

10

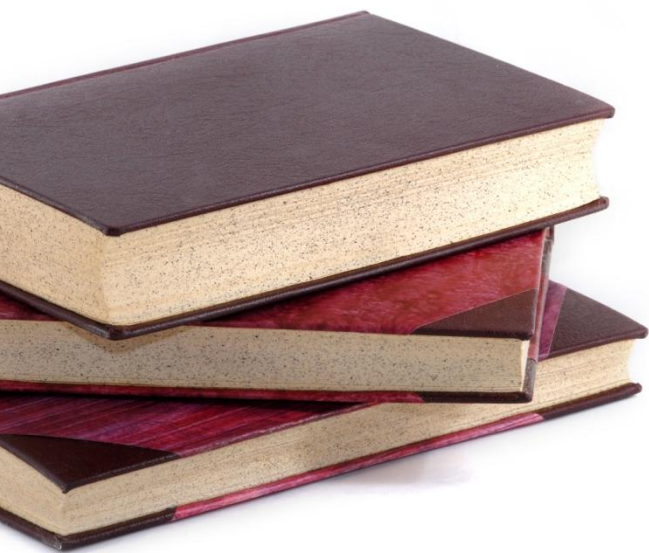
Servicii de rețea

6-7 decembrie 2011

- World Wide Web
- Protocolul HTTP
- Certificate și HTTPS
- Domain Name System

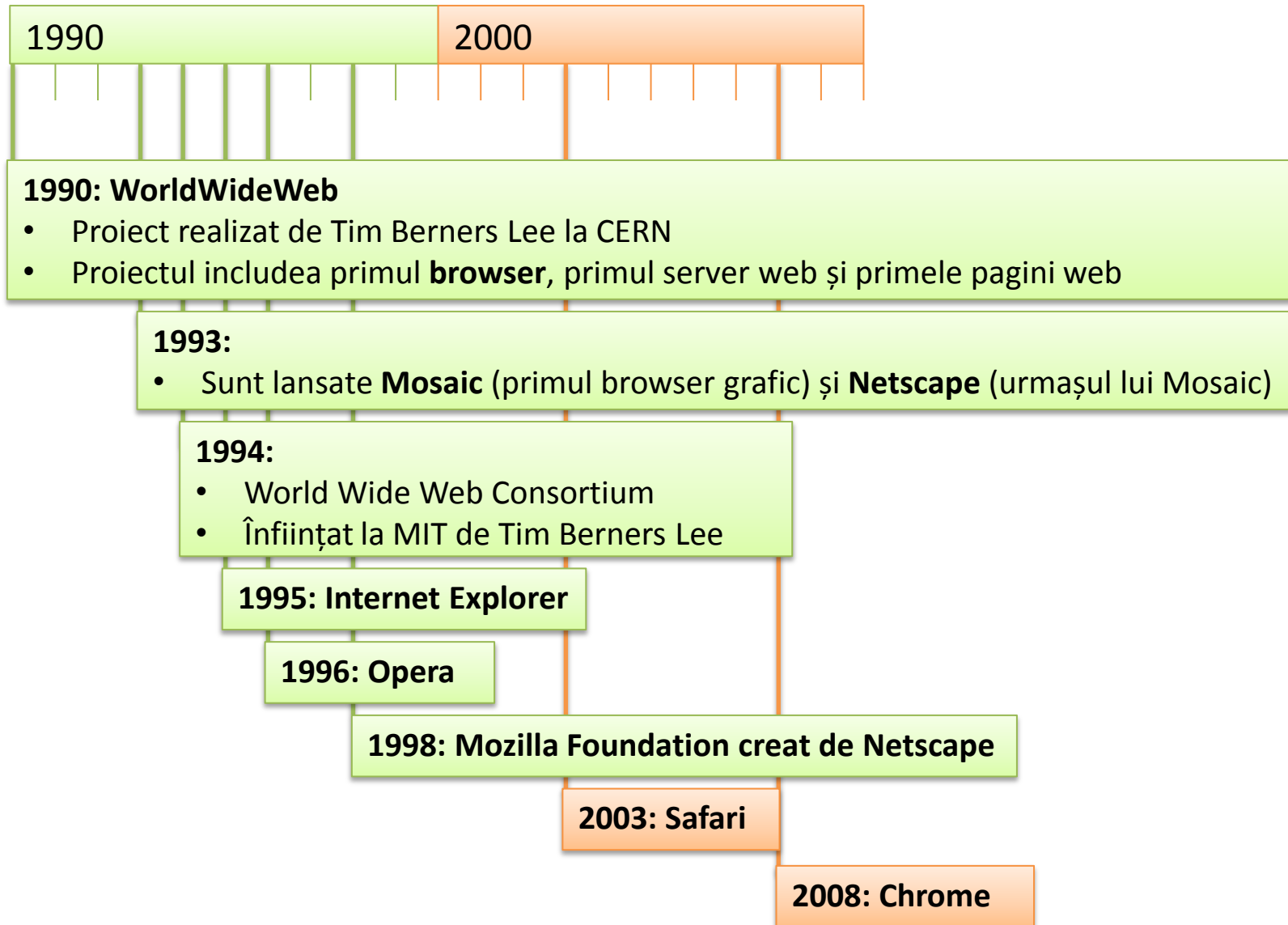
WWW

- Istoric
- Tehnologii
- HTTPS și certificate



- World Wide Web
- Rețea de resurse interconectate prin **hyperlink**-uri
 - Pagini html statice sau generate dinamic
 - Documente css
 - Imagini
 - Etc.
- Oferă resurse identificate prin **URI**
- Resursele sunt accesat prin **browsere**
- Care este diferența dintre Internet și WWW?
 - **R:** Internetul este o rețea de dispozitive. WWW este un serviciu oferit de o parte din această rețea.





- Uniform Resource Identifier
 - URN: Uniform Resource Name
 - URL: Uniform Resource Locator
- Șir folosit pentru identificarea unică a unei resurse în WWW
- Sintaxa unui URL:

protocol://user@domeniu:port/cale?interogări#fragment

<http://cs.curs.pub.ro/2011/course/view.php?id=215>

<http://141.85.241.139>

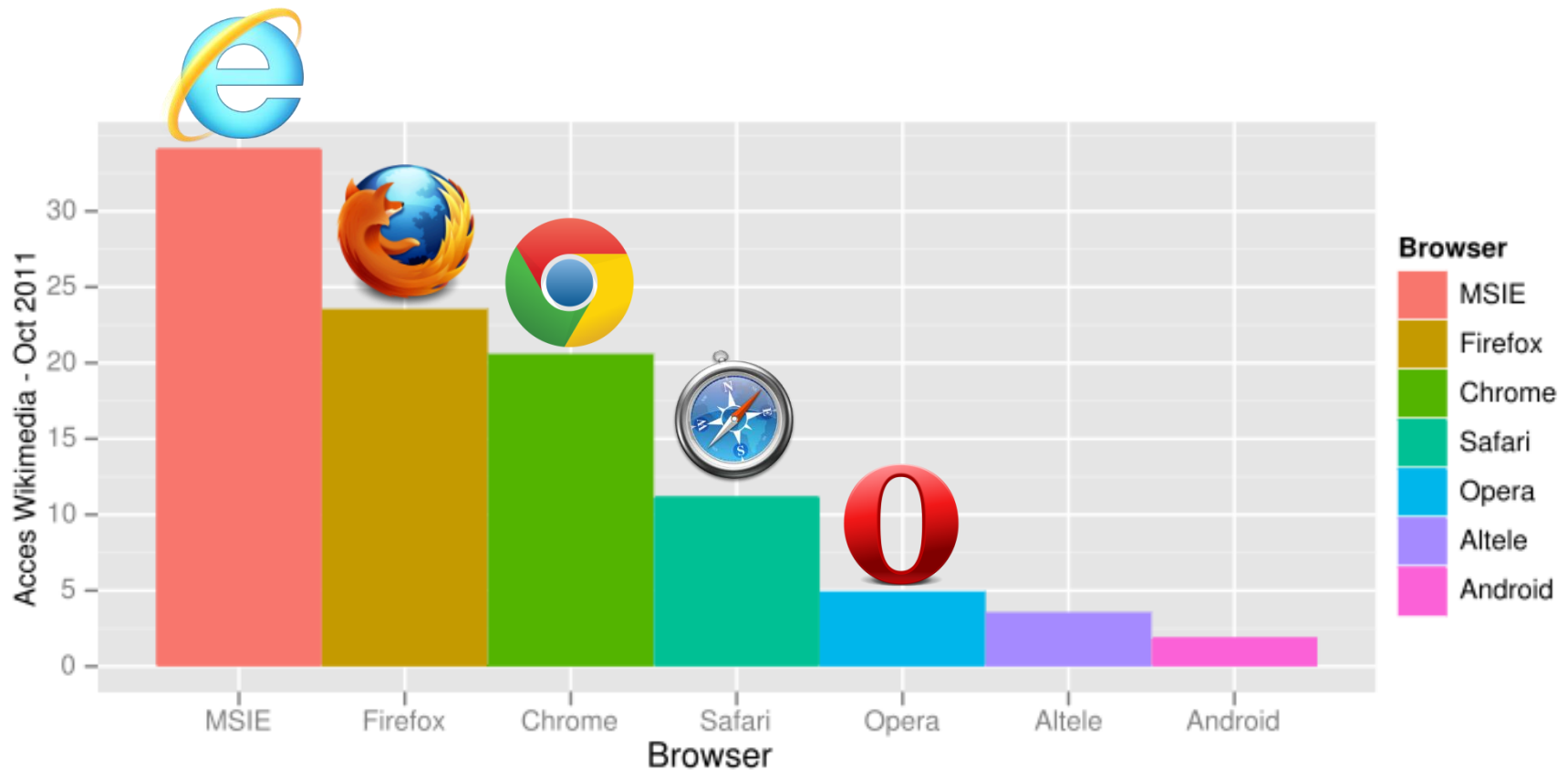
http://dynagen.org/tutorial.html#_Toc193248007

<telnet://student@example.com:25>

[https://\[2001:b30:800:f011:192:168:6:139\]/login/index.html](https://[2001:b30:800:f011:192:168:6:139]/login/index.html)

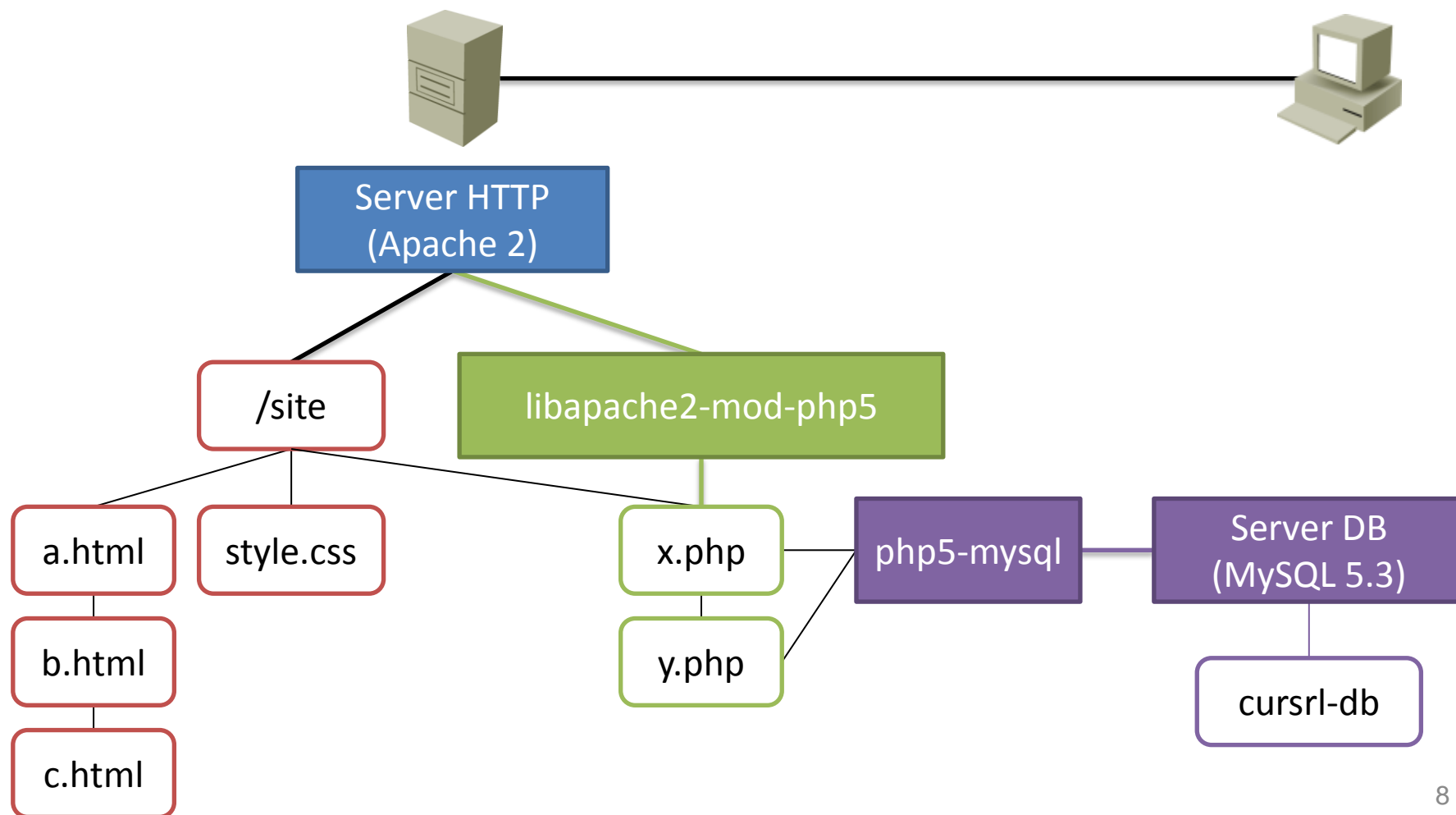
Browser Wars

- 1990: WorldWideWeb – primul browser (scris de Tim Berners Lee)
- 1993: Mosaic – primul browser popular (ulterior numit Netscape)
- Anii 90: Primul browser war (Netscape versus Internet Explorer)



