Serveis de directori

Antoni Martínez-Ballesté

PID_00177497

Universitat Oberta de Catalunya

Índex

Introducció

Un dels propòsits dels sistemes informàtics és guardar informació. Els sistemes gestors de bases de dades ofereixen eines per a guardar quantitats ingents d'informació. Aquesta informació pot arribar a tenir estructures realment complexes, amb moltes entitats i relacions.

Durant la implantació dels sistemes informàtics, ha calgut desenvolupar eines centrades a gestionar un tipus molt concret d'informació. Un exemple clàssic d'això és la informació referent al sistema de fitxers d'una unitat d'emmagatzematge. Es tracta d'un concepte molt específic, amb la informació que s'ha de guardar clarament definida. Ara bé, cal que la implementació sigui robusta i escalable, ja que un sistema de fitxers fàcilment en pot arribar a contenir milions.

Els directoris són un tipus específic de bases de dades amb un propòsit també específic: emmagatzemar la informació sobre un objecte (individu, recurs de xarxa, document). El paper que tenen és clau en qualsevol organització que vulgui tenir la informació sobre els seus treballadors, usuaris de xarxa, etc., catalogada i accessible des de moltes aplicacions. A més, l'ús d'un servei de directori facilita la gestió de la identitat dels usuaris dels sistemes d'informació en una organització.

En aquest mòdul estudiarem els conceptes bàsics dels directoris, el disseny que tenen i la implantació. Ens centrarem en LDAP (protocol d'accés a directoris lleugers o *lightweight directory access protocol*), ja que és l'estàndard més usat des de ja fa uns quants anys. N'estudiarem el concepte, la utilitat, el disseny i la implantació.

Objectius

Els objectius que haurà assolit l'estudiant en acabar aquest mòdul són els següents:

1. Comprendre el concepte i la utilitat d'un servei de directori.
2. Entendre el concepte d'*espai de noms*.
3. Saber les operacions que ofereix un servei de directori com LDAP.
4. Dissenyar un espai de noms per a un servei de directori.
5. Fer el disseny de l'esquema d'un servei de directori, manejant els conceptes d'*atribut* i *classe*.
6. Comprendre quins aspectes incideixen en la seguretat, l'eficiència i la disponibilitat en un servei de directori.
7. Conèixer implantacions de serveis de directori.

1. Concepte i ús dels directoris

La idea de directori està relacionada amb la informació. Des dels inicis dels sistemes de fitxers fins a arribar als primers sistemes operatius per a ordinadors PC, la paraula *directori* s'ha associat a l'esquema amb què estan organitzats els arxius a les unitats d'emmagatzematge. En anglès, *telephone directory* és el nom amb què es coneix la guia telefònica: la relació d'abonats al servei telefònic, els seus números d'abonat i opcionalment l'adreça física del seu domicili. Sigui com sigui, el concepte de *directori* està estretament lligat a l'organització de dades.

Un **directori** és una estructura jeràrquica que organitza i emmagatzema dades sobre elements. És un tipus concret de base de dades.

Així, doncs, els fitxers, malgrat que realment estan guardats en el suport sense organització aparent, s'organitzen de manera lògica en directoris i subdirectoris. En un directori de sistema informàtic en xarxa, la informació que es guarda té a veure amb els recursos del sistema (servidors, impressores, etc.) i amb els usuaris d'aquest sistema. A més, cal que el sistema tingui un servidor per a consultar la informació, emmagatzemar-la o simplement dissenyar-la. Aquesta informació, com veurem més endavant, pot estar emmagatzemada de forma distribuïda o replicada. En conseqüència, s'ha de tenir un servei de directori.

Un **servei de directori** és una plataforma que proporciona mètodes per a gestionar i emmagatzemar les dades que conté el directori.

Un servei de directori permet la cerca de valors a partir d'un determinat nom (o identificador), de manera similar com ho fa un diccionari. De la mateixa manera que un vocable té diferents accepcions, hi ha diferents vocables apuntant a un mateix significat, hi ha paraules derivades, famílies de paraules, etc., els objectes que emmagatzema un directori poden estar relacionats de moltes maneres.

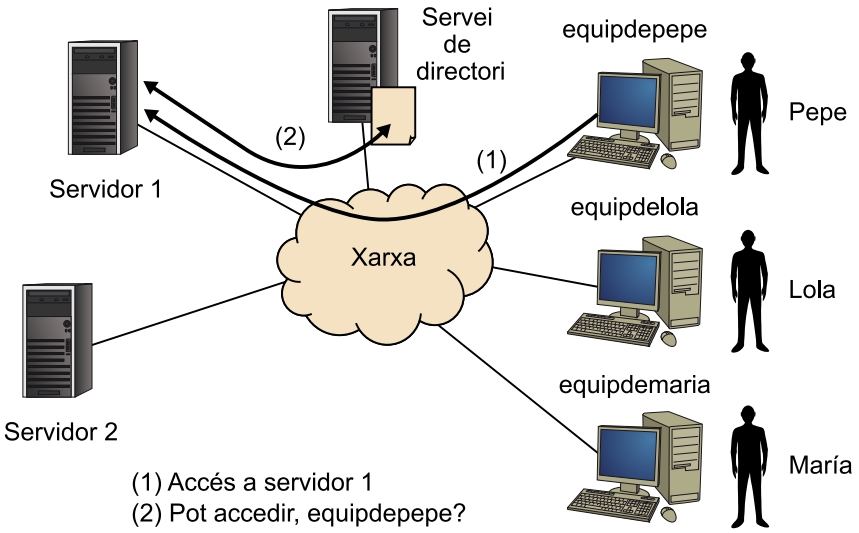
Exemple d'un servei de directori

Imaginem-nos un servei que, dins d'un entorn de xarxa local, retorna una adreça IP i una sèrie de característiques a partir d'un nom d'equip. També podem considerar que hi ha noms que defineixen grups d'equips sobre els quals s'han d'aplicar determinats permisos. Doncs bé, una aplicació que guardi i permeti manejar aquesta informació és un servei de directori. Qualsevol aplicació pot obtenir informació sobre els equips i els grups; només cal accedir al servei de directori.

En la figura hi ha un servei de directori que controla els equips i els recursos d'una xarxa. Quan *Servidor1* rep una petició des d'*equipdepepe*, el recurs sol·licita al servei de directori si aquest equip té permís per a accedir al recurs. Dins del servei de directori, s'ha creat

un espai de noms, és a dir, una concreció de quin és l'identificador de cada recurs de la xarxa. Més endavant estudiarem a fons aquest concepte.

Exemple de consulta a un servei de directori per a accés a recursos



La idea d'un servei de directori és tan recent (o tan antiga, segons com es miri) com l'inici dels sistemes basats en xarxa. El 1988, l'ITU[1] i l'ISO[2] es van unir per dur a terme el desenvolupament d'una sèrie d'estàndards sobre serveis de directori, conegut com a *X.500* o *protocol d'accés a directoris* o *directory access protocol* (DAP). Aquest sistema era complex i, a més, implementava moltes eines que molts clients no van acabar usant mai. Això no propiciava un ampli ús d'X.500, de manera que es va començar a treballar en una versió "lleugera", això és, LDAP (protocol d'accés a directoris lleugers o *lightweigth directory access protocol*), el primer esbós del qual es va publicar a l'RFC 1487. Aquesta primera versió, dissenyada per personal de la Universitat de Michigan, s'utilitzava com a passarel·la. LDAP connectava clients de directori amb serveis X.500. En vista de l'acceptació d'LDAP, es va treballar perquè aquest sistema fos realment el servei de directori i no una mera passarel·la. Així, doncs, LDAP es va convertir en un estàndard *de facto*, sobretot perquè va comportar la base per a diferents productes de servei de directori que gaudeixen de gran popularitat. En el decurs d'aquest mòdul, farem referència en general a LDAP.

(1)*ITU* és la sigla d'*Unió Internacional de Telecomunicacions*.

(2)*ISO* és la sigla d'*Organització Internacional per a la Normalització*.

Document RFC

Els documents de petició de comentaris o *request for comments* (RFC) són documents estàndard o propostes d'estàndard usats per la Internet Engineering Task Force (IETF).

Encara que el concepte de *directori* es relacioni amb dades, és ben cert que hi ha diverses diferències entre un servei de directori i una base de dades:

- En els directoris es fan moltes més lectures de dades que escriptures.
- Els directoris poden modificar més fàcilment el disseny de les entitats que contenen. En canvi, en una base de dades, canviar el disseny d'aquesta base *a posteriori* pot ser més complex.
- Les dades dels directoris solen estar distribuïdes i replicades amb més freqüència que en les bases de dades.

- Els directoris permeten, en general, consultes simples, i no consultes que requereixin la fusió de dades provinents de diverses taules (consultes *join* de les bases de dades).

Vegeu també

En el subapartat següent aprofundirem en alguns d'aquests aspectes mitjançant exemples.

1.1. Exemples i tipus de directoris

Abans de passar a estudiar conceptes concrets sobre els serveis de directori, veurem una panoràmica de diferents tipus de directoris depenent del propòsit o de la implantació que tenen.

Un tipus senzill de directoris és el que està inclòs en **aplicacions de programari**, com per exemple les llibretes d'adreces. Una aplicació informàtica de correu electrònic pot incloure un directori en què cada entrada és un contacte, i entre la informació emmagatzemada hi ha, és clar, l'adreça de correu electrònic del contacte.

Un pas més enllà és que aquesta aplicació de llibreta d'adreces funcionés com una aplicació informàtica independent, o potser fos un element més del sistema operatiu. En aquest cas, caldria establir un **estàndard d'intercanvi d'informació** perquè els altres programes poguessin fer ús d'aquesta llibreta d'adreces.

Un exemple d'aquests sistemes és l'LDIF (format d'intercanvi de dades en LDAP o *LDAP data interchange format*), que és utilitzat com a mitjà habitual d'exportació de dades de llibreta d'adreces a un fitxer de text imprimible.

Aquesta aplicació podria ser una **aplicació de xarxa** executant-se en un servidor. D'aquesta manera, la informació dels contactes estaria disponible per a tots els equips clients que fessin una consulta al servidor. En aquest cas, s'hauria d'establir un protocol de comunicacions en l'àmbit d'aplicació per a fer diferents operacions:

- Consultes sobre la informació d'un contacte.
- Missatge d'error en cas que no es trobi el contacte.
- Opcionalment, un protocol d'identificació de l'usuari.
- Operacions d'alta, baixa i modificació de contactes només executables per un usuari amb permisos d'administrador.

Els **directoris de sistemes operatius en xarxa** emmagatzemen dades de recursos d'una xarxa. Alguns exemples d'aquests directoris són l'Active Directory de Microsoft o l'eDirectory de Novell. De la mateixa manera que una llibreta d'adreces pot estar integrada en una aplicació, executant-se en un sistema o funcionant en un servidor, és clar que serà una aplicació concreta: guardar informació sobre contactes. En el cas dels directoris de sistemes operatius en xarxa, l'ús que se'n fa és més ampli.

