

# DNS

**Główne zadanie:** zamiana nazw domenowych na adr IP

Zawiera także inne informacje o nazwach domenowych...

Usługa DNS utrzymuje tablice rekordów (**RR**) tej postaci:

(nazwa\_domenowa, typ, wartość, ?), gdzie typ to ...

A (adr ip), MX (MTA), NS (autorytatywny ser dns domeny),

CNAME (nazwa kanon), i inne ... patrz: **rfc 1035**

Działa nad prot UDP lub TCP (większe ser dns...)

Zasada działania:

- mój serwer DNS ma informacje o wielu nazw dom (w cache!!)

- jeśli nie ma informacji np. o „lts.wmi.amu.edu.pl” to wtedy:

pyta korzeń o adres ser dns obsługującego domene „pl”

ten serwer powinien wiedzieć wszystko o nazw „\*.pl” (subdomeny)

czyli powinien znać adr ser dns obsługującego domenę „edu.pl”

ten powinien znać adres ser dns obsługującego „amu.edu.pl”

...itd... aż dojdziemy do tzw *serwera autorytatywnego*,

znającego nazwę „lts.wmi.amu.edu.pl”

„Autorytatywny ser dns” dla danej nazwy to ten, w którym zarejestrowano (a nie zcacheowano) nazwę hosta lub domeny

