante représente la démarche de modélisation UML que nous avons choisie pour concevoir notre application

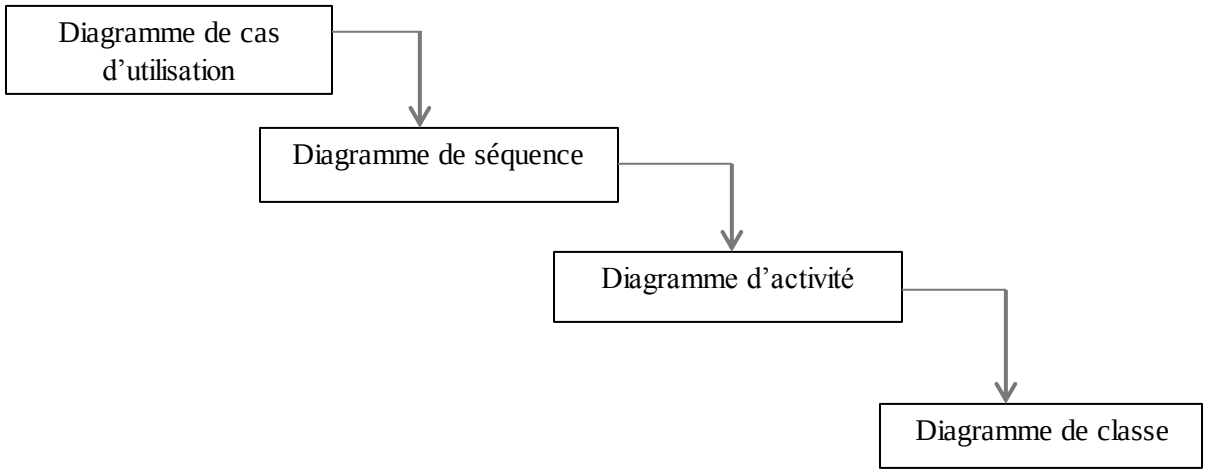


Figure1 : Les démarches de modélisation de notre application

1. Les objectifs d'application :

Le but principal de notre projet est de concevoir et de réaliser un outil d'interrogation pour un administrateur, pour leur facilite lors taches. Pour ce faire nous avons opté pour l'utilisation d'active directory et le Windows server 2008.

2. Analyse :

2.1.Identification des acteurs et leurs taches :

Un acteur représente un ensemble cohérent de rôles joués par des entités externes (Utilisateurs humains, dispositif matériel ou autre système) qui interagissent directement avec le système étudié.

Identification des acteurs :

- **Acteurs principaux :**
  - Administrateur car l'application s'exécutant uniquement au niveau serveur :

- Les taches :
  - Les taches de l'administrateur sont les suivantes :
    1. Gérer les utilisateurs
    2. Gérer les groupes

3. Gérer les ordinateurs
4. Gérer les contacts

2.2. Le diagramme de cas d'utilisation :

Les cas d'utilisations sont une technique puissante pour traduire le comportement du système. C'est un outil formel qui permet d'exprimer les interactions et les dialogues des acteurs.

Relations entre les cas d'utilisation

Afin d'optimiser la formalisation des besoins en ayant recours à la réutilisation des cas d'utilisation, trois relations peuvent décrites entre cas d'utilisation :

- Une relation d'inclusion formalisée par le mot clé « **include** ».

Un cas A inclut un cas B si le comportement décrit par le cas A inclut le comportement du cas B : le cas A dépend du cas B. Lorsque A est sollicité, B l'est obligatoirement comme une partie de A. Les inclusions permettent essentiellement de factoriser une partie de la description d'un cas d'utilisation qui serait commune à d'autres cas d'utilisation.

- Une relation d'extension formalisée par le mot clé « **extend** ».

On dit qu'un cas d'utilisation A étend un cas d'utilisation B lorsque le cas d'utilisation A peut-être appelé au cours de l'exécution du cas d'utilisation B. Exécuter B peut éventuellement entrainer l'exécution de A. Contrairement à l'inclusion, l'extension est optionnelle.

2.2.1. Le diagramme de cas d'utilisation globale :

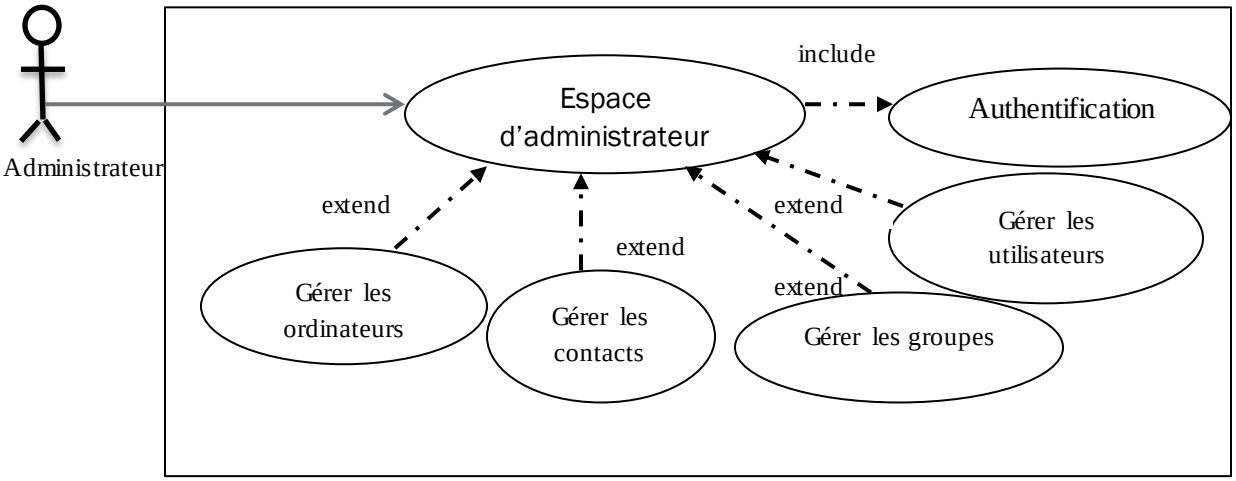


Figure 2.2.1. Le diagramme de cas d'utilisation globale

2.2.2. Diagramme détaillé du cas d'utilisation « gérer les groupes » :

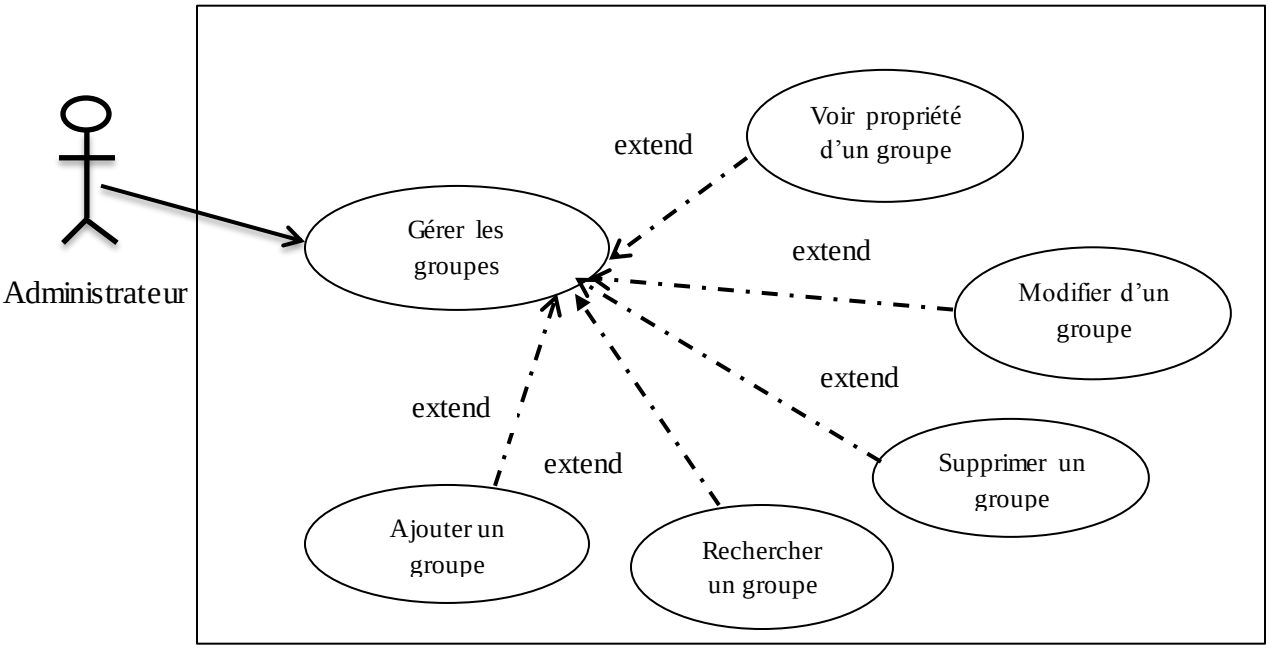


Figure 2.2.2. Diagramme détaillé du cas d'utilisation « gérer les groupes»

2.2.3. Diagramme détaillé du cas d'utilisation « gérer les utilisateurs» :

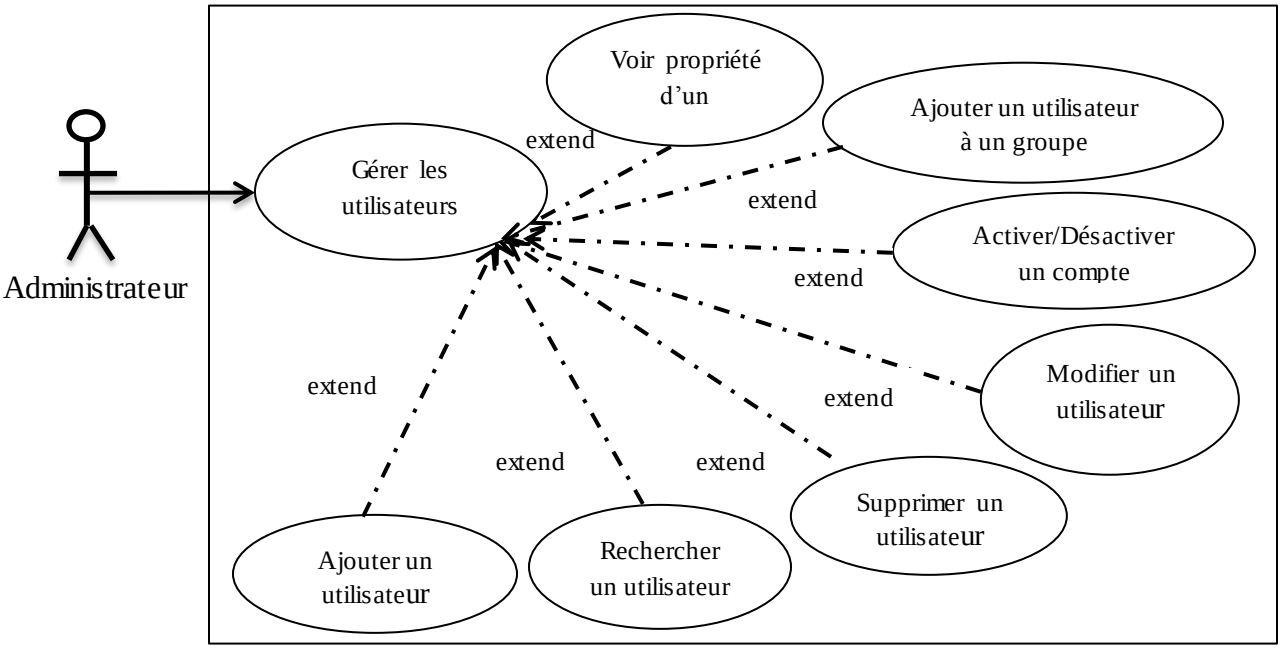


Figure 2.2.3. Diagramme détaillé du cas d'utilisation « gérer les utilisateurs »

2.2.4. Diagramme détaillé du cas d'utilisation « gérer les ordinateurs» :

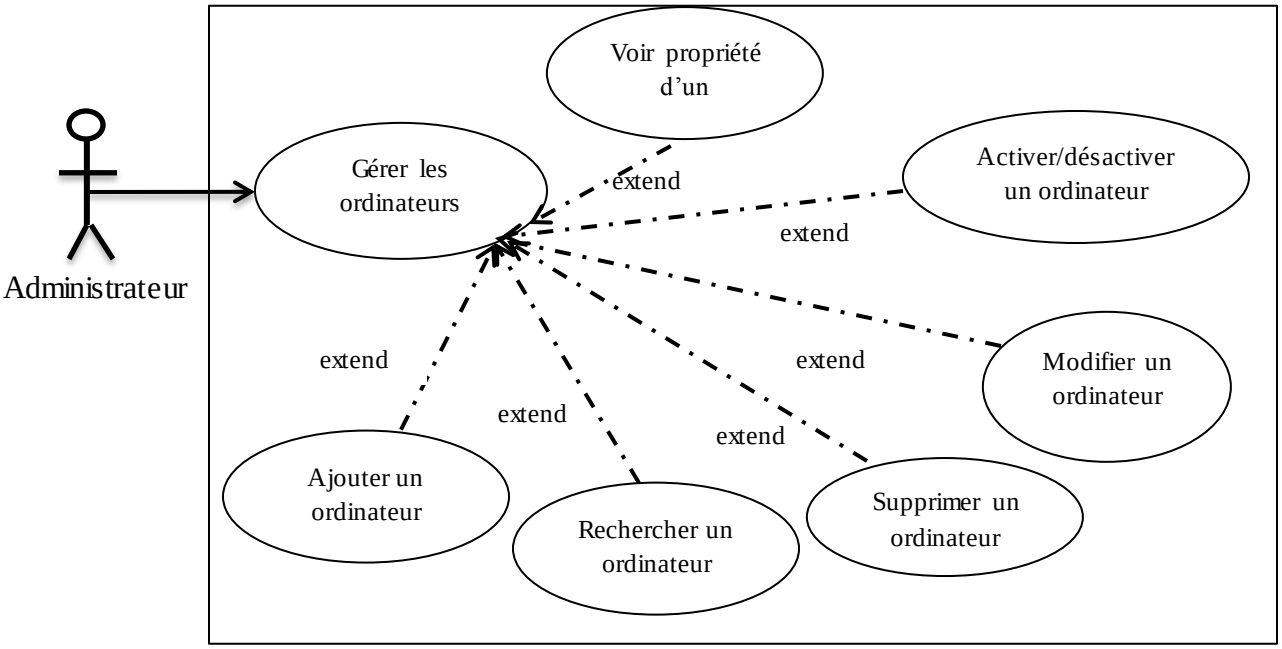


Figure 2.2.4.Diagramme détaillé du cas d'utilisation « gérer les ordinateur »

2.2.5. Diagramme détaillé du cas d'utilisation « gérer les contacts» :

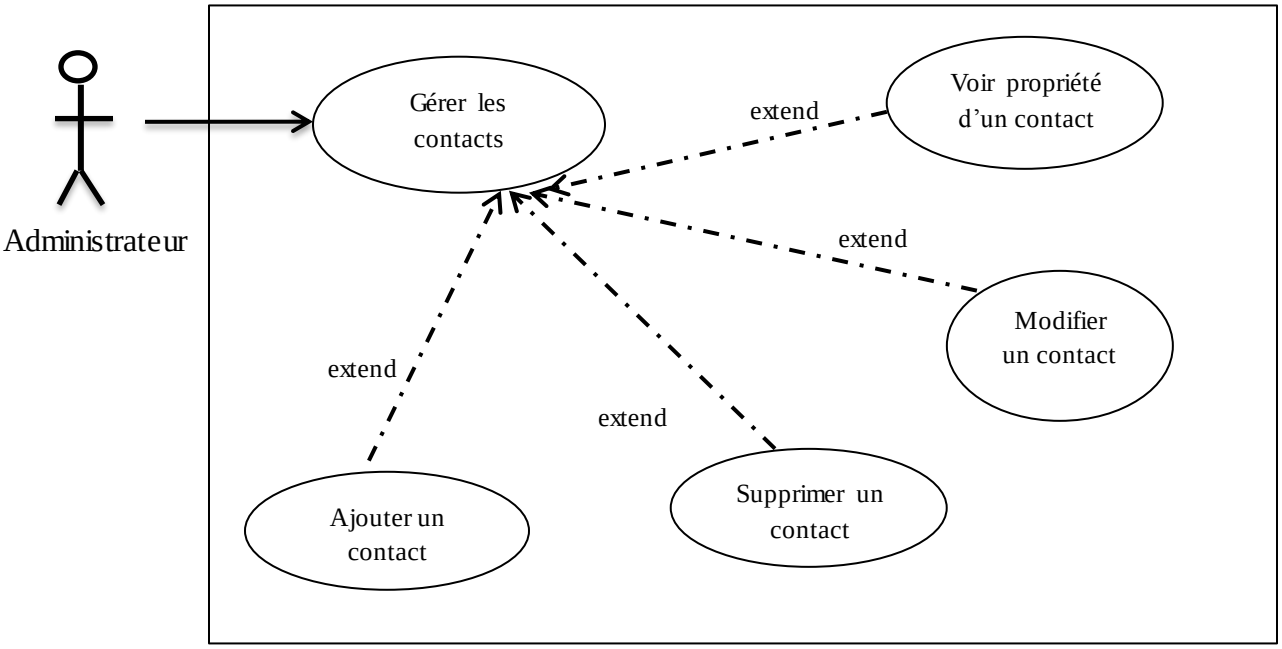


Figure 2.2.5.Diagramme détaillé du cas d'utilisation « gérer les contacts »

3. Conception :

3.1. Diagramme de séquence :

Les diagrammes de séquences sont formés avec des classes traduisant la dynamique du système et qui seront utilisés par la suite dans l'activité de conception.
Il permet de mieux visualiser la séquence des messages par une lecture de bas en haut. L'axe vertical représente le temps, et l'axe horizontal représente les objets qui collaborent. Une ligne verticale en pointillé est attachée à chaque objet et représente sa ligne de vie.

