Tutorial LDAP

```
cn: Marco Ferrante
mail: marco@csita.unige.it
o: Università di Genova
ou: CSITA

cn: Tiziana Podestà
mail: tiziana@csita.unige.it
o: Università di Genova
ou: CSITA
```

nota di copyright

introduzione

Nel corso del tutorial verranno trattati i seguenti argomenti:

-generalità del protocollo LDAP

-panoramica delle applicazioni LDAP enabled

-LDAP per l'autenticazione e la sicurezza

-procedure di installazione e configurazione di un server LDAP open-source

-un esempio di applicazione: sistema di posta

cos'è un servizio di *directory*

Un servizio di directory consente di dare dei nomi a degli oggetti e di associare agli oggetti degli attributi.

- Finger (rfc 742)
- CCSO Nameserver (Ph)
- NIS (YP)/NIS+
- DNS
- X.500

agenda

- applicazioni
- protocollo, struttura dati e operazioni
- autenticazione e autorizzazione
- strumenti e interscambio dati
- progettazione di un servizio
- OpenLDAP
- esempio

Lightweight Directory Access Protocol

LDAP è lo standard più diffuso per l'implementazione di directory *general purpose*

Fra le possibili applicazioni:

- *white pages* (elenco telefonico e indirizzario)
- autenticazione e autorizzazione
- routing posta elettronica
- distribuzione certificati X.509 e CRL
- persistenza di oggetti e classi Java (via JNDI)
- *backend* per altri servizi di directory
- memorizzazione di profili utente

white pages

I principali client di posta elettronica prevedono la connessione ad un server LDAP per le funzioni di rubrica

- Microsoft Outlook/Outlook Express
- Netscape Messenger/Mozilla/Thunderbird
- Eudora 5/6, Pegasus Mail, Sylpheed

Gateway LDAP

- Eudora Ph2LDAP Adapter (per Unix)

  http://www.eudora.com/techsupport/worldmail/ldap.html
- web2ldap (in Python)

  http://web2ldap.de/

server di autenticazione

Un servizio LDAP può essere usato per validare le credenziali utente per l'accesso alle workstation

- PAM_LDAP (Unix/Linux)
- pGina (Windows 2000/XP)

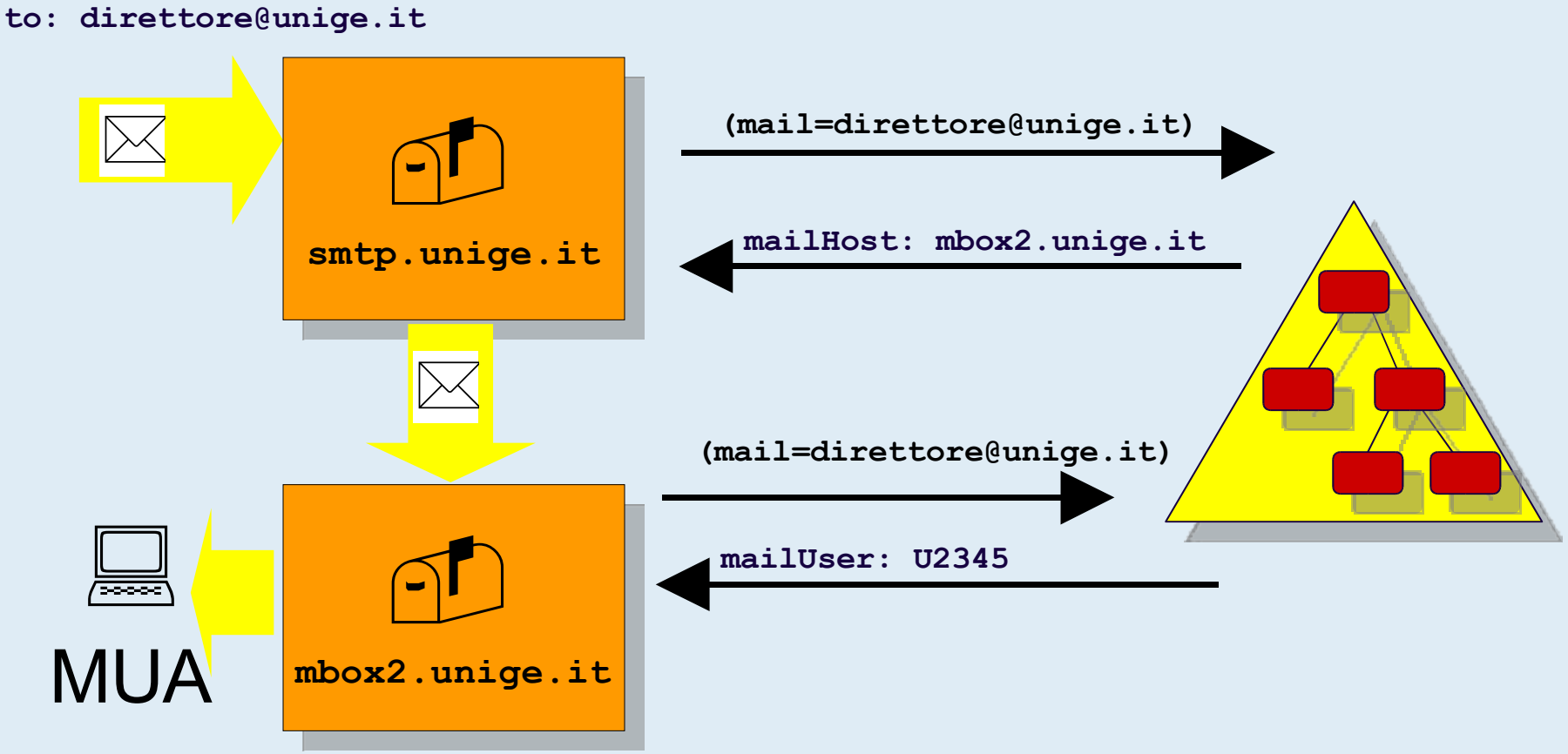
  http://pgina.xpasystems.com/

Applicazioni server possono utilizzare LDAP come backend per autenticazione e autorizzazione

- Apache + mod_auth_ldap
- Jakarta Tomcat
- uPortal

routing dei messaggi di posta

LDAP sostituisce il file “aliases” in modo distribuito



mail server

MTA con funzioni base

- sendmail versione 8
- PostFix

MTA con funzioni avanzate (memorizzano i profili utente completi o riscritture complesse)

- Netscape Messaging Server
- QMail-ldap

MDA

- Cyrus IMAPD
- CourierIMAP

applicazioni...

RADIUS server

- FreeRADIUS http://www.freeradius.org/
- Radiator http://www.open.com.au/radiator/

Server FTP

- Pure-FTPd http://www.pureftpd.org/

Web cache e proxy

- Squid http://devel.squid-cache.org/

Server SSH

- OpenSSH http://ldappubkey.gcu-squad.org

...applicazioni...

Server SMB

- Samba http://www.samba.org/

Gateway LDAP-DNS

- LdapDNS http://www.nimh.org/code/
- ldap2dns http://ldap2dns.tiscover.com/

PKI

- OpenCA http://www.openca.org/
- IDX-PKI http://idx-pki.idealx.org/
- pyCA http://www.pyca.de/

...applicazioni

Gestione di mailing list

- Sympa http://www.sympa.org/

Controllo remoto

- Timbuktu

agenda

- ✔ applicazioni
- protocollo, struttura dati e operazioni
- autenticazione e autorizzazione
- strumenti e interscambio dati
- progettazione di un servizio
- OpenLDAP
- esempio

LDAP

LDAP è un protocollo di accesso, in modalità client-server, a servizi di directory

LDAP specifica le modalità di:

- connessione: *open(), bind(), unbind()*
- confronto di attributi: *compare()*
- ricerca di oggetti: *search()*
- modifica degli oggetti: *add()*, *delete()*, *modify()*
- operazioni sui nomi: *modifyDN()*
- operazioni estese

versioni LDAP

LDAPv2 [rfc 1777]

- Caratteri codificati T.61
- Obbligo di autenticazione dopo la connessione

LDAPv3 [rfc 2251-2256]

- Caratteri codificati UTF-8
- Gestione dei referral da parte dei client
- “autodescrizione”

basi del protocollo

I messaggi sono descritti in formato ASN.1 (Abstract Syntax Notation) e codificati BER

