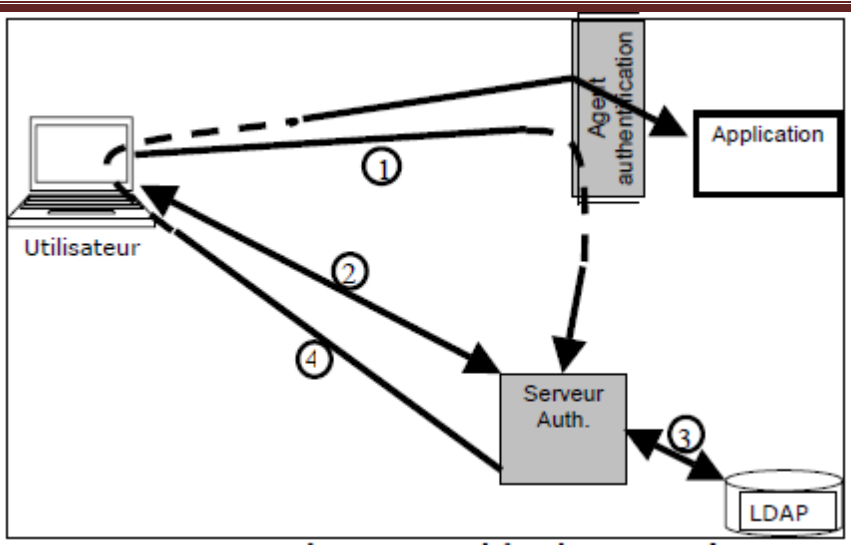


Figure 4: Architecture simple d'un SSO web

3. CAS(*Central Authentification Service*)

3.1. Définition

Développé par l'Université de Yale, CAS (*Central Authentication Service*) met en œuvre un serveur d'authentification accessible, CAS est une architecture pour implémenter un système d'authentification unique (SSO) en s'appuyant sur des systèmes d'authentification tiers comme LDAP, Active Directory, une base SQL, etc. L'architecture est générique et ne dépend pas du système d'authentification choisi. ***[19]***

3.2. Le mécanisme d'Architecture CAS

Le serveur CAS

L'authentification est centralisée sur une machine unique, le serveur CAS. Ce serveur est le seul acteur du mécanisme CAS à avoir connaissance des mots de passe des utilisateurs. Son rôle est double :

- **authentifier** les utilisateurs ;
- **transmettre et certifier l'identité** de la personne authentifiée (aux clients CAS).

*Les navigateurs (*web*)*

Les navigateurs doivent satisfaire les contraintes suivantes pour bénéficier de tout le confort de CAS :

- disposer d'un moteur de chiffrement leur permettant d'utiliser le protocole HTTPS.

- savoir effectuer des redirections HTTP(accéder à une page donnée dans un entête Location lors d'une réponse 30x à une première requête HTTP) et interpréter le langage JavaScript.
- savoir stocker des cookies. En particulier, les *cookies* privés ne devront être retransmis qu'au serveur les ayant émis pour garantir la sécurité du mécanisme CAS.

Ces exigences sont satisfaites par tous les navigateurs classiquement utilisés, à savoir *Microsoft Internet Explorer* (depuis 5.0), *Netscape Navigator* (depuis 4.7) et *Mozilla*.

3.3. Les clients CAS

Une application web muni d'une librairie cliente ou un serveur web utilisant le module *mod_cas* est alors appelé « client CAS ». Il ne délivre les ressources qu'après s'être assuré que le navigateur qui l'accède se soit authentifié auprès du serveur CAS. Parmi les clients CAS, on trouve :

- des librairies correspondant aux langages communément employés en programmation web dynamique (*Perl*, *Java*, JSP,PHP, ASP) .
- un module *Apache*, qui permet de protéger des documents statiques.
- un module PAM, qui permet d'authentifier les utilisateurs au niveau système.

3.4. Authentification d'un utilisateur

Un utilisateur non déjà précédemment authentifié, ou dont l'authentification a expiré, et qui accède au serveur CAS se voit proposer un formulaire d'authentification, dans lequel il est invité à entrer son nom de connexion et son mot de passe :

Si les informations sont correctes, le serveur renvoie au navigateur un *cookie* appelé TGC (*Ticket Granting Cookie*)

Le *Ticket Granting Cookie* (TGC) : est le passeport de l'utilisateur auprès du serveur CAS. Le TGC, à durée de vie limitée (typiquement quelques heures), est le moyen pour les navigateurs d'obtenir auprès du serveur CAS des tickets pour les clients CAS sans avoir à se ré-authentifier. C'est un *cookie* privé (n'est jamais transmis à d'autres serveurs que le serveur CAS) et protégé (toutes les requêtes des navigateurs vers le serveur CAS se font sous HTTPS). Comme tous les tickets utilisés dans le mécanisme CAS, il est opaque (ne contient aucune information sur l'utilisateur authentifié) : c'est un identifiant de session entre le navigateur et le serveur CAS.***[10]***

4. L'authentification unifiée

Le concept d'authentification unifiée c'est le concept qu'on a utilisé dans notre mémoire se compose d'un annuaire LDAP qui est un référence fournissent un référentiel d'informations sur les autre plateformes. Le référentiel central utilisé pour les données LDAP qui est connecté et bien configurer avec le serveur de messagerie Zimbra et les plateformes Dokeos et Joomla , cette authentification va se concentrer beaucoup sur le travail du serveur LDAP qui regroupe tous les comptes des utilisateurs dans un seul annuaire , avec ce dernier l'utilisateur doit s'authentifier et accéder au plateformes , donc c'est une authentification unifiée qui unifier ces plateformes Zimbra , Joomla (CMS), Dokeos (LMS) .