Sistema de noms per a Internet

Un altre exemple de directori de propòsit específic és el sistema de noms per a Internet, DNS[3]. L'accés a serveis basats en Internet es fa per connexions o enviant datagrames cap a una determinada adreça IP. El sistema DNS resol, a partir d'un nom, quina és l'adreça IP del recurs. La particularitat d'aquest sistema és que la informació està distribuïda per tota la xarxa Internet. Per exemple, depenent del TLD[4] (és a dir, si el nom de domini es correspon amb un *.com*, amb un *.es*, etc.), les dades són en un servidor o un altre. Les dades corresponents als noms locals –en general noms que identifiquen serveis o màquines dins d'un determinat domini– es troben en servidors de noms autoritzats. Finalment, hi ha la replicació de dades, un dels exemples de la qual són els servidors de nom arrel, els que contenen informació sobre on es troben els servidors de TLD, els quals, al seu torn, també estan replicats per temes d'eficiència.

(3)*DNS* és la sigla de *servidor de noms de domini* o *domain name service*.

(4)*TLD* és la sigla de *domini de nivell superior* o *top level domain*.

Número d'IP

L'adreça IP és una adreça d'Internet o número únic que identifica una màquina connectada a Internet; per exemple, 85.34.123.170.

Així com DNS es pot veure com un servei de directori una mica específic, n'hi ha de **propòsit general**. Aquest és el cas del servei LDAP. Encara que en certs casos l'ús que se'n fa es remet a tenir informació sobre els usuaris d'una sèrie de serveis en xarxa (permetent, per exemple, l'accés a molts serveis mitjançant un únic usuari i contrasenya), LDAP permet definir solucions per a un ampli espectre d'escenaris.

1.1.1. Directoris i ús d'aquests directoris en seguretat i autenticació

Una aplicació interessant dels serveis de directori és la seguretat. D'una banda, un servei de directori es pot utilitzar, com hem apuntat més amunt, per a gestionar la informació dels usuaris de diversos serveis de xarxa. En concret, es pot pensar en un servei de directori per a una organització que contingui, per a cada usuari d'un sistema, la informació següent:

- Nom i cognoms
- Departament de l'organització
- Identificador d'usuari
- Contrasenya
- Data de l'últim canvi de contrasenya

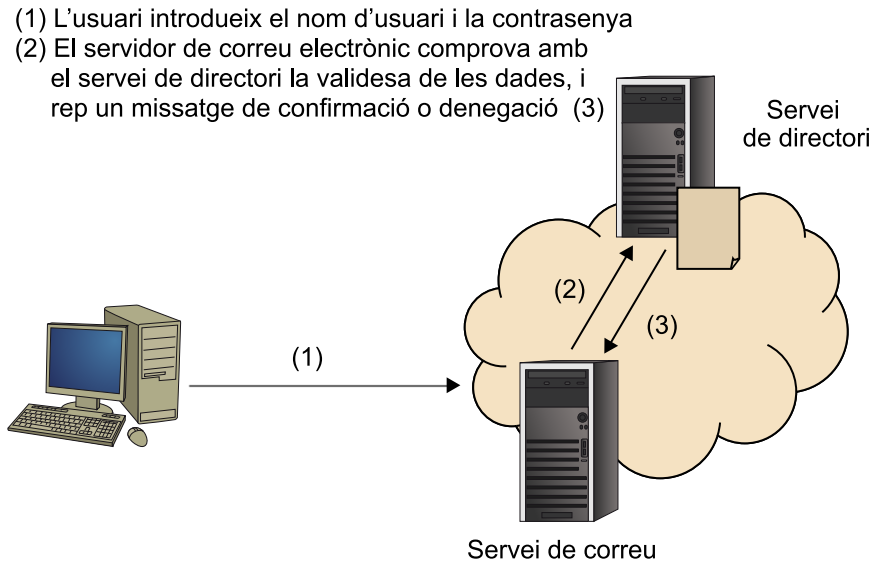
En l'organització hi hauria diversos serveis en xarxa: un correu electrònic, un calendari, un sistema de compartició de documents i una aplicació web de gestió de nòmines. Amb un únic identificador d'usuari i la contrasenya corresponent, els usuaris de l'organització es podrien autenticar davant qualsevol dels serveis. En realitat, el servidor de correu electrònic fa una consulta al servei de directori per comprovar que hi hagi l'identificador d'usuari i que, evidentment, la contrasenya introduïda sigui la correcta.

Si en qualsevol moment l'usuari vol canviar la contrasenya, ha de fer aquesta operació mitjançant el servei de directori. Al seu torn, aquest servei podria avisar l'usuari, per exemple, amb un correu electrònic, de la pròxima caducitat de la contrasenya. O bé si un usuari té caducada la contrasenya, el mateix servidor de correu electrònic mostraria un avís d'impossibilitat d'autenticar l'usuari perquè ha caducat la contrasenya.

Finalment, pel que fa a l'aplicació web de gestió de nòmines, quan un usuari hi accedeix, es consulta el servei de directori del departament a què pertany. Si l'usuari no pertany al departament, per exemple, de gestió, només pot consultar les seves nòmines. En cas contrari, pot fer consultes sobre les seves nòmines i les dels altres treballadors, i també crear i calcular noves nòmines.

En la figura es mostra aquest exemple d'ús del servei de directori, en concret, l'autenticació al servidor de correu de l'organització.

Ús del servei de directori com a suport en l'autenticació d'un usuari davant un servei de correu electrònic



Un altre dels sistemes que es basen en un directori és la infraestructura de clau pública o *public-key infrastructure* (PKI). Aquestes plataformes fan no solament la distribució de certificats i claus a clients i servidors, sinó que també s'encarreguen d'altres funcions, de les quals una de les més importants és la revocació de claus i certificats.

Els directoris ajuden a resoldre dos dels problemes que sol implicar la implantació d'una PKI. L'un és la gestió de cicle de vida d'un certificat, és a dir, com es creen, mantenen i destrueixen els certificats. L'altre fa referència a la localització de certificats, és a dir, com es pot trobar en confiança les claus públiques i els certificats necessaris per a comunicar-se amb serveis i individus.

El fet d'usar un directori pot donar solució, clarament, a aquests problemes. El servei de directori actua de punt central d'administració durant el cicle de vida de la PKI. La creació de certificats i claus es fa mitjançant el mateix servei de directori. Quan cal revocar un certificat, el directori proporciona l'eina adequada: el directori ofereix informació sobre la llista de certificats revocats (i que ja no són vàlids). Les propietats de redundància i distribució de la informació són igualment necessàries en entorns de PKI, i són proporcionades clarament pels serveis de directori.

Una vegada vistos diferents exemples de serveis de directori i algunes de les característiques d'aquests serveis, ens centrarem en la identificació d'objectes i les operacions bàsiques que utilitzen els clients del servei de directoris.

1.2. Espai de noms

En els serveis de directori, cada objecte està identificat mitjançant un nom. Podem definir l'espai de noms d'un directori com el conjunt d'identificadors que s'utilitzen, o es poden utilitzar potencialment, per a identificar de manera unívoca els objectes del directori.

L'identificador d'objecte ha de ser un nom únic dins del servei de directori.

A més d'identificar objectes, els identificadors també poden identificar grups d'objectes, de manera que es pot dissenyar una estructura jeràrquica. Vegem a continuació alguns exemples d'espais de noms.

1.2.1. Sistema DNS

En un sistema DNS, l'espai de noms permet identificar unívocament un equip connectat a Internet. De fet, diversos noms poden apuntar a un mateix equip. Per a identificar un equip dins de l'espai de noms, el DNS s'ajuda d'una estructuració jeràrquica iniciada en el domini (.), que és el domini arrel.

A partir d'aquest domini, s'estableixen una sèrie de dominis de nivell superior, els TLD apuntats més amunt. S'estableixen TLD geogràfics (*.es*, *.fr*, *.pt*, *.uk*, etc.) i genèrics (*.com*, *.net*, etc.). Hi ha una sèrie d'entitats acreditades per a gestionar els noms pertanyents a cadascun dels TLD. Per exemple, hi ha una entitat encarregada de gestionar els dominis *.cat*. Quan una organització vol tenir visibilitat pública a Internet, sol sol·licitar un domini a l'entitat corresponent a la seva TLD (tret que per a visitar els seus serveis s'hagi d'usar una adreça IP, que, com que és una sèrie de números, pot ser difícil de recordar).

A més, l'organització pot tenir la potestat de contenir un servidor de DNS dins dels seus sistemes i definir internament els noms de domini perquè es pugui accedir als seus serveis interns des de l'exterior. En aquest sentit, es poden definir noms per a diferents nivells de domini, fins a arribar a identificar un servei o equip.

Exemple

Imaginem-nos que una organització universitària com la Universitat Oberta de Catalunya (UOC) sol·licita un nom de domini (*uoc.edu*). Aquesta organització podrà decidir establir una jerarquia de serveis Internet, mitjançant els diferents nivells del sistema de noms. Per exemple, imaginem-nos que vol utilitzar tres campus virtuals diferents segons els estudis que contingui. Els dominis poden ser els següents:

- Domini dels Estudis d'Enginyeria: *enginyeria.uoc.edu*.

- Domini dels Estudis de Ciències: *ciencies.uoc.edu*.
- Domini dels Estudis d'Humanitats: *humanitats.uoc.edu*.

A més, es podria pensar que en els Estudis d'Informàtica hi ha un servei web que conté les pàgines sobre docència, un altre que conté les pàgines corresponents a grups d'investigació i les seves publicacions, i un últim web amb informació d'administració d'estudis (expedient, matrícules, etc.). En aquest cas, es podrien definir els dominis següents:

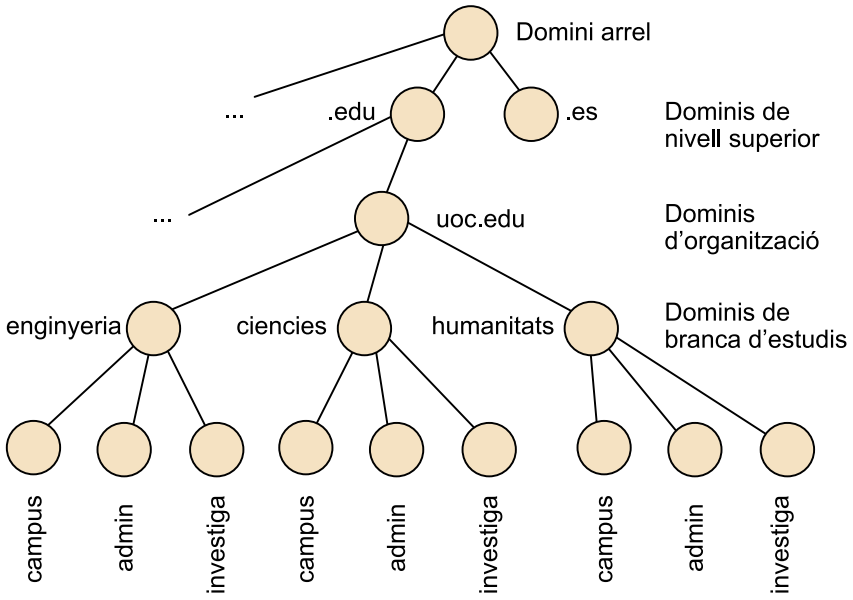
- *campus.enginyeria.uoc.edu*
- *investiga.enginyeria.uoc.edu*
- *admin.enginyeria.uoc.edu*

El resultat seria la identificació unívoca de diferents serveis, fins i tot coincidint els noms en algun dels nivells:

- *campus.enginyeria.uoc.edu* és diferent de *campus.ciencies.uoc.edu*.
- *campus.enginyeria.uoc.edu* és diferent de *investiga.enginyeria.uoc.edu*.

