# Capitolul 4

# World Wide Web (WWW)

Este un serviciu complex pentru accesul la documente, raspandite pe calculatoare din intreaga lume. In 5 ani a evoluat de la o aplicatie pentru transmiterea datelor de cercetare in domeniul fizicii, la o aplicatie pe care majoritatea oamenilor o confunda cu reteaua Internet. Are o interfata usor de utilizat chiar de catre incepatori si ofera o cantitate imensa de informatii in orice domeniu.

Permite:

- localizarea si accesarea informatiei de pe calculatoare aflate la distanta intr-un mod interactiv
- afisare text, imagini
- transmisiuni audio si video in direct
- accesarea informatiilor de la mai multe servicii Internet folosind un singur mecanism

## 4.1 Notiuni Web

**Web** (**WWW** sau **W3**) a aparut in 1989 la CERN (**C**entre **E**uropean de **R**echerche **N**ucleaire) in Elvetia. **Tim Berners-Lee** a inventat WWW ca mod de organizare a informatiei pentru a pune la dispozitia cercetatorilor din intreaga lume informatii din domeniul fizicii particulelor: colectii de rapoarte, planuri, desene si fotografii aflate intr-o continua modificare.
Propunerea sa a fost facuta in martie 1989, iar primul prototip functional in mod text a aparut 18 luni mai tarziu. Prima interfata grafica a aparut in februarie 1993, numita Mosaic.

In **1994** CERN si MIT (**M**assachusetts **I**nstitute of **T**echnology) au semnat un acord pentru a forma **Consortiul World Wide Web**, organizatie ce are ca obiectiv dezvoltarea web-ului, standardizarea protocoalelor si incurajarea interoperabilitatii intre site-uri. Are aproximativ

450 organizatii membre (universitati si companii) din intreaga lume.

Din punctul de vedere al utilizatorului consta dintr-o vasta colectie de documente raspandite in intreaga lume, numite pagini web. Fiecare pagina poate contine legaturi catre alte pagini, ce se pot afla pe acelasi calculator ca si documentul de la care s-a facut referirea, sau pe un calculator aflat oriunde in lume. Documentele care contin legaturi catre alte documente se numesc **hypertext**. Paginile pot fi vizualizate cu ajutorul unui program de navigare (**browser**).

Acesta transfera pe calculatorul local pagina ceruta, interpreteaza comenzile de formatare continute in document si afiseaza textul din acesta formatat corespunzator. Caracterele ce reprezinta referiri catre alte pagini se numesc **hyper-legaturi** (hyper-link) sau simplu **link**, acestea fiind afisate in mod diferit fata de restul textului (subliniate si folosind alte culori - modul implicit) pentru a fi recunoscute de catre utilzator.

