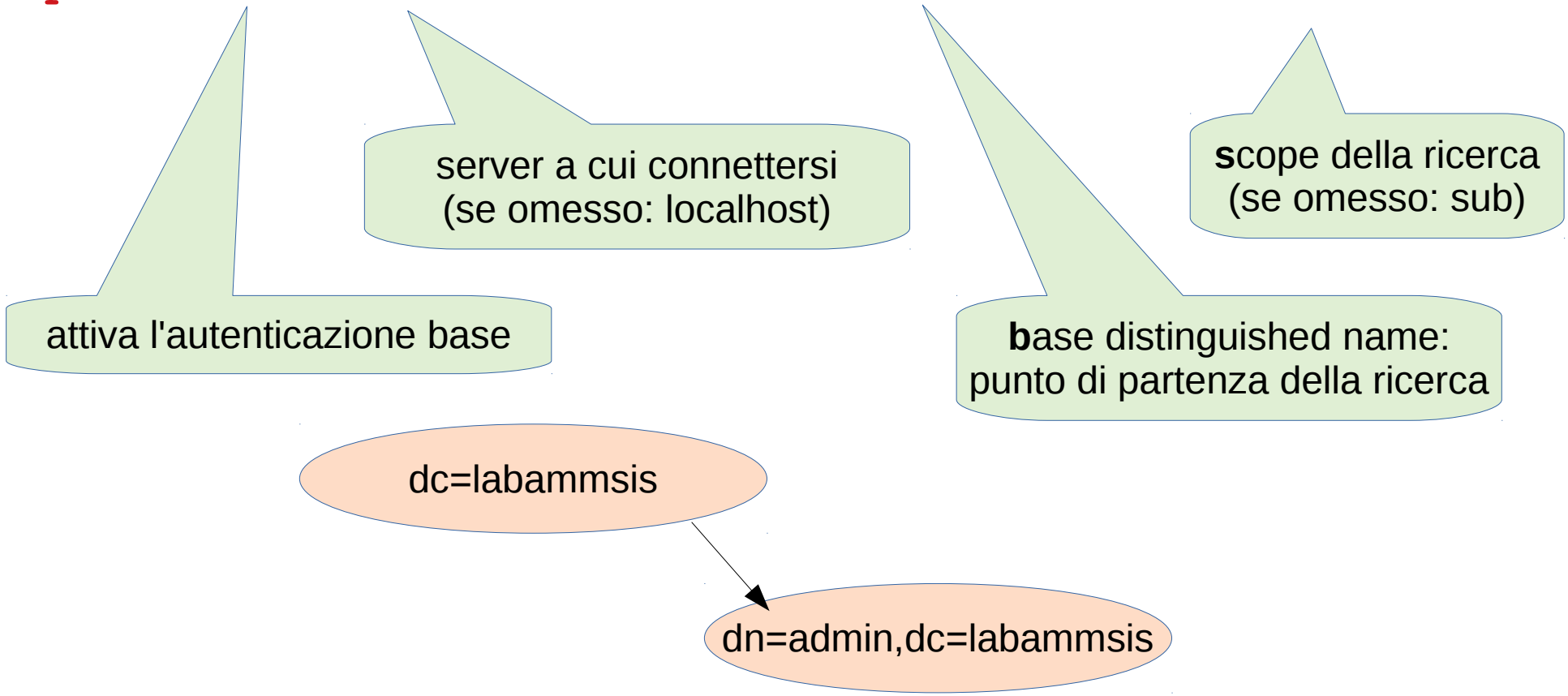
LDAP client CLI - search

- Esplorare per intero la directory del laboratorio
  - ripetere con *one* e *base* al posto di *sub*

```
ldapsearch -x -h 192.168.56.203 -b "dc=labammsis" -s sub
```



LDAP client CLI - search

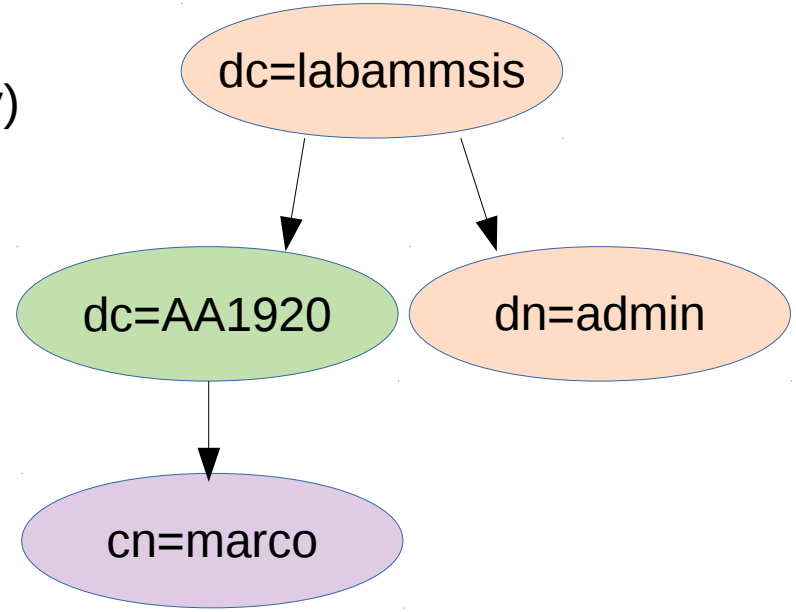
- Query autenticate
  - esempio precedente: bind come guest
  - mostra una interessante entry
  - è l'utente amministratore della directory, con password=admin
  - provare:

```
ldapsearch -x -D "cn=admin,dc=labammsis" -w "admin"
  -h 192.168.56.203 -b "dc=labammsis" -s sub
```