La figura mostra la jerarquia definida amb aquests nivells de DNS.

Exemple de jerarquia d'un sistema DNS



Encara que el DNS va ser creat amb el propòsit que els equips siguin accessibles des d'Internet, qualsevol equip d'una xarxa sense adreça de xarxa pública, o bé que els sistemes de connexió a la xarxa no permetin la connexió des de l'exterior d'aquesta xarxa, pot disposar d'un nom que l'identifiqui dins d'un espai de noms. Mitjançant aquest nom es poden establir connexions internes entre equips de la mateixa xarxa.

1.2.2. Sistema WINS

El sistema WINS (servei de noms d'Internet de Windows o *Windows Internet name service*) és, tal com indica el nom, un equivalent del que ofereix DNS però pensat per a equips Windows. Actualment, els sistemes operatius de Microsoft permeten la convivència de tots dos serveis de noms, de manera que és possible identificar un equip mitjançant un nom dins d'un espai DNS i també un nom WINS. Aplicacions com la compartició de recursos mitjançant Windows es resolen amb el nom WINS.

L'espai de noms WINS va ser dissenyat per a donar suport al sistema NetBIOS[5]. Aquest sistema proporcionava connectivitat de xarxa a mitjan anys vuitanta en els ordinadors PC. Aquest sistema va ser heretat i reimplementat per Microsoft, que el va arribar a incloure com a estàndard en les seves primeres implementacions de Windows, de manera que va permetre el treball en xarxa. Per tant, WINS es considera un sistema de resolució de noms per a les aplicacions que usen NetBIOS.

(5)*NetBIOS* significa *sistema bàsic d'entrada/sortida en xarxa* o *network basic input/output system.*

Aquests noms tenen una longitud de quinze caràcters ASCII per a identificar l'equip. Els sistemes Windows poden prescindir d'aquest servei de noms, ja que, per a resoldre els noms NetBIOS en xarxes petites, n'hi ha prou de fer peticions de difusió o *broadcast*. És a dir, que es difonen per tota la xarxa. A més, per a evitar la generació d'aquest trànsit de xarxa, es pot usar un fitxer (lmhosts) com a servei de directoris per a resoldre noms.

Malgrat que NetBIOS i WINS van gaudir de gran popularitat en els sistemes operatius i xarxes basades en productes Microsoft, l'auge d'Internet va fer abandonar aquests sistemes i passar a especificar noms en el sistema DNS.

1.2.3. Sistema LDAP

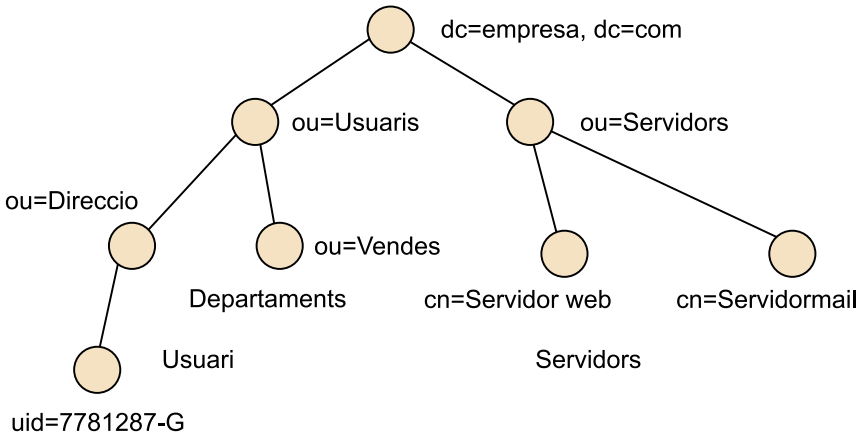
El sistema LDAP, amb les diferents versions que té, s'ha convertit en un estàndard genèric de servei de directori. Tant és així que molts dels productes àmpliament usats per a implantar serveis de directori es basen en LDAP. Per aquest motiu, ara ens centrarem en la manera com és l'espai de noms que es defineix en LDAP. Aquest sistema hereta moltes característiques dels sistemes X.500.

El sistema de noms que usa LDAP permet l'organització dels objectes de manera jeràrquica.

Exemple de directori LDAP

En la figura s'observa un exemple de directori LDAP.

Exemple d'espai de noms d'un directori basat en LDAP



En aquesta organització (*empresa.com*) hi ha definits dos grups d'objectes: els usuaris i els servidors. Els usuaris es divideixen en departaments. Com veurem més endavant, LDAP i

Lectura recomanada

L'RFC 2256 dóna més detalls sobre els atributs.

Vegeu també

En l'apartat 2 d'aquest mòdul aprofundirem en el concepte d'*esquema* i el disseny d'aquest esquema mitjançant regles.

altres implementacions derivades permeten l'agrupació d'elements i crear una jerarquia. LDAP no fixa cap jerarquia ni cap nombre determinat de nivells: l'espai de noms permet donar flexibilitat per a adaptar-se a molts usuaris.

Cada objecte es plasma com una **entrada de directori**. En l'exemple hi ha vuit entrades. Cada entrada té un nom *distinguished name* (DN). Per exemple, l'organització té com a DN `dc=empresa, dc=com`.

Cada entrada de directori està composta per una sèrie d'**atributs**, cadascun dels quals descriu diversos aspectes de l'objecte que l'entrada identifica. En l'exemple es defineix un usuari. Aquest objecte podria tenir els atributs descrits en la taula següent:

Exemple d'una entrada LDAP

Atribut	Valor
objectclass	person
cn	José María López García Pepe López García
sn	López García
telephoneNumber	1789
mail	josem.lopez@empresa.com
jpegPhoto	nU6KNyVIYS817zVdf5YKF1FrNb...

La definició de quins atributs formen part del directori es coneix com a **esquema del directori**. A continuació s'explica el significat dels atributs anteriors:

- **objectclass**. Especifica a quina classe pertany l'objecte.
- **Common name (cn)**. Nom de l'usuari, pot tenir més d'un valor. En aquest cas, es considera nom de l'usuari tant José María com Pepe.
- **Surname (sn)**. És el cognom de l'usuari. Pot ser útil per a ordenar alfabèticament per cognom.
- **telephoneNumber**. Com indica el nom, serveix per a emmagatzemar el número de telèfon de l'usuari. En aquest cas, es tracta d'una extensió de centraleta.
- **mail**. Emmagatzema l'adreça de correu electrònic.
- **jpegPhoto**. Conté una petita imatge de l'usuari.

En l'exemple de la figura anterior, s'utilitzen a més els atributs **domain component (dc)** i **organizationa unit (ou)**.

Vegeu també

En l'apartat 2, sobre disseny del directori, aprofundirem en el concepte d'*objecte*.

En LDAP cada objecte s'identifica mitjançant un nom, el ***distinguished name*** (DN), format per diversos atributs i els valors d'aquests atributs.

En l'exemple de la figura anterior, diem que hi ha un usuari amb DN `uid=7781287-G, ou=Direccio, ou=Usuaris, dc=empresa, dc=com`. Un atribut concret, per exemple `uid=77781287-G`, es coneix com a ***distinguished name* relatiu** o ***relative distinguished name*** (RDN).

Per norma, el valor d'RDN per a cada nivell ha de ser únic. Per tant, no hi pot haver dos usuaris amb RDN `uid=7781287-G` situats en el grup "Direccio". El que sí que està permès és que hi hagi d'un usuari també amb RDN `uid=7781287-G` però que pertanyi al grup de "Vendes".

Per a construir el DN s'usa tota la cadena d'RDN des de la fulla de l'arbre fins a l'arrel. Una variant de l'RDN es dóna quan, en lloc d'usar un únic atribut com a RDN, se n'usen diversos de l'entrada. Així, doncs, és possible tenir dues entrades amb l'RDN `cn=Servidor web`, sempre que s'usi un altre atribut, com per exemple `description`, per a diferenciar entre les dues entrades. En aquest cas, es tracta d'un RDN multivalor o *multivalued*, de manera que l'objecte DN `cn=Servidor web+description=Web tenda, ou=Servidors, dc=empresa, dc=com` és diferent de l'objecte DN `cn=Servidor web+description=Web nomines, ou=Servidors, dc=empresa, dc=com`. Encara que sigui possible l'ús de diversos atributs d'una entrada de directori a fi d'identificar una entrada, es recomana que només s'usi un atribut de cada entrada, per motius bàsicament d'eficiència.

Identificació mitjançant DN

En certa manera, la identificació mitjançant DN recorda la identificació d'un fitxer dins d'una estructura de directoris (per exemple, /usr/bin/grep).

Vegeu també

En l'apartat 2 aprofundirem en el disseny de l'espai de noms d'un servei de directori basant-nos en LDAP.

1.3. Operacions de client

Els directoris no tindrien sentit sense eines que permetessin consultar-los. D'una banda, hi ha eines que poden connectar amb el servei de directori per a fer consultes sobre els usuaris i els recursos. D'altra banda, hi ha serveis que consulten el directori amb diferents finalitats, com per exemple validar un usuari o comprovar els permisos d'accés a un servei. A més, cal disposar d'eines per a modificar la informació que conté el directori. En aquest subapartat definim les operacions més freqüents pel que fa a interacció d'un client amb un servei de directori.

1.3.1. Operacions d'interrogació

Tal com indica el nom, aquest tipus d'operacions permeten al client buscar informació. En concret, es defineixen dues operacions bàsiques: la comparació i la cerca. La comparació s'usa per a comprovar si una entrada en particular conté un valor concret per a un atribut. El servidor respon cert o fals depenent del resultat d'aquesta comparació. La comparació té un ús en casos molt limitats.

En canvi, l'operació de cerca és molt més potent i permet, en el fons, fer també tasques de comparació. L'operació de cerca és una eina potent d'interrogació a servidors de directori.

L'operació de **cerca** o *search* permet buscar en el directori i obtenir informació de les entrades.

Aquesta operació disposa de fins a vuit paràmetres, que es descriuen a continuació:

1) ***Base***. S'utilitza per a indicar a partir de quina entrada o quin objecte es vol començar a fer la cerca.

2) ***Scope***. Permet definir l'àmbit de la cerca. Els valors possibles són els següents:

a) ***Onelevel***: permet buscar només en el nivell següent al definit en el paràmetre base.

b) ***Sub***: s'utilitza per a buscar en tot el subarbre, és a dir, des de la base fins a les fulles.

c) ***Base***: s'usa per a buscar només en la base mateixa, amb l'objectiu d'obtenir informació d'aquesta base.

