

Programare Web



Ciprian Dobre
ciprian.dobre@cs.pub.ro

Curs



Mit... si realitate

- Oricine poate face programare Web
 - Reteta: programator (la nivelul documentelor HTML?) cu ceva cunostinte de ASP.NET si SQL
 - Programarea Web = (mult) mai mult de atat
- Aplicatii desktop vs. Aplicatii web?
 - Aplicatii precum Photoshop, Word, Excel... in JavaScript? ... Deja se intampla.
 - Business-ul are nevoie de Web. Piata de Web a intrecut de mult piata aplicatiilor (inca) Desktop.
 - Browserele devin mai puternice, limbajele, tehnologiile, platformele mai evaluate.
- Programarea Web = mediu challenging



Web-ul in zilele noastre...

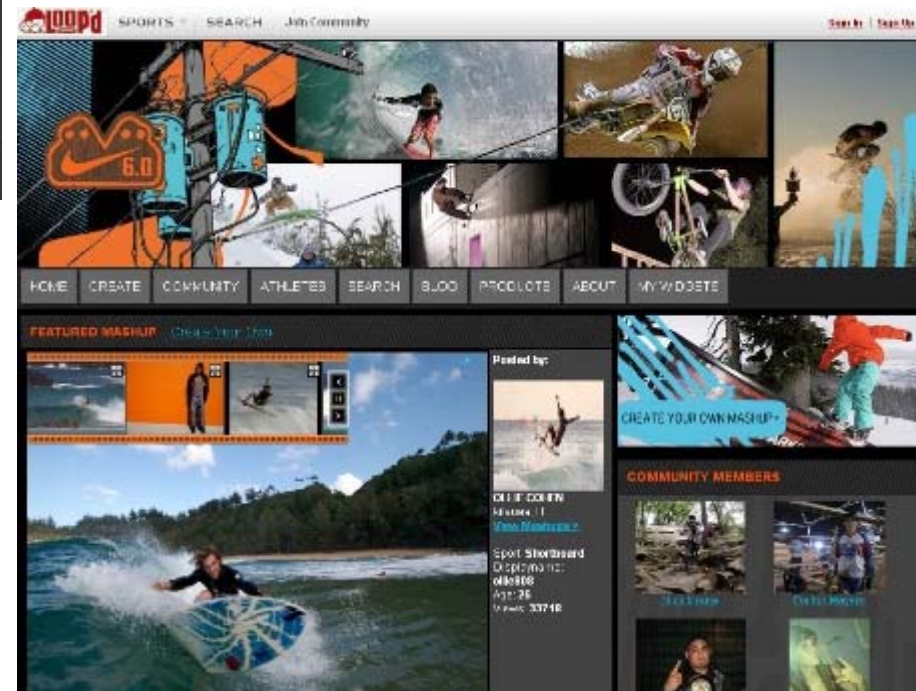


Exemple...



Dell furnizeaza o varietate de alternative pentru o mai buna colaborare sociala cu proprii clienti, inclusiv o insula in lumea virtuala din Second Life

Nike a creat o comunitate sociala pe Loopd pentru conectarea atletilor din diverse activitati sportive cu universul Nike



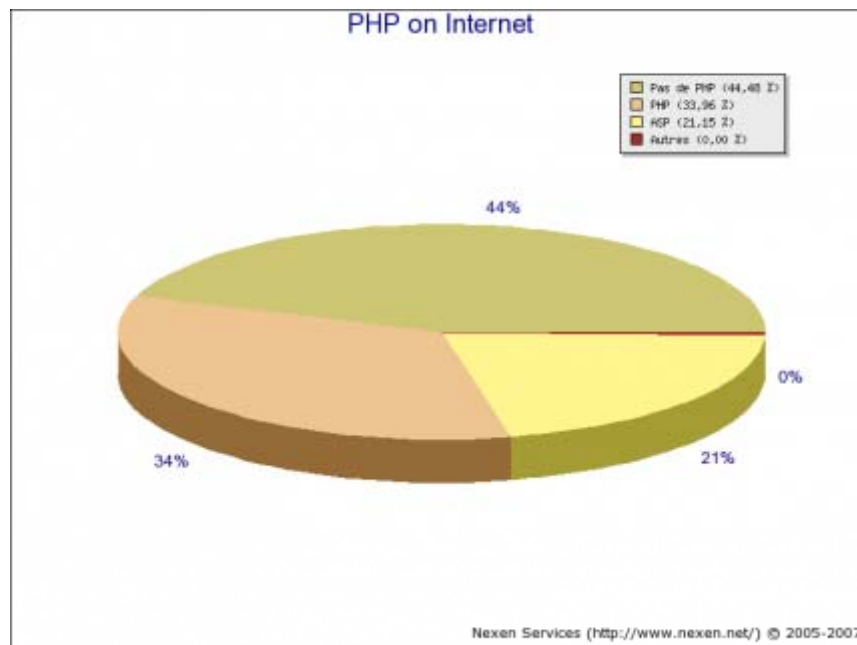


Despre Curs

- Site: <http://curs.cs.pub.ro>
- Competente:
 - Utilizarea instrumentelor IT pentru crearea de site-uri web
 - Cunoasterea si intelegerea de concepte, principii si teorii ale domeniului programarii web
 - Modelarea si proiectarea de sub-sisteme software luand cele mai bune decizii privind raportul costuri-rezultate in ceea ce priveste deciziile de proiectare
 - Asigurarea calitatii produselor si serviciilor web

