Faculté des sciences exactes - Département informatique
Mémoire de fin de cycle en vu d'obtention d'un diplôme
Master **P**rofessionnel en
Administration et sécurité des réseaux

Thème

Conception et réalisation d'une application web pour la gestion des services réseau : DNS, DHCP et SQUID sous Linux

Président de jury: M^r^ ACHROUFEN
Examinateurs: M^me^ BELKHIRI Luiza
M^me^ BRAHMI Nassima

Présenté par :
M^lle^ ABASSI Samia
M^lle^ ABDELFETTAH Fahima

Encadreur : M^me^ Ferroudja ZIDANI

Année 2011-2012

Remerciements

Nos remerciements vont avant tout Au bon dieu qui nous a donné le courage et la patience de réaliser ce modeste travail.

Notre profonde gratitude aux membres de jury qui ont accepté d'´evaluer notre modeste travail.

On remercie vivement et particulièrement notre promotrice: Mme. Ferroudja ZIDANI, pour ses conseils précieux, ses remarques pertinentes et qui a su nous transmettre son expérience.

Nous adressons aussi nos vifs remerciements à nos familles respectives qui ont mis à notre disposition les moyens nécessaires pour la finalisation de ce présent travail.

Notre profonde gratitude à tous nos amis de la promotion qui nous ont beaucoup aidés durant ce projet par leurs conseils et encouragements.

A toute personne ayant contribué de prêt ou de loin à l'aboutissement du projet.

On dédie ce modeste travail

A nos très chers parents,

A nos familles,

A nos amis…

Liste des tableaux

Liste des figures

Sommaire

Introduction générale

La gestion de l'infrastructure réseau est la tâche quotidienne de tout administrateur réseau. Il s'agit de superviser le réseau, de définir des procédures et de gérer les mots de passe, de prendre en charge le suivi des sauvegardes et résoudre les éventuels incidents qui peuvent survenir. Au-delà, anticiper les évolutions technologiques et intégrer de nouveaux outils de gestions, vérifier le fonctionnement optimal de chaque matériel, paramétrer celui-ci, ou de le mettre à jour régulièrement.

Ainsi, dans les réseaux informatiques d'aujourd'hui, la gestion est un problème d'actualité. Le recours à des techniques d'administration efficace est important pour une bonne gestion des moyens de communication dont on dispose. C'est à ce niveau qu'intervient l'administration réseau.

Notre projet consiste à réaliser une application web dynamique pour la gestion des services réseau de l'université de Bejaia. Cette application nous permettra d'assurer les taches que doit effectuer l'administrateur réseau en temps minimal.

A travers ce document, nous allons présenter le travail en quatre chapitres : Nous allons commencer par quelques généralité sur les réseaux et internet, où on va citer les topologies existantes et les principaux services d'internet pour introduire le contexte de notre application, puis dans le chapitre deux une vue globale sur l'administration réseau et les services réseau concernés, suivie d'une analyse et de conception du système crée et on terminera par la réalisation.

Chapitre I

Généralités sur
Les réseaux et Internet

Introduction

Le réseau informatique s'est progressivement intégré dans la société afin de devenir aujourd'hui un outil communément utilisé pour communiquer et échanger tout type d'information. Chaque jour, dans le monde entier, des millions de personnes se connectent pour consulter leurs courriers électroniques, pour chercher des informations pratiques, pour enrichir leurs connaissances ou plus simplement pour discuter avec d'autres internautes.

I. Réseaux

I.1. Définition

Un réseau est un ensemble d'ordinateurs et périphériques interconnectés. Il permet de faire circuler des données informatiques et ainsi d'échanger du texte, des images, de la vidéo ou du son entre chaque équipement selon des règles et protocoles bien définis **[W1]**.

I.2. Topologies de réseaux

Il convient de distinguer la topologie logique de la topologie physique:

La topologie logique : décrit le mode de fonctionnement du réseau, la répartition des nœuds et le type de relation qu'ont les équipements entre eux **[W2]**.

La topologie physique : Décrit la mise en pratique du réseau logique (câblage). Il y a plusieurs topologies, pour des performances différentes. A savoir, les débits, le nombre d'utilisateurs maximum, le temps d'accès, la tolérance aux pannes, la longueur de câblage et les types d'applications différentes **[W13].**

Les différentes topologies de réseaux sont les suivantes :

I.2.1. Topologie en bus

Topologie réseau dans laquelle des noeuds multiples accèdent à un même bus partagé. Cette topologie est l'organisation la plus simple d'un réseau. Cette topologie a pour avantages d'être facile à mettre en œuvre **[W2]**.

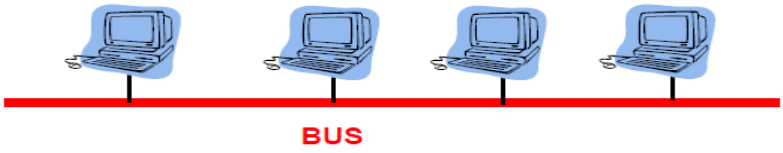

Figure1: Topologie en bus **[W3]**

I.2.2. Topologie en étoile

Dans cette topologie chaque ordinateur ou imprimante est relié au nœud central. Les performances d'un réseau dépendent principalement du nœud central. C'est un type de réseau relativement efficace et économique **[W2]**.

Figure2 : Topologie en étoile **[W3]**

I.2.3. Topologie en anneau

Organisation de réseau en anneau fermé où les systèmes présents s'échangent un ou des jetons (token), qui leur donne le droit d'envoyer des données sur le réseau. Le débit est de l'ordre de quelques Mbit/s, Les deux principales topologies logiques utilisant cette topologie physique sont Token ring (anneau à jeton) et FDDI **[W2]**.

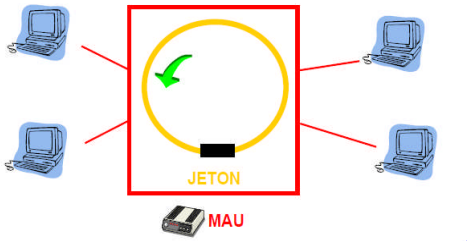

Figure 3 : Topologie en anneau **[W3]**

I.2.4. Topologie en maille

Topologie réseau dans laquelle des segments relient les nœuds dans maillage ne pouvant se réduire à un cas plus simple. Dans un maillage intégral (full mesh), chaque nœud est directement relié à tous les autres. L'usage le plus fréquent des réseaux maillés, sont des réseaux WAN **[W2]**.

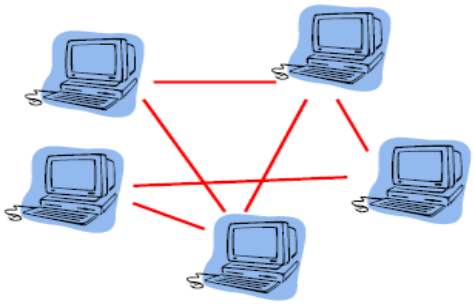

Figure 4 : Topologie en maille **[W3]**

I.3. Le modèle OSI (Open Systems Interconnections)

Lorsque les réseaux informatiques ont pris de l'importance, l'ISO (International Standards Organization) et l'UIT_T (International Télécommunications Union) ont décidé de créer un modèle de base pour décrire les différentes fonctions que doit remplir un réseau.

Ce modèle, baptisé modèle OSI (Open System Interconnect), est basé sur le principe suivant : les fonctions remplies par un système de télécommunication sont segmentées en couches permettant de diviser l'ensemble des fonctions en modules, possèdent chacun une tâche bien défini **[W4].**

I.3.1. Fonctions des couches

Le modèle OSI comprend 7 couches ayant chacune un rôle précis, comme le décrit les lignes suivantes.

Niveau1 : physique

Fournit les moyens mécaniques et électriques nécessaires à l'activation, au maintien et à la désactivation des connexions physiques destinées à la transmission des éléments binaires (bits) **[W4]**.

Niveau2 : Liaison de donnée

S'occupe de l'établissement, du maintien et de la libération des connexions, en se basant sur les moyens fournis par la couche physique. Elle s'occupe aussi du partage d'un support physique unique entre plusieurs machines **[W4]**.

Niveau3 : Réseau

Doit acheminer des paquets d'information jusqu'à leur destination finale. Elle gère donc les flux d'informations (en évitant les embouteillages), leur routage, ainsi que l'adressage. Elle s'appuie par les moyens fournis par les couches physiques et liaison.

Niveau4 : Transport

Complète les fonctions des couches précédentes en gérant les erreurs et en optimisant le transport **[W4]**.

Niveau5 : Session

Établit et maintient des sessions **[W4]**.

Niveau6 : Présentation

S'occupe de la mise en forme des données à envoyer dans un format compréhensible pour le destinataire. Elle gère par exemple le cryptage et la compression des données **[W4]**.

Niveau7 : Application

Au-dessus de toutes les autres, représente les applications qui utilisent la connexion réseau, comme par exemple un logiciel de courrier électronique ou un logiciel de transfert de fichiers **[W4]**.

I.4. Le modèle TCP/IP

TCP/IP est basé sur un modèle de référence à quatre couches. Tous les protocoles appartenant à la suite de protocoles TCP/IP sont situés dans les trois couches supérieures de ce modèle.

Comme le montre l'illustration suivante, chaque couche du modèle TCP/IP correspond à une ou plusieurs couches du modèle de référence à sept couches OSI **[W5]**.

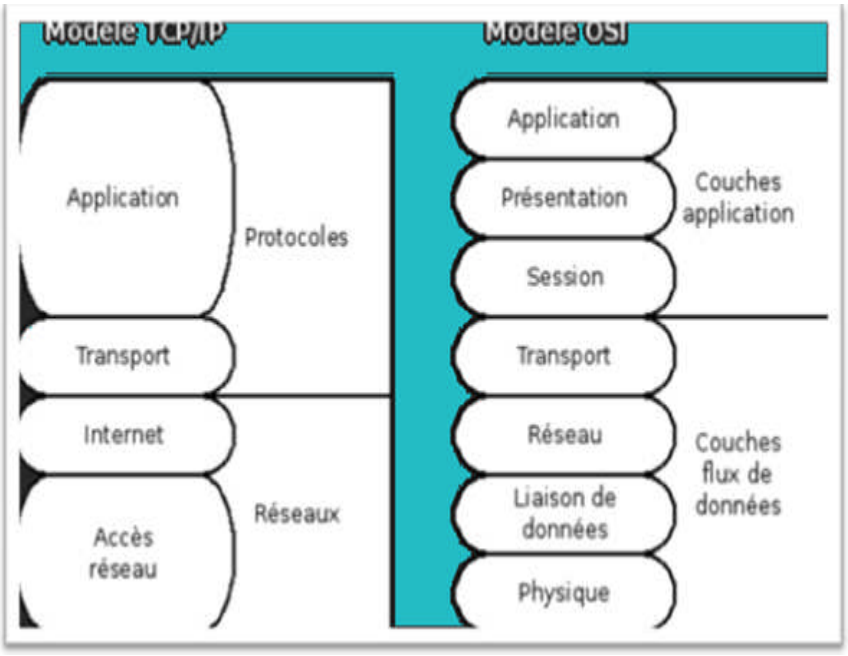

Figure 5 : comparaison entre le modèle TCP/IP et le modèle OSI **[W5]**

Les types de services proposés et quelques protocoles utilisés sur chaque couche du modèle TCP/IP sont décrits en détail dans le tableau suivant **[W5]**.

Couches	**Description**	**Protocole**
Application	Définit les protocoles d'application TCP/IP et explique comment l'hôte programme l'interface avec les services de couches de transport pour utiliser le réseau.	HTTP, Telnet, FTP, TFTP, SNMP, DNS,DHCP,SMTP, autres protocoles d'application
Transport	Propose la gestion des sessions de communication entre les ordinateurs hôtes. Définit le niveau de service et l'état de la connexion utilisés lors du transport des données.	TCP, UDP, RTP
Internet	Regroupe les données en datagrammes IP qui contiennent des informations sur les adresses de source et de destination utilisées pour transmettre les datagrammes entre les hôtes et à travers les réseaux. Effectue le routage des datagrammes IP.	IP, ICMP, ARP, RARP

Interface réseau	Donne des détails sur le mode d'envoi physique des données à travers le réseau, y compris sur la façon dont les bits sont électriquement signalés par les périphériques matériels jouant directement le rôle d'interface avec un support réseau, comme un câble coaxial, une fibre optique ou un fil de cuivre à paire torsadée.	Ethernet, Token Ring, FDDI, X.25, Frame Relay, RS-232, v.35

Tableau1 : Les 4 couches du modèle TCP/IP **[W2]**

I.4.1. Adressage IP

Pour que l'acheminement des données fonctionne sur un réseau TCP/IP, chaque ordinateur doit posséder une adresse IP unique.

I.4.1.1. Structure des adresses IPV4

Les adresses IP sont des nombres de 32 bits qui contiennent 2 champs :

- **Un identificateur de réseau (NET-ID):** tous les systèmes du même réseau physique doivent posséder le même identificateur de réseau, qui doit être unique sur l'ensemble des réseaux gérés.
- **Un identificateur d'hôte (HOST-ID):** il identifie une station de travail, un serveur, un routeur ou tout autre périphérique TCP/IP au sein du réseau.

La concaténation de ces deux champs constitue une adresse IP unique sur le réseau.

Pour éviter d'avoir à manipuler des nombres binaires trop longs, les adresses 32 bits sont divisées en 4 octets. Ce format est appelé la notation décimale pointée, cette notation consiste à découper une adresse en quatre blocs de huit bits. Chaque bloc est ensuite converti en un nombre décimal.

Chacun des octets peut être représenté par un nombre de 0 à 255 **[w07].**

I.4.1.2. Classes d'adresses

La communauté Internet a défini trois classes d'adresses appropriées à des réseaux de différentes tailles. Il y a, a priori, peu de réseaux de grande taille (classe A), il y a plus de réseaux de taille moyenne (classe B) et beaucoup de réseaux de petite taille (classe C). La taille du réseau est exprimée en nombre d'hôtes potentiellement connectés.

Le premier octet d'une adresse IP permet de déterminer la classe de cette adresse **[w07].**

I.4.1.3. Masque de sous réseau

Le rôle du masque de réseau (netmask) est d'identifier précisément les bits qui concernent le numéro de réseau et le numéro d'hôte d'une adresse.

Un bit à 1 dans le masque, précise que le bit correspondant dans l'adresse IP fait partie du numéro de réseau **;** à l'inverse, un bit à 0 spécifie un bit utilisé pour coder le numéro d'hôte**.**

Ainsi, on a un masque dit "par défaut" qui correspond à la classe de ce réseau. Exemple: dans un réseau de classe A sans sous-réseau, le premier octet correspond à l'adresse du réseau donc le netmask commence par 11111111 suivi de zéros soit 255.0.0.0 **[w07].** D'où le tableau suivant :

Classe	Netmask
A	255.0.0.0
B	255.255.0.0
C	255.255.255.0

Tableau2 : masque de sous réseau **[w07]**

I.4.1.4. Adresses réservées

Les adresses réservées ne peuvent désigner une machine TCP/IP sur un réseau.

L'adresse d'acheminement par défaut : (route par défaut.) Tous les paquets destinés à un réseau non connu, seront dirigés vers l'interface désignée par 0.0.0.0.et également l'adresse utilisée par une machine pour connaître son adresse IP durant une procédure d'initialisation (DHCP).

L'adresse de bouclage (loopback): l'adresse de réseau 127 n'est pas attribuée à une société, elle est utilisée comme adresse de bouclage dans tous les réseaux. Cette adresse sert à tester le fonctionnement de votre carte réseau. Un ping 127.0.0.1 doit retourner un message correct. Le paquet envoyé avec cette adresse revient à l'émetteur.

Toutes les adresses de type 127.X.X.X ne peuvent pas être utilisées pour des hôtes. La valeur de 'x' est indifférente. On utilise généralement **127.0.0.1**

L'adresse de diffusion : est une adresse dont tous les bits d'hôte sont positionnés à 1 (ex : 128.10.255.255 adresse de diffusion du réseau 128 de classe B).

Elle est utilisée pour envoyer un message à tous les postes du réseau.

II. Internet

II.1. Définition d'Internet

Réseau informatique mondial constitué d'un ensemble de réseaux nationaux, régionaux et privés qui sont reliés par le protocole de communication TCP/IP et qui coopèrent dans le but d'offrir une interface unique à leurs utilisateurs.

L'Internet est aussi un système d'échange de documents électroniques : textes, fichiers, images, sons et séquences audiovisuelles ainsi que d'autres services **[W6].**

II.2. Les services d'Internet

L'architecture logicielle du réseau Internet fonctionne sur le mode client/serveur, c'est-à-dire qu'un ordinateur relié au réseau Internet peut demander des informations à un ordinateur serveur, envoyer des informations à un ordinateur client ou encore faire les deux à la fois.

Le type de service délivré par un serveur à un client est différent selon le protocole de communication établi entre les deux ordinateurs.

Dans les prochains paragraphes, nous allons décrire quelques services utilisables par l'internaute en débutant par le plus médiatisé d'entre eux : le World Wide Web **[w015].**

II.2.1. Word Wide Web

Le concept du Web repose sur la notion d'hypermédia, c'est à dire la réunion de documents multimédia (texte, son, image…) par l'intermédiaire de liens préétablis.

Le protocole utilisé est le HyperText Transfer Protocol (HTTP) qui permet de transférer à partir d'un serveur web des pages écrites dans le langage de programmation Hyper Text Mark-up Language (HTML) **[W14].**

II.2.2. Transfert de fichiers (FTP)

Le FTP (File Transfert Protocol) est un protocole de communication destiné à l'échange informatique de fichiers sur un réseau TCP/IP. Il permet, depuis un ordinateur, de copier des fichiers vers un autre ordinateur du réseau, d'administrer un site web, ou encore de supprimer ou de modifier des fichiers sur cet ordinateur **[W8].**

II.2.3. Accès a distance (Telnet)

Le protocole Telnet permet de se connecter à un serveur distant à l'aide d'un identifiant et d'un mot de passe. Il est utilisé pour exécuter des programmes à distance comme les bases de données et catalogues de bibliothèque **[W7].**

II.2.4. Les forums

L'ensemble des services permettant le rassemblement d'opinions sur un sujet particulier est regroupé sous la dénomination commune de forums, le but étant de

constituer une communauté virtuelle où chaque participant peut être lecteur (passif) et rédacteur (actif) **[W14].**

II.2.5. Les groupes de discussion (newsgroups)

Service hérité du réseau Usenet et crée en 1979. Il permet de rassembler et de classer selon une arborescence de thèmes les messages des internautes du monde entier. Ces messages sont stockés et dupliqués dans des serveurs de news et envoyés aux internautes qui en font la demande **[W14].**

II.2.6. Les listes de diffusion (Mailing List)

Elles reposent sur le même principe que les groupes de discussions mais en diffèrent par leur aspect privé nécessitant une inscription préalable, la présence d'un modérateur (optionnel dans les groupes de discussion) et par le mode de distribution des messages. Ici, chaque message posté par un des membres de la liste n'est plus stocké sur un serveur mais est directement diffusé à l'ensemble des boîtes aux lettres des participants **[W14].**

II.2.7. Courrier électronique (E-mail)

C'est un des services les plus couramment utilisés d'Internet permettant d'envoyer des messages sous forme de fichier texte à un ou plusieurs destinataires. **[W10]**

II.2.8. La visioconférence

La visioconférence est la technologie qui permet à plusieurs personnes distantes géographiquement de se regrouper dans un lieu virtuel matérialisé par leurs écrans afin de converser comme si elles étaient dans un même lieu bien réel.

Un logiciel fourni avec le matériel de visioconférence, permet le transfert de fichiers et le partage d'applications et de données sur un espace de l'écran commun aux différents interlocuteurs, mis à jour en temps réel.