- Un site din WWW adesea folosește o multitudine de tehnologii

curs-rl.ro (141.85.241.139)



- Clasificare în funcție de locul unde are loc interpretarea codului

Server



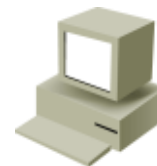
php

ruby

python

ASP

Client



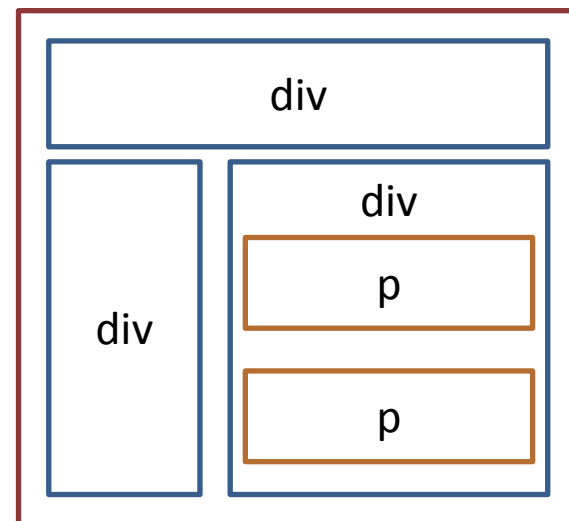
Javascript

AJAX

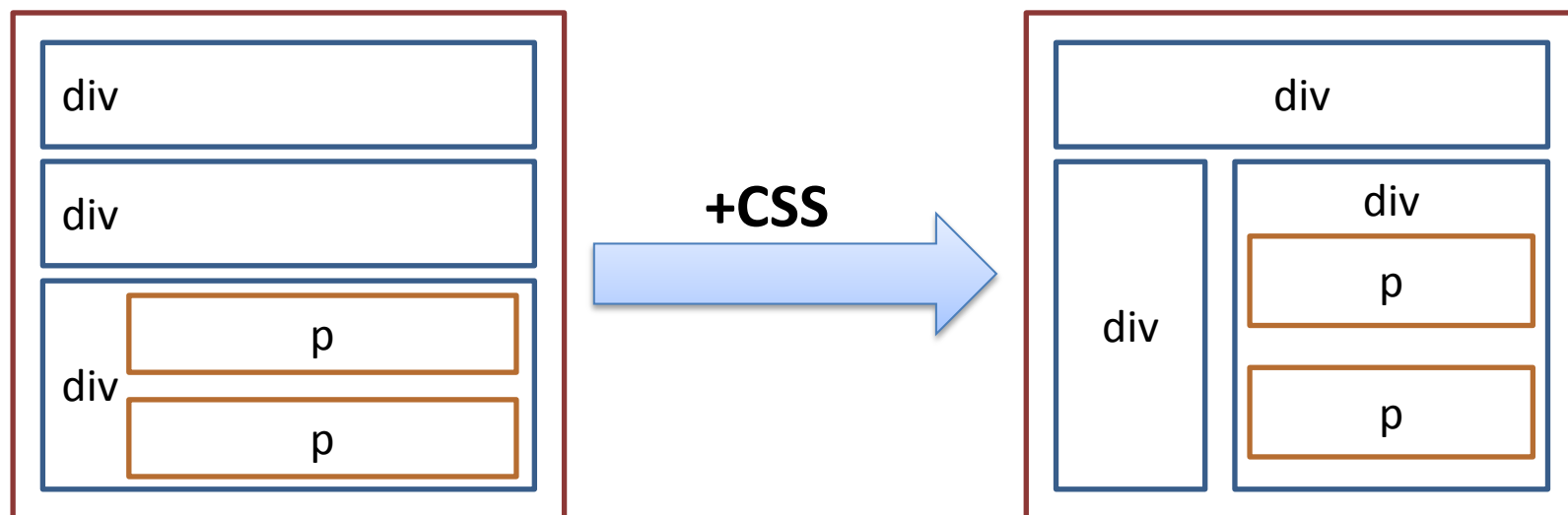
- Dezvoltat de Apache Software Foundation
- Cel mai popular server HTTP din Internet
 - 63% din site-uri foloseau Apache2 în Mai 2011
- Server modular
 - Permite încărcarea și descărcarea modulelor în funcție de cerințe
 - Exemple de module:
 - php5 – permite interpretarea codului php
 - wsgi – permite interpretarea codului python
 - userdir – publică pe server pentru fiecare utilizator un director în care acesta poate pune fișiere



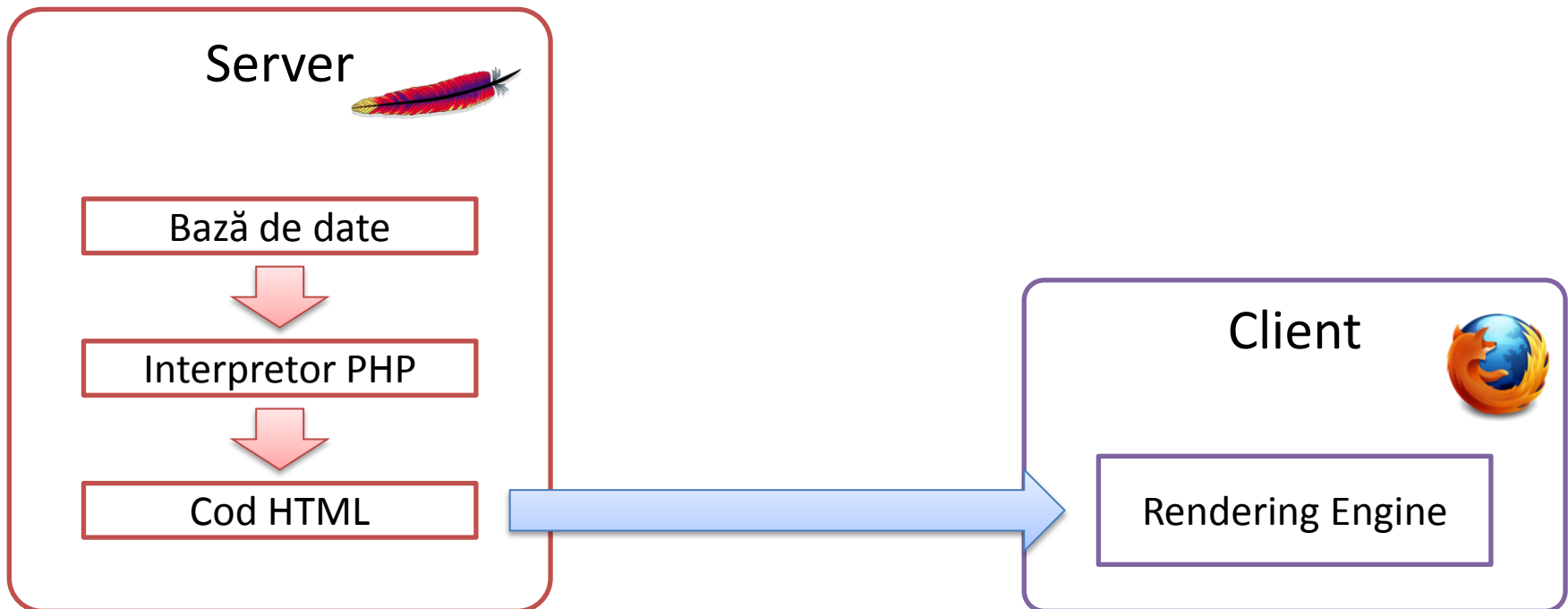
- Primul draft: CERN în 1991
- HyperText Markup Language
- Limbaj pentru descrierea structurii unei pagini
- Specificația HTML5 este în dezvoltare
- XHTML
 - eXtensible HyperText Markup Language
 - Variantă XML a HTML
 - Sintaxă mai strictă decât HTML
 - Mai ușor de parsat
 - În paralel cu HTML5 este definit și XHTML5



- Cascading Style Sheets
- Limbaj pentru particularizarea aspectului structurilor HTML
- Versiune: CSS3



- Majoritatea site-urilor actuale se bazează pe generarea de pagini dinamice
- Codul interpretat (PHP, Ruby, Python) are rolul de a construi codul HTML trimis clientului



- Codul php creează în mod dinamic o pagină statică
- Pentru a crea comportament dinamic într-o pagină (animații, evenimente), codul trebuie executat pe client
- Javascript este folosit pentru a controla elementele din pagină
- Javascript nu este utilizat doar în cadrul site-urilor:
 - Documente pdf
 - Module de browser



- Hypertext Transfer Protocol
- Portul TCP 80
- Versiunea actuală: 1.1
- Arhitectură client-server
- Clientul folosește metode pentru a comunica



Metodă	Rol
GET	Cere o reprezentare a unei resurse
HEAD	Cere informații despre o resursă (fără conținut)
POST	Trimite date pentru procesare de către server
OPTIONS	Cere operațiile suportate de server pentru o resursă

- Protocolul HTTP nu are stare
 - Accesări succesive ale unor pagini în relație logică nu puteau beneficia de informații de sesiune
- Cookies sunt un mecanism pentru persistența stării:
 - Termenul inițial era magic cookie și a fost introdus de inginerii Netscape
 - Obiecte similare unor variabile ce sunt reținute de browser
 - Asociate unui domeniu
 - Pot fi citite sau scrise de site prin intermediul javascript



Exemplu de cerere HTTP

Metodă

Versiune

Locație

Nume de domeniu

Cookies

```
Hypertext Transfer Protocol
GET /2011/ HTTP/1.1\r\n
[Expert Info (Chat/Sequence): GET /2011/ HTTP/1.1\r\n]
[Message: GET /2011/ HTTP/1.1\r\n]
[Severity level: chat]
[Group: Sequence]
Request Method: GET
Request URI: /2011/
Request Version: HTTP/1.1
Host: cs.curs.pub.ro\r\n
User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64; rv:8.0) Gecko/20100101 Firefox/8.0\r\n
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8\r\n
Accept-Language: en-us,en;q=0.5\r\n
Accept-Encoding: gzip, deflate\r\n
Accept-Charset: ISO-8859-1,utf-8;q=0.7,*;q=0.7\r\n
Connection: keep-alive\r\n
[truncated] Cookie: __utma=246491611.14351673.1318718530.1322913674.1322950554.80; __utmz=
Cache-Control: max-age=0\r\n
\r\n
```

De ce e utilă trimiterea numelui de domeniu în cerere?

- Se bazează pe perechi de chei aflate într-o relație matematică:
 - Cheia publică (K^+)
 - Cheia privată (K^-)
- Dându-se un mesaj M , există următoarea relații:

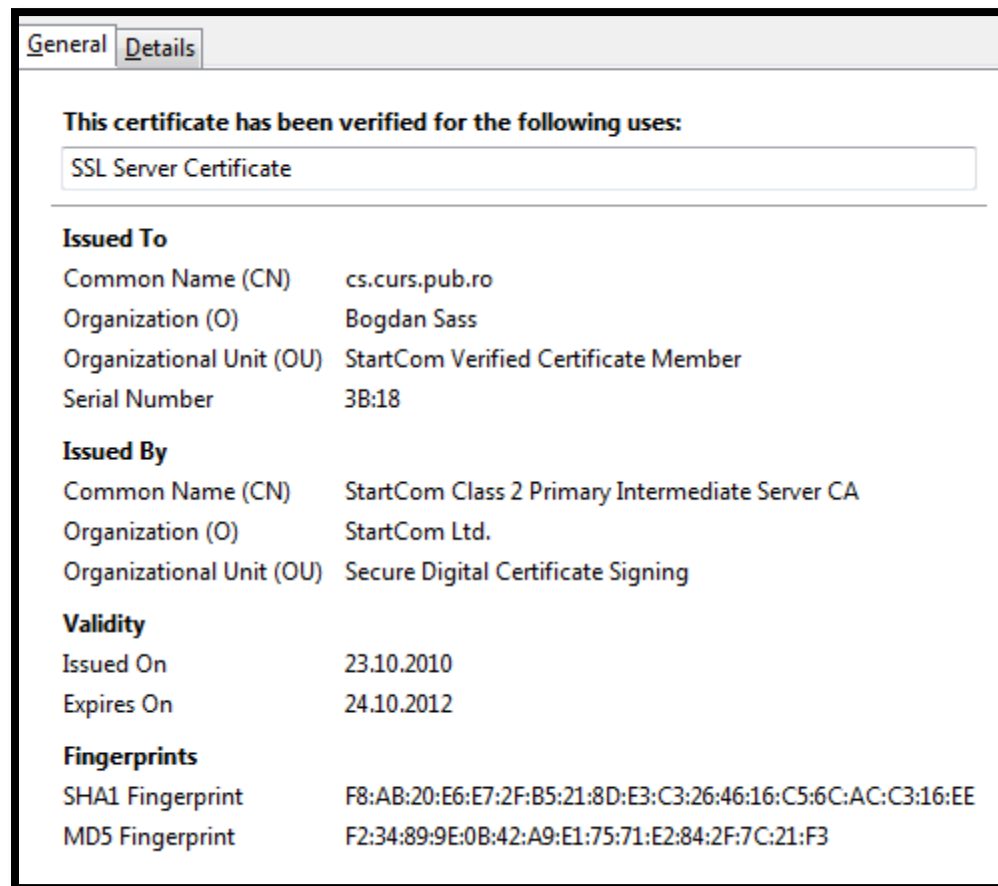
$$K^+(K^-(M)) = M$$

$$K^-(K^+(M)) = M$$

- Cu alte cuvinte, un client poate:
 - Avea configurată pe server cheia sa publică K^+ (de un administrator de exemplu)
 - Cripta un mesaj cu K^-
 - Serverul va putea decripta mesajul cu K^+