3.1.1. Diagramme de séquence de la création d'un compte utilisateur :

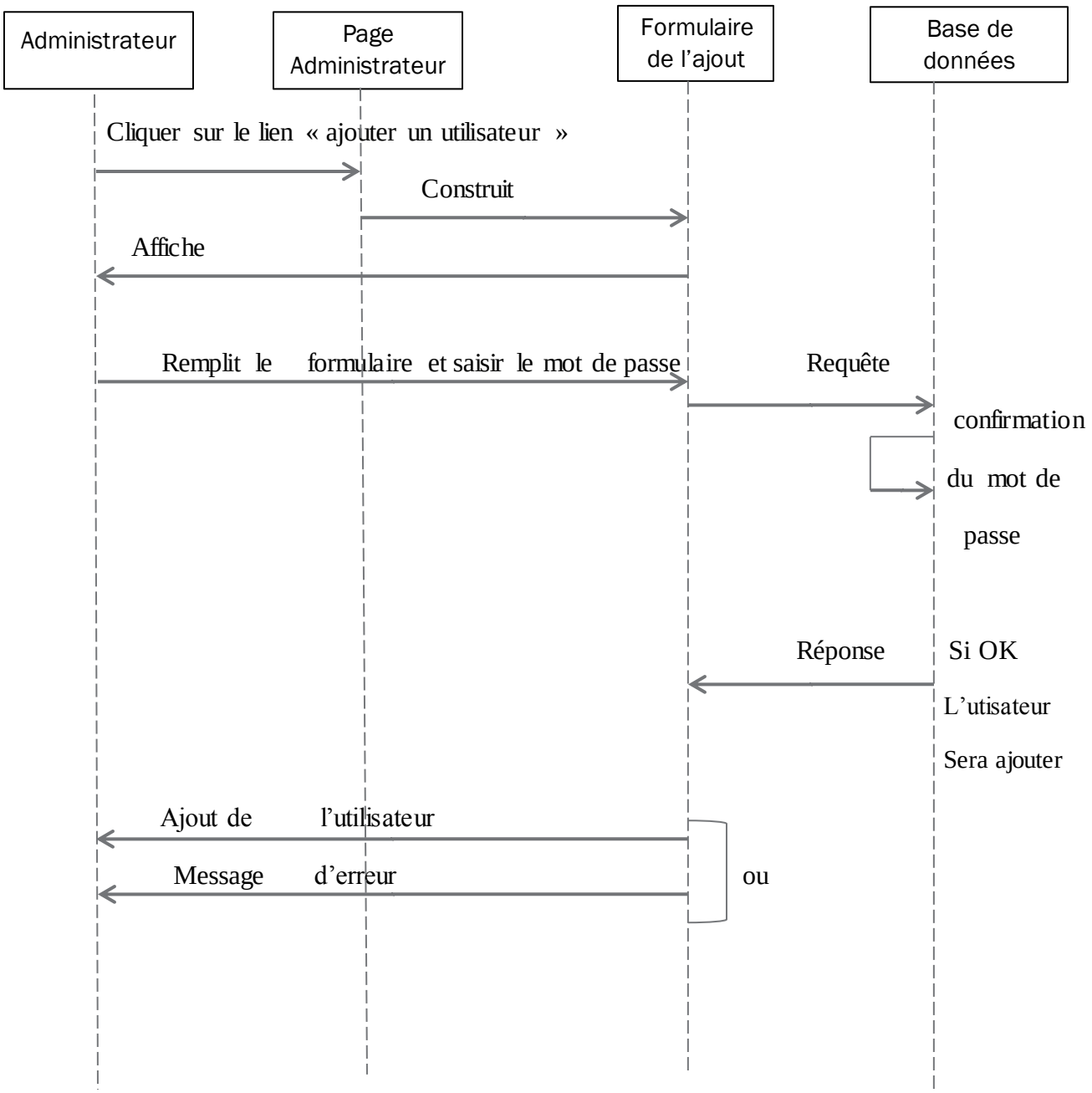


Figure 3.1.1. Diagramme de séquence pour le cas d'utilisation ajouter un utilisateur

3.1.2. Diagramme de séquence de recherche d'un utilisateur :

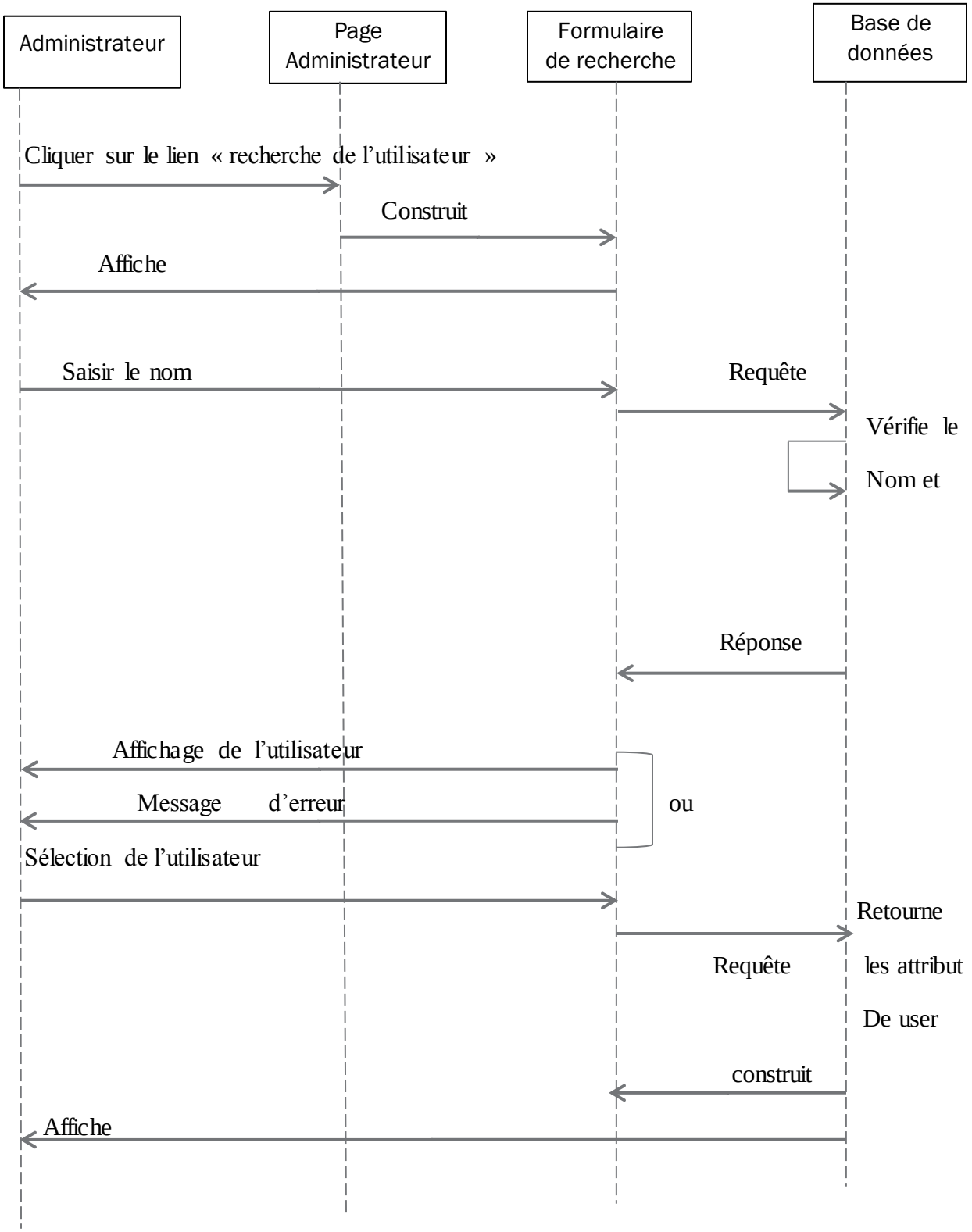


Figure 3.1.2 Diagramme de séquence pour le cas d'utilisation rechercher un utilisateur :

3.1.3. Diagramme de séquence modifier un groupe :

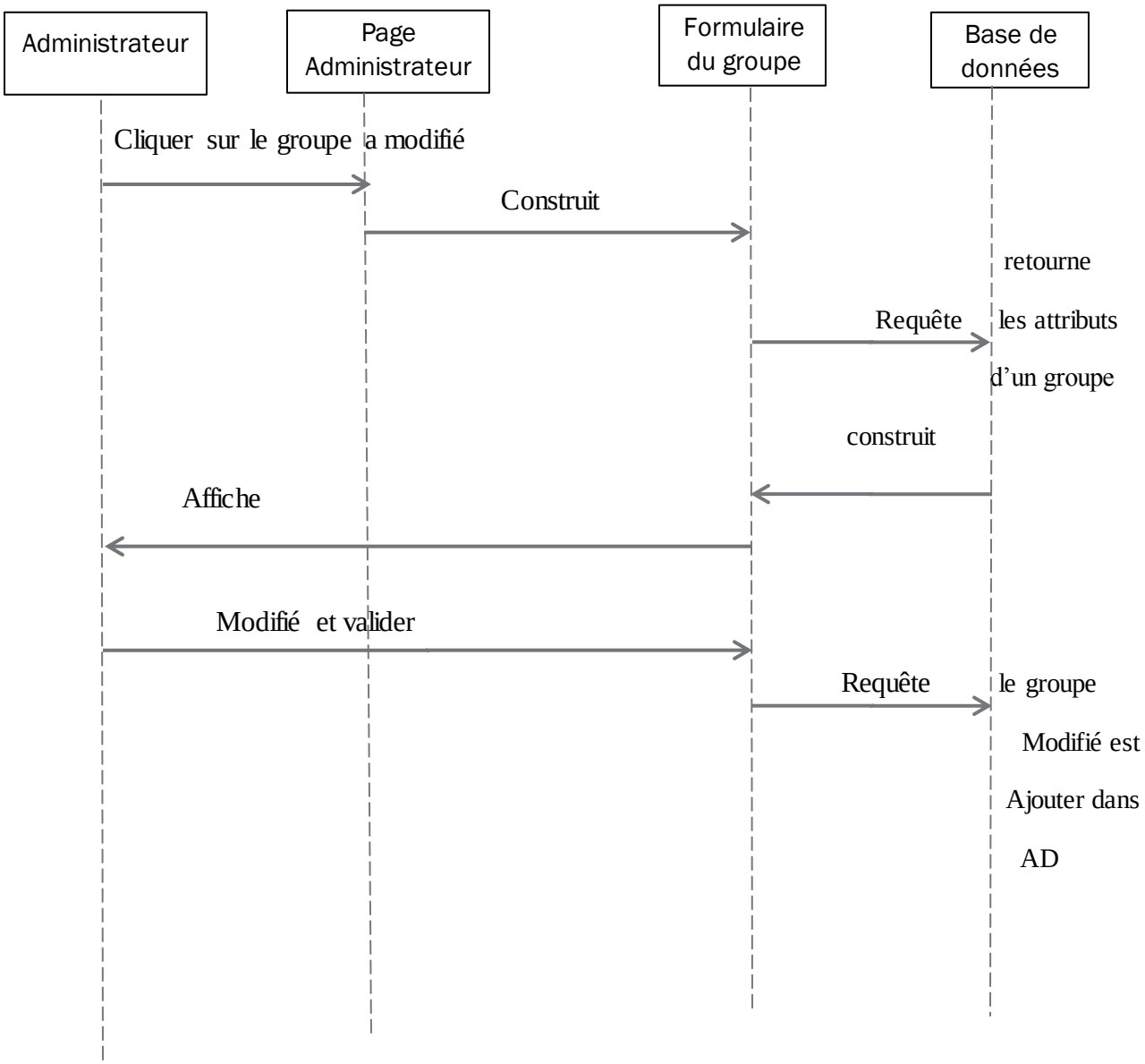


Figure 3.1.3. Diagramme de séquence pour le cas d'utilisation modifier un groupe

3.1.4. Diagramme de séquence supprimer un contact:

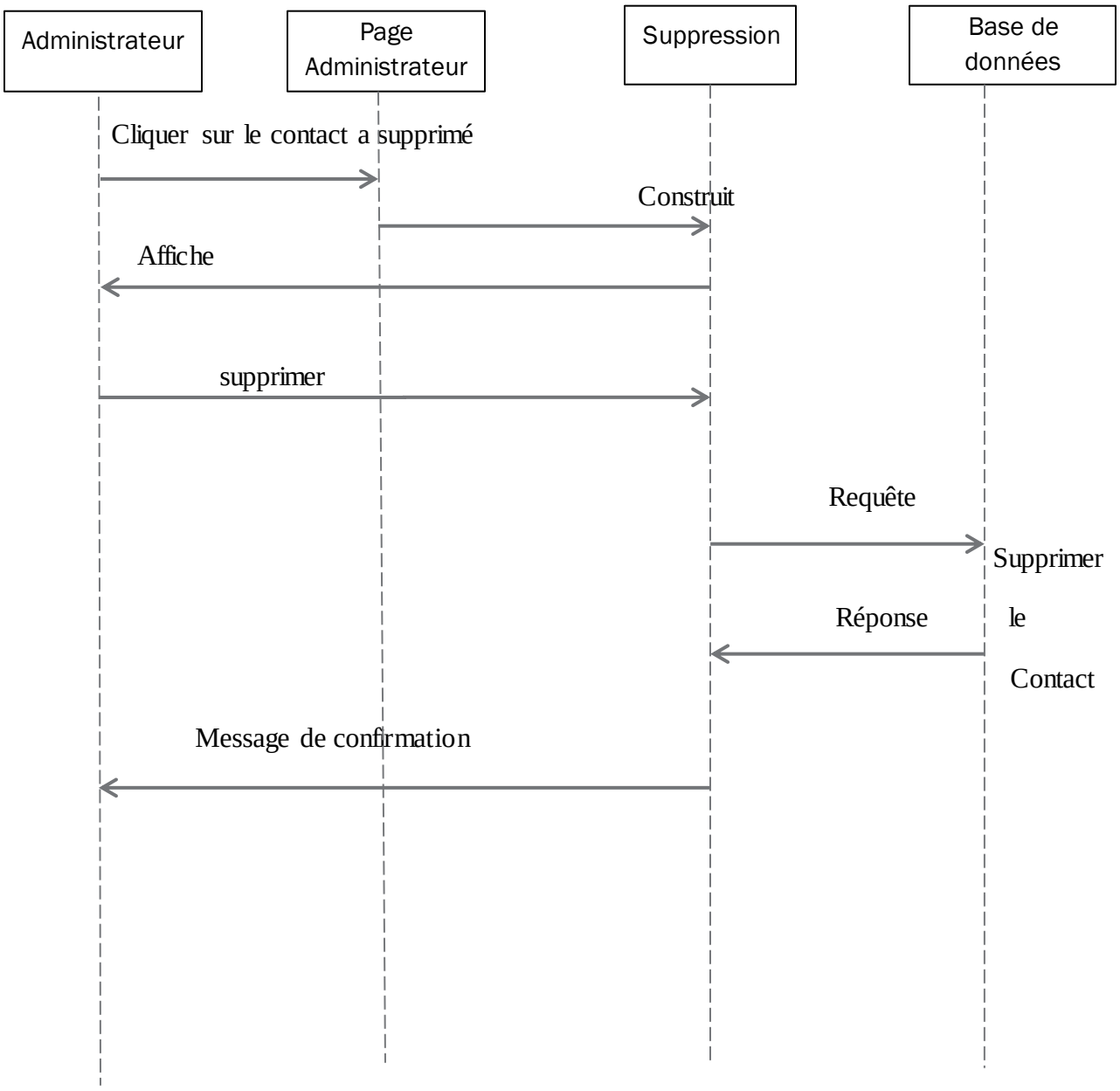


Figure 3.1.4. Diagramme de séquence pour le cas d'utilisation supprimer un contact