II.2.9. Commerce électronique

On utilise souvent le terme commerce électronique pour parler du marketing, de la distribution, de la vente ou de la fourniture de marchandises ou de services par des moyens électroniques. Cependant il faut préciser que le commerce électronique existait bien avant Internet, l'EDI par exemple, mais c'est Internet par sa puissance et par son accès facile qui est arrivé à transformer les échanges et les relations entre entreprises et consommateurs, ce qui a donné un essor considérable au commerce électronique.

Dans le détail, on peut dire que le commerce électronique c'est :

- La présentation de ses biens et de ses services sur le web ;
- Le traitement des commandes et la facturation des clients en ligne ;
- L'automatisation du processus de traitement des questions de la clientèle ;
- Le paiement en ligne et la gestion des transactions électroniques **[W15].**

II.2.10. Téléphonie sur l'Internet ou Voice Over Internet Protocol (VOIP)

VoIP signifie communication vocale sur protocole Internet. Un service téléphonique VoIP est essentiellement un service qui utilise la technologie de communication appelée protocole Internet (ou IP) plutôt que les systèmes analogiques conventionnelles **[W11].**

II.2.11. Messagerie Instantanée

Une messagerie est un petit logiciel Internet permettant à un internaute connecté de rentrer en temps réel en communication avec des individus faisant partie d'une liste d'amis préétablie.

Le logiciel informe l'utilisateur de la présence en ligne des personnes faisant partie de sa liste et lui permet d'échanger avec eux instantanément des messages textes saisis au clavier **[W12].**

II.3. L'intranet

Le mot intranet est devenu familier depuis une décennie bien qu'il renferme différentes interprétations. Un intranet est avant tout lié aux protocoles d'Internet, IP, TCP et UDP, et à leurs différentes incarnations. Le paradigme intranet correspond au système d'information de l'entreprise utilisant les applicatifs d'Internet. L'intranet désigne aussi parfois l'infrastructure de l'entreprise pour réaliser ses communications internes **[B2].**

III. Le Web

III.1. Définition

Le "Web" (la toile d'araignée) est une application géante, qui utilise le réseau Internet, et rend possible l'échange d'informations sur ce réseau, notamment grâce à l'utilisation des navigateurs et du langage HTML. On dit du Web, que c'est une "couche logique d'Internet".

Lorsque deux personnes discutent entre elles, elles ont besoin de différents outils:

- Du matériel, qui permet physiquement l'échange de données: bouche, cordes vocales, oreilles(Internet).
- D'un mode de communication: une langue commune, qui leur permet de se comprendre, et rend possible l'échange d'idées (Le Web) **[Ww1].**

III.2. Les outils du Web

III.2.1. Le navigateur

Un navigateur web est un logiciel informatique qui permet d'utiliser le web. Pour être plus précis, ce type de logiciel permet de consulter le *World Wide Web* (WWW). L'utilisation la plus répandue de ces logiciels étant de visualiser les pages web et d'utiliser les liens hypertextes dans le but d'aller de pages en pages. Exemples : Mozilla fireFox, Google Chrome, Safari **[Ww2].**

III.2.2. Adresse Web URL

Chaque page Web possède une adresse unique définie une seule fois à travers le monde. Cette adresse s'appelle URL (Uniforme Resource Locator). La structure de l'URL est : <Protocole>://< nom du serveur : port>/<nom du fichier>/…, Exemple : http://www.pmarchand.fr/internet/index.html.

Signification des différents composants d'une URL

- Le protocole couramment utilisé sur le Web est HTTP (HyperText Transfert Protocol), protocole de transmission de document hypertextes.
- Le serveur est compris entre les deux slashs de début et la barre oblique suivante ou va jusqu'à la fin de l'URL. Les pages Web demandées se trouvent sur cette machine.
- Chaque page Web est un fichier dont le chemin est indiqué dans la troisième partie de l'URL.
- Le port est le numéro qui caractérise le canal sur le quel le serveur attend les requêtes provenant des différents clients, par exemple, Telnet est identifié par le numéro 23, la messagerie sur 25, Gopher 70, le port par défaut est le port numéro 80…etc.

Les documents HTML utilisent assez souvent un raccourci pour faire les liens sur d'autres documents situés sur le même serveur, on parle d'URL relatives, ce raccourci est transformé par le Navigateur en une URL complète (absolue) avant d'envoyer la requête sachant que le document situé sur l'adresse absolue contient une référence sur l'adresse relative **[B1].**

III.2.3. Serveur Web

Un serveur web est un ordinateur connecté à Internet qui héberge des données ou fichiers et qui gère les requêtes provenant des navigateurs des internautes.

Les serveurs peuvent être spécialisés en fonction des types de données envoyées (html, vidéo, etc) et en fonction de leur rôle (serveur de données, serveur d'applications) **[Ww3].**

La convivialité des systèmes clients (navigateurs ou browser) et la richesse fonctionnelle apportée par le concept d'hyperliens (liens entre documents) ont contribué au succès rapide des serveurs Web qui sont devenu le standard du développement de base d'information sur le Net. Parmi les serveurs les plus utilisés, on cite le serveur **Apache [B1].**

III.3. Le fonctionnement d'une application web

III.3.1. Définition d'une application web

Une application web est un logiciel applicatif manipulable grâce à un navigateur Web (ex : Firefox, Internet Explorer ou Safari). Contrairement à une application classique, l'application web est installée sur un serveur distant et accessible par l'utilisateur depuis un ordinateur connecté au réseau Internet, sans installation préalable sur le poste. L'usage du navigateur web comme logiciel client permet la disponibilité de l'application sur tous les systèmes d'exploitation équipés d'un navigateur compatible (Windows, Mac OS, Linux, …) **[Ww4]**

III.3.2. Fonctionnement d'une application web statique

Avec application Web statique, la seule interactivité dont dispose l'internaute est de pouvoir passer d'une page HTML à l'autre par un simple clic sur les liens hypertextes présents sur une page. À chaque fois que l'internaute clique sur un lien, une requête HTTP est envoyée, établissant du même coup une communication avec le serveur. Cette communication est de type synchrone, c'est-à-dire que dès l'émission de la requête, la communication reste en place jusqu'à la réception de la réponse du serveur. Pendant le temps de traitement de la requête, le navigateur reste figé, bloquant ainsi toute action possible de l'internaute. À chaque requête, le serveur retournera une réponse sous la forme d'une page HTML complète comme le montre la figure suivante **[B5]** :

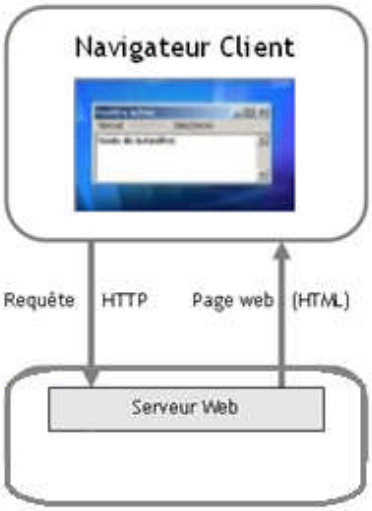

Figure 6 : Fonctionnement d'une application web statique

III.3.3. Fonctionnement d'une application web dynamique

Nous avons vu précédemment le traitement d'une simple requête par le serveur mais d'autre cas peuvent se produire, notamment lors de l'envoi d'un formulaire.

Dans ce cas, la requête est constituée d'une ligne de requête (précisant la méthode utilisée et le protocole HTTP), d'un corps (qui contient les données envoyées au serveur dans le cas d'une requête émise avec la méthode POST) et d'une série d'en-têtes qui définit les spécificités de la requête (nature du navigateur utilisé, type d'encodage…) qui permettront au serveur de traiter correctement les informations. En général, lors de l'envoi d'un formulaire, le traitement côté serveur est réalisé par une page contenant un programme (en Perl par exemple).

Les données reçues pouvant être traitées directement par le programme ou entraîner un échange avec un serveur de base de données afin de les mémoriser ou d'émettre une requête SQL. À l'issue de ce traitement, une nouvelle page web sera construite à la volée et renvoyée au navigateur, ce qui clôturera le processus, débloquant le navigateur de la même manière qu'avec un site statique. Comme l'illustre la figure suivante **[B5]**:

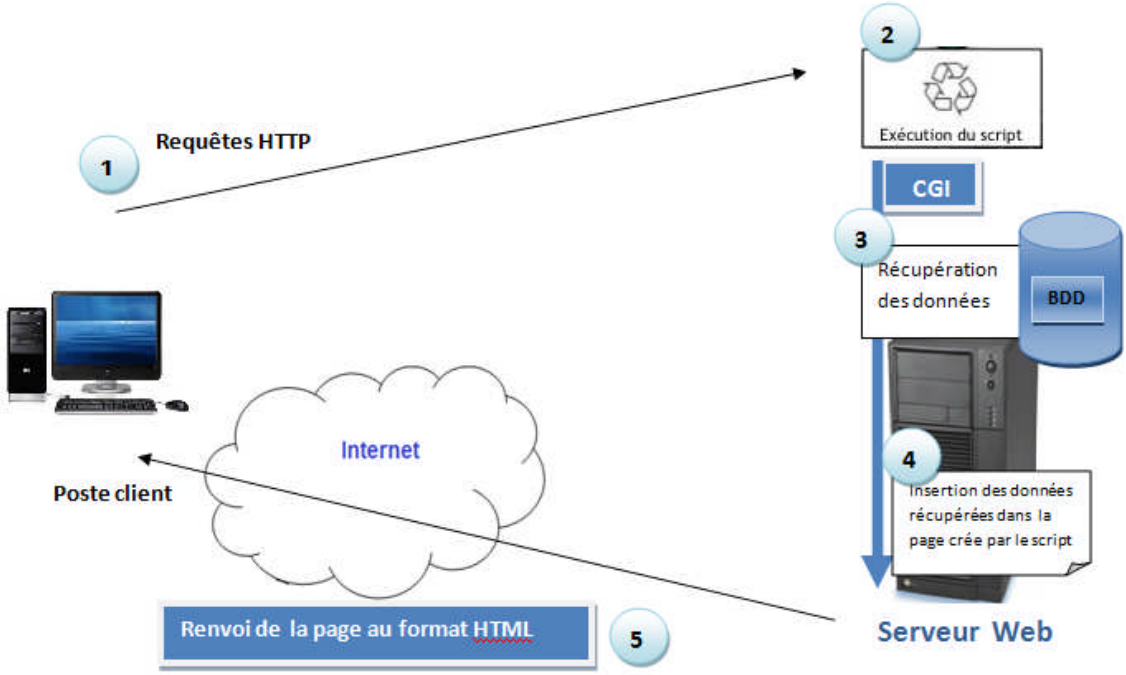

Figure7 : Fonctionnement d'une application web dynamique **[Ww4]**

IV. Architecture Client/Serveur

IV.1. Présentation de l'architecture d'un système Client/Serveur

De nombreuses applications fonctionnent selon un environnement client/serveur, cela signifie que des machines clientes (machines faisant partie du réseau) contactent un serveur, une machine généralement très puissante en termes de capacités d'entrée-sortie, qui leur fournit des services. Ces services sont des programmes fournissant des données telles que l'heure, des fichiers, une connexion, etc. Les services sont exploités par des programmes, appelés programmes clients, s'exécutant sur les machines clientes. Dans un environnement purement client/serveur, les ordinateurs du réseau (les clients) ne peuvent voir que le serveur, c'est un des principaux atouts de ce modèle **[Ww5].**

IV.1.1. Fonctionnement d'un système client/serveur

Le client : C'est le navigateur de l'internaute.

Le serveur web : est un ensemble ordinateur/logiciel paramétré pour pouvoir traiter certains types de pages et notamment celles qui contiennent des instructions de programmation. Il reconnaît ces pages grâce à l'URL qu'il reçoit, effectue les traitements demandés et transmet le résultat au format html au browser de l'internaute **[Ww10].**

Un système client/serveur fonctionne comme suit:

- le client émet une requête vers le serveur grâce à son adresse et le port, qui désigne un service particulier du serveur.
- Le serveur reçoit la demande et répond à l'aide de l'adresse de la machine client et son port.

IV.1.2. Types d'architecture client /serveur

IV.1.2.1. Architecture à deux niveaux

Ce type d'échange, bien construit, permet une totale transparence au réseau : chaque document peut se trouver n'importe où sur Internet, aussi bien sur la même machine que le client qu'aux antipodes. Dans la plupart des cas, les documents servis étant indépendants (en termes informatiques) les uns des autres, peut importe le serveur qui fournit la donnée. Différents éléments d'une même page (HTML, images, CSS) peuvent donc être fournis par plusieurs serveurs; l'assemblage est réalisé par le navigateur **[Ww6].**

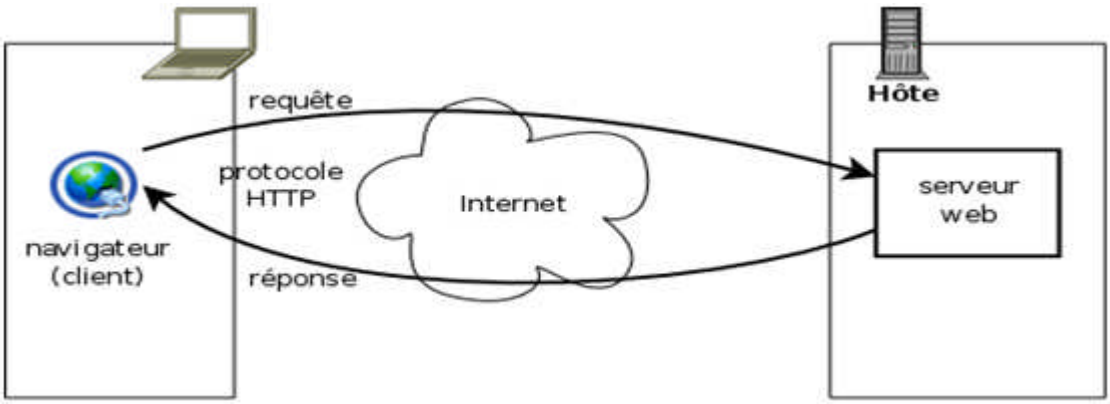

Figure8 : Architecture à deux niveaux

IV.1.2.2. Architecture à trois niveaux

Dans l'architecture à 3 niveaux (appelée architecture 3-tiers), il existe un niveau intermédiaire, c'est-à-dire que l'on a généralement une architecture partagée entre :

1. Un client, c'est-à-dire l'ordinateur demandeur de ressources, équipé d'une interface utilisateur (généralement un navigateur web) chargé de la présentation.
2. Le serveur d'application (appelé également **middleware**), chargé de fournir la ressource mais faisant appel à un autre serveur.
3. Le serveur de données, fournit au serveur d'application les données dont il a besoin **[Ww7].**

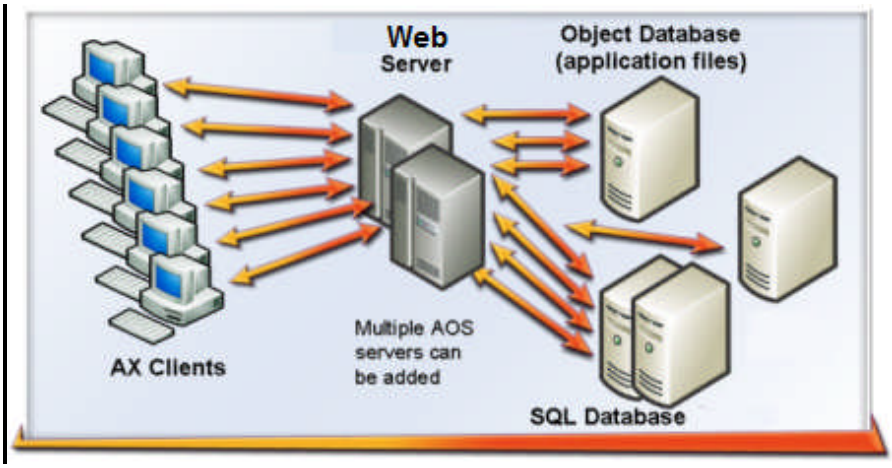

Figure9: Architecture à trois niveaux **[Ww8]**

IV.1.2.3. L'architecture multi-niveaux

L'architecture n-tiers a été pensée pour pallier aux limitations des architectures trois 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 trois tiers met en œuvre une approche objet pour offrir une plus grande souplesse d'implémentation et faciliter la réutilisation des développements.

Théoriquement, ce type d'architecture supprime les inconvénients des architectures précédentes :

- Elle permet l'utilisation des interfaces utilisateur riche ;
- Elle sépare nettement tous les niveaux de l'application ;
- Elle offre de grandes capacités d'extension ;
- Elle facilite la gestion des sessions **[Ww9]**.

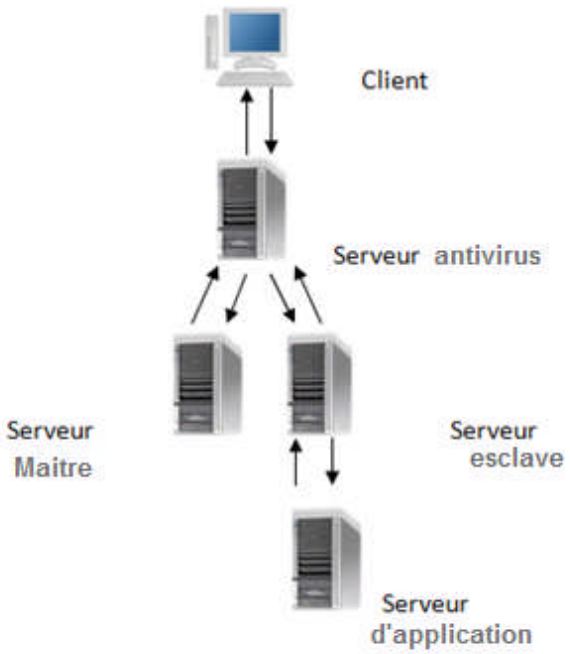

Figure10: exemple d'architecture à N niveaux

Conclusion

Les généralités abordées dans ce chapitre nous ont permis de nous familiariser avec les notions des réseaux et du net, afin d'introduire le contexte et l'environnement du développement de notre application. Dans ce qui suit nous verrons les différents services réseau existants et l'administration réseaux.

Chapitre II

Administration des services réseaux

Introduction

L'administration d'un réseau d'une entreprise consiste à gérer ses différents services réseaux DNS, DHCP et SQUID.

Ces services sont à la disposition de l'administrateur du réseau, ils lui permettent de vérifier le bon fonctionnement du réseau et de modifier ou corriger la configuration du réseau.

I. L'administration réseau

L'administration réseau est l'ensemble des moyens humains, techniques financières de contrôle, de coordination et de surveillance des livres de diverses ressources mises en œuvre **[artc8].**

On se focalisera dans ce qui suit sur le système d'exploitation Linux qui est utilisé par le commanditaire.

I.1. Le système d'exploitation Linux

I.1.1. Définition

Linux est un système d'exploitation, au même titre que Windows ou MacOS. Un système d'exploitation est tout simplement un énorme logiciel qui s'occupe de faire la liaison entre les logiciels, les utilisateurs et le matériel. Par exemple, pour taper un rapport ou une lettre, vous utilisez un traitement de texte. Lorsque vous appuyez sur une touche, c'est le système d'exploitation qui va détecter l'appui de cette touche et va transmettre un message à votre logiciel de traitement de texte qui interprétera ce message soit comme un caractère à afficher soit comme un raccourci clavier **[artc1].**

Linux est multi plateforme. Ce terme assez barbare veut tout simplement dire que Linux est disponible sur plusieurs types de machines. Ainsi, on va pouvoir trouver Linux sur une machine de type PC tel que vous connaissez et que vous utilisez certainement, mais aussi sur les Macintosh, ou encore sur les super calculateurs **[artc1].**

Linux est dans la majorité des cas Open Source. Sans rentrer dans les détails des licences Open Source, ceci veut dire que vous pouvez accéder directement au code source de façon gratuite (dans la plupart des cas) et librement (aussi dans la plupart des cas). Ceci veut donc dire que si vous êtes un bon programmeur, vous pouvez accéder au code source et le modifier selon vos souhaits pour l'adapter à vos besoins **[artc1].**

I.1.2. Architecture et fonctionnement de linux

I.1.2.1. Le noyau linux

Le noyau (kernel en anglais) d'un système d'exploitation est l'interface entre le matériel et les applications. Le noyau gère la mémoire, les périphériques (disques durs, carte son, ...) mais pas le mail ni l'interface graphique. Ce sont les applications qui font ça.