Continut

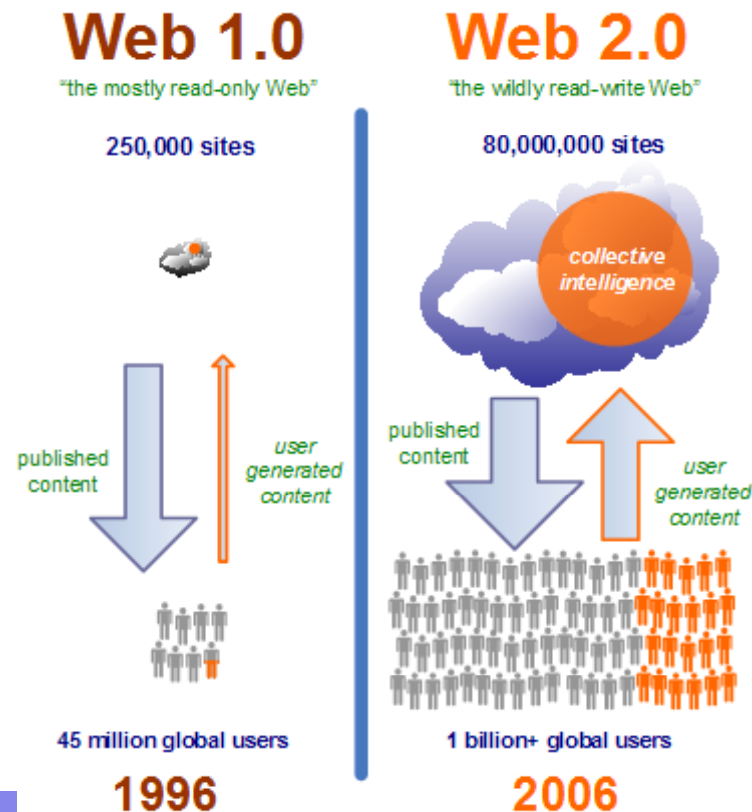
- Noțiuni, metodologii și tehnici de dezvoltare a aplicațiilor web-based
- Limbaje de dezvoltare web server-side (PHP)



```
    'role_id' => $role_details['id'],  
    'resource_id' => $resource_details['id'],  
    );  
if ( $this->rule_exists( $resource_details['id'], $role_details['id'] ) )  
    if ( $access == false ) {  
        // Remove the rule as there is currently no need for it  
        $details['access'] = !$access;  
        $this->_sql->delete( 'acl_rules', $details );  
    } else {  
        // Update the rule with the new access value  
        $this->_sql->update( 'acl_rules', array( 'access' => $access ) );  
    }  
foreach( $this->rules as $key=>$rule ) {  
    if ( $details['role_id'] == $rule['role_id'] && $details['resource_id'] == $rule['resource_id'] ) {  
        if ( $access == false ) {  
            unset( $this->rules[ $key ] );  
        } else {  
            $this->rules[ $key ]['access'] = $access;  
        }  
    }  
}
```

Continut

- Limbaje de dezvoltare web client-side (JavaScript, AJAX, DOM...)



Continut

- Elemente de stil în programarea web. CSS.
Structurarea paginilor Web. Principii de design web.

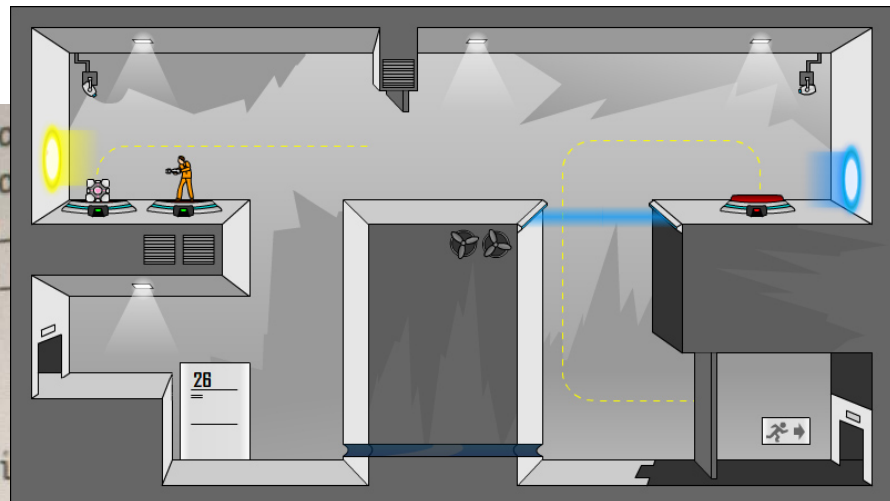


Vs.



Continut

- Elemente multimedia pentru programarea web. Flash și ActionScript.



```
er(Event.COMPLETE, o  
r(Event.COMPLETE, o  
r(IOErrorEvent.IO  
r(IOErrorEvent.IO  
  
uration*1000)/myTi  
Delay,myTimerStepsMax);  
Event.TIMER,onFadeProgress);  
Event.TIMER_COMPLETE,onFadeComplete);  
  
(imageDuration*1000)/myTimerStepsMax;  
r(durationDelay,myTimerStepsMax);  
stener(TimerEvent.TIMER_COMPLETE,onDurationComplete);  
roy, baseUrl:String):void
```