5. Conclusion

Dans ce chapitre on a parlé de SSO single Sign-ON son objectif et ses différentes architectures et le CAS qui est une autre architecture de l'authentification unifiée et pour le chapitre suivant on détermine les outils nécessaires pour notre implémentation.

Chapitre III :Les outils d'authentification unifiée

Les outils utilisés dans notre travail

1. Introduction :

Comme on a parlé dans le chapitre précédent sur les techniques d'authentification qui est le SSO (singl-sign on) et le CAS (central authentification service) dans ce chapitre on commence à présenter les outils utiles à l'authentification, pour notre travail on a montré les outils utilisable à notre environnement universitaire qui est premièrement le serveur LDAP c'est l'annuaire pour la base de données des utilisateurs avec le messagerie électronique Zimbra et les deux plateformes Dokeos et Joomla .

2. Présentation de l'environnement

2.1. Le CentOS :

CentOS est une distribution dérivée de RedHat Entreprise Linux. Son avantage est de proposer la stabilité de RedHat sans avoir à souscrire un contrat de support, il était au final entièrement une open-source parmi ces nouveautés on notera l'amélioration des outils de virtualisation ou les outils de développements et de monitoring qui intéresseront particulièrement les développeurs C++ et Python. Dans ce domaine Cent OS 6.1 propose

également une version mise à jour d'Eclipse.
Autre point mis en avant par le site officiel, Modified (Yum) le gestionnaire de paquets en ligne de commande. Bien que restant très proche de RHEL 6.2 (Red Hat Enterprise Linux), la communauté de CentOS a également modifié le Kernel Linux, ainsi que les paquets Firefox et Apache HTTP.

2.1.1. Pourquoi choisir la distribution CentOS Enterprise Linux ?

La question qu'un utilisateur qui n'a jamais installé le système CentOS Linux, distribution Serveur, peut poser est : Pourquoi choisir CentOS Linux plutôt qu'une autre distribution Linux Serveur ? Avant de répondre à cette question, permettons –nous de révéler un peu l'historique de ce système d'exploitation.
Le toute premier release du système CentOS (Community Enterprise Operating System) créé par le groupe CentOS Développent Team est sortie au mois de mai 2004. Etant depuis une distribution 100% Open Source et totalement gratuite, CentOS est basée sur la distribution

RedHat Entreprise Linux (RHEL). Elle utilise les sources de la RHEL (téléchargeables librement sur Internet) pour régénérer la RedHat à l'identique. On peut donc considérer la CentOS comme une version gratuite de la RedHat. Le support technique est alors de type communautaire : il se fait gratuitement et ouvertement via les listes de diffusion et les forums de la communauté CentOS.

Revenons maintenant à la fameuse question: Pourquoi choisir ce système ?

Chaque système d'exploitation a ses qualités et ses défauts. Il faut tout simplement les distinguer et faire ensuite son propre choix par rapport à ses besoins et attentes. Voici donc les avantages qui m'ont fait porter notre choix sur cette distribution :

- Support gratuit. Mises à jour applicatives et les patchs de sécurité réguliers.
- Stabilité quasi-équivalente à la distribution RedHat utilisé dans de gros environnements de production.
- Cycle de développement suivant celui de RedHat (7ans pour un release).
- L'outil "YUM" facilitant l'exploitation et la gestion des paquets au format RPM.
- Nombreux manuels en ligne (en anglais et en français) de RedHat, 100% compatibles CentOS Linux.

Bien sûr, il n'y a pas d'avantages sans inconvénients, mais il faut vivre avec :

- Limite au niveau des dépôts standards fournissant les paquets RPM.
- Difficulté de création de ses propres paquets RPM.[11]

2.2. Annuaire LDAP

2.2.1. Introduction

L'informatique et la gestion de l'information prend une place de plus en plus importante dans notre société, particulièrement en entreprises. La multiplication des applications et des serveurs rend cette information difficile à maîtriser car très volatile et éparse. Ceci entraîne bien souvent une obsolescence, voire une incohérence des données stockées. Les annuaires LDAP offrent une réponse à ce problème en proposant de centraliser les informations.

2.2.2. Historique et aperçu des annuaires existants

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

Malheureusement, le protocole DAP s'avère difficile à mettre en œuvre et ne fonctionne pas sur les réseaux TCP/IP. En 1993, l'Université du Michigan réfléchit donc à un moyen de pallier ces deux problèmes : elle met en place le protocole LDAP (Lightweight Directory AccessProtocol), au départ simple "connecteur" TCP/IP avec des annuaires X.500.