## Hierarchia DNS...

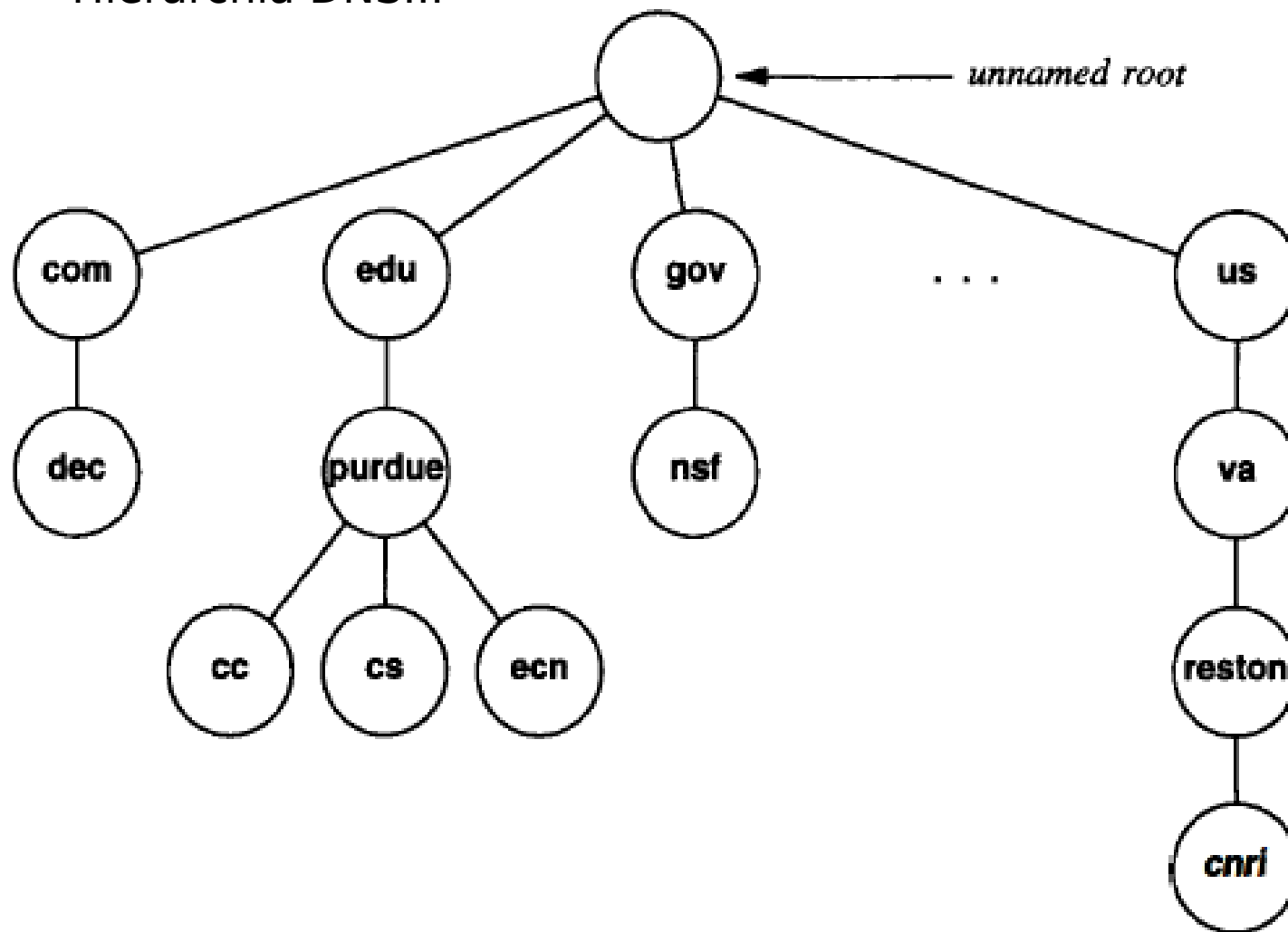
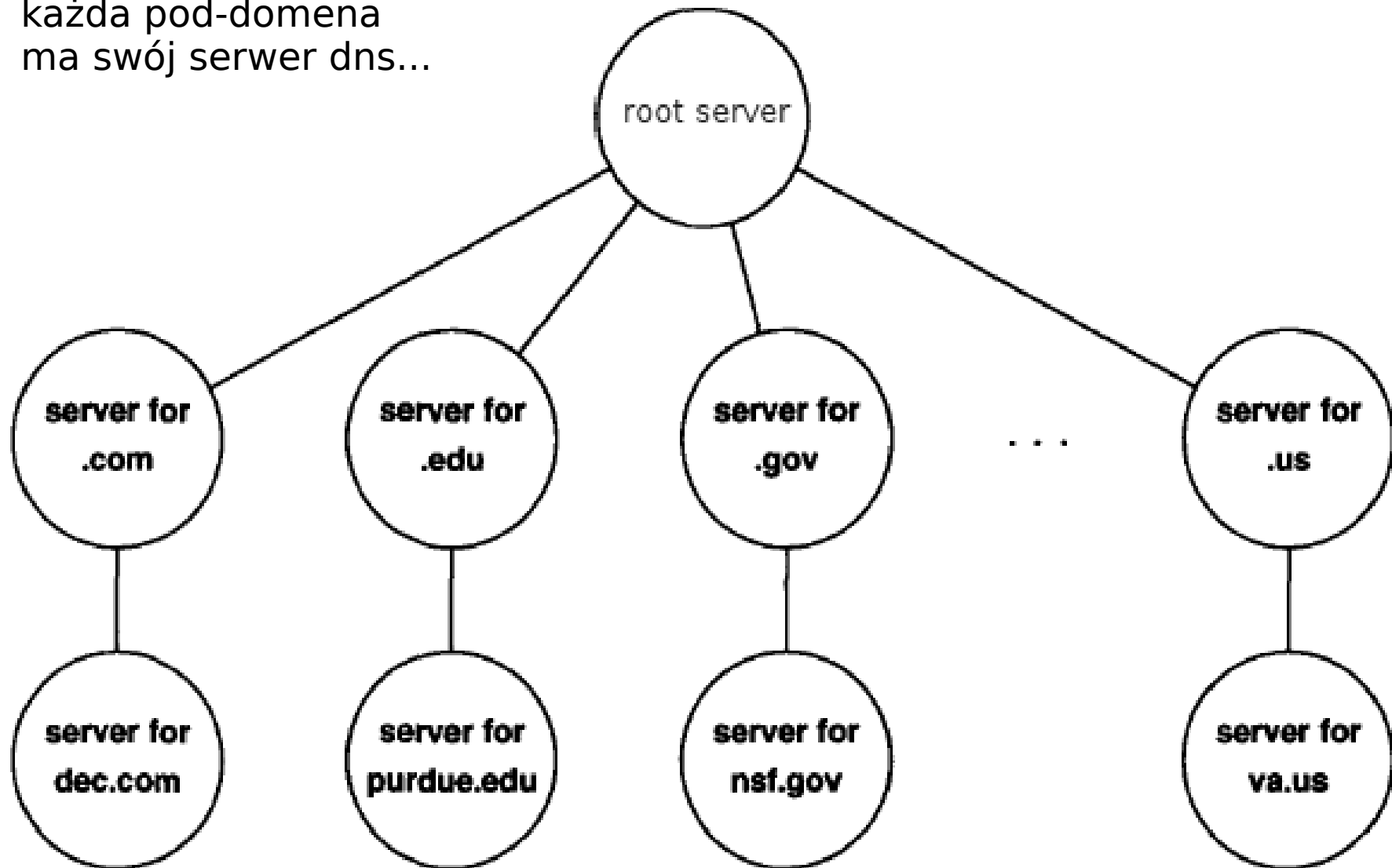


Figure 24.2 A small part of the Internet domain name hierarchy (tree). In practice, the tree is broad and flat; most host entries appear by the fifth level.

Podjęcie wyidealizowane:  
każda pod-domena  
ma swój serwer dns...



Podejście realistyczne:  
bardziej „spłaszczone” ...

