

2015 - 2016

Programarea Clientului Web

s.l. dr. ing. Simona Caraiman

mailto: sarustei@cs.tuiasi.ro

**Universitatea Tehnica “Gheorghe Asachi” din Iasi
Facultatea de Automatica si Calculatoare**

ORGANIZARE

[https://moodle.cti.ace.tuiasi.ro/course/
view.php?id=352](https://moodle.cti.ace.tuiasi.ro/course/view.php?id=352)

- cursuri, laboratoare, alte resurse ...

ORGANIZARE

Structura disciplinei

PCW – curs (2h) + lab (1h) – **Colocviu** – 4k

PCW – proiect (1h) – **Colocviu** – 1k

ORGANIZARE

Titulari

Curs:

s.l. dr. ing. Simona Caraiman

Aplicatii:

ing. Tiberiu Dumitriu

ORGANIZARE

Evaluare

PCW

- colocviu – 50 % (nota min. 5)
- test curs – 25 %
- laborator – 25 % (nota min. 5)

PCW – proiect

- colocviu – 70 % (nota min. 5)
- activitate proiect – 30 % (nota min. 5)

ORGANIZARE

Calendar

- **PCW Curs**
 - **S8** (19.11.2015) – **test grila!!**
 - **S14 – colocviu**

- **PCW Proiect**
 - alege teze proiect: **S3**
 - **predare proiect: L13-L14**

Programarea (clientului) Web

Programarea **clientului** Web

- **World Wide WEB**
 - Internet vs. Web
 - definitie, caracterizare
 - arhitectura
- **CLIENT**
 - aplicatii client-server
 - tipuri de aplicatii client
 - client Web
- **Programarea aplicatiilor Web**
 - context: Web 2.0
 - Rich Internet Applications (RIA)

Care este diferența dintre...

1. Internet & Web

2. Website & Web App

Ce este World Wide Web?

“Panza de paianjen mondială”

Sistem global de distributie a informatiilor hipermedia*

Sistem informational pe **Internet** care permite conectarea documentelor

**hipermedia = multimedia + hiperlegaturi*

Ce este World Wide Web?

“Panza de paianjen mondială”

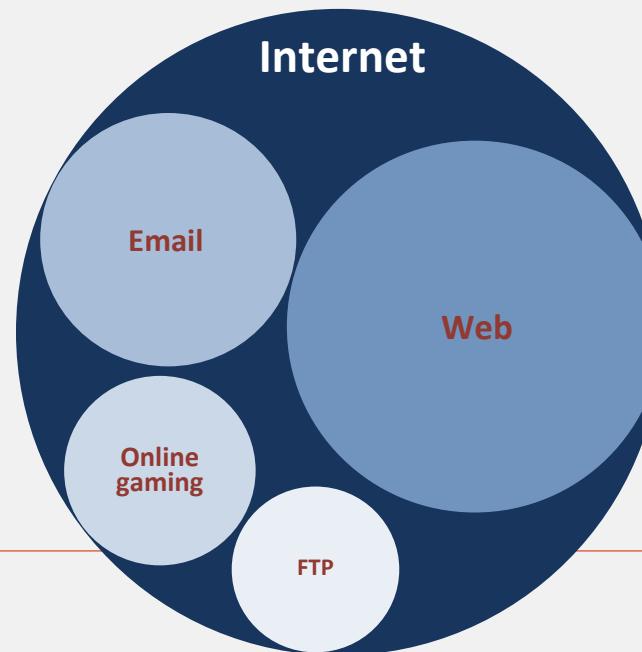
Sistem global de distribuție a informațiilor hipermedia*

Serviciu Internet, deci **Web ≠ Internet**

**hipermedia = multimedia + hiperlegături*

Ce este Internetul?

sistem global de retele de calculatoare interconectate



Internet vs WWW

Internet – servicii:

- **World Wide Web**
- **Communication**
 - e-mail
 - VoIP, Mobile VoIP
 - IPTV
 - Internet fax
 - Instant messaging
- **Data transfer**
 - file sharing, file transfer (FTP)
 - streaming media

World Wide Web

Sir Timothy John “Tim” Berners-Lee (CERN)



A inventat WWW si a scris primul web browser si primul web server (1990)
[http://www.w3.org/History/
1989/proposal.html](http://www.w3.org/History/1989/proposal.html)

Bazat pe modelul ***client-server*** si pe ***hipertext***

WWW – arhitectura*

- **resursele** (*documente sau pagini Web*) contin **marcaje** (adnotari)
- identificarea resurselor prin **adresa** (*URI – Uniform Resource Identifier*)
- protocol de acces la continutul resurselor: **HTTP** - HiperText Transfer Protocol
- program software (**web server**) care raspunde la cereri HTTP
- limbaj de publicare a resurselor (**HTML**)
- program software (**web browser**) care realizeaza cereri HTTP si afiseaza codul HTML(CSS/JS) primit

*<http://www.w3.org/TR/webarch/> PCW - C1.Introducere

World Wide Web

□ Reglementari:

- Consorțiul Web (W3C) www.w3.org – *HTML, XHTML, CSS, DOM*
- Internet Engineering Task Force (IETF) – *URI, HTTP*
- ECMA International – *ECMAScript (JavaScript)*

Ce este un Website?

un grup de pagini conectate pe **World Wide Web**

Ce este o Aplicatie Web?

- website-urile sunt definite de **continut si informatie**
- aplicatiile web sunt definite de **interactiunea cu utilizatorul**, in mod uzual *action-driven*

Web App vs. Desktop App

Avantaje ale aplicatilor web :

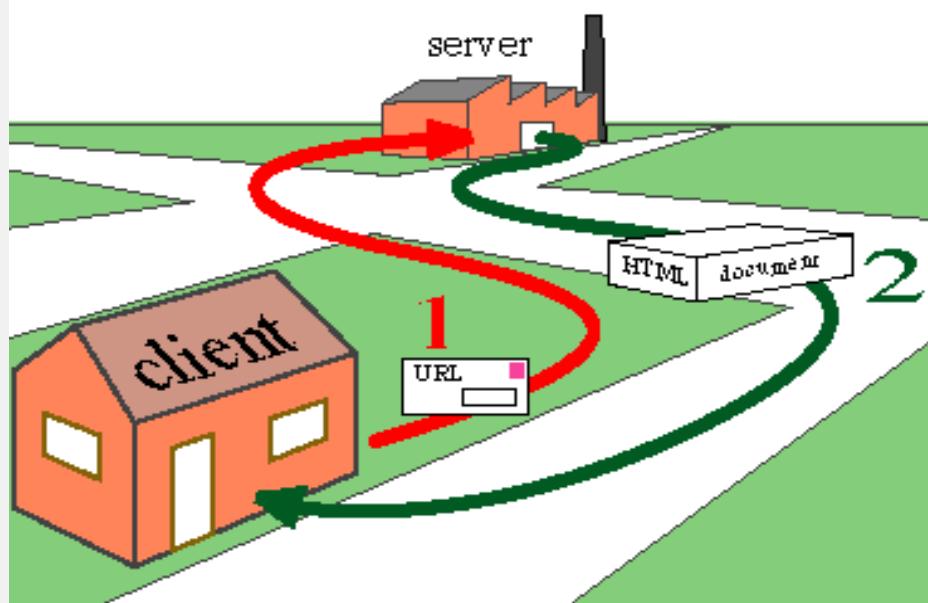
- Accesibile de la orice computer conectat la internet
- Utilizabile sub sisteme de operare si browsere diferite
- Usor de updatat (este necesara doar updatarea software-ului pe server nu si pe fiecare statie dintr-o organizatie)
- Stocarea centralizata pe server (important in cazul informatiilor sensibile)

Web App vs. Desktop App

Dezavantaje ale aplicatiilor web :