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

LDAP est donc une évolution de la norme X.500. Sa version actuelle est la version 3 (RFCs 2251, 4511, 4512, 4513), elle propose les évolutions suivantes par rapport à la version 2 :

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

2.2.3.Types d'annuaires

D'autres types d'annuaires existent, vous les utilisez très certainement :

• DNS : Domain Name Services
• NIS : Network Information Services
• Whois : base d'information concernant les noms de domaines

2.2.4.Les annuaires LDAP

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

• Open LDAP
• Apache Directory Server
• Sun (One/Java) Directory Server
• Active Directory

2.2.5. Les concepts du protocole LDAP

On a coutume de regrouper les caractéristiques et fonctionnalités de l'annuaire LDAP sous la forme de quatre modèles :

• Le modèle de nommage : définit comment l'information est stockée et organisée
• Le modèle fonctionnel : définit les services fournis par l'annuaire (recherche, ajout, ...)
• Le modèle d'information : définit le type d'informations stockées
• Le modèle de sécurité : définit les droits d'accès aux ressources

2.2.6. Le protocole

LDAP signifie "Lightweight Directory Access Protocol".
LDAP est un protocole, ce qu'il signifie que son rôle est de présenter des informations. Un serveur LDAP agit en tant qu'intermédiaire entre une source de données et un client.
Nous verrons qu'en tant qu'intermédiaire il définit quelques conventions, notamment l'organisation des données qu'il présente qui sera sous forme hiérarchique, mais aussi un format d'échange standard.
LDAP fonctionne sur le port TCP 389 (par défaut).

2.2.7. Organisation des données (modèle de nommage)

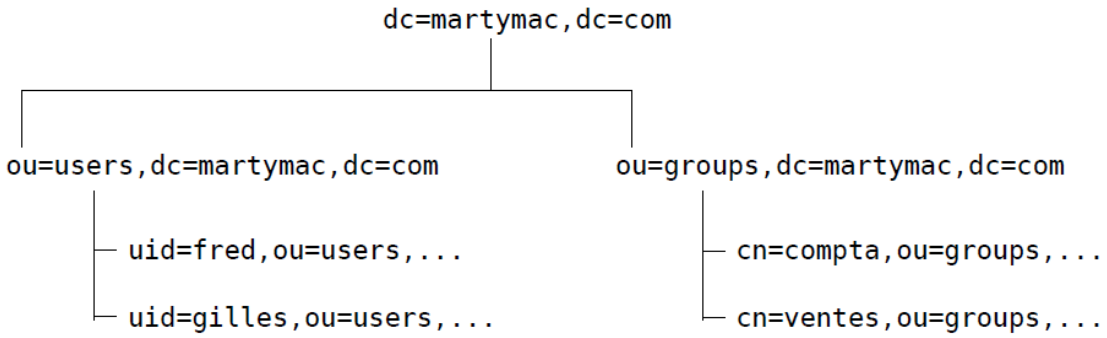
2.2.7.1. Introduction

Le modèle de nommage est la manière dont sont organisées les données dans l'annuaire. Etudions cette organisation plus en détails...

2.2.7.2. La représentation hiérarchique des données

LDAP organise les données de manière hiérarchique dans l'annuaire. Ceci signifie que toutes les informations découlent d'une seule et même "racine".

Voici un exemple d'arborescence LDAP :



Cette arborescence est liée au nommage de chaque élément : un élément marque son appartenance à l'élément supérieur en reprenant le nom, qu'il complète par le sien.

Ainsi, en étudiant simplement le nom de l'élément :

"cn=ventes, ou=groups, dc=martymac, dc=com."

il est possible de le situer dans la hiérarchie : il est situé sous l'élément "**ou=groups**" qui lui-même est situé sous l'élément "**dc=martymac, dc=com.**".

2.2.7.3. Termes à connaître

Voici quelques termes :

• Chaque élément est appelé une **entrée** (an entry). Une entrée peut être un branchement (un **noeud**) ou un élément terminal (une **feuille**).

• Chaque élément possède un **DN** (Distinguished Name). Le DN est le nom complet de l'élément qui permet de le positionner dans l'arborescence. Il est unique dans l'annuaire.

Exemple : "cn=ventes, ou=groups, dc=martymac, dc=com"

• Chaque élément possède également un **RDN** (Relative Distinguished Name). Le RDN est la partie du **DN** de l'élément qui est relative au **DN** supérieur. Le RDN d'un élément ne permet pas de l'identifier de manière absolue dans l'annuaire.

Exemple : "cn=ventes"

• La **racine** est l'élément supérieur de tous les autres, c'est la base de l'arborescence. On l'appelle **root** en anglais, parfois on parle de "**root DN**".

Exemple : "dc=martymac, dc=com"

Les DN de chaque entrée sont composés au moins d'un attribut de l'élément (par exemple

"cn" ou "uid") et de sa valeur. Un attribut est l'une des caractéristiques de cet élément.
Remarquez que la racine choisie ici est composée du nom du domaine où est hébergé notre serveur LDAP, martymac.com, décomposé en "dc" (Domain Components) pour obtenir dc=martymac, dc=com .
L'arbre se découpe ensuite en deux "ou" (Organisational Units) qui constituent deux branchements : "users" et "groups", dans lesquels nous trouvons ensuite les entrées feuilles de notre arbre : les utilisateurs et les groupes.
Chacune des entrées de notre arbre correspond à un type de donnée particulier, défini par une classe d'objet. Nous étudierons ces notions par la suite.

2.2.7.4. Règles de nommage

La RFC 2253 (rendue obsolète par la RFC 4514) normalise l'écriture des DN et conseille de nepas ajouter d'espaces autour du signe "=", ni à la fin du DN. Les espaces sont autorisés par contre pour les valeurs des entrées. Ainsi, le DN suivant est correct :
"cn=Ganael Laplanche, cn=ventes, ou=groups, dc=martymac, dc=com"
Alors que celui-ci ne l'est pas :
"cn = Ganael Laplanche, cn = ventes, ou = groups, dc = martymac, dc = com"
Les majuscules seront ou non prises en compte en fonction du type d'attribut utilisé et de ses particularités.