Continut

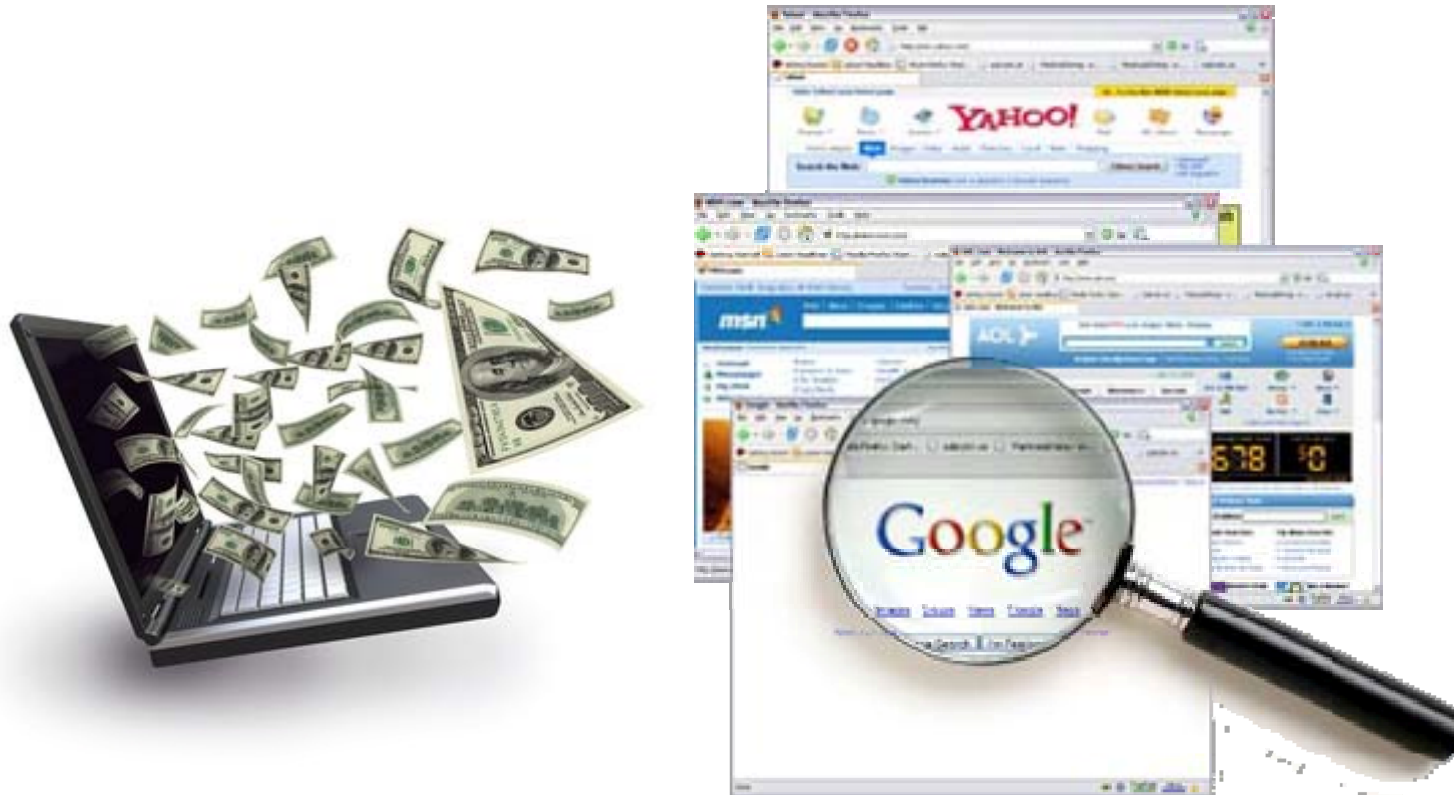
- Servicii web. Studii de caz: programare Web folosind API-urile Amazon, Google.

The collage consists of three main parts:

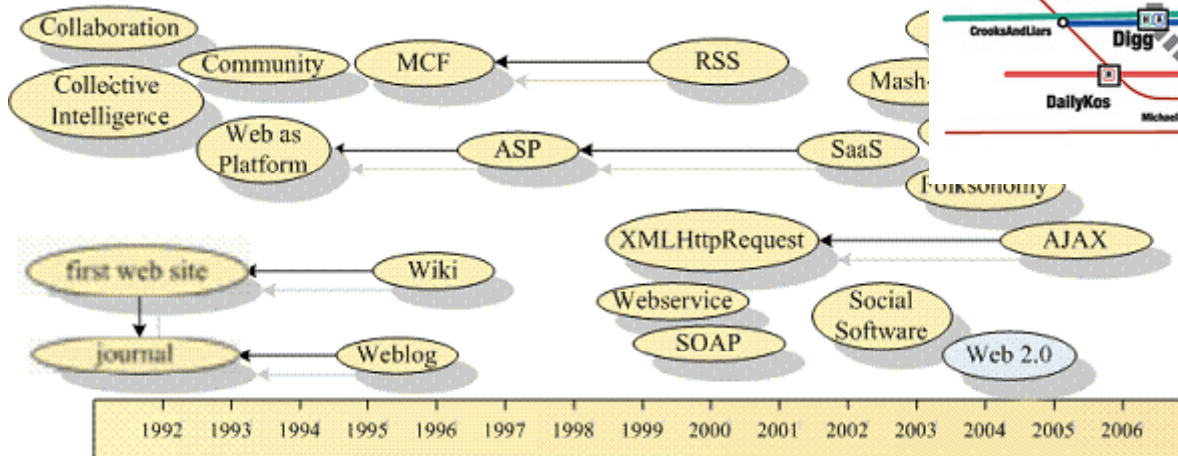
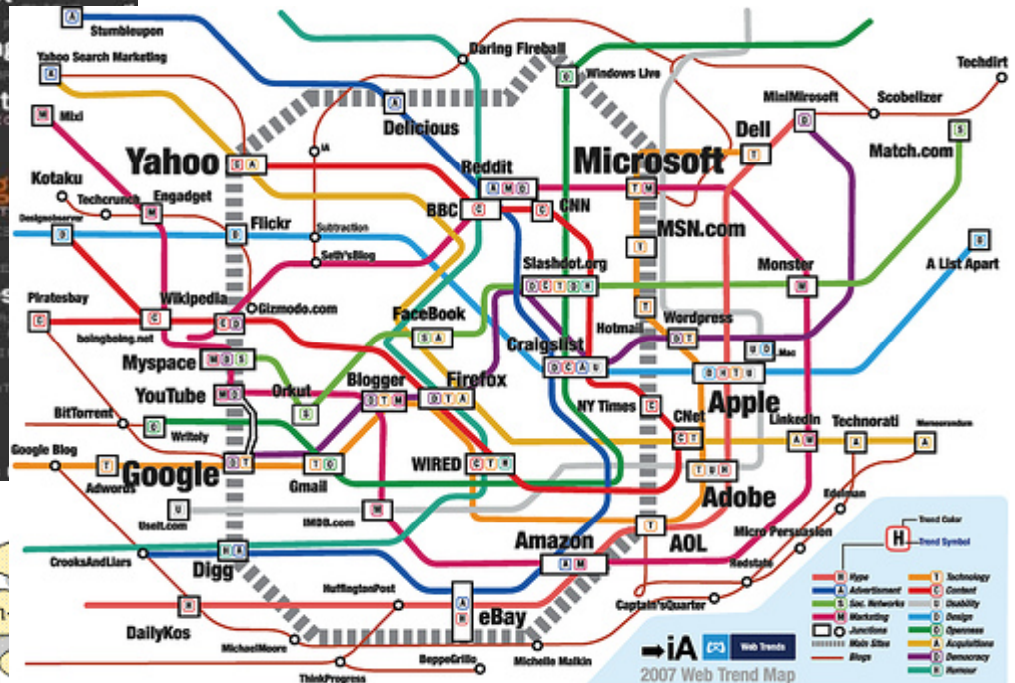
- Map:** A Google Map of Austin, Texas, showing several red location pins. A callout box for 'Lakeline Mall' is visible, with the address '11200 Lakeline Mall Dr, Austin, TX 78613'.
- Spreadsheet:** A screenshot of a Google Docs spreadsheet showing a pie chart. An 'Edit chart' dialog box is open, allowing for chart customization. The chart has four segments in blue, green, orange, and red.
- API Interface:** A screenshot of the Google AdWords API interface. It shows a 'Quota units remaining' section, a 'Your Developer Token' field with a '[reset]' link, and a 'Google AdWords API Terms and Conditions' link. A red arrow points to the developer token field with the text 'This is your API Token ID'.