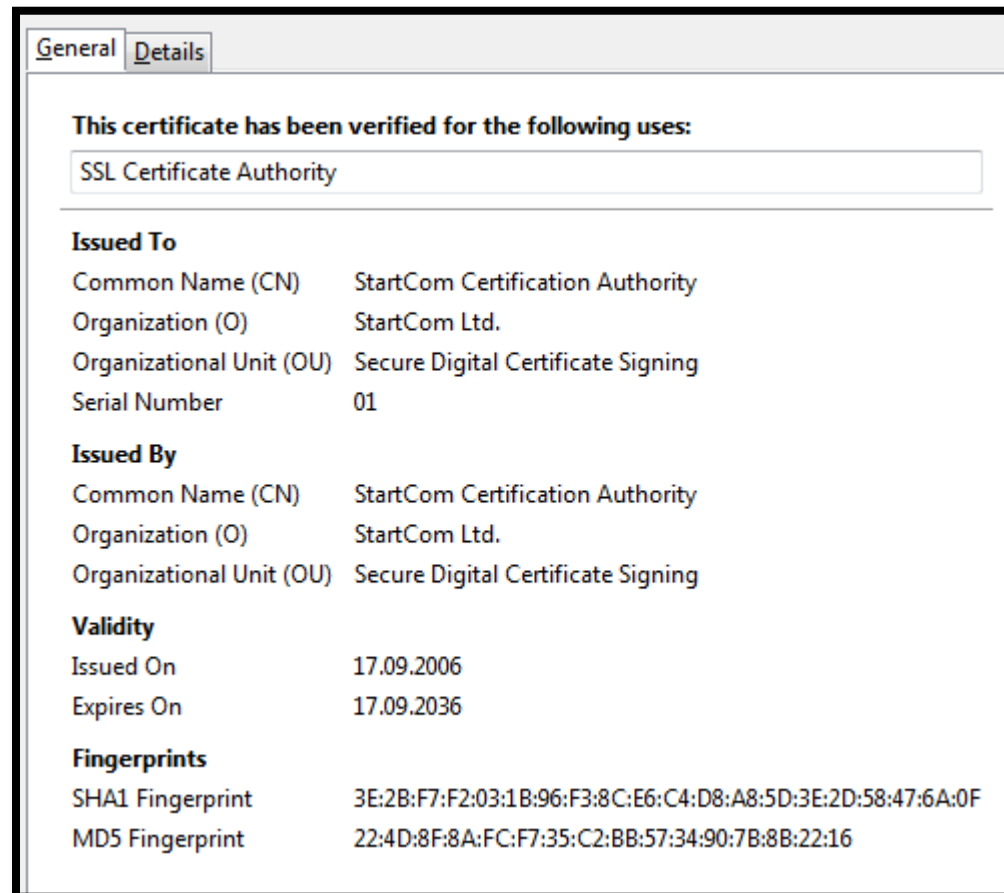
- HTTP Secure
- Portul TCP 443
- Folosește SSL/TLS pentru a stabili un canal criptat sigur între client și serverul HTTPS
- Site-ul se autentifică prin intermediul unui **certificat**
- Un certificat conține:
 - Numele site-ului (CN)
 - Numele organizației (O)
 - CA-ul care urmează să valideze certificatul (CA – CN, CA – O)
 - Data de emitere / expirare
 - Un hash pentru verificarea integrității
 - **Cheia publică a site-ului**

- Browser-ul clientului primește certificatul site-ului la accesarea acestuia
- Exemplu cs.curs.pub.ro în Firefox:

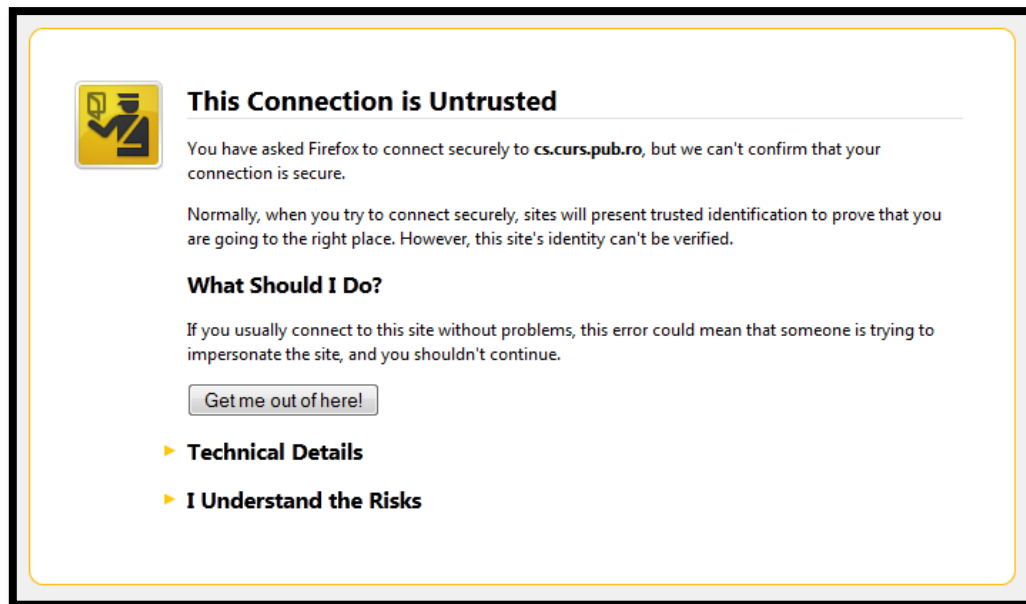


- HTTPS depinde de existență CA-urilor (Certificate Authority)
- Un CA este o entitate ce garantează autenticitatea unui certificat
- CA-urile semnează ierarhic
- În exemplul de mai sus:
- Certificat `cs.curs.pub.ro`
 - Semnat de către StartCom Class 2 Primary Intermediate Server CA
 - Semnat de către StartCom Certification Authority
- Browserele vin cu un set de CA-uri importante predefinite
 - În Firefox 8.0 (Windows):
 - Firefox/Options/Advanced/Encryption/View Certificates
 - Cele două CA-uri StartCom sunt incluse de la instalare deci accesarea `cs.curs.pub.ro` nu lansează avertizări

- Autenticitatea unui CA primar este garantată de el însuși
- Certificatele self signed ale CA-urilor importante vin cu browser-ul



- Dacă nu este cunoscut CA-ul site-ului (de exemplu certificatul este self-signed):
 - Browserul oferă o avertizare asupra faptului că site-ul nu poate fi autentificat
 - Poate semnaliza o încercare de phishing
 - Permite adăugarea unei excepții pentru certificatul respectivă



Instalare

1. Generare chei

2. Cerere

3. Semnare

4. Instalare

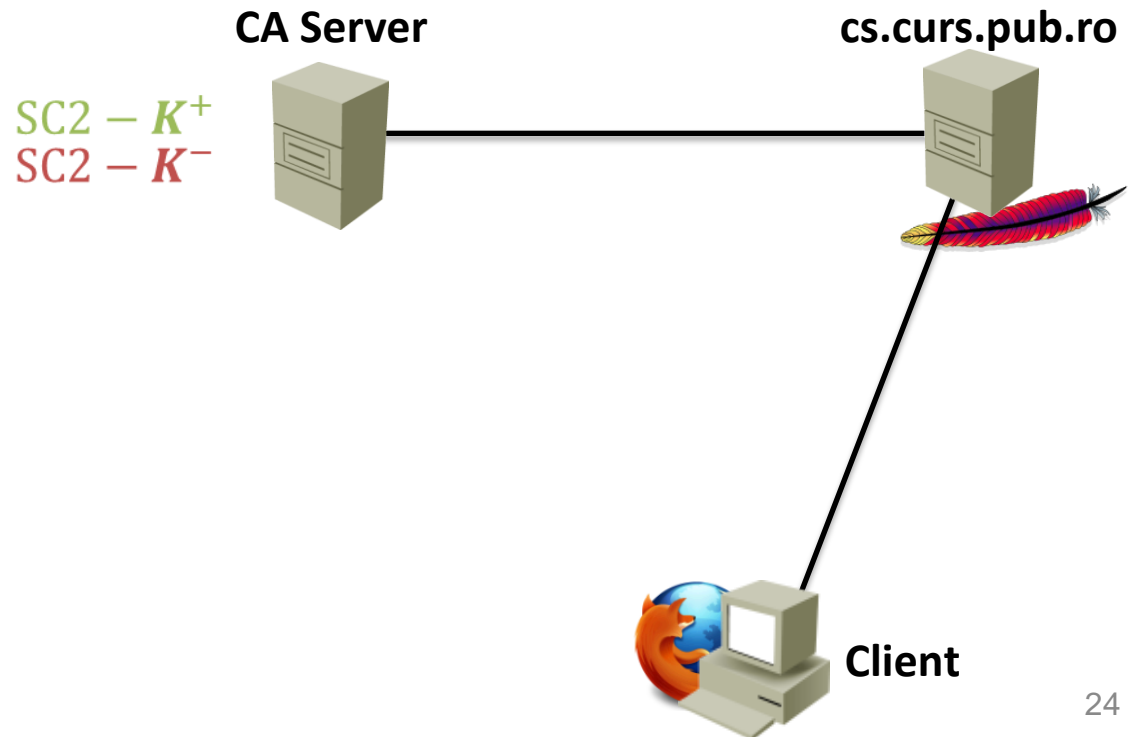
Utilizare

1. Primire certificat

2. Validare CA

3. Validare site

- Serverul CA-ului are deja generată o pereche de chei publice și private ($StartCom2 - K^+$, $StartCom2 - K^-$)