Souvent, on parle de Linux comme d'un système d'exploitation complet. En fait, c'est un abus de langage. Linux n'est que le noyau du système. Il est inutile sans les applications GNU qui se greffent dessus. On devrait normalement parler de GNU/Linux. La figure çi dessus illustre la composition et le rôle du noyau dans un système d'exploitation **[artc 2].**

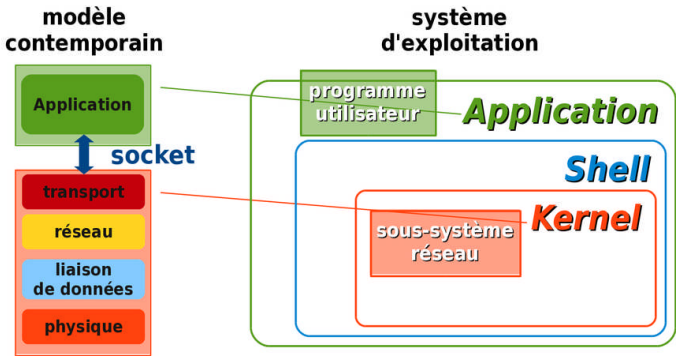

Figure 11 : Le rôle du noyau d'un système d'exploitation **[artc 5]**

I.1.2.2. L'interpréteur des commandes

Les différents langages informatiques peuvent être scindés en deux groupes : les langages compilés et les langages interprétés. L'écriture d'un programme dans un langage compilé commence par une phase de saisie du code source. Celui-ci est ensuite soumis à un utilitaire spécial nommé compilateur qui le transforme – souvent en plusieurs étapes – en code exécutable binaire, directement compréhensible par le microprocesseur. À l'inverse, un

programme écrit dans un langage interprété ne subit pas de modifications, mais il ne peut être exécuté directement par le processeur. Il faut qu'une application particulière – l'interpréteur – isole le code source et le décode instruction par instruction pour demander au microprocesseur d'exécuter telle ou telle tâche **[B5].**

I.1.2.3. Le système de gestion de fichiers

Toutes les applications informatiques doivent enregistrer et retrouver des informations. La solution consiste à stocker les informations dans des fichiers sur des disques. UNIX, et donc Linux, reconnaît deux types de dispositifs : les périphériques à accès direct par blocs (comme les disques), et les périphériques caractères (comme les bandes et les liaisons séries), dont certains peuvent être à accès direct et d'autres à accès séquentiel.

Chaque périphérique supporté est représenté dans le système de fichiers (filesystem) par un fichier spécial sous /dev. Lorsqu'on lit ou écrit dans un tel fichier, les données proviennent du, ou vont dans, le pilote logiciel du périphérique qu'il représente.
ex : less /dev/dev/hda, /dev/cdrom, /dev/ramdisk, /dev/console, /dev/mouse0, /dev/lp0. **[artc 6].** La figure suivante illustre l'arborescence des fichiers système :

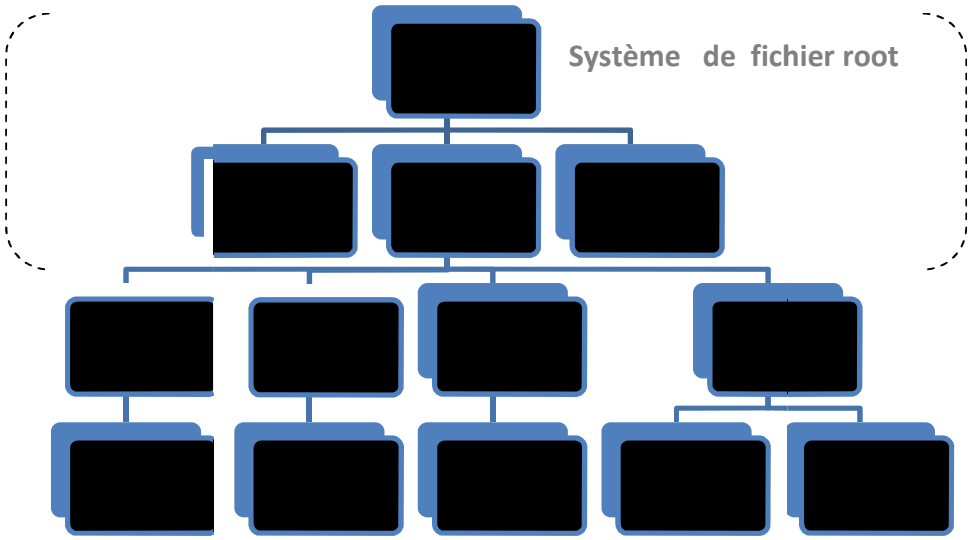

Figure 12 . L'arborescence des FS.

I.1.2. Quelques avantages de linux

Linux est un système :

- **Efficace**. Contrairement à des systèmes beaucoup plus répandus, il n'utilise pour ses besoins propres que très peu de ressources. Les logiciels que vous utilisez pour votre travail disposent donc de beaucoup plus de puissance pour fonctionner **[artc 3].**
- **Fiable**. Une machine sous Linux fonctionne 24h/24 si besoin sans se plaindre (si le matériel est prévu pour, en particulier au niveau thermique) **[artc 3].**
- **Robuste**. Une erreur d'un utilisateur ou un ``plantage'' éventuel d'une application n'affectent pas le reste du système. D'autre part, il est exceptionnel de devoir l'arrêter: la quasi-totalité des opérations de configuration, mise au point, etc, ne nécessitent pas l'arrêt du système **[artc 3].**
- **Très bon marché**. Le prix demandé par les sociétés qui vendent Linux sur CDROM ne sert qu'à couvrir leurs frais et à leur permettre de financer dans une certaine mesure la poursuite de cette activité. Linux étant développé par des passionnés pour le plaisir, personne n'a à supporter le coût de son développement **[artc 3].**

Enfin, Linux est conforme à la norme POSIX et aux standards du marché, en particulier de l'Internet. Cela signifie qu'un logiciel conçu pour un autre système de la même famille (Solaris de SUN, Digital Unix, AIX d'IBM, SCO Unix...) peut être rapidement porté sous Linux et vice-versa, ce qui assure une protection de l'investissement logiciel en cas d'obligation de changement de système **[artc 3].**

I.1.3. Inconvénients de linux

- La configuration est souvent moins simple (mais souvent plus puissante) que dans les logiciels commerciaux. Linux demande plus d'investissement intellectuel pour un débutant, mais cet investissement se trouve par la suite rentabilisée.
- Il y a beaucoup de docs traduites en français, mais beaucoup ne sont qu'en anglais. Il est difficile d'utiliser Linux si on ne connait pas un minimum d'anglais informatique.
- Certains types de logiciels tels que les jeux sont beaucoup plus fournis sous MS-Windows que sous Linux. Il existe des logiciels de bureautique pour Linux

(exemples : OpenOffice.org, AbiWord, KOffice) mais beaucoup moins que pour Windows.

- Linux n'est pas Windows ni Mac, il y a des choses différentes et il faut changer certaines habitudes pour ceux habitués à un autre OS. Des fois il faut lire de la doc.
- Il a besoin de plus de mémoire à la base que Windows. Si on veut faire tourner l'environnement graphique et des applis telles que Netscape, il faut compter un minimum de 32Mo **[artc 4].**

I.2. Administration d'un réseau sous linux

I.2.1. Définition

Sous linux comme sur tous les systèmes Unix, l'administration des services réseau consiste principalement à éditer et modifier des fichiers de configuration appropriés.

I.2.2. Le rôle de l'administrateur réseau

Trois grandes familles de tâches incombent à l'administrateur Unix : gérer le système, les services et la sécurité.

1. Gérer le système :

- Surveiller et assurer la bonne marche du système au quotidien :
- Surveiller les ressources (disque, mémoire, CPU, etc.).
- Planifier l'ajout de ressources.

2. Administrer les services déployés :

- Configurer ou modifier les fichiers de configuration des serveurs installés (DNS,DHCP,SQUID,...etc).
- Gérer les utilisateurs.
- Installer et configurer les applications.
- Planifier les migrations.

3. Prévoir et gérer les incidents et les intrusions (tâches transversales) :

- Installer les correctifs de sécurité.

- « Durcir » le système et les applications.
- Mettre en œuvre un plan de sauvegarde.
- Superviser le système et les applications **[artc8].**

I.3. Le contrôle d'accès sous linux

I.3.1. Rappel sur les permissions standards

Chaque objet du système de fichier est associé à trois groupes de permissions qui sont : propriétaire, groupe et autres. Pour chaque groupe trois types de permissions peuvent êtres attribuées : *read*, *write* et *execute*. Au total seul 9bits sont utilisés. Les permissions standard ont donc l'avantage de ne pas occuper beaucoup de place **[B4].**

I.3.2. Définition d'une ACL

Une ACL (Access Control List) est la propriété d'un objet contrôlant les droits d'accès à cet objet par des utilisateurs ou d'autres objets.

Lorsque l'on applique cette définition à un système de fichier Linux, le but d'une ACL est de permettre de définir une liste d'utilisateur associée à un objet (fichier ou répertoires) avec des permissions d'accès **[B4].**

Schématiquement :
Objet(fichier, périphérique…)-sujet(processus utilisateur)-droits (read,write,execute)

I.3.3. Pourquoi les ACL ?

Grace aux ACL on peut permettre à n'importe quel utilisateur, ou groupe, un des droits d'accès (lecture, écriture et exécution) sans être limité à un groupe et à un utilisateur propriétaire.

Voici un exemple de problème que nous pouvons avoir avec les permissions classiques :

On veut donner l'accès à des ressources (applications téléchargeables) à trois groupes d'utilisateurs spécifiques se connectant via l'intranet.

La solution consiste, avec les ACL, à attribuer simplement l'accès de ces trois groupes à nos ressources **[B4].**

I.3.4. Gestion avancée des Liste de contrôle d'accès

On utilise les commandes **getfacl** et **setfacl** pour consulter puis modifier les ACL sur un fichier.

Les ACL se divisent en fait en deux classes principales, l'ACL minimale et l'ACL étendue ; notions auxquelles il faut rajouter celle d'ACL par défaut (qui ne va s'appliquer qu'aux répertoires).

L'ACL minimale est exclusivement composé d'éléments de type « Propriétaire » (owner), « Groupe »
(owning group) et « Autres » (other) et correspond à la traduction littérale du modèle de gestion des droits Unix traditionnels que nous avons abordés au début de l'article : Les ACL ne remplacent pas mais étendent le modèle standard afin de préserver la compatibilité avec l'existant**[artc7]**. (Voir la figure)

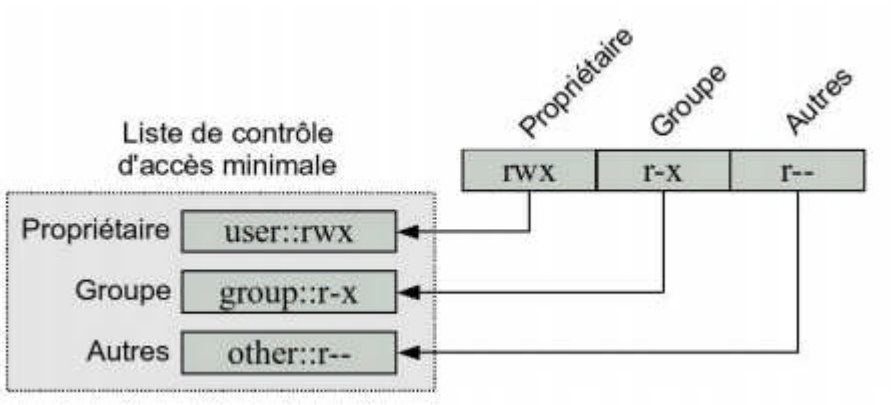

Figure13 : liste de contrôle d'accès minimale **[artc7]**

Une ACL étendue prolonge les droits de l'ACL minimale et introduit la notion de « Masque » (mask), d' « Utilisateurs nommés » (named user) et de « Groupes nommés » (named group) **[artc7]**. (Voir la figure)

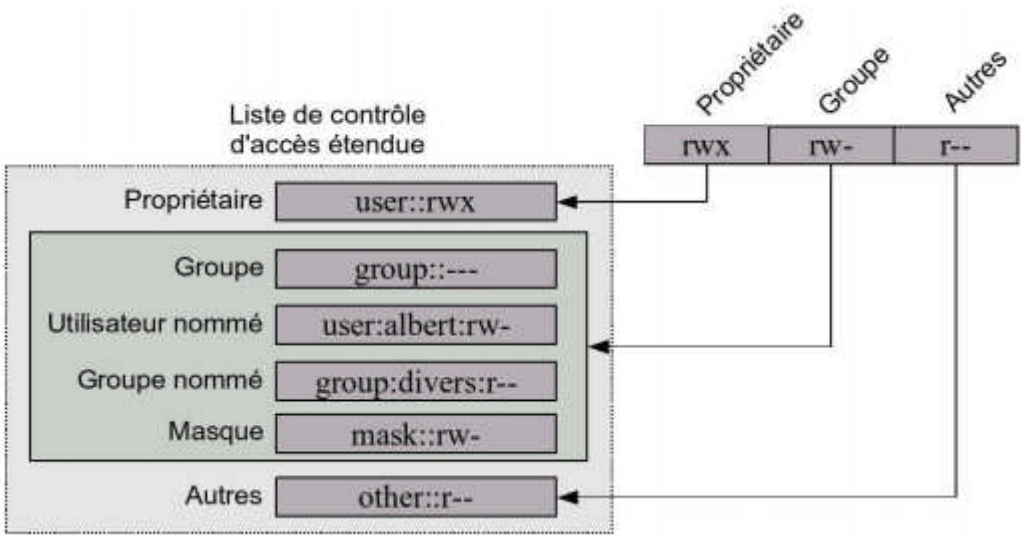

Figure 14: Liste de contrôle d'accès étendue **[artc7]**

Il n'est pas difficile de comprendre le rôle des « Utilisateurs nommés » et des « Groupes nommés » qui vont nous permettre d'attribuer individuellement des droits à, respectivement, des utilisateurs et des groupes **[artc7]**.

I.3.5. Configuration des ACL sous LINUX

Pour interdire l'accès à certains sites on peut utiliser les ACL. On peut interdire en fonction du domaine, du protocole, de l'adresse IP, du numéro de port, d'un mot, on peut aussi limiter sur une période.

La syntaxe d'une ACL est illustrée ci dessus:

acl	aclname	acltype	string [string2]
http_access		allow | deny	[!]aclname
icp_access		allow | deny	[!]aclname

II. Les services réseaux

Un service réseau est un logiciel ayant pour but de répondre à une demande provenant du réseau. Nous expliquons les principaux services dans les lignes qui suivent, pour bien comprendre leurs fonctionnements et leurs utilités.

II.1. DHCP

II.1.1. Définition

DHCP signifie Dynamic Host Configuration Protocol. Il s'agit d'un protocole de configuration dynamique d'hôte qui permet d'allouer à la demande des adresses IP aux équipements se connectant au réseau. DHCP est un protocole de la couche application utilisant IP/UDP. C'est est extension du protocole BOOTP (Bootstrap Protocol), l'enrichissant de nouvelles fonctionnalités, comme l'allocation dynamique d'adresses IP **[Th4].**

II.1.2. Fonctionnement du protocole DHCP

DHCP fonctionne sur le modèle client-serveur : un serveur qui détient la politique d'attribution des configurations IP, envoie une configuration donnée pour une durée donnée à un client donné (typiquement, une machine qui vient de démarrer). Le serveur va servir de base pour toutes les requêtes DHCP (il les reçoit et y répond), aussi doit-il avoir une configuration IP fixe. Dans un réseau, on peut donc n'avoir qu'une seule machine avec adresse IP fixe : le serveur DHCP. Le protocole DHCP s'appuie entièrement sur BOOTP : il en reprend le mécanisme de base (ordre des requêtes, mais aussi le format des messages). DHCP est une extension de BOOTP.

Quand une machine vient de démarrer, elle n'a pas de configuration réseau (même pas de configuration par défaut), et pourtant, elle doit arriver à émettre un message sur le réseau pour qu'on lui donne une vraie configuration. La technique utilisée est le broadcast : pour trouver et dialoguer avec un serveur DHCP, la machine va simplement émettre un paquet spécial, dit de broadcast, sur l'adresse IP 255.255.255.255 et sur le réseau local. Ce paquet particulier va être reçu par toutes les machines connectées au réseau (particularité du broadcast).

Lorsque le serveur DHCP reçoit ce paquet, il répond par un autre paquet de broadcast contenant toutes les informations requises pour la configuration. Si le client

accepte la configuration, il renvoie un paquet pour informer le serveur qu'il garde les paramètres, sinon, il fait une nouvelle demande **[Th4].**

II.1.3. Avantages de DHCP dans l'administration d'un réseau

- Le protocole DHCP offre une configuration de réseau TCP/IP fiable et simple, empêche les conflits d'adresses et permet de contrôler l'utilisation des adresses IP de façon centralisée**.** Ainsi, si un paramètre change au niveau du réseau, comme, par exemple l'adresse de la passerelle par défaut, il suffit de changer la valeur du paramètre au niveau du serveur DHCP, pour que toutes les stations aient une prise en compte du nouveau paramètre dès que le bail sera renouvelé. Dans le cas de l'adressage statique, il faudrait manuellement reconfigurer toutes les machines.
- Economie d'adresse : ce protocole est presque toujours utilisé par les fournisseurs d'accès Internet qui disposent d'un nombre d'adresses limité. Ainsi grâce à DHCP, seules les machines connectées en ligne ont une adresse IP.
- Les postes itinérants sont plus faciles à gérer **[w01].**

II.1.4. Configuration du dhcp sous linux

Sous linux, la configuration du DHCP consiste au minimum à configurer le fichier dhcpd.conf qui se trouve par défaut dans le répértoire : /etc/dhcp/.

Ce fichier est composé de plusieurs sections, chacune limitée par des accolades {et} :

- des paramètres globaux qui s'appliquent à tout le fichier ;
- shared-network ;
- subnet ;
- host ;
- group.

Après chaque modification de la configuration, il faut relancer le serveur avec la commande :

/etc/init.d/squid restart

II.2. DNS

II.2.1. Définition

Le système de nom de domaine (DNS) est utilisé pour faire correspondre des noms de domaine et des adresses IP afin de pouvoir localiser des hôtes sur des réseaux distants par le biais de noms, plus facilement mémorisables qu'une adresse IP.

Ce processus s'articule autour d'une relation client/serveur ou le client, nommé resolver, effectue une requête auprès d'un serveur de noms.

Lorsque l'utilisateur entre l'URL,http://www.google.com par exemple, le navigateur envoie une requête au serveur de domaine de fournisseur d'accès, qui essaie de déterminer l'adresse IP correspondante comme le montre la figure suivante **[B4].**

figure15 : résolution d'adresse avec DNS

II.2.2. Les composants d'un serveur DNS

Pour répondre à toutes ces conditions, le DNS a été organisé en trois composantes.