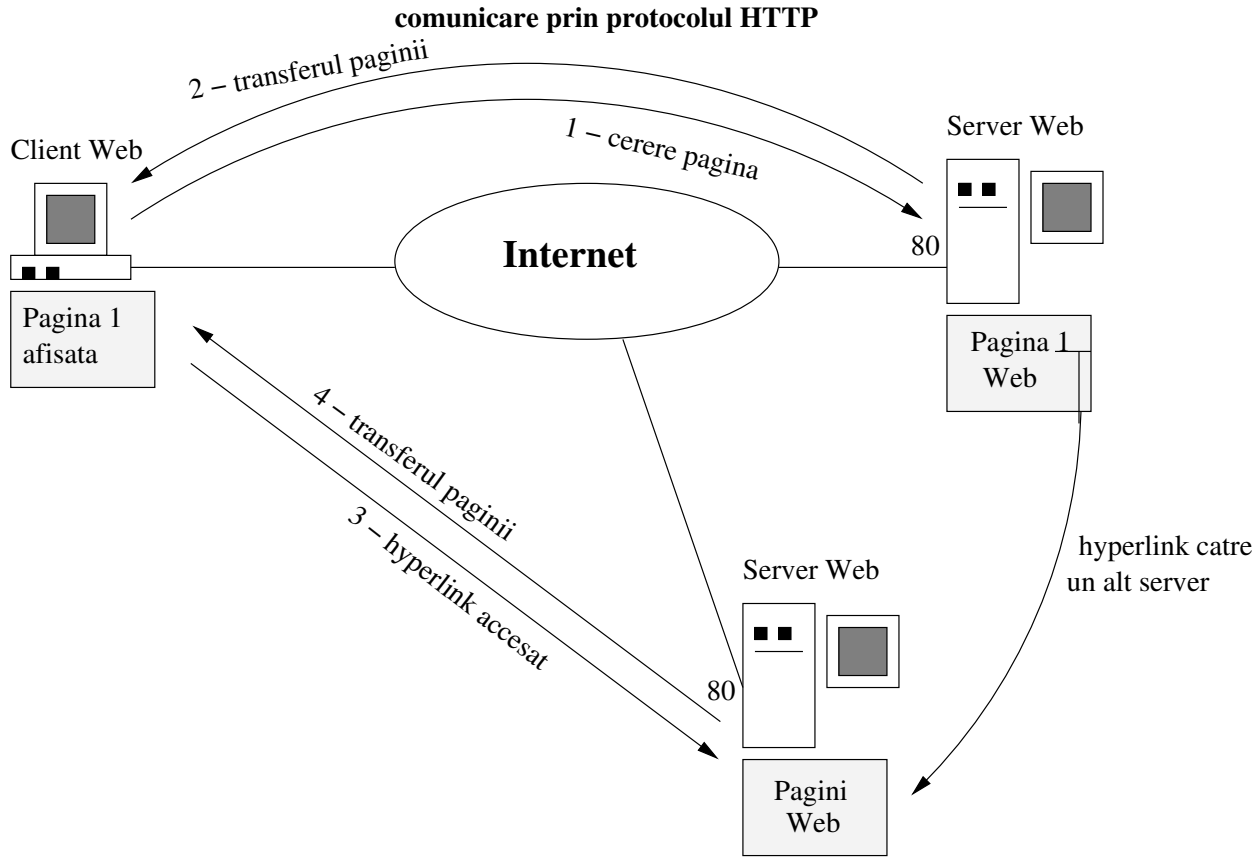


Figura 4.1: Functionare Web

In figura 4.1 este indicat modul de comunicare intre server si client. Serverul de web este un program care accepta conexiuni de la clienti (programele de navigare) pe portul 80. Dupa ce clientul stabileste legatura cu serverul trimite cererea pentru pagina web, si serverul va trimite

inapoi raspunsul: fisierul respectiv sau un mesaj de eroare daca acel fisier nu se gaseste. Dupa aceea conexiunea este inchisa de catre server. Daca se acceseaza un nou link in pagina ceruta, clientul stabileste din nou conexiunea la server si cere documentul respectiv. Fiecare cerere consta dintr-o comanda de genul: "GET nume_fisier".
Protocolul folosit pentru transmiterea paginilor web este **HTTP** (**H**yper**T**ext **T**ransfer **P**rotocol). Este folosit in interactiunea dintre client si serverul de web.
In principiu un server de web de baza este un program destul de simplu care preia cererile si trimite inapoi raspunsul. La aceasta functie de baza insa se mai adauga si alte functii cum ar fi CGI sau posibilitatea de securizare a accesului. De exemplu se pot proteja anumite pagini, fiind accesibile numai celor care au un nume de utilizator si parola. Un alt nivel de securitate este de a realiza o conexiune criptata intre navigator si server prin care se transmit date confidentiale (cum ar fi numarul cartilor de credit, informatii despre conturi bancare etc.).
Exista doua elemente cheie ce stau la baza functionarii WWW-lui:

- paginile web sunt scrise in limbajul HTML care este un limbaj de "marcare" a textului. Acestea sunt cuvinte cheie pe care le recunoaste navigatorul si conform acestora formateaza textul si il afiseaza utilizatorului
- fiecare pagina web are o adresa unica, numita **URL**.

Astfel cei doi factori - codarea HTML si URL-urile unice ofera Web-lui flexibilitate si usurinta in utilizare:

- HTML permite ca intr-un fisier text simplu sa se includa liste, tabele, formulare, imagini si legaturi la alte fisiere
- URL-ul permite referirea cu usurinta la o alta pagina Web. Astfel se pot face referiri la alte informatii relevante legate de documentul respective, disponibile pe alte servere Web din lume.

### 4.1.1 URL (Uniform Resource Locator)

Pentru a putea localiza o pagina Web este necesar un mecanism care sa permita adresarea lor - **URL**. Sunt necesare trei elemente pentru a localiza o pagina Web:

- adresa serverului web (numele de domeniu al serverului)
- unde este localizata pagina pe serverul de Web (numele fisierului, cu intreaga cale catre acesta)
- protocolul prin care se acceseaza pagina.

Aceste trei elemente formeaza URL-ul, care identifica tipul documentului, locatia pe Internet si numele fisierului.

Forma generala a unui URL este:
<**protocol**>://<**nume.de.domeniu.server**>/<**cale_catre_pagina** >/<**nume_pagina**> .

Referinta la un director reprezinta de fapt si acela un fisier, de obicei **index.html**, **index.htm**, **index.php**, **index.asp**, **index.shtml**.

### 4.1.2 HTML (HyperText Markup Language)

Este limbajul folosit pentru scrierea paginilor Web, fiind un limbaj de marcare care descrie browserului modul de formatare a textului din cadrul paginii.
Versiunea HTML 1.0 functiona intr-o singura directie: utilizatorul putea transfera pagini de pe un server web, dar nu putea transmite informatii in sens invers. Tot mai multe organizatii comerciale si-au facut prezenta pe Web si a aparut necesitatea comunicatiei in ambele directii (de exemplu: preluare comenzi, la motoarele de cautare se introduc cuvinte pe baza carora se fac cautarile). Asemenea elemente au fost introduse incepand cu HTML 2.0. Dupa aceea a aparut suport pentru tabele, formulare.
Versiunea actuala (noiembrie 2003) este 4.01. Mai multe informatii gasiti la pagina consortiului WEB: **www.w3.org**.
Paginile HTML folosesc limbajul **CSS** (**C**ascading **S**tyle **S**heets) pentru a imbunatati designul unei pagini. **CSS** ofera dezvoltatorilor paginilor web o unealta simpla si eficienta pentru a simplifica operatia de adminstrare/actualizare a unui site web, si ofera proprietati de design sofisticate. CSS este recomandarea consortiului World Wide Web, aparuta in anul 1996. Un **style sheet** este un set de proprietati pentru browser ce indica cum sa fie formatate diferitele taguri ale unui document.
Exista 2 versiuni: **CSS1** sau nivelul 1 si **CSS2** sau nivelul 2. CSS2 ofera compatibilitate cu CSS1 si are in plus multe optiuni noi.
Un style sheet CSS1 contine cinci tipuri de baza de informatii de prezentare, numite proprietati:

- Proprietati pentru culori si fundal.
- Proprietati pentru fonturi.
- Proprietati pentru text (spatierea intre cuvinte, intre litere, etc.)
- Proprietati pentru blocuri (marginea si intre elementele blocului, etc.).
- Clasificari (control asupra stilului listelor si formatarea elementelor)

Problema principala in trecut a fost lipsa suportului pentru CSS in multe din browsere, astfel incat pagina nu era afisata cum dorea designerul. In ultimii 2 ani majoritatea browserelor au introdus suport si pentru CSS, cum ar fi Internet Explorer 5+, Netscape 6+, astfel CSS fiind utilizat fara in paginile web.