3.1.5. Diagramme de séquence activer/désactiver un ordinateur :

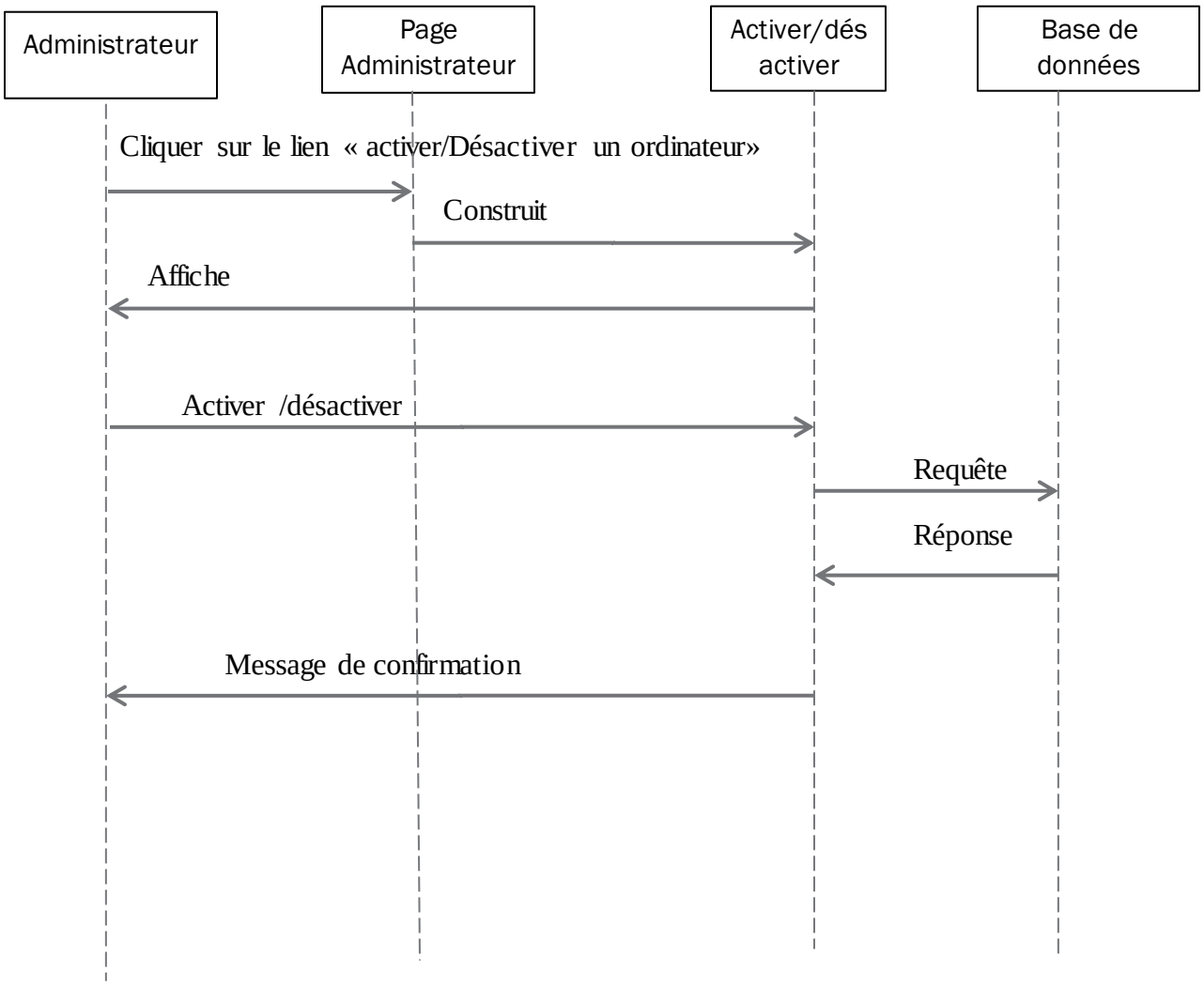


Figure3.1.5.Diagramme de séquence pour le cas d'utilisation activer/désactiver un ordinateur

3.2.Diagramme d'activé :

Un diagramme d'activité ressemble à un organigramme traditionnel.il permet d'identifié d'un coup d'œil la famille des scénarios d'un cas d'utilisation qui décrivent toute les réactions du système

3.2.1. Diagramme d'activité de l'ajout de l'utilisateur :

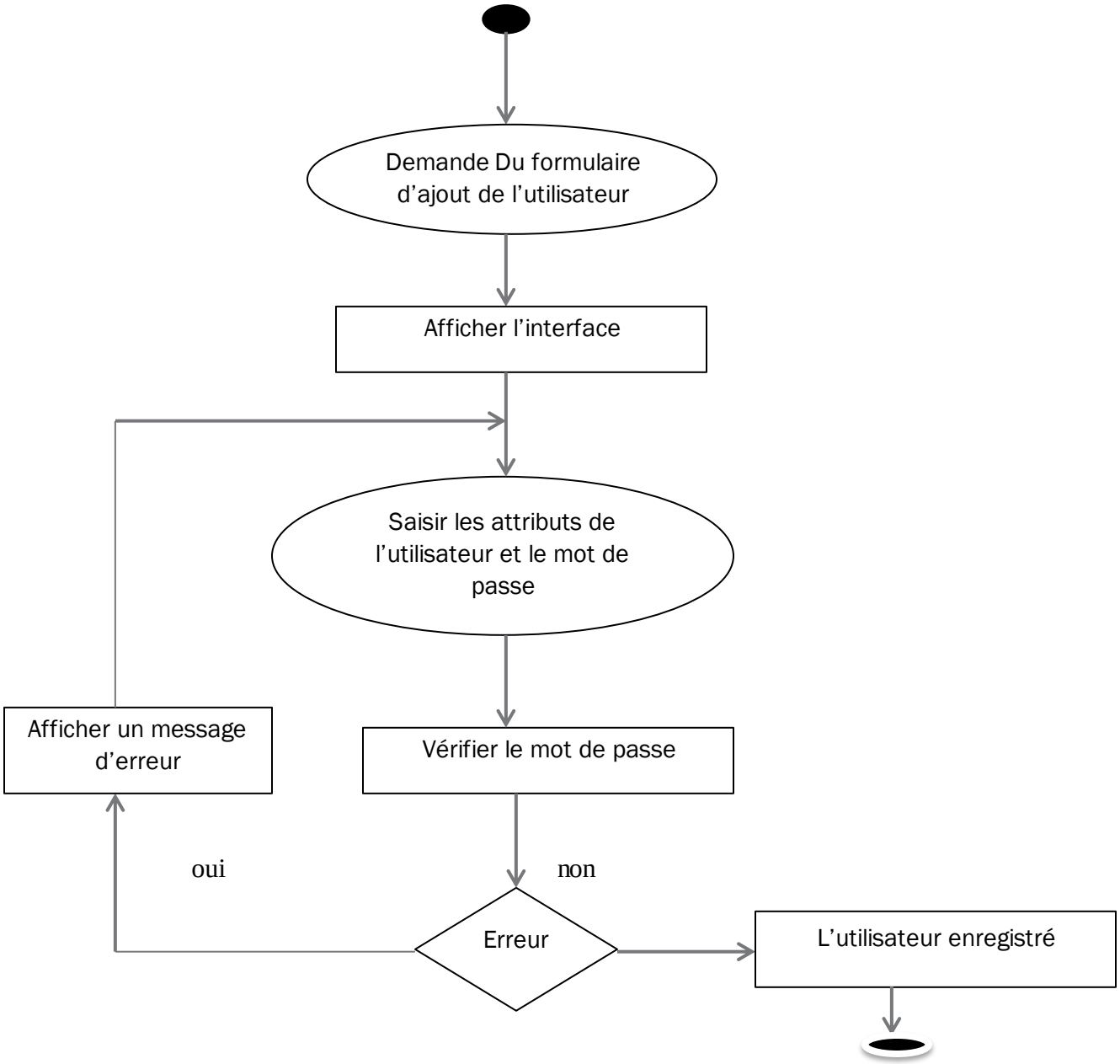


Figure 3.2.1. Diagramme d'activité de l'ajout de l'utilisateur

3.2.2. Diagramme d'activité de recherche d'utilisateur :

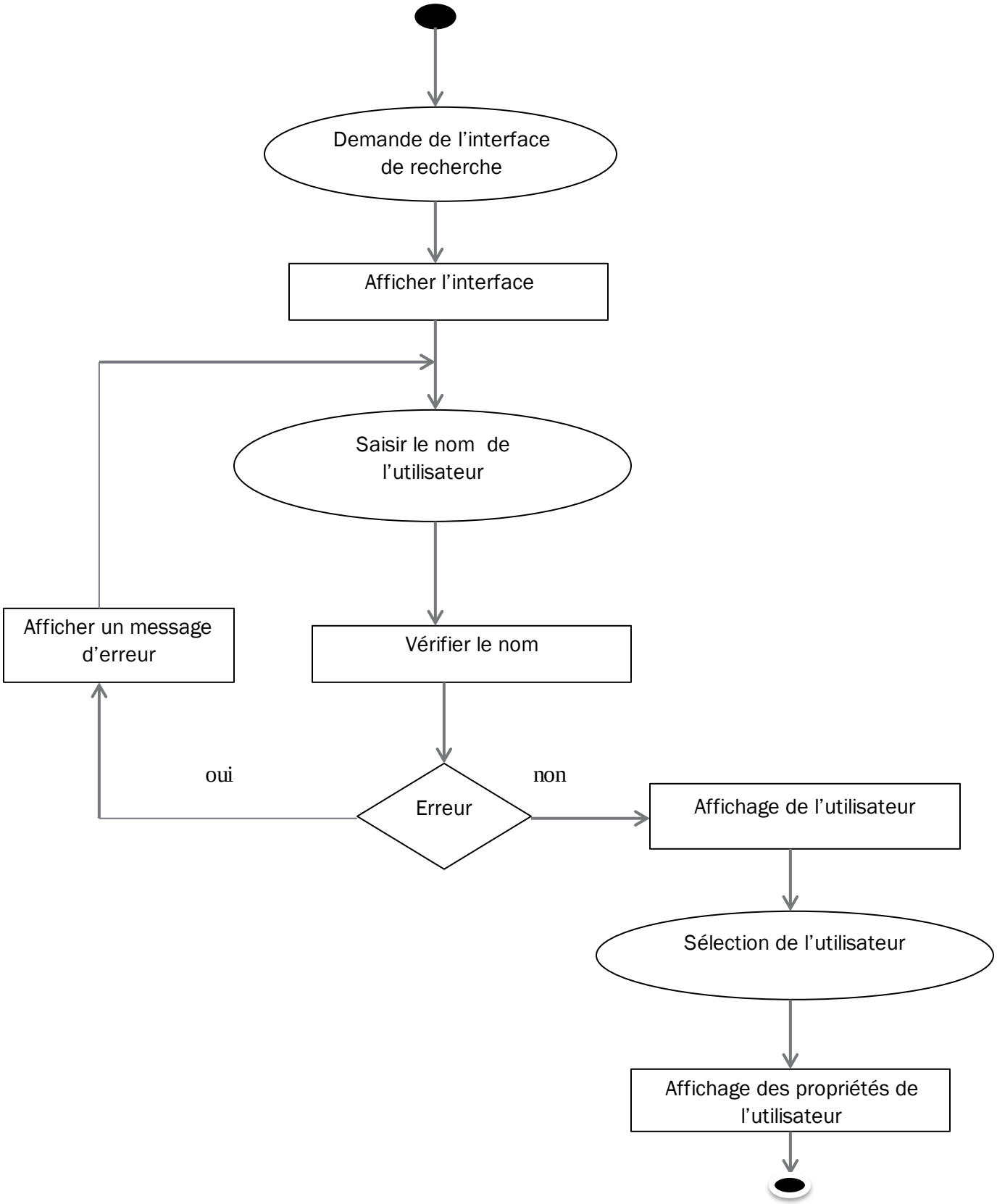


Figure 3.2.2. Diagramme d'activité de recherche d'utilisateur

3.2.3. Diagramme d'activité de modification d'un groupe :

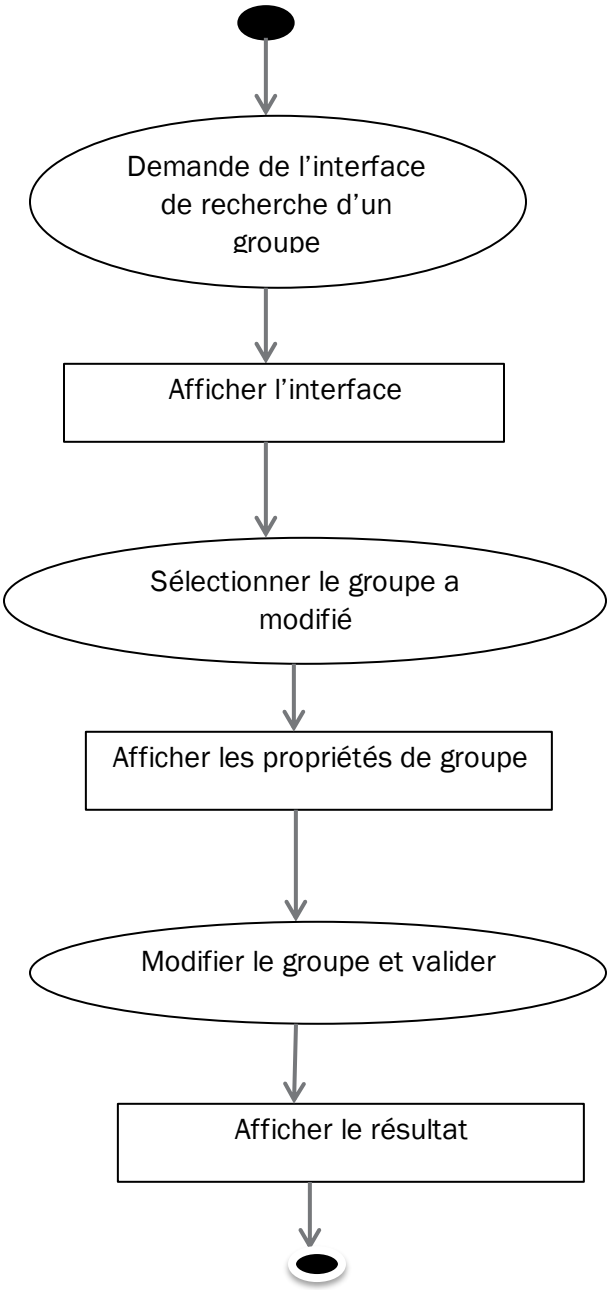


Figure 3.2.3. Diagramme d'activité de modification d'un groupe

3.2.4. Diagramme d'activité de suppression d'un contact :

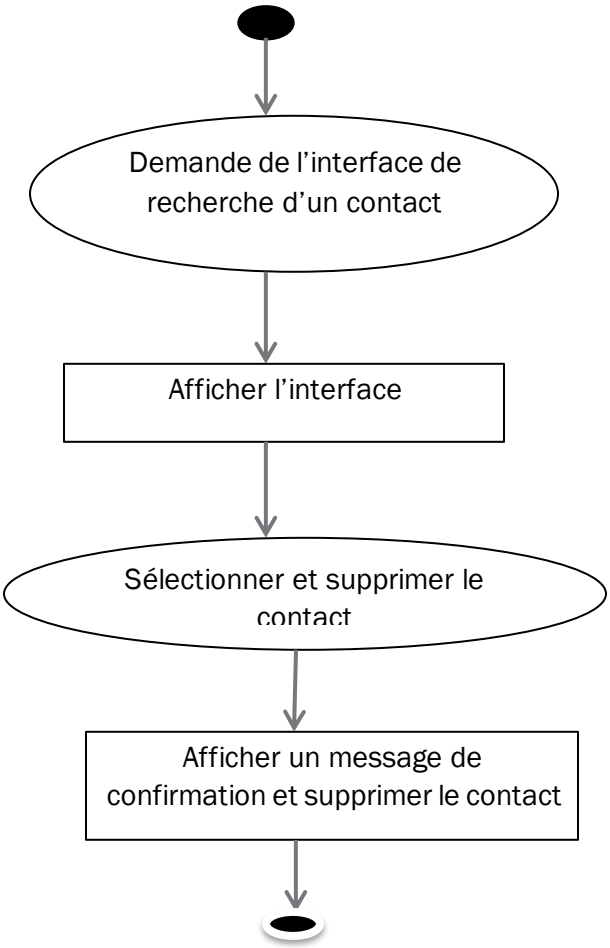


Figure 3.2.4.Diagramme d'activité de suppression d'un contact

3.2.5. Diagramme d'activité de Désactiver un ordinateur:

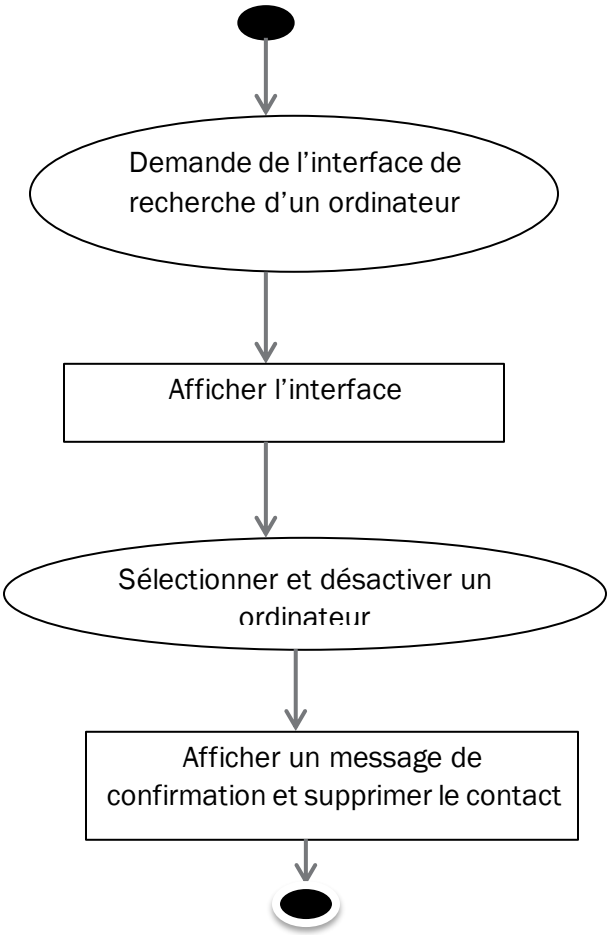


Figure 3.2.5. Diagramme d'activité de Désactiver un ordinateur

3.3. Diagramme de classe :

Le diagramme de classe représente une vue statique car on ne tient pas compte du facteur temporel dans le comportement du système. Le diagramme de classes modélise les concepts du domaine d'application ainsi que les concepts internes créés de toutes pièces dans le cadre de l'implémentation d'une application. Il contient principalement des classes ainsi que leurs associations mais on peut aussi y trouver des objets. Il permet de fournir une représentation abstraite des objets du système qui vont interagir ensemble pour réaliser les cas d'utilisation.