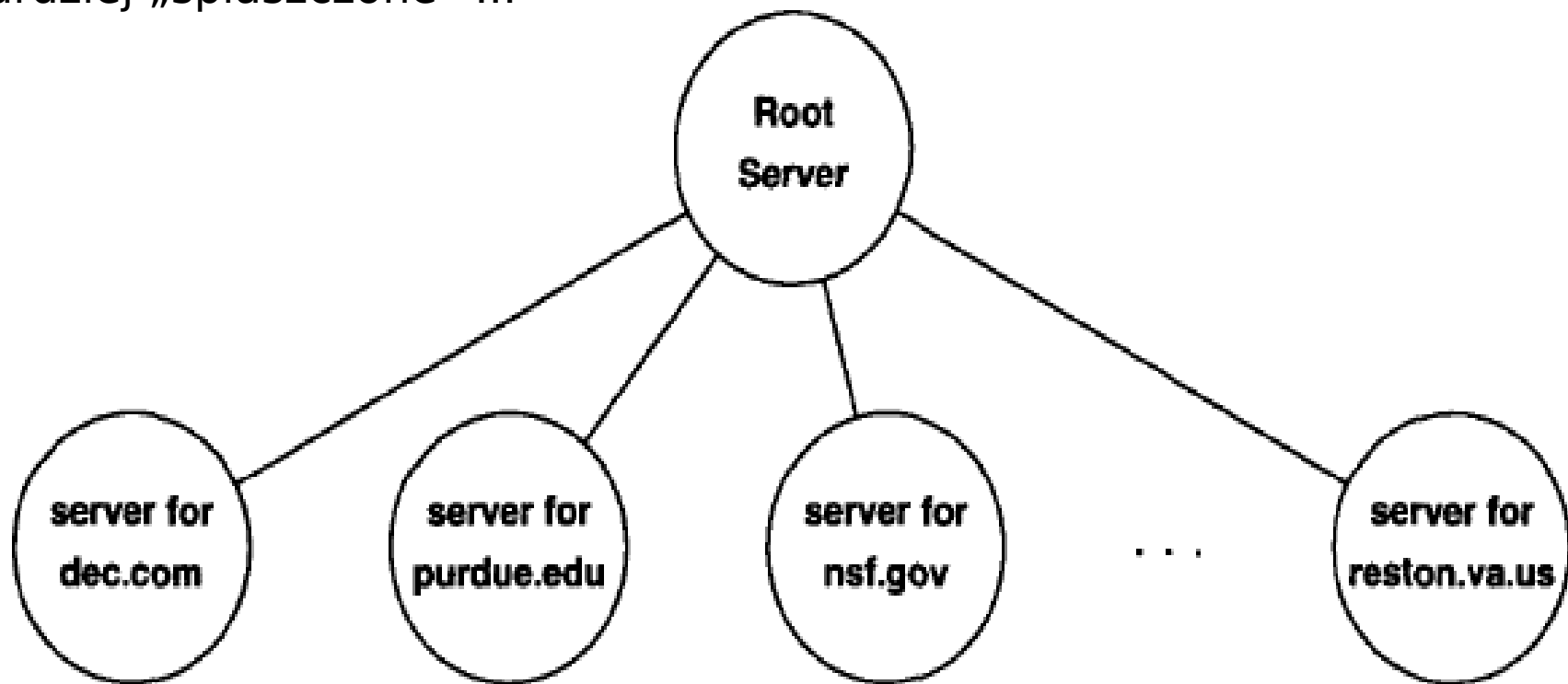
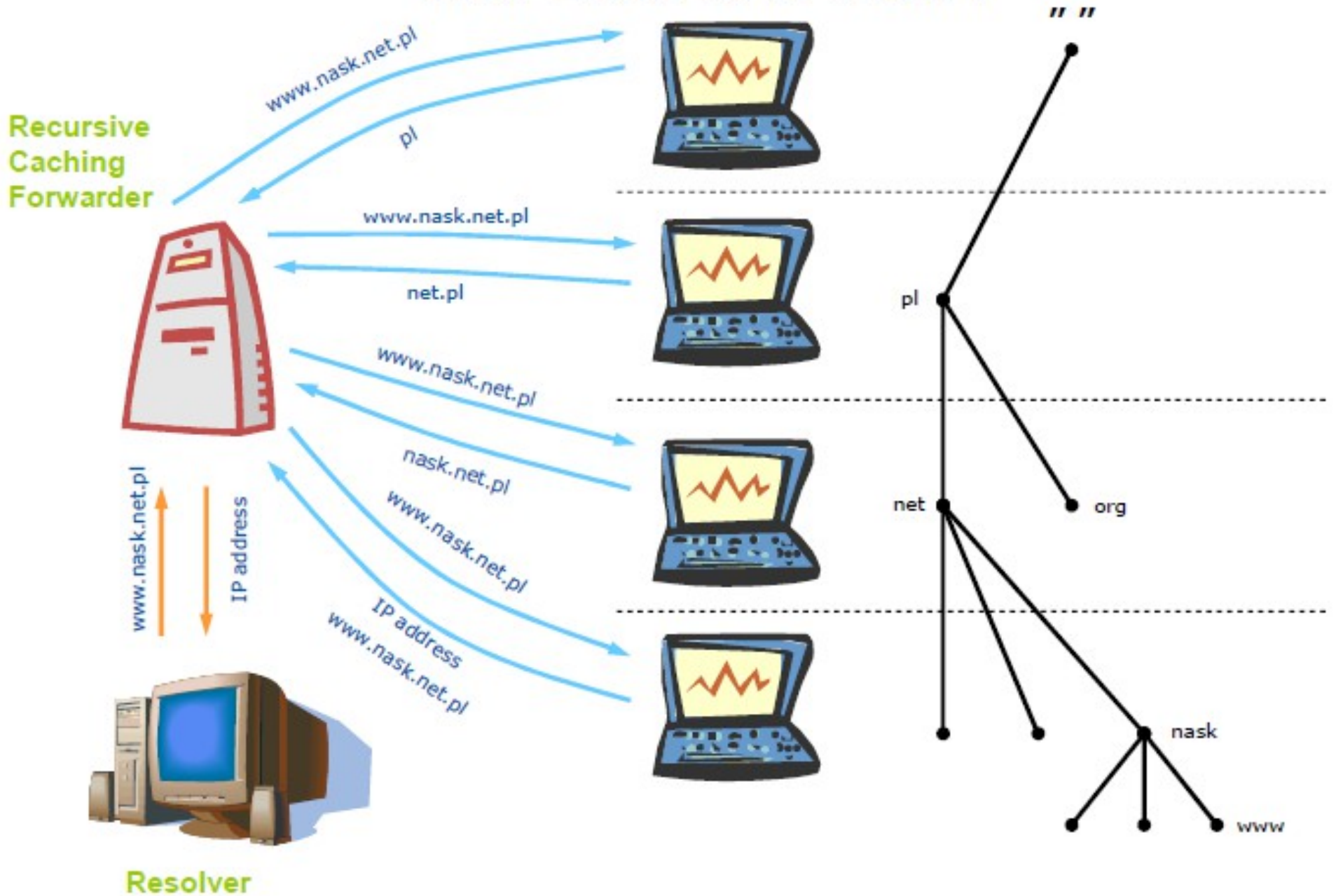