Instalare

1. Generare chei

2. Cerere

3. Semnare

4. Instalare

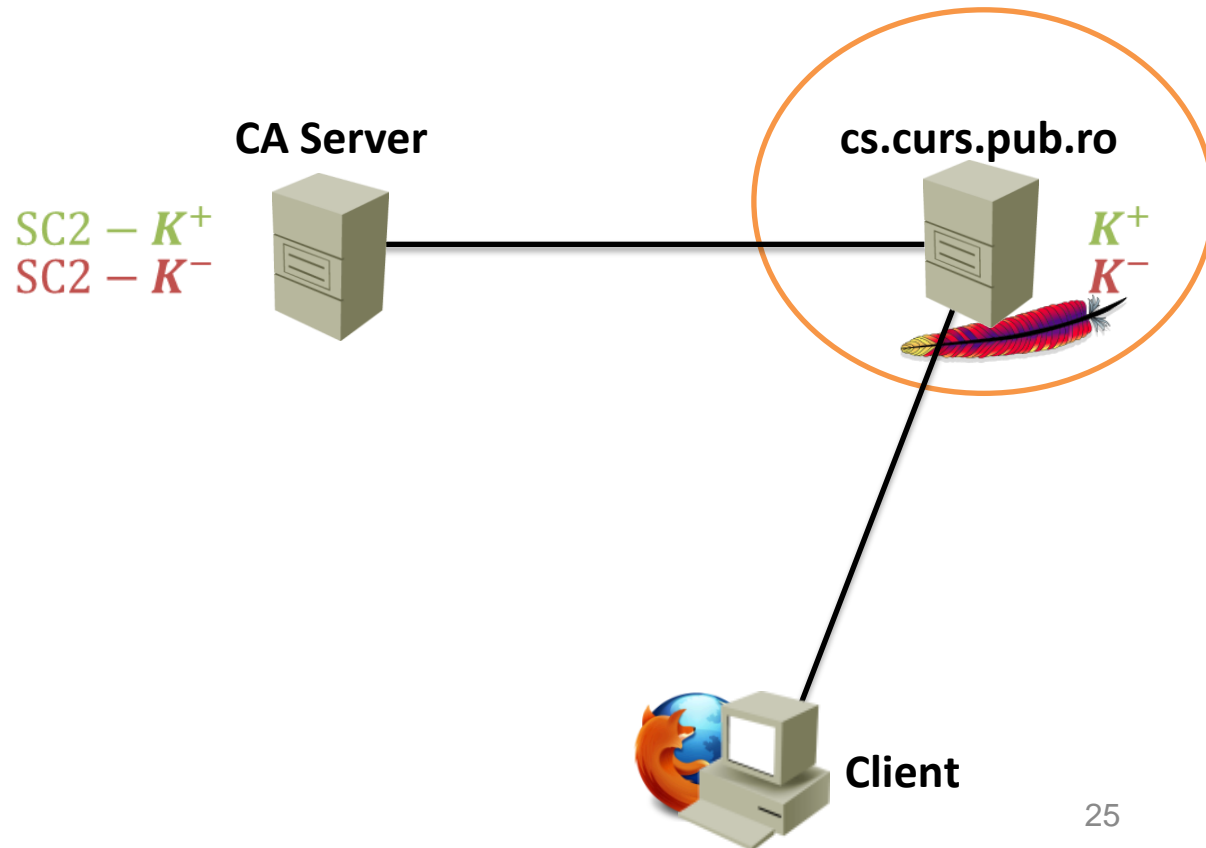
Utilizare

1. Primire certificat

2. Validare CA

3. Validare site

1. Serverul **cs.curs.pub.ro** generează o pereche de chei asimetrice (K^+ , K^-)



Instalare

1. Generare chei

2. Cerere

3. Semnare

4. Instalare

Utilizare

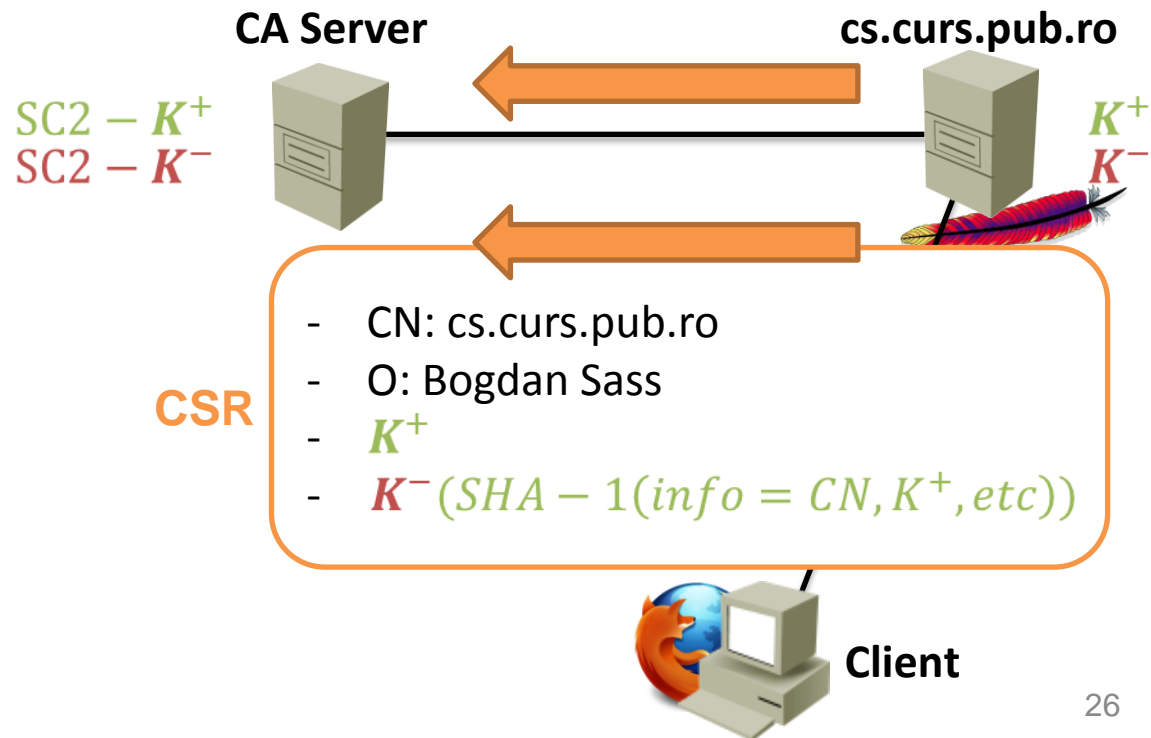
1. Primire certificat

2. Validare CA

3. Validare site

2. O cerere de semnare este trimisă către CA:

- cererea include detalii despre `cs.curs.pub.ro` și K^+
- cererea este semnată cu K^- înainte de trimitere (De ce?); Cum poate valida semnătura cu K^- CA-ul?
- **CSR** = Certificate Signing Request



Instalare

1. Generare chei

2. Cerere

3. Semnare

4. Instalare

Utilizare

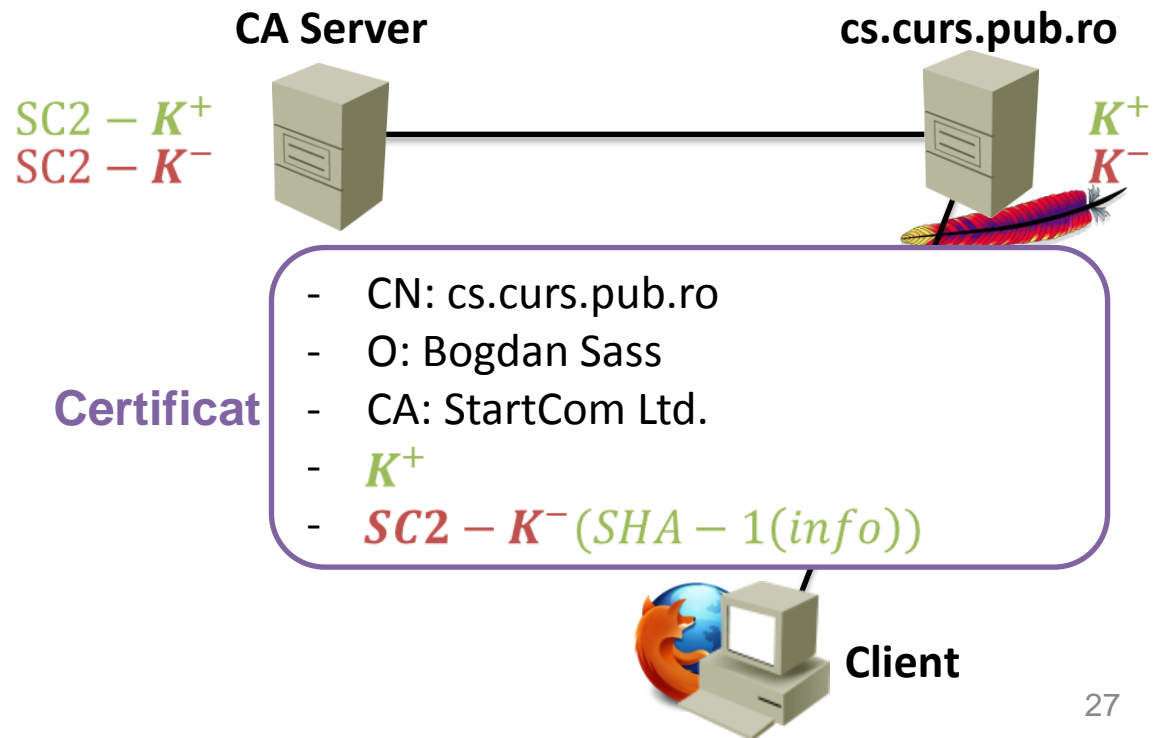
1. Primire certificat

2. Validare CA

3. Validare site

3. CA-ul folosește cheia sa privată pentru a semna cererea

- Semnarea constă în calcularea unui hash peste informațiile din cerere
- Hash-ul este apoi criptat cu $StartCom2 - K^-$ și inclus în certificat



Instalare

1. Generare chei

2. Cerere

3. Semnare

4. Instalare

Utilizare

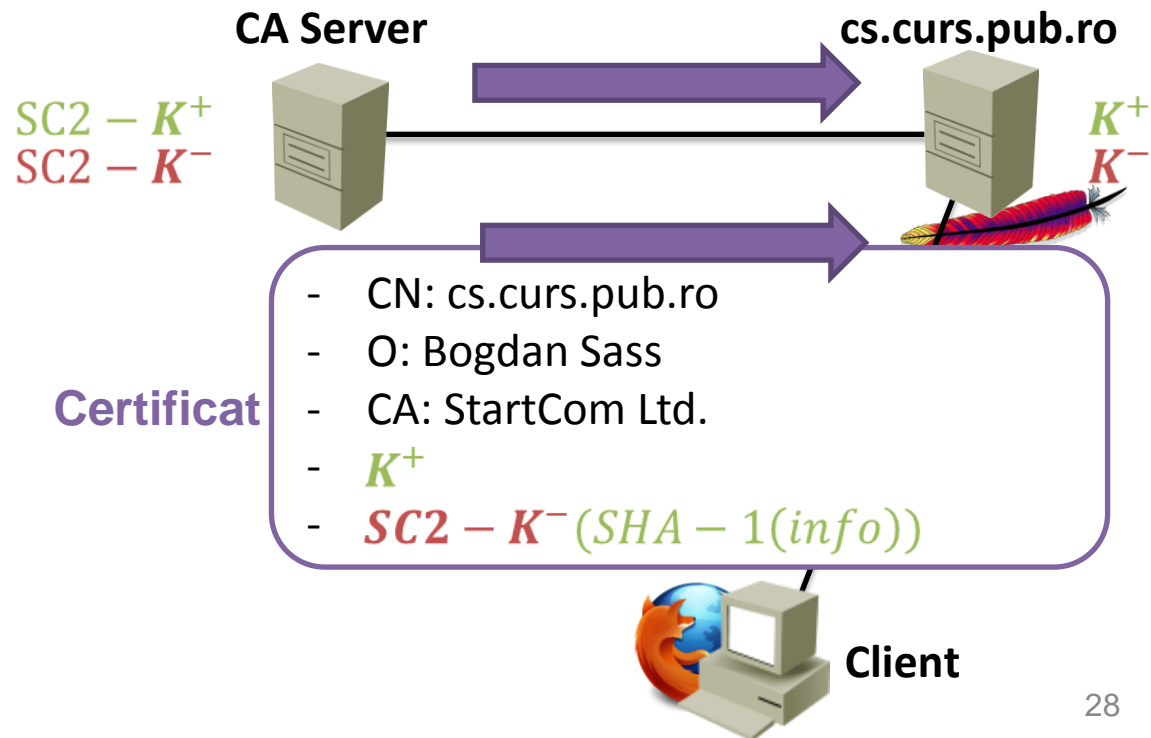
1. Primire certificat

2. Validare CA

3. Validare site

4. CA-ul trimite certificatul semnat serverului

- Certificatul trebuie configurat în serverul HTTPS folosit
- La accesarea serverului, certificatul va fi prezentat clienților



Instalare

1. Generare chei

2. Cerere

3. Semnare

4. Instalare

Utilizare

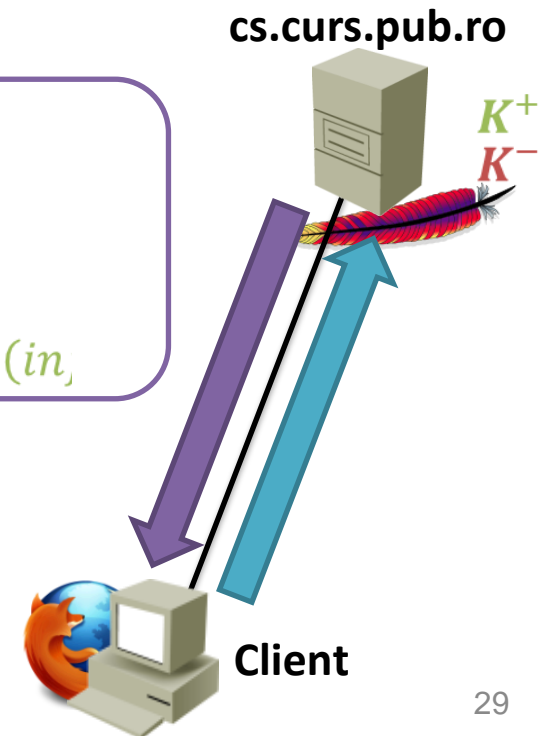
1. Primire certificat

2. Validare CA

3. Validare site

1. Clientul accesează prin HTTPS cs.curs.pub.ro
 - Serverul îi trimite certificatul obținut anterior

- CN: cs.curs.pub.ro
- O: Bogdan Sass
- CA: StartCom Ltd.
- K^+
- $SC2 - K^-(SHA-1(in_)$



Instalare

1. Generare chei

2. Cerere

3. Semnare

4. Instalare

Utilizare

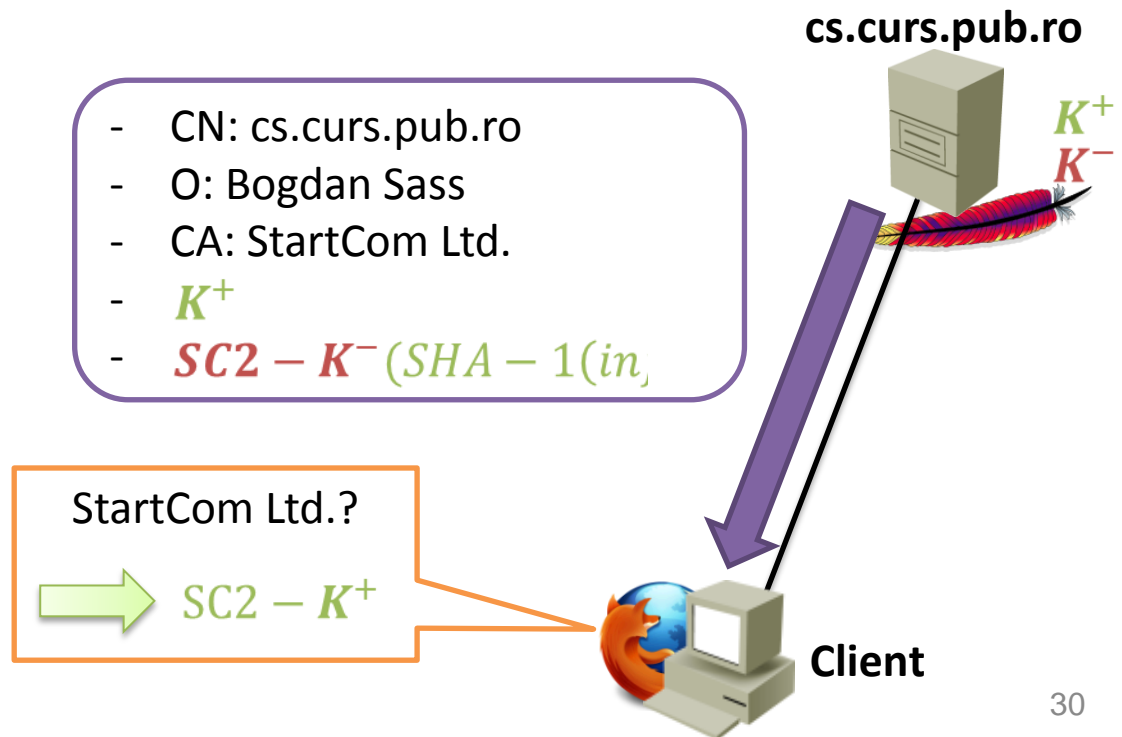
1. Primire certificat

2. Validare CA

3. Validare site

2. Clientul se uită în certificatul site-ului care e CA-ul care a semnat

- După ce e obținut CA-ul, se uită în baza de date locală a browser-ului dacă CA-ul este de încredere
- Dacă da, clientul ia din certificatul local **StartCom2** – K^+ pentru a verifica integritatea și autenticitatea mesajului