3.3.1. Diagramme de classe gérer les utilisateurs :

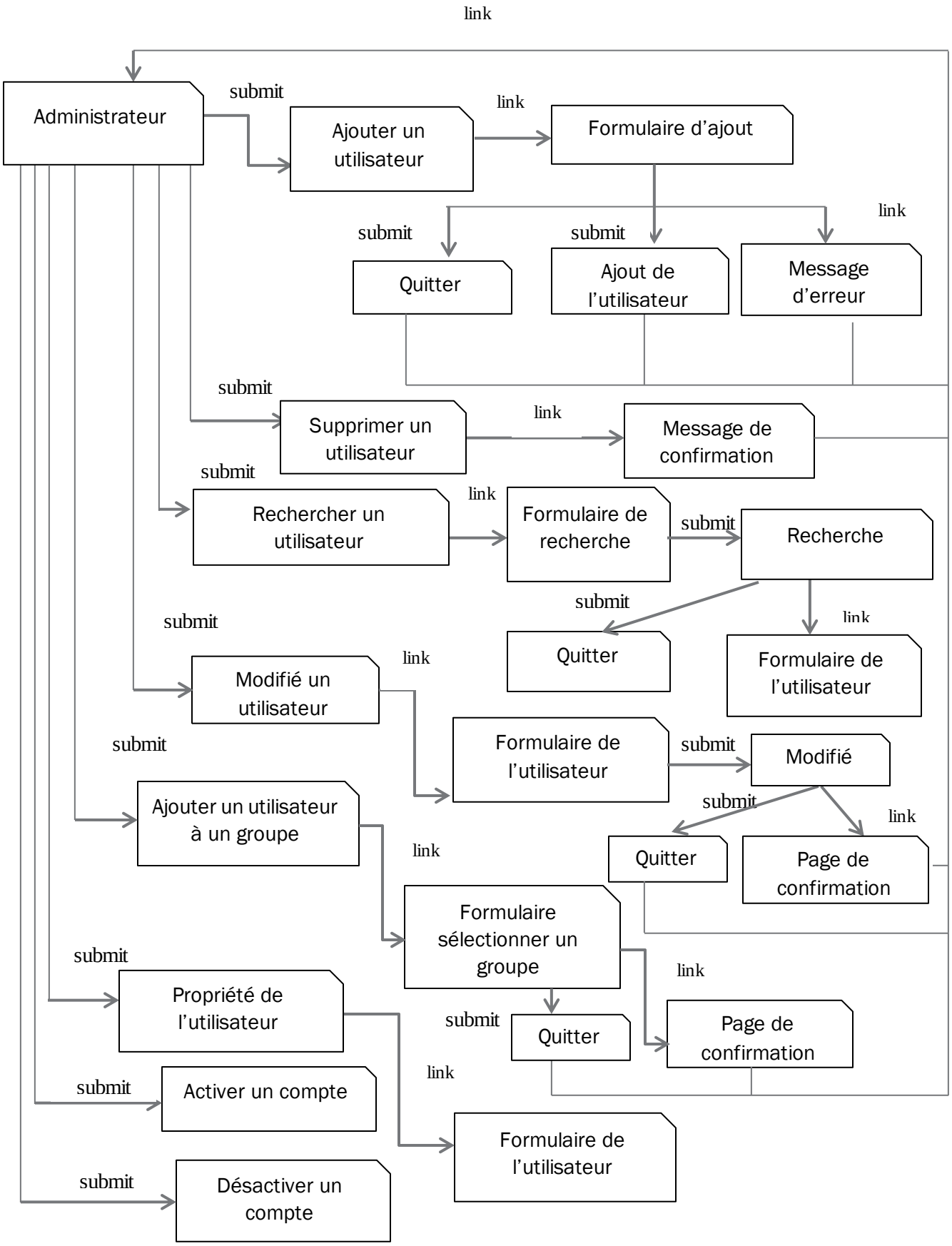


Figure 3.3.1. Diagramme de classe gérer les utilisateurs

3.1.2. Diagramme de classe gérer les groupes :

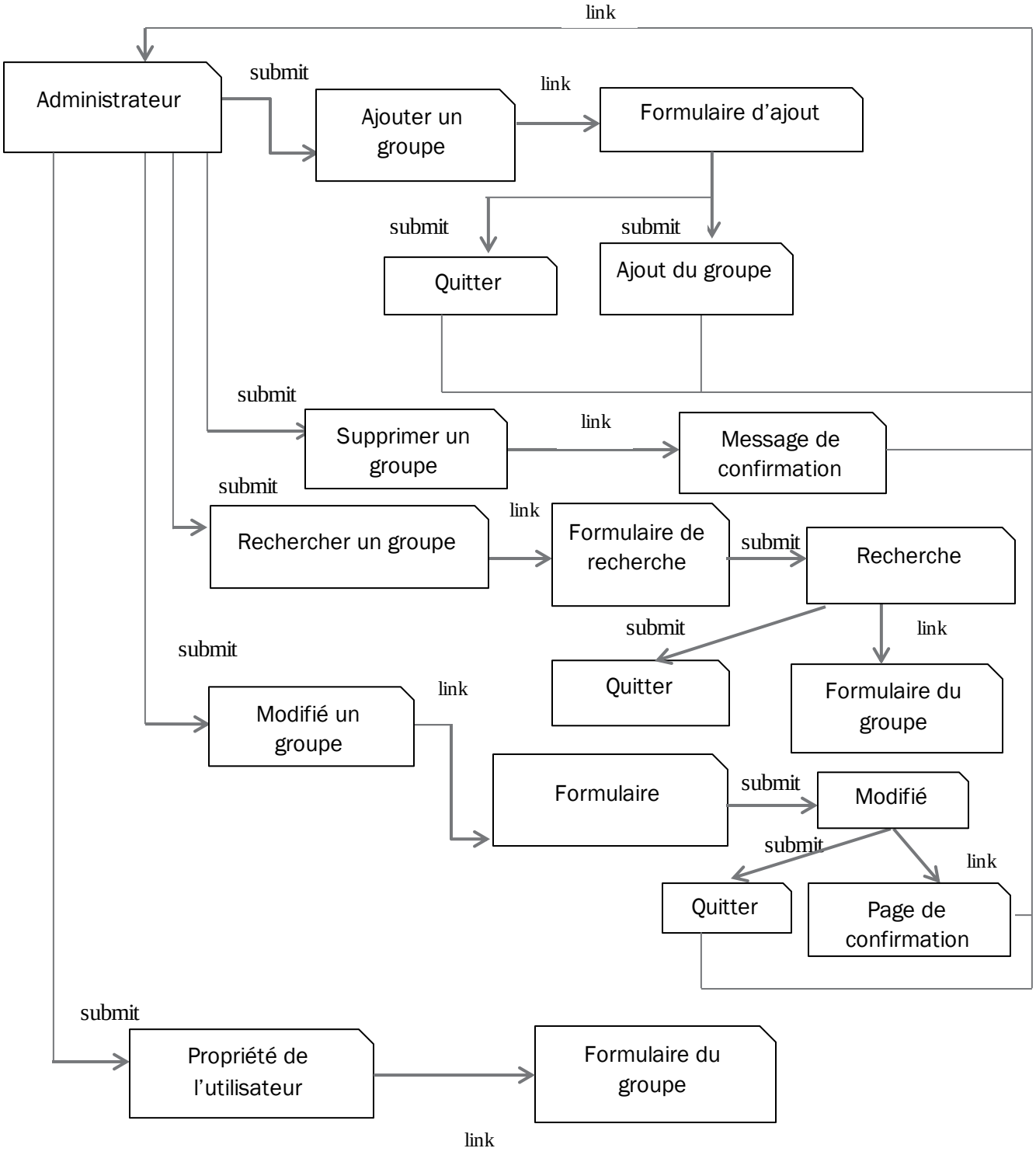


Figure 3.1.2.Diagramme de classe gérer les groupes

3.1.3. Diagramme de classe gérer les contacts:

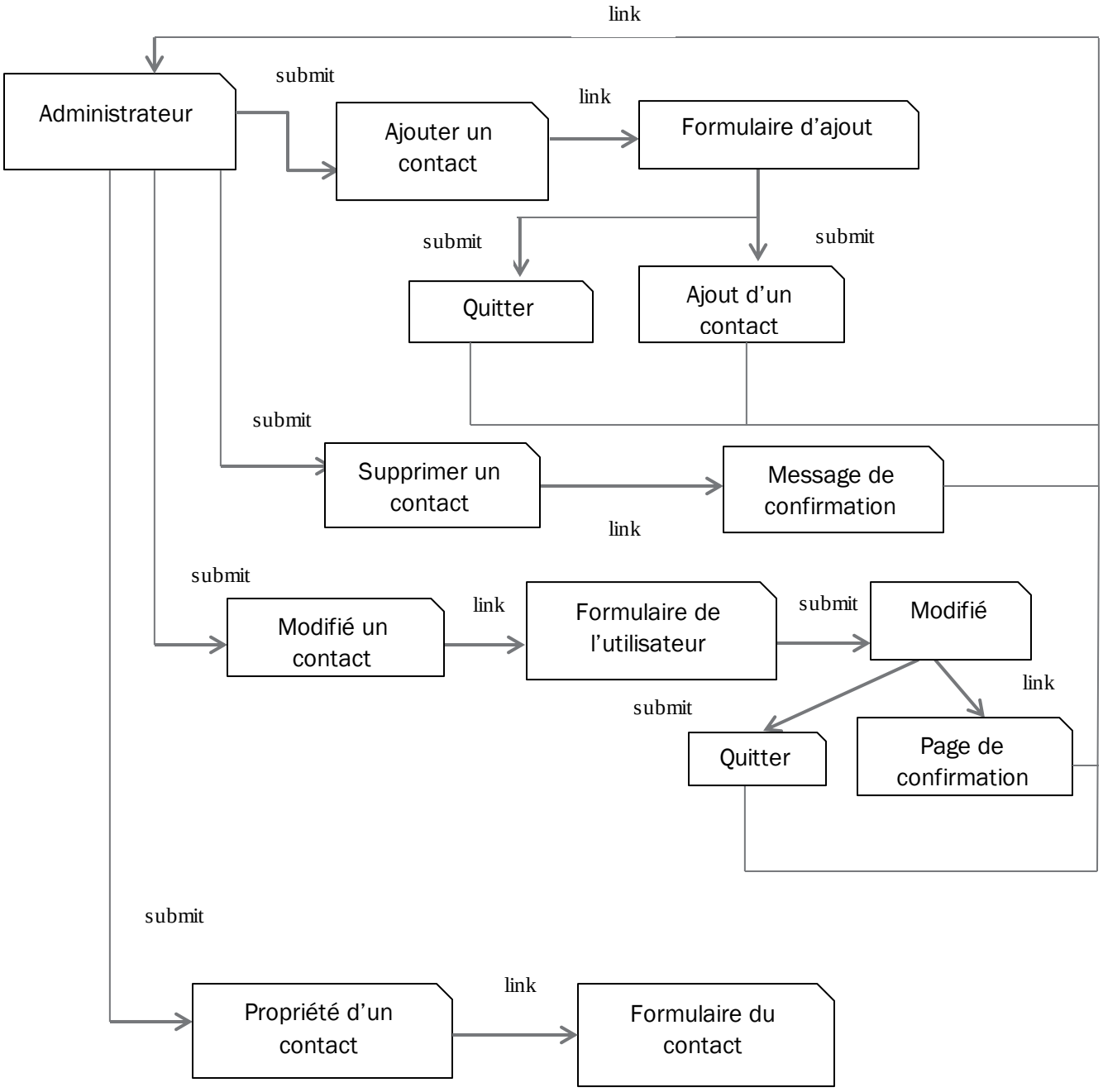


Figure 3.1.3.Diagramme de classe gérer les contacts

3.1.4. Diagramme de classe gérer les ordinateurs :

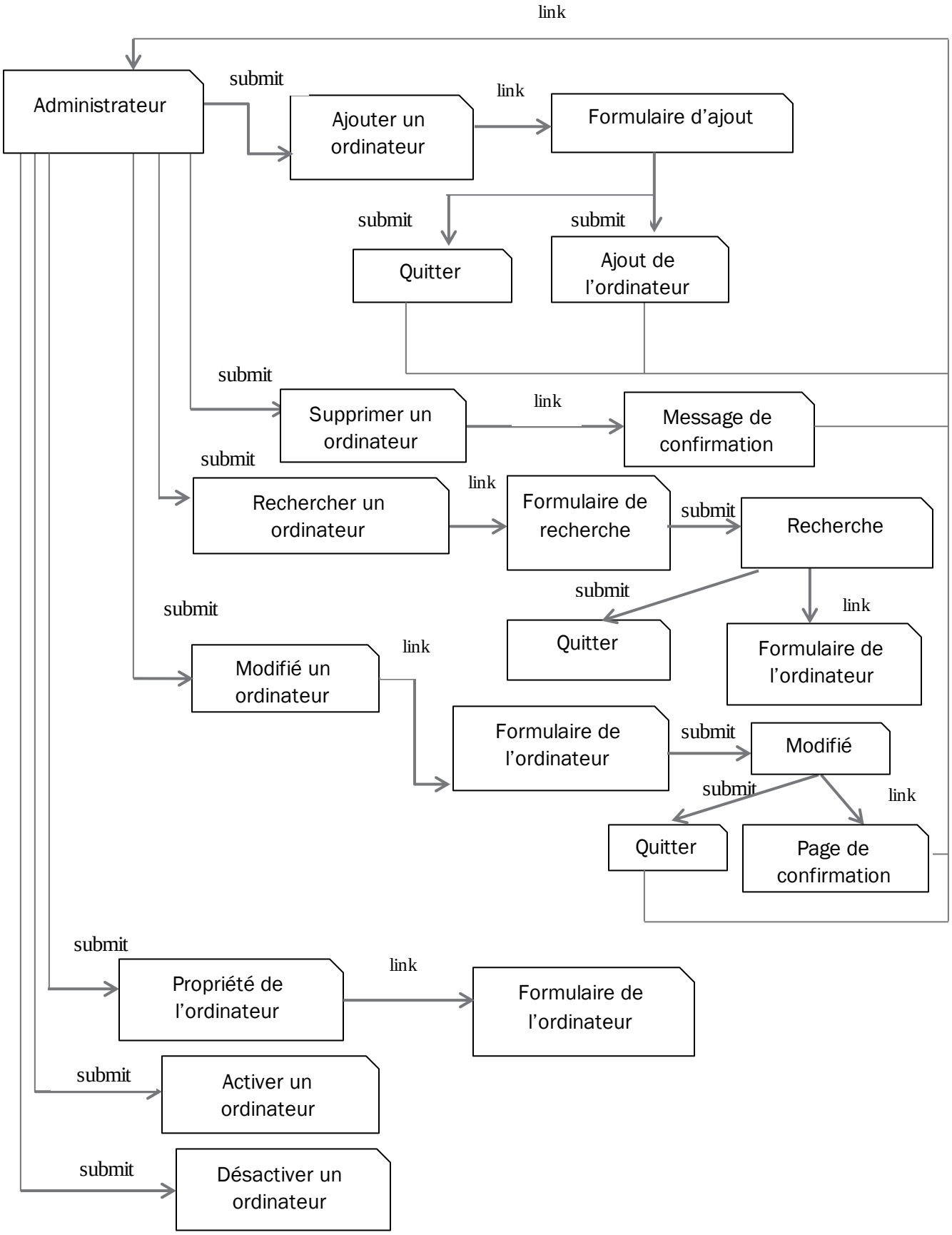


Figure 3.1.4. Diagramme de classe gérer les ordinateurs

Conclusion :

Tout au long de ce chapitre, nous avons présenté les différents besoins auxquels le système doit répondre. Nous avons également décrit les différents cas d'utilisation système et nous les avons formellement illustrées par des diagrammes de cas d'utilisation, diagramme de séquence, diagramme d'activités et diagramme de classe

Dans le prochain chapitre, nous allons montrer à l'aide de quels outils nous avons pu mettre en place notre application.

CHAPITRE IV:
LA REALISATION

Introduction :

Dans ce chapitre, nous commençons par présenter l'environnement logiciel et de programmation, ainsi que les choix techniques de développement. Ensuite, nous terminerons par une présentation des principales interfaces de notre application

1. Environnement de programmation :

Le langage Visual basic :

est un langage de programmation événementielle de troisième génération ainsi qu'un environnement de développement intégré, créé par Microsoft pour son modèle de programmation COM. Visual Basic est directement dérivé du BASIC et permet le développement rapide d'applications, la création d'interfaces utilisateur graphiques, l'accès aux bases de données en utilisant les technologies DAO, ADO et RDO, ainsi que la création de contrôles ou objets ActiveX. Les langages de script tels que *Visual Basic for Applications* et VBScript sont syntaxiquement proches de Visual Basic, mais s'utilisent et se comportent de façon sensiblement différente.