3) ***Alias deferencing options***. LDAP permet la definició d'àlies, mitjançant els quals es poden enllaçar entrades del directori (una és l'àlies i l'altra, l'objecte original).

Nota

L'operació de cerca permet tractar amb àlies, però aquest concepte és fora de l'objectiu d'aquest mòdul.

4) ***Size limit***. Aquest paràmetre indica al servidor el nombre màxim d'entrades trobades amb èxit que es vol obtenir.

5) ***Time limit***. Especifica el temps màxim en segons durant el qual es pot executar la cerca. Si el valor és 0 vol dir que no hi ha temps límit establert.

6) ***Attributes-only***. Permet especificar al servidor de directori que retorni únicament els atributs que contenen les entrades oposades, i no els valors d'aquests atributs.

7) ***List of attributes to return***. És la llista dels atributs que es vol obtenir. Si no s'especifiquen, s'entén que es volen obtenir tots els atributs.

8) ***Search filter***. És una expressió que descriu el tipus d'entrades que es vol obtenir. Sense pretendre ser exhaustius, a continuació mostrem alguns exemples de filtres de cerca, juntament amb el significat que tenen:

- **(sn=prados)**. Aquest criteri de cerca significa que ha de retornar les entrades amb l'atribut "sn" (cognom) amb el valor "prados".

- **(sn=mar*)**. En aquest exemple l'asterisc indica que el criteri és que el cognom comenci per *mar*, és a dir, *marquez*, *martin*, *martinez*, etc. L'asterisc permet buscar els cognoms que acabin amb *ez*: **(sn=*ez)**.

- **(sn~=fernandez)**. En aquest cas, es retornen cognoms semblants a *fernandez* (*hernandez*, *ferrandiz*, etc.).

- **(age>=18)**. Aquest filtre retorna els usuaris majors d'edat. S'ha de tenir en compte que (age<18) no és possible. Sempre hi ha d'haver un igual en la comparació per als filtres LDAP. En aquest cas, podem usar una negació: **(!(age>=18))**. Els operadors >= i <= es poden usar també en cadenes de caràcters (per exemple, cognoms); en aquest cas s'usa un ordre lexicogràfic.

- **(jpegPhoto=*)**. Retorna totes les entrades que inclouen una imatge JPEG.

Lectura recomanada

Per a saber més coses sobre el filtre de cerca o *search filter* consulteu l'obra següent:

T. A. Howes i altres (2003). *Understanding and Deploying LDAP Directory Services* (2a. ed.). Boston, MA: Addison-Wesley.

Lectura recomanada

L'RFC 2254 recull la definició dels filtres de cerca d'LDAP, i exemples per a comprendre'ls més bé.

També es pot construir filtres compostos mitjançant els operadors lògics & (and) i | (or).

Per exemple, amb l'expressió **(&(objectClass=person)(|(givenName=Pedro)(!(age<=50))))** s'obtenen les entrades de classe persona el nom de pila de la qual sigui Pedro i sigui més gran de cinquanta anys.

JXplorer

JXplorer és una eina de programari lliure per a connectar-se a servidors LDAP i obtenir informació. Té una interfície gràfica d'usuari.

Les implementacions de client LDAP ens ofereixen una eina de cerca. L'eina pot tenir una interfície gràfica d'usuari; en aquest cas disposarem de quadres de diàleg per a introduir els anteriors paràmetres de cerca, o bé disposarem d'assistents que ens ajudaran a crear-los. L'eina pot ser mitjançant l'indicador d'ordres, com **ldapsearch**.

ldapmodify

Així com ldapsearch és una eina per a consultar informació, ldapmodify permet variar les dades contingudes en el directori.

Ús de ldapsearch

Vegem un exemple d'ús de l'eina ldapsearch:

ldapsearch –h ldap.exemple.com –s sub –b "ou=enginyers" "(cn=Juan Prados)"

En aquest cas es busca en el servidor ldap.exemple.com, dins de l'apartat d'enginyers, l'entrada corresponent a Juan Prados. L'eina ldapsearch ha fet una consulta al servei de directori que no ha necessitat una connexió autenticada o, dit d'una altra manera, ha usat una **connexió anònima**.

1.3.2. Operacions d'actualització

LDAP té quatre operacions d'actualització de dades: afegir, esborrar, modificar i reanomenar/moure.

L'operació d'afegir (**add**) permet crear una nova entrada i introduir-ne les dades. Com a paràmetres de l'operació, hi ha el DN per a la nova entrada (que servirà, evidentment, per a situar l'entrada dins de l'arbre) i la llista d'atributs amb els valors d'aquests atributs. Aquesta llista de valors ha de coincidir amb l'esquema del directori, és a dir, amb el model de dades pel que fa a atributs, que estudiarem més endavant.

L'operació d'esborrar (**delete**) elimina una entrada del directori. Solament necessita el DN de l'entrada que es vol eliminar. És important que l'entrada no tingui fills en l'arbre per a poder-la eliminar.

L'operació de modificar (**modify**) permet canviar valors d'atributs d'una entrada, afegir-los o bé esborrar-los.

Finalment, una operació més complexa i alhora potent és la de reanomenar. Aquesta operació (**rename**) permet canviar el DN de les entrades. Així, doncs, en el fons, també permet moure l'entrada d'una ubicació a una altra de l'arbre. Té quatre paràmetres:

1) El DN de l'entrada a reanomenar/moure.

2) El nou RDN que tindrà l'entrada.

3) El DN del que serà el nou pare de l'entrada (el paràmetre és opcional si s'empra usant un DN diferent que el del pare actual de l'entrada; es farà un moviment de l'entrada).

4) Finalment, un indicador per a esborrar l'antic RDN. Si no s'esborra aquest antic RDN, el nou RDN de l'entrada s'afegeix com a atribut de l'entrada.

1.3.3. Operacions d'autenticació

La connexió a un directori no està exempta de les implicacions en la seguretat del mateix servei de directori. Per exemple, pot ser que la consulta d'informació sigui pública per a qualsevol usuari, mentre que la modificació només sigui possible per a certs usuaris amb rol d'administradors del servei de directori. D'altra banda, si la implementació del directori guarda dades sensibles, com per exemple contrasenyes, sense encriptar (gens recomanable), s'han de prendre precaucions especials per a garantir la seguretat de les dades. Per a fer una connexió (operació **bind**), s'ha d'especificar el DN de qui fa la connexió. Es pot usar una contrasenya per a autenticació, i també diferents mètodes de seguretat. Aquests models els revisarem més endavant en el subapartat 2.3.

1.3.4. Accés des d'altres aplicacions

Una altra manera en la qual està disponible LDAP és com a interfície per a implementar programes (API). D'aquesta manera, per exemple, en llenguatge C és possible usar funcions contra un servei de directori LDAP, com `ldap_search()`, `ldap_bind()` o `ldap_add()`.

També hi ha interfícies per a accedir a serveis LDAP des de C#, C++ o PHP.

En llenguatge Java, el Java Naming and Directory Interface (JNDI) proporciona eines per a usar serveis de directori.

L'exemple següent mostra una connexió anònima a un servei LDAP mitjançant C#:

```
LdapConnection ldapConn= new LdapConnection();

ldapConn.Connect ("ldap.exemple.com",389);

ldapConn.Bind (null, null);
```

L'exemple següent fa el mateix, però aquesta vegada en PHP:

```
<?php

$ldapconn = ldap_connect("ldap.exemple.com")
   or die("Impossible connectar.");


if ($ldapconn) {
   $ldapbind = ldap_bind($ldapconn);
}

?>
```

Finalment, aquest exemple il·lustra la connexió a un directori LDAP mitjançant JNDI:

```
DirContext ctx = new InitialDirContext(env);

env.put(Context.INITIAL_CONTEXT_FACTORY,"com.sun.jndi.ldap.LdapCtxFactory");

env.put(Context.PROVIDER_URL, "ldap://direccio:389");

env.put(Context.SECURITY_AUTHENTICATION,"simple");
```

```
env.put(Context.SECURITY_PRINCIPAL,"cn=usuari");


env.put(Context.SECURITY_CREDENTIALS,"contrasenya");
```

2. Disseny del directori

L'espai de noms és un element estretament lligat amb l'organització dels objectes que inclou el directori. Tal com hem vist, en LDAP els objectes s'identifiquen mitjançant una sèrie de valors específics d'atributs, en principi un per cada nivell de l'espai de noms.

Així, doncs, es pot pensar que hi hauria d'haver certa relació entre l'organització de la institució i el disseny del seu directori. D'altra banda, s'ha de tenir clar quins elements convé emmagatzemar per a cada entrada, és a dir, quins són els atributs que definiran els objectes del directori. En aquest apartat, posem l'èmfasi en els aspectes de disseny del directori, tant pel que fa a espai de noms com pel que fa a l'esquema. També fem algunes reflexions entorn de la seguretat, la robustesa i l'eficiència del sistema.

2.1. Disseny de l'espai de noms

Per al disseny de l'espai de noms es poden tenir en compte tres elements: l'elecció del sufix, l'estructura o complexitat del directori i la identificació d'objectes dins del directori.

2.1.1. Elecció del sufix

L'organització per la qual es dissenya el directori pot tenir un àmbit local o bé pot formar part d'una organització més gran, que també tingui un directori. Sigui com sigui, l'important és que el sufix (el nom de la part "arrel" de l'arbre del directori) identifiqui unívocament l'organització. Hi ha tres alternatives vàlides per a triar el sufix:

1) La primera consisteix a complir la recomanació de l'RFC 2247, en la qual s'especifica que és convenient mapar el DN del directori amb el nom DNS que tingui assignat (o pretengui tenir assignat algun dia) l'organització.

> Si l'organització té, per exemple, un lloc web amb el domini *www.empresa-exemplar.com*, és convenient que el DN del domini, el sufix, sigui DN `dc=empresa-exemplar, dc=com`.

2) La segona alternativa, lleugerament diferent de la primera, consisteix a usar tot el nom de domini, usant aquesta vegada l'atribut *o* (d'**organization**). En aquest cas, DN `o=empresa-exemplar.com`.

3) La tercera alternativa, finalment, té a veure amb la localització geogràfica de l'empresa, tal com es formulava en les recomanacions per a X.500.

Aquestes nomenclatures s'usen, per exemple, per a identificar organismes i entitats dins dels certificats electrònics. Per a fer-ho, s'usa l'atribut *c* (**country**). En cas que l'empresa sigui a Espanya i es digui Empresa Exemplar, SA, tindrem DN `o=Empresa Exemplar\, SA, c=ES`. Fixem-nos en l'ús del caràcter (\) per a denotar que la coma pertany al valor de l'atribut i l'intèrpret no l'ha de confondre amb l'inici d'un nou atribut.