Instalare

1. Generare chei

2. Cerere

3. Semnare

4. Instalare

Utilizare

1. Primire certificat

2. Validare CA

3. Validare site

3. Certificatul e validat:

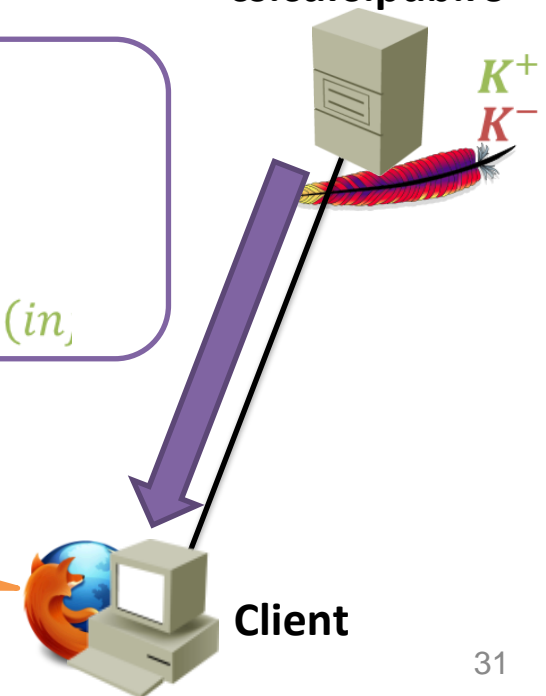
- Sa calculează hash-ul $SHA - 1(Info)$ pe câmpurile din certificat
- Dacă relația de mai jos e adevărată, certificatul este validat și browser-ul permite accesarea site-ului

$$SC2 - K^+ (SC2 - K^- (SHA - 1(Info))) = SHA - 1(Info)$$

- CN: cs.curs.pub.ro
- O: Bogdan Sass
- CA: StartCom Ltd.
- K^+
- $SC2 - K^- (SHA - 1(in_$

Certificatul e valid.
Site-ul poate fi
accesat.

cs.curs.pub.ro



- Unde eșuează verificarea dacă site-ul minte referitor la identitatea sa?

- R: Utilizare – pasul 3

- $SC2 - K^-(SHA - 1(Info))$ nu poate fi modificat deoarece e semnat cu o cheie privată necunoscută (a CA-ului)
- Când e decriptat hash-ul cu cheia publică a CA-ului, va fi diferit de ce e calculat acum pe mesaj

$$SC2 - K^+ \left(SC2 - K^-(SHA - 1(Info)) \right) = SHA - 1(Info) \neq SHA - 1(Info - mod)$$

- De ce nu va funcționa un self signed certificate?

- R: Utilizare – pasul 2. Nu va fi găsit $cs.curs.pub.ro - K^+$ în browser.

- Ce se întâmplă dacă site-ul spune că e semnat de StartCom2, dar de fapt e self-signed?

- R: Utilizare – pasul 3

- Se va încerca aplicarea unei chei publice peste un mesaj pe care s-a aplicat o cheie privată din altă pereche
- Autentificarea va eșua deoarece nu va fi respectată egalitatea

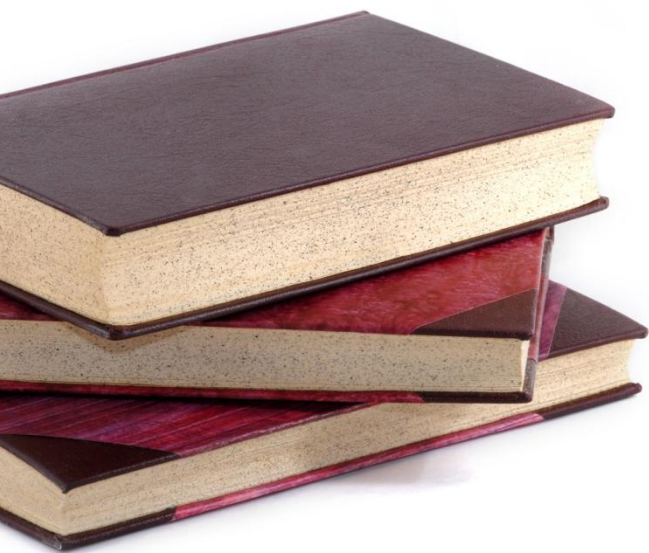
$$SC2 - K^+ \left(K^-(SHA - 1(Info)) \right) \neq SHA - 1(Info)$$

- Ce se întâmplă dacă e compromis **StartCom2 – K^-** ?
 - R:
 - Oricine va putea construi certificate validate de browser.
 - Pentru astfel de situații se folosesc liste de revocare (liste ce spun ce CA-uri nu mai sunt de încredere)
 - Exemplu de compromitere: Sony PlayStation 3 Private Root Key (pentru certificarea digitală a pachetelor software)

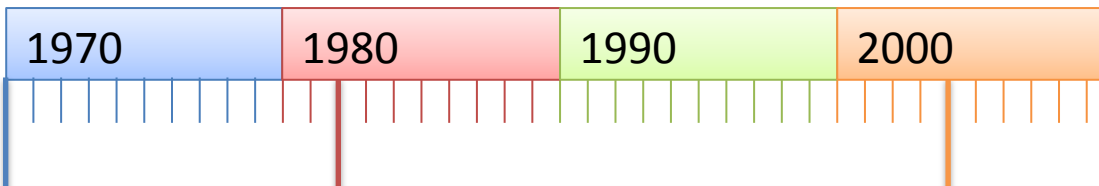


DNS

- Istoric
- Structură
- Servere DNS
- Interogări DNS



- Domain Name System
- Sistem ierarhic distribuit de asocieri nume – adresă
- Poate răspunde la întrebările:
 - Care sunt adresele IPv4 și IPv6 ale lui cs.curs.pub.ro?
 - Cine este 141.85.241.139?
 - Ce IP are serverul de mail pentru domeniul pub.ro?
- Port TCP și UDP 53
- Model client-server
- Serverul stochează informații numite **înregistrări**
- Clienții interoghează serverul pentru a afla valorile din înregistrări



Since forever ☺ - HOSTS.TXT

- Fișier static ce tine mapări Nume – Adresă
- Folosit în sistemele actuale în special pentru maparea localhost – 127.0.0.1
- Începând cu anii 80 nu mai putea face față numărului în creștere de host-uri

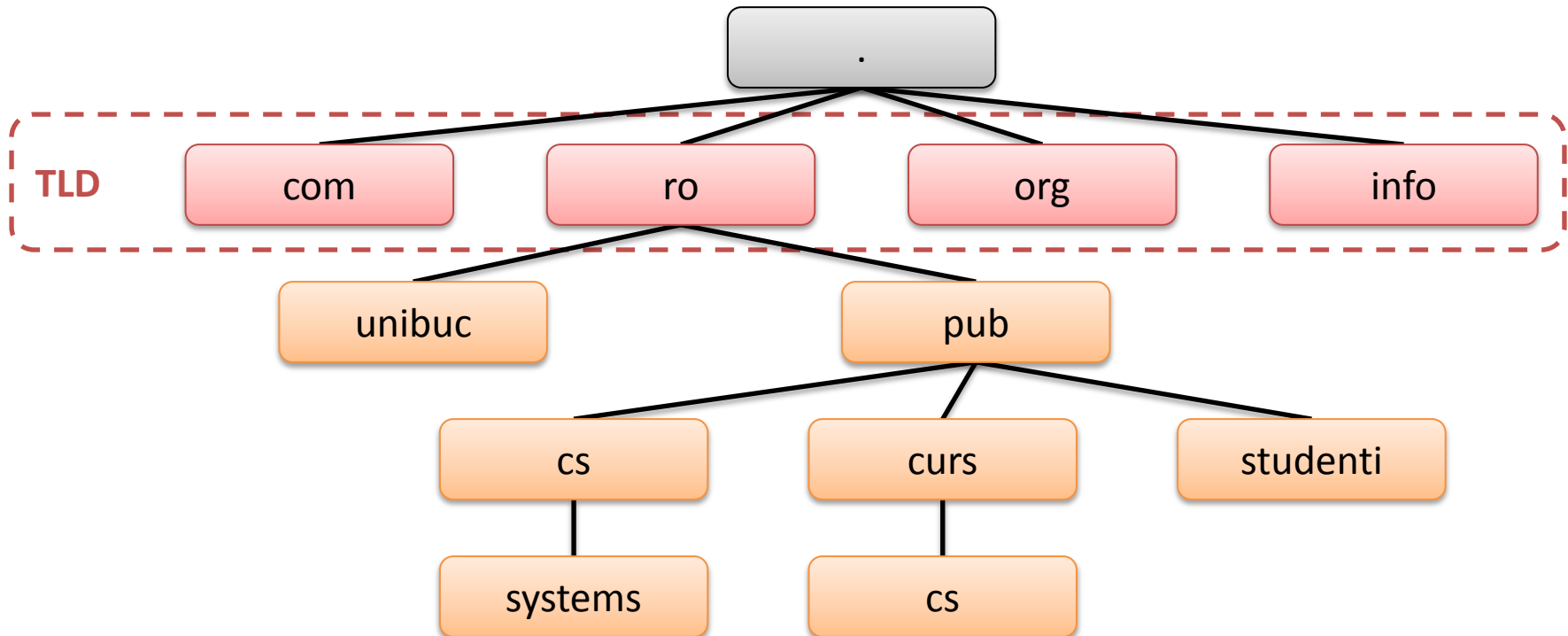
1982 – DNS (Domain Name System)

- Inventat de Paul Mockapetris
- Sistem distribuit de mapări Nume - adresă
- Folosit și pentru a obține alte informații

2005 – DNSSEC

- DNS nu are mecanisme de securitate
- Un atacator poate rescrie IP-ul din răspunsul DNS
- DNSSEC garantează autenticitatea (dar nu și confidențialitatea) răspunsului

- ICANN – Internet Corporation for Assigned Names and Numbers
- ICANN:
 - administrează lista de TLD (Top Level Domains)
 - delegă altor organizații autoritatea de a alocă numele de domenii dintr-un TLD



-
- ```

graph TD
 Root[.] --- com[com]
 Root --- ro[ro]
 Root --- org[org]
 Root --- info[info]
 ro --- unibuc[unibuc]
 ro --- pub[pub]
 pub --- cs1[cs]
 pub --- curs[curs]
 pub --- studenti[studenti]
 cs1 --- systems[systems]
 curs --- cs2[cs]

```
- Diagram illustrating a hierarchical DNS structure for a university:
- Root (grey box with a dot) branches into four Top Level Domains (TLDs): **com**, **ro**, **org**, and **info** (red boxes).
  - The **ro** TLD branches into **unibuc** (orange box) and **pub** (orange box).
  - The **pub** domain is further divided into two subdomains:
    - ns1.cs.pub.ro** (purple box): Contains **cs** (orange box) and **systems** (orange box).
    - pub.pub.ro** (blue box): Contains **pub** (orange box), **curs** (orange box), and **studenti** (orange box).
  - The **pub** domain further branches into **cs** (orange box), **curs** (orange box), and **studenti** (orange box).
  - The **cs** domain further branches into **systems** (orange box).
  - The **curs** domain further branches into **cs** (orange box).