Un programme en VB peut être développé en utilisant les composants fournis avec Visual Basic lui-même. Les programmes écrits en Visual Basic peuvent aussi utiliser l'API Windows, ceci nécessitant la déclaration dans le programme des fonctions externes. De ce fait on a choisie Visual studio comme environnement de développement et la figure suivante présente son interface principale

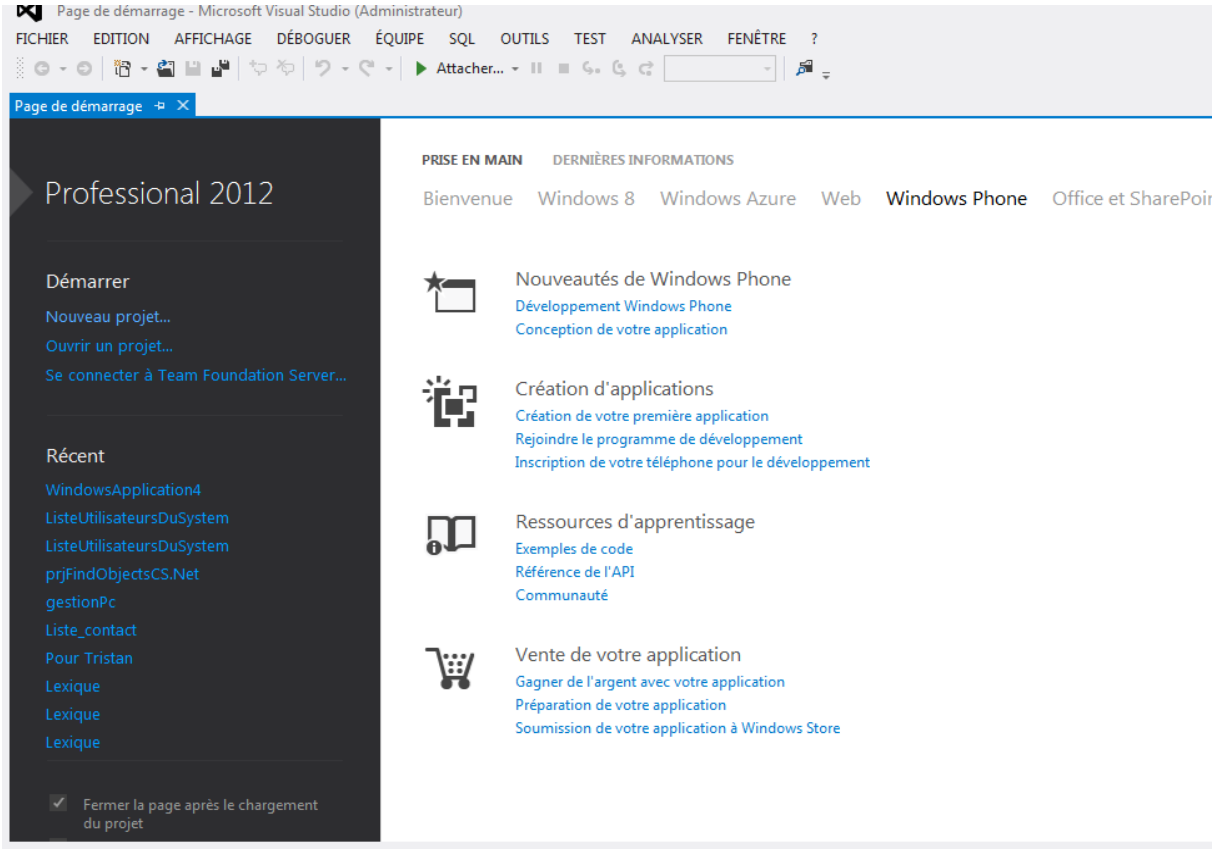


Figure 1 : interface principale de Microsoft Visual Studio

2. La plate-forme d'exécution :

Celle-ci repose sur un système d'exploitation (Windows Server 2003) faisant office de serveur d'administration avec le service d'annuaire Active Directory permettant de gérer et d'administrer tous les objets d'un réseau.

2.1.Système d'exploitation :

Microsoft Windows Server 2008 est le système d'exploitation Windows Server de nouvelle génération qui aide les administrateurs système à optimiser leur contrôle sur l'infrastructure. Il offre une disponibilité et des fonctionnalités sans précédent.
Les administrateurs bénéficient d'un environnement serveur davantage sécurisé, fiable et robuste. Par ailleurs, Windows Server 2008 propose aux organisations une nouvelle valeur ajoutée en garantissant à tous les utilisateurs l'accès à l'ensemble des services du réseau, où qu'ils se trouvent. Windows Server 2008 offre également une vue approfondie sur les fonctions du système d'exploitation et de diagnostic, permettant aux administrateurs de consacrer davantage de temps à la valeur métier de l'entreprise.

2.2.Service d'annuaire Active Directory

Active Directory permet une gestion de tous les types d'objets réseau au sein d'une structure hiérarchique et évolutive tout en apportant une tolérance de pannes et un équilibrage de charge entre les contrôleurs de domaine. L'administration est simplifiée et l'évolutivité du système est assurée. Il limite le temps écoulé pour la configuration de chaque machine cliente dans le déploiement et le paramétrage de l'environnement de travail, grâce aux stratégies de groupes. Il est compatible avec plusieurs protocoles standards (DNS, LDAP, etc.).

3. Installation et configuration d'Active directory :

Pour commencer l'installation il va falloir ajouter le rôle. ***Gestionnaire de Serveur -> Ajouter Rôle***

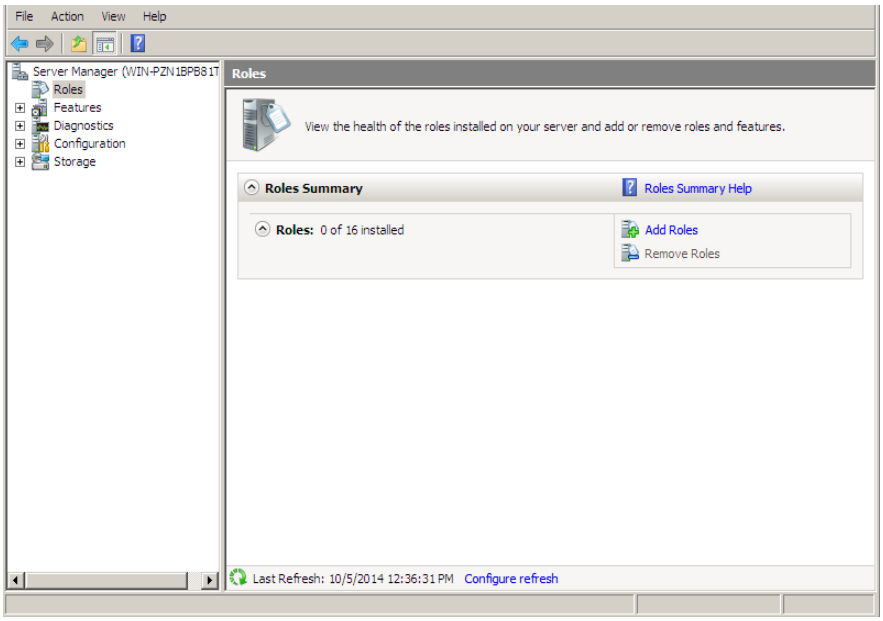


Figure 3.1. Gestionnaire de serveur

Ensuite on choisit Service de Rôles Active Directory. Lancer l'installation et ajoutez les fonctionnalités (proposées par l'assistant) qu'il vous manque.

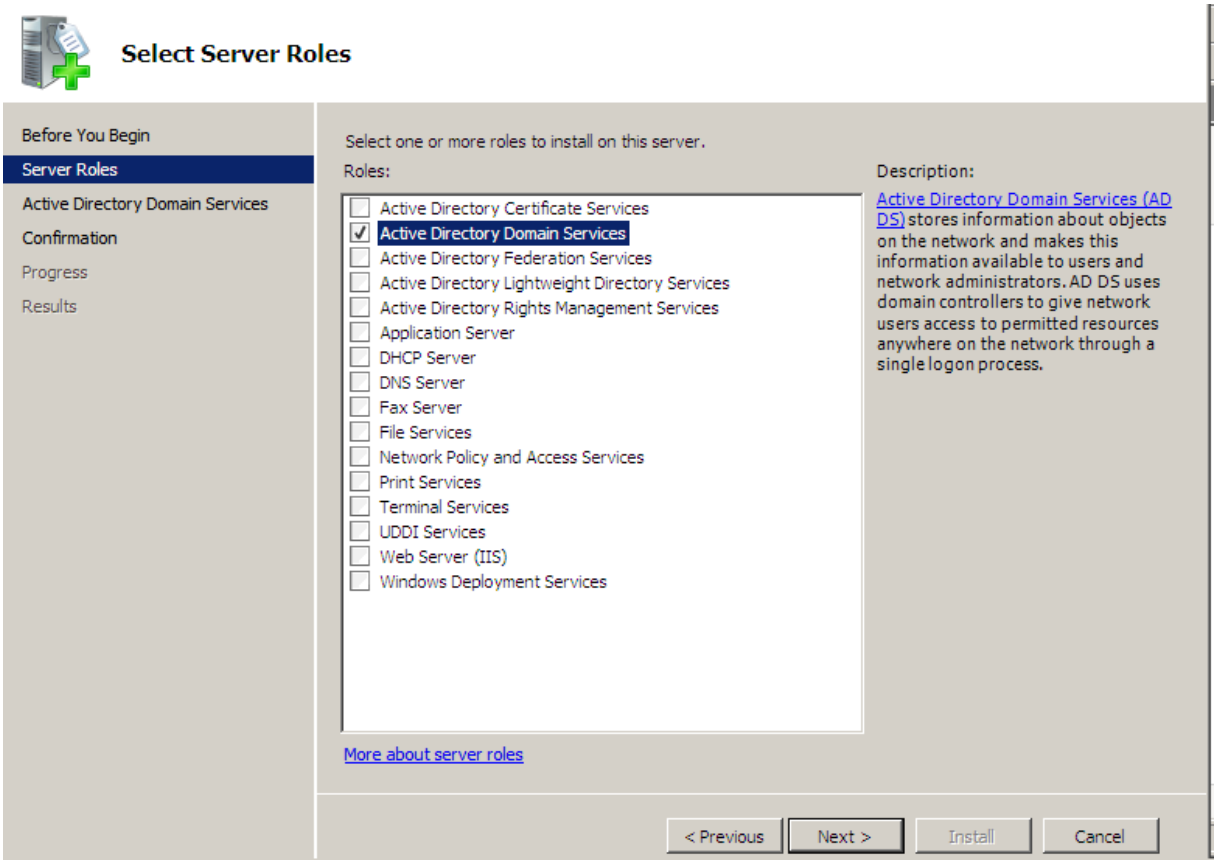


Figure 3.2.Service de rôle serveur

Une fois terminé, nous avons automatiquement le *« dcpromo.exe »* qui se lance. Ceci est l'assistant de configuration pour notre Active Directory. La première chose que vous propose l'assistant c'est d'installer en mode avancé. Le mode avancé vous permet de configurer votre Active Directory via une sauvegarde.

Vu que notre but est d'en installer un, comme étant le tout premier contrôleur de domaine, nous ne sélectionnerons pas cette option.

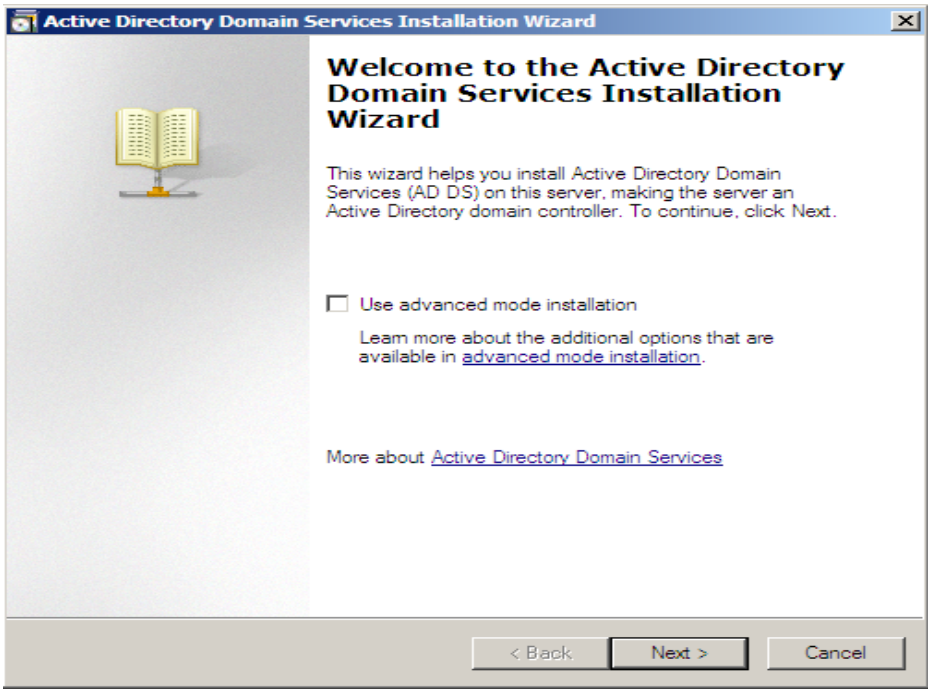


Figure 3.3. Lancement de l'assistant d'installation d'Active Directory

Avant de commencer les étapes de configuration, Windows nous informe que sous Windows Serveur 2008 et 2008 R2 les algorithmes de chiffrement sont très forts, du coup il pourrait avoir des soucis avec des machines trop anciennes ou des OS trop vieux.

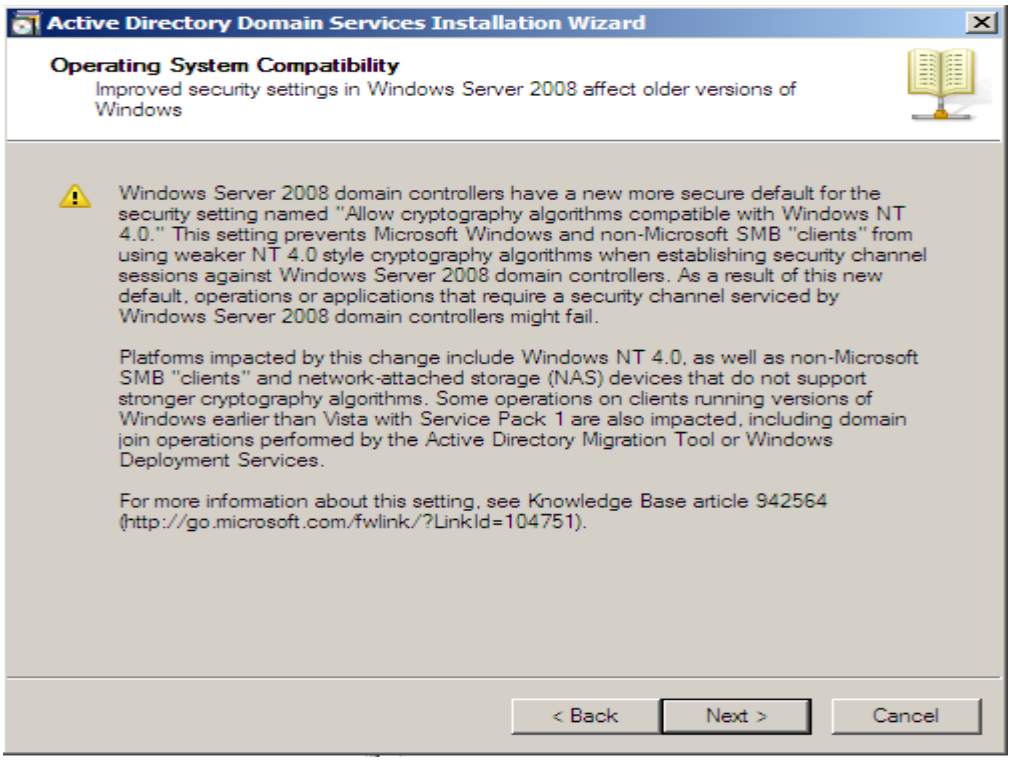


Figure 3.4.information sur le Windows server

Maintenant nous allons commencer à configurer notre Active Directory. La première étape est de soit rejoindre une forêt existante ou en créer une dans notre cas il nous faut créer une forêt

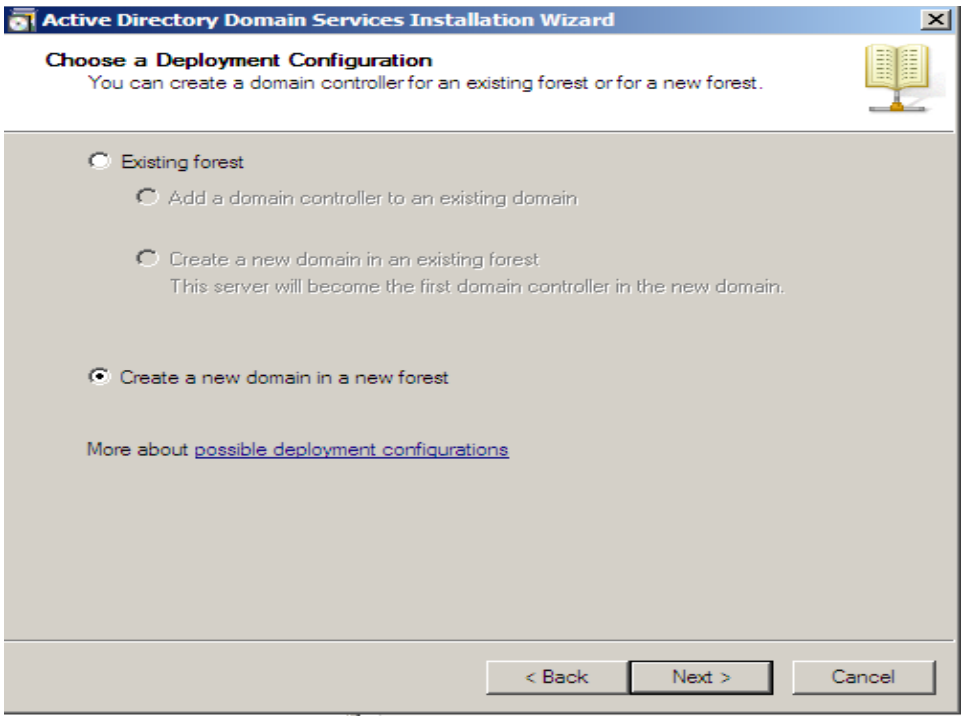


Figure 3.5.Selection du type de domaine

Ensuite on va lui donner un nom qu'est : dep.info.to

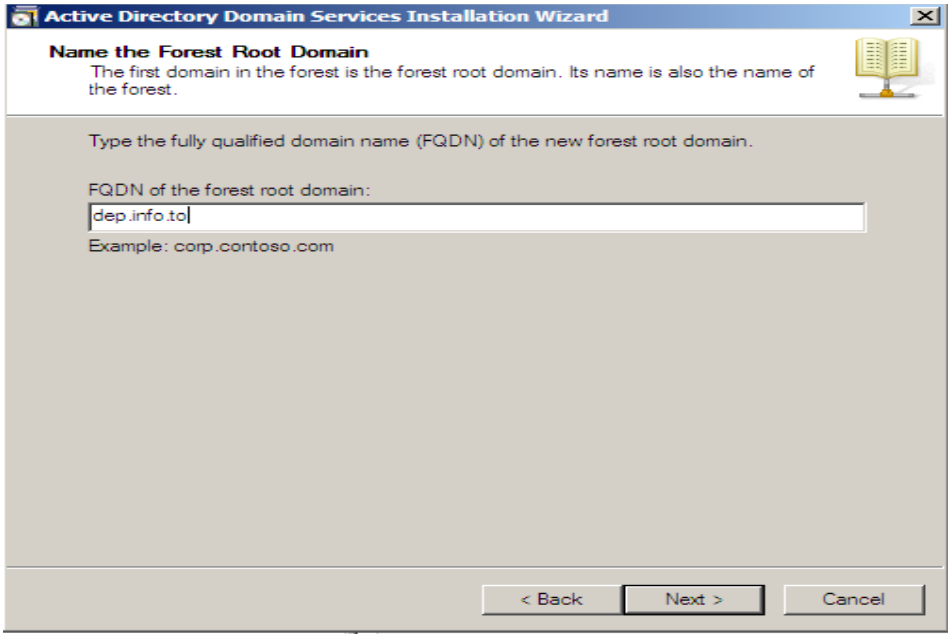


Figure 3.6. Spécification du nom pour le nouveau domaine

Une fois notre nom affecté, Windows nous demande de choisir le niveau fonctionnel de notre foret Active Directory. Dans notre cas on choisir le niveau fonctionnel 2008

