

# LDAP: introduzione teorica

Giacomo Tenaglia

CNR Biblioteca di Area

19 Dicembre 2006

# Cos'è LDAP

- Protocollo per accedere a informazioni condivise in rete
- Accesso client-server a collezione di informazioni: ricerca, lettura, inserimento, modifica e cancellazione dei dati
- LDAP server: implementa LDAP, le informazioni sono immagazzinate al suo interno
- LDAP gateway: implementa LDAP, utilizza altri server per reperire le informazioni
- Directory services illustri: X.500, WHOIS, NIS/YP...

# Cosa non è LDAP

- Non è un sostituto di un database relazionale
- Non è un file system per oggetti di grandi dimensioni
- Non è ottimizzato per immagazzinare informazione particolarmente dinamica
- Non è utile senza applicazioni

# Applicazioni comuni

- White pages via LDAP o interfacce web
- Autenticazione e autorizzazione
- Roaming profiles
- PKI

# Breve storia

- In principio era X.500/DAP (ISO/OSI)..
- 1991: definizione di LDAP (TCP/IP)
- Da gateway verso X.500 a server LDAP standalone..
- 1996: LDAPv3

# Principi di base

Come è organizzata l'informazione?

- struttura gerarchica ad albero
- molto comune una struttura che rispecchia il DNS

Come ci si riferisce all'informazione?

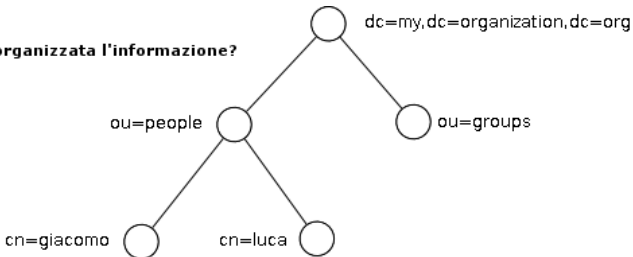
- Distinguished Name (DN):  
entry,antenati,nell,albero,della,directory

Che tipo di informazione può essere immagazzinata?

- entry contenenti attributi (tipo/valore)
- sintassi ben precisa
- l'identificazione delle entry tramite DN è univoca

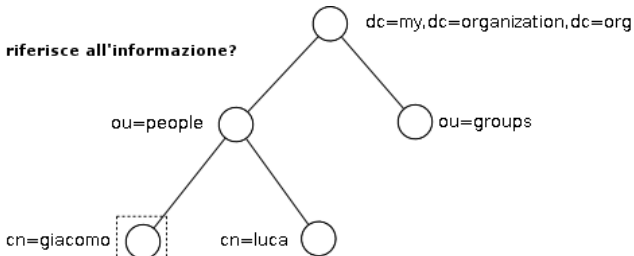
# Principi di base

Come e' organizzata l'informazione?



# Principi di base

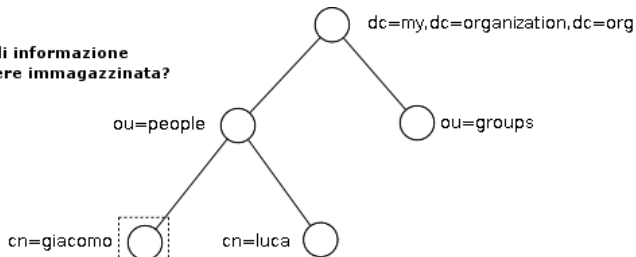
Come ci si riferisce all'informazione?



dn: cn=giacomo,ou=people,dc=my,dc=organization,dc=org

## Principi di base

Che tipo di informazione  
puo` essere immagazzinata?



dn: cn=giacomo,ou=people,dc=my,dc=organization,dc=org

cn: giacomo  
objectClass: person  
objectClass: inetOrgPerson  
objectClass: posixAccount  
mail: giacomo.tenaglia@my.organization.org  
telephoneNumber: +390511234567

# LDAP Data Interchange Format (LDIF)

- Standard per immagazzinare informazioni su configurazione e contenuto di una directory LDAP in file di testo
- Usato per importare/esportare nuovi dati nella directory o per effettuare modifiche sui dati
- File LDIF: collezione di entry, separate da linee vuote
- Mappatura tipo attributo: valore per ogni entry

# LDAP Data Interchange Format (LDIF)

```
dn: cn=giacomo,ou=people,dc=my,dc=organization,dc=org
cn: giacomo
objectClass: person
objectClass: inetOrgPerson
objectClass: posixAccount
mail: giacomo.tenaglia@my.organization.org
telephoneNumber: +390511234567
```

```
dn: cn=luca,ou=people,dc=my,dc=organization,dc=org
cn: luca
sn: ferroni
objectClass: person
objectClass: inetOrgPerson
objectClass: posixAccount
mail: luca.ferroni@my.organization.org
```

# Attributi, objectClass e schemi

## Attributi:

- usati per immagazzinare i dati
- sintassi definita nello schema corrispondente, specifica formato dei valori e possibilità di assumere più valori contemporaneamente

```
attributetype ( 0.9.2342.19200300.100.1.3
NAME ( 'mail' 'rfc822Mailbox' )
DESC 'RFC1274: RFC822 Mailbox'
EQUALITY caseIgnoreIA5Match
SUBSTR caseIgnoreIA5SubstringsMatch
SYNTAX 1.3.6.1.4.1.1466.115.121.1.26{256} )
```