2.2.8. Accéder à l'annuaire (modèle fonctionnel)

Il existe plusieurs types d'opérations que l'on peut effectuer sur l'annuaire, voici les plus importantes :
• Rechercher une entrée suivant certains critères
• S'authentifier
• Ajouter une entrée
• Supprimer une entrée
• Modifier une entrée
• Renommer une entrée
Certaines de ces actions, notamment la recherche, nécessitent des outils particuliers pour nous faciliter l'accès à l'annuaire

2.2.8.1. La base

La base est le DN à partir duquel nous allons agir. Pour une recherche, il s'agir du nœud à partir duquel est effectuée la recherche. Il peut s'agir de la racine de l'arbre pour une recherche sur la totalité de l'arbre, par exemple "dc=martymac, dc=com".

2.2.8.2. La portée

La portée (scope) est le nombre de niveaux sur lesquels l'action va être effectuée. Il existe 3 niveaux différents :

• SUB : l'action est effectuée récursivement à partir de la base spécifiée sur la totalité de l'arborescence.

• ONE : l'action est effectuée sur un seul niveau inférieur par rapport à la base spécifiée (les fils directs). Si l'on effectuait une recherche avec la portée ONE à partir de "dc=martymac, dc=com", nous pourrions trouver "ou=users, dc=martymac, dc=com" et "ou=groups, dc=martymac, dc=com".

• BASE : l'action est effectuée uniquement sur la base spécifiée. Une recherche sur "dc=martymac ,dc=com" avec la portée BASE renverrait cette entrée uniquement.

2.2.8.3. Les filtres

Le troisième outil à notre disposition est le filtre. Un filtre va permettre d'effectuer des tests de correspondance lors d'une recherche. Il s'agit en quelques sortes du critère de la recherche.
Il existe 4 tests basiques, qui peuvent ensuite être combinés :

• Le test d'égalité :**X=Y**

• Le test d'infériorité :**X<=Y**

• Le test de supériorité :**X>=Y**

• Le test d'approximation : **X~=Y**

Les autres opérateurs (<, >) ou des tests plus complexes peuvent être mis en place par combinaison, il faut alors utiliser les parenthèses ( ) et l'un des opérateurs suivants :

• L'intersection (et) : **&**

• L'union (ou) : **|**

• La négation (non) : **!**

Un test d'infériorité stricte pourrait donner ceci : (&(X<=Y)(!(X=Y)))
On peut combiner plus de deux éléments : (&(X=Y)(Y=Z)(A=B)(B=C)(!(C=D)))

Ces filtres seront appliqués sur des attributs choisis pour sélectionner finement les données que nous voulons extraire de notre annuaire.

2.2.8.4. Les URLs LDAP

Récemment est apparue une méthode concise et simplifiée pour interroger un annuaire LDAP. Il s'agit d'un format d'URL combinant toutes les notions que nous avons étudiées. En une seule ligne, il est possible de spécifier tous les éléments de notre requête. Voici le format de cette URL (RFC 2255, rendue obsolète par la RFC 4516) :

ldap[s]://serveur[:port][/[base[?[attributs à afficher][?[portée][?[filtre][?extensions]]]]]]

L'exemple ci-dessous recherche tous les uid de notre arbre, à partir de la branche users :

ldap://localhost:389/ou=users,dc=martymac,dc=com?uid?sub

2.2.9. Les données contenues dans l'annuaire (modèle d'information)

2.2.9.1. Les attributs

Nous avons jusqu'ici évoqué la notion d'attribut sans trop l'expliquer. Un attribut est une valeur contenue dans une entrée. Une entrée peut bien entendu contenir plusieurs attributs. Prenons l'exemple de l'entrée LDAP complète d'un compte utilisateur POSIX :

```
dn: uid=martymac,ou=users,dc=martymac,dc=com
objectClass: account
objectClass: posixAccount
cn: martymac
uid: martymac
uidNumber: 10001
gidNumber: 10001
homeDirectory: /home/martymac
userPassword:: e0NSWVBUfWJjT29IUk5SbG1HbC4=
loginShell: /bin/sh
gecos: martymac
description: martymac
```

Figure 5: Entrée complète avec le format LDIF

Ceci correspond à une entrée complète, extraite par une interrogation de l'annuaire. Le format affiché est le format **LDIF**.

Ce paragraphe présente tous les attributs, un par ligne, que comprend notre entrée. Un attribut est séparé de sa valeur par ":". Suivant son type, un attribut peut avoir plusieurs valeurs : dans ce cas, il est dit "multi-valué" et apparaît sur plusieurs lignes avec des valeurs différentes.

Nous pouvons observer ici des attributs nommés "dn", "objectClass", "cn", "uid", ...

L'attribut "dn" qui est indiqué en première ligne est le nom unique de notre entrée dans l'arbre dont nous avons parlé précédemment. Il constitue un attribut à part entière dans notre entrée.
Il est composé du dn de l'entrée supérieure, ainsi que du rdn.