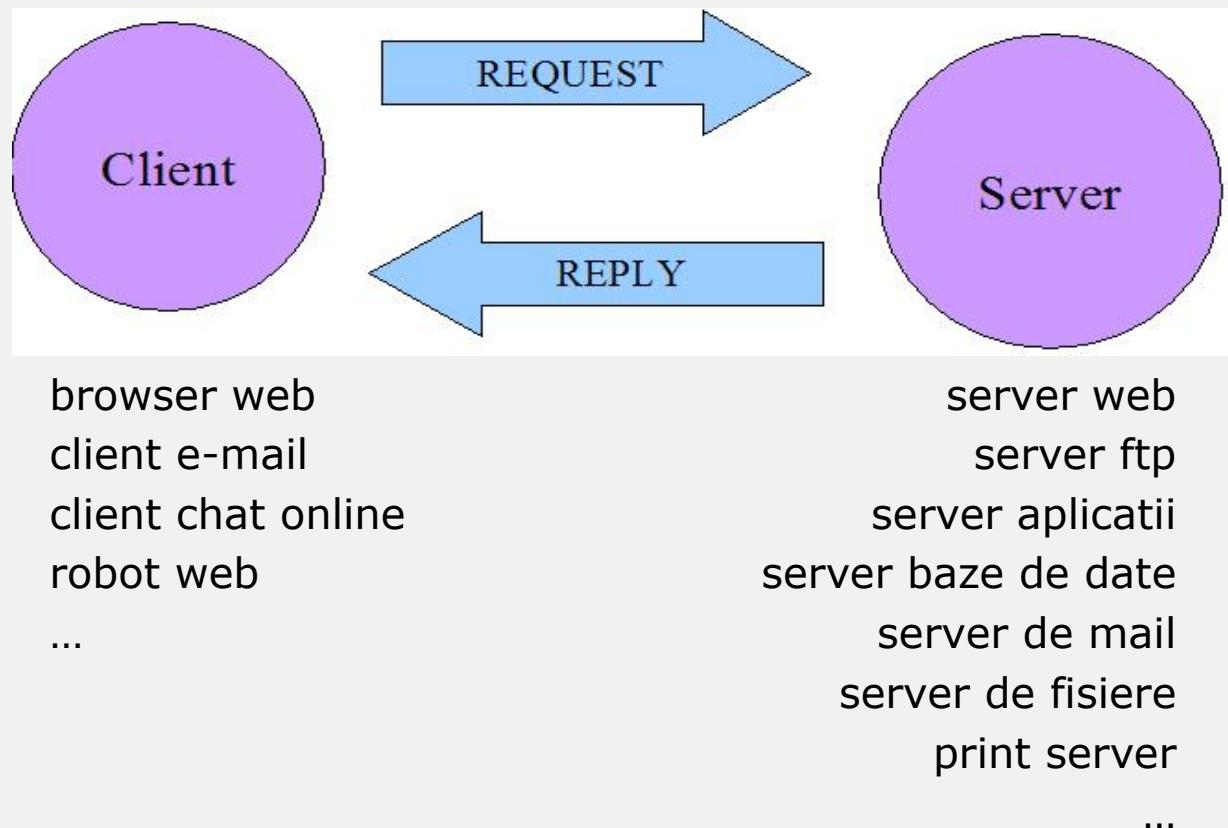
- Necesitatea unei conexiuni active la Internet (nu intotdeauna disponibila peste tot)
- Securitatea datelor sensibile vehiculate pe internet
- Aspectul neuniform al resurselor in browser-e diferite
- Accesul limitat la SO

WWW – arhitectura client-server



<http://public.web.cern.ch/public/en/About/WebWork-en.html>

WWW – arhitectura client-server



Aplicatii client

- **fat/rich/thick client**
 - procesarea are loc local
 - necesita, totusi, conexiuni periodice la server
 - **av:** servere mai putin performante (ieftine), lucru offline, performante multimedia mai bune, flexibilitate.
 - **ex:** video gaming
- **thin client**
 - procesarea are loc pe server
 - **av:** costuri reduse, utilizare usoara, securitate mai buna
 - **ex:** remote desktop apps
- **hybrid client**
 - procesare locala, stocare date pe server
 - suport multimedia, performante mari
 - flexibilitate, utilizare usoara

	Local storage	Local processing
Fat Client	Yes	Yes
Hybrid Client	No	Yes
Thin Client	No	No

Client Web

Program ce permite accesarea resurselor in
World Wide Web

Deci,

- identifica resursele prin URI-uri
- acceseaza resurse printr-un protocol web (“vorbeste” HTTP)
- prezinta documentele web utilizatorului (“cunoaste” limbaje de marcare si de interpretare a scenariilor)

Server Web

Single machine

vs.

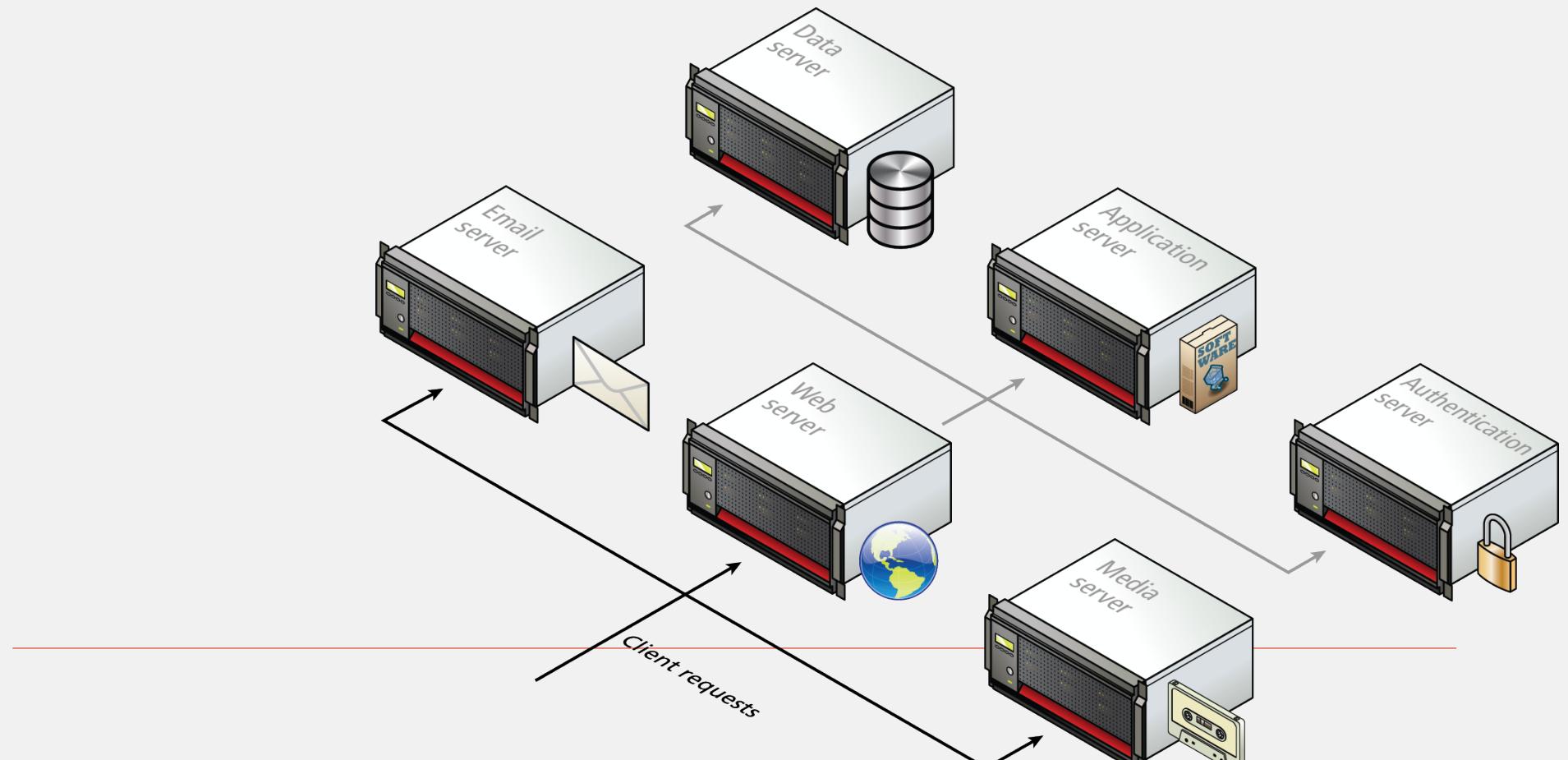
Real-world:

Ex., Facebook: milioane de cereri/sec

- tipuri diferite de servere
 - servere replicate
-

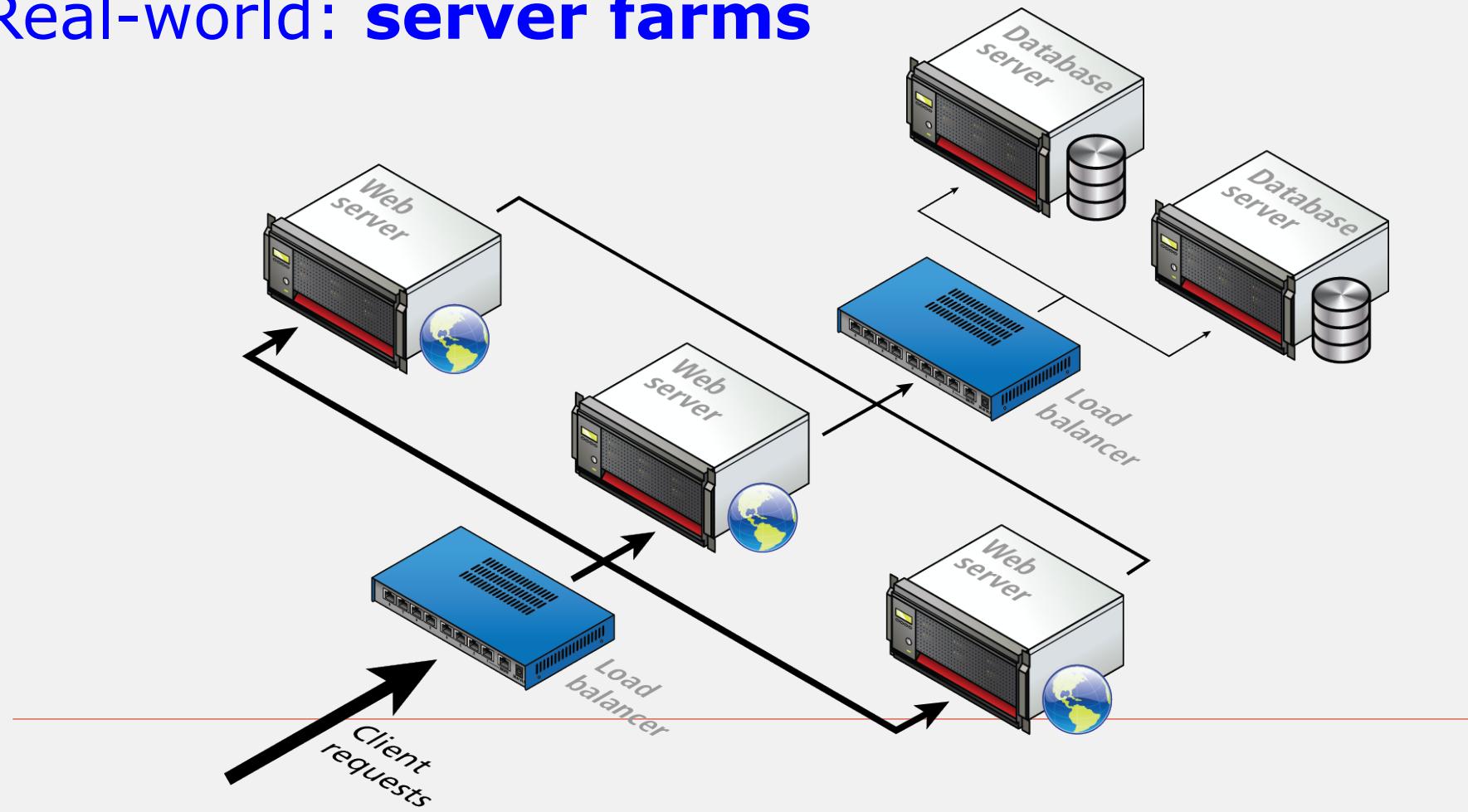
Server Web

Server with split functionality



Server Web

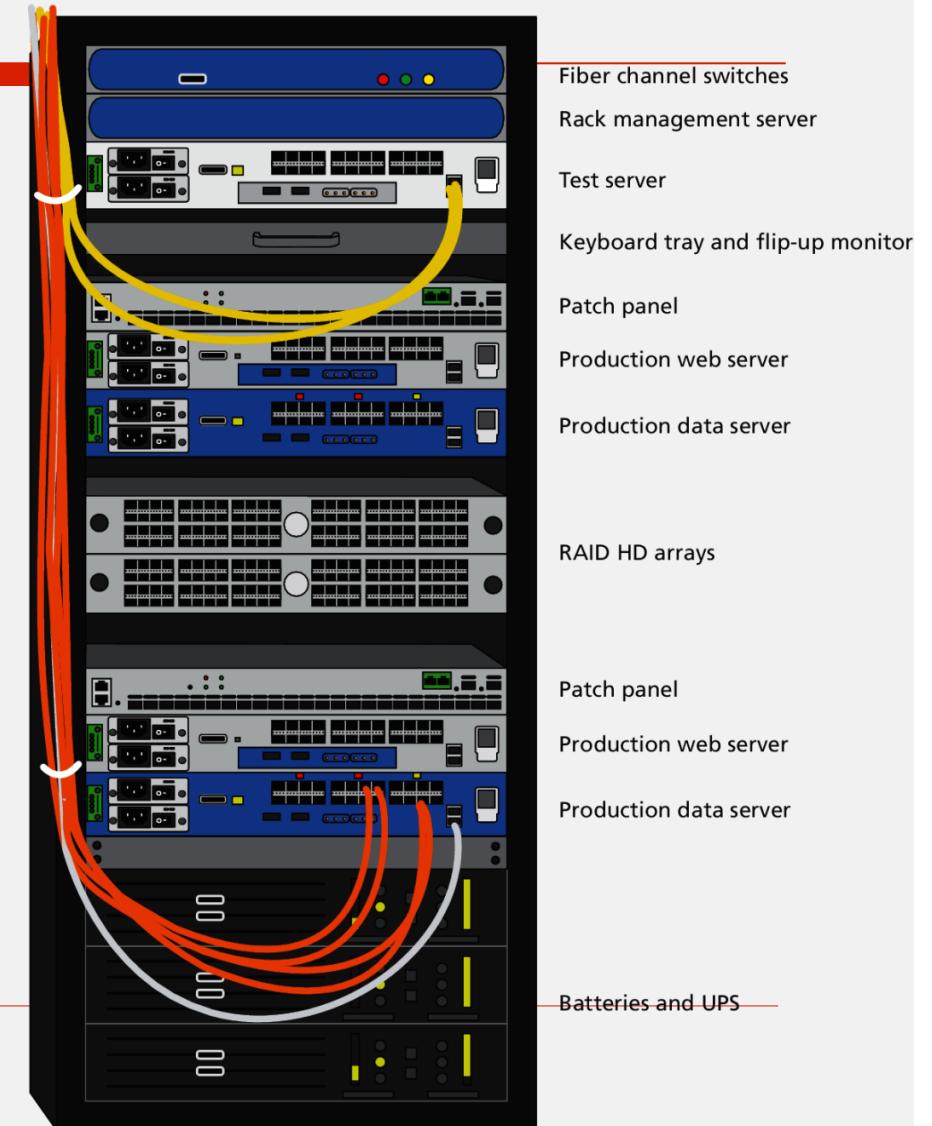
Real-world: **server farms**



Server Web

Real-world: **server farms**

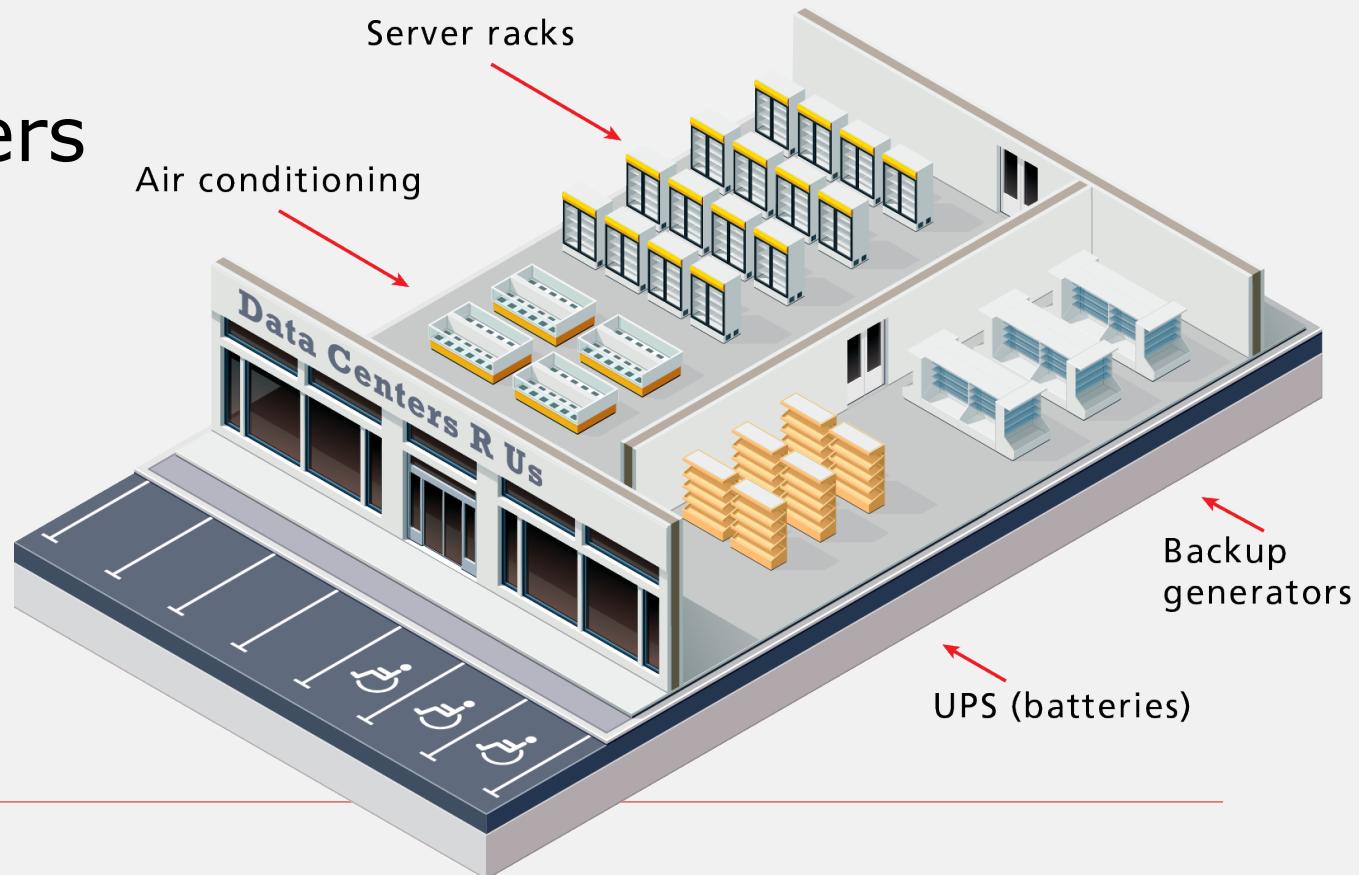
- server racks



Server Web

Real-world: **server farms**

- in data centers



Programarea (clientului) web

CONTEXT:

WEB 2.0*

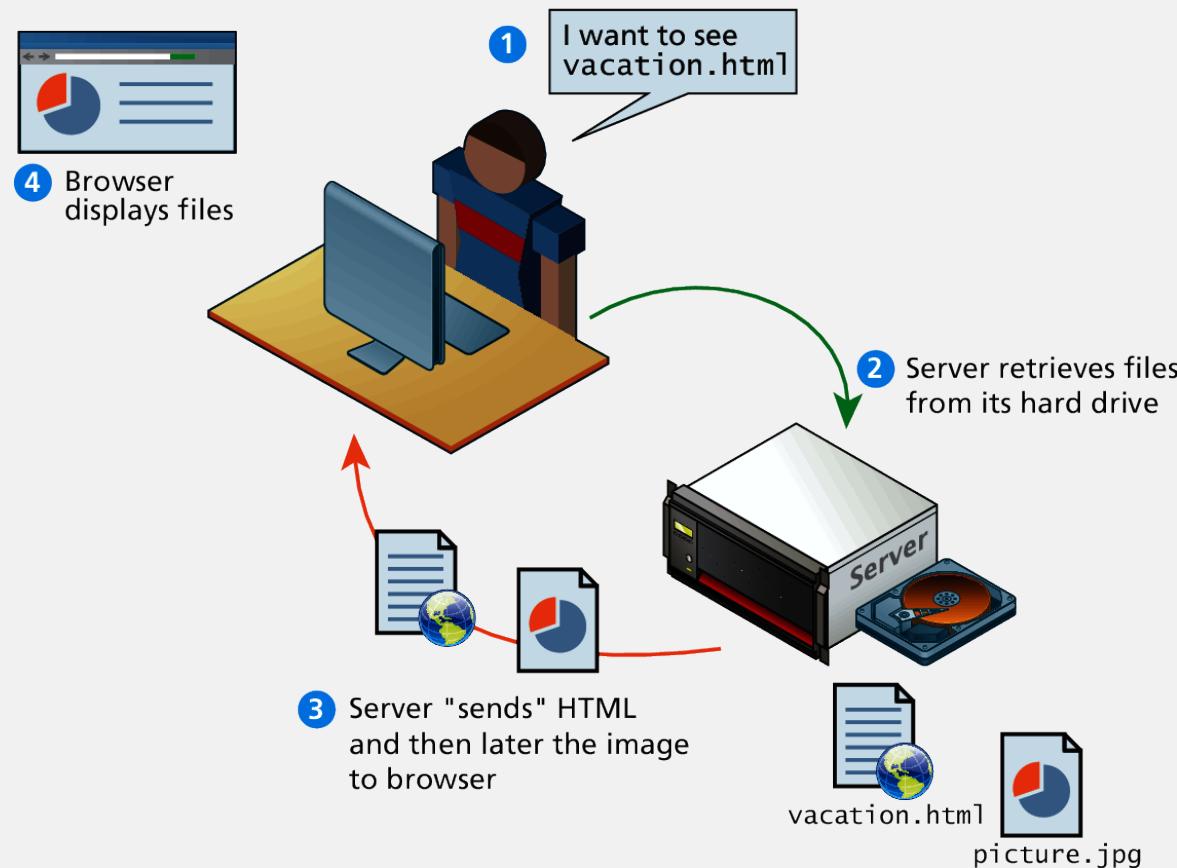
WEB-ul prezent = WEB Social



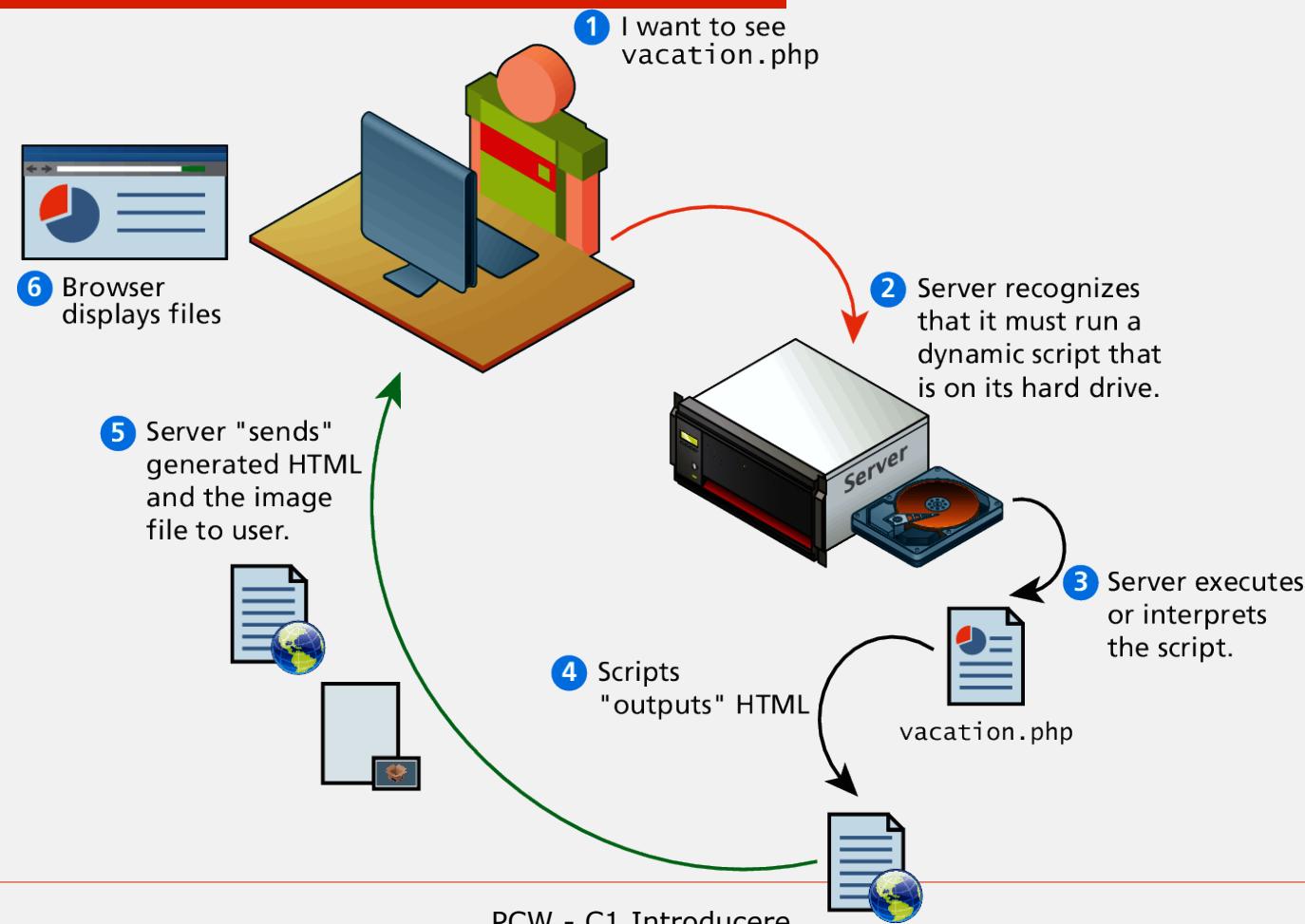
WEB Semantic

*<http://conferences.oreillynet.com/web2con/>,
Tim O'Reilly, "What Is Web 2.0", O'Reilly Media, (2005)

Web 1.0 (primii ani ai Web-ului)



WEB 2.0 – (continut generat dinamic)



WEB 2.0 – (continut generat dinamic)

□ Pentru USER:

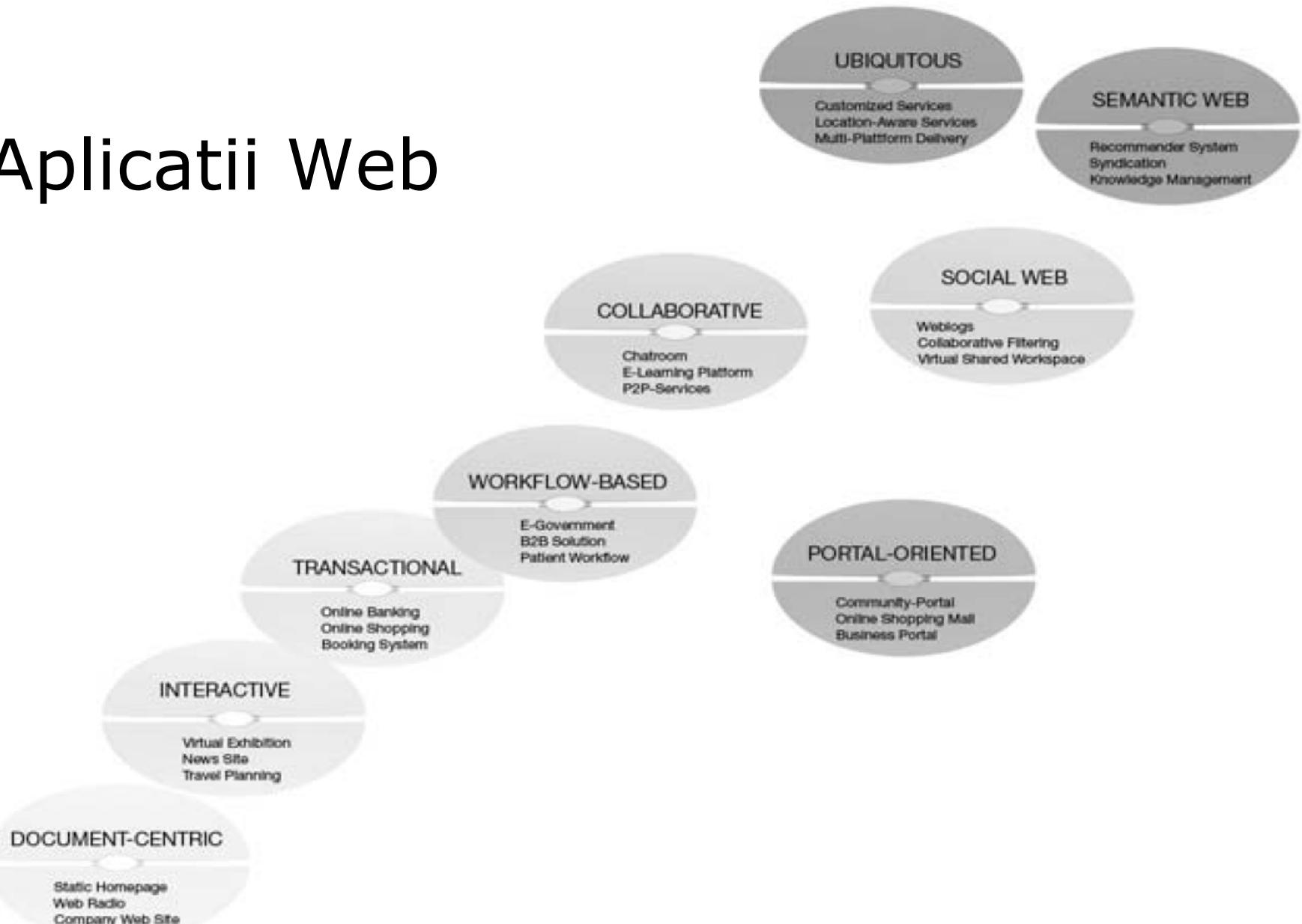