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

# Attributi, objectClass e schemi

objectClass:

- tutte le entry devono avere un attributo objectClass, che deve avere sempre almeno un valore
- ogni valore di objectClass è template per i dati che possono essere immagazzinati nella entry
- sono definiti gli attributi obbligatori e quelli opzionali
- anche questo è specificato negli schemi corrispondenti ad ogni objectClass

# Attributi, objectClass e schemi

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

```
objectClass ( 2.16.840.1.113730.3.2.2
NAME 'inetOrgPerson'
DESC 'RFC2798: Internet Organizational Person'
SUP organizationalPerson
STRUCTURAL
MAY (
audio $ businessCategory $ carLicense $ departmentNumber $
displayName $ employeeNumber $ employeeType $ givenName $
homePhone $ homePostalAddress $ initials $ jpegPhoto $
labeledURI $ mail $ manager $ mobile $ o $ pager $
photo $ roomNumber $ secretary $ uid $ userCertificate $
x500uniqueIdentifier $ preferredLanguage $
userSMIMECertificate $ userPKCS12 )
)
```

# Attributi, objectClass e schemi

## Schemi:

- definiscono sintassi degli attributi e struttura delle objectClass
- ogni definizione è identificata in modo univoco da un Object Identifier (OID)
- gli OID sono assegnati univocamente dalla IANA (vedi indirizzi IP)

# Attributi, objectClass e schemi

```

attributetype ( 0.9.2342.19200300.100.1.3
NAME ( 'mail' 'rfc822Mailbox' )
DESC 'RFC1274: RFC822 Mailbox'
EQUALITY caseIgnoreIA5Match
SUBSTR caseIgnoreIA5SubstringsMatch
SYNTAX 1.3.6.1.4.1.1466.115.121.1.26{256} )

```

```

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

```

```

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

```

```

objectClass ( 2.16.840.1.113730.3.2.2
NAME 'inetOrgPerson'
DESC 'RFC2798: Internet Organizational Person'
SUP organizationalPerson
STRUCTURAL
MAY (
audio $ businessCategory $ carLicense $ departmentNumber $
displayName $ employeeNumber $ employeeType $ givenName $
homePhone $ homePostalAddress $ initials $ jpegPhoto $
labeledURI $ mail $ manager $ mobile $ o $ pager $
photo $ roomNumber $ secretary $ uid $ userCertificate $

```

# Autenticazione e ACL

Perchè è necessario (e utile) autenticarsi?

- il poter compiere operazioni su una directory dipende dal livello di autorizzazione che l'utente autenticato possiede
- simile all'autenticazione su di un database
- è possibile definire anche la granularità del controllo degli accessi, dalla directory intera, al sottoalbero, all'attributo specifico

# Autenticazione e ACL

## Binding:

- l'atto di essere autenticato da una directory LDAP
- username: si specifica una entry tramite il suo DN
- password: deve corrispondere al valore dell'attributo userPassword della entry

## Tipi di autenticazione LDAPv3:

- Anonymous Authentication: password vuota
- Simple Authentication: password trasmessa in chiaro
- Simple Authentication over SSL/TLS: ldaps:// oppure startTLS su ldap://
- Simple Authentication and Security Layer: Kerberos, GSSAPI, SKEY/MD5 e altri plugin

# Autenticazione e ACL

```
access to attrs=userPassword
  by dn="cn=admin,dc=my,dc=organization,dc=org" write
  by anonymous auth
  by self write
  by * none

access to *
  by dn="cn=admin,dc=my,dc=organization,dc=org" write
  by * read
```

# Perchè un directory service?

- Autenticazione centralizzata
- Mappa di personale, gruppi, macchine e servizi
- Ottimizzato per operazioni di lettura
- Replicazione facilitata (scalabilità e load balancing)
- Struttura basata su schemi facilmente estendibile
- Schemi facilmente modificabili
- Integrazione con un numero di applicativi sempre crescente

# Struttura della directory

## Nomenclatura

- base DN: dc=dominio,dc=ente,dc=it
- una entry ou=nomeistituto per ogni istituto
- entry standard per utenti (ou=People), gruppi (ou=Group), macchine (ou=Hosts), ... all'interno di ogni istituto

## Che informazione immagazzinare nella directory?

- inizialmente migrazione del sistema di autenticazione esistente
- informazioni aggiuntive riguardo gli utenti (telefono, email..)
- si potrebbero mappare anche macchine, servizi per gestire anche l'autorizzazione in maniera centralizzata

# Struttura della directory

Schemi: inetOrgPerson, posixAccount ...eduPerson

...

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

# LDAP è la Risposta?

Contro:

- Impatto iniziale non dei più amichevoli
- Bisogna “entrare nella logica” del directory service

Pro:

- Standard sempre più diffuso
- Riduce il mal di testa dell'amministratore di sistema
- Riduce il mal di testa dell'utente

## Riferimenti

- RFC 2251-2256: Lightweight Directory Access Protocol (v3) (1996)
- RFC 2829: Authentication Methods for LDAP (2000)
- G.Carter, “LDAP System Administration”, O’Reilly (2003)
- C.Donley, “LDAP Programming, Management and Integration”, Manning (2003)
- <http://www.openldap.org>