Sur un annuaire LDAP la racine est toujours composée des attributs "dc" (Domain Component)associés à chacune des parties du nom de domaine où est hébergé le serveur ("dc=martymac, dc=com" pour le domaine martymac.com). Ceci est une convention. X500 préconisait les attributs "o", "l" et "c", mais LDAP a simplifié le procédé (cf. RFCs 2247, 4519,4524).L'attribut "ou" constitue une "Organisational Unit", c'est à dire une unité organisationnelle : en quelque sorte un regroupement. Nous avons choisi d'en créer deux dans notre exemple :
"users", qui accueillera nos utilisateurs et "groups", nos groupes.
Nous n'allons pas étudier chacun des attributs présents ici, cependant, nous souhaiterons porter votre attention sur l'un des attributs les plus importants, il s'agit de la classe d'objet, ou "objectClass"...

2.2.9.2. Les classes d'objets

A première vue, l'entrée présentée ci-dessus constitue un amalgame de différentes informations qui ne semblent pas organisées. Toutes ces entrées sont induites par la présence des objectClass.
L'objectClass d'une entrée est un attribut qui permet de cataloguer cette entrée. Un objectClass définit un regroupement d'attributs obligatoires ou autorisés pour une entrée.
Une entrée peut posséder un ou plusieurs objectClass. Ce sont ces objectClass qui définissent la présence de tous les autres attributs.
Ici, l'objectClass "posixAccount" rend obligatoire les attributs cn, uid, uidNumber, gidNumber et homeDirectory. Il rend possible l'utilisation des 4 autres attributs userPassword, loginShell, gecos et description.

2.2.9.3. Les schémas

Comment savoir quels sont les objectClass disponibles et quels attributs ils contiennent. C'est très simple, la syntaxe et la liste des attributs connus de l'annuaire sont écrits dans ce que l'on appelle les "schémas". Un annuaire LDAP a la capacité de charger en mémoire plusieurs schémas. A travers ces schémas, il est possible de définir de nouveaux attributs et de

nouveaux objectClass. Cette souplesse permet de définir très finement ce qui sera stocké dans notre annuaire.

Concrètement, un schéma est un fichier qui décrit un à un les attributs disponibles (leur nom, leur type, etc...), ainsi que les objectClass qui y font appel. Au démarrage du serveur LDAP, le ou les fichiers de schéma spécifiés dans sa configuration seront chargés.

Dans notre exemple, l'objectClass posixAccount est défini dans le fichier **nis.schema**. Etudions une partie de ce fichier, livré avec OpenLDAP et situé dans **/etc/ldap/schema** :

```
# [...]
attributetype ( 1.3.6.1.1.1.1.0 NAME 'uidNumber'
        DESC 'An integer uniquely identifying a user in a domain'
        EQUALITY integerMatch
        SYNTAX 1.3.6.1.4.1.1466.115.121.1.27 SINGLE-VALUE )

# [...]
objectclass ( 1.3.6.1.1.1.2.0 NAME 'posixAccount' SUP top AUXILIARY
        DESC 'Abstraction of an account with POSIX attributes'
        MUST ( cn $ uid $ uidNumber $ gidNumber $ homeDirectory )
        MAY ( userPassword $ loginShell $ gecos $ description ) )
```

Figure 6: Attribut utilise par posixAccount et le objectClass

Le fichier est assez volumineux et a été tronqué.

Le premier paragraphe définit l'un des attributs utilisés par le posixAccount : uidNumber. Le second, l'objectClass posixAccount. Nous n'allons pas étudier en détail ces deux définitions, simplement, sachez que :

• A chaque définition correspond un OID (Object IDentifier), qui permet de rendre unique l'attribut spécifié. Ces OIDs sont déposés auprès de l'IANA et sont donc officiels.

• Un attribut définit un type d'égalité à mettre en œuvre lors d'une recherche (ici, intergerMatch) ainsi que le type de données qu'il contient (l'OID spécifié après SYNTAX).

• Un objectClass définit les attributs que l'objet doit présenter (MUST) et ceux qu'il peut posséder (MAY).

Les schémas constituent donc une source d'information très importante. En cas de doute concernant le type ou le nom des attributs à spécifier dans une entrée, n'hésitez pas à vous y reporter. Enfin, sachez qu'il est tout à fait possible de créer ses propres schémas, cependant, il faut penser à réutiliser les schémas existants .

2.2.9.4. Le format LDIF

Les données contenues dans l'annuaire sont présentées dans un certain format : il s'agit du format LDIF (LDAP Data Interchange Format - RFC 2849). Nous en avons vu un exemple dans le paragraphe précédent.

Sachez que toute interaction avec un annuaire se fait par le biais de ce format : l'ajout, la modification, la suppression d'entrées, l'interrogation de l'annuaire y compris.

Dans ce format, chaque entrée constitue un paragraphe, et, au sein de chaque paragraphe, chaque ligne constitue un attribut. Voici un exemple un peu plus complet, incluant le groupe de notre utilisateur :[04]

```
# [...]
dn: cn=utilisateurs,ou=groups,dc=martymac,dc=com
objectClass: posixGroup
cn: utilisateurs
gidNumber: 10001

dn: uid=martymac,ou=users,dc=martymac,dc=com
objectClass: account
objectClass: posixAccount
cn: martymac
uid: martymac
uidNumber: 10001
gidNumber: 10001
homeDirectory: /home/martymac
userPassword:: e0NSWVBUfWJjT29IUk5SbG1HbC4=
loginShell: /bin/sh
gecos: martymac
description: martymac
# [...]
```

Figure 7: Entrée incluant le groupe avec le format LDIF

2.3. Plateforme

Une plate-forme est en informatique une base de travail à partir de laquelle on peut écrire, lire, développer et utiliser un ensemble de logiciels.