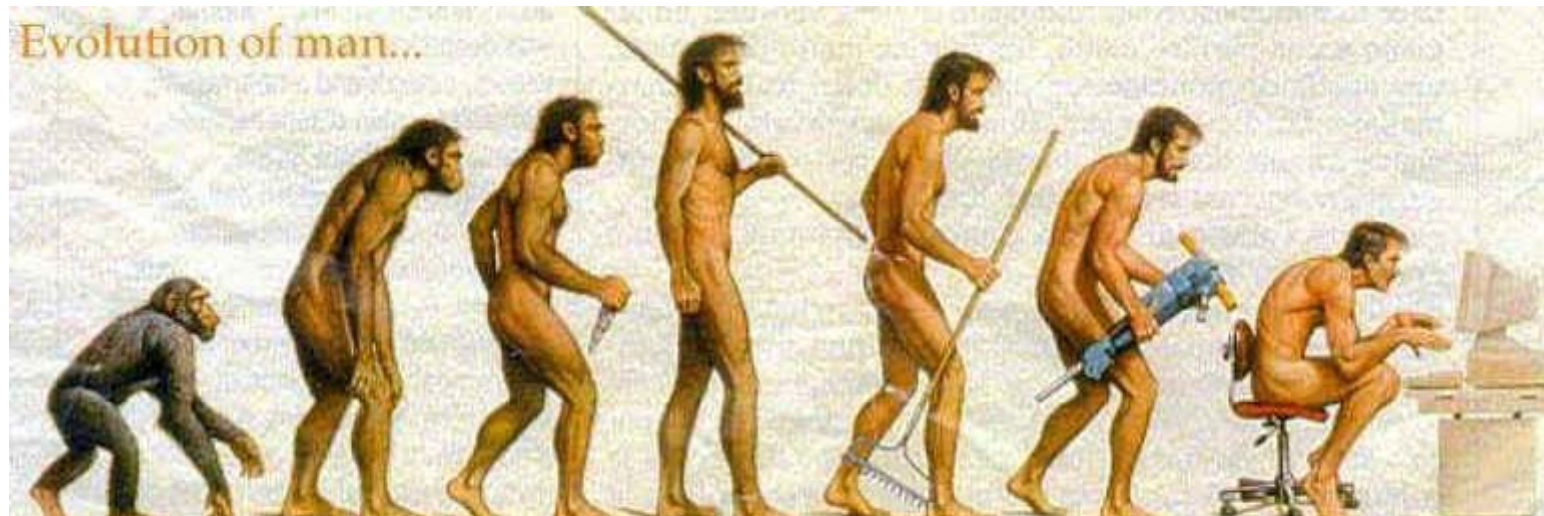
Continut

- Optimizarea paginilor Web pentru indexarea de catre motoare de cautare.



In plus...





Evolutia omului e legata de Web...





Capitole principale (1)

- Arhitectura unei aplicatii Web.
- *Servere HTTP: caracteristici, exemple. Limbajul HTML: elemente de baza, tabele, cadre, formulare. Scripturi scrise in limbaje compilate.*
- Folosirea limbajului SQL pentru programarea web.
ODBC.
- Limbaje de scripting server-side, PHP
- Programare web client-side, javascript



Capitole principale (2)




- Continut si design in programarea web, CSS, DHTML
- Elemente de multimedia pentru programarea web
- Administrarea continutului site-urilor Web: optimizarea paginilor web pentru indexarea de catre motoarele de cautare, solutii profesionale
- Elemente de securitate si optimizare
- Framework-uri Web

Resurse

- Materiale postate pe site

participanți, întrucât conțin informații referitoare la drepturi

General

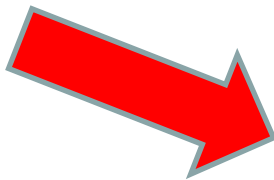
-  Forum stiri
-  Forum general
-  Formular feedback
- Prezenta Laborator

Regulamente

-  Regulament Curs Programare Web
-  Ghid de etica

Resurse curs

-  Prezentrari curs






Notarea

- Regulamentul - pagina cursului



participanți, întrucât conțin informațiile referitoare la drepturi

General

-  Forum stiri
-  Forum general
-  Formular feedback
- Prezenta Laborator



Regulamente

-  Regulament Curs Programare Web
-  Ghid de etica

Resurse curs

-  Prezentari curs



Notare

- Nota finala la cursul de **Programare Web** se calculeaza dupa urmatoarea formula:

$$\text{Nota} = \text{ROUND}(\text{NotaCurs} * 0,1 + \text{PunctajParcurs} * 0,6 + \text{NotaExamen} * 0,4)$$

unde:

- NotaCurs, PunctajParcurs sunt obtinute pentru activitatile din timpul semestrului;
- NotaExamen este nota obtinuta la examen
- Examenul este promovat **numai daca** se obtine minim 50% atat din nota de la examenul final, cat si din punctajul din timpul semestrului



Notare

- NotaCurs se calculeaza pe baza
 - *activitatii* la curs (prezenta)
 - a unor *teste* (neanuntate) si a unor
 - *teme de casa*
- Temele de casa date la curs – de dificultate medie – se predau la cursul imediat urmator celui in care au fost propuse si nu se refac.