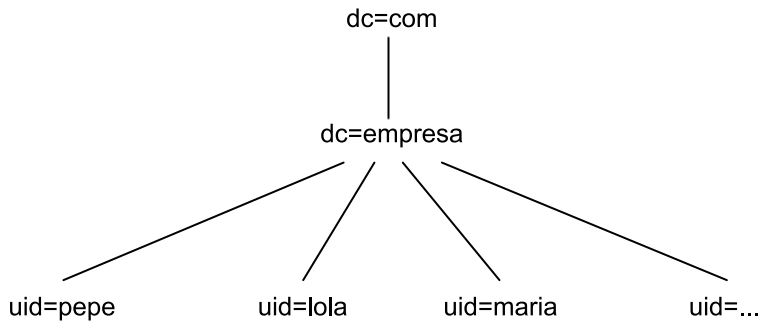
Finalment, és possible que un directori tingui més d'un sufix. Això cal si es fusionen directoris pertanyents a organitzacions amb sufixos diferents.

2.1.2. Estructura del directori

Un altre element que s'ha de tenir en compte en dissenyar un espai de noms és l'estructura del directori, que en el fons també tindrà implicacions en la complexitat del directori pel que fa a la jerarquia.

L'espai de noms més simple és un espai de noms pla; per exemple, sense departaments ni grups d'usuaris. La figura mostra un espai de noms pla:

Espai de noms pla



Això és adequat per a organitzacions amb pocs usuaris o recursos. Ara bé, en grans organitzacions és recomanable seguir una estructura jeràrquica.

En organitzacions que preveuen departaments o diferents perfils d'usuaris, s'aconsella crear unitats organitzacionals (identificades amb l'atribut *ou*). A més, el fet de distribuir els recursos en grups pot ser útil, per exemple, per a qüestions de control d'accés a recursos.

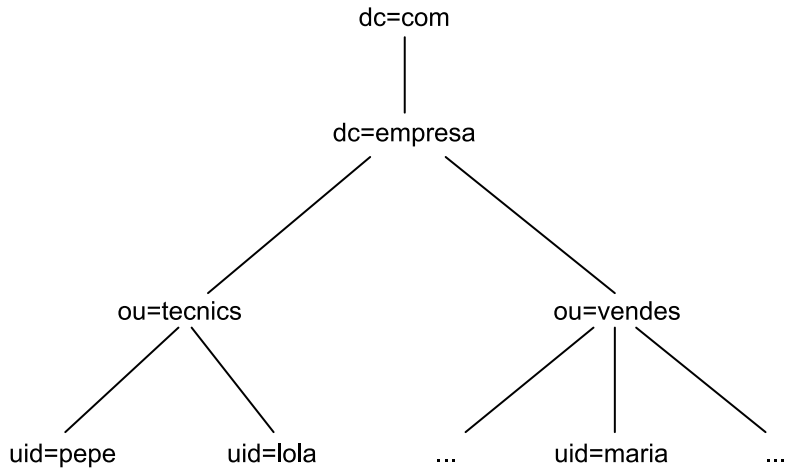
Vegeu també

El control d'accés a recursos es tracta en el mòdul "Identificació, autenticació i control d'accés".

Exemple

Imaginem-nos la distribució d'usuaris d'una empresa com la que mostra la figura:

Exemple d'estructura jeràrquica



Usant una estructura com l'anterior, és fàcil permetre l'accés a determinats recursos als usuaris del grup "Tècnics" o als del grup "Vendes".

2.1.3. Identificació d'objectes en el directori

L'últim element que analitzarem sobre el disseny de l'espai de noms té a veure amb la identificació dels objectes dins del directori. Tal com hem vist, les restriccions que imposa LDAP a l'hora d'identificar objectes són dues:

- L'RDN d'una entrada es forma a partir d'un atribut (o més d'un, encara que no és recomanable).

- L'RDN ha de ser únic entre les entrades germanes (aquelles que pengen del mateix pare en l'estructura).

Encara que solament hi hagi establertes aquestes dues restriccions, s'aconsella en general que la identificació dels objectes (especialment quan es tracta de persones o usuaris) es faci amb un identificador únic.

Per a fer això, s'ha de garantir que el nom d'usuari sigui únic. Clarament, hi podria haver dues persones amb el mateix nom. Encara que una organització sigui petita, resultaria poc pràctic que hi hagués un usuari identificat amb un nom molt comú, com per exemple *pepe*, perquè fàcilment hi pot haver o hi haurà algun dia un altre usuari *pepe* per a emmagatzemar en el directori (i no seria gaire estètic anomenar-lo *pepe2*). Per aquest motiu s'ha d'utilitzar un atribut que identifiqui unívocament l'usuari; per exemple, es pot usar el seu número d'identificació fiscal (NIF).

Clarament, una altra opció per a gestionar la unicitat dels atributs d'identificació és que l'administrador creï els identificadors i que controli que són únics. Es podrien usar les inicials, i addicionalment nombres, per a evitar repeticions. O bé si el directori s'usa per a accedir a serveis que funcionen en

sistemes operatius del tipus Unix o Linux, estaria bé que l'identificador d'usuari coincidís amb el nom d'usuari que es defineixi per als serveis. En aquest cas s'utilitzaria l'atribut *uid*, que significa identificador d'usuari o *user identifier*.

Finalment, una altra opció, encara que no recomanable, és assignar una cadena aleatòria com a identificador de l'usuari. Diem que no és recomanable perquè no seria fàcil recordar aquesta cadena aleatòria.

Malgrat que té moltes opcions per a identificar usuaris i recursos, la solució ha de ser pràctica i estètica.

2.2. Esquema del directori

Fins ara hem parlat d'objectes en el sentit ampli de la paraula. Hem vist que un directori conté entrades amb els diferents atributs d'aquestes entrades. Una entrada representa un objecte la informació del qual és emmagatzemada en el directori. I es pot entreveure que les entrades poden ser de diversos tipus: individus, recursos, grups, etc. Més amunt hem apuntat en què consisteix l'esquema del directori; ara tractarem diversos aspectes relacionats amb el disseny.

L'**esquema del directori** és la definició de quins tipus d'objectes guarda un directori i quins atributs s'utilitzen per a definir-los.

Les implementacions de serveis de directori ja solen incloure les seves pròpies definicions d'esquema, els quals se solen poder ampliar sense problemes per a cobrir qualsevol necessitat.

En el decurs d'aquest subapartat tractarem dels conceptes d'*atribut* i *classe*, essencials en la definició de l'esquema del directori.

2.2.1. Atribut

Els atributs serveixen directament per a guardar informació (nom de persona, número de telèfon, fotografia, etc.). Per a definir un atribut en LDAP, cal disposar d'una sèrie d'informació:

- Un nom que identifica l'atribut que es defineix. En el cas d'LDAP, ja hi ha alguns noms estàndard definits (*common name*, *telephoneNumber*, etc.), alguns dels quals, amb una abreviatura també coneguda (per exemple, *cn* per a *common name*). LDAP no distingeix entre majúscules i minúscules.

- Un OID (identificador d'objecte) que també identifiqui l'atribut. Els OID són cadenes de números que permeten localitzar de manera precisa un objecte de dades. Els OID també defineixen un espai de noms amb una jerarquia. Per exemple, 2.5.4.16. és l'OID de *postalAddress* i el valor 2 significa que la resta d'estructura ha estat proposada amb l'ITU en conjunció amb l'ISO. La notació que segueixen els OID està definida en l'ASN.1 (notació de sintaxi abstracta número u o *abstract syntax notation number one*). Com que en el fons no cal usar l'OID tret que s'hagi d'interactuar amb sistemes X.500, l'ús d'OID no sol ser gaire popular, ja que és més còmode treballar amb noms a l'hora d'identificar l'atribut.

- Una descripció textual de l'atribut per a permetre anotacions. Aquesta descripció permet fins a 1.024 caràcters.

- Una sintaxi de l'atribut, la qual defineix com es representen les dades. La sintaxi s'identifica mitjançant un OID.

  Per exemple, si s'especifica 1.3.6.1.4.1.1466.115.121.1.15, ens referim al fet que la sintaxi de l'atribut és la cadena de text. O bé, si es tracta d'un número de telèfon, que comença per (+), seguit del prefix internacional i del número d'abonat, ens referim a la sintaxi 1.3.6.1.4.1.1466.115.121.1.50.

- Unes regles de coincidència, que són utilitzades en les cerques d'informació en el directori. Per exemple, si s'especifica *caseIgnoreMatch*, s'indica que les majúscules no s'han de tenir en compte a l'hora de comparar cadenes de caràcters.

- Altres valors, com per exemple la longitud màxima.

Els atributs poden derivar d'altres atributs. És a dir, donada la definició general d'un atribut com per exemple *name*, hi pot haver una sèrie d'atributs que en derivin i, en conseqüència, que n'heretin les característiques.

ASN.1

L'ASN.1 defineix un marc d'identificació de dades amb independència del mètode de representació que usi una màquina. Correspon a la capa de presentació de l'OSI (interconnexió de sistemes oberts o *open systems interconnection*).

X.500

Recordeu que X.500 va ser el primer sistema estàndard de directori.

Enllaç d'interès

Trobareu informació sobre els OID al lloc web següent: http://www.oid-info.com.

Definició de nous atributs

Abans de definir nous atributs, convé saber quins estan definits a l'eina de servei de directori que usarem. Per a aquest objectiu, hem de consultar els fitxers de definició d'esquema que inclogui el nostre producte de servei de directori.

Exemple

L'esquema de dades d'LDAP, que especifica els atributs estàndard que ja estan definits per a un servei de directori, inclou la definició de l'atribut *name*:

```
attributetype ( 2.5.4.41 NAME 'name'

        EQUALITY caseIgnoreMatch

        SUBSTR caseIgnoreSubstringsMatch

        SYNTAX 1.3.6.1.4.1.1466.115.121.1.15{32768} )
```

A l'hora, però, de definir l'atribut *cn* o *commonName*, especifica que el seu superior és *name*:

```
attributetype ( 2.5.4.3 NAME ( 'cn' 'commonName' ) SUP name )
```

En l'exemple de la definició de l'atribut *name* s'observa l'ús d'`EQUALITY` i `SUBSRT` per a especificar regles de coincidència, mentre que la sintaxi s'especifica mitjançant `SYNTAX` i el corresponent OID. En la mateixa sintaxi s'especifica una longitud màxima entre claus.

Per a acabar el subapartat sobre atributs, presentem la taula següent, que conté alguns dels atributs estàndard en LDAP.