### 4.1.3 XML (eXtensible Markup Language)

Este un meta limbaj ce permite utilizatorilor sa structureze si defineasca informatiile continute intr-un document. XML a fost dezvoltat pentru a **descrie datele**, si utilizeaza taguri, elemente si atribute pentru a descrie intr-o maniera clara continutul unui document. Nu are tag-uri predefinite, programatorul trebuie **sa defineasca tag-urile**, de aceea este numit limbaj extensibil. XML nu este un inlocuitor pentru HTML, ci o extensie a acesteia. HTML este utilizat pentru a formata documentul ce trebuie afisat, iar XML este folosit pentru a defini informatia din cadrul documentului.
XML nu a fost dezvoltat pentru a "face" ceva, exista doar pentru a structura, stoca si a trimite datele. Mai jos aveti un exemplu XML:

<nota>
<to>Ion</to>
<from>Radu</from>
<heading>Reminder</heading>
<body>Nu uita sa-mi trimiti actele pana la week-end!</body>
</nota>

Exemplul de mai sus este doar informatie intre taguri XML. Trebuie scris un soft pentru a trimite, receptiona si afisa aceasta informatie.
Ca o definitie XML este **o unealta independenta de hardware si software pentru transmiterea datelor**. Cateva din avantajele XML:

- permite distribuirea unei parti importante a procesarii datelor de la server la client
- este un limbaj extensibil
- ofera informatii despre continutul documentului (tag-urile, atributele si elementele ofera informatie de context ce poate fi folosita pentru a interpreta **sensul continutului**, oferind noi posibilitati de exemplu pentru motoare de cautare foarte eficiente).
- documentele XML contin diverse tipuri de date (audio, video, PDF, ActiveX, appleturi Java)

### 4.1.4 Etapele transferarii unei pagini web

Continutul unei pagini web poate fi text, imagine, inregistrare audio sau video, transmisiuni directe audio, video. Din momentul introducerii URL-ului unei pagini pana la afisarea acesteia etapele parcurse sunt:

1. navigatorul **contacteaza serverul DNS local** pentru a afla adresa IP pentru numele de domeniu al destinatiei.

2. navigatorul primeste adresa IP si **stabileste conexiunea** cu serverul web de la acea adresa pe portul 80, sau pe cel specificat in URL

3. clientul **trimite comanda de cerere** al fisierului HTML si daca aceasta adresa are si cookie-uri pe hard-disk, il trimite si pe acesta odata cu cererea.

4. serverul web **transmite fisierul** cerut si cookie-urile, daca exista

5. serverul **inchide conexiunea** dupa ce a transmis fisierul catre client. Pentru a transfera o alta pagina de la acel server, clientul trebuie sa deschida o noua conexiune catre server.

6. navigatorul **afiseaza pagina** formatata corespunzator pe baza tag-urilor HTML

Programele de navigare (Internet Explorer, Netscape, Opera, etc. ) afiseaza in bara de stare a navigatorului majoritatea etapelor care se executa, astfel incat utilizatorul poate determina daca problema este la serverul DNS, sau nu este buna adresa, sau momentan nu este disponibil serverul Web respectiv.
Pentru fiecare imagine care apare pe o pagina Web navigatorul stabileste o noua conexiune si aduce acea imagine. Acest lucru se datoreaza faptului ca implementarea protocolului http a fost mai simplu astfel. Daca o pagina web contine multe imagini atunci transferul acelei pagini se efectueaza bineinteles mai lent.

## 4.2 Interactiune cu utilizatorul in paginile web

### 4.2.1 CGI (Common Gateway Interface)

Etapele descrise mai inainte reprezinta modul de servire a unei pagini "statice". Sunt insa si pagini dinamice, cum ar fi de exemplu cazul in care introduceti cuvintele pe pagina unui motor de cautare si acesta va afiseaza rezultatul. Sau cand completati un formular acesta trebuie procesat de catre un program.
Toate acestea presupun ca odata primita cererea de la client serverul de web sa transmita aceasta cerere catre o aplicatie (program pe serverul de web) care proceseaza datele primite (datele unui formular, cerere de cautare etc.), si dupa rulare transmite datele rezultate inapoi la server-ul de web, care le transmite la randul lui clientului care a facut cererea.
Metoda standard de a transmite o asemenea cerere de catre server catre o aplicatie este **CGI** (**C**ommon **G**ateway **I**nterface). In acest caz exista deci un flux de date bidirectional intre utilizator si server, ce permite:

- accesarea unei baze de date si a oferi rezultatul documentelor HTML
- generare de formulare HTML pentru introduceri de date
- interactiune cu documente on-line pentru a realiza cautari

Pentru a intelege cum functioneaza CGI-urile in urmatoarele paragrafe se va face o scurta descriere a functionarii serverelor web. Cererea trimisa de client catre serverul web contine mai multe elemente ce sunt numite **campuri**. Singurul camp care este necesar este campul **cerere**, care este primul camp al unei cereri. Asa cum sugereaza si numele, acest camp este suficient pentru a cere o pagina web de la un server (acest lucru insa nu inseamna ca serverul va transmite URL-ul corespunzator, in formatul corepsunzator fara a avea informatia din restul campurilor: inseamna doar ca serverul va raspunde la cerere).
Cand primeste o cerere, serverul efectueaza mai multe operatii inainte sa trimita inapoi clientului URL-ul cerut:

- decide locatia URL-ului pe sistemul local de fisiere. Serverul trebuie sa "rezolve" URL-ul pe sistemul local de fisiere.
- decide daca URL-ul necesita autentificare pe baza de nume si parola. Diverse locatii pe sistemul de fisiere pot fi protejate prin diverse metode, incluzand necesitatea unui nume utilizator si parola pentru a permite acces numai anumitor utilizatori.
- decide daca URL-ul necesita tratare speciala. In functie de locatia URL-ului, extensia fisierului si alti factori, serverul poate fi configurat sa efectueze o operatie speciala. Una din aceste tratari speciale presupune rularea unei aplicatii externe de pe acelasi server - un script CGI.
  De exemplu liniile de mai jos specifica ca URL-ul ce incepe cu **/cgi-bin/** (scipturi CGI ) va fi tratat ca un script executabil, si indica locul (directorul) in care se afla acele scripturi.
  De exemplu configuratia utilizatat pe un server web Apache:

```
# ScriptAlias: This controls which directories contain server scripts.
# Format: ScriptAlias fakename realname

ScriptAlias /cgi-bin/ /usr/local/etc/httpd/cgi-bin/
```

- dupa efectuarea unor operatii speciale trimite URL-ul clientului. Serverul inchide conexiunea si serveste o alta cerere.

In cazul tratarilor speciale este posibil ca serverul sa citeasca si sa proceseze un document inainte sa-l trimita clientului. Acest lucru se numeste server-side includes, desi termenul **server parsing** este mai exact.
Un script CGI poate fi utilizat in doua moduri:

1. poate fi inclus in pagina web si atunci cand utilizatorul acceseaza pagina scriptul CGI va fi executat si rezultatul acestuia este inserat in pagina trimisa inapoi utilizatorului. Aceasta este metoda numita **server parsing**.
2. este apelat direct de catre utilizator prin specificarea numelui scriptului in URL-ul trimis la server.

Prima metoda este o versiune mai simpla a celei de-a doua metode, fara a necesita prea multe operatii din partea serverului, de aceea se va descrie ce-a de-a doua metoda.

Atunci cand serverul web primeste de la client cererea, creaza variabile de mediu pe baza mesajului de cerere. Pentru a transmite informatii scriptului CGI, serverul foloseste atat argumente in linia de comanda cat si variabile de mediu, ce este metoda principala utilizata de CGI de a primi informatii.