Les plates-formes informatiques sont généralement conçues, développées, construites, mises en service et maintenues par des constructeurs informatiques, ou des prestataires de services. Dans le cas des plates-formes logicielles, elles sont plutôt maintenues par les organismes (par exemple l'INRIA, le CNRS, le CEA, l'INRA) qui hébergent la base de travail et les logiciels associés.

Lorsqu'on parle de plate-forme web, il peut s'agir du logiciel serveur web, de ce même logiciel avec son système d'exploitation sous-jacent, du logiciel serveur web avec son système d'exploitation et son matériel, d'un ensemble de machines avec serveur web, ou encore d'un même ensemble en tenant compte des infrastructures réseau et connectivité à Internet.

2.3.1. Zimbra

2.3.1.1. Introduction

Le courrier électronique est l'une des applications les plus indispensables à la viabilité d'une petite ou moyenne entreprise. Outil de communication à son origine, il est devenu la solution d'archivage de fait pour les données métier dans de nombreuses entreprises, et sert de plate-forme quasi universelle de messagerie, coordination et collaboration. Son évolution va se poursuivre :

• *Fonctionnalités* : les attentes des utilisateurs en termes d'expérience homogène et facile s'avèrent élevées, et les préférences d'interface utilisateur fortes

• *Plates-formes* : à l'origine installé sur des serveurs d'entreprise, le courrier électronique fut l'une des premières applications à adopter la virtualisation , les périphériques mobiles et le Cloud

Les attentes croissantes des utilisateurs se heurtent constamment aux barrières économiques du coût de la solution et des tâches d'administration. Alors que les contrats de support arrivent à expiration pour certaines solutions de messagerie répandues, de nombreuses entreprises recherchent des alternatives.

2.3.1.2. Définition de Zimbra

Zimbra est un logiciel serveur collaboratif (ou groupware) qui permet à ses utilisateurs de stocker, organiser et partager rendez-vous, contacts, courriels, liens, documents et plus.

Zimbra offre une solution éprouvée dans des environnements de production, un choix d'options de déploiement en local avec gestion intégrée du stockage hiérarchique ou un hébergement par l'un des nombreux partenaires fournisseurs de services VMware vCloud. Zimbra propose également une appliance virtuelle logicielle reposant sur VMware vSphere, qui se déploie en moins de 10 minutes, dotée d'une interface d'administration simplifiée et conjuguant l'application et le système d'exploitation dans une seule procédure de gestion du

cycle de vie pour réduire les tâches de maintenance. L'appliance virtuelle de collaboration Zimbra utilise la plate-forme vSphere pour assurer une haute disponibilité, une sauvegarde et une reprise d'activité intégrées dans une véritable solution métier.
La messagerie Zimbra est conçue pour fonctionner de manière optimale sur les principales plates-formes informatiques (matériel, OS, virtualisation et Cloud), et avec des applications intégrées et des services Web hébergés . Sécurité, extensibilité, évolutivité et pérennité : Zimbra est une solution ouverte et simple à gérer à laquelle font confiance des millions d'utilisateurs, d'entreprises et de prestataires de services dans le monde entier. S'adossant aux ressources et l'assistance de VMware et ses partenaires, Zimbra incarne un choix sûr.

VMware Zimbra est un leader des logiciels de messagerie et de collaboration open source de nouvelle génération. Zimbra simplifie l'informatique et s'établit en référence de la collaboration sur le Web et le Cloud avec une expérience utilisateur novatrice et évolutive, intégrant une interface Web AJAX enrichie. Administration simplifiée, mobilité avancée et options de déploiement en local ou en hébergement sur le Cloud : Zimbra est une plate-forme de collaboration privilégiée pour les entreprises, les prestataires de services, les services publics et le monde de l'enseignement. Zimbra est l'un des principaux fournisseurs de messagerie, en croissance rapide.***[02]***

2.3.1.3. Zimbra Composants

L'architecture Zimbra inclut des intégrations open-source à l'aide standard de l'industrie protocoles. Le logiciel tiers énuméré ci-dessous est fourni avec Zimbra logiciels et installé dans le cadre du processus d'installation. Ces composant sont été testés et configurées pour fonctionner avec le logiciel.

- Jetty, le serveur d'applications Web que le logiciel Zimbra doit fonctionner
- Postfix, un agent de transfert de courrier open source (MTA) qui achemine le courrier messages vers le serveur Zimbra approprié
- logiciel Open LDAP, une implémentation open source du Lightweight

Directory Access Protocol (LDAP) qui stocke la configuration système Zimbra, la liste d'adresses globale Zimbra, et les fournisseurs d'authentification de l'utilisateur. Zimbra

peuvent également travailler avec les services de GAL et authentification fournies par externe Annuaires LDAP comme Active Directory

- la base de données du logiciel MYSQL
- Lucene, une open source texte complet et moteur de recherche
- Anti-virus et des composants open source anti-spam, contient:

• ClamAV, un scanner anti-virus qui protège contre les fichiers malveillants

• SpamAssassin, un filtre de messagerie qui tente d'identifier le spam

•James / Sieve de filtrage, utilisé pour créer des filtres pour le courrier électronique

2.3.1.4. Architecture du système

La conception architecturale ZCS est affichée dans la Figure de l'architecture ZCS serveur de collaboration. Cela montre le logiciel open-source livré avec le ZCS et d'autres applications tierces recommandées.