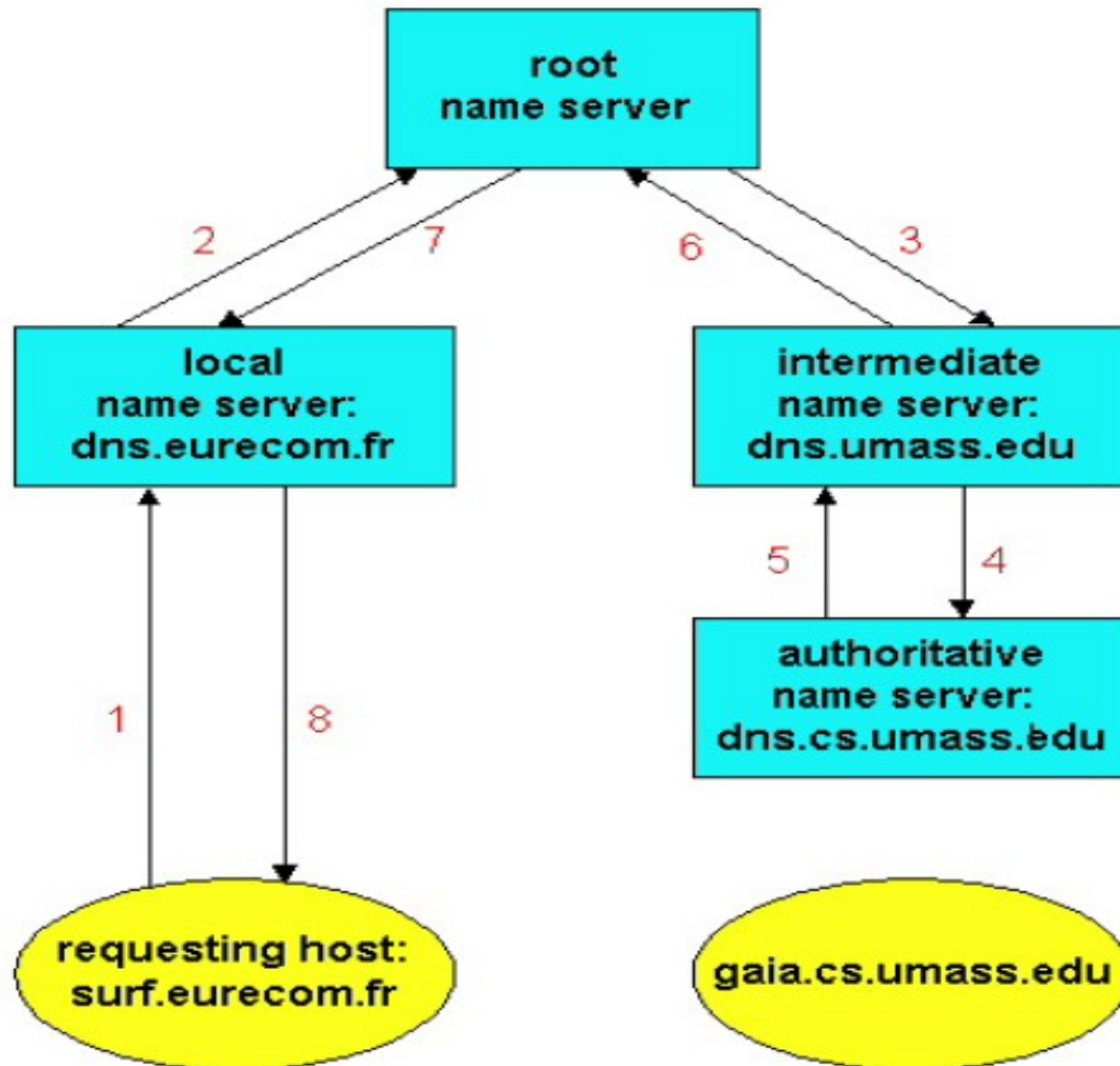