Notare

- PunctajParcurs se calculeaza pe baza
 - notelor la temele de laborator (4p),
 - a activitatii de la laborator si
 - a task-urilor propuse după fiecare laborator (1p)
- Se tine cont nu doar de prezenta fizica in laborator, ci este notata mai ales implicarea studentului in realizarea lucrarilor practice din fiecare saptamana.
- Test final laborator (1p)
- Laborator: <http://elf.cs.pub.ro/pw>



Proiect științific (1)

- Suplimentar, până la 1p pot fi recuperate prin participarea în cadrul unui experiment științific
 - Aplicație disponibilă la http://cipism.hpc.pub.ro/MobileExp_bluetooth.html
 - Necesită un telefon cu Android (nu neapărat al vostru, puteți să cooptați un coleg în cadrul experimentului – la înscriere acesta va specifica în rubrica observații pentru cine participă)
 - Reguli:
 - Porniți aplicația de câte ori ajungeți la facultate (respectiv, puteți închide aplicația când plecați de la facultate)
 - Datele culese includ și timpul cât aplicația rulează, se vor verifica datele culese și se vor puncta numai studenții ce au ținut aplicația pornită cel puțin 90% din timpul cât au stat în facultate

Proiect științific (2)

- Înscrierea se face accesând <https://docs.google.com/spreadsheet/ccc?key=0AICMhP0LuoqydE9abklzMV90MHJTbVNDZWpERIE5eGc>
- Urmează și un al doilea experiment ... de trafic





Cap. 1.

Arhitectura unei aplicatii Web



Evolutie

- Arhitectura unei aplicatii informatice folosita intr-o organizatie a trecut prin mai multe etape:

1.Etapa Mainframe

2.Etapa Client-Server

3.Etapa Web Based (aplicatii web sau web-enabled)

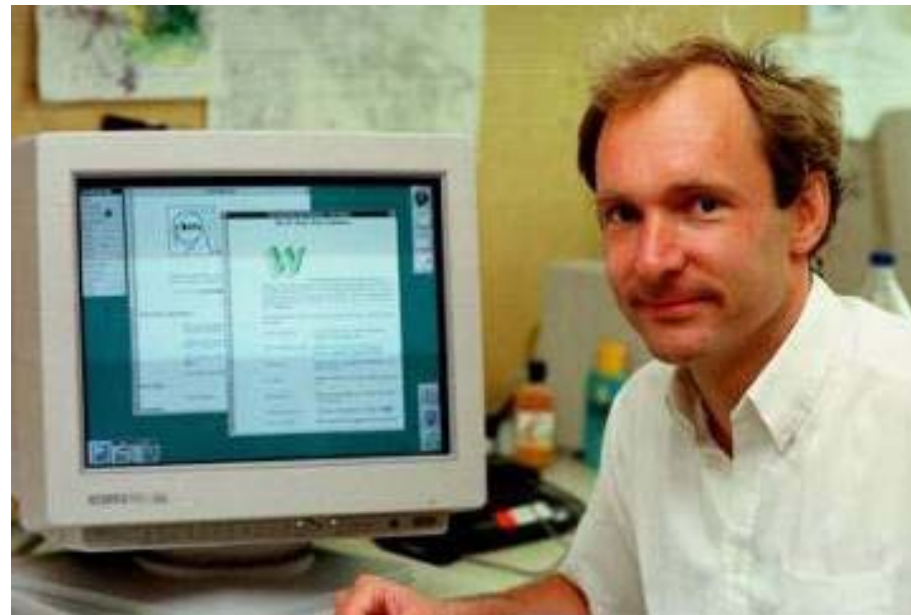
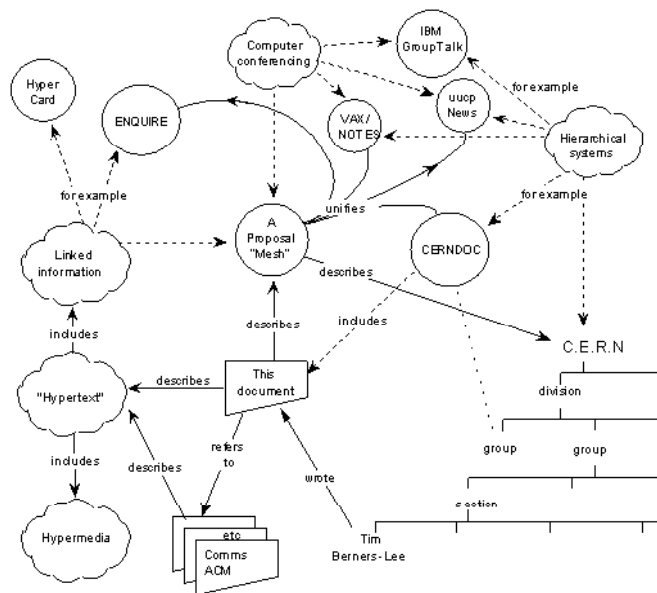
Etapa client-server

- In prima perioada a etapei client-server aplicatiile erau mai ales departamentale:
- In cadrul organizatiei fiecare **departament** (productie, financiar, resurse umane, relatii cu clientii, etc.) avea propria sa **aplicatie** accesata in retea locala a departamentului.