  - notate che anche in lettura cambia qualcosa
  - parametri indispensabili per le scritture

LDAP client CLI - add

- Aggiunta di entry
  - (nella figura è mostrato solo il RDN delle entry)



new.ldif:

```
dn: dc=AA1920,dc=labammsis
objectClass: dcObject
objectClass: organization
dc: AA1920
o: labammsis
(riga vuota)
dn: cn=marco,dc=AA1920,dc=labammsis
objectClass: ???
cn: marco
sn: prandini
```

perché devo inserirli?
https://www.ietf.org/rfc/rfc2247.txt
https://www.ietf.org/rfc/rfc2256.txt

posso trovare una classe adatta?
https://oav.net/mirrors/LDAP-ObjectClasses.html

```
ldapadd -x -D "cn=admin,dc=labammsis" -w "admin"
  -h 192.168.56.203 -f "new.ldif"
```

se omesso:
ldif letto da STDIN

LDAP client CLI - search

- Le ricerche si possono fare per contenuto delle entry, oltre che per posizione nel DIT

```
ldapsearch -x -h 192.168.56.203 -b "dc=labammsis" -s sub
'cn=marco'

ldapsearch -x -h 192.168.56.203 -b "dc=labammsis" -s sub
'(!(o=*))'

ldapsearch -x -h 192.168.56.203 -b "dc=labammsis" -s sub
'(&(cn=marco)(sn=prandini))'

ldapsearch -x -h 192.168.56.203 -b "dc=labammsis" -s sub
'(|(&(cn=marco)(sn=prandini))(o=*))'
```

LDAP client CLI - delete

- uno per uno

```
ldapdelete -x -D "cn=admin,dc=labammsis" -w "admin"
  -h 192.168.56.203 "cn=marco,dc=AA1920,dc=labammsis"
```

- o anche lista via STDIN

```
dn_list:

cn=marco,dc=AA1920,dc=labammsis

dc=AA1920,dc=labammsis

cat dn_list | ldapdelete -x -D "cn=admin,dc=labammsis"
 -w "admin" -h 192.168.56.203
```

LDAP client CLI - modify

- LDIF con attributo changetype: modify seguito da add o replace o delete che specificano su quale attributo agire
- Molte modifiche possibili
  - aggiungere un valore di un attributo
  - sostituire i valori di un attributo
  - rimuovere completamente un attributo
  - rimuovere un valore di attributo
- Vedere esempi nel prontuario sul sito

Configurazione LDAP server

- La configurazione del server è conservata in un DIT separato dai dati
  - completamente consultabile via LDAP stesso (da localhost)

    `ldapsearch -Y EXTERNAL -H ldapi:/// -b "cn=config"`
    - sui nostri sistemi, il meccanismo di autenticazione "EXTERNAL" essenzialmente significa fidarsi dell'utente Unix → *root* è autorizzato a configurare il servizio
  - modificabile via LDAP stesso con **`ldapadd`**, **`ldapmodify`**, **`ldapdelete`**
    - sempre con l'autenticazione "speciale"
- Un'operazione che può essere necessaria è la definizione di nuove classi di oggetti e nuovi tipi di attributo, se quelli standard non sono adatti alla modellazione del nostro DIT
- Vedere guida a LDAP sul sito del corso