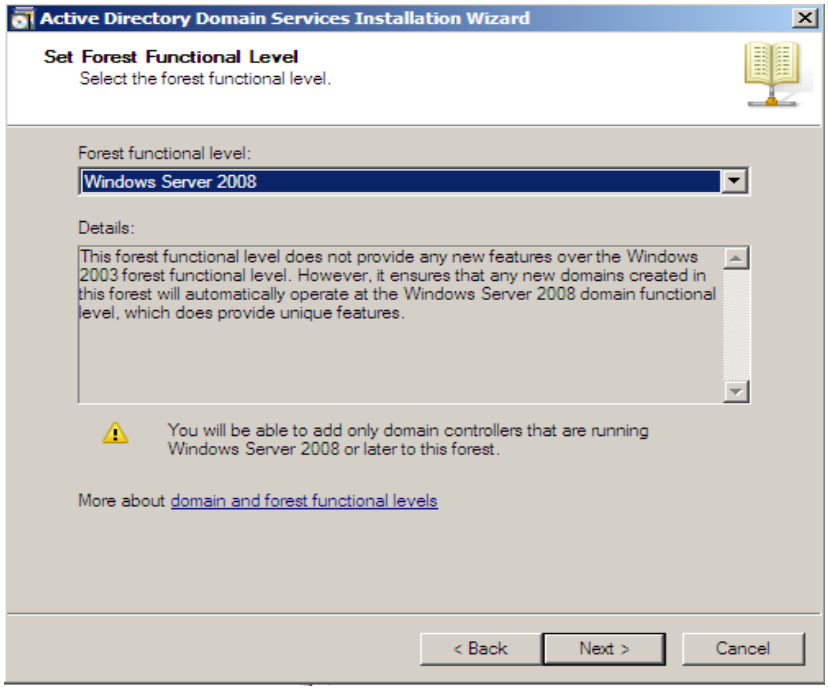


Figure 3.7.choisir le niveau fonctionnel

Ensuite Windows nous propose d'installer des options supplémentaires. Dans notre Cas on n'a pas de Serveur DNS parent qui gère le « .adds ». Il va essayer de le contacter mais vu qu'il n'existe pas il va nous générer un message d'erreur ce qui est tout normal.

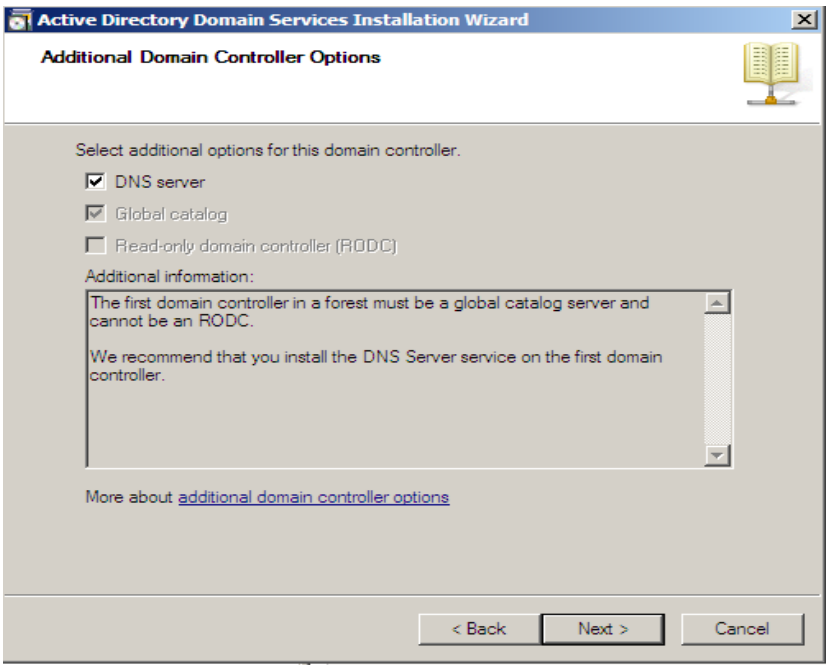
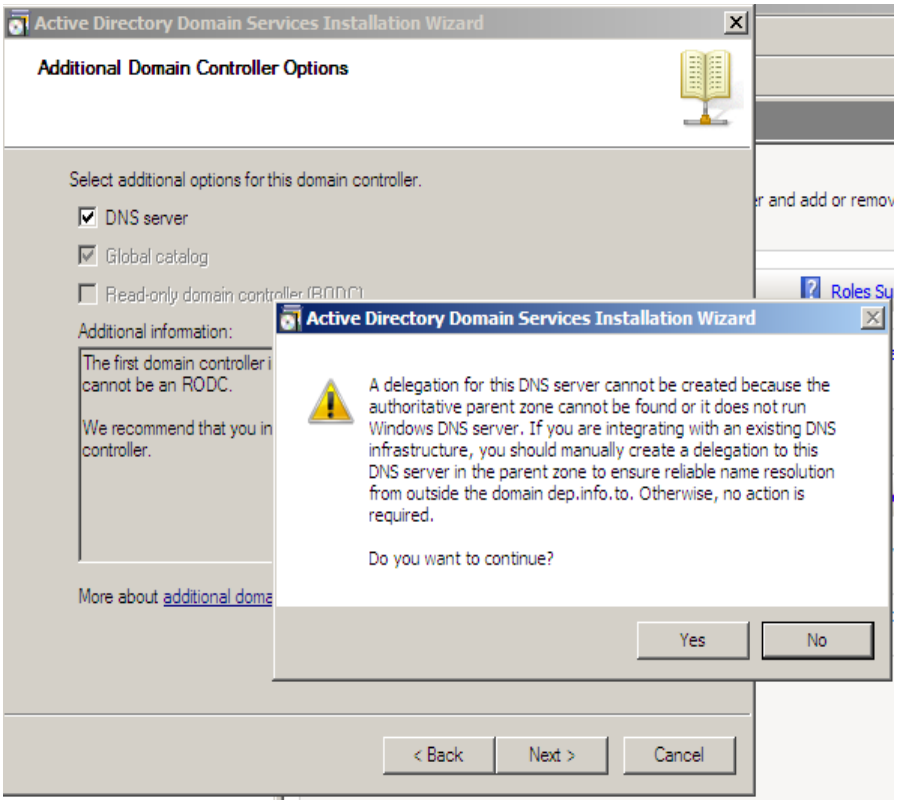


Figure 3.8.installation du DNS



Ensuite, on nous demande de choisir l'emplacement des fichiers de bases de données, de logs. Des fichiers qui vont être utiles à l'Active Directory.

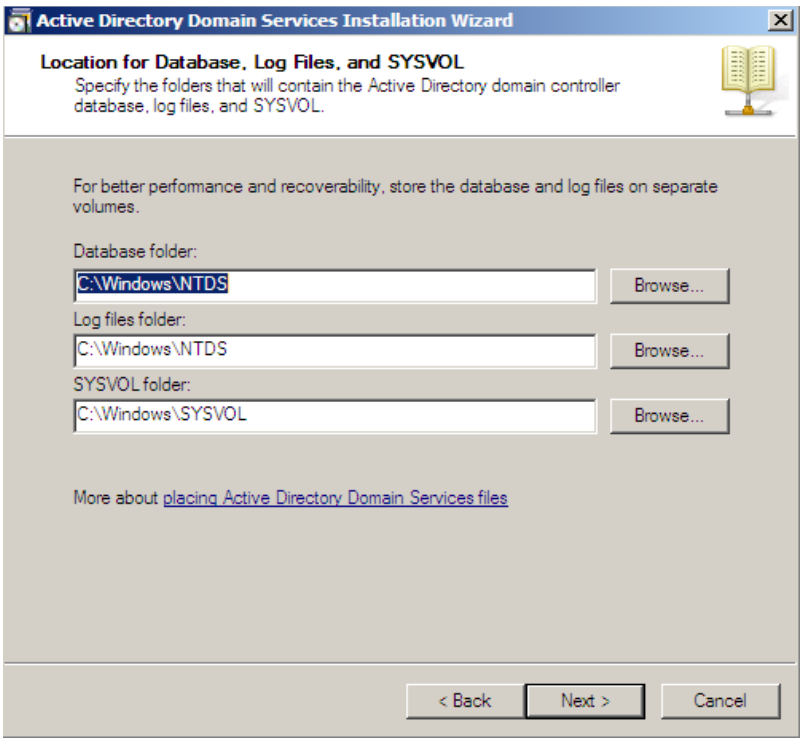


Figure 3.9. Emplacement du dossier Sysvol

Dans le but de protéger les restaurations de votre Active Directory, Windows nous propose d'y affecter un mot de passe juste pour la restauration

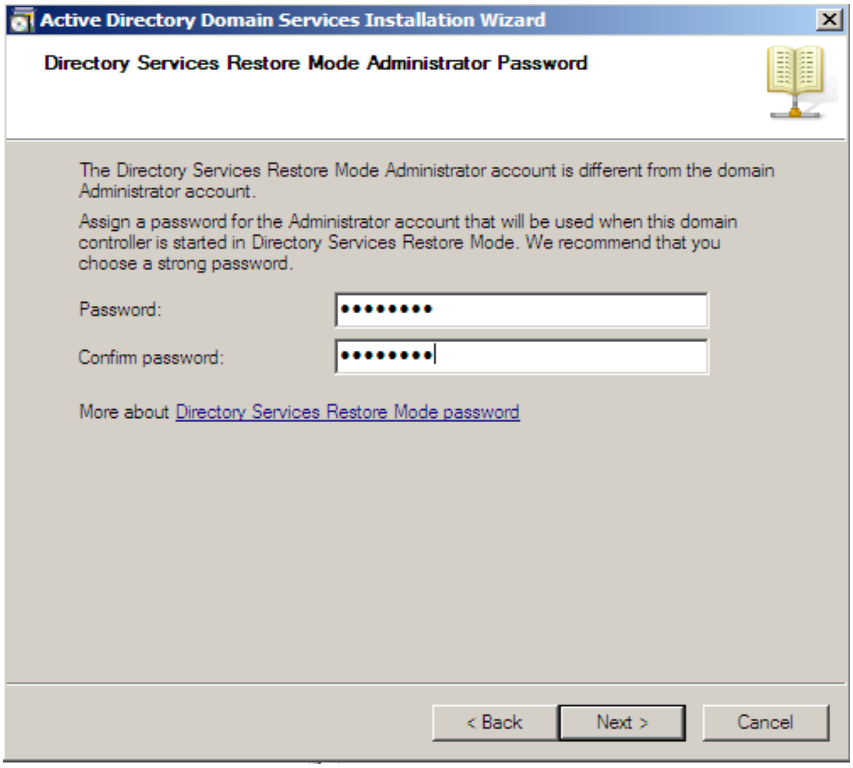


Figure 3.10.Mot de passe pour la restauration

Nous aurons droit à un petit résumé et une fois accepté les services seront installés et configurés. Une fois cliqué sur « fin » le système devra redémarrer.