CERN si Web Wide Web

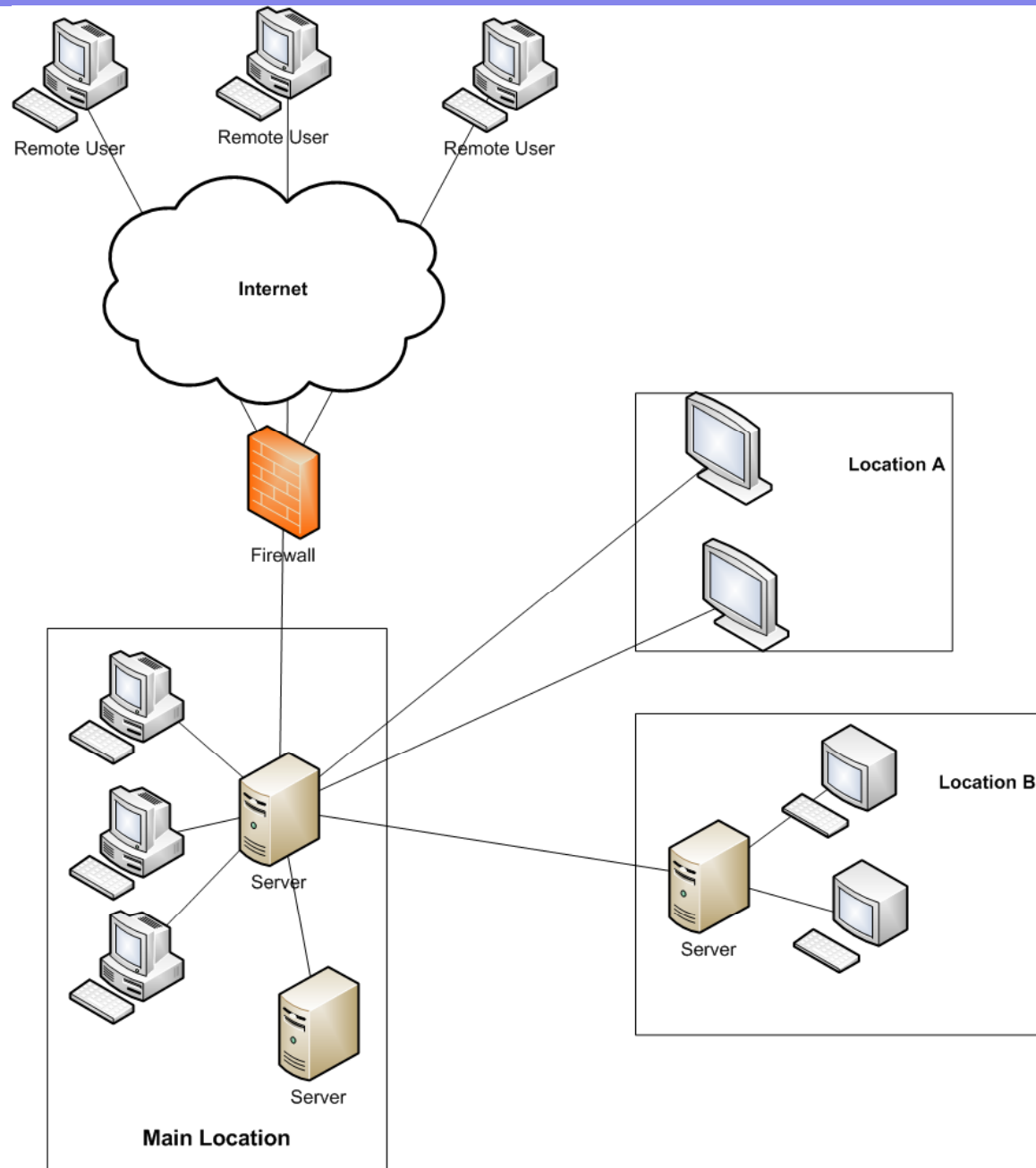
- 1989: **Tim Berners-Lee** publica “Information Management: A proposal” pentru a ajuta viitorul proiect Large Hadron Collider.
- 1991: Primele **sisteme www introduse** fizicienilor prin intermediul bibliotecii de program de la CERN. Primul server web – localizat in laboratoarele de fizica Europeana.
- 1993: Primul browser Mosaic; web-ul atinge 500 de servere si 1% din totalul traficului de Internet.





Etapa web-based

- Deosebirea principala in acest caz este accesarea aplicatiei
 - Printr-un client standard (browserul)
 - De foarte multe ori din afara locatiei unde se afla aplicatia
 - Comunicatia se face nu prin canale private / dedicate ci prin Internet
- O arhitectura tipica este in figura urmatoare:





Conversația cu un server Web

- Conexiune TCP la server pe portul 80 (implicit)
- Browser-ul trimite peste TCP ceva de genul:

```
GET /index.html HTTP/1.0
User-Agent: Mozilla/4.73 [en] (X11; U;
Linux 2.0.35 i686)
Host: www.yahoo.com
Accept: image/gif, image/x-xbitmap,
image/jpeg, image/pjpeg, image/png,
*/*
Accept-Encoding: gzip
Accept-Language: en
Accept-Charset: iso-8859-1,*,utf-8
Cookie: B=2vsconq5p0h2n
```




Conversația cu un server Web

- Serverul răspunde cu:

```
HTTP/1.0 200 OK
Content-Length: 16018
Content-Type: text/html
```

```
<html><head><title>Yahoo!</title><base
  href=http://www.yahoo.com/>
...etc.
```