LDAP usa TCP e mantiene le connessioni

Le operazioni possono essere asincrone e concorrenti

oggetti

Un oggetto (*entry*) LDAP è costituito da un insieme di coppie attributo-valore (AVA *Attribute Value Assertion*)

un attributo particolare (objectClass) definisce il comportamento strutturale dell'oggetto in termini di attributi obbligatori e ammessi

```
objectClass: top
objectClass: person
objectClass: organizationalPerson
cn: Marco
sn: Ferrante
ou: csita
mail: marco@csita.unige.it
```

attributi

LDAP utilizza la semantica X.500 per gli attributi degli oggetti

Per ogni attributo sono specificati:

- OID (*Object IDentifier*) IANA
- nome e eventuali alias
- significato
- sintassi
- regole di confronto
- molteplicità (uno o molti)

esempi attributi

```
attributetype ( 2.5.4.6
NAME ('c' 'countryName')
DESC 'RFC2256: ISO-3166 country 2-letter code' SUP
name SINGLE-VALUE )

attributetype (2.5.4.4 NAME ('sn' 'surname')
DESC 'RFC2256: last (family) name(s) for which the
entity is known by' SUP name)

attributetype ( 2.5.4.23
NAME ('facsimileTelephoneNumber' 'fax')
DESC 'RFC2256: Facsimile (Fax) Telephone Number'
SYNTAX 1.3.6.1.4.1.1466.115.121.1.22 )
```

classi

LDAP utilizza la semantica X.500 per le classi di oggetti

per ogni classe sono specificati:

- OID
- descrizione
- categoria (astratta, strutturale, ausiliaria)
- attributi obbligatori
- attributi opzionali

esempio classe

```
objectclass ( 2.5.6.5 NAME
'organizationalUnit'
DESC 'RFC2256: an organizational unit'
SUP top STRUCTURAL
MUST ou
MAY ( userPassword $ searchGuide $ seeAlso $
businessCategory $
x121Address $ registeredAddress $
destinationIndicator $
preferredDeliveryMethod $ telexNumber $
teletexTerminalIdentifier $
telephoneNumber $ internationaliSDNNumber $
facsimileTelephoneNumber $ ... ) )
```

schema

L'insieme delle definizioni di attributi e classi prende il nome di schema

Lo schema è specifico dell'installazione

Le modifiche agli oggetti che violano lo schema verranno rifiutate

Alcuni sistemi rafforzano i vincoli di schema

- Novell eDirectory
  - specifica le classi che ammettono discendenti
- OpenLDAP 2.1.x
  - limita a uno le classi strutturali per oggetto

inetOrgPerson

person
SUP top STRU
sn
cn
userPasswor
telephoneNum
seeAlso
description

organizationalPerson
SUP person STRUCTURAL
title
x121Address
registeredAddr
destinationInd
preferredDeliv
telexNumber
teletexTermina
internationali
facsimileTelep
street
postOfficeBox
postalCode
postalAddress
physicalDeliv
ou
st
l

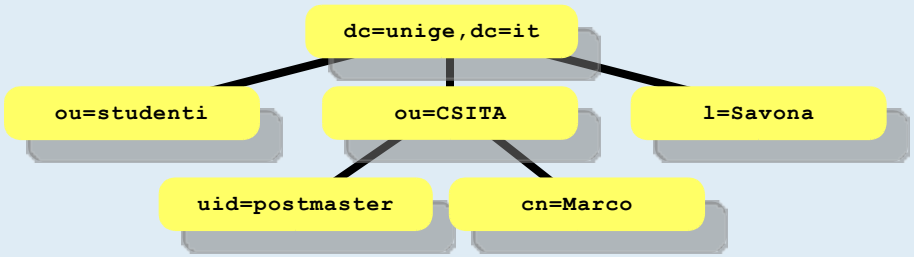
inetOrgPerson
SUP organizationalPerson STRUCTURAL
audio
businessCategory
carLicense
departmentNumber
displayName
employeeNumber
employeeType
givenName
homePhone
homePostalAddress
initials
jpegPhoto
labeledURI
mail
manager
mobile
o
pager
photo
roomNumber
secretary
preferredLanguage
uid
userCertificate
userPKCS12
x500uniqueIdentifier
userSMIMECertificate

struttura dei dati

LDAP organizza i dati in modo gerarchico

Ogni oggetto può avere al più un genitore, mentre non ci sono limiti al numero di figli

L'albero delle entry prende il nome di DIT (*Directory Information Tree*)



Per ogni servizio,

viene identificato un elemento radice

- c=IT
- dc=garr,dc=it

entry speciali

I server LDAPv3 forniscono alcune entry al di fuori della radice

- Root DSE (base “”)

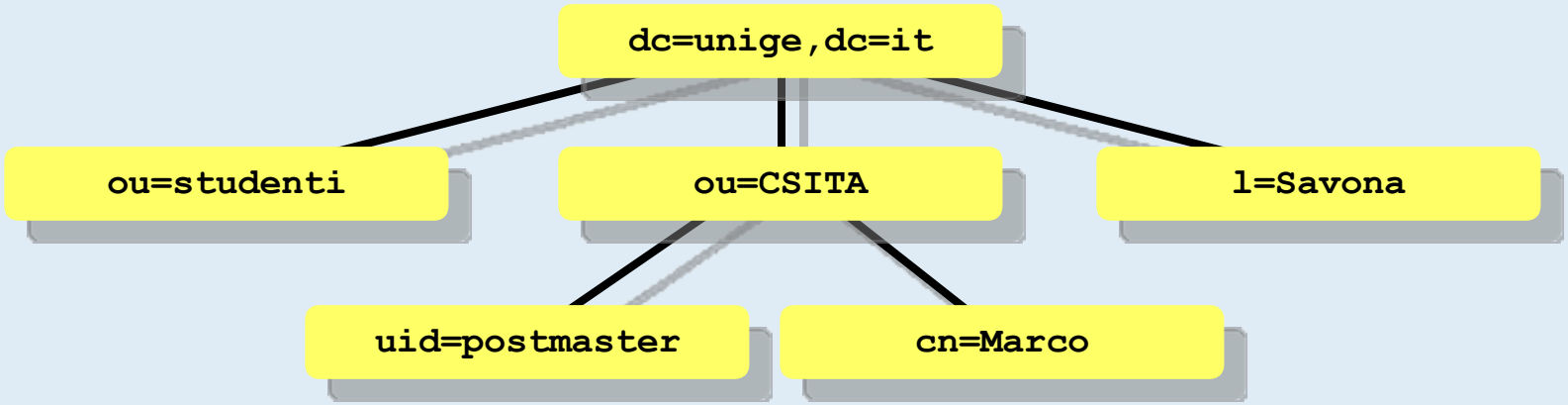
```
namingcontexts: dc=unige,dc=it
changelog: cn=changelog
supportedldapversion: 2
supportedldapversion: 3
subschemasubentry: cn=schema
supportedsaslmechanisms: EXTERNAL
```

- schema

RDN

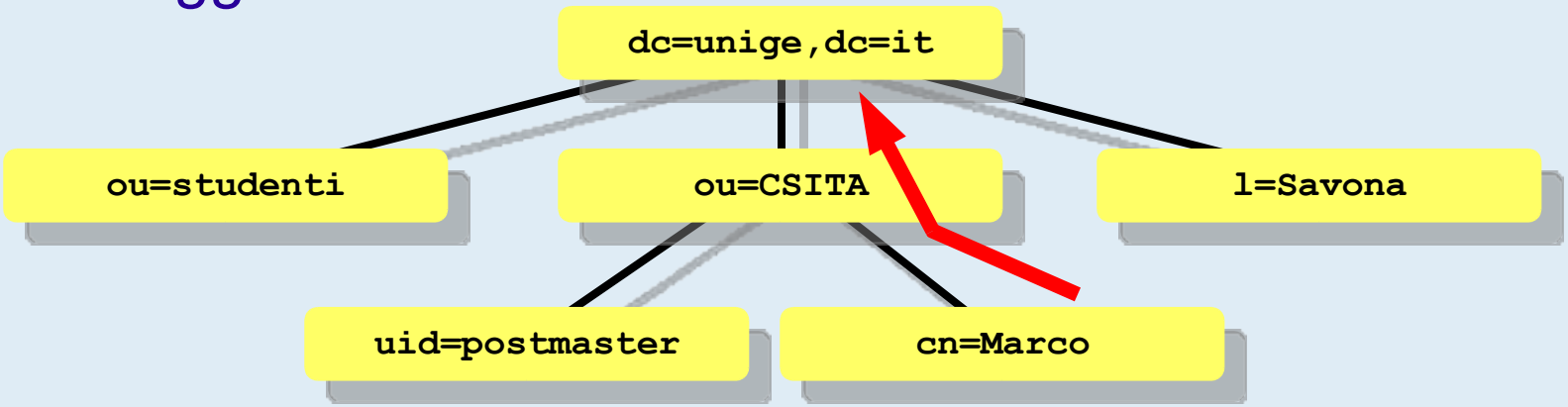
Ogni oggetto ha un AVA (o più) elettivo (RDN *Relative Distinguished Name*)