- Experienta interactiva cu site-ul web
- Contribuie la continutul web si consuma continut web
- Experienta web user-driven

□ Pentru DEVELOPER:

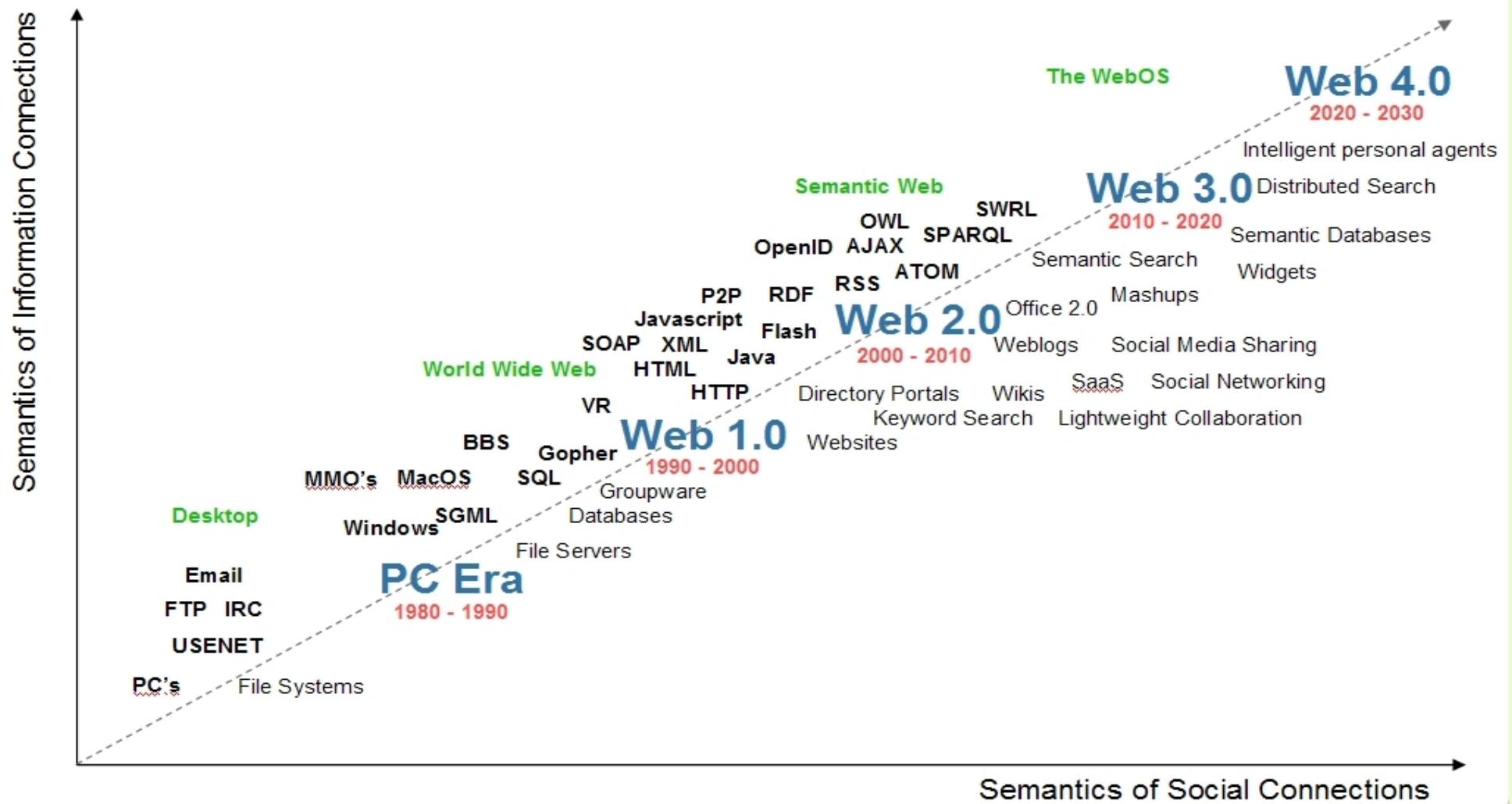
- Schimbare de paradigma in ceea ce priveste crearea site-urilor web dinamice
- Logica de programare, care exista inainte doare pe server, a inceput sa migreze in browser
- JavaScript, tehnici de comunicatii asincrone, limbaje de marcare