## Master/Slave

- Ambele servere răspund cererilor de DNS

## Server forwarder

- Ajută alte servere DNS din rețeaua locală cu rezolvarea cererilor externe
- Folosit pentru a masca serverele DNS locale și pentru a reduce traficul extern prin caching
- Un server trimite la forwarder o cerere dacă nu este autoritar peste domeniul cerut și nu are informația în cache

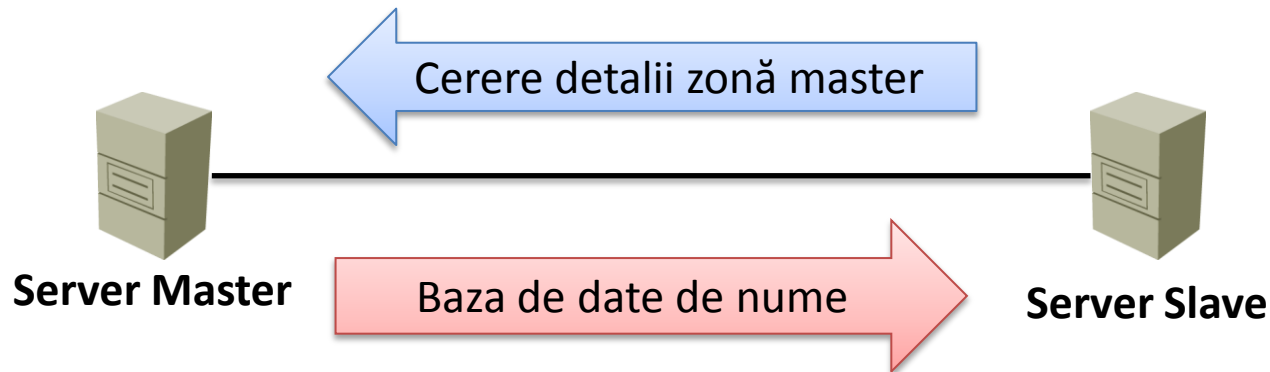
## Caching-only

- Server autoritar doar pe domeniul 0.0.0.127-in.addr.arpa
- Face caching de înregistrări des folosite pentru a reduce traficul DNS

## Root Server

- Administrează TLD-uri

- Fiecare domeniu trebuie să aibă o zonă de master pentru a putea genera răspunsuri autoritare pe domeniul gestionat
- **Serverul Slave** va contacta periodic **Serverul Master** pentru a obține lista de înregistrări configurate
- Zona slave trebuie să precizeze explicit zona master





- **Recursiv**
  - Cererea trebuie rezolvată de serverul interogat
  - Trimise în general de aplicațiile client
- **Nerecursiv**
  - Cererea primește răspuns doar dacă serverul interogat:
    - Are intrarea în Cache
    - E autoritar pentru cerere
  - Dacă nu sunt respectate condițiile:
    - Este returnat faptul că nu poate fi rezolvată cererea
    - Se indică un alt server de nume

- Pentru a cere anumite informații se poate folosi comanda host:

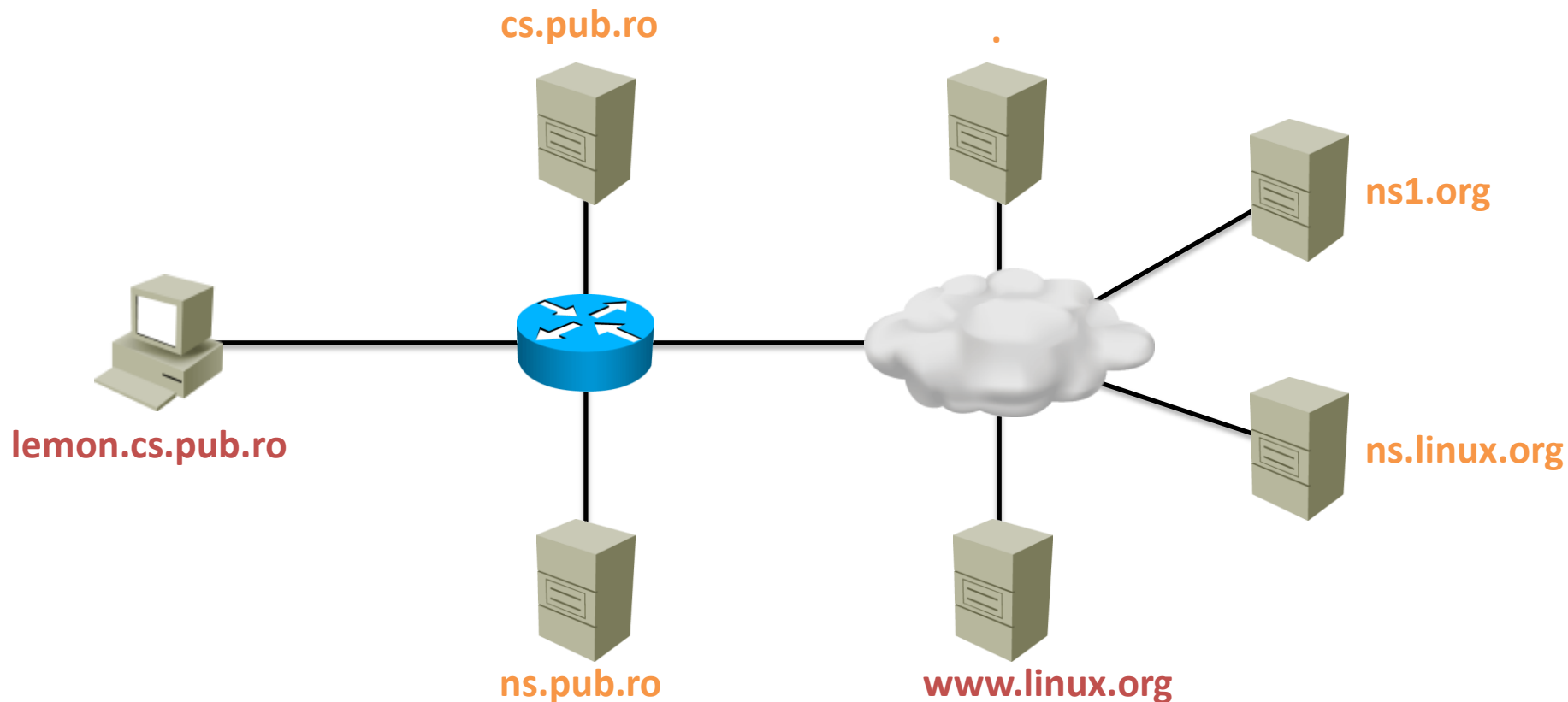
```
linux> host -t MX cs.pub.ro
```

| Tip         | Rol                                                                  | Exemplu                                                                                            |
|-------------|----------------------------------------------------------------------|----------------------------------------------------------------------------------------------------|
| <b>A</b>    | Descoperire adresă IPv4                                              | cs.pub.ro has address 141.85.227.111                                                               |
| <b>AAAA</b> | Descoperire adresă IPv6                                              | cs.pub.ro has no AAAA record                                                                       |
| <b>NS</b>   | Name Server – identifică serverele de nume asociat cu un domeniu     | cs.pub.ro name server ns1.cs.pub.ro.<br>cs.pub.ro name server ns2.cs.pub.ro.                       |
| <b>SOA</b>  | Start of Authority – întoarce diverși parametri specifici zonei      | cs.pub.ro has SOA record ns1.cs.pub.ro.<br>admin.cs.pub.ro. 2011120301 28800 7200<br>604800 86400  |
| <b>MX</b>   | Mail Exchanger – identifică serverele de mail asociate cu un domeniu | cs.pub.ro mail is handled by 5 mail.cs.pub.ro.<br>cs.pub.ro mail is handled by 20 vmail.cs.pub.ro. |
| <b>PTR</b>  | Pointer – folosite pentru rezolvare inversă                          | 111.227.85.141.in-addr.arpa domain name pointer<br>cursuri.cs.pub.ro.                              |

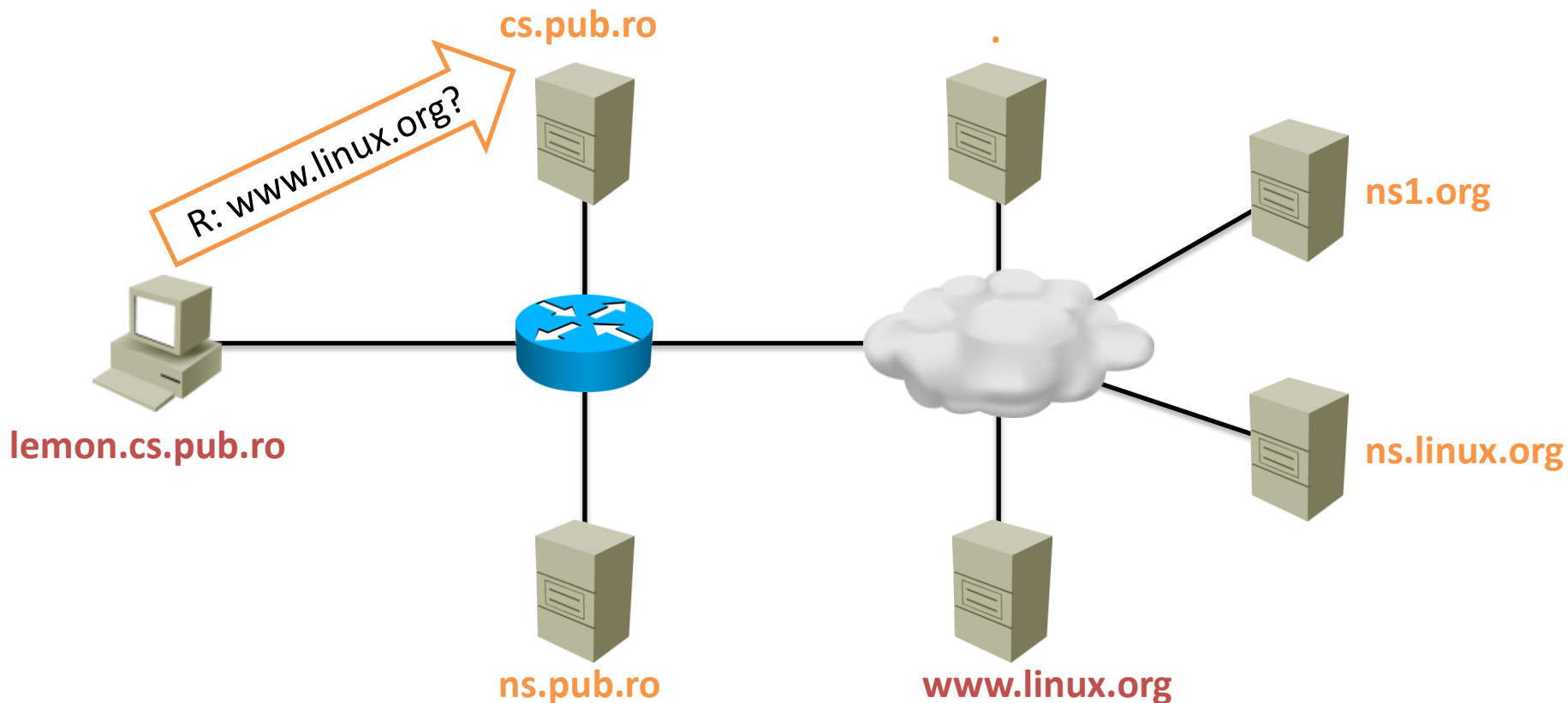
# Exemplu fișier de zonă

```
;cs.pub.ro.db
$ORIGIN pub.ro.
cs IN SOA ns.cs.pub.ro. nsmaster.cs.pub.ro. (
 2011120301 ; Serial
 8H ; Refresh
 2H ; Retry
 1W ; Expire
 1D ; TTL
)
 TXT "Computer Science Department"
 NS ns1.cs
 NS ns2.cs
 A 141.85.227.111
$ORIGIN cs.pub.ro.
 MX 5 mail.cs.pub.ro
 MX 20 vmail.cs.pub.ro
ns1 A 141.85.226.5
ns2 A 141.85.241.113
mail A 141.85.227.3
vmail A 141.85.227.3
```

- Stația **lemon.cs.pub.ro** dorește să acceseze serverul **www.linux.org**
- Stația **lemon.cs.pub.ro** are configurat ca server DNS **cs.pub.ro**

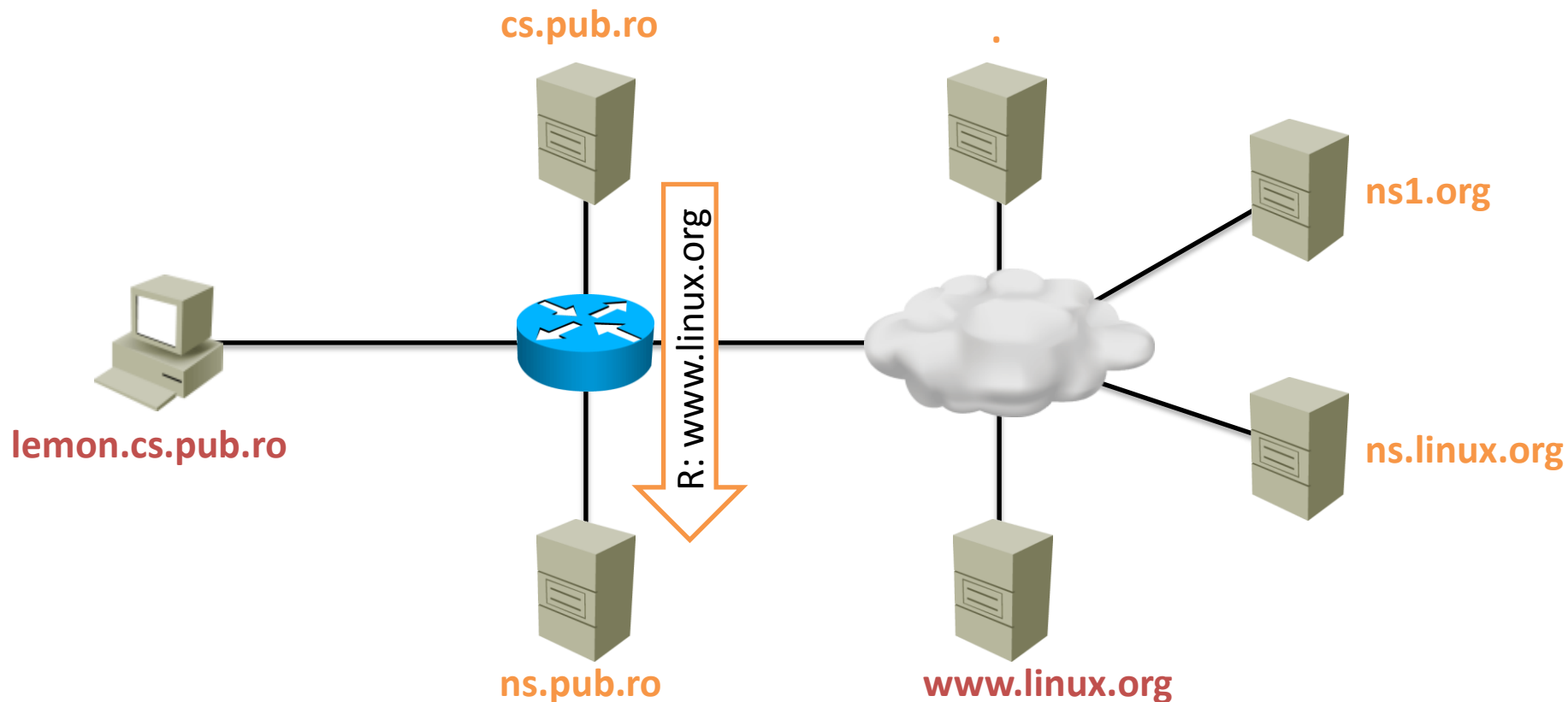


- Stația **lemon.cs.pub.ro** trimite o cerere recursivă către **cs.pub.ro**
- **cs.pub.ro** verifică dacă este autoritar peste **www.linux.org** → nu
- **cs.pub.ro** verifică dacă are în cache **www.linux.org** → nu