Tutti figli di un determinato nodo devono avere un RDN differente



DN (*Distinguished Name*)

La sequenza degli RDN forma il nome univoco dell'oggetto



Tradizionalmente, il DN LDAP viene scritto come la sequenza ordinata degli RDN letti dalla foglia alla radice

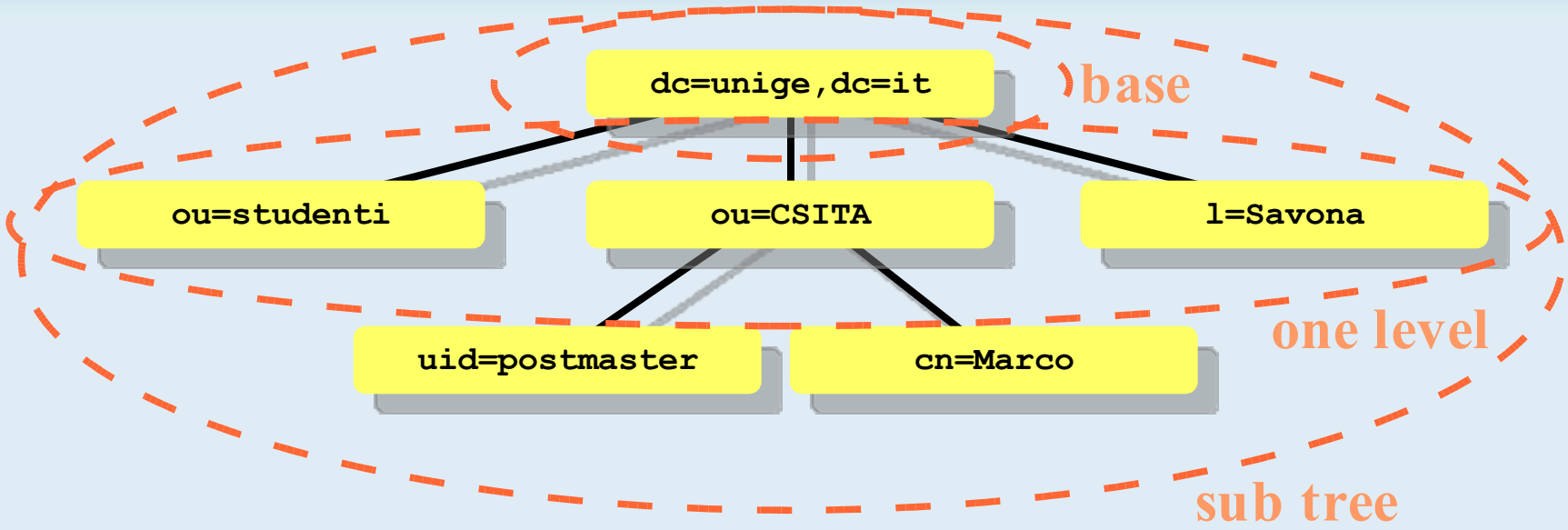
```
cn=Marco,ou=csita,dc=unige,dc=it
```

ricerca e lettura sul DIT

Per cercare le entry che soddisfano determinate condizioni si specifica in un *search()*

- da dove iniziare la ricerca (base)
- l'ambito della ricerca (scope)
- il criterio (filtro)
- gli attributi da restituire o null per tutti gli attributi
  - alcuni attributi potrebbero non essere accessibili
  - alcuni attributi sono restituiti solo se specificati

scope di ricerca



base: solo nella entry specificata

one level: solo nei figli diretti della entry specificata

subtree: nella entry specificata e in tutti i suoi discendenti diretti e indiretti

filtri

La sintassi del filtro è in notazione prefissa

operatori logici: **& | !**

comparatori: **= <= >= ~=** (somiglianza fonetica)

carattere jolly: *

ricerca per nome:

```
(cn=Marco Ferrante)
```

tutti coloro che hanno la posta sul dominio:

```
(mail=*@unige.it)
```

filtri complessi

numeri di telefono:

```
(telephoneNumber=*0103532102)
telephoneNumber: +39 010 353-2102
```

nome o cognome “bruno”:

```
(|(sn=bruno)(givenName=bruno))
```

posta sul server senza forward:

```
(&(mailHost=mbox.unige.it)
  (!(mailForwardingAddress=*)))
```

risultati particolari

Di norma, un *search()* restituisce l'elenco di risultati o un errore

Se il server non è “autoritativo” per la base richiesta, può restituire un *referral*

```
ldap://root.openldap.org:389
```

Se il server supporta gli alias, essi vengono restituiti direttamente o risolti a seconda delle opzioni di connessione

```
objectClass: alias
aliasedObjectName: cn=pippo,dc=unige,dc=it
```

operazione di connessione

L'accesso ad una directory LDAP inizia con un operazione di connessione *open()* o *init()*

Nei più diffusi SDK è possibile specificare:

- più server scelti in *round robin*
  - se copie dei dati si trovano su diversi server (repliche)
- un URL ldap://server/base
- l'uso SSL

Al termine delle operazioni, occorre un'operazione di disconnessione

operazioni sulle *entry*

L'aggiunta di una entry ad un DIT si ottiene con un'operazione

```
add(<target DN>, <lista attributi>)
<attributi> = <nome>, <insieme valori>
```

gli attributi devono contenere i valori distintivi nell'RDN

La cancellazione di una entry

```
delete(<target DN>)
```

operazioni sugli attributi

La modifica degli attributi di una entry si ottiene con l'operazione

```
modify(<target DN>, <lista modifiche>)
<modifica> = <[add|modify|delete]>
<attributo>
<attributo> = <nome> <insieme valori>
```

LDAP non prevede la transazionalità ma tutte le modifiche di un set vengono applicate in un'unica operazione atomica

Una violazione di schema lascerà il DIT in uno stato consistente

modifica del DN

L'operazione per modificare il DN di una entry è

```
modifyDN(<old RDN>, <new RDN>,
  <nuovo genitore DN>, <cancella old RDN>)
```

Secondo specifiche, tutte le combinazioni di rinominazione/spostamento della entry dovrebbero essere possibili

In pratica, la maggior parte dei server accettano solo null come DN del nuovo genitore

agenda

- ✔ applicazioni
- ✔ protocollo, struttura dati e operazioni
- ■ autenticazione e autorizzazione
- ■ strumenti e interscambio dati
- ■ progettazione di un servizio
- ■ OpenLDAP
- ■ esempio

autenticazione (*bind*)

Nella fase di *bind()*, vengono presentate al server le credenziali per l'accesso

Le credenziali identificano direttamente o indirettamente un oggetto del DIT

- direttamente: DN, userPassword
- indirettamente: mappa di un certificato X.509

Molti server LDAP supportano SASL (*Simple Authentication and Security Layer*), ma le forme più robuste di autenticazione richiedono che le password siano conservate in chiaro

userPassword

La password del DN è memorizzata in uno specifico attributo standard

Hanno un comportamento omogeneo con gli altri attributi, quindi va protetto in modo particolare

Di norma, le password sono memorizzate dopo essere state sottoposte ad una funzione di *hash*

Per permettere al server il confronto, l'algoritmo usato è specificato esplicitamente

```
userPassword:
  {SHA}axB9Sy8te6HQkOBo22y1yMV8oaOb=
```

precauzioni nell'uso della password

Le password hanno alcuni comportamenti peculiari

- accedere con una password nulla equivale all'accesso anonimo
- le password devono essere codificate dal client; il server le tratta come normale testo
- le password possono essere multiple

Alcuni programmi mal progettati tentano erroneamente il confronto della password per l'autenticazione

autorizzazione LDAP

La terna

<il soggetto può compiere l'azione sull'oggetto>

è particolarmente potente su LDAP perché

- il soggetto è una entry (dalle credenziali)
- l'azione è un'operazione LDAP
- l'oggetto è un insieme di entry specificato in stile LDAP

Le ACL prendono quindi la forma generale

<DN oggetto; operazione LDAP; LDAP URL>

ACL semplice

Il formato e l'espressività delle ACL non è standard:

OpenLDAP (in *slapd.conf*)

```
access to attr=userpassword
    by self write
    by anonymous auth
```