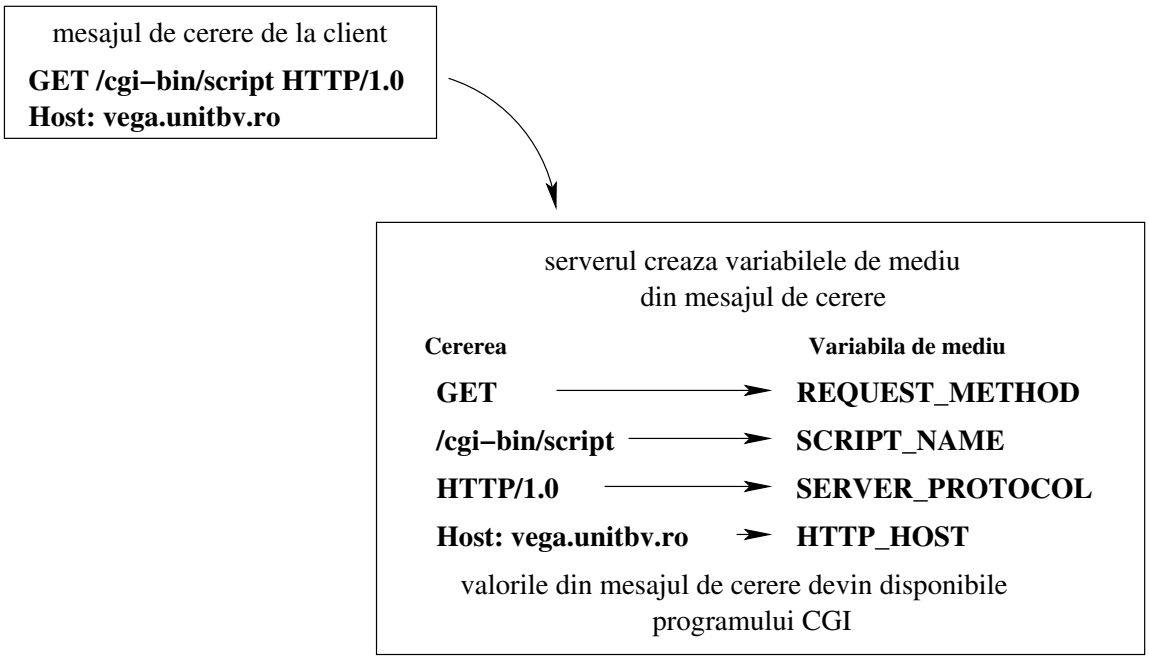


Figura 4.2: Crearea variabilelor de mediu

Daca clientul apeleaza scriptul CGI cu parametrii, acestea sunt salvate in variabila de mediu **QUERY_STRING**. Deoarece majoritatea cererilor sunt procesate in browser prin utilizarea unui formular, sirul de cerere este automat creat de catre navigator. Este posibila accesarea scriptului CGI fara introducerea datelor de catre utilizator, caz in care sirul de cerere trebuie creat manual si atasat la sfarsitul URL-ului. Sirul de cerere pentru un script CGI intotdeauna incepe cu semnul **?**:
http://vega.unitbv.ro/cgi-bin/script**?param**, unde (**param** este sirul de cerere).
Aceata este metoda de a transfera informatii serverului folosind metoda GET, care este metoda implicita utilizata de navigatoare pentru accesarea URL-urilor. Acesata are limitari serioase, deoarece lungimea sirului de cerere poate fi de numai 1024 de caractere. Unele browsere se blocheaza cand lungimea cererii depaseste limita, si multe servere nu accepta intregul sir daca este prea lung. Pentru a transmite cantitati mai mari de date serverului, se utilizeaza metoda POST. O comunicare POST presupune ca clientul trimite urmatoarele elemente serverului:

- un antet, mesajul de cerere, cu metoda POST
- trimite o linie goala, pentru a termina antetul
- datele, sau continutul, numita si entitate

Evident serverul rezolva cererea POST diferit fata de cererea GET. Acestea sunt etapele simplificate (figura 4.3):

- serverul citeste mesajul de cerere, vede ca este o cerere POST si salveaza datele de la client
- serverul creaza variabilele de mediu din toate elementele mesajului de cerere, ca si la metoda GET. Unul din variabilele importante este **CONTENT_LENGHT**, care este lungimea in octeti a entitatii trimise (continut) de client.
- serverul gaseste metoda necesara pentru a trimite continutul la URL-ul cerut. In cele mai multe cazuri URL-ul este un script CGI executabil, astfel incat serverul trimite continutul primit de la client la intrarea scriptului CGI. Similar, scriptul scrie rezultatul la iesirea standard sau un fisier, care este redirectat de server catre clientul care a facut cererea.
- in final serverul inchide conexiunea
- clientul pregateste datele primite de la server pentru afisare, bazandu-se pe informatia din antet. Daca de exemplu antetul specifica **Content-type** de tipul **audio/basic**, navigatorul il va trimite la aplicatia corespunzatoare pentru a rula fisierul de sunet.

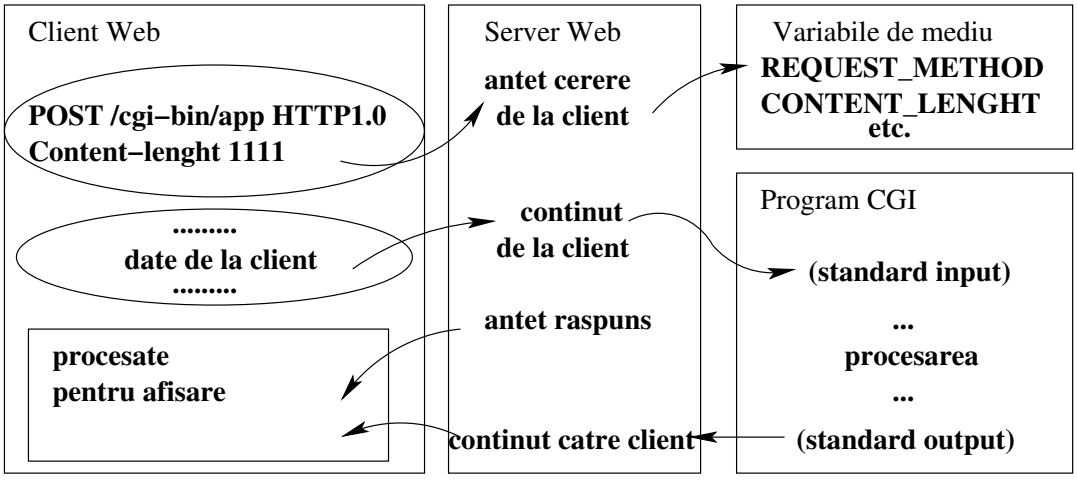


Figura 4.3: Tranzactia completa POST

Scripturile CGI sunt scrise in limbaje de programare ca C, C++, Perl, TCL sau Python. In general aplicatiile CGI sunt scripturi pentru ca sunt mai usor de scris si depanat decat programele compilate C sau C++.
Alternative la scripturile CGI sunt Active Server Pages (ASP) de la Microsoft si PHP.

### 4.2.2 ASP (Active Server Pages)

ASP este un limbaj scriptic, codul fiind inclus intr-o pagina HTML si care este procesat pe un server web Microsoft inainte ca pagina sa fie trimisa la utilizator. ASP-ul este similar cu CGI-uri in sensul ca ambele implica programe ce **ruleaza pe calculatorul** pe care se afla serverul de web, si care transmit rezultatul inapoi serverului web. In mod tipic script-ul de pe server primeste ca intrare cererea de la utilizator de a accesa datele dintr-o baza de date si apoi le serveste in forma HTML serverului care il transmite navigatorului. Scripturile ASP sunt create in **VBScript** (Visual Basic Scripting Edition) sau **JScript** (limbajul dezvoltat de Microsoft, echivalentul limbajului Javascript dezvoltat de Netscape), incluzandu-le in pagina HTML sau folosind instructiuni **ActiveX Data Objects** (ADO) in fisierul HTML.
Paginile ce contin ASP au extensia **.asp**

### 4.2.3 PHP

PHP este un limbaj scriptic distribuit gratuit. **PHP**, a carei initiale au originea de la **P**ersonal **H**ome **P**age **T**ools, a devenit acronimul pentru: **P**HP: **H**ypertext **P**reprocessor. Este o varianta alternativa pentru ASP pentru sistemele de operare Linux, dar care poate rula si pe Windows. Acest limbaj poate fi rulat de serverele web Apache (care are varianta si pentru sistemul de operare Windows) si mai nou si de IIS (Microsoft), deci practic limbaul PHP este disponibil ambelor sisteme de operare, pe cand ASP poate rula numai pe Windows.
Scriptul PHP este inclus in pagina web. Inainte ca pagina sa fie transmisa utilizatorului serverul web apeleaza interpretorul PHP pentru a efectua operatiile mentionate in scriptul PHP.
Paginile ce au inclus scripturi PHP au una din extensiile extensia **.php**, **.php3** sau **.phtml**.