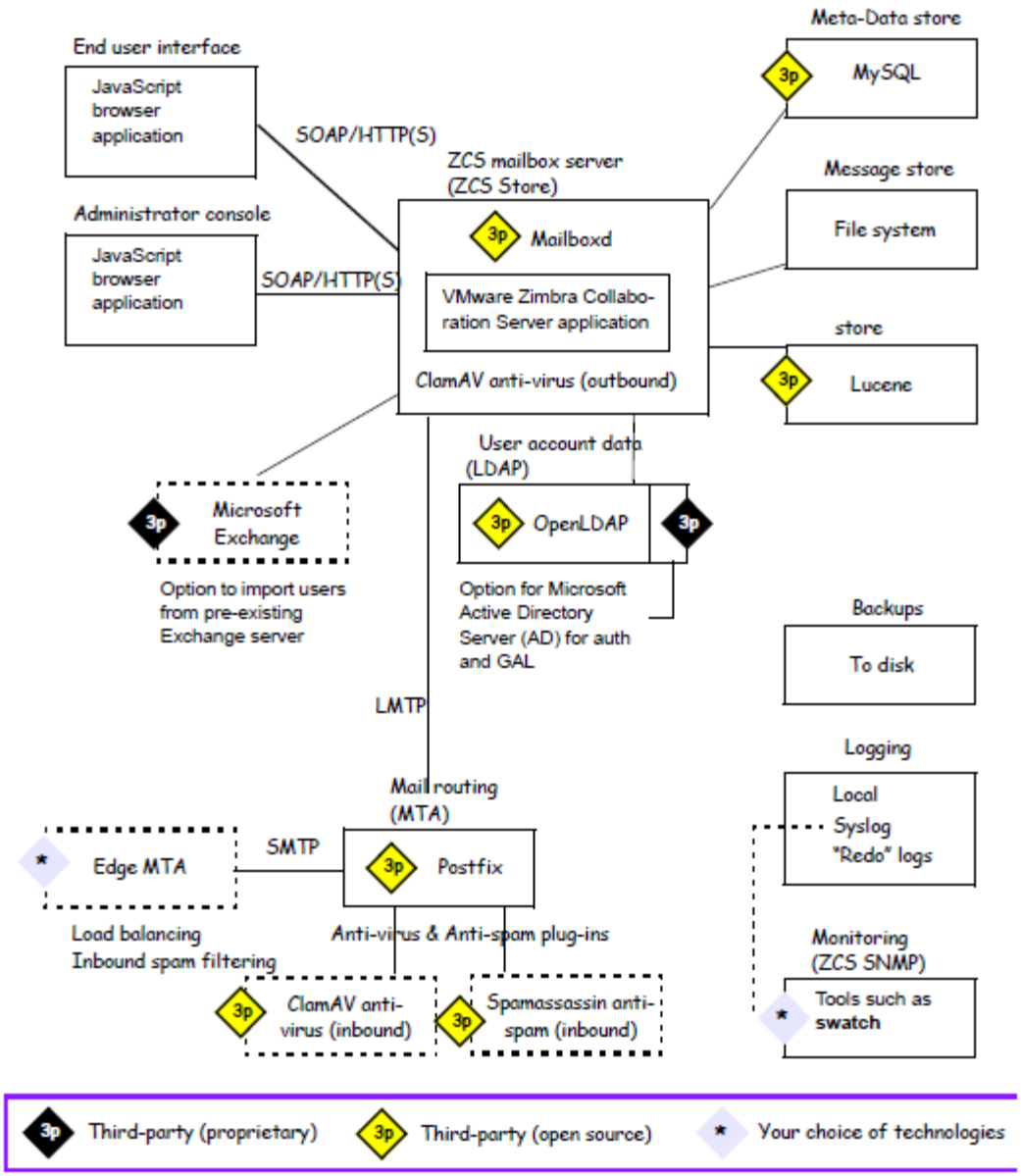


Figure 8: Architecture collaborative du serveur ZCS

2.3.1.5. Le packages d'applications Zimbra

Les paquets d'application inclut a ZCS :

2.3.1.5.1. Zimbra Core(base) : Comprend les bibliothèques, utilitaires, outils de suivi et les fichiers de configuration de base *.zmconfigd* fait partie de Zimbra-core et est automatiquement activée et fonctionne sur tous les systèmes.

2.3.1.5.2. Zimbra LDAP : ZCS utilise le logiciel OpenLDAP, une open source du serveur d'annuaire LDAP. L'authentification des utilisateurs, la liste d'adresses globale Zimbra et les attributs de configuration sont les services fournis par OpenLDAP. Notez que le GAL Zimbra et services d'authentification peuvent être fournis par un annuaire LDAP externe comme Active Directory.

2.3.1.5.3. Zimbra MTA : Postfix est l'agent de transfert de courrier open source (MTA) qui reçoit l'email via SMTP et routes chaque message au serveur de messagerie Zimbra approprié à l'aide locale Mail Transfer Protocol (LMTP).Le MTA Zimbra inclut également l'anti-virus et anti-spam composants.