Netscape Directory Server (nella *root entry*)

```
aci: (target="ldap:///dc=unige,dc=it")
  (targetattr=userPassword)
  (version 3.0;acl "ProteggiPassword";
  allow (write) userdn="ldap:///self";)
```

gruppi

Un attributo può contenere un DN

Entry gruppo:

```
dn: cn=admin group, dc=unige,dc=it
objectClass: groupOfNames
member: cn=marco,ou=CSITA,dc=unige,dc=it
member: cn=tiziana,ou=CSITA,dc=unige,dc=it
```

Entry ruolo:

```
dn: cn=direttore,ou=CSITA,dc=unige,dc=it
objectClass: organizationalRole
roleOccupant:
 cn=Luca,ou=CSITA,dc=unige,dc=it
```

gruppi e ruoli

Nelle regole delle ACL è possibile specificare che il DN della connessione deve comparire in un attributo di una entry

Inoltre, le ACL prevedono il soddisfacimento di un filtro, concedendo l'autorizzazione a determinate categorie di entry

Si possono comporre regole che assegnino specifici privilegi a utenti di determinati gruppi o in determinati ruoli

regole con ACL

Ogni *owner* può amministrare i membri del proprio gruppo (Netscape Directory Server)

```
(targetattr="member")(version 3.0;
  acl "Owner dei gruppi"; allow (all)
  groupdnattr=
    "ldap:///ou=staff,dc=unige,dc=it?owner"; )
```

Il direttore può cancellare un utente dal dipartimento (OpenLDAP 1.x)

```
access to dn.regex="^(.*),(.*),dc=unige,dc=it$$"
by group/organizationalRole/roleOccupant.regex=
  "^cn=direttore,$2,dc=unige,dc=it$$" write
```

LDAP per validare gli accessi

Un servizio di directory può essere usato come *repository* degli utenti per applicazioni server

Gli utenti useranno le stesse credenziali per diversi servizi

Putroppo, gli utenti difficilmente ricordano il proprio DN e, in ogni caso, esso è troppo lungo

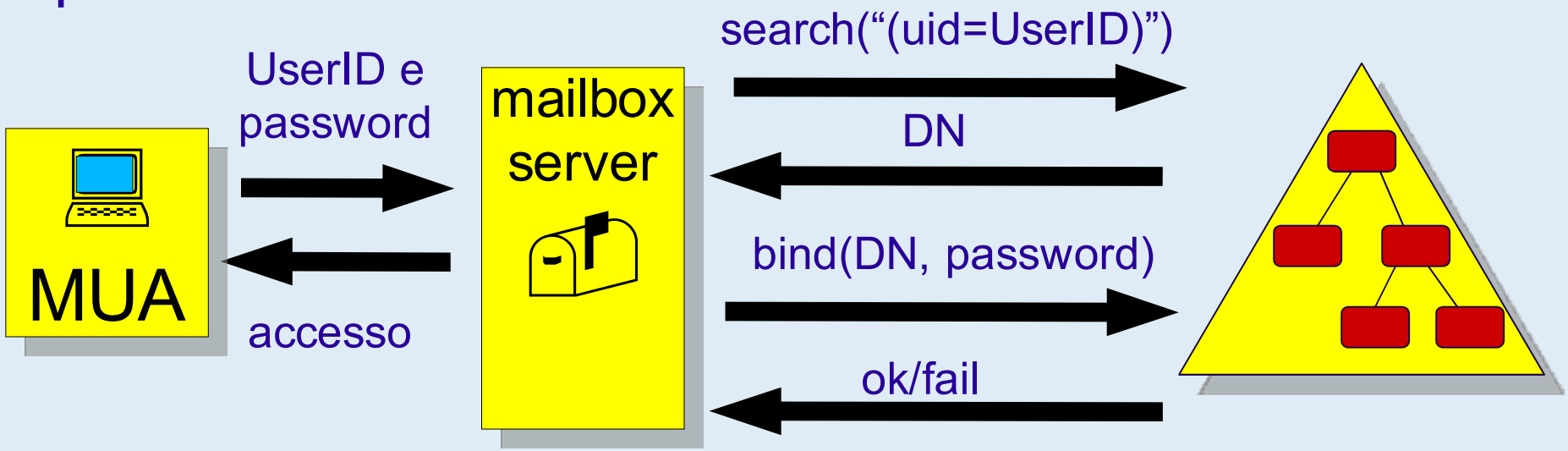
Le applicazioni server possono utilizzare un sistema di autenticazione in due fasi

autenticazione in due fasi

L'utente fornisce un valore univoco (es. uid o mail)

Il server si connette e cerca la entry con questo attributo

Il server tenta il *bind()* con il DN trovato e la password utente



autorizzazione per applicazioni server

I server che usano LDAP per l'autenticazione possono utilizzarlo anche per generazione di un profilo di autorizzazione

filtri aggiuntivi

```
(&(uid=%1)(objectClass=unigePerson))
```

assegnazione a un ruolo (RBA)

```
administrator ⇔ (ou=manager)
```

appartenenza a gruppi

agenda

- ✔ applicazioni
- ✔ protocollo, struttura dati e operazioni
- ✔ autenticazione e autorizzazione
- ■ strumenti e interscambio dati
- ■ progettazione di un servizio
- ■ OpenLDAP
- ■ esempio

importazione e esportazione dati

L'input e l'output degli strumenti per LDAP prevede l'uso del formato testuale standard LDIF (*LDap Interexchange Format*)

```
dn: cn=Tiziana, ou=CSITA, dc=unige, dc=it
cn: Tiziana
sn:: UG9kZXN0w6A=
mail: tiziana@csita.unige.it
userPassword: {SHA}
 xB9Sy8te6HQkOBo22y1yMV8oaOb=
                              riga vuota
```

I codici non ASCII sono codificati Base64

LDIF esteso (*replog*)

Un file LDIF può contenere la sequenza delle operazioni da applicare

```
dn: cn=Ford Perfect,dc=esempio,dc=it
changetype: add
objectclass: person
cn: Ford
cn: Ford Perfect
sn: Perfect

dn: cn=Ford Perfect,dc=esempio,dc=it
changetype: modify
add: description
description: scrittore di guide
```

strumenti

Tool “standard” a riga di comando

- ldapsearch

  ldapsearch -D <bind DN> -b “dc=unige,dc=it”

  -s sub “(mail=marco@csita.unige.it)” [attributi]

  -L esporta il risultato in formato LDIF

- ldapadd

  importa in file LDIF

- ldapmodify

  applica le modifiche indicate in file LDIF esteso

replog e repliche

Il formato *replog* è supportato dal tool *ldapmodify*

Il programma *slurpd*, parte della distribuzione OpenLDAP, utilizza un file *replog* generato runtime da *slapd*, per sincronizzare le repliche

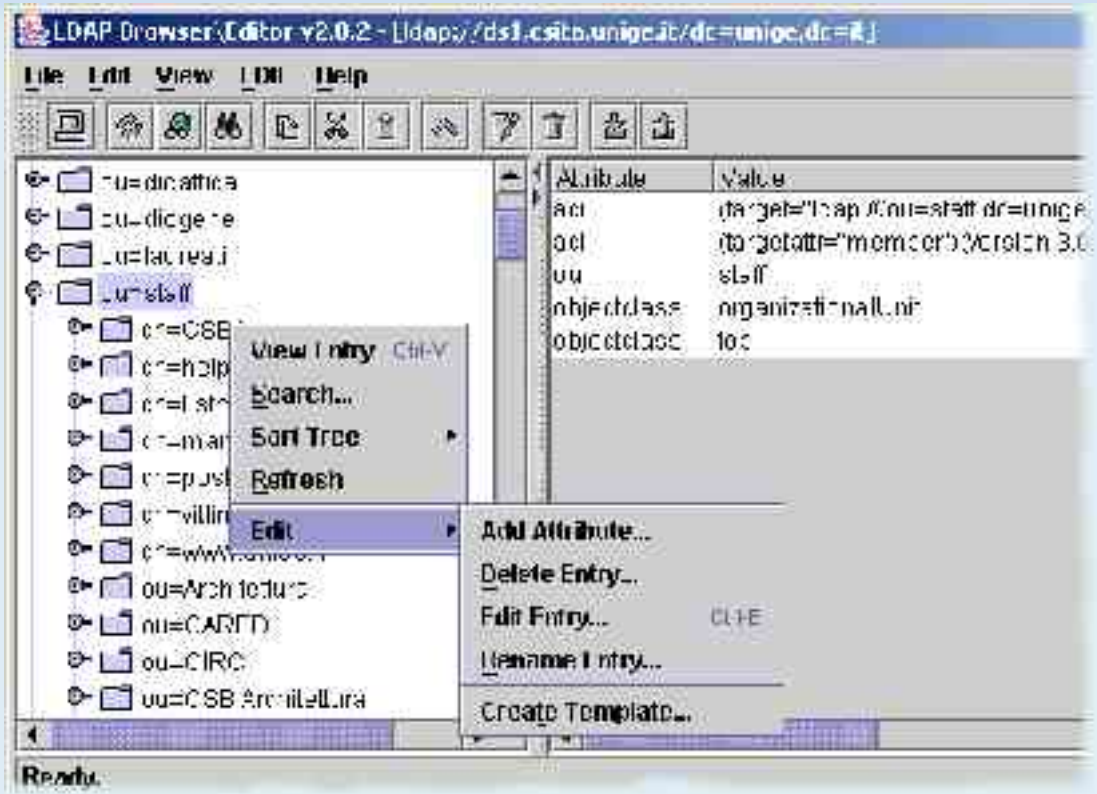
Poiché *slurpd* opera con il protocollo LDAP stesso, è possibile utilizzare una master *slapd* con repliche di altre implementazioni