Descripció d'alguns dels atributs més habituals en LDAP

Atribut	Descripció
cn, *commonName*	Nom de l'objecte. Si l'objecte és una persona, serveix per a especificar el seu nom complet
sn, *surname*	Cognom d'una persona
serialNumber	Número de sèrie d'un dispositiu o recurs
c, *countryName*	Nom del país usant dos caràcters, tal com especifica l'ISO 3166
st, *stateOrProvinceName*	Nom de l'estat, província, comunitat autònoma, etc.
street, *streetAddress*	Adreça postal (carrer, número, etc.)
o, *organizationName*	Nom de l'organització
ou, *organizationalUnitName*	Nom del departament o una altra unitat organitzacional
title	Títol de la persona dins d'una organització (presidenta, directora, etc.)
description	Descripció de l'objecte, de manera comprensible per als humans
postalCode	Codi postal
telephoneNumber	Número de telèfon
preferredDeliveryMethod	Descripció de la manera com vol una persona que li sigui lliurada la informació (per exemple, per fax o correu electrònic)
member	Es tracta d'un DN que indica de qui és membre l'objecte en l'arbre del directori
uid, *userid*	Identificador d'usuari, en general usat per a autenticar-se en un servei o sistema
userPassword	Contrasenya

Atribut	Descripció
dc, domainComponent	Especificació d'un domini DNS segons les RFC 1274 i 2247; per exemple, *es* o *uk*
buildingName	Nom de l'edifici on hi ha una organització o una persona
preferredLanguage	Idioma preferit per la persona per a comunicar-se oralment o per escrit
userSMIMECertificate	Certificat o cadena de certificats per a l'autenticació i comprovació de signatures electròniques

2.2.2. Classe d'objecte

Fins ara hem vist la definició d'atributs. Amb aquesta operació es poden definir atributs específics per al nostre servei de directori. Una vegada han estat definits els atributs, es poden utilitzar classes d'objectes per a definir com són les entrades del directori. Cada entrada del directori pertany a un tipus d'objecte determinat. Com en el cas dels atributs, ja hi ha algunes classes d'objectes definides com a estàndard.

Amb les classes d'objectes s'especifica quins atributs ha de contenir l'entrada de manera obligatòria i quins atributs pot contenir definits de manera opcional. També és possible usar la classe d'objecte a l'hora de buscar informació en el directori.

Exemple

Recordem que en l'exemple sobre operacions d'interrogació que hem vist més amunt, especificàvem *objectClass=person* per a buscar només persones (en el fons, entrades corresponents a la classe *person*). Vegem un exemple de definició de la classe *person*:

```
objectClasses: ( 2.5.6.6 NAME 'person'

 DESC 'RFC2256: a person' SUP top

 STRUCTURAL MUST ( sn $ cn )

    MAY ( userPassword $ telephoneNumber $ seeAlso $ description ) )
```

Fixem-nos que, d'una manera similar a la definició d'atributs en l'esquema, la definició d'una classe demana l'ús d'un OID (2.5.6.6). També hi ha un camp de descripció textual en què s'informa que la definició de *person* es correspon amb el que estableix l'RFC 2256.

S'especifica que la classe és de tipus estructural. Això vol dir que la classe descriu propietats bàsiques de l'objecte; en general, gairebé totes les classes definides són d'aquest tipus. La descripció d'altres tipus s'escapa dels objectius d'aquest mòdul.

L'apartat `MUST` indica que cal que un objecte del tipus "person" inclogui el cognom (*sn*) i el nom complet (*cn*). Els camps que pot contenir, però que no són essencials, estan indicats en `MAY`. En la definició d'algunes classes, no hi ha l'entrada `MUST`. Això vol dir que no es requereix cap camp en concret.

Finalment, veiem que en les classes també hi ha la definició de *superiors*; en aquest cas, el superior de *person* és *top*.

En la taula s'especifiquen algunes característiques per a algunes de les classes típicament definides en un esquema.

Característiques d'algunes de les classes típicament definides en un esquema

Classe	MUST	MAY
alias (permet que l'entrada de directori apunti cap a una altra entrada)	aliasedObjectName	
organization	o	userPassword, seeAlso, businessCategory, preferredDeliveryMethod, telephoneNumber, facsimileTelephoneNumber, street, postOfficeBox, postalCode, postalAddress, physicalDeliveryOfficeName, description, etc.
organizationalUnit	ou	userPassword, seeAlso, businessCategory, preferredDeliveryMethod, telephoneNumber, internationaliSDNNumber, facsimileTelephoneNumber, street, postOfficeBox, postalCode, description, etc.
person	sn, cn	userPassword, telephoneNumber, seeAlso, description
device	cn	serialNumber, seeAlso, owner, ou, o, l (localització), description
account	userid	description, seeAlso, l, o, ou, host
document	documentIdentifier	commonName, description, seeAlso, l, o, ou, documentTitle, documentVersion, documentAuthor, documentLocation, documentPublisher
room	commonName	roomNumber, description, seeAlso, telephoneNumber

Fixem-nos que classes com *organizationUnit* poden tenir contrasenya o número de telèfon.

2.3. Seguretat, eficiència i disponibilitat del directori

Després d'haver analitzat els punts clau del disseny de l'espai de noms i de l'esquema del directori, veurem breument altres aspectes estretament relacionats amb la implantació real del servei de directori.

2.3.1. Seguretat en el directori

L'objectiu d'aplicar seguretat al directori és protegir la informació d'accessos no autoritzats. Aquest model va més enllà de permetre l'accés al servei de directori, i fins i tot pot detallar quines operacions es poden dur a terme amb quins atributs, i qui pot fer aquestes operacions.

Una opció per a protegir la informació és usar protecció en la capa de transport de la informació. Així, per exemple, es pot usar un **canal de comunicació SSL/TLS**[6] per a proporcionar confidencialitat i autenticitat a la informació que es transmet entre el servei de directori i l'eina amb què interactua.

(6)*SSL* és la sigla de *capa de sòcol segur* o *secure socket layer*; i TLS, de *seguretat de nivell de transport* o *transport layer security*.

Vegeu també

L'emmagatzematge de contrasenyes s'estudia en el mòdul "Identificació, autenticació i control d'accés".

Una de les informacions que convé protegir de l'accés indegut és la contrasenya. En els estàndards no s'especifica com s'ha de fer aquesta protecció, de manera que es deixa en mans de l'implementador. El que sí que és habitual és que en lloc de guardar la contrasenya en clar es guardi un resum de la contrasenya en lloc de la contrasenya original.

En el cas de la contrasenya, guardant-ne el resum no és computacionalment possible saber quin n'és el valor original. Ara bé, com que la funció de resum que s'utilitza per a guardar la contrasenya és coneguda, és relativament fàcil que un atacant creï una contrasenya i en guardi el resum a l'entrada de qualsevol usuari. Amb aquest atac, l'atacant pot suplantar la identitat de l'usuari davant qualsevol servei o sistema que usi el directori com a eina d'autenticació.

En relació amb aquest tema, pot ser que calgui la definició de **llistes de control d'accés**. Mitjançant aquestes llistes, el servei de directori té definits clarament diversos paràmetres sobre l'accés a les dades, el seguiment de les quals s'ha de procurar de fer davant l'accés de consultes i modificacions.

Exemple

Per a cada atribut de cada tipus d'entitat, es poden definir permisos de lectura i escriptura depenent del perfil de qui executa la petició, a més de poder-se especificar mètodes de validació específics. Així, doncs, per a veure l'extensió telefònica d'un treballador d'una organització, no fa falta autenticar-se (n'hi ha prou amb una connexió anònima). O bé, per a modificar el valor d'un atribut *salari*, cal autenticar-se com un usuari del grup d'administradors.

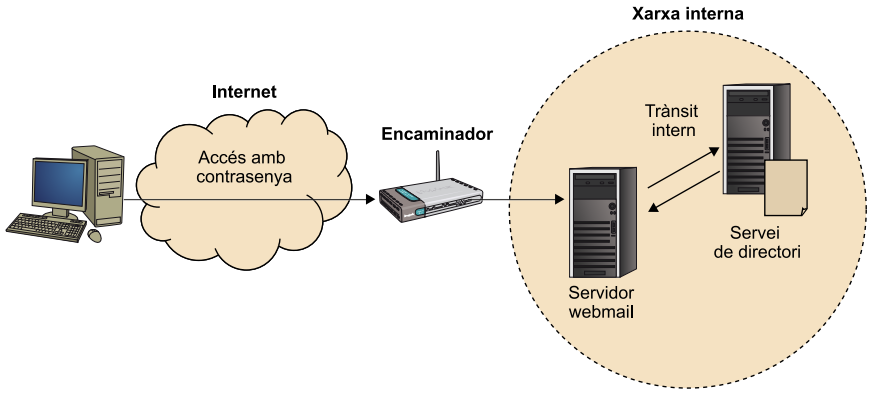
La majoria d'implementacions de servei de directori permeten l'autenticació del client (recordem que pot ser un usuari o bé una altra aplicació telemàtica) mitjançant un certificat electrònic. Una altra manera de restringir les dades que conté un servei de directori és fent que aquestes dades estiguin aïllades dels usuaris o equips que no poden accedir al servei.

Exemple

Si un servei de directori està pensat per a fer autenticacions en servidors de correu o intranets, el servidor que controla el directori pot ser a la xarxa interna de l'organització. El portal web de la intranet, que ha de ser ben accessible des de l'exterior per a oferir els seus serveis, fa una consulta a una màquina (el servidor del directori) que pot estar ben bé situada a la xarxa interna. Una altra alternativa és protegir el servidor de directori mitjançant un tallafoc que permeti l'accés a determinats perfils d'adreça de xarxa.

En la figura, el client es connecta a un servidor de correu web per Internet, i per a autenticar-se envia la contrasenya. La comprovació de la contrasenya es fa a escala interna amb el servei de directoris que es troba sense connexió a la xarxa exterior.

Ubicació del servei de directori en una xarxa interna

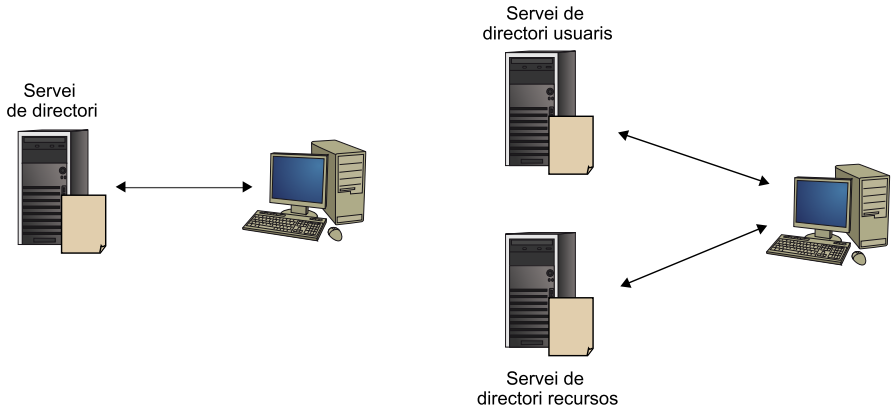


2.3.2. Eficiència i disponibilitat

D'altra banda, tan importants com la seguretat són l'eficiència i la disponibilitat. És cert que un petit directori per a una organització petita es pot executar en una única màquina, de la qual es pot fer una còpia de seguretat diària de les dades. Ara bé, si el nombre potencial de peticions d'informació al servei de directori és molt elevat, pot ser convenient que, per motius d'eficiència, les dades estiguin distribuïdes en més d'una màquina.