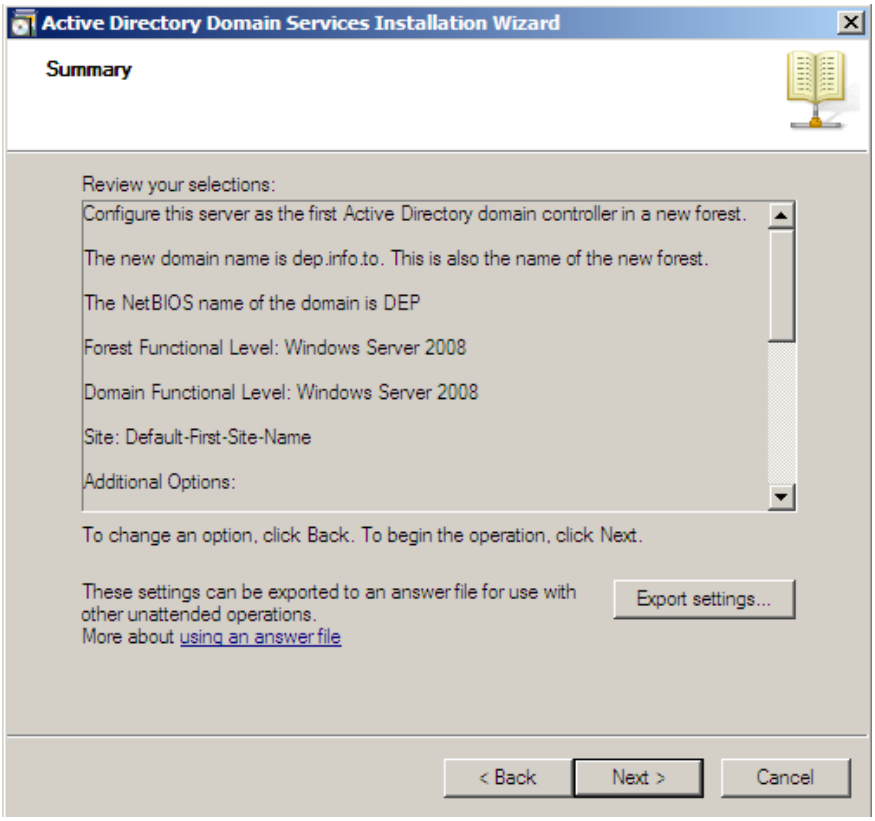


Figure 3.11.Resumer des services

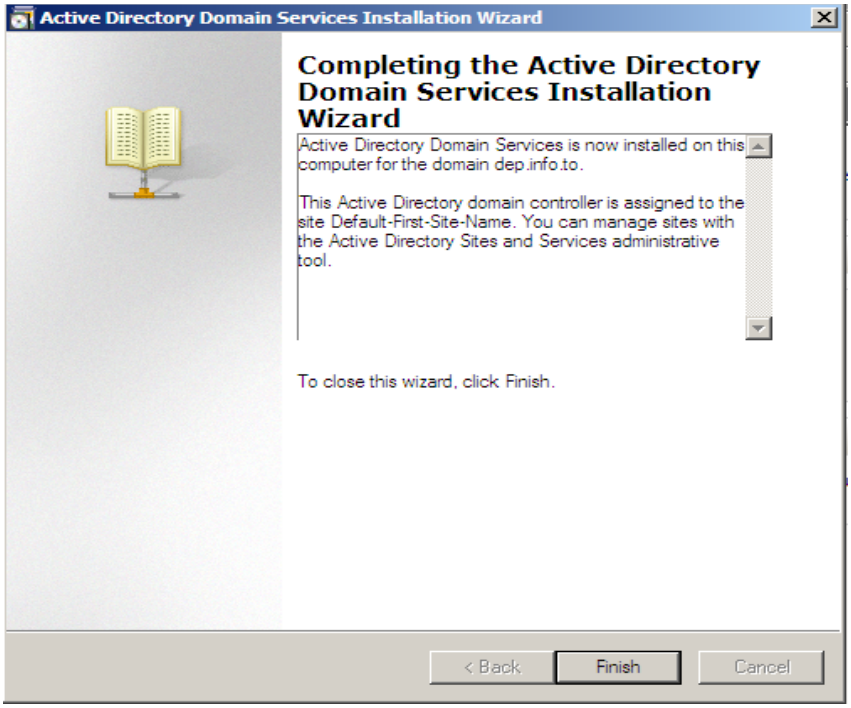


Figure 3.12. Fin de l'installation Active Directory

4. Installation de .NetFramework4.0 :

- Après l'extraction des fichiers l'installation peut commencer

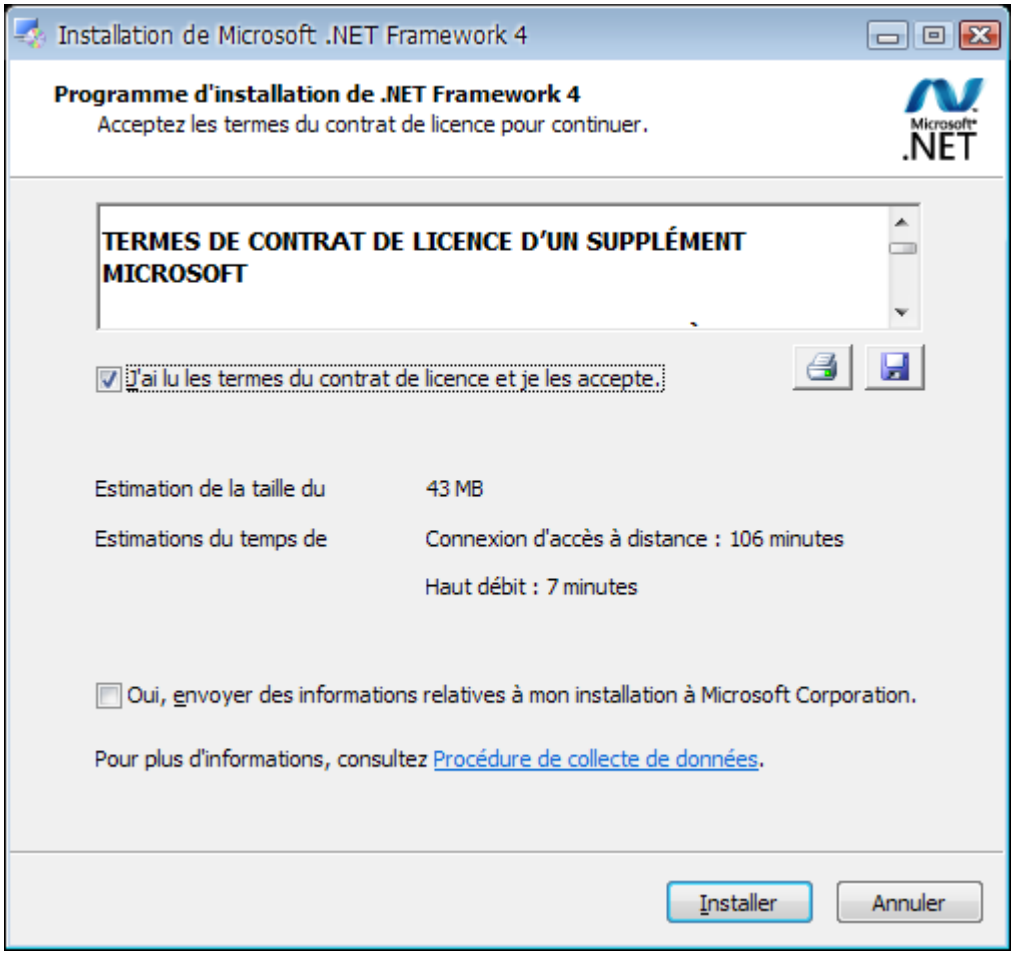


Figure 4.1. Assistant de l'installation de Microsoft .Net Framework

- Cliquer sur **[Installer]**

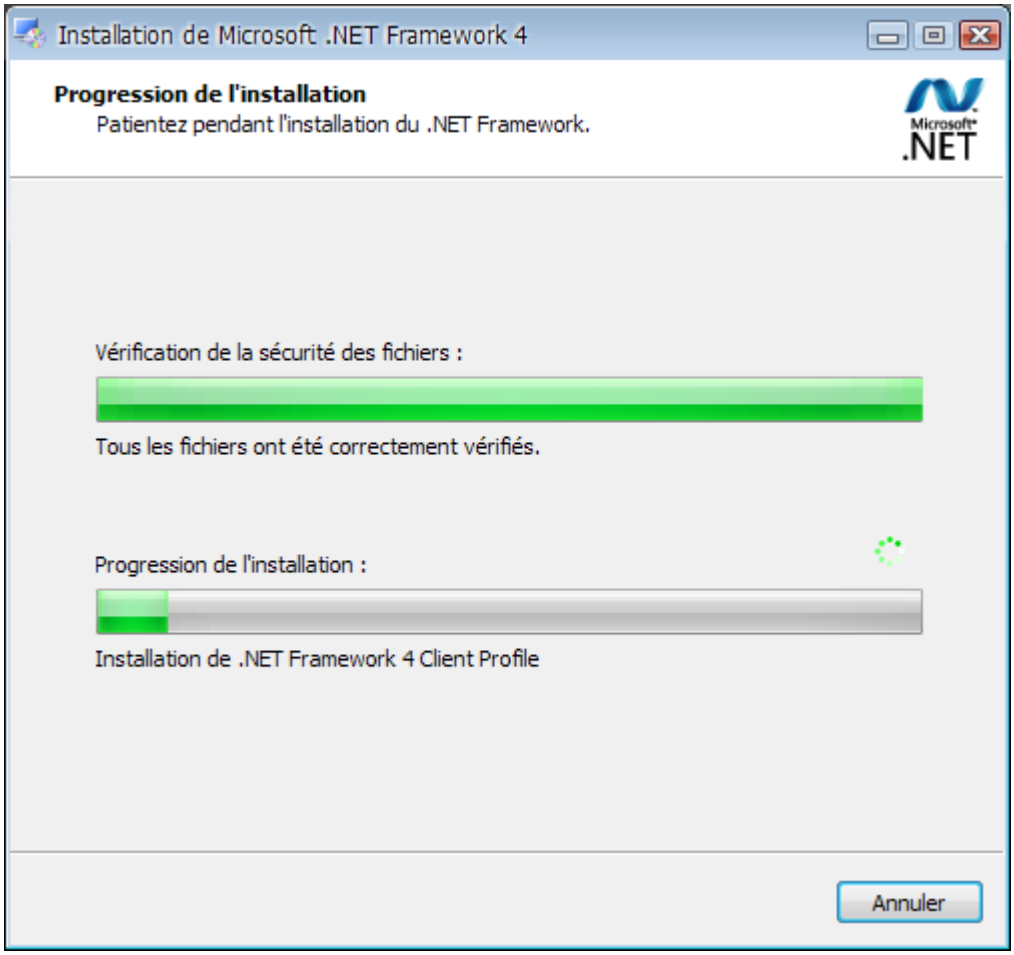


Figure 4.2.Installation de .Net Framework

- L'installation peut durer quelques minutes en fonction de l'ordinateur

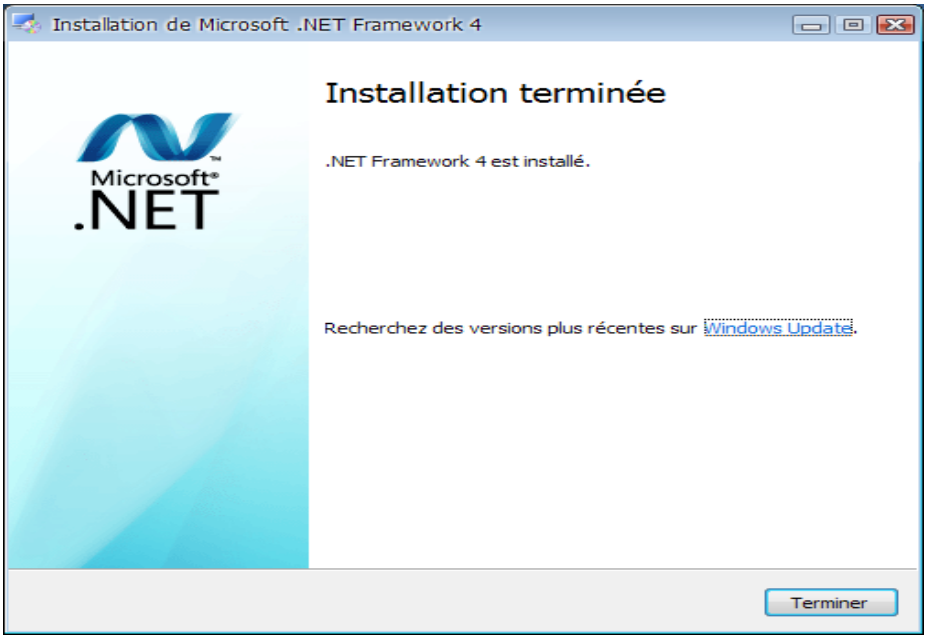


Figure 4.3. Fin de l'installation de .Net Framework

- A la fin de l'installation, il est demandé de redémarrer l'ordinateur

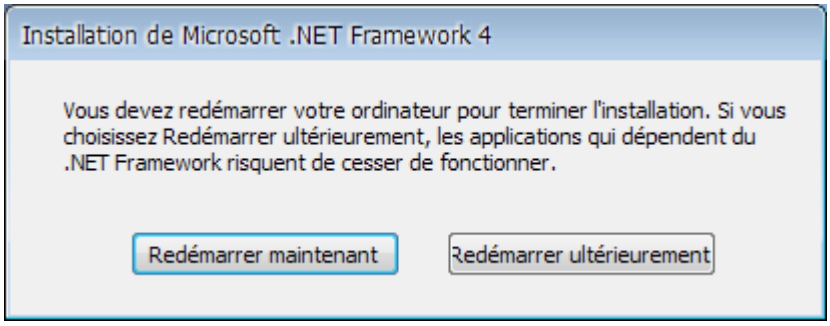


Figure 4.4.Redémarrage pour finir l'installation

5. Présentation de quelque code source :

5.1. Connexion à l'annuaire active directory :

Ce code permet de se connecter à active directory

```
Dim de As New DirectoryEntry("LDAP://CN=Users,DC=dep,DC=info,DC=to")
```

5.2. Ajout d'utilisateur :

Ce code permet d'ajouter un utilisateur à active directory

```
Dim de As New DirectoryEntry("LDAP://CN=Users,DC=dep,DC=info,DC=to")
Dim maRecherche As DirectorySearcher = New DirectorySearcher
'on cherche si l'utilisateur a creer existe dans ad
's'il n'existe pas ilsera creer sinon un message sera renvoyer
maRecherche.SearchRoot = de
maRecherche.Filter = "(&(objectClass=user) (ANR=" & TxtLogin.Text & "))"
Dim results As SearchResultCollection = maRecherche.FindAll()
If results.Count = 0 Then
    If (TxtPassword.Text.Length > 7) And (TxtPassword.Text = TxtOk.Text) Then
        'on va creer un nouvelle utilisateur
        Dim newUser As DirectoryEntry = de.Children.Add("CN=" & TxtLogin.Text, "user")
        newUser.Properties("sAMAccountName").Value = TxtLogin.Text
        newUser.Properties("sn").Value = TxtNom.Text
        newUser.Properties("givenName").Value = TxtPrenom.Text
        'Envoyer les modification au serveur
        newUser.CommitChanges()
        'on va activer le compte
        Dim val As Integer = CType(newUser.Properties("userAccountControl").Value, Integer)
        newUser.Properties("userAccountControl").Value = val And Not 2
        newUser.CommitChanges()
        If TxtDescription.Text = "" Then
            newUser.Properties("description").Clear()
            newUser.CommitChanges()
        Else
            newUser.Properties("description").Value = TxtDescription.Text
            newUser.CommitChanges()
        End If
        MessageBox.Show("Objet ajouter")
    Else
        MsgBox("ajout echoué")
    End If
Else
    MessageBox.Show("l'objet existe déja")
End If
```

5.3.Recherche d'objet:

Permet de faire la recherche dans active directory

```
Using Searcher As System.DirectoryServices.DirectorySearcher = New System.DirectoryServices.DirectorySearcher(Root)
    If Not (tbUserID.Text.Length = 0) Then
        Searcher.Filter = "(&(|(objectClass=user)(objectClass=group))(ANR=" & tbUserID.Text & " * ))" 'chercher les
    End If
    Searcher.SearchScope = SearchScope.OneLevel
    'determiner les proprieté a recuprer
    Searcher.PropertiesToLoad.Add("sAMAccountName")
    Searcher.PropertiesToLoad.Add("description")
    Searcher.PropertiesToLoad.Add("givenName")
    Searcher.PropertiesToLoad.Add("sn")
    Searcher.PropertiesToLoad.Add("distinguishedName")
    Searcher.PropertiesToLoad.Add("groupType")
    Searcher.Sort.PropertyName = "sAMAccountName"
    Searcher.Sort.Direction = System.DirectoryServices.SortDirection.Ascending 'A-Z
    Using users = Searcher.FindAll
        If users.Count > 0 Then
            Dim Item1 As String = Nothing
            Dim Item2 As String = Nothing
            Dim Item3 As String = Nothing
            Dim Item4 As String = Nothing
            Dim Item5 As String = Nothing
            Dim strDisplyName As String = Nothing
            Dim GroupName As String = Nothing
            For Each Item As SearchResult In users 'in recuper les entrées trouver
                Try
                    Item2 = CStr(Item.Properties("sAMAccountName").Item(0))
                Catch
                    Item2 = ""
                End Try
                Dim lv As ListViewItem = (ListView1.Items.Add(Item2))
                If Not (Item.Properties.Contains("groupType")) Then
                    Item1 = CStr(Item.Properties("sAMAccountName").Item(0))
                    Item1 = "User"
                Else
```

6. Présentation de quelques interfaces :

6.1.Page d'accueil :

Avant d'accéder à la page principale de l'application, l'administrateur doit s'authentifier, et ça pour s'assurer que d'autres personnes n'utilise pas l'application en se passant pour l'administrateur.

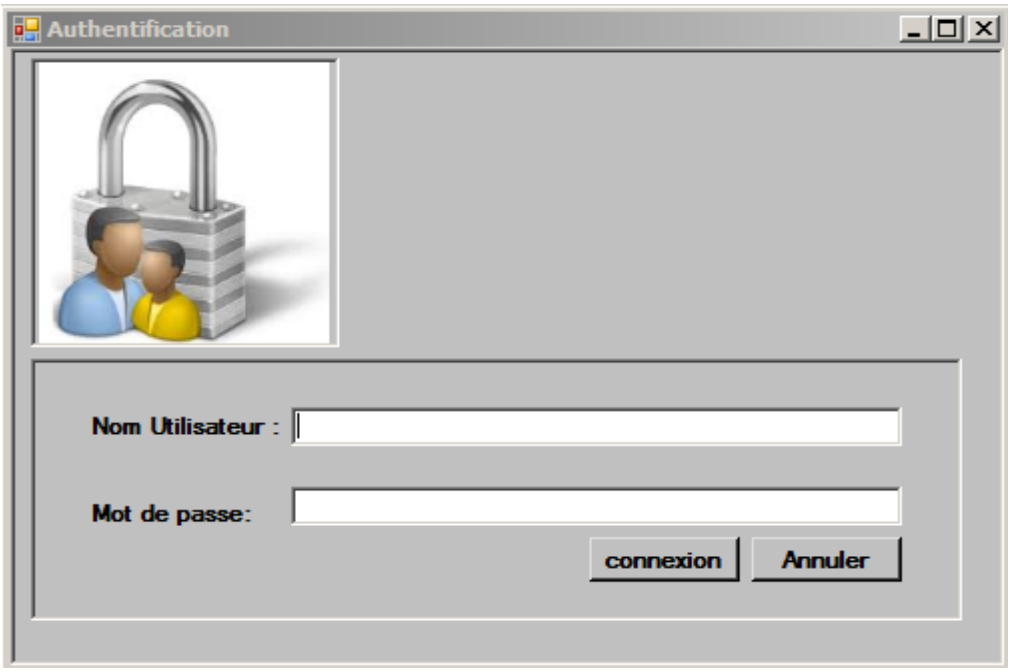


Figure 6.1.1.Page d'authentification

Un message d'erreur sera affiché si le mot de passe ou le nom d'utilisateur sont incorrecte

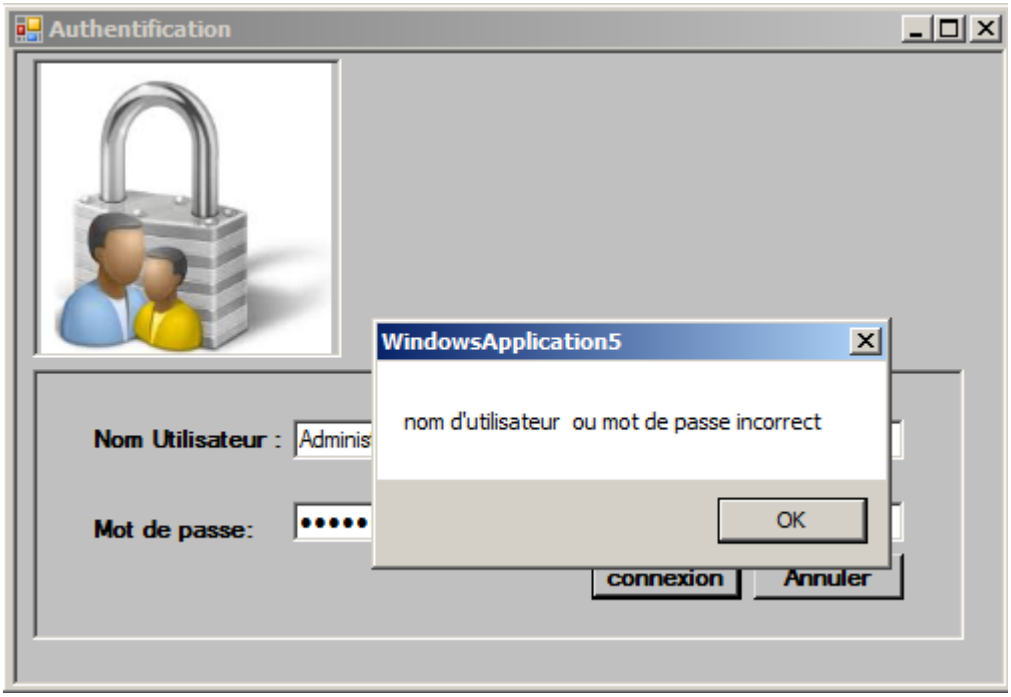


Figure 6.1.2.Message d'erreur

6.2.Page d'ajout d'utilisateur :

Cette interface permet d'ajouter un utilisateur a la liste existante en saisissant les attributs du l'utilisateur en suite en clique sur valider.