Sono state verificate come funzionanti repliche verso Netscape Directory Server 4.16 e Novell eDirectory 6

LDAP Browser\Editor

LDAP Browser\Editor è un'interfaccia grafica per la gestione di directory LDAP

Utilizzabile su sistemi con Java



http://www.iit.edu/~gawojar/ldap/

DSML

DSML (*Directory Services Markup Language*) è un alternativa XML a LDIF

Gli equivalenti DSML di ldapadd e ldapsearch sono disponibili sul sito http://www.dsmltools.org/

DSML 2 prevede anche un protocollo di accesso
Al momento è implementato nativamente da Sun Directory Server 5.2 e, attraverso un gatweay SOAP, da Microsoft Active Directory

agenda

- ✔ applicazioni
- ✔ protocollo, struttura dati e operazioni
- ✔ autenticazione e autorizzazione
- ✔ strumenti e interscambio dati
- progettazione di un servizio
- OpenLDAP
- esempio

organizzazione del DIT

- Una prima scelta irreversibile è il DN della radice (*naming context*)
- Un server può ospitare più di una radice, ma in generale, ogni DIT avrà vita separata dagli altri
- Sotto la radice, il DIT viene partizionato in rami
- Nei rami occorre registrare gli oggetti con uno schema di denominazione coerente
- Le entry devono essere aggregate
- Esigenze particolari possono richiedere l'estensione dello schema di base

radice stile X.500

Tradizionalmente la radice viene scelta:

`o=<organizzazione>,c=<codice paese>`

esempio: `o=Università di Genova,c=it`

Pro:

- compatibile X.500 e quindi X.509
- default per alcune implementazioni

Contro:

- “o” dev'essere registrato presso ISO (a pagamento)
- “o” non è sempre univoco o concorde
  - es. Università degli Studi di Genova
- caratteri “problematici”

radice stile RFC 2247

Recentemente si è diffusa la scelta:

`[dc=...,]dc=<dominio 2 livello>,dc=<TLD>`

oppure

`dc=<dominio 2 livello>.<TLD>`

esempio: **`dc=unige,dc=it`** oppure **`dc=unige.it`**

Pro

- non richiede registrazione
- si integra con il DNS mediante i record SRV, quindi
  - autodiscovery dei server dal DN
  - recupero di certificati dall'indirizzo email

partizionamento del DIT

Di norma, le entry non vengono registrate tutte direttamente sotto la radice

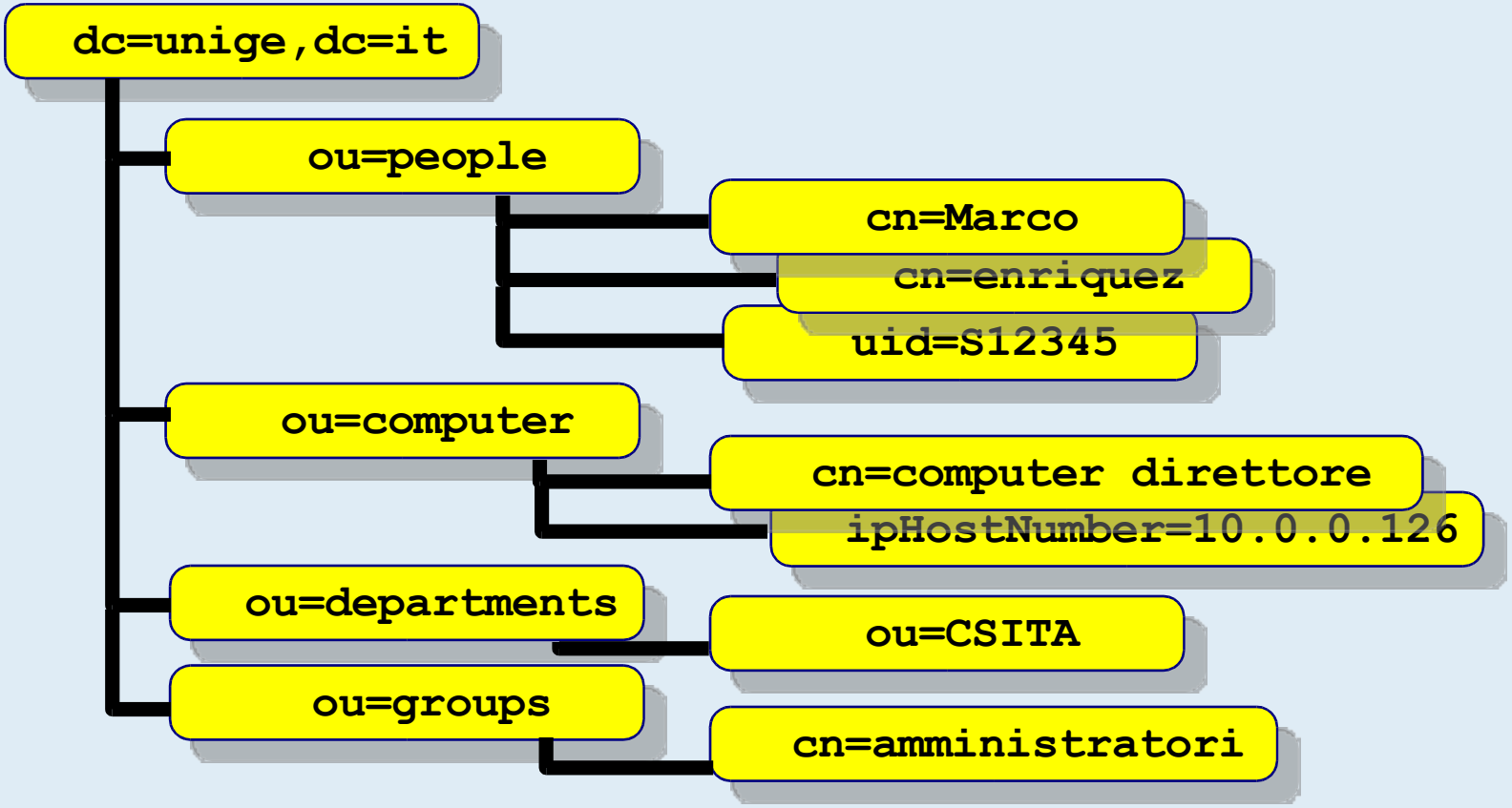
Per motivi di maneggevolezza e organizzazione si suddivide l'albero in diversi rami principali mediante entry di classe *organizationalUnit* o *locality*

Anche in questo caso, esistono due scuole di pensiero:

- partizionamento per oggetti omogenei
- analogia con la struttura organizzativa reale

partizionamento per classi omogenee...

I rami principali possono essere organizzati in modo da contenere oggetti di classe omogenea.