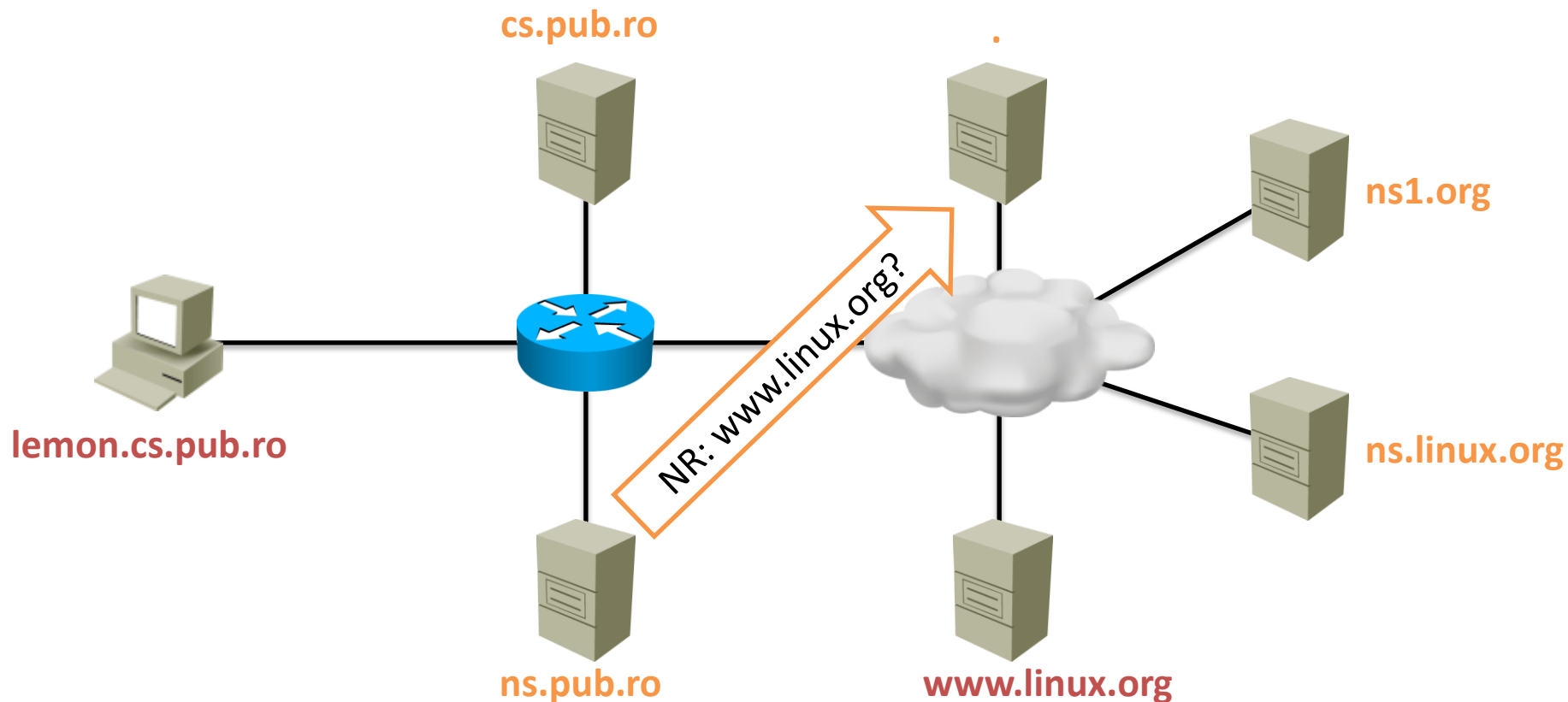
## Exemplu – Pasul 2

- **cs.pub.ro** are **ns.pub.ro** configurat ca forwarder
- **cs.pub.ro** trimite o cerere recursivă către **ns.pub.ro**



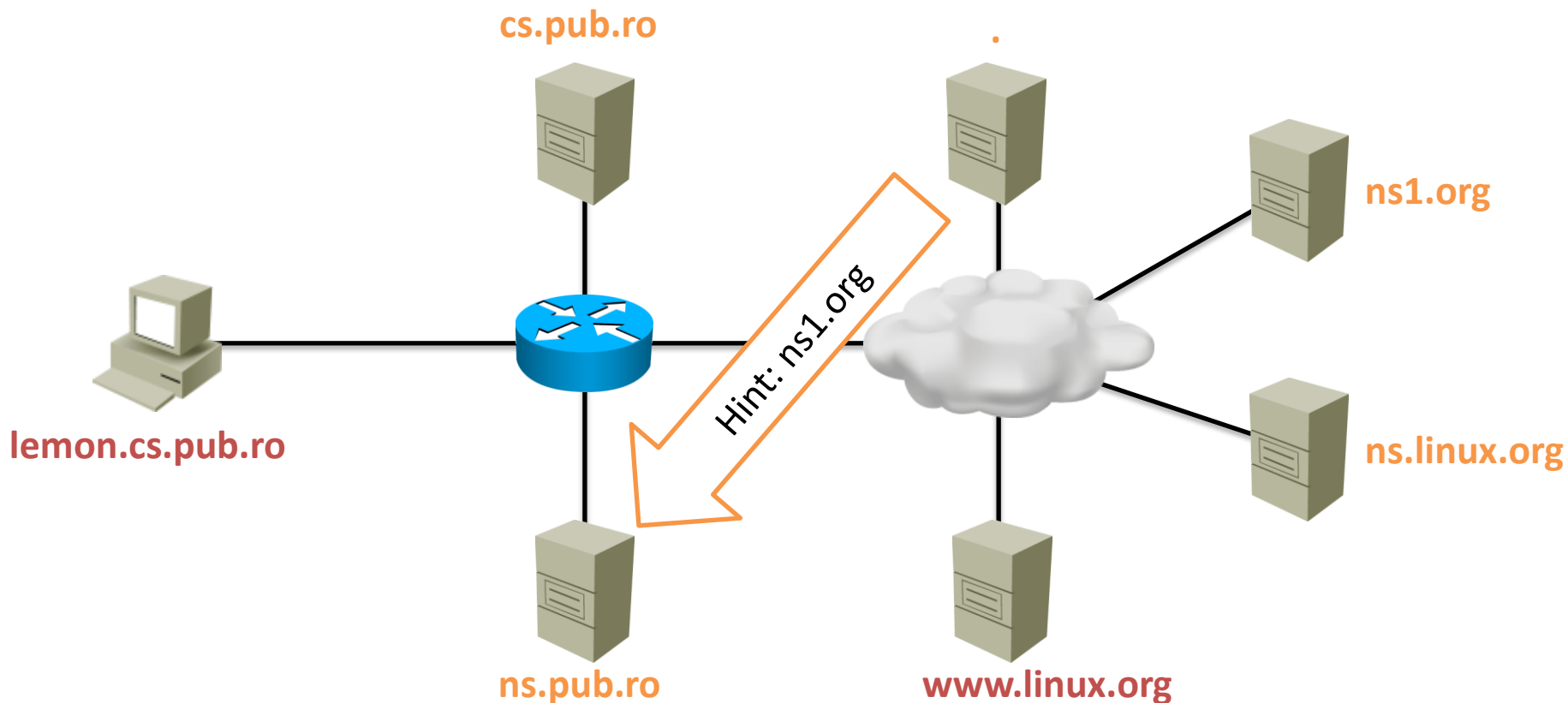
## Exemplu – Pasul 3

- **ns.pub.ro** verifică dacă are în cache **www.linux.org** → nu
- **ns.pub.ro** trimite o cerere nerecursivă către **.** (**root**)



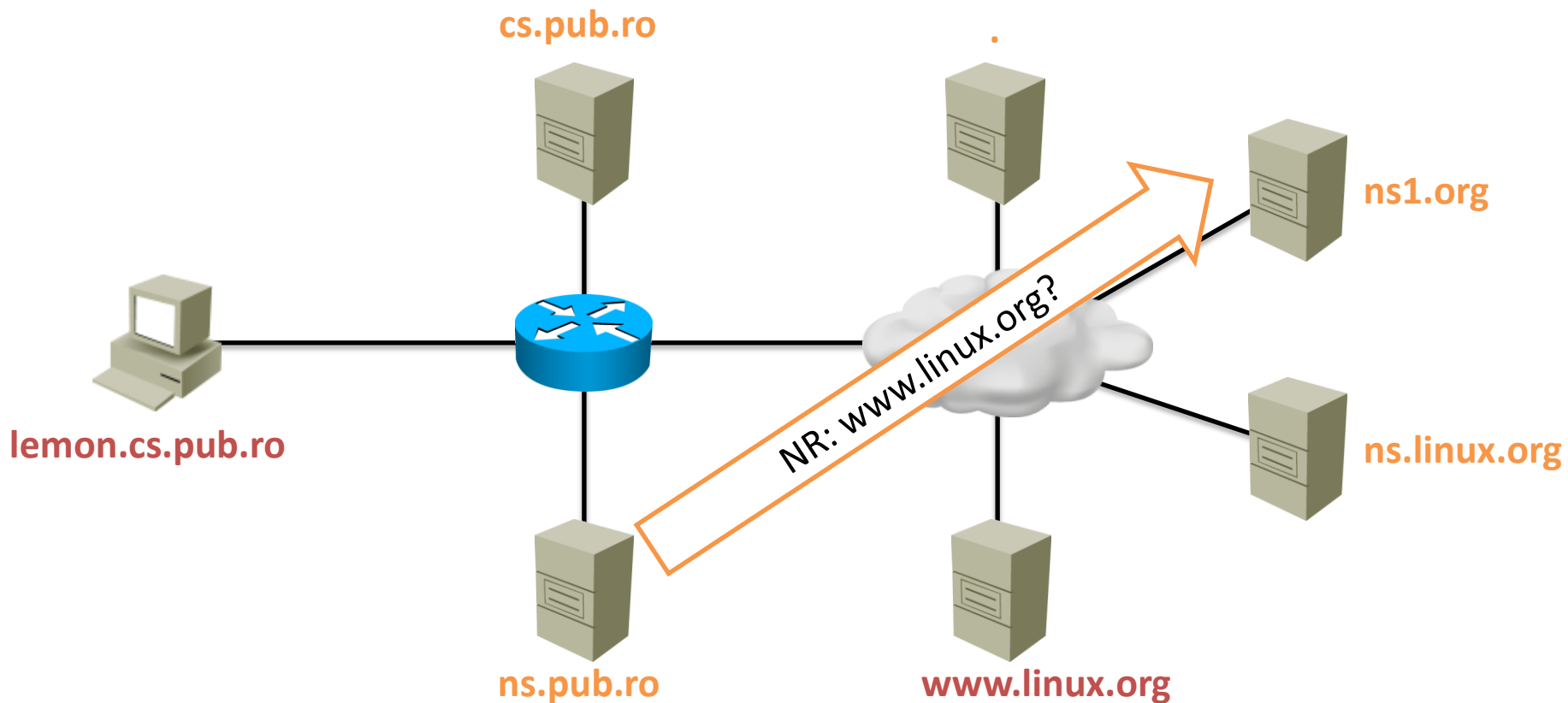
## Exemplu – Pasul 4

- **.(root)** verifică dacă are în cache **www.linux.org** → nu
- **.(root)** răspunde negativ cu hint-ul **ns1.org**