### 4.2.4 JavaScript

**JavaScript** este un limbaj scriptic interpretat, orientat pe obiect, dezvoltat pentru web de catre Netscape. Chiar daca numele sunt asemenatoare, nu are nici o legatura cu limbajul Java. In timp a devenit mult mai popular decat Java in ceea ce priveste utilizarea pe web, pentru ca Javascript este un limbaj puternic, usor de utilizat, rapid, si este cel mai favorit limbaj pe web. Utilizarea originala si cea mai populara a Javascript-ului este schimbarea imaginilor cand se pozitioneaza mouse-ul peste o imagine.
Secventa de cod JavaScript este inclusa in pagina HTML, si este executata de catre browser in momentul incarcarii paginii.

Dupa ce Javascript a inceput sa castige popularitate, Microsoft a creat versiunea proprie de Javascript numita **JScript**, prima versiune aparand in iulie 1996 in Internet Explorer3.0. Aceasta era o versiune redusa a Javascript1.1 Din momentul respectiv exista atat Javascript cat si Jscript, de unde vin si diferentele de interpretare a scripturilor de catre navigatoare. In acelasi timp Microsoft a lansat si propriul limbaj **VBScript**, care este insa mult mai putin folosit decat JavaScript.

Pentru a aduce versiunea JScript la compatibilitate cu JavaScript1.1, Microsoft a mai lansat o noua versiune de JScript, insa unele scripturi rulau bine in Netscape si nu in Internet Explorer si invers.
Netscape si Sun au apelat la o terta parte, la ECMA (European Computer Manufacturers Association - http://www.ecma.ch), pentru a standardiza limbajul JavaScript. Noul standard a fost numit **ECMAScript**, deci in momentul de fata sunt trei standarde separate: JavaScript, JScript si ECMAScript.

Un lucru important de avut in vedere cand sa utilizeaza JavaScript este compatibilitatea cu navigatoarele. Cel care creaza pagina web trebuie sa se asigure ca versiunea limbajului pe care utilizeaza este compatibila cu cat mai multe navigatoare.
Cateva exemple pentru care JavaScript este foarte popular si mult utilizat:

- continut dinamic, cum ar fi un text ce ruleaza in fata utilizatorului
- HTML dinamic: manipulari de imagini, pozitionare elemente pe pagina
- **validarea si preluarea** datelor dintr-un formular
- client-side CGI, permite ca aplicatiile sa fie scrise in Javascript si astfel nu trebuie trimisa cerere pentru procesare catre server.

### 4.2.5 Limbajul Java

**Java** este un limbaj de programare orientat pe obiecte dezvoltat de **Sun Microsystems** in anul 1995 pentru utilizarea in mediul distribuit heterogen al Internetului. Poate fi utilizat pentru a crea aplicatii ce pot rula pe un singur calculator sau sa fie distribuit printre mail multe servere si clienti intr-o retea. De asemenea poate fi folosit pentru a scrie un modul mic de aplicatie sau **applet** pentru a fi inclus intr-o pagina web. Appleturile permit interactiunea utilizatorului cu pagina.
Daca navigatorul gaseste un applet incorporat in codul HTML, acesta este transferat pe calculatorul utilizatorului si este executat local. Motivele utilizarii appleturilor:

- interactivitate cu utilizatorul (de ex. jocuri)
- formulare complexe (de tip calcul)
- adaugare animatie si sunet la paginile web fara a utiliza programe de vizualizare externe.

### 4.2.6 Macromedia Flash

Flash de la firma Macromedia este instrumentul de animatie **vectoriala** folosit pe web. Ofera posibilitatea crearii unor animatii de foarte buna calitate, scalabilitate, iar marimea este suficient de mica pentru a putea fi incarcata chiar si la viteze mai lente.

Exista doua tipuri de imagini utilizate pe web:

- imaginile **raster** constau din mici patratele (pixeli) de diverse culori. Calitatea imaginii depinde de rezolutie si imaginile par "sterse" cand se maresc. Cel mai bine sunt utilizate la fotografii si la imagini create in programe de manipulare a imaginilor ca Photoshop, Paint Shop Pro, CorelDraw.
- imaginile **vectoriale** constau din linii si curbe (numite vectori) descrise de ecuatii matematice. Astfel aceste imagini sunt independente de rezolutie si pot fi scalate si se pot muta, li se poate schimba culoarea usor fara a pierde din detalii, si vor fi afisate corect. De aceea sunt foarte utile in animatii.