...partizionamento per classi omogenee

Dato che gli oggetti non cambiano di classe (una persona non diventa un computer), minimizza i problemi legati a cambiamenti di status (es. una persona che cambia dipartimento)

D'altra parte, molti oggetti non sono facilmente classificabili (es. un ruolo) e le associazioni diventano difficili (es. afferenza)

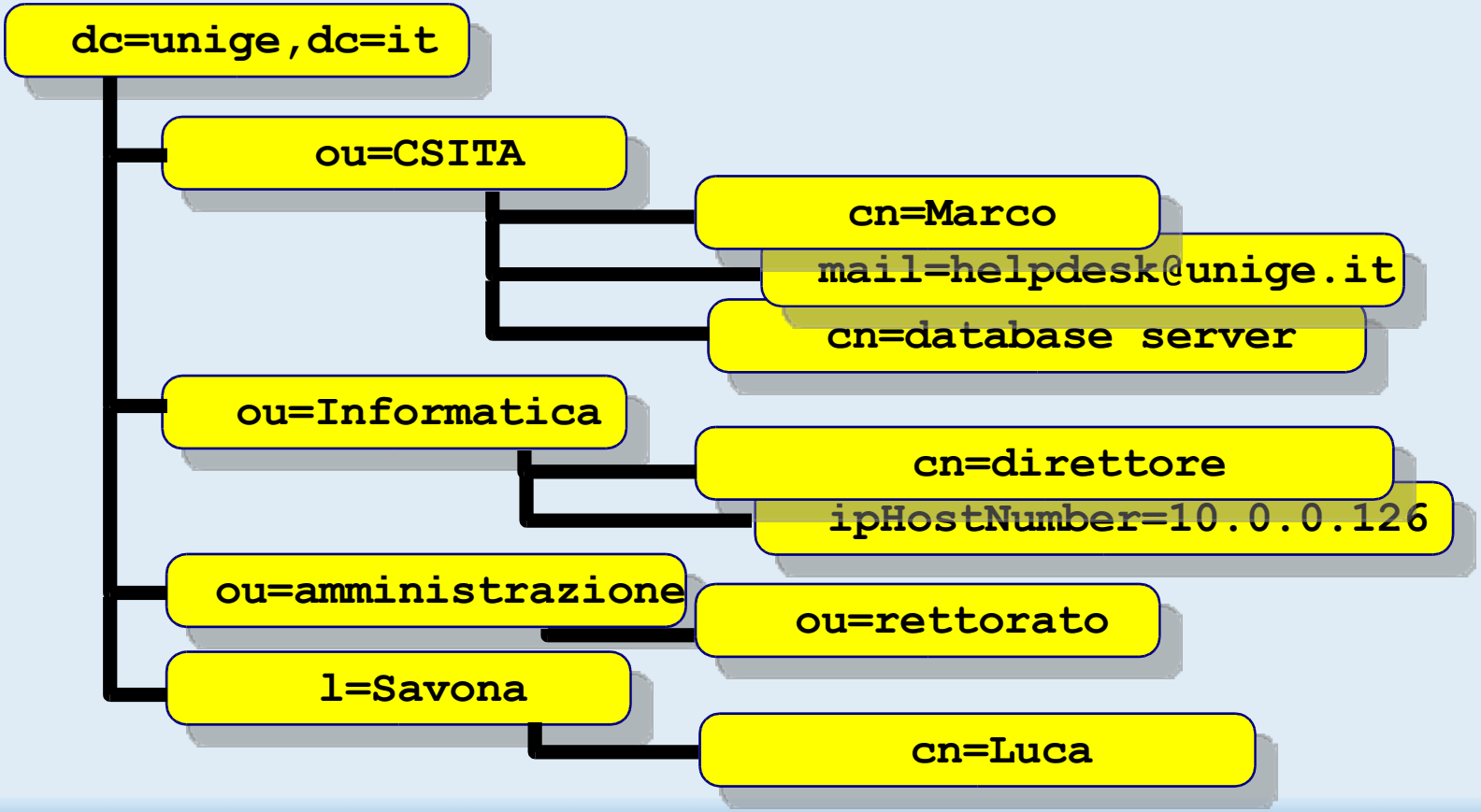
Molte applicazioni richiedono questo tipo di struttura perché applicano mappature tipo:

```
cn=%1,ou=%2,dc=unige,dc=it
```

per cercare gli utenti

partizionamento per organigramma...

I rami principali possono essere organizzati in modo da riflettere l'organizzazione dell'ente



...partizionamento per organigramma

Permette di raggruppare elementi assortiti mantenendo le relazioni

Risolve i problemi con gli oggetti con più aspetti

```
ou=csita,dc=unige,dc=it
objectClass: organizationalUnit
objectClass: dcObject
```

È necessario per delegare l'amministrazione o il servizio stesso alle strutture

Introduce molti problemi per lo spostamento di oggetti, ma non è detto che sia un comportamento scorretto

nomi delle *entry*

Ogni AVA è candidato a diventare l'RDN della *entry*

Una denominazione coerente semplifica l'amministrazione. Un buon candidato:

- è obbligatorio (strutturalmente o per convenzione)
- è stabile
- è *collision free*
- usa caratteri ASCII base

cn=Marco Ferrante

uid=P1234

mail=marco@unige.it

~~cn=Marco~~

~~employeeNumber=123~~

~~ou=Istituto di Fisica~~

DN e *accounting*

LDAP permette un archivio unico di utenti

Nello stile RFC 2247, il DN è unico a livello globale

La correlarezione tra utenti di sistemi diversi semplifica le funzioni di *accounting*

Se la topologia del DIT prevede modifiche ai DN è necessario tenerne traccia storica

```
dn: cn=zaphod,ou=Informatica,dc=unige,dc=it
objectClass: unigeLoginProperties
formerDN: cn=zaphod,ou=Fisica,dc=unige,dc=it
```

aggregare le *entry*

Le aggregazioni svolgono un ruolo essenziale per le funzioni di autorizzazione

Possibili criteri di aggregazione:

- soddisfacimento di un filtro
  - discendenti o figli di uno stesso nodo
  - possesso di un determinato AVA
- oggetti con DN elencati in un'altra entry
  - gruppi
  - ruoli

raggruppamenti per enumerazione

Raggruppamenti particolari richiedono oggetti di classe groupOfName o organizationalRole

L'uso di queste aggregazioni richiede due fasi di interrogazione. A seconda dei casi, le applicazioni devono:

- applicare un confronto con il DN utente (ad esempio, per concedere autorizzazioni)
- leggere un attributo dalla entry indicata (ad esempio, per generare delle liste di posta)

raggruppamenti per proprietà

Una proprietà può essere usata per generare mailing list o report

I criteri possono essere memorizzati in una entry

```
dn: cn=utenti,dc=unige,dc=it
memberURL: ldap://dc=unige,dc=it??sub?
   (userPassword=*)
```

Sun Directory Server, IBM SecureWay e Novell eDirectory supportano i gruppi dinamici nelle ACL

backlink

Se una entry è elencata in un gruppo

```
dn: cn=docenti,dc=unige,dc=it
uniqueMember:
   uid=1234,ou=staff,dc=unige,dc=it
```

sarebbe comodo che la entry mantenesse un riferimento al gruppo di appartenenza (*backlink*)

```
dn: uid=1234,ou=staff,dc=unige,dc=it
memberOf: cn=docenti,dc=unige,dc=it
```

Nessun LDAP server provvede automaticamente alla gestione dei *backlink*

È possibile scrivere un plugin apposito

integrità referenziale dei gruppi

In un DIT a organigramma

```
dn: cn=preside,ou=Lettere,dc=unige,dc=it
roleOccupant:
  uid=1234,ou=Lettere,dc=unige,dc=it
```

in caso di cambio di facoltà, si creerà un errore di riferimento. Alcuni server gestiscono automaticamente il cambiamento, assegnando ad ogni oggetto un identificatore univoco

Non sempre questo comportamento è desiderabile: ad un cambio di ramo può corrispondere una perdita di prerogative

estendere lo schema

Esigenze particolari possono non trovare le classi o gli attributi adatti nello schema di default del server

Lo schema può essere esteso, con modalità che variano a seconda delle implementazioni

Attenzione: su alcuni server, l'estensione dello schema è un'operazione irreversibile!