Figure 24.4 A realistic organization of servers for the naming hierarchy of Figure 24.2. Because the tree is broad and flat, few servers need to be contacted when resolving a name.

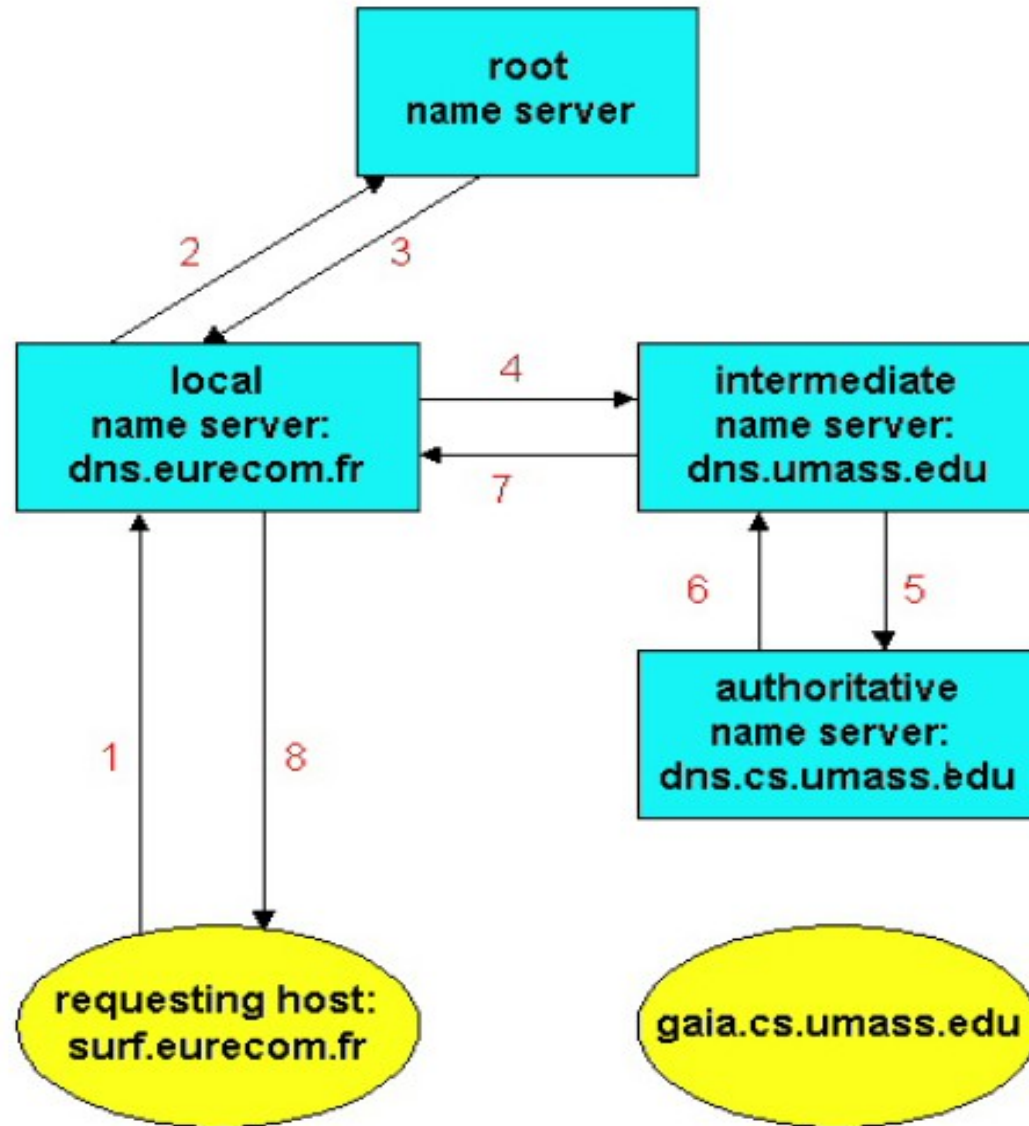
# How Does DNS work ?



Szukamy gaia.cs.umass.edu; tryb pracy: rekurencyjny



Tryb pracy: iteracyjny oraz rekurencyjny



```
## pakiet dns, j. Tcl
```

```
#
```

```
package re udp; # !!
```

```
  #% 1.0.11
```

```
package re dns
```

```
  #% 1.3.3
```

```
set n1 [dns::resolve "wp.pl"]
```

```
dns::wait $n1
```

```
dns::address $n1
```

```
  #% 212.77.98.9
```

```
join [dns::result $n1] \n
```

```
  #% name wp.pl type A class IN ttl 231 rdlength 4 rdata 212.77.98.9
```

```
set n1 [dns::resolve "amu.edu.pl" -type NS]; # serwery DNS dla tej domeny
```

```
dns::wait $n1
```

```
join [dns::result $n1] \n
```

```
  #% name amu.edu.pl type NS class IN ttl 3192 rdlength 7 rdata dns2.amu.edu.pl
```

```
name amu.edu.pl type NS class IN ttl 3192 rdlength 6 rdata dns.amu.edu.pl
```

```
name amu.edu.pl type NS class IN ttl 3192 rdlength 7 rdata dns3.amu.edu.pl
```

```
set n1 [dns::resolve "amu.edu.pl" -type MX]; # mail/MTA dla tej domeny
```

```
dns::wait $n1
```

```
join [dns::result $n1] \n
```