## 4.3 Portal

**Portal** este termenul folosit pentru un site web care este sau isi propune sa fie un punct de pornire principal pentru utilizatori cand acestia se conecteaza la web, sau pe care utilizatorii il viziteaza ca pe un site de "baza".
In general serviciile oferite de un portal sunt:

- categorii de site-uri web, pe diverse domenii de interes
- posibilitatea de a face cautari pentru diverse informatii
- stiri, prognoza meteo
- informatii despre tranzactii de actiuni
- cautari de harti si numere telefon, forumuri etc

Un fapt interesant este aparitia primelor portaluri ce permit utilizatorilor sa-si individualizeze dupa preferinte interfata portalului cum ar fi culoarea fundalului, modul de aranjare a informatiilor pe site.
Exista portaluri generale cum ar fi Yahoo, Netscape, Microsoft Network, AOL, si portaluri pentru un anumit domeniu, de exemplu pentru investitori, administratori de retea (SearchNetworking.com) etc.
Multi provideri ofera portaluri pentru utilizatorii sai, majoritatea adoptand stilul de la Yahoo, site cu multe categorii usor de utilizat, in mare parte in mod text ce permite incarcare rapida pe care utilizatorii sa-l foloseasca cu usurinta si la care se intorc cu placere. Companiile cu portaluri au atras mereu investitorii de pe piata actiunilor, pentru ca aceste site-uri au foarte multi vizitatori astfel castigand o audienta mare.

Cateva pagini romanesti, care au fost create pentru a fi portaluri: **www.kappa.ro**, **www.lastart.ro**, **www.rol.ro**, **www.d-toate.net/portal**, **www.acasa.ro**, **www.apropo.ro**

## 4.4 Motoare de cautare

Pe web exista o mare varietate de informatie in diverse domenii pe milioane de site-uri. Problema este de a gasi informatia care il intereseaza pe utilizator, caz in care se apeleaza la un **motor de cautare (search engine)**. Acestea sunt site-uri speciale ce permit utilizatorilor sa gaseasca informatii de pe alte site-uri, pe baza unor cuvinete de cautare.
Modul de realizarea a acestora difera, dar in principiu realizeaza aceleasi functii:

- cauta informatii pe Internet
- mentin un index al cuvintelor gasite, si unde anume au fost gasite
- permit utilizatorilor sa caute cuvinte sau combinatii de cuvinte in acest index

Astazi un motor de cautare se refera la cautarea informatiei pe web, insa inainte ca web-ul sa devina un serviciu atat de popular existau alte servicii de cautare a informatiei pe Internet, cum ar fi gopher, archie, veronica etc.
La primele motoare de cautare indexul era format din cateva sute de mii de pagini, si primeau cateva mii de cereri pe zi. In ziua de azi un motor de cautare contine in index **sute de milioane de pagini** si raspund la **zeci de milioane cereri pe zi**.

Pentru a gasi informatie pe sutele de milioane de site-uri, motoarele de cautare folosesc programe speciale numite **paianjeni** (spiders) penru a crea liste de cuvinte gasite pe web.
Procesul de creare a acestor liste se numeste **web crawling**. Cautarea incepe la serverele cele mai utilizate si populare. Spider-ul incepe indexarea cuvintelor gasite pe un site popular si continua la toate link-urile gasite pe pagina respectiva. Prin acest sistem cautarea incepe sa se extinda pe partea cea mai utilizata a web-ului (figura 4.4).

De exemplu **Google** a fost creat ca un motor de cautare academic, si fondatorii lui au descris modul de functionare al sistemului, pentru a arata cat de eficient este acesta. Sistemul avea mai multi paianjeni simultani, in general 3, fiecare din ele avand deschise simultan 300 de conexiuni la pagini web. La performanta maxima, cu 4 paianjeni, sistemul era capabil sa parcurga peste 100 de pagini pe secunda generand aproximativ **600kiloocteti** de date pe secunda. La vizitarea unei pagini paianjenul nota 2 informatii: **cuvintele** din cadrul paginii si **unde au fost gasite** cuvintele. Paianjenul nota toate cuvintele semnificative de pe o pagina, ignorand articolele a (o), an (un) etc.
Alte spidere folosesc alte metode, in incercarea de a face sistemul cat mai rapid sau a oferi utilizatorilor o cautarea mai eficienta. De exemplu spider-ul de pe **Lycos** retine cuvintele din titluri, paragrafe si link-uri, impreuna cu cele mai folosite 100 de cuvinte si fiecare cuvant din primele 20 de linii ale textului. **Altavista** indexeaza toate cuvintele de pe o pagina, inclusiv articolele ca **a**, **an**, **the** si alte cuvinte "nesemnificative".

Tag-urile **META** din limbajul HTML permit celui care creaza pagina web sa specifice cuvintele pe baza carora sa fie indexata pagina respectiva. Exista si situatii cand se introduc cuvinte care