COMPLEXITY ↑

Aplicatii Web



DEVELOPMENT HISTORY →



Source: Radar Networks & Nova Spivack, 2007 – www.radarnetworks.com

WEB Semantic

- **Web of data**
 - "... that can be processed directly and indirectly by machines" (T. Berners-Lee)
- componenta a *Web 3.0*
- sisteme de recomandare, mediatizare, management de cunoștințe
- **formate comune** pt. **integrarea si combinarea datelor** provenind din surse diverse
 - RDF - Resource Description Framework
 - OWL - Web Ontology Language
 - XML - Extensible Markup Language

WEB Semantic

□ Exemplu – *DBpedia*

- publicarea datelor *structurate* extrase din Wikipedia
- complex queries (Faceted Wikipedia Search):
 - Rivers that flow into the Rhine and are longer than 50 kilometers
 - French scientists who were born in the 19th century

Programarea (clientului) Web

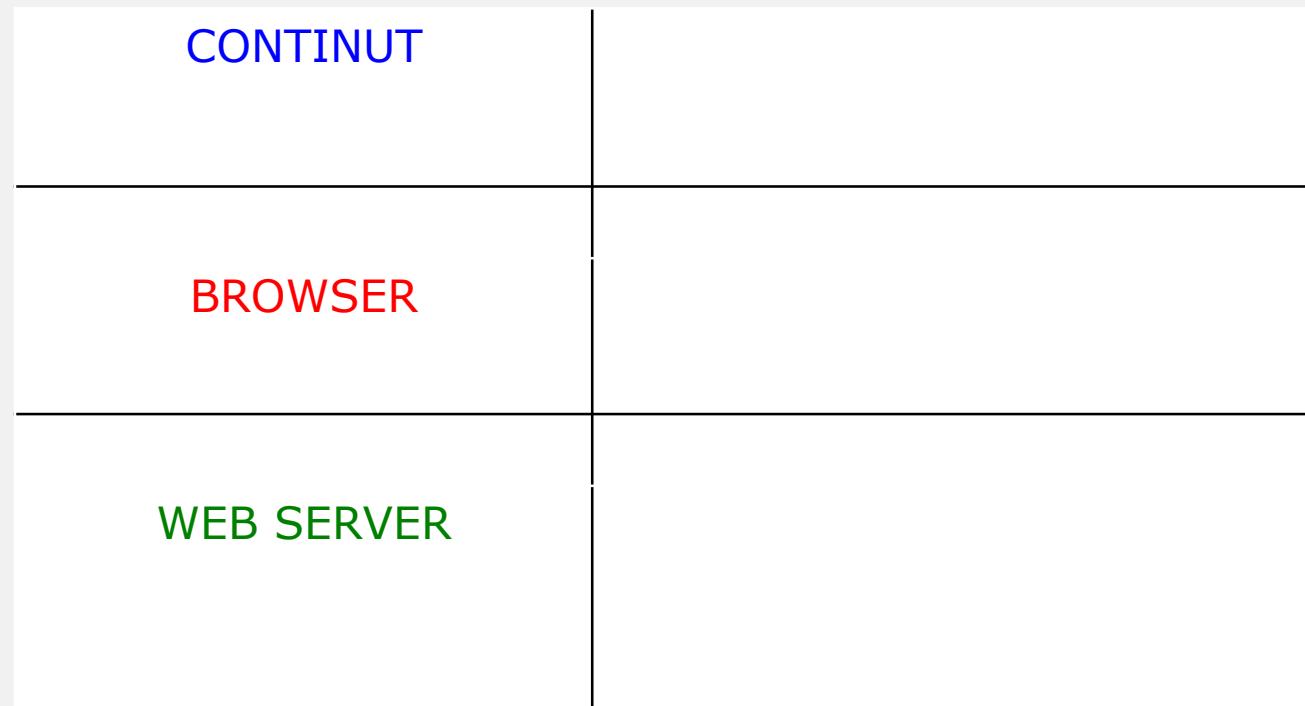
- Cum sunt site-urile web structurate?
- Cum “vorbesc” cu un site web?
- Cum construiesc un site web?

Programarea (clientului) Web

- Cum sunt site-urile web structurate?
- Cum “vorbesc” cu un site web?
- Cum construiesc un site web?