La figura mostra dues disposicions de les dades d'un servei de directori. A l'esquerra, un únic servei de directori conté tota la informació de l'organització. A la dreta, hi ha dos serveis de directori: un d'especialitzat a contenir la informació sobre usuaris i un altre d'especialitzat en informació sobre recursos.

Exemple de distribucions única (esquerra) i distribuïda (dreta)



La compartició d'informació s'ha de fer de manera que el servidor corresponent a usuaris i el servidor corresponent a recursos rebin aproximadament la mateixa càrrega d'operacions. Si no pot ser així, s'ha d'apostar per la replicació de la informació en més d'una màquina.

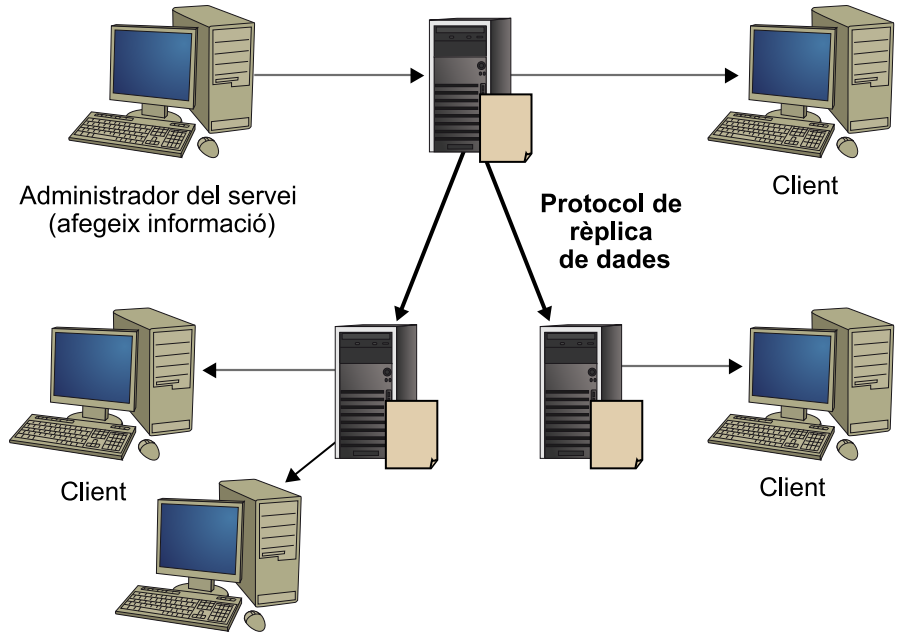
La replicació d'informació, és a dir, el procés de mantenir moltes còpies de dades del directori en diferents ubicacions, propicia una sèrie d'avantatges relacionats amb l'eficiència del sistema. D'entrada, no es col·lapsa un únic servei

amb les peticions dels clients. La replicació de dades permet que hi hagi un servidor que centralitzi les peticions dels clients i, depenent de diferents paràmetres, reencaminar la petició. Per exemple, si s'usa com a paràmetre la càrrega de treball dels servidors de rèplica, es reencamina la petició a la màquina que tingui menys treball en aquell moment. A més, si la xarxa és relativament gran (per exemple, una xarxa de campus o metropolitana), l'accés a dades és més ràpid si aquestes dades estan replicades prop de la ubicació de xarxa del client.

Addicionalment, es resolen temes de disponibilitat: si cau un dels servidors que conté la rèplica, el client pot fer ús d'algun dels servidors de rèplica que siguin actius.

Perquè el sistema de replicació funcioni, tal com es mostra en la figura, cal que es prevegi un protocol de replicació de dades per a donar consistència a la informació continguda en els diferents servidors.

Exemple de replicació de dades



3. Implementacions de servei de directori

Després d'haver fet un acostament teòric al concepte, el disseny i els usos dels serveis de directori, i d'haver estudiat el sistema LDAP, ara estudiarem tres casos concrets d'implementació:

- OpenLDAP
- Apache Directory Server
- Active Directory

Tots tres estan inspirats en LDAP.

3.1. OpenLDAP

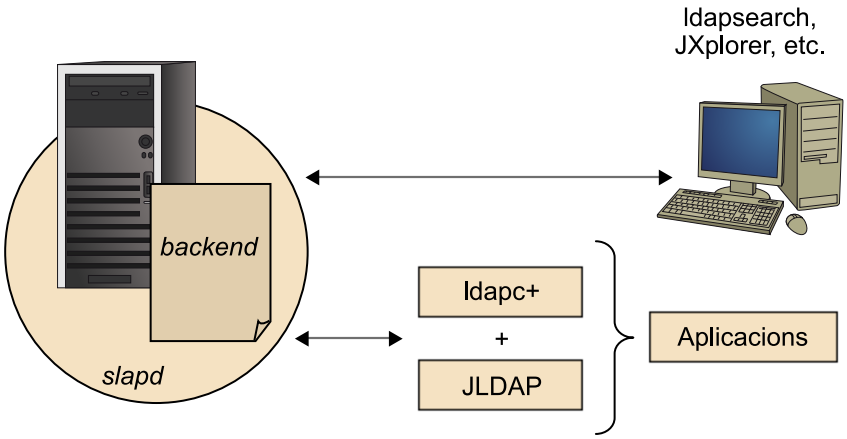
L'OpenLDAP és una implementació d'un servei de directori basat en LDAP amb la filosofia de programari lliure i codi obert. El projecte OpenLDAP és qui s'encarrega de desenvolupar-lo i els seus productes són disponibles en moltes distribucions GNU/Linux. A més, hi ha versions per a altres plataformes de sistemes operatius, com el Mac US X o l'MS Windows.

Origen de l'OpenLDAP

L'OpenLDAP es va desenvolupar a la fi dels anys noranta a partir de la implementació d'LDAP que va fer la Universitat de Michigan, que, com hem apuntat al principi d'aquest mòdul, és la responsable de la creació d'aquest estàndard de serveis de directori.

L'OpenLDAP i els projectes que se'n deriven proporcionen diferents components, alguns dels quals estan inclosos en la llista de l'esquema següent:

Components d'aplicacions i eines relacionades amb LDAP



El servidor **slapd** executa les funcions de servei de directori. S'encarrega d'interaccionar amb el *back-end* corresponent. JLDAP i ldapc++ proporcionen biblioteques perquè es puguin programar aplicacions en Java i en C++, respectivament.

Per al *back-end*, és a dir, el magatzem de dades, hi ha tres tipus de categories:

1) *Back-end* d'emmagatzematge de dades. Aquests sistemes realment emmagatzemen informació. El primer va ser el back-bdb. També hi ha el back-ldif, que, en lloc de guardar les dades en un format propi, les emmagatzema directament en fitxers de text pla en format LDIF.

Lectura complementària

Les especificacions tècniques del format d'intercanvi de dades en LDAP o *LDAP data interchange format* (LDIF) estan recollides en l'RFC 2849.

2) *Back-end* intermediari. Actuen de servidor intermediari o *proxy* entre el servei de directori LDAP i el *back-end* que realment emmagatzema les dades. Per tant, es pot tenir un *back-end* local que realment cursi peticions sobre un *back-end* extern. Per exemple, el back-sql estableix connexions cap a bases de dades de conformitat amb SQL.

SQL

El llenguatge d'estructuració interrogat o *structured query language* (SQL) és un sistema estàndard d'instruccions per a definir, gestionar i consultar dades en una base de dades relacional.

3) *Back-ends* dinàmics. El propòsit d'aquest tipus de *back-ends* és variat. Per exemple, n'hi ha que estan especialitzats en localització de serveis LDAP mitjançant DNS (*back-dnssrv*) o n'hi ha que permeten fer estadístiques de l'ús de *slapd* (*back-monitor*).

A més, com a complement dels diferents *back-end* disponibles, hi ha la figura de l'*overlay*, que és un element que permet incrementar les funcionalitats del sistema, afegint-hi mòduls.

Exemple

Si s'ha de registrar l'activitat de peticions es pot usar l'*overlay auditlog* perquè es generin fitxers de text pla o bé *accesslog* perquè l'activitat quedi registrada usant una base de dades LDAP.

L'OpenLDAP permet la replicació de continguts. En les primeres versions s'utilitzaven els conceptes de *servidors màster* i *esclaus*: el màster acceptava actualitzacions dels clients, mentre que els esclaus només acceptaven actualitzacions des d'un màster. Recordem que l'actualització o sincronització de continguts entre els diferents servidors és essencial per al bon funcionament de la replicació de continguts. Actualment, s'utilitzen els termes *proveïdor* i *consumidor d'actualitzacions*. Això permet definir regles millors d'actualització, que fan possible que un servidor pugui actuar de proveïdor o bé de consumidor segons la necessitat que tingui.

Lectura complementària

A l'RFC 4533 es detallen els protocols de sincronització.

El motor de sincronització per a l'OpenLDAP és **syncrepl**, que usa com a protocol el **LDAP Sync**. Aquest protocol permet tant actualitzacions del tipus *pull*, en què el consumidor fa enquestes periòdiques al proveïdor per veure si hi ha hagut actualitzacions, o del tipus *push*, en què el consumidor està atent a les notificacions d'actualització enviades pel proveïdor.

3.2. Apache Directory Server

L'OpenLDAP és un dels serveis de directori més utilitzats avui dia, encara que també hi ha altres alternatives atractives de programari lliure, com per exemple el servei de directoris que preveu Apache.

L'Apache Directory Server (ApacheDS) és un servei de directori desenvolupat amb el llenguatge de programació en Java i disponible amb la llicència Apache Software. També pot funcionar com a servidor d'autenticació Kerberos, però l'ús principal que se'n fa és de servidor LDAP.

Vegeu també

El servidor d'autenticació Kerberos s'estudia en el mòdul "*Single sign-on* i federació d'identitats".

El sistema es pot encastar sense dificultat en altres projectes basats en components Java, cosa que el fa atractiu com a motor de servei de directori si s'està desenvolupant un sistema d'informació basat en Java. En aquest sentit, si l'OpenLDAP és més indicat per a funcionar com a gran servei de directori, que estigui replicat i distribuït, i funcioni com un servei individual, l'ApacheDS està més pensat per a formar part d'un paquet d'aplicacions que necessitin, en principi internament, un servei de directori.

Les aplicacions Java usen JNDI[7] per a interactuar amb el servei de directori. De fet, l'ApacheDS implementa un JNDI.

(7)*JNDI* són les sigles de *Java Naming and Directory Interface.*

L'ApacheDS utilitza una sèrie d'interfícies anomenades *multipurpose interfaces for network applications* (MINA). Aquestes interfícies contenen mètodes per a generar objectes que implementin noves funcionalitats. Així, doncs, es poden implementar protocols que usin ApacheDS com a mitjà d'emmagatzematge i gestió.

Finalment, igual que l'OpenLDAP, l'Apache DS té una sèrie d'interceptors i altres elements de programari que en permeten l'extensió.

3.3. Active Directory

En els anys noranta, Microsoft va implantar el concepte de *sistema operatiu de xarxa* als seus productes. En concret, Windows NT 3.0 ja implementava un entorn en xarxa, amb diferents recursos accessibles remotament i usuaris, gestionats per un administrador. El concepte de *domini* agrupava diferents recursos i usuaris.