- **L'espace des noms et la liste des ressources :**

L'espace des noms et la liste des ressources définissent un ensemble de noms et les données qui leur sont associées, le tout étant organisé sous la forme d'un arbre. Chaque nœud et chaque feuille de cet arbre contient donc un certain nombre d'informations. Ces informations, stockées dans des structures appelées *Resource Records*, seront consultées grâce à des requêtes. Ainsi, pour accéder à une information donnée, on précisera dans la requête le nom de domaine concerné et le type d'information désirée **[w02]**.

- **Les serveurs de noms :**

Les serveurs de noms sont des programmes qui assurent la traduction des noms en adresses IP et réciproquement. Pour cela, ils possèdent des informations sur l'architecture de l'arbre et sur les données associées. Un serveur de nom peut mémoriser des informations au sujet de n'importe quelle partie de l'arbre, mais en général il contient plutôt des informations complètes sur une partie de l'arbre. Il possède également les références d'autres serveurs susceptibles de fournir des informations sur le reste de l'arbre. Lorsqu'un serveur a la connaissance complète d'une partie de l'arbre, on dit qu'il a autorité sur cette partie de l'arbre **[w02]**.

- **Les solveurs de noms :**

Les solveurs de noms (resolvers) sont des programmes qui, à la suite d'une demande de la part d'un client, obtiennent l'information recherchée auprès des serveurs de noms.

Un solveur doit donc pouvoir interroger un serveur de nom et utiliser l'information reçue pour répondre à la demande ou pour interroger un autre serveur susceptible de connaître la réponse. Le plus souvent, un solveur se présentera sous la forme d'une routine système directement accessible par les programmes utilisateurs **[w02].**

II.2.3. le rôle de DNS

Le système DNS introduit une convention de nommage hiérarchique des domaines qui commence par un domaine racine noté ".". Les domaines situés directement sous le domaine racine sont appelés domaines de premier niveau. Ils sont gérés par l'ICANN et représentent souvent la localisation géographique (fr, be, eu, ru, de ...) ou le type de service (museum, info, org, gov, mail, ...). Les domaines de second niveau sont disponibles pour les entreprises et les particuliers. Ils sont distribués et gérés par d'autres sociétés comme l'InterNIC (une filiale le l'ICANN) ou bien l'AFNIC (Association Française pour le Nommage Internet en Coopération) qui gère le domaine *fr*. Enfin une multitude de sous domaines peuvent être crée à l'intérieur d'un domaine de second niveau.

La figure suivante illustre un extrait de la hiérarchie du serveur DNS **[w015].**

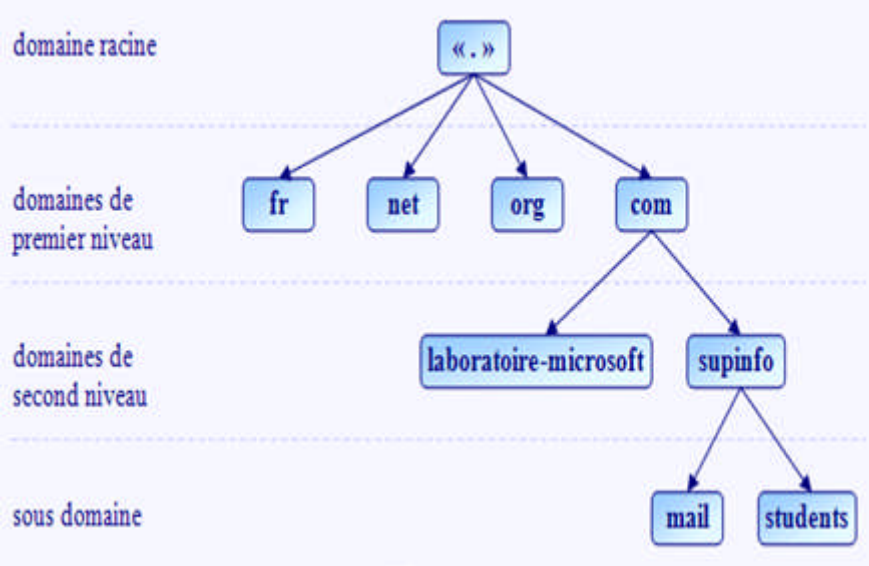

Figure 16: Un extrait de la Hiérarchie du serveur DNS **[w015]**

II.2.4. Configuration du DNS sous linux

La configuration du DNS sous linux consiste à manipuler au minimum le fichier **named.conf** qui est le fichier principal et le plus important de la configuration de ce serveur, il est crée à l'installation du package BIND, il s'agit aussi de manipuler les fichiers des bases de données crées par l'administrateur lors de la configuration du serveur, **exemple** : db.univ-bejaia.dz.

II.3. SQUID

II.3.1. Définition

Un serveur proxy est une machine intermédiaire entre un réseau local et internet. Son but principal est de permettre aux postes du LAN d'accéder à Internet par son biais. En général elle sert aussi de cache, c'est à dire de sauvegarder en mémoire les pages du web les plus utilisées, afin de gagner du temps d'accès et de réduire l'utilisation de la bande passante (proxy-cache).Comme il est illustré dans la figure suivante **[w03]**:

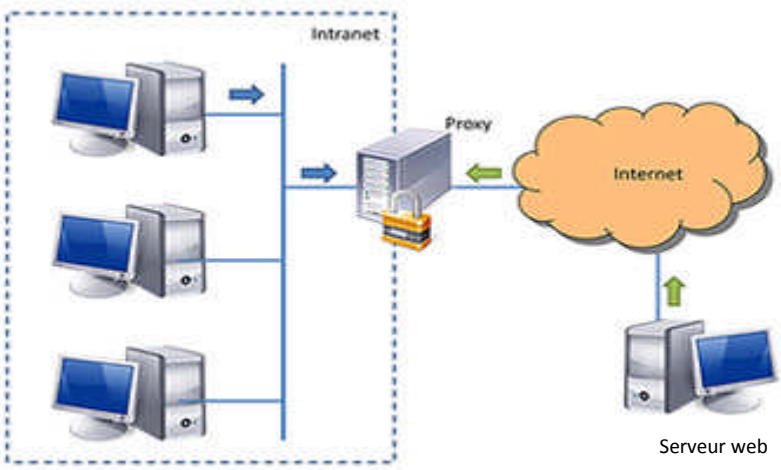

Figure17 : la position de proxy **[w04]**

II.3.2. Fonctionnement du proxy

Il s'agit d'un serveur mandaté par une application pour effectuer une requête sur Internet à sa place. Ainsi, lorsqu'un utilisateur se connecte à Internet à l'aide d'une application cliente configurée pour utiliser un serveur proxy, celle-ci va se connecter en premier lieu au serveur proxy et lui donner sa requête. Le serveur proxy va alors se connecter au serveur que l'application cliente cherche à joindre et lui transmettre la requête. Le serveur va ensuite donner sa réponse au proxy, qui va à son tour la transmettre à l'application cliente **[w05].**

Dans notre cas pratique on se focalise sur le serveur proxy Squid qui s'installe sous linux.

II.3.3. La configuration du serveur Squid

Squid comporte de nombreux paramètres, toute la configuration de Squid se trouve dans le fichier squid.conf que l'on trouve dans **/etc/squid/** ou **dans /usr/local/squid/etc.**

La plupart des options par défaut du fichier squid.conf ne sont pas à changer, on peut laisser le # pour garder les options par défaut.

II.3.4. Les services de Squid

Squid propose beaucoup de services, on site les plus importants qui sont :

Le cache : Le rôle de ce serveur est de stocker les objets demandés par les utilisateurs pour la première fois via les protocoles HTTP, FTP et Gopher. Ainsi, lors des demandes futures sur

un objet présent en cache, s'il n y a pas de mise à jour, le serveur de cache n'aura pas besoin d'aller chercher cet objet sur Internet et retournera directement celui qu'il a en mémoire. Les objets stockés peuvent être de tout type : texte, image, vidéo, …etc **[w06].**

L'authentification: Squid permet d'authentifier les clients avant qu'ils accèdent à la ressource qu'ils demandent. L'authentification marche pour les modes proxy et "httpd accelerator". Il devient alors de plus en plus avantageux d'utiliser Squid en frontale d'un serveur web car ce dernier n'aura à assumer que les rôles primordiaux de service web dynamique **[w06].**

Filtrage: Squid offre la possibilité de filtrer les requêtes des clients. Ainsi, il est possible de restreindre l'accès aux ressources en fonction de plusieurs paramètres différents. Voici une liste de paramètre pouvant intervenir dans le rejet d'une requête répondant à l'un des critères :

- L'URL contient un mot interdit ;
- L'adresse IP source/destinataire est interdite ;
- Le domaine de source/destination est interdit ou contient un mot interdit ;
- La date de la demande ;
- Le port de destination ;
- Le protocole utilisé. Peut permettre de bloquer les transferts FTP par exemple ;
- Le type du navigateur utilisé **[w06].**

II.4. Apache

II.4.1. Présentation d'apache

Avec environ 70% de part de marché, apache est le plus important dispositif de publication de contenu sur Internet. Cet enivrement s'explique sûrement par la conception même d'apache. En effet, apache est un serveur modulaire. C'est-à-dire qu'il est possible de rajouter des modules en fonction des besoins **[B4].**

II.4.2. Configuration d'apache

La configuration d'apache a évolué au fil du temps. Dans sa version 1.3, toute sa configuration se faisait à l'aide d'un fichier httpd.conf. Depuis la version 2.0, il existe plusieurs fichiers de configuration, chacun servant à configurer une fonction particulière d'apache. Par exemple, un fichier spécifique doit être utilisé pour configurer un hôte virtuel.

Par la suite, chaque fichier est inclus dans le fichier httpd.conf. Le fait de séparer les différents points de configuration permet d'en faciliter la lecture et la compréhension. Principalement, on distingue deux fichiers de configuration : un pour configurer le fonctionnement du serveur et un pour définir ce qui est envoyé sur internet **[B4].**

Conclusion

Bien que la configuration des services réseau et le contrôle d'accès suffit pour administrer un réseau informatique mais il est important de prendre en considération la notion du temps, surtout avec l'évolution des technologies et leur influence sur les entreprises, en outre on a envisagé de créer une application dynamique pour la gestion des services réseau.

Chapitre III

Analyse et conception

Introduction

Cette partie est consacrée à l'analyse et à la conception de notre projet, en se basant sur le langage de modélisation unifié UML, guidée par une démarche (UP). Il s'agit d'une étape cruciale dans la réalisation d'une application donnée et le futur d'un logiciel dépend de cette phase.

Le présent chapitre va nous donner un aperçu global de l'application.

I. Présentation de l'entreprise

I.1. Historique

L'université de Bejaia est une organisation éducative crée à la faveur de la décentralisation initiée au début des années 80 pour désengorger les pôles universitaires d'Alger, Constantine, Tizi-Ouzou et de Sétif. Bien que née tardivement (1983) par rapport à son passé scientifique et culturel de renom, l'Université de Bejaia, établissement public de formation supérieure sous tutelle du Ministère de l'Enseignement Supérieur et de la Recherche Scientifique, a pu s'agrandir et passer d'un effectif de 205 étudiants encadrés par 40 enseignants à son ouverture à un effectif de 22 792 étudiants et 698 enseignants en 2006.

Cet essor n'a pu se réaliser qu'avec une diversification judicieuse de ses filières et options. L'Université de Bejaia regroupe quatre Facultés (Faculté des Sciences exactes, Faculté de technologie, Faculté de science de la nature et de la vie, faculté de droit, faculté des lettres et des sciences humaines, faculté des sciences économiques, sciences de gestion et sciences commerciales et faculté de médecine). Elle se déploie sur deux sites principaux (Targa Ouzemmour et Aboudaou). L'Université de Bejaia dispose d'une bibliothèque centrale pluridisciplinaire, d'une annexe opérationnelle depuis 2002 et d'une bibliothèque au campus d'Aboudaou ouverte en 2003. Elle couvre plus de 35 spécialités avec un fond documentaire riche et dépassant en ouvrage 146 000 volumes, 2500 titres de thèses et mémoires, 300 titres de revue dont 79 en cour d'abonnement et environ 400 CD ROM. La diversification de

filière de formation, la valorisation de la recherche et le transfert de compétences lui ont permis d'être parmi les universités les plus performantes du pays.

Devant l'importance grandissante de l'informatique dans les domaines de la vie courante et sa large utilisation dans toutes les filières (technologie, économie, médecine, etc.), l'ouverture de cette spécialité est devenue indispensable. La volonté et le dynamisme des différents responsables ont conduit à l'ouverture d'une Ecole Doctorale en Informatique option Réseaux et Systèmes Distribués : c'est la première du genre dans le pays. L'université de Bejaia, à l'instar des autres établissements de l'enseignement supérieur, a mis en place le nouveau dispositif d'enseignement LMD des la rentrés 2004/2005, avec comme première étape une proposition de 33 licences dont 19 professionnelles et 14 académiques.

I.2. Organigramme de l'université de Bejaia

L'Université de Bejaia comme l'illustre la figure ci-dessous dispose de quatre services principaux : Le rectorat, le secrétaire général, La bibliothèque centrale de l'université et les services communs de l'université. Chaque service dispose d'un ensemble de sous services.

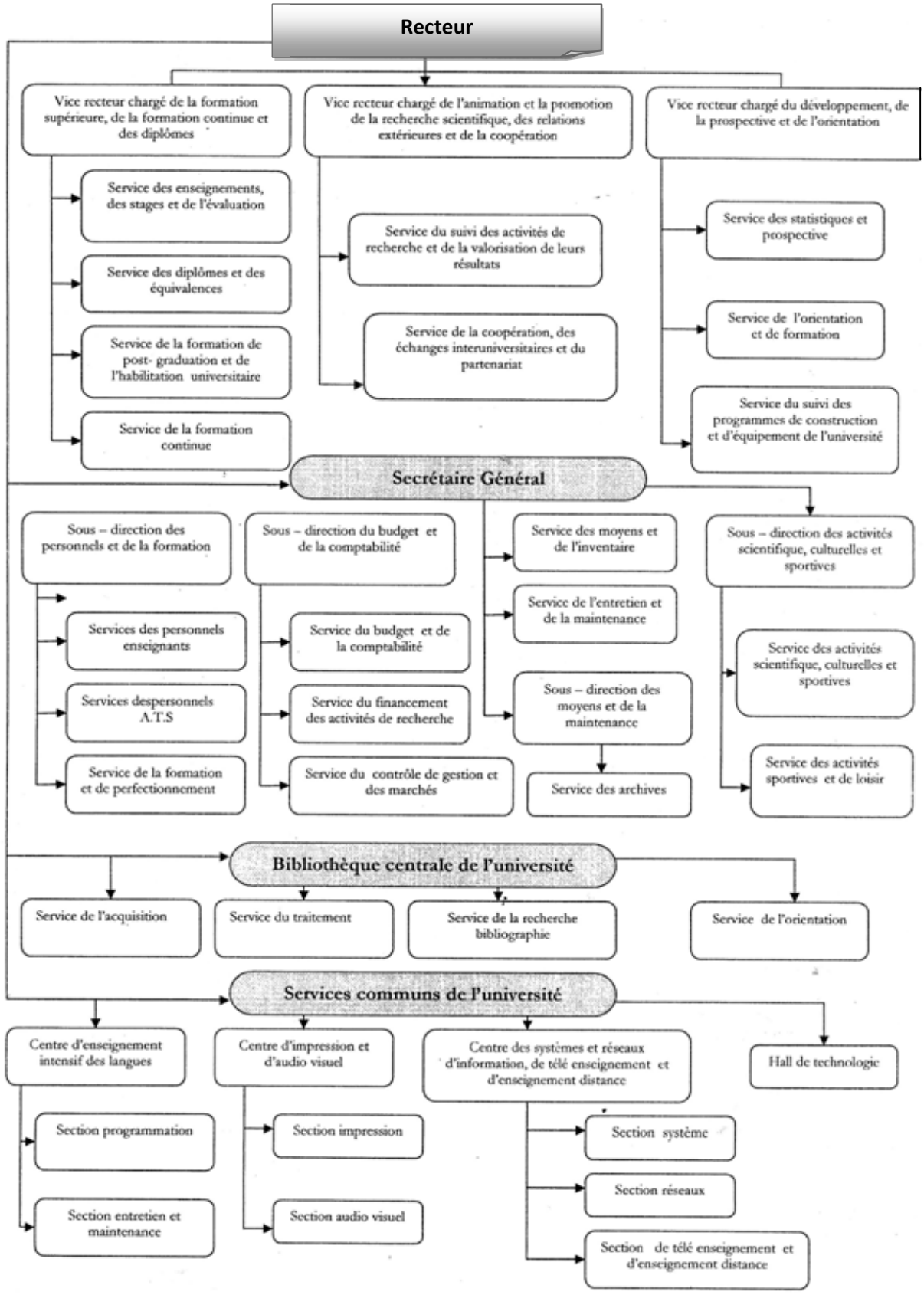

Figure 18 : Organigramme de l'université de Bejaia

I.3. Le réseau de l'université de Bejaia

Le réseau de l'université de Bejaïa dispose de trois serveurs : un serveur proxy, un serveur web et le serveur de messagerie représentant une zone démilitarisée reliée à un pare-feu qui permet l'accès à l'internet en passant par un routeur, le firewall est aussi lié à un Switch fédérateur, le tout compose la première zone de l'université de Bejaïa.

Le Switch fédérateur relie les trois zones restantes à la première zone et au centre d'ABOUDAOU, la figure ci-dessus illustre la schématisation du réseau de l'université de Bejaia.

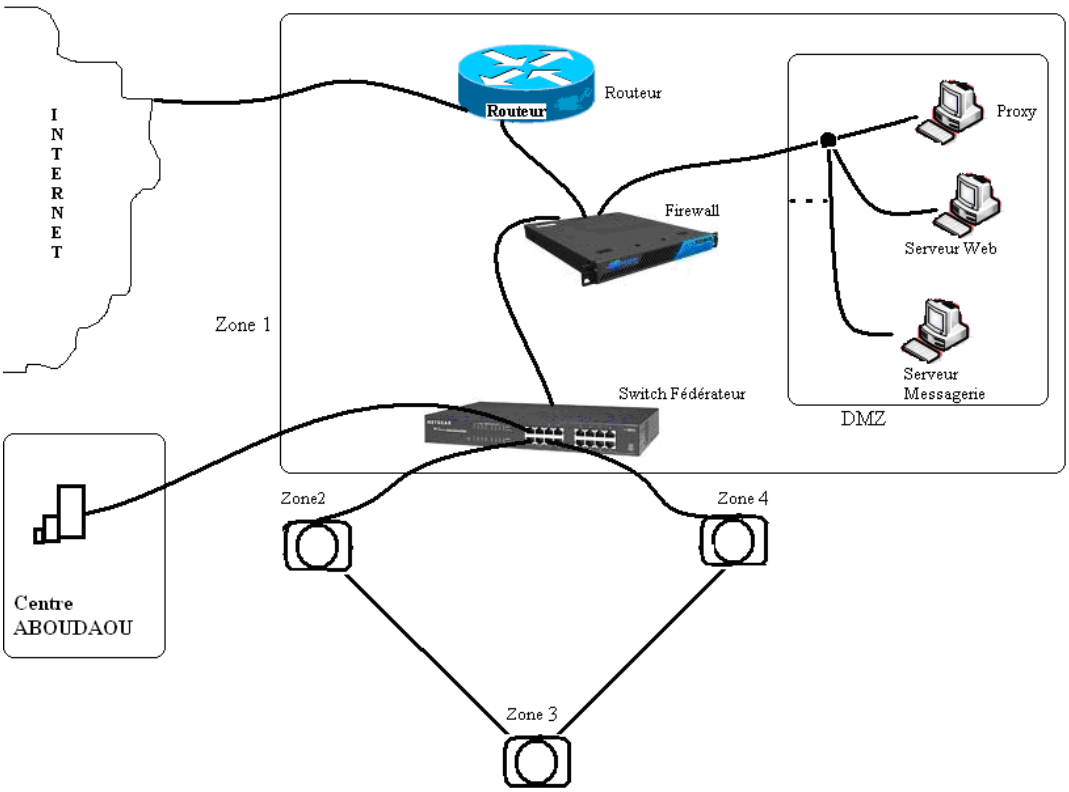

Figure 19 : Topologie intranet du réseau de l'université de Bejaia

II. Cotexte et motivation du projet

II.1. Contexte

Notre application consiste à créer une interface Web, accessible par mot de passe, pour permettre à un administrateur de gérer des serveurs qui sont dans notre application des services réseau installés sur une machine. Notre travail se focalisera sur la tache d'administration réseau : démarrer, arrêter ou redémarrer les serveurs et modifier les fichiers de configuration, qui sont des actions que l'administrateur réseau devra effectuer depuis une machine distante.