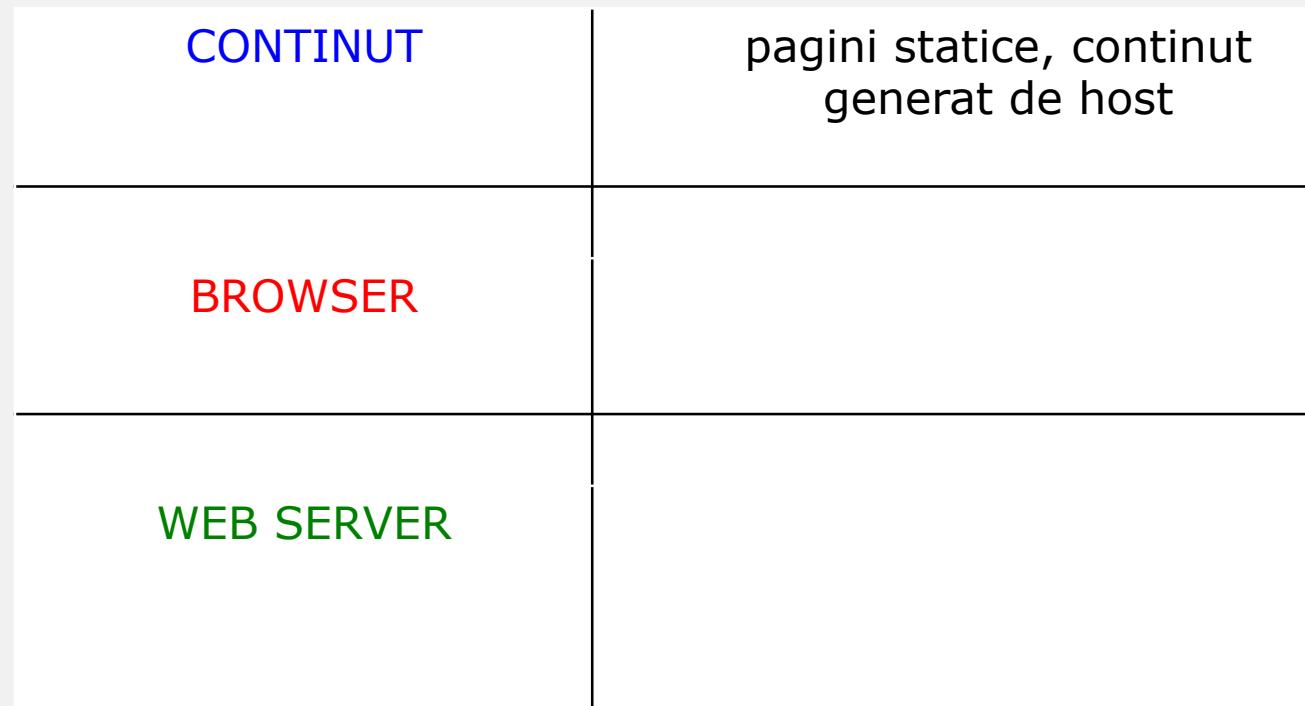
Structura unui site Web

Continut static



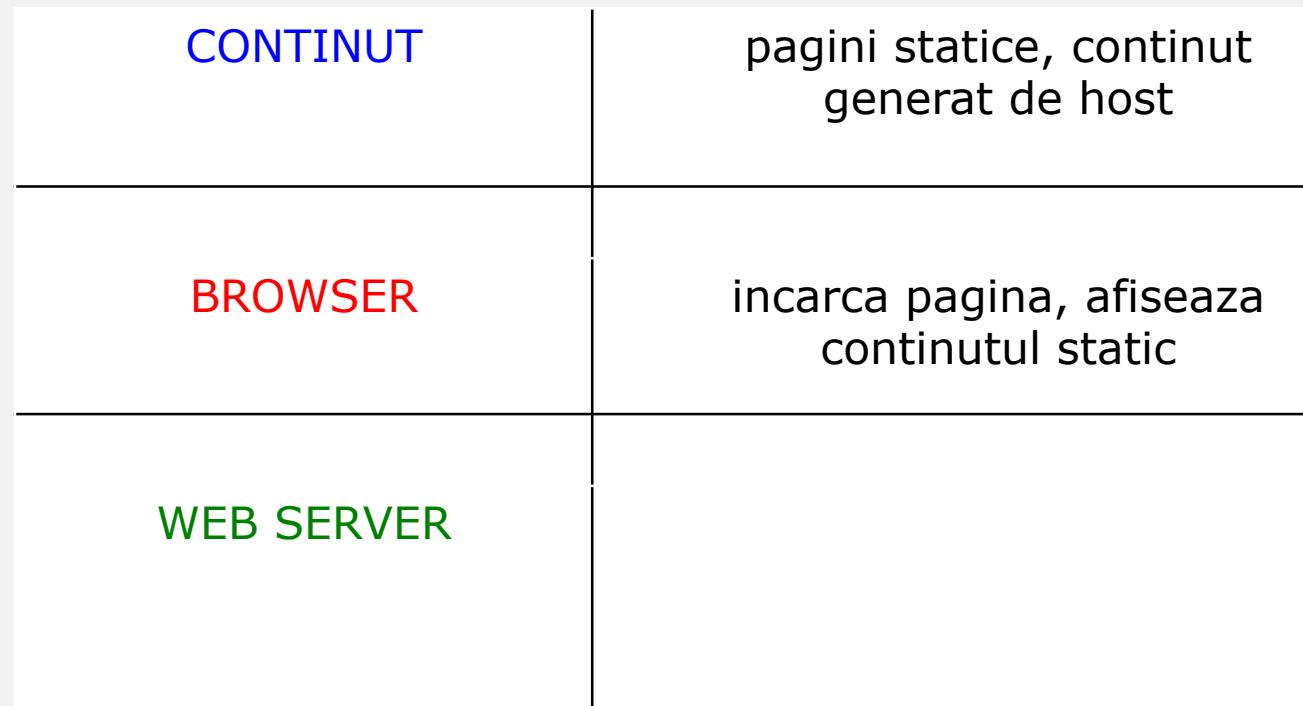
Structura unui site Web

Continut static



Structura unui site Web

Continut static



Structura unui site Web

Continut static

CONTINUT	pagini statice, continut generat de host
BROWSER	incarca pagina, afiseaza continutul static
WEB SERVER	model simplu cerere/raspuns

Structura unui site Web

Ce este un server?

loop indefinitely:

- wait for incoming connection
- accept incoming connection
- handle incoming connection
- send response back

Structura unui site Web

Ce este un server?

loop indefinitely:

- wait for incoming connection
- accept incoming connection
- handle incoming connection
- send response back

Continut static... doar gaseste fisierul cerut si trimite inapoi continutul acestuia!

Structura unui site Web

Ce este un server?

loop indefinitely:

- wait for incoming connection
- accept incoming connection
- handle incoming connection
- send response back

Continut static... doar gaseste fisierul cerut si trimitе inapoi continutul acestuia!

Serverul nu stie ce contine fisierul. Clientul interpreteaza raspunsul

Structura unui site Web

Web 2.0 – ce este un server?

The screenshot shows the Facebook sign-up interface. At the top, there's a blue header bar with the word "facebook" in white. To the right of the logo are fields for "Email or Phone" (containing "john.doe@gmail.com") and "Password" (containing "....."). There are also checkboxes for "Keep me logged in" and a link for "Forgot your password?". Below the header, a large white area features the text "Facebook helps you connect and share with the people in your life." followed by a map of the world with yellow user icons connected by dashed lines. To the right of this, the word "Sign Up" is prominently displayed in large, bold, black letters. Underneath it, the text "It's free and always will be." is shown. The main form consists of several input fields: "First Name" and "Last Name" (both empty), "Your Email" (empty), "Re-enter Email" (empty), "New Password" (empty), and "Birthday" (with dropdown menus for Month, Day, and Year). A question "Why do I need to provide my birthday?" is next to the birthday inputs. At the bottom, there are two radio buttons for gender: "Female" and "Male". A small note at the very bottom states: "By clicking Finish, you agree to our Terms and that you have FROM: FACEBOOK.COM".

Structura unui site Web

Web 2.0 – ce este un server?



Structura unui site Web

Web 2.0 – ce este un server?



English

Home Personal Business

Get to Know PayPal Pay Online Send Money Get Paid Products & Services

Account login

Email address

PayPal password

Go to

My account

Log In

Problem with login?

New to PayPal? [Sign up](#).

WELCOME TO PayPal

The world's most loved way to pay and get paid. [Learn More](#)

Get paid
anywhere.

The new way to
accept credit cards.

Learn More



Accept

+100 million
people use PayPal worldwide.

Sign Up

Pay Online

Send Money

Get Paid

[Get to Know PayPal](#)

Shop and pay online quickly

Send money to just about

Accept online payments fo

Find Highlight all Match case

Structura unui site Web

Ce este un server? – context: Web 2.0

- dynamic web sites, “web applications”
- user-generated content; personalization
- user interaction with content
- the web as a platform or service

Structura unui site Web

Web 2.0 – the server:

loop indefinitely:

- wait for incoming connection
- accept incoming connection
- handle incoming connection
- send response back

ruleaza **programul** specificat în cerere
returnează răspunsul **programului**

Răspuns: HTML, CSS, Javascript

Structura unui site Web

	early web	modern web-app
CONTINUT	pagini statice, continut generat de host	
BROWSER	incarca pagina, afiseaza continutul static	
WEB SERVER	model simplu cerere/ raspuns	

Structura unui site Web

	early web	modern web-app
CONTINUT	pagini statice, continut generat de host	pagini si continut generate dinamic pe server
BROWSER	incarca pagina, afiseaza continutul static	
WEB SERVER	model simplu cerere/ raspuns	

Structura unui site Web

	early web	modern web-app
CONTINUT	pagini statice, continut generat de host	pagini si continut generate dinamic pe server
BROWSER	incarca pagina, afiseaza continutul static	furnizeaza o platforma pt. user interaction
WEB SERVER	model simplu cerere/raspuns	