A la fi dels noranta, va aparèixer l'Active Directory, una concreció del concepte de *directori* adaptat mitjançant tècniques provinents de l'LDAP per a donar resposta a propietats de robustesa i rendiment.

Si Windows NT usava el NetBIOS com a mecanisme primari de comunicació de xarxa (i el WINS com a base de dades de nom d'objectes de xarxa), l'Active Directory requereix l'ús de TCP/IP, i també del servei DNS.

Igual que altres implementacions d'LDAP, ofereix un sistema d'administració centralitzat, i l'organització de les dades en forma distribuïda i replicada.

L'Active Directory incorpora un magatzem de dades per a guardar informació sobre els objectes (objectes com usuaris o impressores). L'estructura que ofereix es basa en quatre conceptes:

1) El domini és l'estructura bàsica, la qual agrupa tots els objectes que s'administren. Un domini es pot identificar mitjançant una estructura DNS.

2) La unitat organitzativa és una unitat inferior, que pot estar composta per altres unitats organitzatives, però també per grups i objectes.

3) Els grups són conjunts d'objectes del mateix tipus.

4) Finalment, la unitat bàsica és l'objecte, és a dir, la representació dels recursos i usuaris del sistema en xarxa.

En cas que hi hagi diversos dominis compartint un espai de nomenclatures i un esquema comú, s'estableix l'estructura d'**arbre de dominis**.

Exemple d'arbre de dominis

Hi pot haver un arbre format pels dominis següents:

- *hospitalCiutat1.hospitals.org*
- *hospitalCiutat2.hospitals.org*
- *hospitalCiutat3.hospitals.org*

Finalment, un **bosc** és una col·lecció d'arbres que, encara que no comparteixen un espai de nomenclatura contigu, tenen un esquema comú. Per exemple: *hospitals.org* o *ajuntaments.org*.

En un sistema basat en l'Active Directory, un servidor pot exercir diversos papers:

- **Controlador de domini**. Aquests servidors pertanyen al domini i contenen una còpia de les dades pertanyents a recursos i usuaris. Contenen una còpia del compte de l'usuari. Són un element indispensable del domini i se'n poden usar diversos per a distribuir o replicar la informació.

- **Servidor de catàleg global**. Inclou informació sobre tots els objectes d'un bosc.

- **Servidor membre**. Pertany també al domini, però no conté còpies dels comptes d'usuari. S'usa per a guardar els arxius i recursos de xarxa.

- **Servidor independent**. Aquest servidor no té res a veure amb la gestió de l'Active Directory (per exemple, un servidor Windows que pertanyi a un grup de treball concret).

Quan hi ha diversos servidors cohabitant per a la replicació o distribució d'informació, és convenient usar un esquema de xarxa per a optimitzar la gestió del trànsit entre diferents servidors. Ara bé, operacions com la modificació de l'esquema només les pot fer un dels servidors, i no des de qualsevol servidor del sistema de rèpliques.

Active Directory en sistemes Unix

Malgrat que és un producte de Microsoft per a plataformes Windows, l'Active Directory es pot integrar a sistemes d'informació basats en sistemes operatius del tipus Unix mitjançant programari de terceres parts.

Resum

En aquest mòdul hem estudiat el concepte de *servei de directori* i hem vist aspectes relacionats amb el disseny que té i la implantació.

En primer lloc, hem estudiat l'ús dels directoris. Hem vist que són un tipus concret de base de dades que ens retorna informació a partir de la identificació d'una entrada (el DN) o bé a partir de cerques en el directori. Hem revisat diferents tipus de directoris, des dels integrats en una aplicació fins als de sistemes operatius de xarxa. Hem vist la relació d'un servei de directori amb la seguretat de la informació i l'accés a aplicacions. Hem descrit diversos exemples d'espai de noms, en concret el DNS, el sistema WINS i l'espai de noms descrit per LDAP. Hem estudiat el concepte de *distinguished name*. Hem acabat aquesta part del mòdul descrivint quines són les operacions que poden fer els clients amb un servei de directori. Hem vist de manera detallada les operacions d'interrogació, especialment la cerca. Així mateix, hem indicat les eines que hi ha perquè altres aplicacions i programes puguin accedir a les dades que conté un directori.

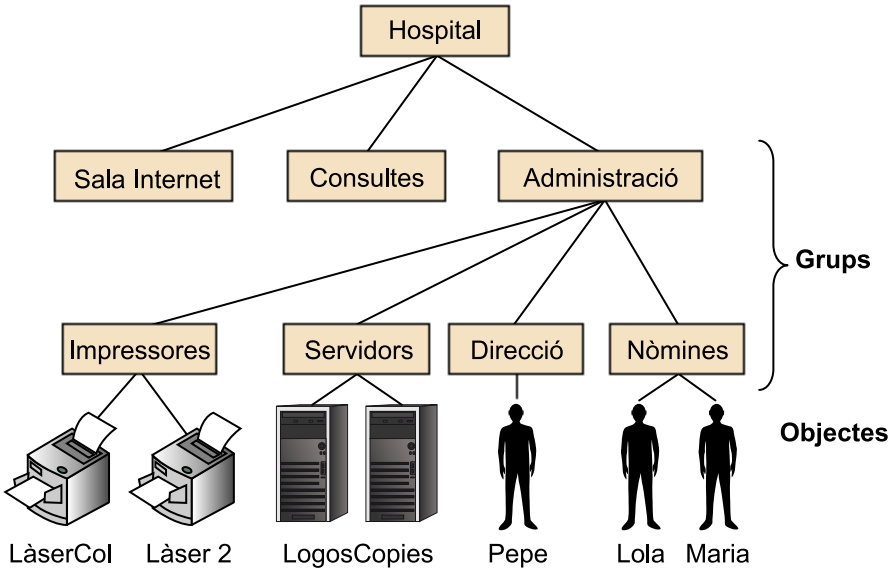
L'apartat següent l'hem dedicat a aspectes de disseny del directori. Amb referència al disseny de l'espai de noms, hem tractat de l'elecció del sufix del directori (com comença l'espai de noms), com s'ha de decidir l'estructura (hem vist que hi ha espais de noms plans i jeràrquics) i com s'han d'identificar els objectes. També hem aprofundit en el concepte d'*esquema del directori*, amb el qual descrivim quina informació es guarda al directori. Hem exposat com es defineix un atribut en LDAP i també com podem definir una classe d'objecte. Per acabar aquest apartat, hem discutit aspectes relacionats amb la seguretat en l'accés al directori, l'eficiència del servei de directori i la disponibilitat de les dades que conté aquest directori.

Per acabar el mòdul, hem descrit tres implementacions de servei de directori basades en LDAP i àmpliament utilitzades: l'OpenLDAP, l'Apache Directory Server i l'Active Directory de Microsoft.

Activitats

1. En la figura es mostra l'estructura d'un directori d'un hospital, que preveu la xarxa amb els recursos i els usuaris. Està dividit en tres unitats organitzatives: una sala d'Internet, els recursos informàtics de les consultes de l'hospital i els recursos informàtics de l'administració. En concret, la unitat organitzativa d'Administració, inclou el grup Direcció (integrat només per un usuari), el grup Nòmines, integrat pels usuaris que pertanyen a la gestió de nòmines, un grup Servidors amb un parell de màquines i un parell d'impressores, situades dins del grup Impressora.

Escenari de directori



Dissenyeu un esquema per a aquesta organització i detalleu quines classes quins atributs utilitzareu.

2. Utilitzant l'esquema anterior, creeu algunes entrades al directori. És a dir, per als usuaris i els recursos de la figura, escriviu els atributs i els valors d'aquests atributs, que s'han d'emmagatzemar al directori.

3. Imagineu-vos que l'usuari Pepe també ha de pertànyer al grup de nòmines. Detalleu la solució, és a dir, els atributs i els valors d'aquests atributs, que s'han d'emmagatzemar al directori per als usuaris de la figura anterior.

4. Mireu d'obtenir i instal·leu un servidor de directoris basat en LDAP (l'OpenLDAP és un bon candidat). Amb l'eina JXplorer, accediu a aquest servidor i intenteu implementar la solució proposada en les activitats anteriors.

5. Usant un llenguatge de programació que conegueu, elaboreu una petita aplicació que permeti fer una llista d'entrades del directori segons paràmetres variats, com per exemple la classe d'objecte de les entrades de les quals cal fer una llista.

Glossari

àlies *m* Entrada que en realitat enllaça amb una altra entrada del directori.

atribut *m* Part de l'entrada destinada a guardar una peça d'informació, com per exemple un cognom o un número de telèfon.

base *f* Objecte o entrada des del qual, en l'operació de cerca específica, es vol començar a buscar informació.

cerca *f* Operació bàsica d'interrogació al servei del directori, mitjançant la qual s'obté informació conforme a una sèrie de criteris.

DAP *sigla* Vegeu **protocol d'accés a directoris**.

directori *m* Tipus especialitzat de base de dades jeràrquica que organitza i emmagatzema dades sobre elements (entrades de directori).

directory access protocol *m* Vegeu **protocol d'accés a directoris**.

distinguished name *m* Relació de DN relatius que identifiquen unívocament una entrada en un directori LDAP.
sigla **DN**

***distinguished name* relatiu** *m* Atribut i valor d'aquest atribut que identifiquen una entrada per a un nivell en concret de l'arbre del directori.

DN *sigla* Vegeu ***distinguished name***.

esquema *m* Definició de quins tipus d'objectes guarda un directori i els atributs usats en la definició dels objectes.

identificador d'usuari *m* Atribut específic per a guardar l'inici de sessió d'un usuari d'un sistema informàtic.
sigla **UE**

LDAP *sigla* Vegeu **protocol d'accés a directoris lleugers**.

lightweight directory access protocol *m* Vegeu **protocol d'accés a directoris lleugers**.

objectclass *m* Atribut d'una entrada de directori basat en LDAP que especifica a quina classe pertany l'entrada (per exemple, *person*).

protocol d'accés a directoris *m* Sistema primitiu d'implementació i gestió de directoris, basat en l'especificació X.500.
en directory access protocol
sigla **DAP**

protocol d'accés a directoris lleugers *m* Interfície entre clients de directori i sistemes DAP, que després va evolucionar cap a un servei de directori.
en light weight directory access protocol
sigla **LDAP**

servei de directori *m* Plataforma que proporciona mètodes per a gestionar i emmagatzemar les dades que conté el directori.

sufix *m* DN per a l'arrel de l'arbre de directori.

user identifier *m* Vegeu **identificador d'usuari**.

Bibliografia

Desmond, R. i altres (2008). *Active Directory: Designing, Deploying and Running Active Directory*. Sebastopol, CA: O'Reilly.

Howes, T. A. i altres (2003). *Understanding and Deploying LDAP Directory Services* (2a. ed.). Boston (Massachusetts): Addison-Wesley Longman.

Raya, J. L. i altres (2008). *Aprenda Microsoft Windows Server 2008*. Madrid: Ra-Ma Editorial.