- Dacă mai avem și imagini embedded images:

```

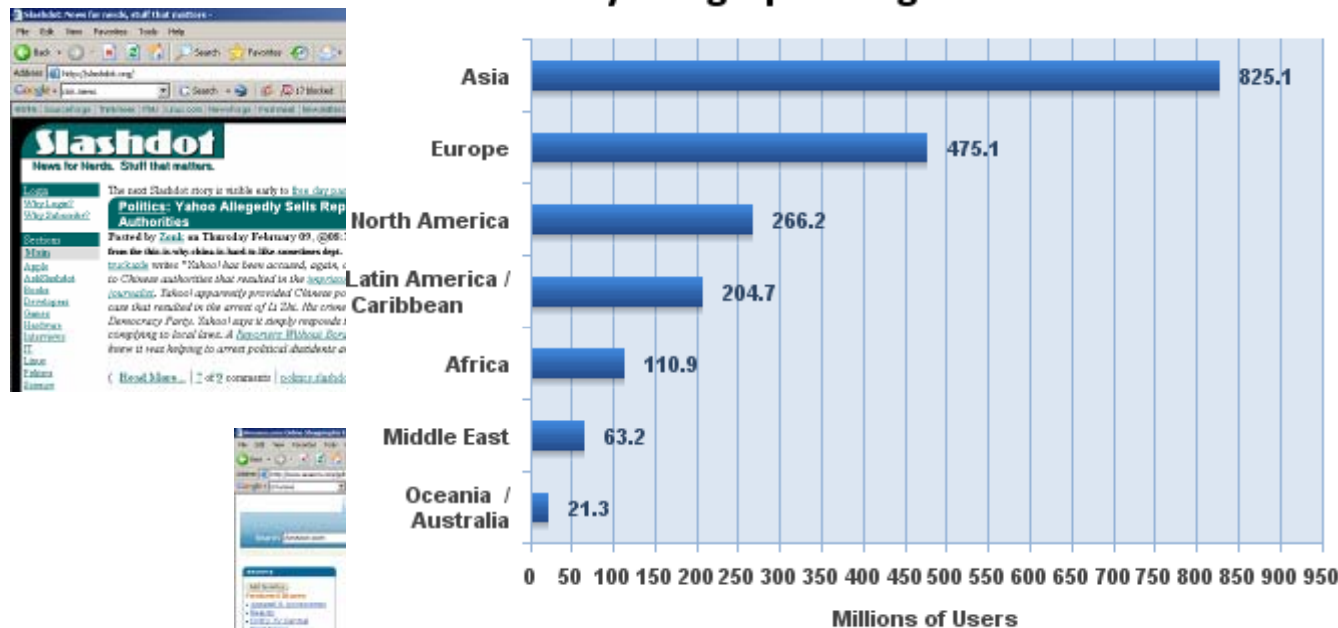
```

...și apoi repetă tot acest proces cu noul URL.

O privire la Web

- Istorie: Decembrie 2005 – al 1 miliard-ulea utilizator online!

Internet Users in the World by Geographic Regions - 2010



Source: Internet World Stats - www.internetworldstats.com/stats.htm
 Estimated Internet users are 1,966,514,816 on June 31, 2010
 Copyright © 2010, Miniwatts Marketing Group



Evolutia Web-ului

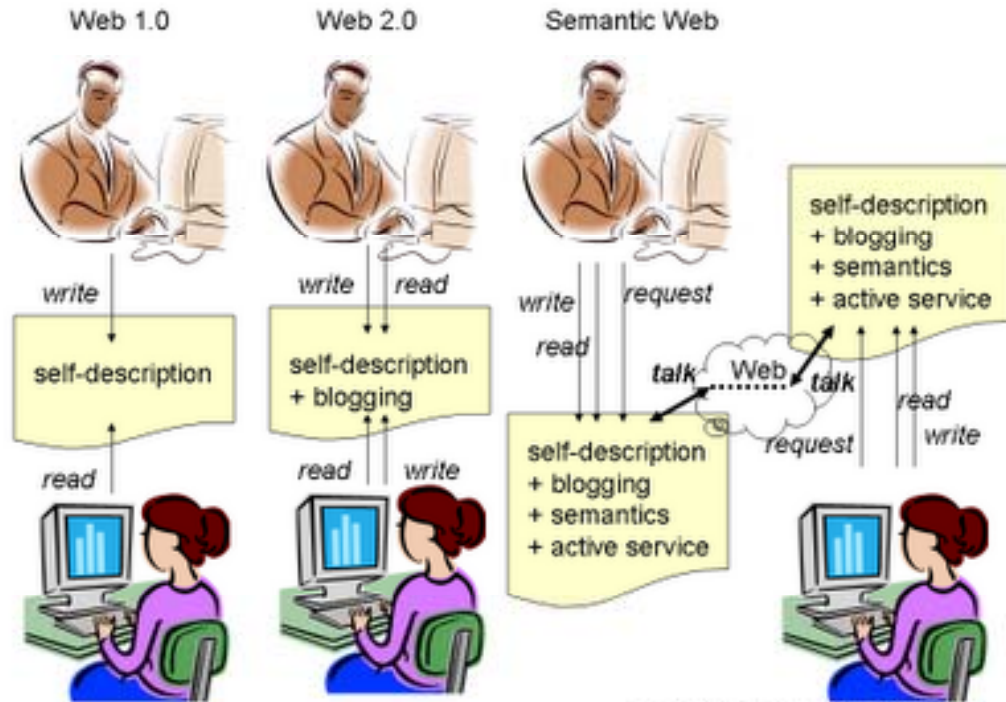


Image from Thinking Space by Yihong Diao

World Wide Web

HTML/CSS & XML/XPath	XLink	RDFa/ µFormats	Ajax/ REST	XSDL & SPath/MPath	(Loc) AtomPub
Documents & Data	Resource Associations	Semantics	Applications	Schemas/ Models	Location-Based Data



Probleme de rezolvat

- In care **locatii** trebuie sa existe server (= bani necesari dotarii cu asa ceva) si in care nu.
- Se analizeaza:
 - Numarul de utilizatori dintr-o anumita locatie
 - Latimea de banda disponibila intre locatia respectiva si locatia principala
 - Timpul de raspuns necesar de asigurat utilizatorilor



Probleme de rezolvat

- Rezolvarea problemelor de **securizare** a transmisiei de date:
 - datele circuland in afara retelei organizatiei trebuie de exemplu criptate corespunzator.
- Problemele legate de **browser**:
 - aplicatia trebuie sa poata fi accesata in acelasi mod si fara pierderi de functionalitate prin oricare din tipurile principale de browser existente.