II.2. Critique de l'existant

La gestion actuelle du réseau de l'université de Bejaïa exige le déplacement de l'administrateur jusqu'aux machines sur lesquelles sont installés les services réseau et qui sont dotées du système d'exploitation Linux, ce qui n'est pas pratique en terme de temps.

L'administrateur gère tout serveur par l'accès au fichier de configuration concerné et en effectuant des modifications sur les lignes et cela n'est pas évident pour des fichiers volumineux.

II.3. Analyse et définition des besoins

La conception et le développement de cette application web a été faite suite à une collecte d'information au niveau du centre de calcul, qui contient les différents serveurs en discutant avec l'administrateur réseau afin de concevoir une application qui répond à ses besoins.

L'étude présentée dans ce chapitre est donc le fruit de ces discussions et de l'analyse de besoins.

II.3.1. Les besoins fonctionnelles

- L'administrateur peut y accéder à l'interface qui lui est destinée après l'authentification.
- L'administrateur peut à tout moment démarrer, arrêter ou redémarrer les serveurs.

- L'administrateur peut y accéder à distance aux fichiers de configuration des serveurs et effectuer des modifications sur ces fichiers.
- L'administrateur peut à tout moment faire un aperçue du fichier de configuration.

III. Modélisation du projet

III.1. Unified Modeling Language (UML)

III.1.1. Définition

La notation UML (Unified Modeling Language) constitue une étape importante dans la convergence des notations utilisées dans le domaine de l'analyse et la conception objet puisqu'elle représente une synthèse des méthodes les plus utilisées **[Th1].**

III.1.2. Démarche de la modélisation UML

A l'heure actuelle UML est dans sa version 2.0 et il comporte 13 diagrammes. Répartis en deux types de vue : vue statique et vue dynamique (comportementale). Comme le montre la figure suivante **[Th2]**:

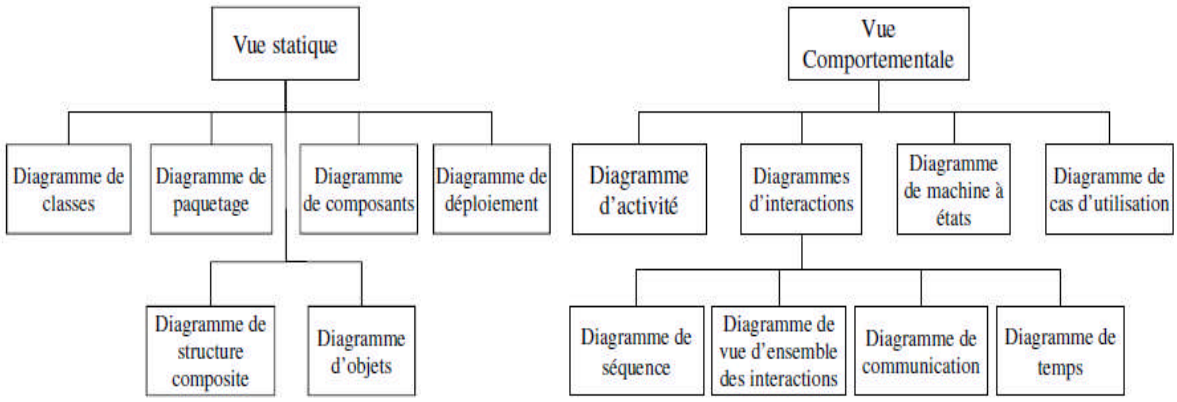

Figure 20: Les 13 diagrammes UML

Afin de modéliser notre application Web nous allons utiliser quatre diagrammes pour la spécification des besoins.

a. Diagrammes de cas d'utilisations

L'objectif des cas d'utilisation est l'expression des besoins en termes de services que doit assurer le système. Les diagrammes de cas d'utilisation dans UML définissent deux concepts principaux : les acteurs, et les cas d'utilisation. Un acteur est une entité extérieure du système qui peut initier l'un de ses cas d'utilisation. Un cas d'utilisation est une fonctionnalité offerte par le système. **La Figure 20** montre un diagramme de cas d'utilisation pour un exemple d'une application bancaire **[Th1].**

Les diagrammes de cas d'utilisation sont importants pour visualiser, spécifier et documenter le comportement d'un élément. Ils rendent les systèmes, les sous – systèmes et les classes plus facile à aborder et plus compréhensibles en présentant une vus externe de la façon dont ces éléments peuvent êtres utilisés en contexte **[B8].**

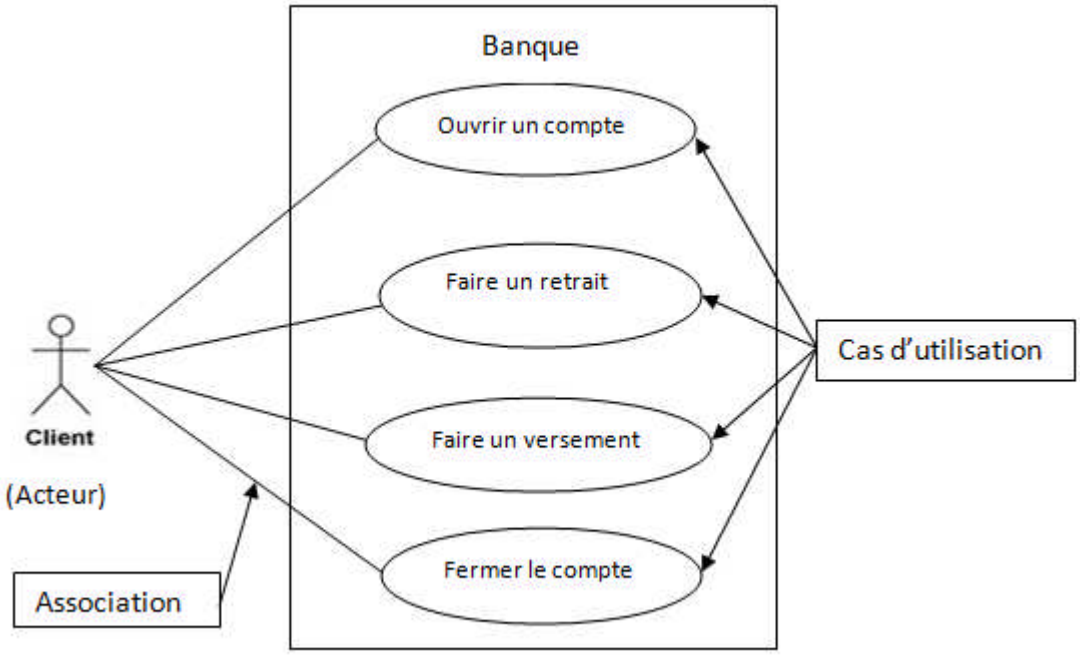

Figure 21 : Exemple de diagramme de cas d'utilisation

Les diagrammes de cas d'utilisation permettent de définir deux types de relations entre les cas d'utilisation : l'inclusion et l'extension. Un cas d'utilisation peut être compris dans un autre cas d'utilisation ; ceci signifie que le service spécifié par le deuxième est compris dans le service du premier. Un cas d'utilisation peut étendre un cas d'utilisation

père ; ceci signifie que le service du premier est une spécialisation du service du deuxième. Les relations entre les cas d'utilisation sont notées comme des flèches de dépendances avec les mots-clés <<include>> pour l'inclusion et <<extend>> pour l'extension **[Th1].**

b. Diagrammes de séquences

Les diagrammes de séquences sont des diagrammes d'interaction qui mettent en évidence l'ordre chronologique des messages.ils représentent un ensemble d'objets ainsi que les messages envoyés et reçus par ces objets. Les objets sont en général des instances de classes désignées aux anonymes, mais ils peuvent également représenter des instances d'autres éléments structurels, comme des collaborations, des composants et des nœuds. On utilise les diagrammes de séquence pour illustrer la vue dynamique d'un système **[B8].**

L'ordre d'envoi d'un message est déterminé par sa position sur l'axe vertical du diagramme; le temps s'écoule "de haut en bas" de cet axe.

La figure 22 illustre l'affichage du résultat d'une recherche bibliothèque en diagramme de séquence.

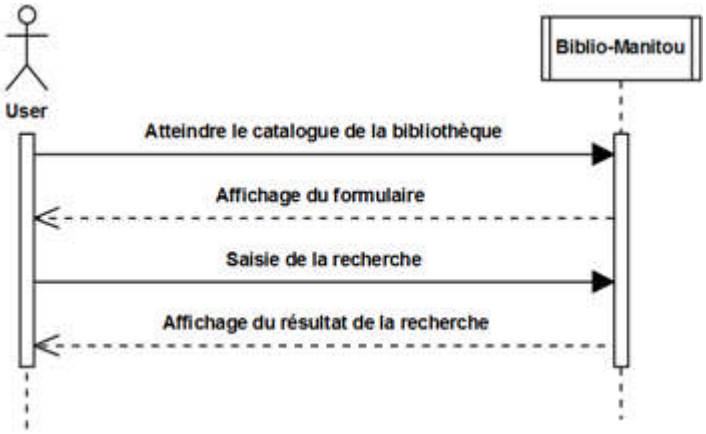

Figure 22: Exemple diagramme de séquences

c. Diagrammes de classes

Les diagrammes de classes sont les diagrammes les plus courants dans la modélisation des systèmes orientés objet. Il permet de décrire statiquement les classes présentes dans un système et les relations entre ces classes **[B8].**

Tout d'abord, une *classe* contient plusieurs informations : un nom, des attributs et des méthodes. La figure 23 montre la notation UML pour une classe LIVRE qui a un attribut titre et trois méthodes getTitre, emprunter et retour. Il existe trois types de *visibilité* pour les classes, les attributs et les méthodes : privé, protégé, public. La visibilité privée restreint l'accès à cet élément aux éléments de la même classe, un élément protégé est accessible aux descendants de la classe et aux éléments d'un même paquetage, et un élément public est accessible à tous les autres éléments du système. La notation UML pour la visibilité est la suivante : **-** pour privé, # pour partagé et + pour public. Enfin, une classe peut être abstraite ou une simple interface. Une *classe abstraite* peut contenir des attributs et des méthodes, mais le corps de certaines méthodes est vide et elles doivent être définies dans des classes filles **[Th3].**

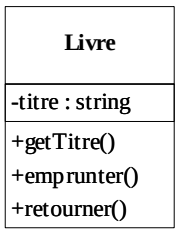

Figure 23 : Exemple de notation UML d'une classe « livre »

Les classes peuvent être regroupées dans des paquetages. Un paquetage UML (Package) est un espace de nommage. La Figure 22 illustre un paquetage UML appelé Bank qui regroupe les deux classes Account et Customer **[Th1].**

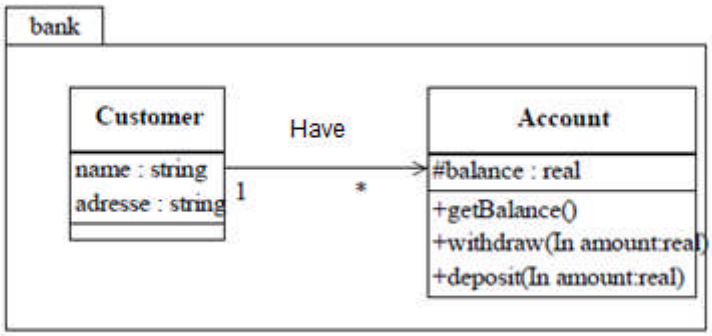

Figure 24 : Extrait d'un diagramme de classes d'une application bancaire **[Th1]**

d. Diagrammes de déploiement

Un diagramme de déploiement décrit la disposition physique des ressources matérielles qui composent le système et montre la répartition des composants sur ces matériels comme l'illustre la figure suivante. Chaque ressource étant matérialisée par un nœud (exemple : utilisateur, web browser (HTTP), Serveur web et le serveur de base de données), le diagramme de déploiement précise comment les composants sont répartis sur les nœuds et quelles sont les connexions entre les composants ou les nœuds **[B6].**

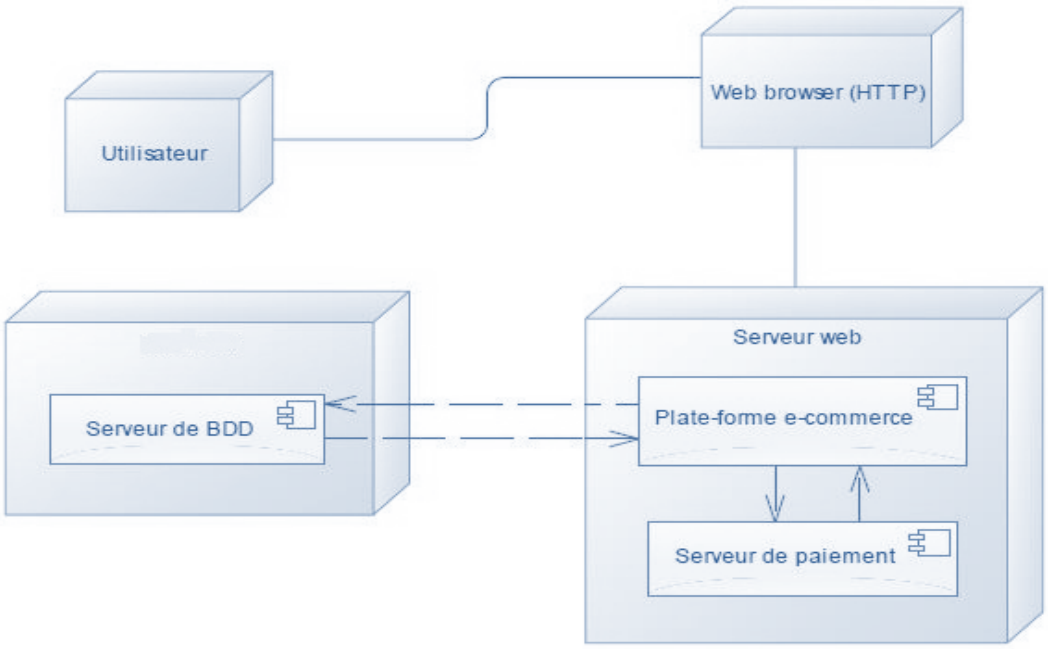

Figure 25 : Exemple d'un diagramme de déploiement

IV. UP - LE PROCESSUS UNIFIE

IV.1. Définition

Le processus unifié est un processus de développement logiciel, c'est à dire qu'il regroupe les activités à mener pour transformer les besoins d'un utilisateur en système logiciel comme le montre la figure suivante.

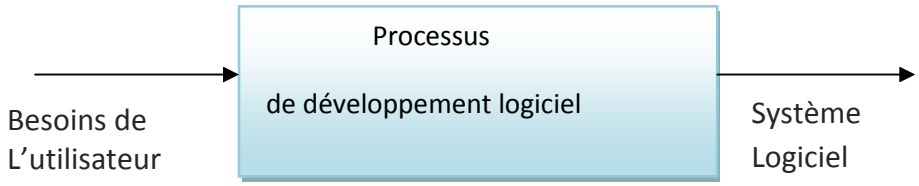

Figure 26 : Processus de développement logiciel

Néanmoins, les trais véritablement Distinctifs du processus unifié tiennent en trois expressions clés : itératif et incrémental, centré sur l'architecture et piloté par les cas d'utilisation **[B7].**

1. itératif et incrémental

Vu que les projets à réaliser sont de plus en plus complexes et grands, l'idée est de découper le travail en mini projets. Chacun d'entre eux représente une itération qui donne lieu à un incrément. Les itérations désignent des étapes de l'enchaînement d'activités, tandis que les incréments correspondent à des stades de développement du produit. **[B7]**

2. centré sur l'architecture

L'architecture logicielle représente les aspects statiques et dynamiques du système. Elle émerge des besoins de l'entreprise, tels qu'ils sont exprimés par les utilisateurs et reflétés par les cas d'utilisation. L'architecture propose une vue d'ensemble de la conception faisant ressortir les caractéristiques essentielles en laissant de côté les détails secondaires. Il faut noter que tout produit est à la fois forme et fonction. L'une ou l'autre isolément ne

saurait suffire. Les cas d'utilisation et l'architecture doivent s'équilibrer pour créer un produit réussi **[B7].**

3. piloté par les cas d'utilisation

Un cas d'utilisation représente une fonctionnalité qui satisfait un besoin d'un utilisateur.

Le processus suit une voie spécifique, en procédant par une série d'enchaînement d'activités, dérivées d'un cas d'utilisation. Un cas d'utilisation est analysé, conçu, implémenté et enfin testé **[B7].**

IV.2. Cycle de vie du processus unifié

Le processus unifié se déroule en quatre phases, incubation, élaboration, construction et transition. Chaque phase répète un nombre de fois une série d'itérations. Et chaque itération est composée de cinq activités : capture des besoins, analyse, conception, implémentation et test comme la montre la figure suivante **[B7]**:

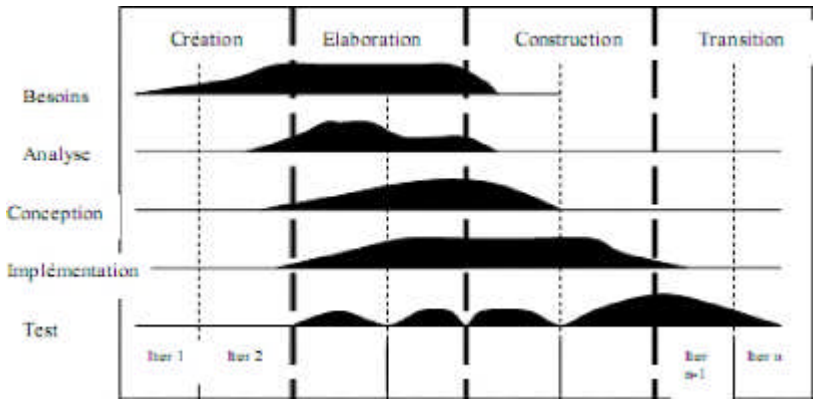

Figure 27: cycle de vie du processus unifié

- **Création (Incubation)**

C'est la première phase du processus unifié. Il s'agit de délimiter la portée du système, c'est-à-dire tracer ce qui doit figurer à l'intérieur du système et ce qui doit rester à l'extérieur, identifier les acteurs, lever les ambiguïtés sur les besoins et les exigences

nécessaires dans cette phase. Il s'agit aussi d'établir une architecture candidate, c'est-à-dire que pour une première phase, on doit essayer de construire une architecture capable de fonctionner. Dans cette phase, il faut identifier les risques critiques susceptibles de faire obstacles au bon déroulement du projet **[B7]**.