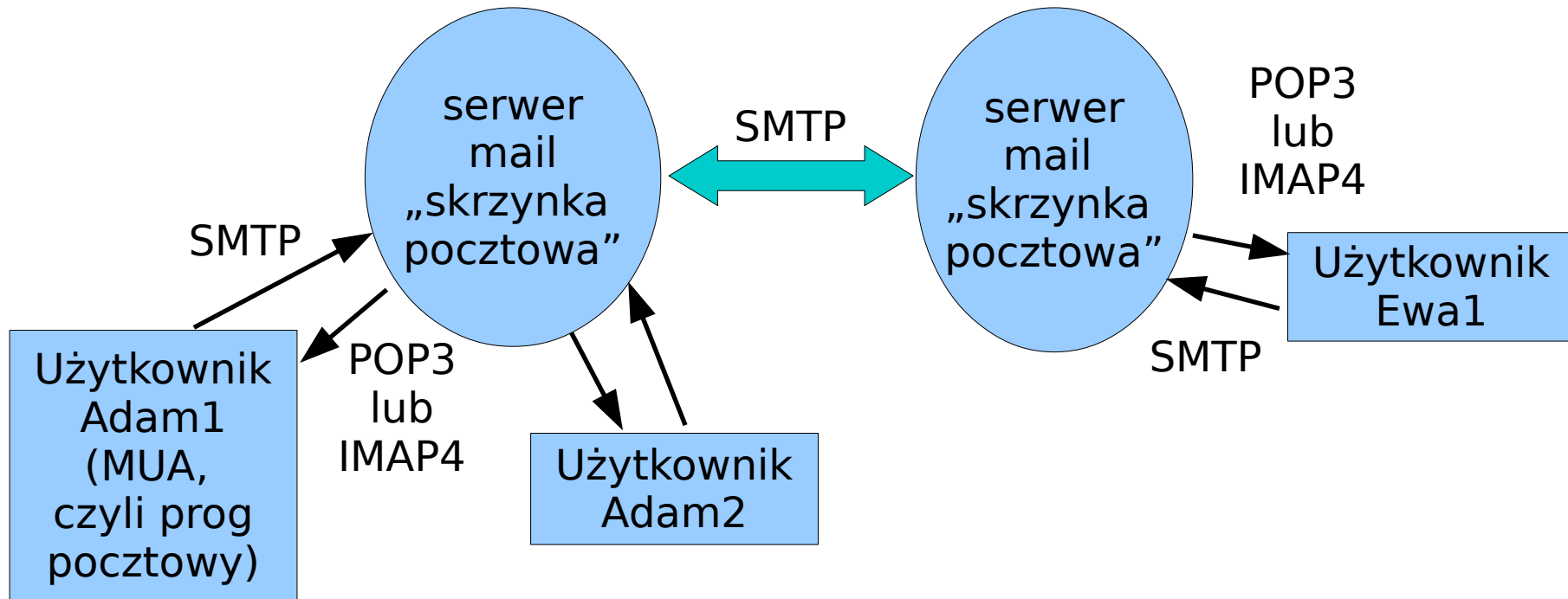
```
  #% name amu.edu.pl type MX class IN ttl 5 rdlength 8 rdata {10 mx2.amu.edu.pl}
```

```
name amu.edu.pl type MX class IN ttl 5 rdlength 8 rdata {10 mx1.amu.edu.pl}
```



# Mail (email)

(mocno uproszczony opis...)



Adres mailowy: `user@serwer`, `user@nazwa_domenowa`  
DNS wyciąga z nazwy dom adr ser mailowego (MX)

Budowa maila:

nagłówki (from:, to:, date:, subject:, content-type:),  
pusta linia,  
treść (w mailach złożonych: części mime; to jest drzewko!!)