Probleme de rezolvat

- Probleme legate de **conectivitate**:
 - din toate locatiile de unde aplicatia e necesar sa fie accesibila timpul de raspuns trebuie sa fie in parametrii ceruti.
- De asemenea trebuie apreciat **traficul** generat de o aplicatie in contextul in care aceasta ruleaza in paralel cu alte aplicatii.
 - Performantele pot fi afectate in cazul in care traficul cumulat depaseste anumite praguri.



Cum continuam?

- In partea a doua a cursului veti studia dezvoltarea de aplicatii web complexe.
- In prima parte insa ne vom concentra pe elementele de baza: HTML, folosirea bazelor de date in aplicatii web si limbajul de scripting PHP.



HTTP

- Comenzi text peste TCP/IP
- La bază un protocol cerere-răspuns, cu inițiere din partea clientului
- **Stateless:** fiecare cerere tratată complet independent
 - Nu există metode implicite de creare de asocieri între cereri distincte
 - Nu se furnizează stări persistente
- În plus, încă din primele zile ale HTTP-ului: cookies
 - Header suplimentar adăugat de server unui răspuns HTTP
 - Cookies sunt opace pentru client
 - Clientul ar trebui să trimită înapoi un cookie în cadrul unei cereri succesive către același server



Script CGI

- Scripturile CGI sunt o modalitate de a furniza pagini dinamice HTML sau de alt tip (imagini, video, sunet, postscript, etc).
- Acestea nu sunt efectiv stocate ca atare pe serverul web ci sunt generate pe baza datelor obtinute de la utilizator (în general conform datelor completate de catre acesta într-un formular).
- Din aceasta cauza ele nu pot fi servite direct de serverul web ci trebuiesc create de catre un program care se numeste script CGI.



Script CGI

- Numele de *script CGI* provine din denumirea specificatiei interfetei între serverul de web si astfel de programe, numita **C**ommon **G**ateway **I**nterface - prescurtat CGI.
- Teoretic orice program executabil care respecta anumite conventii în ceea ce priveste mediul din care își colectează datele initiale si modul în care scrie rezultatul poate fi un script CGI.



Exemplu

- Cel mai simplu exemplu de script este un program care pentru orice date de intrare afiseaza un mesaj standard. Iata un exemplu scris în limbajul C:

```
#include <stdio.h>
int main()
{
printf("Content-Type: text/html\n\n");
printf("<html>\n");
printf("<body>\n");
printf("Sunt un script CGI\n");
printf("</body>\n");
printf("</html>\n");
}
```



Exemplu

- Rezultatul rularii acestui script este urmatorul:

Content-Type: text/html

[Linie goala]

```
<html>
```

```
<body>
```

```
Sunt un script CGI
```

```
</body>
```

```
</html>
```



Script CGI in pagina web

- Un script CGI este un fisier executabil.
 - Poate proveni dintr-un program compilat (C sau alt limbaj) sau
 - Poate fi un fisier executabil scris în shell sau alt limbaj interpretat (cum este Perl).
- Exista si posibilitatea de a include scripturile direct în fisierele HTML.
 - O pagina web va contine atât directive HTML care sunt interpretate obisnuit de browser cât si programe scrise în anumite limbaje care vor fi înlocuite cu rezultatul executiei lor.
- Exista doua abordari în acest caz:



Client side

- **Abordarea client-side:** paginile continând scripturi sunt trimise de serverul web fara nici o modificare catre browser, urmând ca executia lor sa aiba loc acolo.
- Este cazul *limbajului Java* (applet-uri, aplicatii JavaWS, JavaFX) pe care browserele uzuale îl contin nativ.
- O parte a procesarilor se transfera de la server => micșorarea volumului de activitate pentru primul.



Server side

- **Abordarea server-side:** serverul înlocuieste în pagina ceruta programul cu rezultatul acestuia.
- Pentru aceasta serverul trebuie sa aiba suport pentru limbajul respectiv.
- Cursul prezinta limbajul de scripting server side PHP



CGI – date de intrare

- Fiecare formular are asociat o metoda prin care datele completate de utilizator sunt transmise de serverul de web catre scriptul CGI.
- Aceste informatii sunt împachetate sub forma unui sir de perechi de forma:

`simbol1=valoare1&simbol2=valoare2&...&simbolk=valoarek`

unde numele simbolilor este dat de numele câmpurilor formularului iar valorile sunt cele furnizate de utilizator prin completare.



CGI – date de intrare

- Metodele care pot fi folosite sunt în număr de două:

1. Metoda **GET**

- datele sunt stocate într-o variabilă de mediu (environment) cu nume predefinit (QUERY_STRING)

2. Metoda **POST**

- datele sunt plasate în fișierul standard de intrare al scriptului (stdin în cazul scripturilor scrise în limbajul C)
- lungimea șirului care conține datele este stocată într-o variabilă cu nume predefinit (CONTENT_LENGTH)



CGI – date de intrare

- Pe lângă aceste informații serverul setează o serie de alte variabile de mediu conținând o multitudine de alte informații.
- În cazul scrierii de scripturi în limbajul C acestea se pot obține folosind funcția *getenv()* iar în cazul scripturilor care utilizează modulul PHP acestea sunt direct disponibile.



Stocarea pe partea de server

- Cum se opt menține *persistente* datele pentru mai multe cereri HTTP provenite din partea unui același utilizator?
 - Datele pot fi incluse (embedded) în URL, cookie, sau variabile/câmpuri ascunse în formulare
 - Mai bine: Stocăm datele pe server și includem în cereri doar handlere (adevărul e cunoscut doar de server, programul devine defensiv)
- De ce e mai bine să stocăm datele pe server?
 - Clienții nu sunt de încredere, dimensiunea datelor....
 - Handlerul poate fi semnat criptografic, poate avea un termen de expirare, etc. (un client nu poate falsifica datele)



Sumar

- Evolutia arhitecturilor
- Arhitecturi Web
- Probleme cu dezvoltarea aplicatiilor Web
- Scripturi CGI