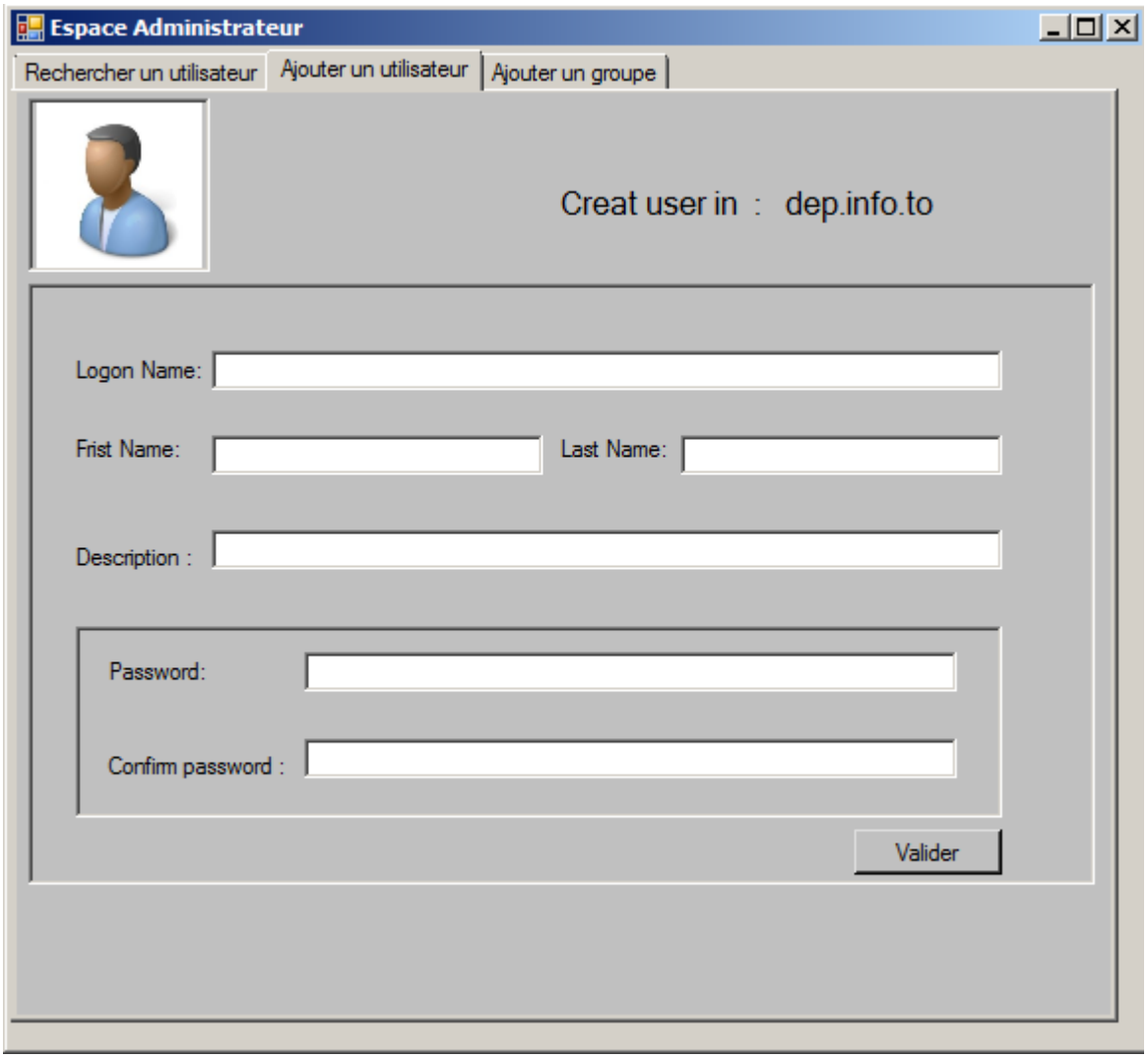


Figure 6.2.Page d'ajout d'utilisateur

6.3.Page d'ajout du groupe :

Cette interface permet d'ajouter un groupe a la liste existante en saisissant les attributs du groupe en suite on clique sur valider pour l'ajout.

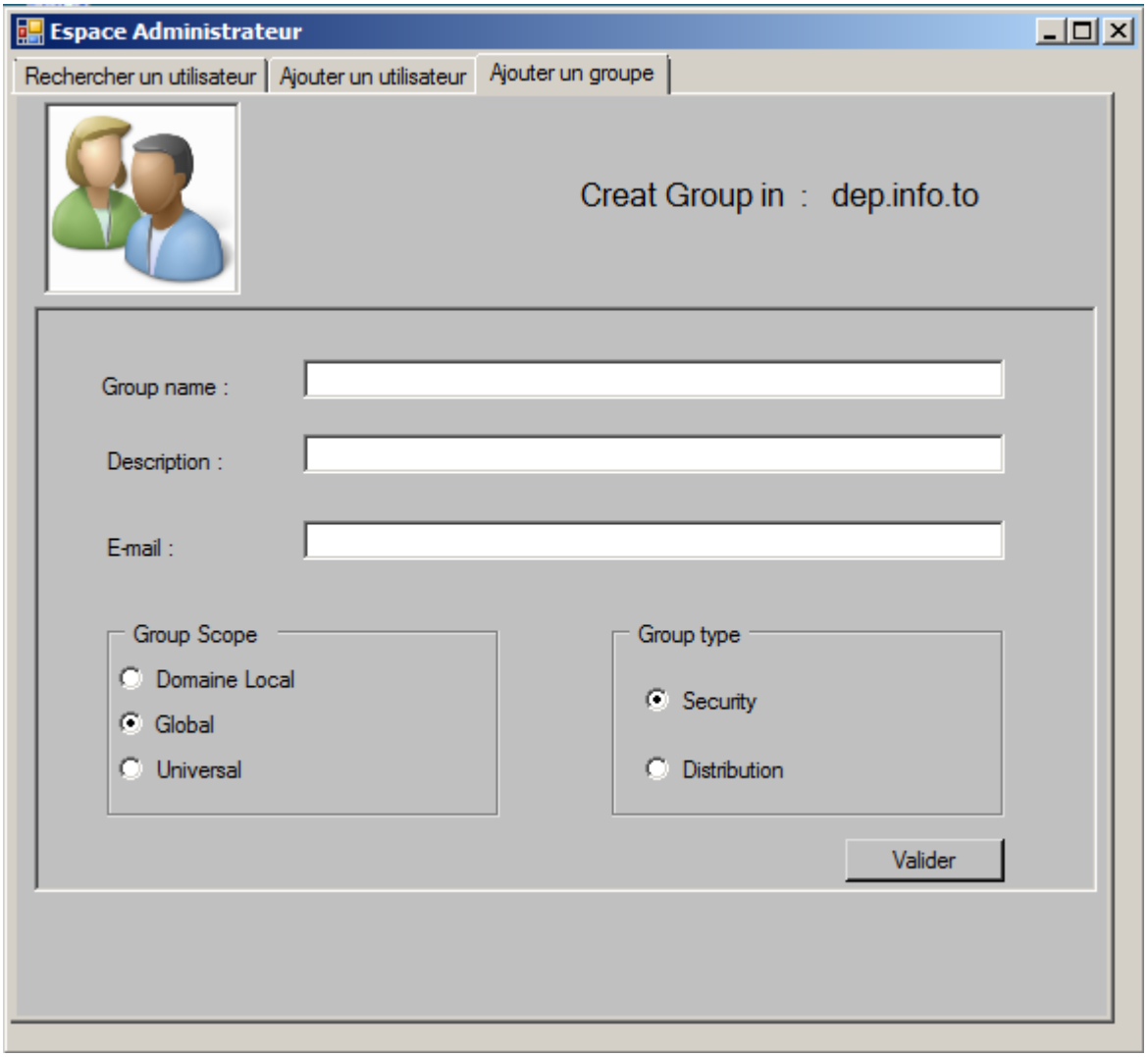


Figure 6.3.Page d'ajout d'un groupe

6.4.Recherche d'objet:

Cette interface permet de rechercher d'un objet

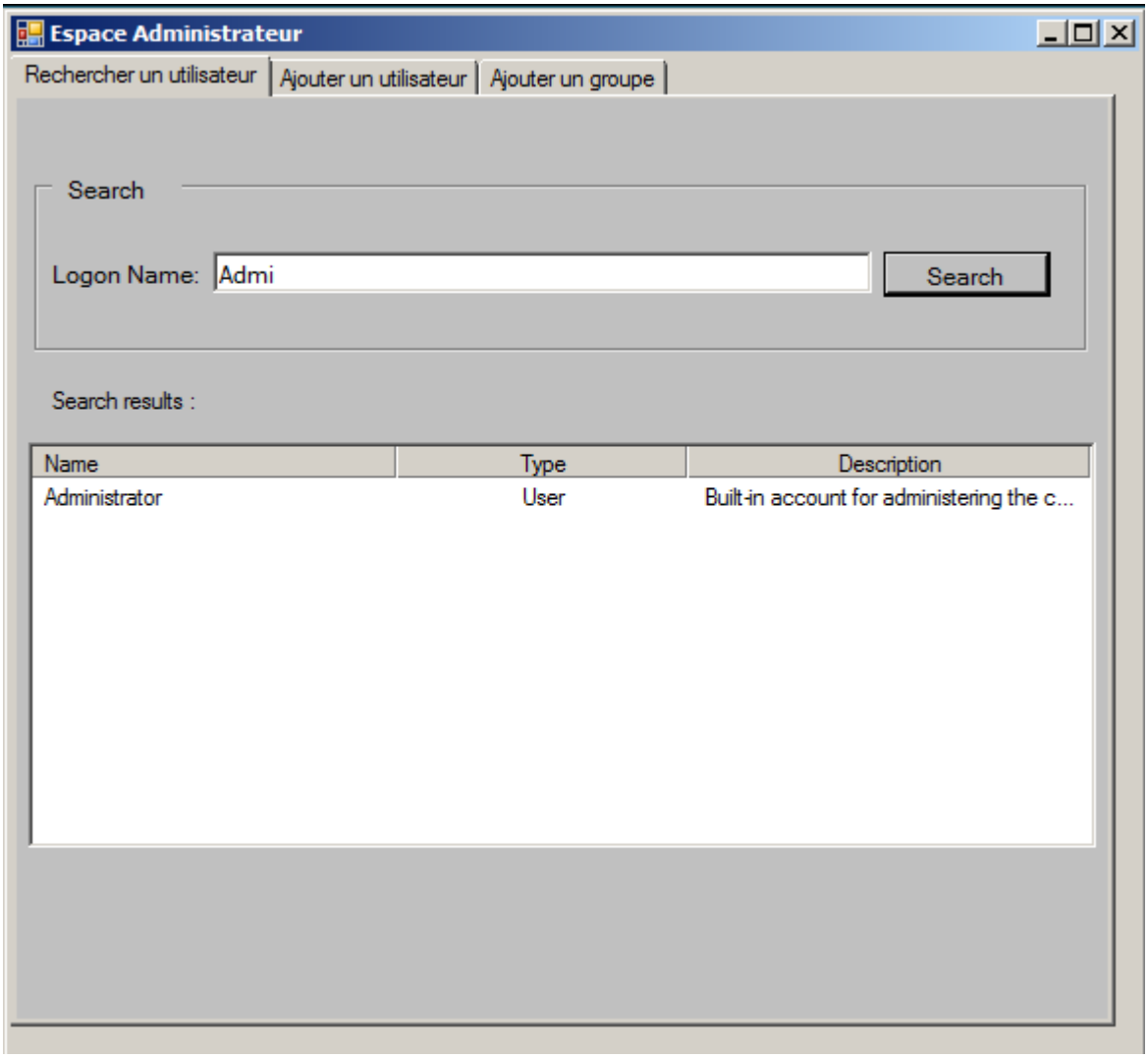


Figure 6.4.Page de recherche d'objet

6.5.Propriété d'utilisateur :

Cette interface ci-dessous noue affiche les propriétés générales de chaque objet

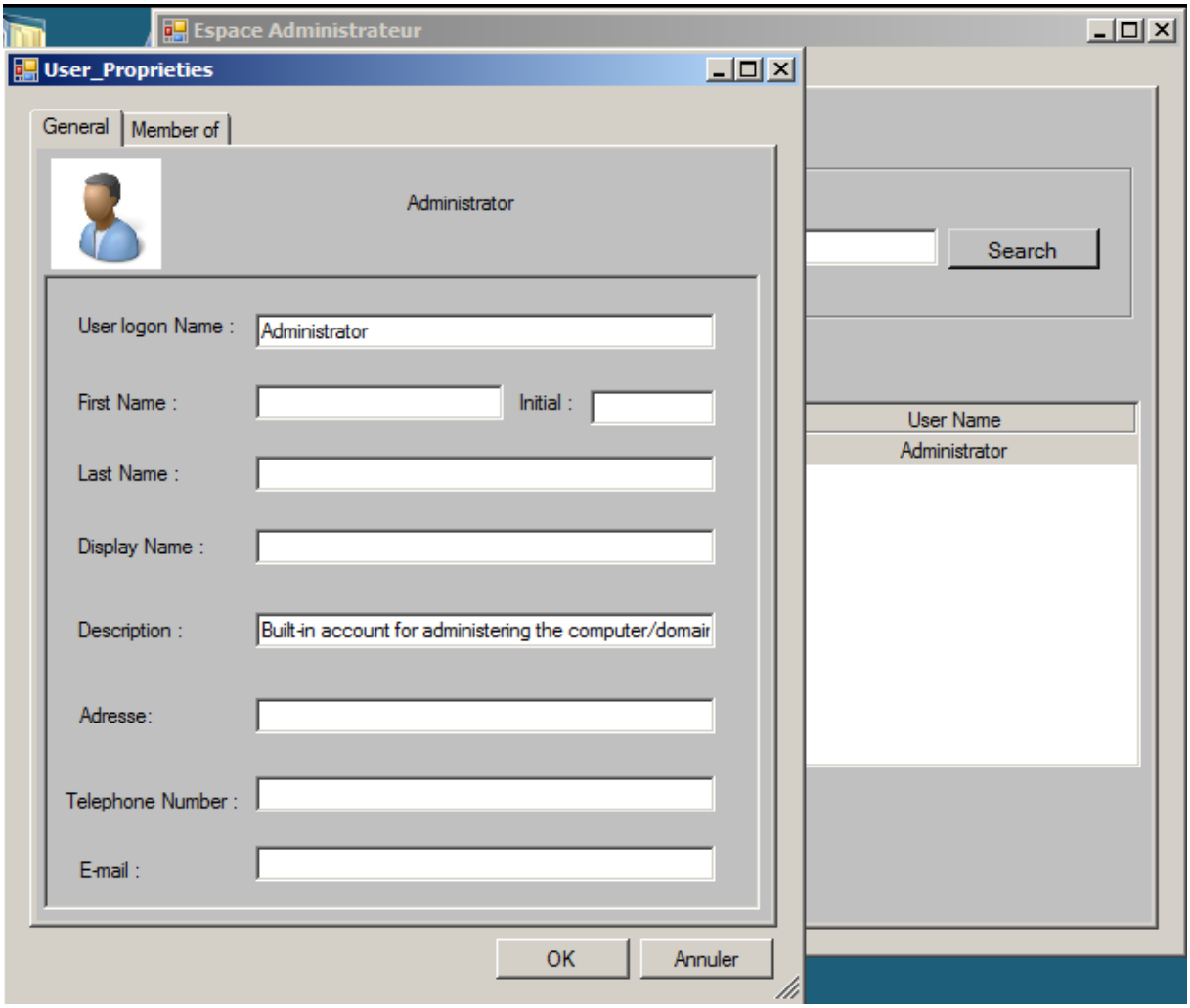


Figure 6.6.Page propriété d'un utilisateur

Conclusion :

Cette ultime phase de développement de notre application, nécessitant un outil de modélisation et un environnement programmation, nous a permis d'approfondir nos connaissances non seulement dans le langage de programmation Visual Basic mais aussi sur les concepts du langage de modélisation UML.

CONCLUSION GÉNÉRALE

Conclusion générale

A l'heure actuelle, l'informatique est omniprésente dans la vie d'une entreprise. En effet, la bonne gestion de l'entreprise dépend de l'outil informatique. Vu que cette dépendance est nettement en hausse, divers outils ont été mis en place pour permettre une meilleure gestion des systèmes d'information. Active Directory est l'un de ces outils que nous avons utilisé pendant la réalisation de notre travail.

Nous avons de ce fait installé et configuré le service d'annuaire Active Directory dans Windows servers 2008.nous avons réalisé un espace administratif à travers des interfaces simples et conviviales pour que l'administrateur ait la possibilité de gérer les comptes utilisateurs de manière simple et rapide.

Ainsi, à travers les différentes étapes de développement de notre application, de la spécification au codage, nous avons pu mettre en pratique des connaissances acquises au cours de notre cursus mais aussi approfondir d'autres concernant UML ou encore le langage de programmation visuel basic.

En guise de perspectives, les contributions suivantes peuvent être apportées à notre travail :

De créer plusieurs domaines pour implémenter la relation d'approbation et pour mieux comprendre le fonctionnement du protocole Kerberos.

De gréer tous les objets d'Active Directory.

BIBLIOGRAPHIE

Bibliographie

[09]<<LDAP :concepts,deploiement>>-laurent mirtain-c.calveleira,c.gross.

[11]<<Administrateur reseaux>> A.Guermouche

[12]<<Exposé nouvelle technologie reseau LDAP>>sylvain pernot-sebastien larué.

[13] <<Administation système deployement de LDAP>>-stephane galland.

[16] <<The LDAP Content Synchronisation Operation – IETF Internet Draft >>-<draft-zeilenga-ldup-sync-06.txt> ,Zeilenga (K.), Choi (J.) –, September 2004

[17]<<LDAp(lighweight directory access protocol)>>-annewei-CNAM PARIS.

22]<< Domain names - concepts and facilities >> Mockapetris (P.), November 1987

[18] www.actifdirectory.pdf

[20] www.module9.pdf (installation active directory)

[21] www.architectureactivedirectory.html

[14] www.definitionldap.com

Les mémoires de fin d'étude :

[4] conception et realisation d'une application WEB pour la gestion et le suivi du partimoine informatique.

Réalisé par :M^{r} Kacel Bilal,Boukellal Mohand-arezki

Diriger par : M^{r} Redaoui samir.

[5] Conception et Réalisation d'une application de suivi des affaires client de la SONELGAZ
Cas : SONELGAZ de Tizi Ouzou

Dirigé par: M^{r}. RAMDANE ,Mr. CHERIEF
Réalisé par :M^{me} MANANE Dalila, M^{elle}. CHERIEF HADJ-ALI Fatiha