Esercizio

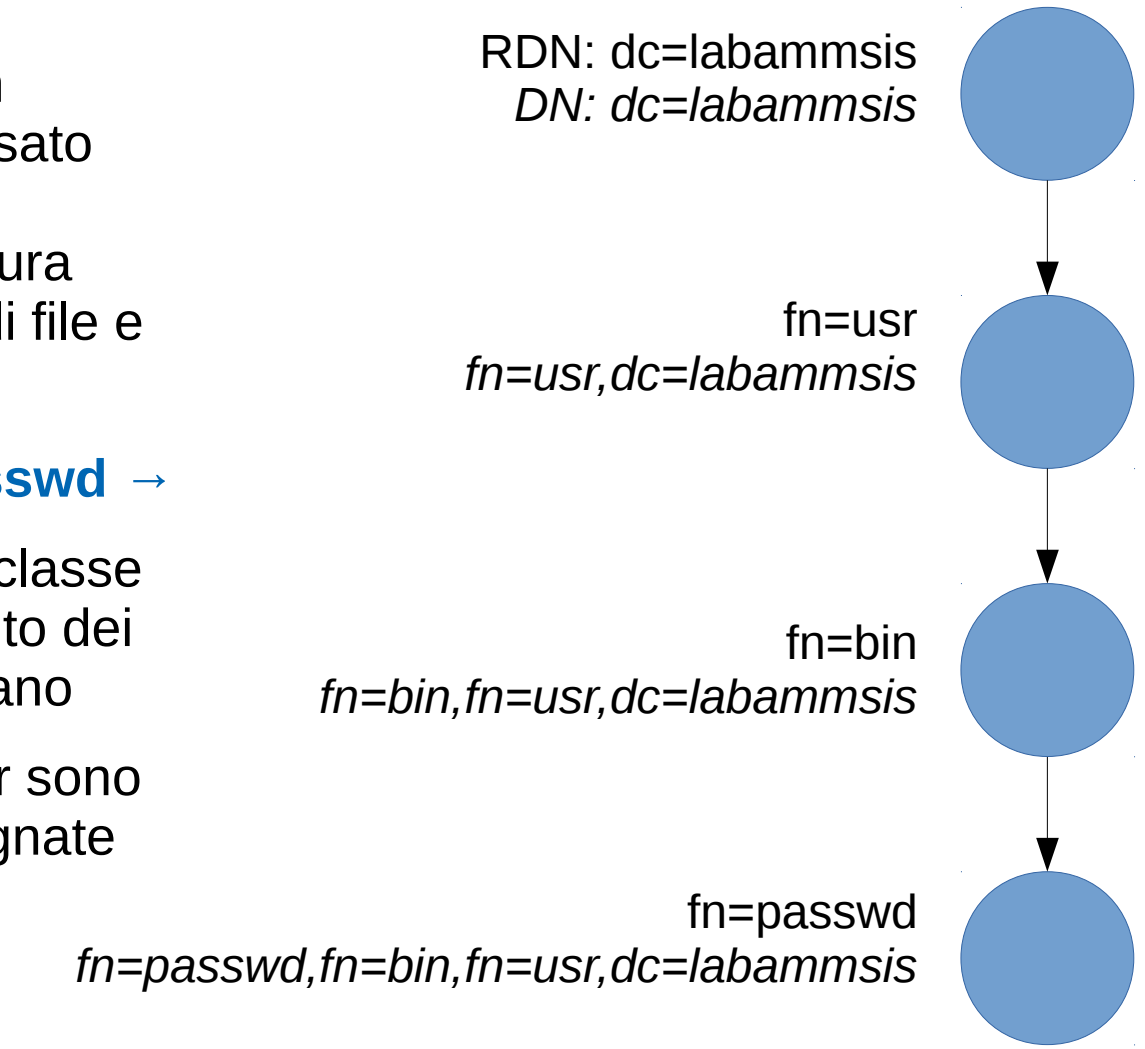
- Definire
  - due nuovi tipi di attributo
    - **fn** di tipo adatto a rappresentare un nome di file
    - **fs** adatto a rappresentare una dimensione in byte
  - due nuove classi *ausiliarie*
    - **dir** che contenga obbligatoriamente **fn** e facoltativamente **fs**
    - **file** che contenga obbligatoriamente sia **fn** che **fs**
    - nota: definiamo queste classi come ausiliarie solo a fini didattici, per ricordarci poi che le classi ausiliarie non possono essere usate da sole, non c'è un motivo concettuale

Esercizio - inserire entry

- **ldap-fs-store.sh** -
  memorizzare nella directory un sottoalbero del filesystem, passato come parametro allo script, riproducendo con i DN la struttura gerarchica della collocazione di file e directory.

  es. /usr/bin/passwd →

  - scegliere per ogni nodo la classe più appropriata e tener conto dei vincoli MUST che ne derivano
  - si noti che le classi file e dir sono ausiliarie: vanno accompagnate da una classe strutturale
  - scegliamo **organization**



Esercizio - Modificare entry

- **ldap-fs-sumspace.sh** - esplorando la directory LDAP, calcolare per ogni entry che rappresenta una directory lo spazio occupato dai file presenti in tale directory, ed aggiornare l'entry con la somma

- es: in LDAP ho due entry con `objectClass=file`
  - `fn=pippo,fn=lib,fn=usr,dc=labammsis` supponiamo con `fs=10`
  - `fn=pluto,fn=lib,fn=usr,dc=labammsis` supponiamo con `fs=20`

  → aggiorno l'entry `fn=lib,fn=usr,dc=labammsis`

  (che ovviamente avrà `objectClass=dir`)

  impostando `fs=30`

Eliminare entry / estensioni

- **ldap-fs-purge.sh** - esplorare la directory LDAP, e verificare se i file in essa rappresentati esistono ancora sul filesystem. In caso contrario rimuoverli da LDAP.
- **Estensioni proposte**
  - introdurre nello schema attributi adatti a rappresentare ownership e permessi dei file
    - la directory va svuotata (come?), riconfigurata, e ripopolata
  - modificare ldap-fs-purge per
    - controllare se i permessi dei file esistenti sono uguali a quelli memorizzati nella directory LDAP
    - ripristinare i permessi sul filesystem prendendoli da LDAP nel caso siano diversi, segnalando l'evento nei log di sistema