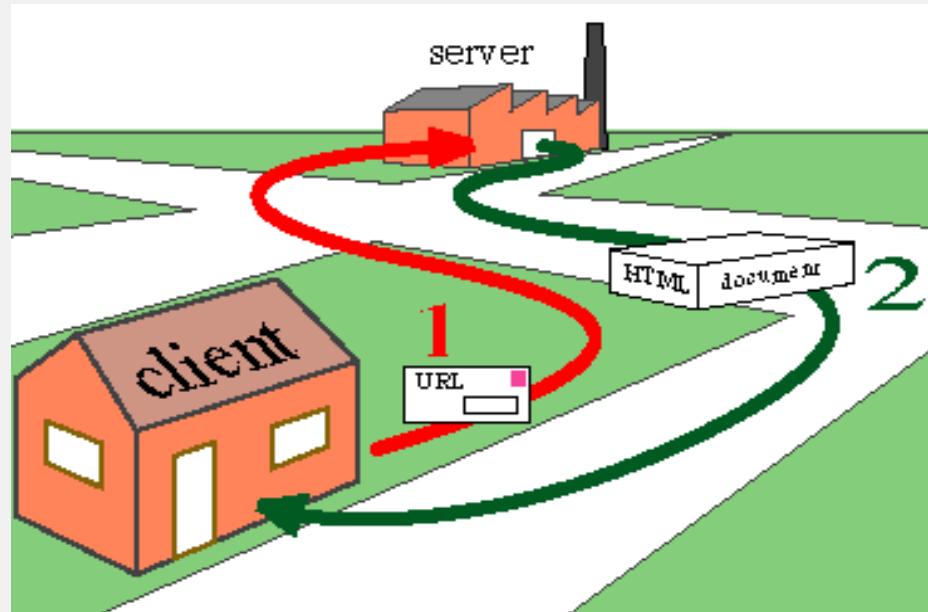
Structura unui site Web

	early web	modern web-app
CONTINUT	pagini statice, continut generat de host	pagini si continut generate dinamic pe server
BROWSER	incarca pagina, afiseaza continutul static	furnizeaza o platforma pt. user interaction
WEB SERVER "web stack"	model simplu cerere/raspuns	scripturi, framework-uri, baze de date

Programarea (clientului) Web

- Cum sunt site-urile web structurate?
- Cum “vorbesc” cu un site web?
- Cum construiesc un site web?

Comunicatia client-server



URI, DNS, HTTP

Adresarea resurselor Web

URI – *Uniform Resource Identifier*

(RFC 3986)

secventa compacta de caractere ce identifica o
resursa abstracta sau fizica

Ex.:

<ftp://ftp.is.co.za/rfc/rfc1808.txt>

<http://eureka.cs.tuiasi.ro/~sarustei/PCW/>

<mailto:John.Doe@example.com>

<tel:+1-816-555-1212>

<telnet://192.0.2.16:80/>

URI, URL si URN

URI – locator, nume, amandoua

URL – *Uniform Resource Locator*

- **localizeaza** resursa prin descrierea mecanismului de acces
- <http://users.cs.tuiasi.ro/~sarustei/PCW/>
- <ftp://ftp.is.co.za/rfc/rfc1808.txt>

URN – *Uniform Resource Name*

- **identifica** resursele prin nume, intr-o forma persistenta, chiar daca resursa a disparut (“urn:”<NID>”:<NSS>)
 - NID – Namespace ID, NSS – Namespace Specific String
 - registered NID: **ietf, pin, issn, oid, newsml, oasis, xmlorg, publicid**
- urn:isbn:0451450523
- urn:issn:0167-6423
- urn:ietf:rfc:2648

URL vs URN <=> “unde” vs “ce”

URN – Uniform Resource Name

- nume simbolice asignate resurselor de catre organizatii cu rol de autoritate (IANA)
- nu contin informatii despre locatia/locatiile fizice ale resursei
 - pot mapa url-uri multiple
 - decizia de a utiliza un anume url poate fi amanata pana in momentul accesului efectiv la resursa
 - daca un url nu este disponibil, poate fi utilizat un altul fara necesitatea interventiei utilizatorului
 - pot fi obtinute informatii despre resursa fara a fi necesara accesarea efectiva a acesteia
- Rezolvarea urn-urilor
 - NU este standardizata
 - poate fi implementata folosind HTTP, DNS
 - Implementari la nivel corporativ
 - Dependente de necesitatile organizatiei/corporatiei respective

URL – Uniform Resource Locator

□ Absolut

absolute-URL = scheme ":" hier-part ["?" query]["#" fragment]

- [http://example.org/absolute/URI/with/absolute/path/to/
resource.txt](http://example.org/absolute/URI/with/absolute/path/to/resource.txt)
- <ftp://example.org/resource.txt>

□ Relativ

- ../../resource.txt
- ./resource.txt#frag01
- #frag01

□ Transformare URL relativ in URL absolut: **RFC 3986**

URL – Uniform Resource Locator

Sintaxa generica

- se constă dintr-o secvență hierarhică de componente: *scheme*, *authority*, *path*, *query*, *fragment*.

URL = **scheme ":" hier-part ["?" query] ["#" fragment]**

foo://example.com:8042/over/there?name=ferret#nose
_ / _____ / ____ / __ / __ /
| | | | |
scheme authority path query fragment

URL - Sintaxa

Schema

- Consta dintr-o seventa de caractere care incepe cu o litera urmata de orice combinatie de litere, cifre, plus ("+"), punct ("."), sau minus ("-").
- case-insensitive
- officiale (IANA – *Internet Assigned Number Authority*):
 - [file](#), [ftp](#), [http](#), [https](#), [ldap](#), [mailto](#), [telnet](#), etc.
- neoficiale:
 - [fish](#), [skype](#), [callto](#), [view-source](#), [ymsgr](#), etc.

Ex:

ymsgr:sendIM?<screenname>

view-source:http://en.wikipedia.org/wiki/URI_scheme

http://eureka.cs.tuiasi.ro/~sarustei/PCW/

URL - Sintaxa

Authority

authority = [userinfo "@"] host [":" port]

- precedata de //
- se termina la primul caracter “/”, “?” sau “#” intalnit, sau la sfarsitul URL-ului

foo://example.com:8042/over/there?name=ferret#nose

URL - Sintaxa

Path

- contine date (organizate ierarhic), care, impreuna cu componenta *query*, servesc la identificarea resursei in contextul schemei si a autoritatii specificate
- se termina la primul caracter "?" sau "#" sau la sfarsitul URL
- daca un URL contine componenta *autoritate*, atunci componenta *path* trebuie sa fie precedata de "/"

`foo://example.com:8042/over/there?name=ferret#nose`

URL - Sintaxa

Query

- Contine date non-ierarhice, care impreuna cu componenta *path*, servesc la identificarea resursei in contextul schemei si autoritatii specificate
- este indicata prin intermediul “?” si se termina la primul caracter “#” sau la sfarsitul URL-ului

foo://example.com:8042/over/there?**name=ferret**#nose

URL - Sintaxa

Fragment

- permite identificarea indirecta a unei resurse secundare prin referinta la o resursa primara si informatii aditionale de identificare
- componenta fragment este indicata de prezenta caracterului “#” si se termina la sfarsitul URL
- evaluat client-side
 - cererea trimisa de client serverului nu contine identificatorul de fragment

`foo://example.com:8042/over/there?name=ferret#nose`

URI

- utilizeaza caractere dintr-un subset al setului ASCII:

ALPHA DIGIT - . _ ~

- caractere rezervate

- pt. delimitarea componentelor generice ale unui URI:

: / ? # [] @

- pt. delimitarea sub-componentelor:

! \$ & ` () * + , ; =

- pot fi codificate folosind %HH-escaping

- & - %26

- OBS: codarea caracterului *spatiu*

- %20 vs + (*in query strings*)

IRI

IRI = *Internationalized Resource Identifier*

(RFC 3987)

- complementar URI-ului
- sevența de caractere din Universal Character Set (Unicode/ ISO 10646)

	IRI (utf8/%HH)	URI (us-ascii/%HH)
March	March	March
März	März	M%C3%A4rz
http://wine.org/rosé	http://wine.org/rosé	http://wine.org/ro%E9

Curs 1 - overview

Internet vs. WEB

Website vs. Web app. vs. Desktop App

Programare (client) Web – Context

Programare (client) web – Basics

- structura website (static vs. dinamic)
- comunicatia client-server pe Web
 - adresarea resurselor ([URI](#))

Curs 2+ - preview

Programare (client) web – Basics (cont.)

- comunicatia client server pe Web
 - DNS, HTTP
- dezvoltarea unui site Web
 - responsabilitati client/server
 - web stacks – tehnologii client/server
 - page load steps

Curs 2+ - preview

Client side:

- arhitectura browser, Javascript, AJAX
- client-side performance
- plugins, extensii, add-ons
- Web Workers

Server side:

- web app architecture,
- web frameworks (Node.js)

Curs 2+ - preview

Formate de manipulare a datelor

- familia XML
- JSON

Persistenta

- cookies, sesiuni
- Web Storage

Web Sockets, WebRTC, Server-sent events

Web mobile

Web semantic

Securitate Web