- **Elaboration**

C'est la deuxième phase du processus. Après avoir compris le système, dégagé les fonctionnalités initiales, précisé les risques critiques, le travail à accomplir maintenant consiste à stabiliser l'architecture du système. Il s'agit alors de raffiner le modèle initial de cas d'utilisation, voire capturer de nouveaux besoins, analyser et concevoir la majorité des cas d'utilisation formulés, et si possible implémenter et tester les cas d'utilisation initiaux **[B7].**

- **Construction**

Dans cette phase, il faut essayer de capturer tous les besoins restants car il n'est pratiquement plus possible de le faire dans la prochaine phase. Ensuite, continuer l'analyse, la conception et surtout l'implémentation de tous les cas d'utilisation. A la fin de cette phase, les développeurs doivent fournir une version exécutable du système **[B7].**

- **Transition**

C'est la phase qui finalise le produit. Il s'agit au cours de cette phase de vérifier si le système offre véritablement les services exigés par les utilisateurs, détecter les défaillances, combler les manques dans la documentation du logiciel et adapter le produit à l'environnement (mise en place et installation) **[B7].**

- **Expression de besoins**

Recenser les besoins fonctionnels et non fonctionnels du système, le diagramme UML de cas d'utilisation est utilisé pour cette phase **[w07].**

- **Analyse**

Détaille les besoins en spécifications détaillées, une ébauche de la conception **[w07].**

- **Conception**

Décide comment sera construit le système durant l'implémentation, définition des sous-systèmes et composants et création d'abstractions de la solution **[w07].**

- **Implémentation**

Traduire les résultats de la conception en un système opérationnel Livrables **[w07].**

- **Tests**

Vérification et validation des résultats de l'implémentation **[w07].**

V. Les phases de conception suivant UP

V.1. Phase de création

Cette phase nous permet de spécifier et d'analyser les besoins issus des cas d'utilisations.

V.1.1. Identification des acteurs

L'acteur principal de notre application est l'administrateur qui peut :

- Modifier les fichiers de configuration.
- Démarrer, redémarrer et arrêter un serveur donné.

V.1.2. Les diagrammes des cas d'utilisations

Dans ce qui suit, nous verrons les diagrammes de cas d'utilisation général et détaillé pour l'administrateur.

- **Cas général**

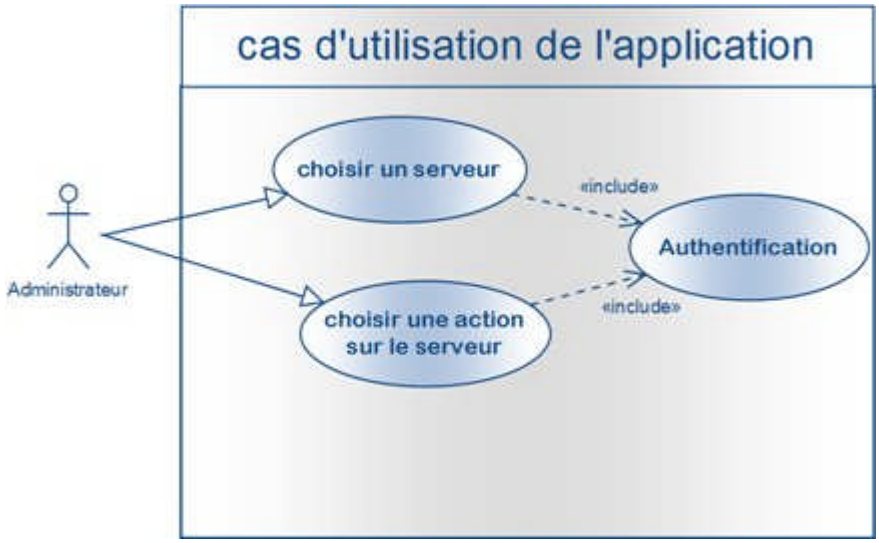

Figure 28: cas d'utilisation général

Description : Après l'authentification, l'administrateur peut effectuer deux taches principales qui sont :

1) Le choix d'un serveur à configurer ;
2) Le choix d'une action sur un serveur.

- **Cas choix d'un serveur**

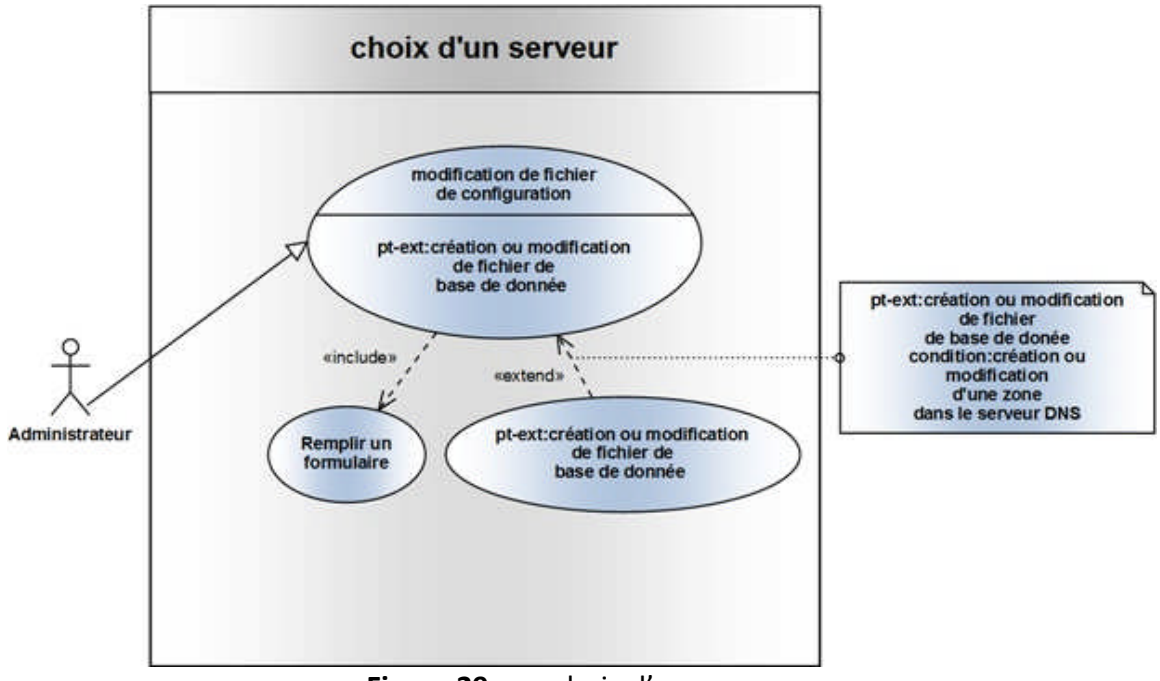

Figure 29: cas choix d'un serveur

- **Cas choix d'une action**

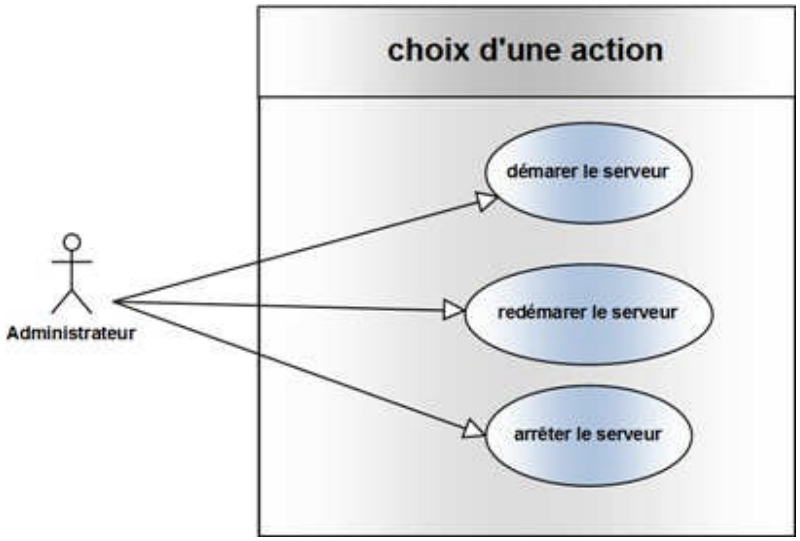

Figure 30 : cas choix d'une action

V.2. Phase d'élaboration

Dans cette phase nous allons détailler la spécification des besoins, et cela en décrivant les différentes séquences et scénarios de l'administrateur, le diagramme de classe ainsi que l'architecture de l'application.

V.2.1. Diagrammes de séquences et scénarios

- **Séquence «Modification de fichier de configuration »**

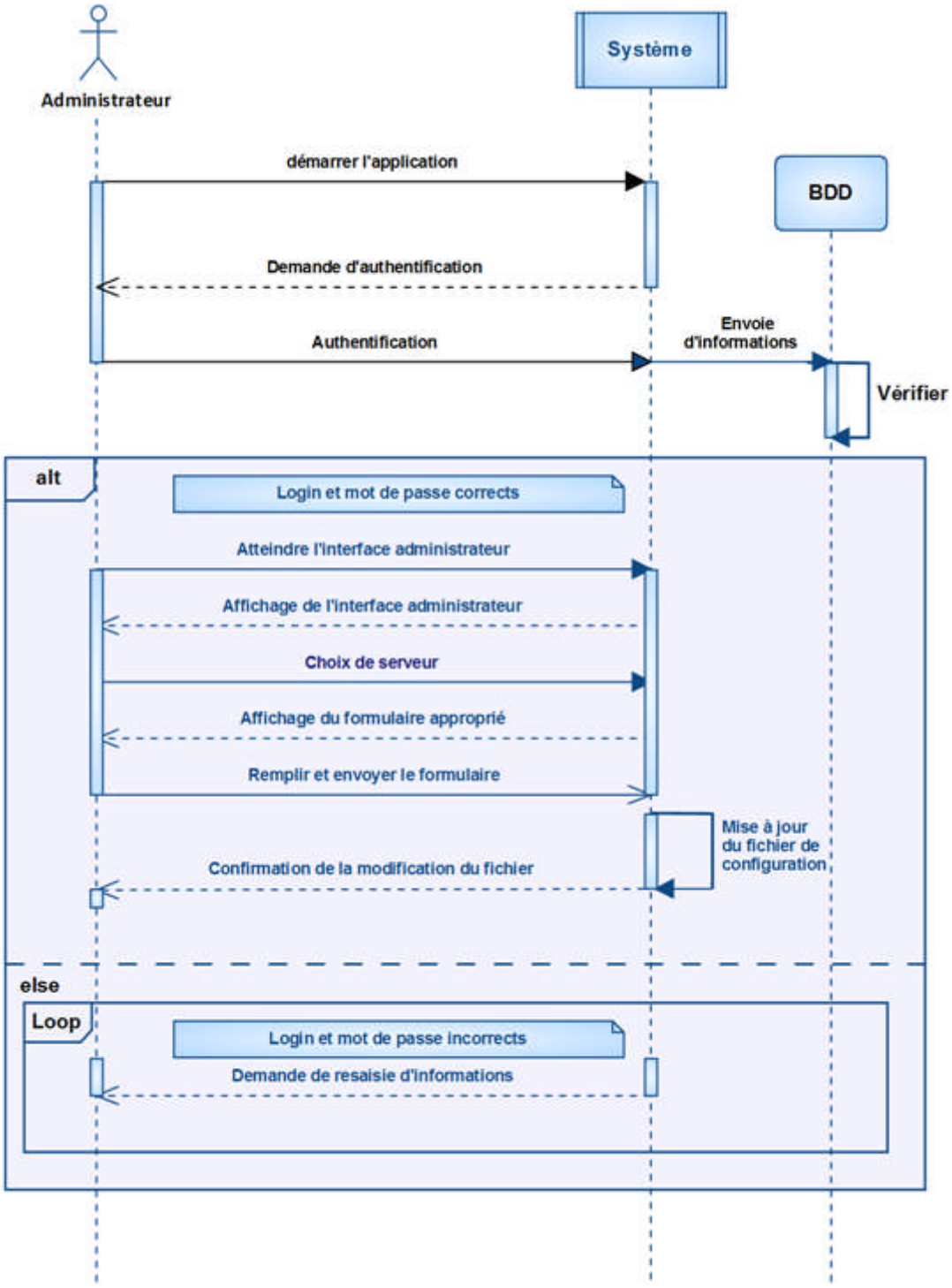

Figure 31 : Diagramme de séquence Modification de fichier de configuration

Description : Dans cette séquence l'administrateur à la possibilité de modifier un fichier de configuration d'un serveur donné après le scénario suivant:

- ✓ Atteindre l'interface administrateur ;
- ✓ Affichage de page d'authentification par le système d'application web ;
- ✓ L'utilisateur remplit et envoie le formulaire et attend la réponse ;
- ✓ Le système envoie les informations au serveur de base de données ;
- ✓ Les informations saisies par l'administrateur sont vérifiées dans la base de données : Si les informations saisies sont corrects, le système affiche l'interface administrateur sinon demande-la ressaisie d'informations ;
- ✓ Le système affiche le formulaire de modification approprié au serveur ;
- ✓ L'administrateur rempli et envoie le formulaire ;
- ✓ Le système effectue les mises à jour sur le fichier de configuration et envoie la confirmation à l'administrateur.

- **Séquence de « choix d'une action sur le serveur »**

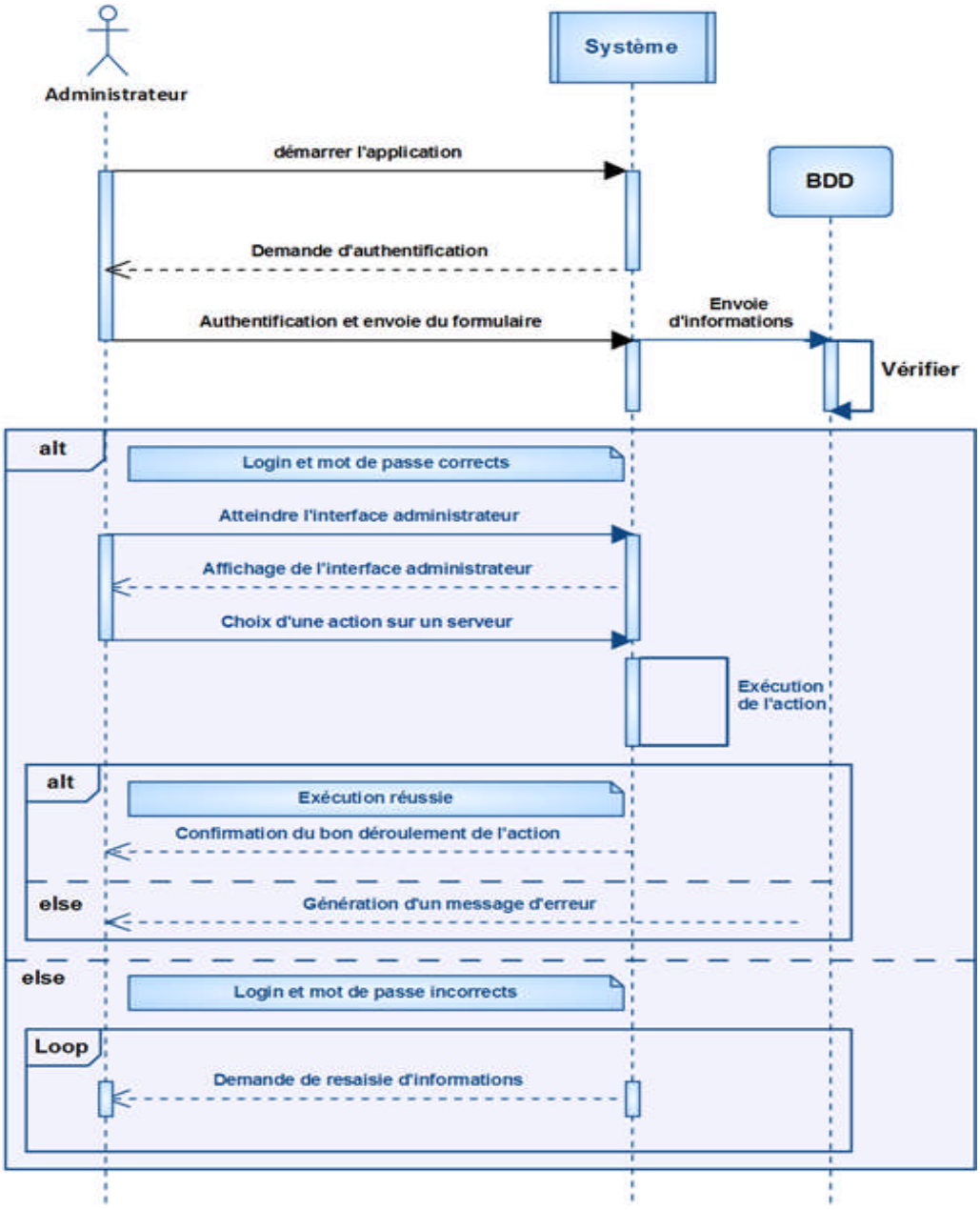

Figure 32 : Diagramme de séquence de choix d'une action

Description : Dans cette séquence l'administrateur à la possibilité de choisir une action (Démarrer, arrêter ou redémarrer) que le système va exécuter sur un serveur donné après le scénario suivant:

- ✓ L'administrateur atteint l'interface administrateur après la bonne authentification ;
- ✓ L'administrateur effectue le choix de l'action à exécuter sur un serveur donné ;
- ✓ Le système exécute l'action choisie ;

- ✓ Le système confirme le bon déroulement de l'action sinon génère un message d'erreur.

V.2.2. Diagramme de classe

Dans la figure suivante, on a décrit statiquement les classes présentes dans notre système et les relations entre ces classes.

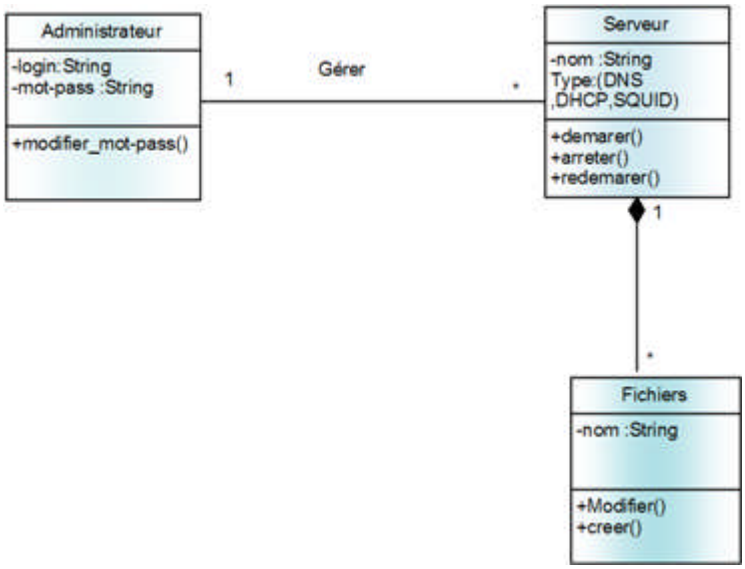

Figure 33 : Diagramme de classes

V.2.3. Architecture de l'application

Pour l'implémentation de cette application, on s'est basés sur l'architecture trois tiers, donc le déroulement des taches s'effectue à travers le client, le serveur d'application et le serveur de base de données.

V.3. Phase de construction

Dans cette phase on a essayé de capturer tous les besoins sous forme de cas d'utilisations que nous avons mis au point pour cette application.

Les objectifs principaux de l'implémentation sont de planifier les intégrations des composants pour chaque itération comme c'est illustré dans la figure 34. A la fin de cette phase on a fourni une version exécutable du système.