La progettazione di un nuovo schema è un'operazione complessa che richiede un'approfondita conoscenza del domino dei dati da rappresentare, dei dettagli specifici del server e del funzionamento di LDAP in generale

cataloghi di schemi

Prima di creare un'estensione proprietaria, si può consultare un catalogo di schemi

- **DAASI**
  http://www.daasi.de/home-e.html
- **Educause**
  http://www.educause.edu/eduperson/
- **Linux Center LDAP Schema Repository**
  http://ldap.akbkhome.com/
- **IBM LDAP Directory Schema**
  http://www-1.ibm.com/servers/eserver/iseries/ldap/schema/
- **Microsoft Active Directory Schema**
  http://msdn.microsoft.com/library/en-us/adschema/adschema/active_directory_schema.asp
- **Netscape Universal Schema Reference**
  http://developer.netscape.com/docs/manuals/directory/schema2/41/contents.htm
- **Novell NDS Schema Reference**
  http://developer.novell.com/ndk/doc/ndslib/schm_enu/data/h4q1mn1i.html

schema eduPerson

```
( 1.3.6.1.4.1.5923.1.1.2
  NAME 'eduPerson' AUXILIARY
  MAY (
eduPersonAffiliation $
eduPersonNickname $
eduPersonOrgDN $
eduPersonOrgUnitDN $
eduPersonPrimaryAffiliation $
eduPersonPrincipalName $
eduPersonEntitlement $
eduPersonPrimaryOrgUnitDN ) )
```

OID (*Object IDentifier*)

Internamente, un server LDAP non usa i nomi delle classi o degli attributi

Tutte le operazioni vengono svolte attraverso gli OID. Un OID è una sequenza tipo

```
1.3.6.1.4.1.4203
```

Non usate OID di fantasia o copiati da altri!

La gestione delle sequenza usa un sistema di delega. L'assegnazione di un OID per la propria organizzazione può essere richiesta, gratuitamente per usi interni, a IANA per scopi relativi a SNMP

dimensionamento hardware

LDAP non è particolarmente avido di risorse

I dati del servizio dell'Università di Genova

- Sun Netra CPU Sparc 400 MHz (dedicato)
- Netscape Directory Server 4.16
- 1 Gbyte di RAM

Un server gestisce

- ~ 12.000 search/ora
- ~ 7.000 bind/ora
- ~ 45.000 entry
- ~ 250 Mbyte di dati su disco (31 indici)

architettura hardware

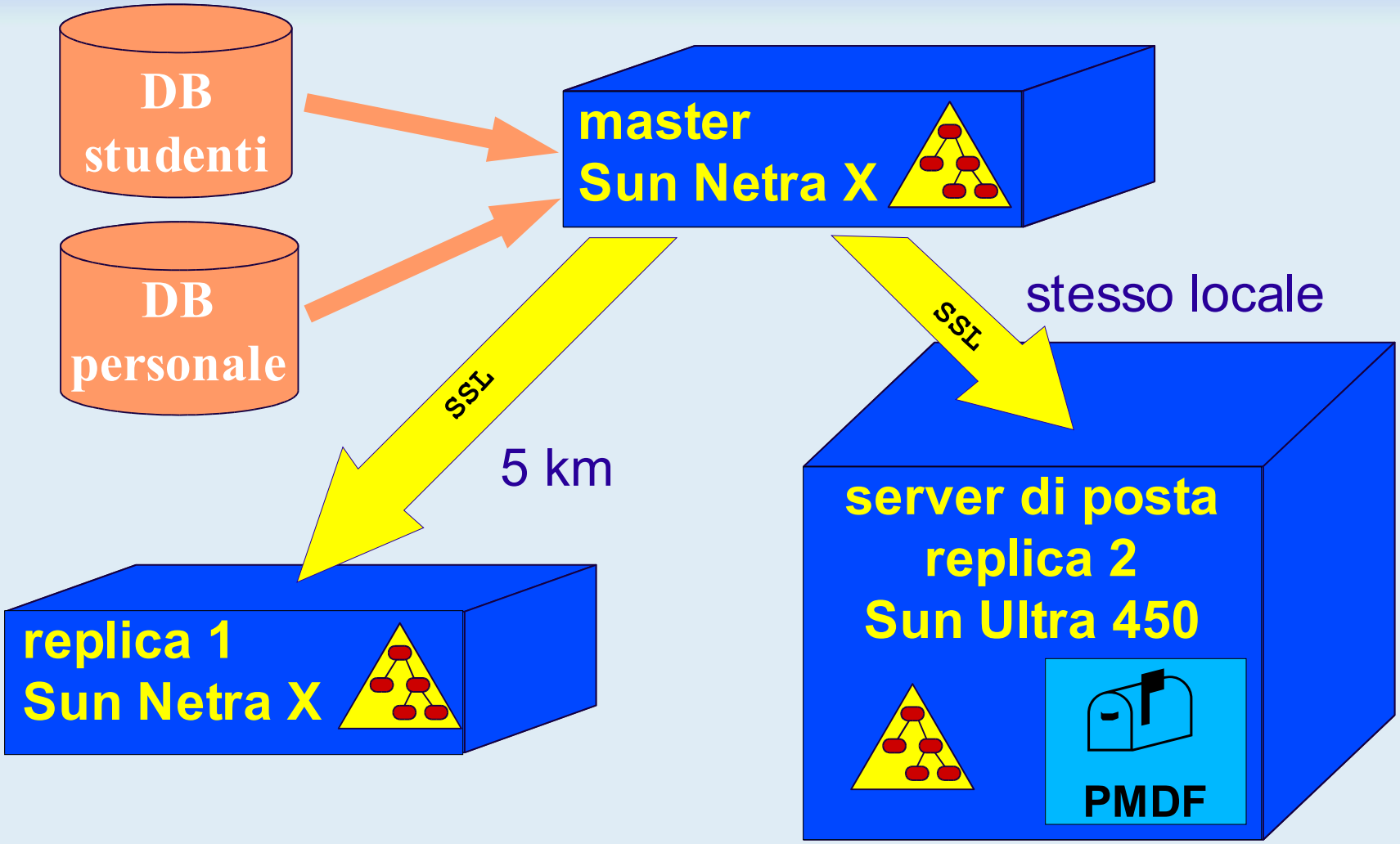
LDAP usato per l'autenticazione costituisce un servizio critico

LDAP prevede meccanismi di ridondanza dei server in modalità master/replica

- un solo server accetta le modifiche (master)
  - alcune implementazioni sono multimaster
- la sincronizzazione delle repliche può avvenire in modalità push o pull

Il meccanismo di replica non è standardizzato

sistema Directory di Ateneo



uso delle repliche

I principali SDK per LDAP prevedono la possibilità di specificare più di un server (separati da spazi) a cui tentare il collegamento

LDAPv3 prevede l'implementazione distribuita

- se il server non dispone dei dati richiesti, può restituire il referral del server che li mantiene
- una replica può indicare il master come referral in risposta a un tentativo di scrittura

implementazioni

Le principali implementazioni server disponibili sono:

- OpenLDAP (open source)
- Sun (Netscape/iPlanet) Directory Server
- Microsoft Active Directory
- Novell NDS e eDirectory
- Oracle Internet Directory (OID)
- IBM SecurWay

agenda

- ✔ applicazioni
- ✔ protocollo, struttura dati e operazioni
- ✔ autenticazione e autorizzazione
- ✔ strumenti e interscambio dati
- ✔ progettazione di un servizio
- ▪ OpenLDAP
- ▪ esempio

OpenLDAP

OpenLDAP (http://www.openldap.org/) è un progetto open-source che ha prodotto o mantiene:

- server LDAPv2 (openldap 1.x)
- server LDAPv3 (openldap 2.x)
- SDK in C per i client LDAP
- librerie Java per LDAP (contributo Novell)
- bridge JDBC/LDAP (contributo OctetString)

Il server OpenLDAP 2.x è utilizzabile su piattaforme Linux/Unix

È compilabile su Windows con CygWin

versioni OpenLDAP

Versione 2.0.x

- primo supporto per LDAPv3

Versione 2.1.x

- implementazione del draft di LDAPv4/LDAP bis

Versione 2.2.x

- in beta

installazione di OpenLDAP

Dipendenze:

- Sleepycat Berkeley Database 4.1
- Thread support (per *slurpd*)
- OpenSSL 0.9.6 (per TLS/SSL)
- Cyrus-SASL (per SASL)

L'installazione procede come al solito:

```
./configure [opzioni]
make depend
make
make install
```

configurazione OpenLDAP

La configurazione è governata dal file *slapd.conf* (solitamente in */usr/local/etc/openldap/*) e dai file che esso include