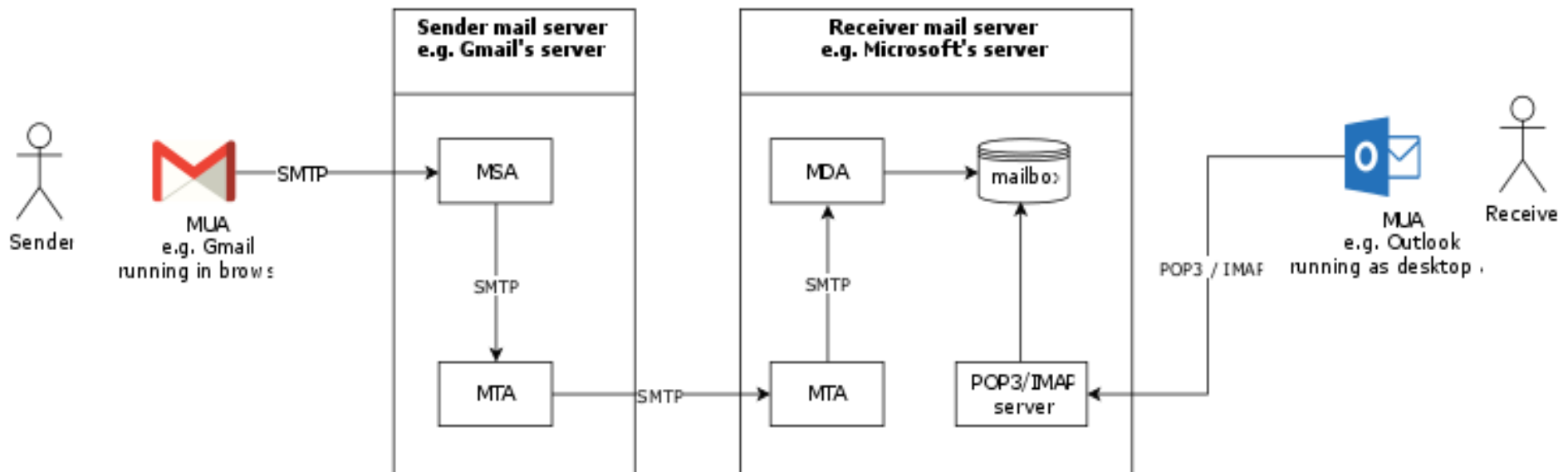
# Serwery mail - więcej szczegółów...

dlaczego MX ma inny adr ip niż ser SMTP ???

MTA = Mail Transfer Agent

MSA = Mail Submission Agent

MDA= Mail Delivery Agent



```
## pakiet pop3, j. Tcl
#
package re tls
package re pop3
source pop3_tls.tcl; # dodaje obsługę tls...

set password ???
set p [pop3::open -ssl 1 pop3.amu.edu.pl mhanckow $password 995]

set maile [split [pop3::list $p] \n]; llength $maile; # mam tyle maili...
#% 694
lrange $maile end-5 end
#% {689 7011} {690 11797} {691 4464} {692 30434} {693 44913} {}

set raw0 [pop3::top $p 693 10]
# + naglowki maila + 10 linijek tresci maila...

set raw1 [pop3::retrieve $p 693]; string len $raw1
#% 44286
# + jednak sciagamy caly mail...
```

```
## pakiet mime, j. Tcl
```

```
#
```

```
package re mime
```

```
set m1 [mime::initialize -string $raw1]
```

```
mime::getheader $m1 from
```

```
  #% {Alicja Adamczyk <aliada2@st.amu.edu.pl> }
```

```
mime::getheader $m1 date
```

```
  #% {Wed, 3 Jun 2020 10:56:32 +0000}
```

```
mime::getproperty $m1 content
```

```
  #% multipart/mixed
```

```
set m2 [mime::getproperty $m1 parts]
```

```
  #% ::mime::1-1 ::mime::1-2
```

```
mime::getproperty [lindex $m2 0] content
```

```
  #% multipart/alternative
```

```
mime::getproperty [lindex $m2 1] content
```

```
  #% text/plain
```

```
set m3 [lindex $m2 0]
```

```
  #% ::mime::1-1
```

```
mime::getproperty $m3 content
```

```
  #% multipart/alternative
```

```
set m4 [mime::getproperty $m3 parts]
```

```
  #% ::mime::1-1-1 ::mime::1-1-2
```

```
mime::getproperty [lindex $m4 0] content
```

```
  #% text/plain
```

```
mime::getproperty [lindex $m4 1] content
```

```
  #% text/html
```

# Usługa katalogowa, X.500, LDAP

usługa katalogowa czyli opis zasobów w sieci...

zasoby: użytkownicy, grupy użytkowników, komputery,  
zakłady, departamenty, drukarki, ...

drzewo obiektów/folderów

ścieżka od korzenia do wierzchołka (obiektu lub folderu)

ścieżka jest postaci "attr1=value1, attr2=value2, ..."

obiekty także mają atrybuty (np. cn, mail, description, displayName, ...)