V.3.1. Diagramme de déploiement

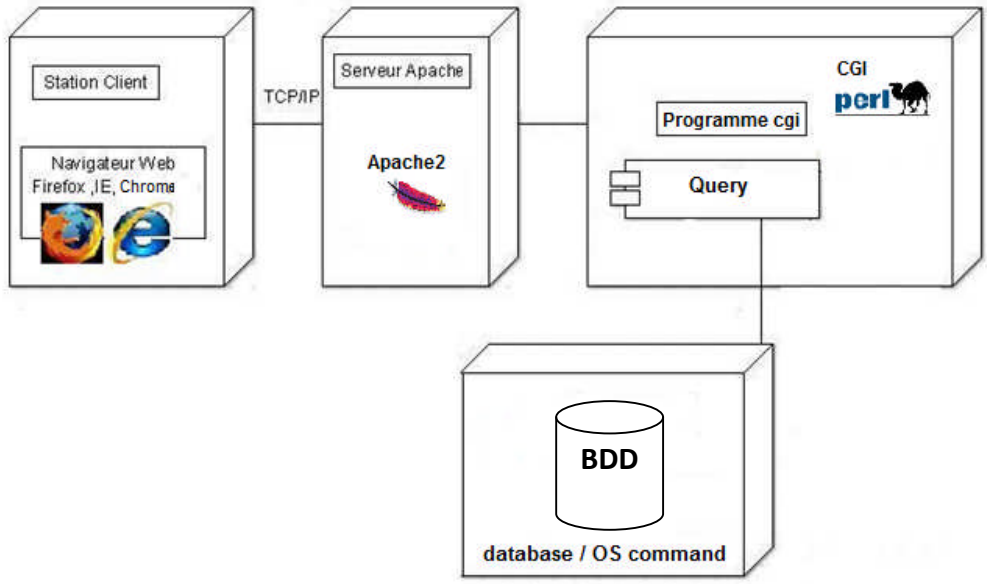

Figure 34 : diagramme de déploiement

V.3.2. Quelques détails de l'implémentation

Dans ce qui suit nous allons présenter quelques exemples de codes utilisés.

- **Utilisation de bases de données**

- **Choix de la base**

La gestion de la base de données se fait à travers les scripts CGI et SQL, grâce aux commandes de requêtes illustrées çi dessus :

Illustration

```
$var= $info->param('textfield'); // cette instruction nous permet récupérer le contenue du paramètre « textfield » via la requète $info.
```

- **Dynamisme du contenu des pages**

Une des particularités de notre application est que les aperçus des fichiers de configuration sont crées à partir des fichiers se trouvant dans le répertoire approprié.

Illustration

```
print "Content-type:text/html\n\n";
print "<html><head><title>Recopie d'un fichier</title></head>";
print "<body><p>Aper&ccedil;u du fichier squid.conf</p>";
$file="/etc/squid/squid.conf";
open(INPUT,"$file") or die "impossible d'ouviri le fichier";
while (<INPUT>) {
print;
print "<br>";
}
close INPUT;
print "<hr></body></html>";
```

- **La sécurité**

Pour contrôler étroitement notre serveur, on a interdit l'utilisation des fichiers **.htaccess** qui permettent de passer outre les fonctionnalités de sécurité qu'on a configurée. Et cela en ajoutant dans le fichier de configuration du serveur les lignes suivantes :

<Directory />
AllowOverride None
</Directory>

Ceci interdit l'utilisation des fichiers .htaccess dans tous les répertoires, sauf ceux pour lesquels c'est explicitement autorisé.

On a aussi pensé à utiliser La protection des pages par un mot de passe se fait de façon simple par ajout de la directive ci-dessous dans le fichier access.conf

<Files /répertoire/page.html>
AuthName Libellé affiché dans la fenêtre protégée
AuthType Basic
AuthUserFile /etc/users.http

require valid-user

</Files>

- /répertoire/page.html : est la page protégée
- AuthName : est le libellé qui est affiché dans la fenêtre ainsi protégée
- AuthType est en général positionné avec la valeur basic
- AuthUserFile est le fichier contenant le nom des utilisateurs qui ont le droit d'accéder à la page ainsi que le mot de passe associé
- require est une directive qui peut prendre la valeur valid-user

Les fichiers htaccess peuvent utiliser le fichier htpasswd comme base de secret, ce fichier contient les utilisateurs et les mots de passes cryptés qui peuvent se connecter à l'application web.

La fabrication d'un mot de passe se fait par le programme htpasswd qui prend la syntaxe suivante :

htpasswd -c passwordfile username

- Avec l'option -c le fichier est créé ; sans l'option -c le fichier est complété par la nouvelle entrée.
- passwordfile est le nom du fichier contenant les mots de passe
- username est le nom de l'utilisateur

On a restreint l'accès à notre application par l'authentification qui permet la vérification des données pour chaque demande d'accès, comme illustré ci-dessous :

Illustration

```
@user_parameters=split(/ /,$ligne);
for (@user_parameters)
{
if ((@user_parameters[0] eq $entre)and (@user_parameters[1] eq $entree)){
open(INPUT,"/cgper.html")or die "impossible d'ouviri le fichier";
while (<INPUT>) {
print;
```

```
print "<br>";
}close INPUT;
 }else
{
print "erreur d'authentification";
}}}
```

V.4. Phase de transition

Dans cette phase on a effectué de différents tests sur notre application afin de détecter les anomalies et de les corriger, comme les tests de compatibilité avec les systèmes d'exploitation (Linux,Windows XP,2000,…etc) et étant notre application programmé en scripts CGI, les tests ont été positifs.

Vu la portabilité de notre application, elle assure son fonctionnement avec les différents navigateurs (Mozilla Firefox, Google Chrome,… etc)

V.4.1. Diagrammes de GANTT

Le diagramme de GANTT est un outil permettant de modéliser la planification de tâches nécessaires à la réalisation d'un projet. Il s'agit d'un outil inventé en 1917 par Henry L. GANTT **[W013]**. (Voir le tableau ci-dessous)

Mois / Tâches	Mars				Avril				Mai				Juin			
	S1	S2	S3	S4	S1	S2	S3	S4	S1	S2	S3	S4	S1	S2	S3	S4
Réception du projet																
Rencontre avec le commanditaire et Définition des besoins																

Analyse de l'existant, collecte d'information et le contenu des pages.																
Réalisation de la maquette et remplissage des pages statiques																
Validation de la maquette et rédaction du cahier de charges																
Modélisation UML																
Programmation dynamiques																
Rédaction de la version finale du mémoire																

Tableau3 : Diagramme de Gantt

VI. Les contraintes

Pendant l'élaboration de notre application on a fait fasse à de différentes contraintes qui sont :

- **Contrainte de temps**

L'application doit être finalisée avant le dépôt du mémoire qui date du 19/06/2012 ce qui nous donne à peine tois mois pour faire la conception et la réalisation de notre application.

- **Contraintes ergonomiques**

La structure de notre application doit permettre à l'administrateur d'accéder aux fichiers de configuration en un minimum de clics.

- **Contraintes pédagogiques**

Vue les fonctionnalités offertes par cette application, il a fallu s'auto former dans les outils nécessaires pour le développement tel que le langage perl et les scripts CGI.

- **Contraintes fonctionnelles**

Pour que l'administrateur puisse gérer dynamiquement les serveurs, effectuer des modifications sur les fichiers de configuration et éxécuter des actions sur les serveurs, pour celà on devait créer une plate-forme qui doit etre accessible par n'importe quelle personne pouvant se charger de la gestion du réseau.

Conclusion

Dans ce chapitre, nous avons présenté l'analyse et la conception de notre application en utilisant le langage de modélisation UML guidée par une démarche (UP).

Nous avons abordé cette démarche par la définition des besoins de notre application. Puis nous avons déterminé les acteurs. Ensuite nous avons présenté graphiquement les diagrammes de cas d'utilisation, de séquence et le diagramme de classes globales.

Dans le chapitre suivant, nous présentons la réalisation de notre application.

Chapitre IV

Réalisation

Introduction

Dans ce chapitre on passe à l'implémentation en présentant le contenu des pages de notre application et les outils de développements utilisés avec un aperçu sur les maquettes des interfaces, ainsi que les différentes fonctionnalités de l'application illustrées par des images écrans.

I. Contenu des pages

Le tableau suivant décrit un bref contenu des pages:

Rubriques	Contenu
Page d'index	Représente le portail de l'application, il contient une illustration à propos de l'application et un lien vers le page d'authentification administrateur.
Authentification	Cette page contient le formulaire d'authentification destiné à l'administrateur de l'application.
Accueil	Elle contient les liens vers les serveurs à gérer et les actions qui permettent d'arrêter, de démarrer et de redémarrer ces serveurs.
SQUID	Cette rubrique nous permet de gérer le serveur proxy Squid. Elle contient : ➢ Deux formulaires : l'un pour la modification du contenu de fichier de configuration, l'autre pour l'ajout des paramètres de configuration à la fin du fichier. ➢ Des boutons permettant de démarrer, redémarrer et d'arrêter le serveur SQUID.
DNS	Cette page nous permet de gérer le serveur DNS. Elle contient : ➢ Deux formulaires : l'un pour la modification du contenu de fichier de configuration, l'autre pour l'ajout des paramètres de configuration à la fin du fichier. ➢ Des boutons permettant de démarrer, redémarrer et d'arrêter le serveur DNS. ➢ Un bouton permettant de gérer le cas de modification de plusieurs zones. ➢ Un bouton permettant de créer ou de modifier un fichier de base de données dans le cas d'ajout d'une zone.

DHCP	Cette page nous permet de gérer le serveur DHCP. Elle contient : ➢ Deux formulaires : l'un pour la modification du contenu de fichier de configuration, l'autre pour l'ajout des paramètres de configuration à la fin du fichier. ➢ Des boutons permettant de démarrer, redémarrer et d'arrêter le serveur DHCP. ➢ Des boutons permettant d'effectuer des modifications dans le cas de paramètres répétés.

Tableau 4 : Contenu des pages

II. Les outils de développement

Le développement d'une telle application nécessite le passage par quelques étapes et l'utilisation de quelques outils, nous citons ci-dessous les outils que nous avons utilisés.

- **Macromedia Dreamweaver 8**

Dreamweaver est un éditeur WYSIWYG (what you see is what you get, ce que vous voyez est ce que vous obtenez) destiné à la conception, au codage et au développement de sites, de pages et d'applications Web. Quelque soit l'environnement de travail utilisé, Dreamweaver propose des outils qui nous aident à créer des applications Web. Ce logiciel est édité par Macromedia.

Les fonctions d'édition visuelle de Dreamweaver nous permettent de créer rapidement des pages sans rédiger une seule ligne de code. **[w08]**

- **Pacestar UML Diagrammer**

Un outil qui permet de générer les diagrammes UML rapidement et facilement avec un minimum d'effort de formation.

- **Macromedia Fireworks 8**

Un logiciel qui permet de créer, de modifier et d'améliorer la qualité des images.

- **Macromedia Flash 8**

Nous l'utilisons afin de créer des animations graphiques pour l'application et la prise en charge des autres logiciels tels que Photoshop et Illustrator…

- **Les feuilles de styles en cascade CSS** (Eng : Cascading Style Sheets)

Les feuilles de style en cascade (CSS) regroupent des règles de mise en forme qui déterminent l'aspect du contenu d'une page Web. Les styles CSS nous permettent de gérer en souplesse l'aspect d'une page, tant en termes de précision de l'emplacement de la présentation qu'en termes de choix de polices et de styles de caractères spécifiques. **[w09]**

V.2.1. Langage et environnement de développement

- **Perl**

Perl est un langage de programmation orienté objet qui permet l'extraction et le traitement rapide d'informations.

L'avantage du Perl est que le programme est un langage en script et non un code source à compiler ce qui permet de porter le programme directement sur toutes les plates-formes supportées (Unix, Windows, Mac...) sans autre opération qu'une simple copie de fichier.

Un script Perl est à la base interprété, mais cette période d'interprétation n'est que passagère. En fait, il y a avant chaque exécution une compilation transparente pour l'utilisateur. L'exécutable est généré en mémoire puis exécuté à partir de là. Ce qui augmente légèrement le temps d'exécution par rapport à un programme équivalent en C, mais c'est souvent bien plus rapide à écrire, plus simple à maintenir et à porter **[w010]**.

- **Scripts CGI**

SCGI est une interface permettant l'exécution de programmes externes par un serveur HTTP. Plus généralement, CGI est en fait un standard pour l'écriture de passerelles entre des serveurs d'informations tel que HTTP et des programmes externes. L'interface CGI permet de faire communiquer le programme et le serveur HTTP **[w013].**

La CGI est un programme exécutable qui peut donc utiliser n'importe quel langage de programmation :

- Shell-script
- PERL
- Langages compilés C/C++

Pour des raisons de portabilité, le langage PERL est souvent utilisé pour l'écriture des CGI. **[w011]**

- **HTML**

Les pages HTML sont des pages Hyper Texte, pouvant inclure des images, des sons, et des liens vers d'autres pages HTML, elles doivent être visualisées avec un Navigateur (comme Netscape, Microsoft Explorer, google chrome, mozila firefox…etc.)**[w012].**

- **JavaScript**

JavaScript est un langage de script léger, orienté objet et multiplateforme.

III. Orientation graphique

Pour plus de dynamisme, les bannières et certains boutons de l'application sont crées en flash, ils sont basés sur un font bleu linéaire.

IV. Orientation chromatique

Les interfaces d'application sont un mélange de deux couleurs dominantes Bleu et blanc choisi soigneusement pour leur professionnalisme et pour la sensation de confort qu'ils donnent aux utilisateurs.

Pour ce qui est de la couleur du texte on trouve le noir ou le blanc et le bleu qui sont adéquat à la visibilité et au confort lors de la lecture.

V. Orientation typographique

Le choix de la police est un élément important, on a utilisé deux polices différentes qui sont : Arial et Verdana.

Arial et Verdana qui sont des polices système interprétées par tous les ordinateurs.

VI. Arborescence de l'application web

C'est un schéma représentant les différentes pages de l'application organisées d'une manière logique et hiérarchique sous forme d'un arbre, commençant par la page d'index suivie des autres pages (voir Figure 35).

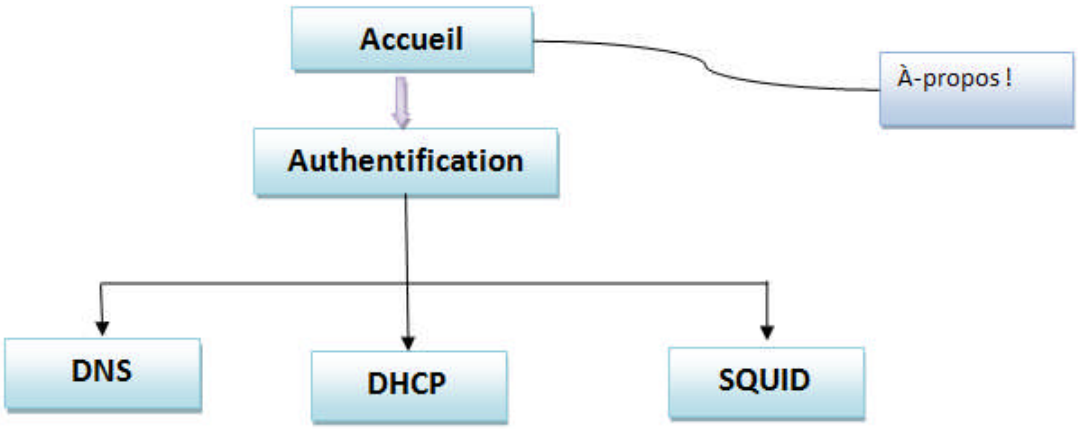

Figure 35 : Arborescence de l'application web

VII. Interface de l'application web

Dans ce qui suit, nous allons présenter quelques interfaces dans notre application web.

- ✓ **Interface d'accueil**

Dans cette page on trouve un lien permettant à l'administrateur d'accéder à la page d'authentification.

Figure 36 : Interface d'accueil

- ✓ **Interface d'authentification**

C'est une page qui permet à l'administrateur d'accéder à son espace à l'aide de saisie d'un login et d'un mot de passe.et on trouve un lien qui permet à l'administrateur de modifier le login et le mot de passe.

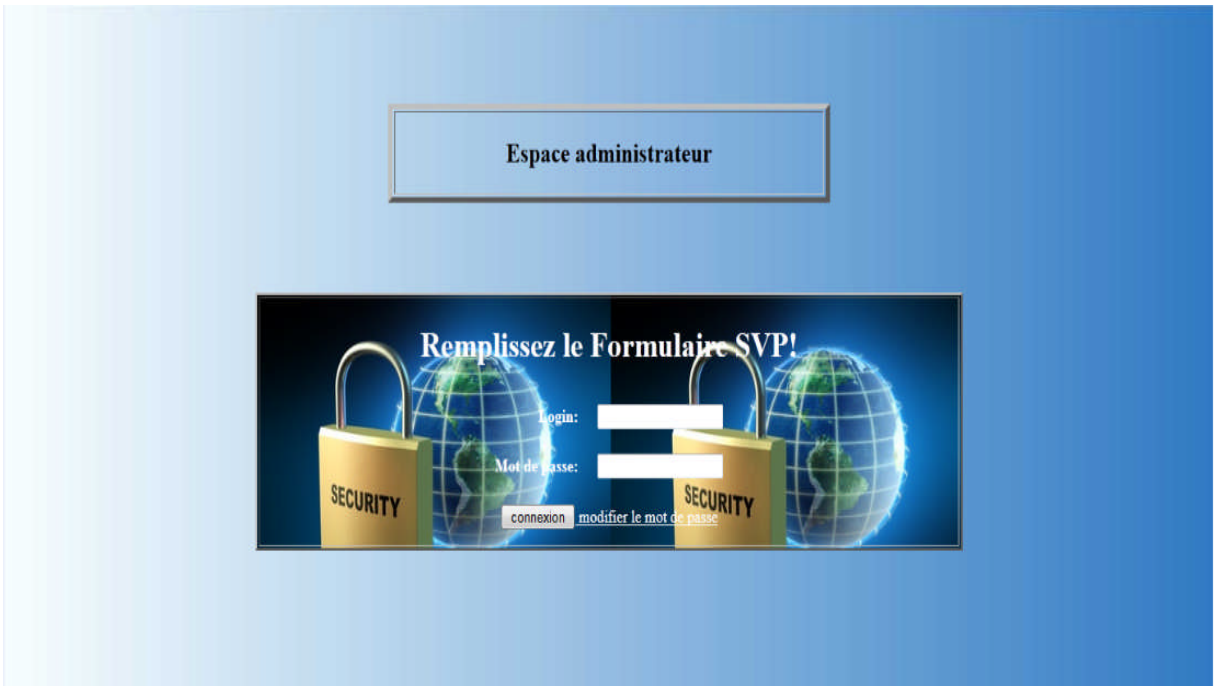

Figure 37 : interface d'authentification

✓ **Interface d'administration**

Cette page offre un aperçu de notre application, on trouve les liens vers les serveurs à gérer par l'administrateur ainsi que les actions à effectuer sur les serveurs.

Figure 38 : Interface d'administration

L'image ci-dessous illustre un exemple de lien vers le serveur DNS ou on trouve un formulaire qui contient les différents paramètres modifiable du fichier de configuration named.conf.