Il file è diviso in una sezione generale e una o più sezioni per i *backend*

La definizione dello schema è solitamente in file esterni (nella sottodirectory *schema/*) e viene inclusa nel file principale

slapd.conf

```
idletimeout <secondi>
sizelimit {<massimo>|unlimited}
timelimit {<massimo>|unlimited}
password-hash [{SSHA} {SHA} {SMD5} {MD5}
  {CRYPT} {CLEARTEXT}]
referral <url>

include /etc/openldap/schema/core.schema
include
  /etc/openldap/schema/inetorgperson.schema
include
  /etc/openldap/schema/eduperson.schema
schemacheck [off|on]
```

impostazioni di sicurezza in slapd.conf

```
allow [bind_v2 bind_anon_credallows
  bind_anon_dn update_anon]
disallow [bind_anon bind_simple bind_krbv4
  tls_2_anon]
require bind  LDAPv3 authc SASL strong none
TLSCACertificateFile <filename> oppure
  TLSCACertificatePath <path>
TLSCertificateFile <filename>
TLSCertificateKeyFile <filename>
```

backend

OpenLDAP supporta diversi *backend* da usare come origine dei dati

- database Sleepycat BerkeleyDB (varie versioni)
- server DNS (solo referral)
- server LDAP singoli e multipli (funzione proxy)
- file passwd
- script di shell, Tlc e Perl
- RDBMS (via ODBC)

bdb *backend*

Per un'istallazione stand-alone, occorre un database. Il default è il backend bdb (Berkeley DB 4.1)

```
database        bdb
directory       /usr/local/var/sldap
suffix          "dc=unige,dc=it"
rootdn          "cn=Manager,dc=unige,dc=it"
rootpw          secret
index   objectClass     eq
```

agenda

- ✔ applicazioni
- ✔ protocollo, struttura dati e operazioni
- ✔ autenticazione e autorizzazione
- ✔ strumenti e interscambio dati
- ✔ progettazione di un servizio
- ✔ OpenLDAP
- ■ esempio

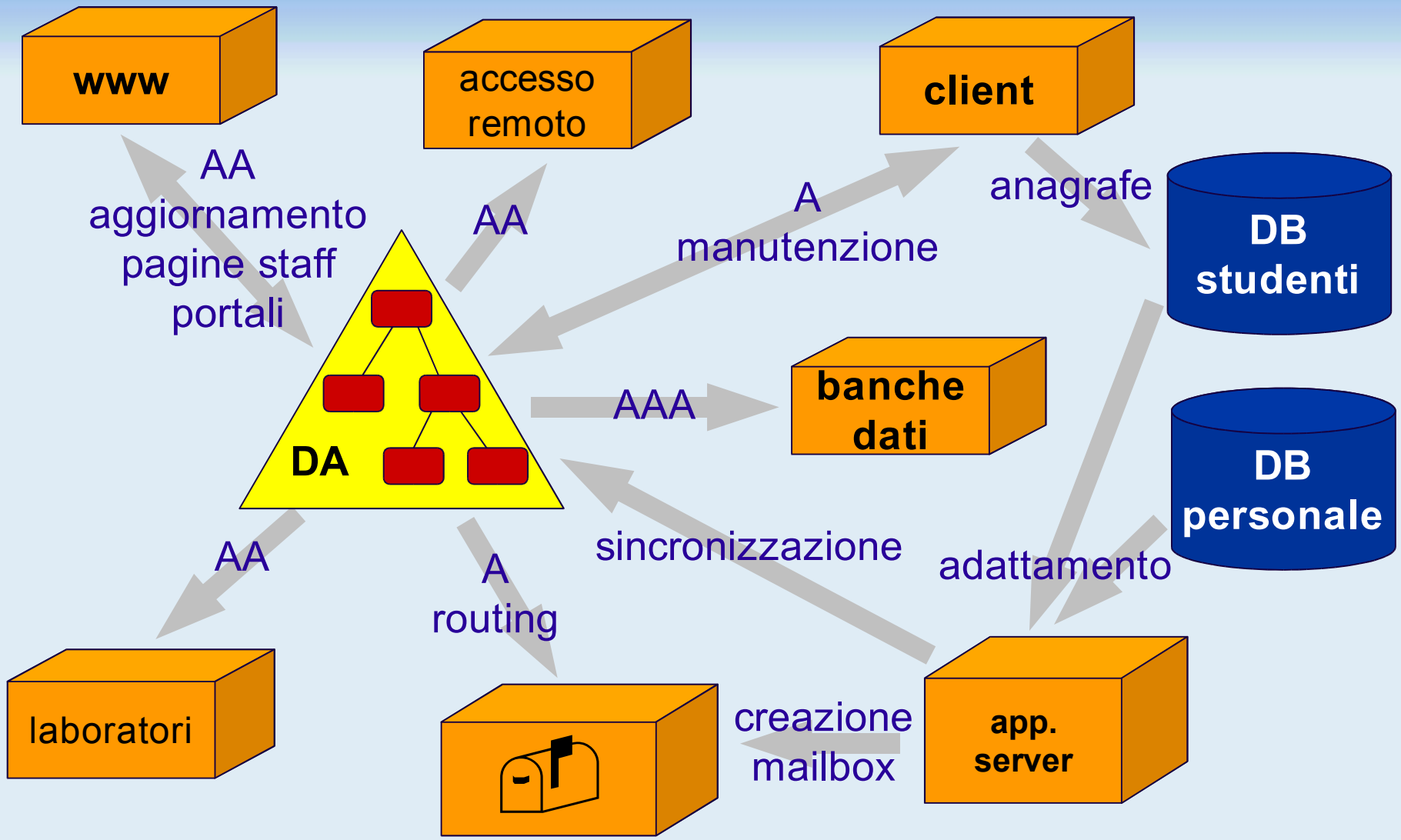
esempio: sistema di posta UniGe

Il sistema è composto da cinque servizi principali:

- PMDF per il personale (~ 3500 utenti)
- PostFix+Cyrus Imap per gli studenti (~ 9500 utenti + ~ 2500 *forward*)
- Postfix+SpamAssassin come *smarthost*
- Sympa per le mailing list
- Imp per le funzioni webmail

Tutti i dati per l'accesso alle mailbox e per l'instradamento dei messaggi sono memorizzati sulla Directory di Ateneo

sistema LDAP UniGe



accesso alle mailbox studenti

Configurazione di PAM (*/etc/pam_ldap.conf*)

```
host ldap-master.csita.unige.it ds2.csita.unige.it
port 389
ldap_version 3
binddn cn=authuser,dc=unige,dc=it
bindpw ********
base dc=unige,dc=it
scope sub
pam_filter mailhost=*  # composto con (uid=%)
pam_login_attribute uid
```

Cyrus usa PAM per autenticare gli accessi

```
auth        required      /lib/security/pam_ldap.so
account     required      /lib/security/pam_ldap.so
```

schema per il routing dei messaggi

Ispirato a Netscape Messaging Server

```
objectClass mailRecipient
  allows
    cn,
    mail,
    mailAlternateAddress,
    mailHost,
    mailRoutingAddress,
    uid,
    userPassword,
    ...
```

attributi per il routing

Il sistema usa tre attributi:

`mailAlternateAddress`

- Tutti gli indirizzi a cui risponde la mailbox

`mailHost`

- server che ospita la mailbox

`mailRoutingAddress`

- indirizzo nella forma <userid>@<nome host>

esempio mailRecipient

Ogni casella di posta è associata a una entry
Per le persone:

```
dn: cn=marco,ou=CSITA,dc=unige,dc=it
objectClass: person
objectClass: mailRecipient
uid: C9999
mailAlternateAddress: marco@csita.unige.it
mailAlternateAddress:
marco.ferrante@unige.it
mailHost: mbox.unige.it
mailRoutingAddress: C9999@mbox.unige.it
```

configurazione PostFix

```
virtual_maps = ldap:studenti
studenti_server_host = ds1.csita.unige.it
  mbox.unige.it ds2.csita.unige.it
studenti_server_port = 389
studenti_search_base = dc=unige,dc=it
studenti_timeout = 60
studenti_query_filter =
  (mailAlternateAddress=%s)
studenti_result_attribute =
  mailRoutingAddress
```

riferimenti

Active Directory

Chadwick, D.W. "**Windows 2000: A Threat to Internet Diversity and Open Standards?**", in IEEE Computer, August 2000, http://sec.isi.salford.ac.uk/download/ComputerStandards.pdf

Progettazione

Michael R. Gettes - **A Recipe for Configuring and Operating LDAP Directories**, Maggio 2000, http://www.georgetown.edu/giia/internet2/ldap-recipe/

Gruppi

Barton, Thomas - **Practices in Directory Groups**, Gennaio 2002, http://middleware.internet2.edu/dir/groups/rpr-nmi-edit-mace_dir-groups_best_practices-1.0.html

riferimenti

Mail routing

ldap, postfix and courier-imap howto

http://www.the-djs.com/~joenix/vriesman.tk/postfix-courier-ldap-howto.html

draft vari

draft-lachman-ldap-mail-routing-03.txt

draft-ietf-asid-email-routing-su-00.shtml

Cyrus IMAP

http://asg.web.cmu.edu/cyrus/imapd/