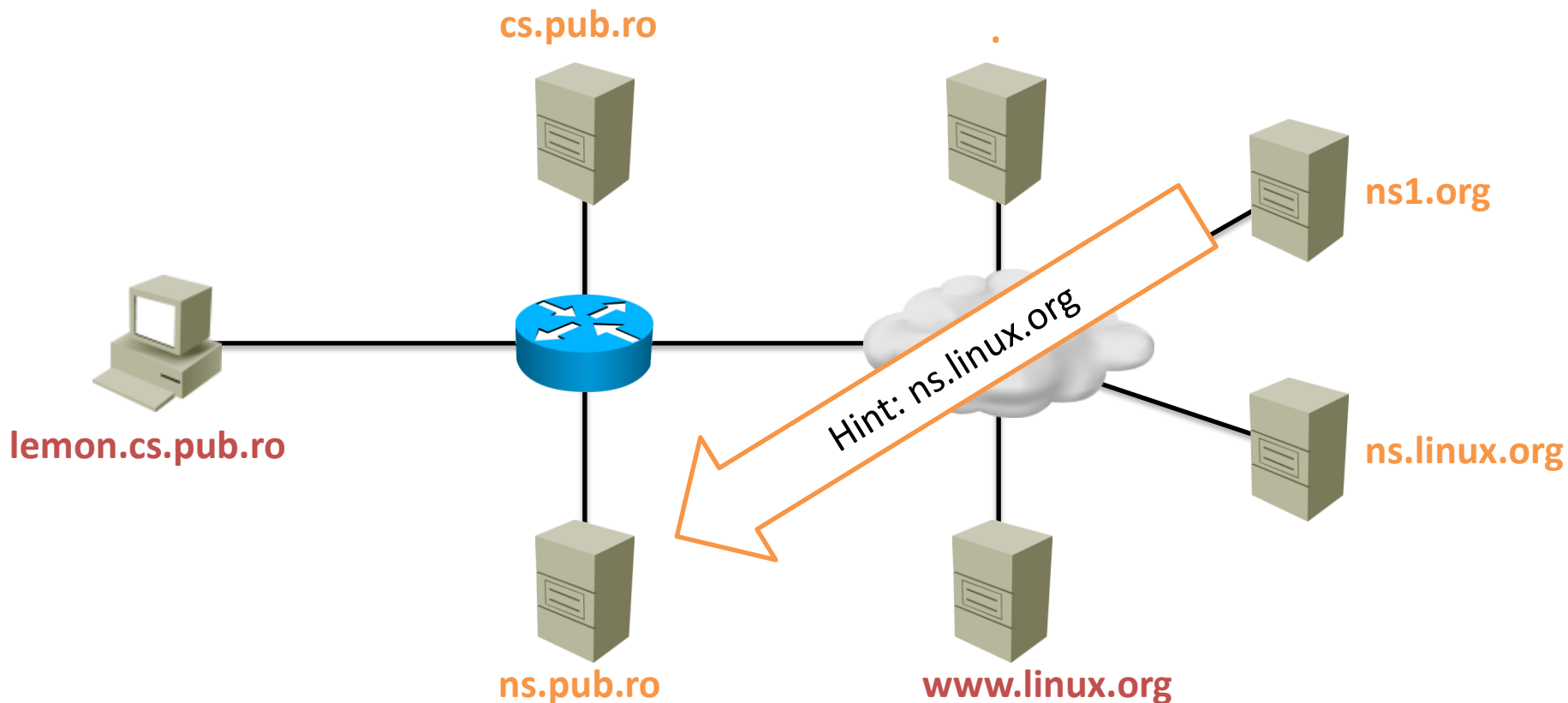


- **ns.pub.ro** trimite o cerere nerecursivă pentru **www.linux.org** către **ns1.org**

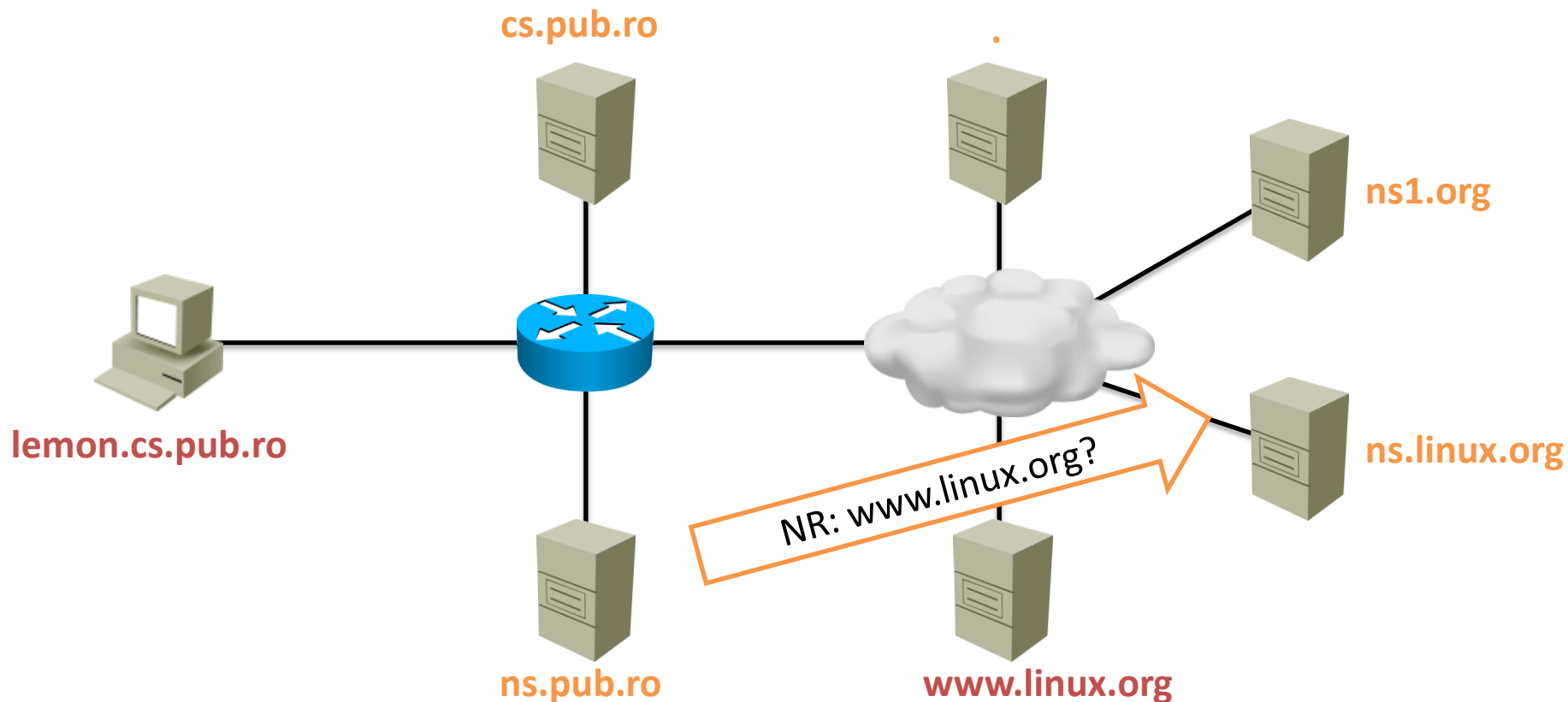


## Exemplu – Pasul 6

- **ns1.org** verifică dacă are în cache **www.linux.org** → nu
- **ns1.org** răspunde negativ cu hint-ul **ns.linux.org**

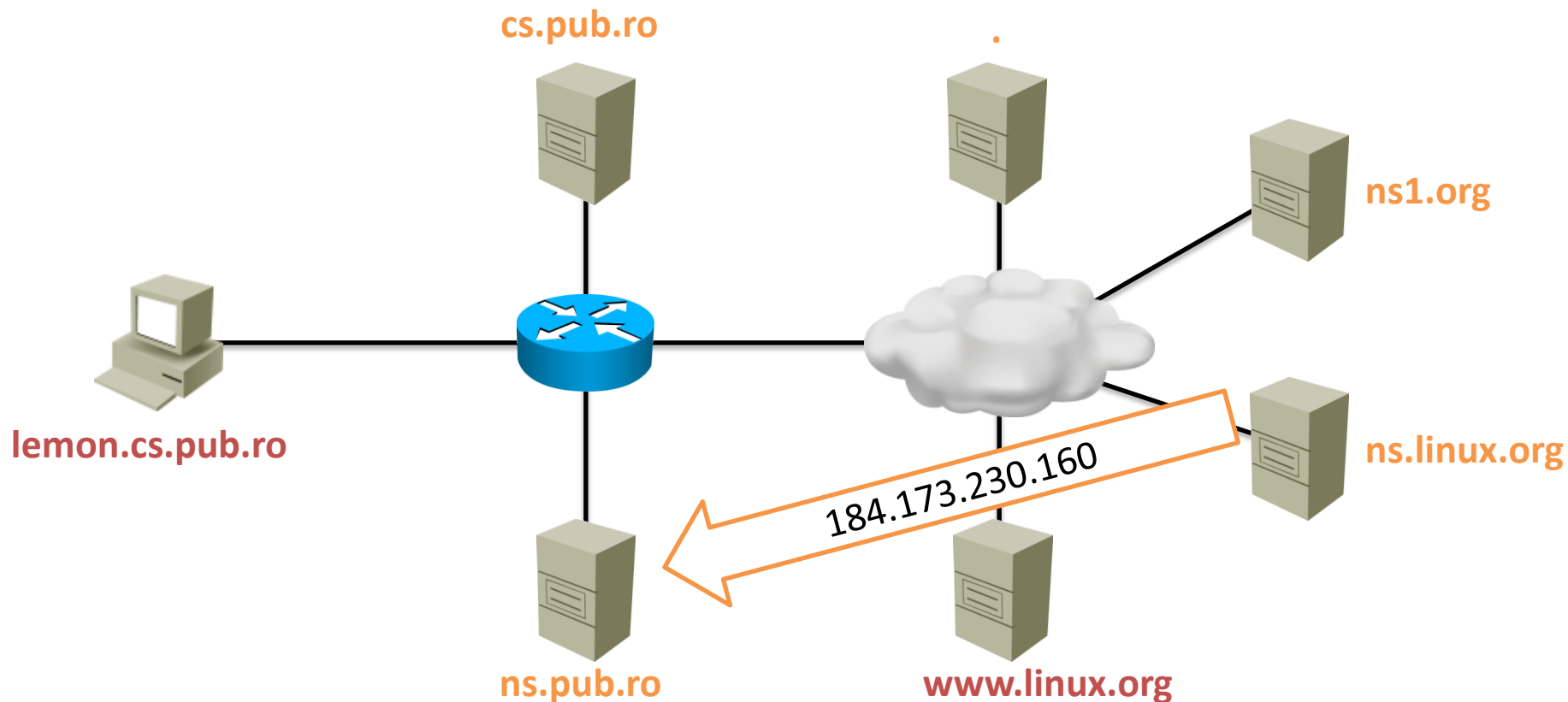


- **ns.pub.ro** trimite o cerere nerecursivă pentru **www.linux.org** către **ns.linux.org**

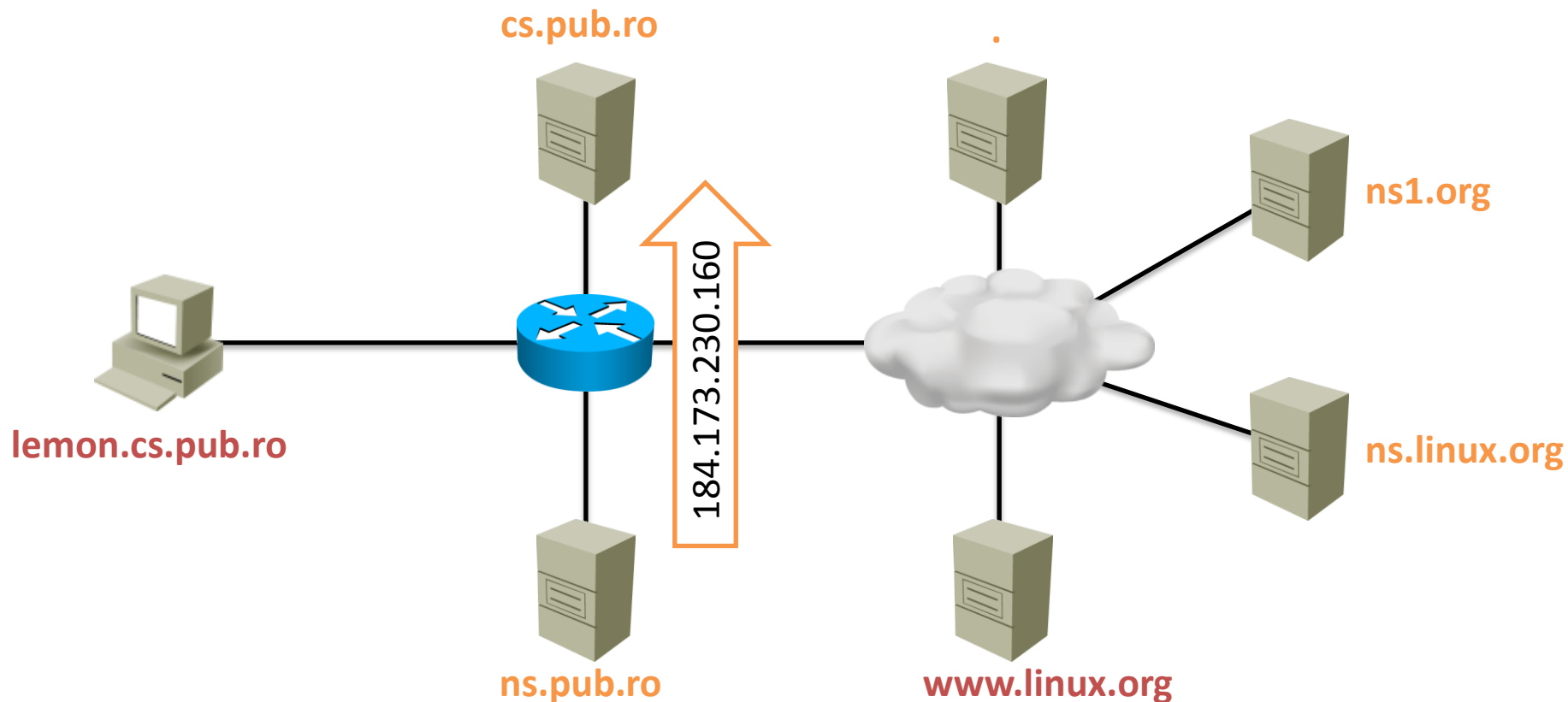


## Exemplu – Pasul 8

- **ns.linux.org** observă că este autoritar peste domeniul din cerere
- **ns.linux.org** răspunde pozitiv cu adresa IP solicitată



- **ns.pub.ro** adaugă înregistrarea în cache
- **ns.pub.ro** trimite răspunsul mai departe către **cs.pub.ro**



- **cs.pub.ro** adaugă înregistrarea în cache
- **cs.pub.ro** trimite răspunsul mai departe către **lemon.cs.pub.ro**