2.3.1.6. Stockage de Zimbra (serveur de messagerie)

Le paquet du stockage Zimbra installe les composants du serveur de boîte aux lettres, y compris Jetty, qui est le conteneur de servlets le logiciel Zimbra gère l'intérieur. Dans ZCS, cette conteneur de servlet est appelée mailboxd. Chaque compte est configuré sur un serveur de boîte aux lettres, et ce compte est associé à une boîte aux lettres qui contient tous les messages électroniques, les pièces jointes, contacts, agenda et les fichiers de collaboration pour ce compte de messagerie.

Chaque serveur Zimbra a son propre stockage autonome de données, stockage de messages, et un stockage d'index pour les boîtes aux lettres sur ce serveur.

Comme chaque e-mail arrive, les horaires de serveur Zimbra un fil pour que le message soit indexé (Index stockage).

Zimbra Zimbra-SNMP : utilise échantillon de regarder la sortie de syslog pour générer des interruptions SNMP.

Zimbra-Logger : Le Zimbra logger installe des outils d'agrégation de syslog, rapports. Si l'enregistreur n'est pas installé, la section des statistiques du serveur de la console d'administration n'est pas affichée.

Zimbra-Spell Aspell : est le correcteur orthographique open source utilisé sur le Zimbra Web Client. Quand Zimbra-sort est installé, le Paquet Zimbra-Apache est également installé.

Zimbra-Proxy : L'utilisation d'un serveur proxy IMAP / POP permet la récupération de courrier pour un domaine à être divisée entre plusieurs serveurs Zimbra sur une base d'utilisateur.

Le paquet de Proxy Zimbra peut être installé avec le LDAP Zimbra , le Zimbra MTA, le serveur de messagerie Zimbra, ou sur son propre serveur.

Zimbra-Memcached est un paquet séparé de zimbra_proxy et est automatiquement sélectionné quand le paquet Zimbra-proxy est installé. Un serveur doit exécuter zimbra_memcached lorsque le proxy est en cours d'utilisation. Tous zimbraproxies installés peuvent utiliser un seul serveur de cache.

2.3.1.7. Service de LDAP Zimbra

Services d'annuaire LDAP fournissent un référentiel centralisé d'informations sur les utilisateurs et les périphériques qui sont autorisés à utiliser votre service Zimbra. Le référentiel central utilisé pour les données LDAP de Zimbra est le serveur d'annuaire OpenLDAP.

Le serveur LDAP est installé lorsque ZCS est installé. Chaque serveur possède sa propre entrée LDAP qui comprend des attributs spécifiant les paramètres de fonctionnement. En outre, un objet de configuration globale définit par défaut pour tous les serveurs dont l'entrée ne précise pas tous les attributs.

Un sous-ensemble de ces attributs peut être modifié via la console d'administration Zimbra et d'autres via l'utilitaire zmprov CLI.

2.3.1.8. La circulation du trafic LDAP

La figure du trafic d'annuaire LDAP montre le trafic entre le Zimbra-LDAP serveur d'annuaire et les autres serveurs de la Collaboration VMware Zimbra Système serveur. La MTA et Zimbra Collaboration VMware Zimbra Serveur de boîte aux lettres du serveur lue ou écrire à la base de données LDAP sur le serveur d'annuaire.

Les clients Zimbra se connecter via le serveur Zimbra, qui se connecte à LDAP.

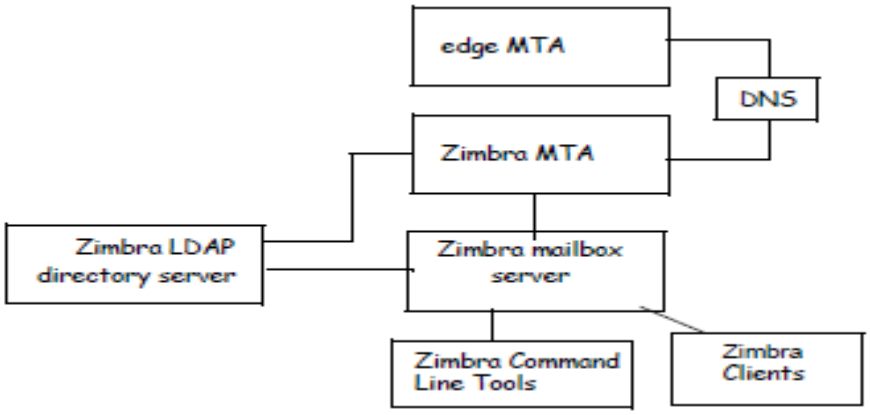


Figure 9: Trafic d'annuaire LDAP

2.3.1.9. Hiérarchie de l'annuaire LDAP

Annuaires LDAP sont disposés en une structure hiérarchique arborescente avec deux types de branches, les branches de messagerie et la branche config. Branches de messagerie sont organisés par domaine. Entrées appartiennent à un domaine, tels que les comptes, des groupes, des alias, sont provisionnés dans le domaine DN dans le répertoire. La branche config contient des entrées de système d'administration qui ne font pas partie d'un domaine. Entrées de la branche config comprennent les comptes du système d'administration, de configuration globale, les subventions globales, COS, les serveurs, les types MIME et Zimlets.

Le chiffre de la hiérarchie LDAP Zimbra montre la hiérarchie LDAP Zimbra. Chaque type d'entrée (objet) a certaines classes d'objets associés.