można zadawać "zapytania o obiekty" z warunkiem logicznym

**LDAP** = Lightweight Directory Access Protocol, dostęp do usł. kat.

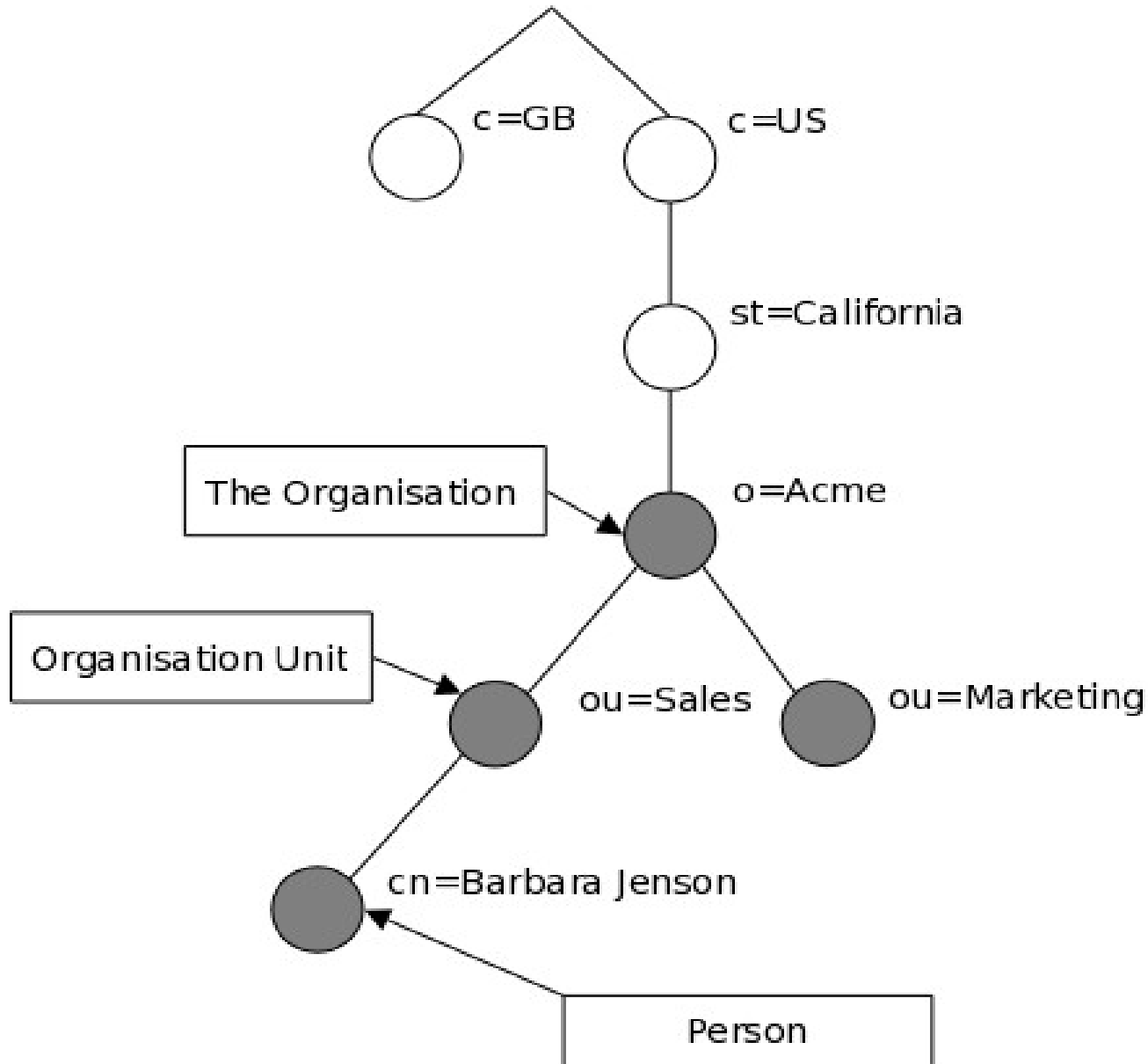
docs: X.500, rfc2251, **rfc4511 (LDAP)**,

<http://www.openldap.org/doc/admin24/intro.html>

# Drzewo LDAP/X.500...

ścieżka do „Barbara Jenson”:

„cn=Barbara Jenson, ou=Sales, o=Acme, st=California, c=US”



```
package re ldap
catch {package re dict}
```

```
# tunel SSH jest niezbędny jeśli działamy spoza WMI !!!
#ssh -L 5000:ldap.wmi.amu.edu.pl:389 ???@lts.wmi.amu.edu.pl
set h [ldap::connect localhost 5000]
```

```
set x1 [ldap::search $h "DC=labs,DC=wmi,DC=amu,DC=edu,DC=pl" "(CN=mhanckow)" ""]
llength $x1
#% 1
lindex $x1 0 0
#% CN=mhanckow,OU=Faculty,OU=People,DC=labs,DC=wmi,DC=amu,DC=edu,DC=pl
# ^ distinguished name
lsort [dict keys [lindex $x1 0 1]]
#% accountExpires badPasswordTime badPwdCount c cn co codePage company countryCode ...
# ^ atrybuty obiektu
```

```
dict get [lindex $x1 0 1] description
#% Pracownik
```

```
dict get [lindex $x1 0 1] distinguishedName
```

```
#% CN=mhanckow,OU=Faculty,OU=People,DC=labs,DC=wmi,DC=amu,DC=edu,DC=pl
```

```
dict get [lindex $x1 0 1] mail
```

```
#% mhanckow@amu.edu.pl
```

```
encoding convertfrom utf-8 [dict get [lindex $x1 0 1] displayName]
```

```
#% {Michał Hańckowiak}
```

```
join [dict get [lindex $x1 0 1] memberOf] \n
```

```
#% CN=Zaklad_Teorii_Algorytmow_i_Bezpieczenstwa_Danych,OU=Departments,
OU=Groups,DC=labs,DC=wmi,DC=amu,DC=edu,DC=pl
```

```
CN=public,OU=Groups,DC=labs,DC=wmi,DC=amu,DC=edu,DC=pl
```

```
CN=faculty_research,OU=Groups,DC=labs,DC=wmi,DC=amu,DC=edu,DC=pl
```

```
CN=faculty,OU=Groups,DC=labs,DC=wmi,DC=amu,DC=edu,DC=pl
```