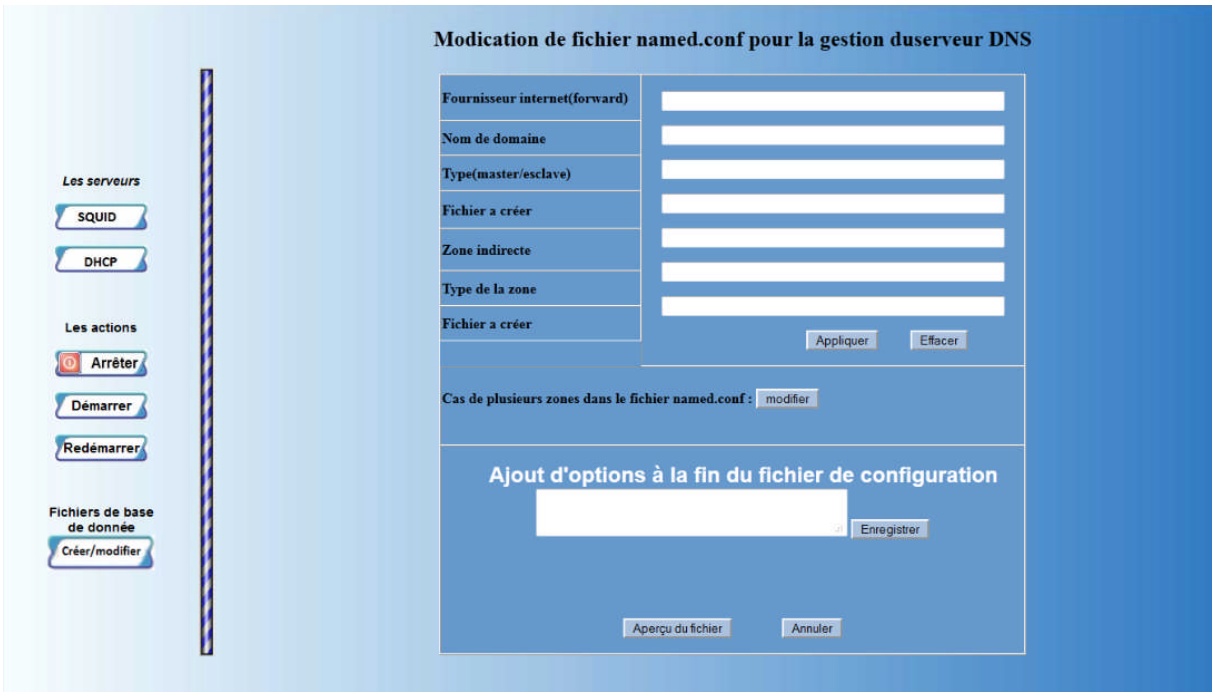

Figure 39 : Interface de gestion du serveur DNS

Conclusion

On a illustré dans ce chapitre une vue globale de l'implémentation de notre application qui est une interaction entre l'HTML et les scripts CGI ainsi que la description des pages dynamiques et leur contenu.

Conclusion et perspectives

L'utilisation croissante des applications web est due aux nombreux avantages qu'elles apportent :

- Aucune installation de logiciel n'est nécessaire sur le poste client, un navigateur Web suffit.
- Possibilité d'accès à distance.
- Compatibilité avec différents systèmes d'exploitation : Windows, Mac OS ou Linux.

Pour saisir ces avantages, notre projet de fin de cycle a été abordé dans le but de réaliser une application Web pour la gestion des services réseau sous linux permettant la configuration et la modification des fichiers.

Après avoir fixé les spécifications des besoins, nous avons procédé à la conception de l'architecture en se basant sur la modélisation UML guidée par une démarche UP.

Nous avons aussi implémenté la solution conçue ainsi que ses tests unitaire, intégration et architecturale et validations pour aboutir à une application fiable et optimale.

Ce projet nous a été bénéfique car il nous a permis de bien nous familiariser avec le système linux, de nous perfectionner en améliorant nos connaissances en programmation et en conception, de bien comprendre et mettre en œuvre le déroulement d'un cycle de vie d'un logiciel et de découvrir le monde d'administration réseau.

Vu le manque de temps, on n'a pas pu ajouter des modules à notre application, pour permettre la mise à jour de l'application en cas de disponibilité de nouvelles versions de serveurs.

Glossaire

Termes	Signification
FDDI	Fiber Distributed Data Interface : un type de réseau informatique LAN ou MAN permettant d'interconnecter plusieurs LAN à une vitesse de 100 Mbit/s sur de la fibre optique.
WAN	Wide Area Network : réseau généralement constitué de plusieurs sous-réseaux hétérogènes et s'étendant sur une région ou un pays entier.
TCP	Transmission Control Protocol : est un ensemble de règles (protocole) utilisées pour le contrôle de transmission de paquets.
TFTP	Trivial File Transfert Protocol : est un protocole simplifié de transfert de fichiers. Il fonctionne en UDP sur le port 69.
SNMP	Simple Network Management Protocol : Il s'agit d'un protocole qui permet aux administrateurs réseau de gérer les équipements du réseau et de diagnostiquer les problèmes de réseau.
SMTP	Simple Mail Transfer Protocol : Dans un réseau TCP/IP, ce protocole gère l'envoi de mails entre différents serveurs. Il est également utilisé pour l'envoi de mails à partir des ordinateurs clients.
UDP	User Datagram Protocol : situé en couche 4. Il n'ouvre pas de session et n'effectue pas de control d'erreur. Il est alors appelé "mode non connecté".
IP	Internet Protocol : Le protocole IP fait partie de la couche Internet de la suite de protocoles TCP/IP. C'est un des protocoles les plus importants d'Internet car il permet l'élaboration et le transport des datagrammes IP.
RTP	Real-time Transport Protocol : e but de RTP et de fournir un moyen uniforme de transmettre sur IP des données soumises à des contraintes de temps réel (audio, vidéo, ...).
ICMP	Internet Control Message Protocol est l'un des protocoles fondamentaux constituant la suite de protocoles Internet. Il est utilisé pour véhiculer des messages de contrôle et d'erreur pour cette suite de protocoles.
ARP	Address resolution Protocol : est un protocole effectuant la traduction d'une adresse de protocole de couche réseau en une adresse MAC .
RARP	RARP (Reverse Address Resolution Protocol) : est un protocole permettant à un équipement d'obtenir son adresse IP en communiquant son adresse Ethernet à un serveur RARP.
X.25	X25 est une norme internationale de transmission de données par paquets qui a vu le jour vers le milieu des années 70 à l'UIT (Union International des Télécommunications).
Frame Relay	Le relayage de trames (ou FR, pour l'anglais *Frame Relay*) est un protocole à commutation de paquets situé au niveau de la couche de liaison du modèle OSI, utilisé pour les échanges intersites (WAN).

GNU	GNU est un système d'exploitation libre lancé en 1984 et maintenu par le projet GNU. Son nom est un acronyme récursif qui signifie en anglais « GNU's Not UNIX » (littéralement, « GNU n'est pas UNIX »). Il reprend les concepts et le fonctionnement d'UNIX
POSIX	Portable Operating System Interface : c'est le standard officiel qui définit les interfaces communes à tous les systèmes de type Unix.
BOOTP	Le protocole Bootstrap (BOOTP) est un protocole de configuration d'hôte développé avant DHCP.
ICANN	Internet Corporation for Assigned Names and Numbers: le rôle de l'ICANN est de superviser le réseau interconnecté gigantesque et complexe d'identifiants uniques qui permettent aux ordinateurs de se reconnaître entre eux sur Internet.
InterNIC	Définition du mot INTERNIC, Ensemble de serveurs regroupant des informations de statistiques sur Internet, mis à disposition des utilisateurs par la fondation américaine NSF.
AFNIC	Association française pour le nommage Internet en coopération.
BIND	Un serveur qui héberge le service DNS est appelé "serveur de noms". Ubuntu est livré par défaut avec BIND (Berkley Internet Naming Daemon).
LAN	Local Area Network:réseau local.

Bibliographie

[B1] : Peter MULLER, « INTERNET : Créer Votre SITE Web », Ed. Micro Application, 2000.

[B2] : Guy PUJOLLE, « Les réseaux », Ed. EYROLLES, 2011.

[B3] : Ivar Jackobson, James Rambaugh, « Administration réseau sous linux», Ed. Eyrolles, 2004.

[B4] : Marc Pybourdin, Emmannuel Macé, David Sztykman, Vincent Steenhoute, « Linux Administration système et réseau », Ed : DUNOD, 2008.

[B5] : Jean- Marie Defrance, « Premières applications Web 2.0 avec Ajax et PHP », Ed : EYROLLES, 2006.

[B6] : Laurent AUDIBERT, « UML 2 - de l'apprentissage à la pratique (cours et exercices) », Ed : Ellipses, 2004.

[B7] : Ivar Jackobson, Grady Boosh, James Rambaugh, «Le processus unifié de développement logiciel», Ed : Editions Eyrolles, 1999.

[B8] : Ivar Jackobson, Grady Boosh, James Rambaugh, « Le guide de modélisation UML », Ed : Editions Eyrolles, 2003.

Webographie

[W1]: http://www.lococheb.net/intro_reseau.php

[W2] : http://www.lococheb.net/topologie.php#Reseau en étoile

[W3]: http://www.httr.ups-tlse.fr/pedagogie/cours/intro/topo.htm

[W4]: http://www.lococheb.net/osi.php#top

[W5]: http://technet.microsoft.com/fr-fr/library/cc786900(v=ws.10).aspx

[W6]: www.ling.umontreal.ca/lhomme/docs/docettermino.pdf

[W7]: http://aeris.11vm-serv.net/cours/internet/internet.html

[W1]: http://c2i-n1.fr/50-fillezila (gl)/35-Fillezilla-ftp.pdf

[W9]: http://www.linux-kheops.com/doc/cours/jallet2/c0605.html

[W10]: http://www.commentcamarche.net/contents/utile/email.php3

[W11]: http://www.crtc.gc.ca/fra/info_sht/t1029.htm

[W12]: http://www.definitions-marketing.com/Definition-Messagerie-instantanee

[Ww1]:http://wwwlemm.univlille1.fr/divers/internet/autofor/html/nav/histoi/histo.htm#d

[Ww2]: http://glossaire.infowebmaster.fr/navigateur-web/

[Ww3]: http://www.definitions-webmarketing.com/Definition-Serveur-web

[Ww4]: http://www.f2x.fr/application-web/

[Ww5]: http://www.netalya.com/fr/reseaux8.asp

[Ww6]: http://www.reseaucerta.org/docs/cotecours/archiTechniquePGI.pdf

[Ww7]: http://autoformation.freehostia.com/ClientServeur/Introduction/Architecture%203-Tier.htm

[Ww8]:http://www.sunriseconsult.com/Solutions/MicrosoftDynamicsAX/TechnicalFeatures.aspx

[Ww9]:http://perso.modulonet.fr/~placurie/Ressources/BTS1-ALSI/Chap-12-%20Le%20client-serveur.pdf

[Ww10]: http://www.awt.be/web/res/index.aspx?page=res,fr,fic,060,002

[w01]:http://www.linux-france.org/prj/edu/archinet/systeme/ch27s02.html74

[w02]:http://www.labouret.net/dns/

[w03]: http://www.reseaumaroc.com/files/Serveur%2*
0Proxy%20%20Squid.pdf
[w04]: http://free.korben.info/index.php/Proxy
[w05]:http://free.korben.info/index.php/Proxy 2005/squid.pdf
[w06]:http://www-igm.univ-mlv.fr/~dr/XPOSE2003/Squid/ch01s02.html
[w04]: http://free.korben.info/index.php/Proxy
[w05]:http://free.korben.info/index.php/Proxy
[w06]:http://www-igm.univ-mlv.fr/~dr/XPOSE2003/Squid/ch01s02.html
[w07]:http://www.slideshare.net/mostefaiamine/cours-gnie-logiciel-cours-2-cycles-de-vie
[w08]:http://webyo.net/article.php?id_article=36
[w09]: http://webyo.net/Dreamweaver-Les-styles
[w010]:http://www.ftls.org/fr/initiation/perl/index2.shtml#s2
[w011]:http://chl.be/glmf/www.linuxmag-france.org/old/lm1/cgi.html
[w012]:http://www.ftls.org/fr/initiation/HTML/index2.shtml
[w013]: http://www.associationadeil.org/spip.php?rubrique46
[w014]:http://www.linux-france.org/article/web/docs/cgi-bin-april/html/cgi-bin-april-1.html
[w015]: www.medicalistes.org/spip/IMG/pdf/Internet-Medecine-Generale.pdf
[artc 1]: http://cyber-avenue.org/SPIP/spip.php?article89
[artc 2]: http://www.pileouface.org/linux/documentation/bases_linux.pdf
[artc 3]: http://aful.org/ressources/presentation/linux
[artc 4]: http://www.maretmanu.org/homepage/inform/linux/quoi.php
[artc 5]: http://www.inetdoc.net/dev/socket-c/socket-c.context.html
[artc 6]:http://linuxfranch-county.org/docs/guideLLLinux/systeme_fichiers_2-2.pdf
[artc7]: Article publié dans GNU/Linux Magazine France, numéro 79 (Janvier 2006) URL : http://chl.be/glmf/lionel.tricon.free.fr/Articles/acl/article_acl.pdf
[artc8]: http://sebastien.nameche.fr/supports/AdministrationLinux.pdf
[Th1] : Tewfik ZIADI, « Manipulation de Lignes de Produits en UML », 2004.
[Th2] : Mohamed HADJ KACEM, « Modélisation des applications distribuées à architecture dynamique : Conception et Validation », 2006.
[Th3] : Benoit Baudry, «Assemblage testable et validation de composants », 2007.
[Th4] : Jacques Demerjian, « Le langage UML », 2000.

ANNEXE

Illustration du fichier named.conf

```
options {
        // DNS tables are located in the /var/named directory
        directory "/var/named";

        // Forward any unresolved requests to our ISP's name server
        // (this is an example IP address only -- do not use!)
        forwarders {
                123.12.40.17;
        };

        /*
         * If there is a firewall between you and nameservers you want
         * to talk to, you might need to uncomment the query-source
         * directive below.  Previous versions of BIND always asked
         * questions using port 53, but BIND 8.1 uses an unprivileged
         * port by default.
         */
        // query-source address * port 53;
};

// Enable caching and load root server info
zone "named.root" {
        type hint;
        file "";
};

// All our DNS information is stored in /var/named/mydomain_name.db
// (eg. if mydomain.name = foobar.com then use foobar_com.db)
zone "mydomain.name" {
        type master;
        file "mydomain_name.db";
        allow-transfer { 123.12.41.40; };
};

// Reverse lookups for 123.12.41.*, .42.*, .43.*, .44.* class C's
// (these are example Class C's only -- do not use!)
zone "12.123.IN-ADDR.ARPA" {
        type master;
        file "123_12.rev";
        allow-transfer { 123.12.41.40; };
};
```

Illustration du fichier dhcpd.conf

```
#
# Sample configuration file for ISC dhcpd for Debian
#
# Attention: If /etc/ltsp/dhcpd.conf exists, that will be used as
# configuration file instead of this file.
#
#

# The ddns-updates-style parameter controls whether or not the
server will
# attempt to do a DNS update when a lease is confirmed. We
default to the
# behavior of the version 2 packages ('none', since DHCP v2
didn't
# have support for DDNS.)
ddns-update-style none;

# option definitions common to all supported networks...
option domain-name "example.org";
option domain-name-servers ns1.example.org, ns2.example.org;

default-lease-time 600;
max-lease-time 7200;

# If this DHCP server is the official DHCP server for the local
# network, the authoritative directive should be uncommented.
#authoritative;

# Use this to send dhcp log messages to a different log file (you
also
# have to hack syslog.conf to complete the redirection).
log-facility local7;

# No service will be given on this subnet, but declaring it helps
the
# DHCP server to understand the network topology.

#subnet 10.152.187.0 netmask 255.255.255.0 {
#}

# This is a very basic subnet declaration.

#subnet 10.254.239.0 netmask 255.255.255.224 {
#  range 10.254.239.10 10.254.239.20;
#  option routers rtr-239-0-1.example.org, rtr-239-0-
2.example.org;
```

```
# This declaration allows BOOTP clients to get dynamic addresses,
# which we don't really recommend.

#subnet 10.254.239.32 netmask 255.255.255.224 {
#  range dynamic-bootp 10.254.239.40 10.254.239.60;
#  option broadcast-address 10.254.239.31;
#  option routers rtr-239-32-1.example.org;
#}

# A slightly different configuration for an internal subnet.
#subnet 10.5.5.0 netmask 255.255.255.224 {
#  range 10.5.5.26 10.5.5.30;
#  option domain-name-servers ns1.internal.example.org;
#  option domain-name "internal.example.org";
#  option routers 10.5.5.1;
#  option broadcast-address 10.5.5.31;
#  default-lease-time 600;
#  max-lease-time 7200;
#}

# Hosts which require special configuration options can be listed
in
# host statements.   If no address is specified, the address will
be
# allocated dynamically (if possible), but the host-specific
information
# will still come from the host declaration.

#host passacaglia {
#  hardware ethernet 0:0:c0:5d:bd:95;
#  filename "vmunix.passacaglia";
#  server-name "toccata.fugue.com";
#}

# Fixed IP addresses can also be specified for hosts.   These
addresses
# should not also be listed as being available for dynamic
assignment.
# Hosts for which fixed IP addresses have been specified can boot
using
# BOOTP or DHCP.   Hosts for which no fixed address is specified
can only
# be booted with DHCP, unless there is an address range on the
subnet
# to which a BOOTP client is connected which has the dynamic-
bootp flag.
```

```
#host fantasia {
#  hardware ethernet 08:00:07:26:c0:a5;
#  fixed-address fantasia.fugue.com;
#}

# You can declare a class of clients and then do address
allocation
# based on that.   The example below shows a case where all
clients
# in a certain class get addresses on the 10.17.224/24 subnet,
and all
# other clients get addresses on the 10.0.29/24 subnet.

#class "foo" {
#  match if substring (option vendor-class-identifier, 0, 4) =
"SUNW";
#}

#shared-network 224-29 {
#  subnet 10.17.224.0 netmask 255.255.255.0 {
#    option routers rtr-224.example.org;
#  }
#  subnet 10.0.29.0 netmask 255.255.255.0 {
#    option routers rtr-29.example.org;
#  }
#  pool {
#    allow members of "foo";
#    range 10.17.224.10 10.17.224.250;
#  }
#  pool {
#    deny members of "foo";
#    range 10.0.29.10 10.0.29.230;
#  }
#}
```

Illustration du fichier squid.conf

```
#     WELCOME TO SQUID 2.7.STABLE9
#     ----------------------------
#
#     This is the default Squid configuration file. You may
wish
#     to look at the Squid home page (http://www.squid-
cache.org/)
#     for the FAQ and other documentation.
#
#     The default Squid config file shows what the defaults for
#     various options happen to be.  If you don't need to
change the
#     default, you shouldn't uncomment the line.  Doing so may
cause
#     run-time problems.  In some cases "none" refers to no
default
#     setting at all, while in other cases it refers to a valid
#     option - the comments for that keyword indicate if this
is the
#     case.
#


#  Configuration options can be included using the "include"
directive.
#  Include takes a list of files to include. Quoting and
wildcards is
#  supported.
#
#  For example,
#
#  include /path/to/included/file/squid.acl.config
#
#  Includes can be nested up to a hard-coded depth of 16
levels.
#  This arbitrary restriction is to prevent recursive include
references
#  from causing Squid entering an infinite loop whilst trying
to load
#  configuration files.


# OPTIONS FOR AUTHENTICATION
#
```

Résumé

Actuellement, l'Administration de Réseau Informatique et la gestion de ses services occupent une place prépondérante. Sans une bonne maîtrise de ces dernières, aucune entreprise ne peut espérer un avenir.

Le dynamisme est l'une des caractéristiques les plus essentielles de l'informatique. C'est ceci qui nous a poussés à créer une application web dynamique pour la gestion des services réseaux, accessible par un administrateur dans un réseau informatique local.

Le but recherché durant l'étude est d'avoir une application fiable, facile à utiliser, elle permet à l'administrateur de configurer les fichiers de différents serveurs sous linux.

Pour la réalisation de notre application on a opté pour le CGI comme langage de script coté serveur pour donner du dynamisme aux pages.

Abstract

Currently, Computer Network Administration and management of its services dominate. Without a good command of the latter, no company can hope for a future.

Dynamism is one of the most essential characteristics of the computer. This is what prompted us to create a dynamic web application for managing network services, accessible by an administrator in a local computer network.

The aim during the study is to have an application reliable, easy to use; it allows the administrator to configure the files from different servers on Linux.

For the realization of our application we opted for the CGI scripting language on the server side to give the dynamic pages.