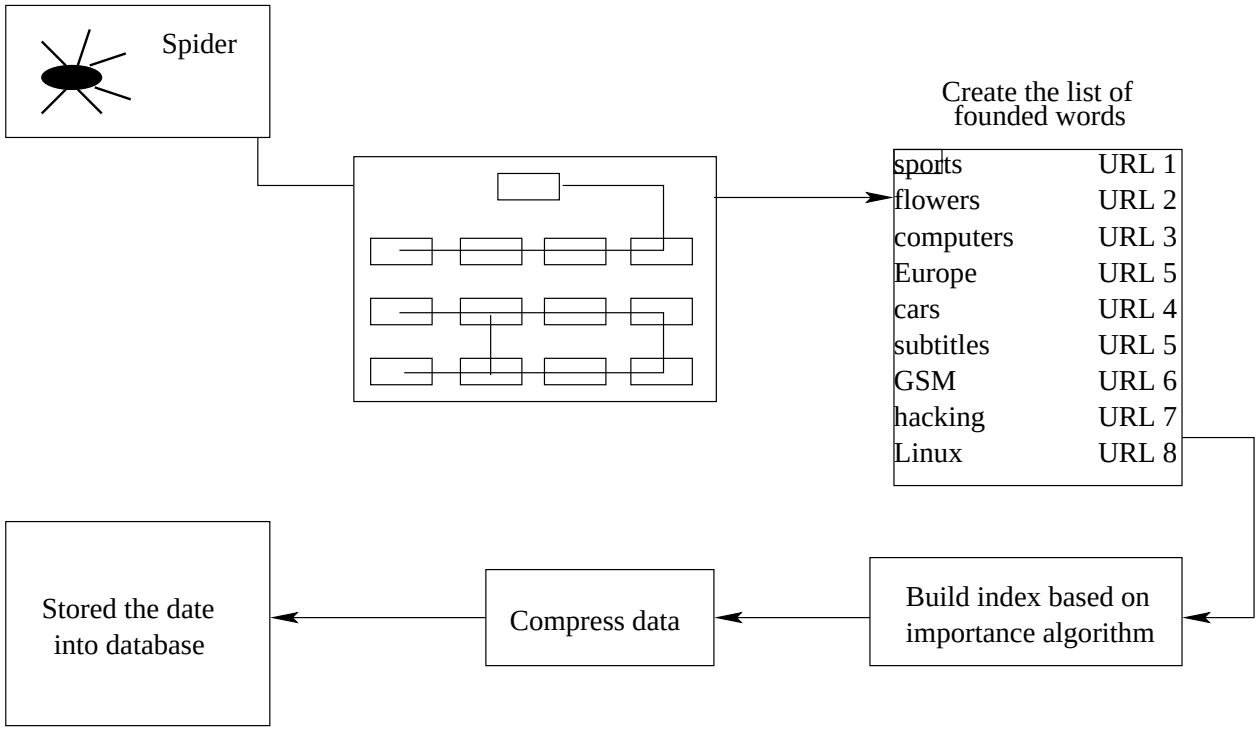


Figura 4.4: Modul de functionare a motorului de cautare

sa permita gasirea paginii pentru subiecte foarte populare, fara ca pagina sa contina ceva legat de subiectul respectiv. Pentru a elimina acest lucru paianjenii coreleaza cuvintele din tag-ul meta cu continutul paginii, rejectand cuvintele ce sunt in META tag, dar nu apar pe pagina.
Exista situatii cand detinatorul paginii nu doreste sa apara pe site-urile de cautare sau nu doreste ca activitatea spider-ului sa acceseze pagina. Pentru aceasta a fost dezvoltat **protocolul de excludere a robotului** (spider rejection protocol). Daca acesta este indicat la inceputul tag-ului META, care se afla la inceputul paginii HTML, spider-ul nu va indexa cuvintele de pe pagina si nici nu va urma link-urile de pe aceasta.
Dupa terminarea cautarii informatiei pe web (datorita modificarii constante a web-ului spiderele cauta in continuu informatie) serverul de cautare trebuie sa organizeze informatia astfel incat sa fie utila.
Exista doua componente ce fac datele cautate accesibile utilizatorilor:

- informatia in sine
- metoda prin care este indexata informatia

In cel mai simplu caz se poate stoca cuvantul gasit si URL-ul unde a fost gasit. In acest fel nu s-ar putea specifica daca cuvantul a fost folosit in sens important sau nerelevant, daca cuvantul a fost utilizat o singura data sau de mai multe ori, daca pagina continea link-uri catre alte pagini ce contin acel cuvant. In acest fel nu s-ar putea specifica **importanta rezultatului**,

daca una este mai semnificativa decat cealalta.

Astfel motorul de cautare memoreaza mai multa informatie, nu numai cuvantul gasit. Poate memora de cate ori apare cuvantul pe pagina. Cuvantului i se asociaza importanta, aceasta incrementandu-se daca apare in prima parte a documentului,in paragrafe, in link-uri, in titlul paginii sau in tag-ul META. Fiecare motor de cautare are o alta metoda de atribuire a importantei, din aceasta cauza la diferite servere de cautare acelasi cuvant poate produce rezultate diferite.

Pentru a minimiza spatiul alocat de aceasta informatie, toate datele se codifica. De exemplu la Google s-au utilizat 2 octeti pentru stocarea informatiei, importanta ocupand 2 sau 3 biti. Dupa compactarea informatiei acesta poate fi indexata, rolul acesteia fiind ca sa fie gasita cat mai repede posibil. Exista mai multe metode pentru a realiza acest lucru, cea mai folosita fiind atasarea unei valori numerice fiecarui cuvant (**hash table**). **Functia de dispersie** (hashing) permite distribuirea in mod egal a intrarilor pe un numar predeterminat de diviziuni (sectiuni). Tabela contine numarul si un indicator (pointer) la data respectiva. Datorita acestei indexari eficiente, se pot obtine rezultate rapide chiar la o cautare mai complexa ce consta din mai multe cuvinte.
Cautarea facuta de utilizator poate fi introducerea unui simplu cuvant, sau mai complexa in care se utilizeaza mai multe cuvinte si operatori booleani (AND sau +, OR, NOT sau -, "") pentru a restrange cautarea.

Cautarea realizata cu operatori booleani este o cautare literala, adica motorul cauta exact cuvintele sau fraza introdusa. Aceasta poate fi o problema daca cuvantul are mai multe intelesuri. Se pot elimina cele care nu sunt de interes, dar ar fi si mai simplu daca serverul ar putea efectua aceasta operatie. Una din directiile de dezvoltarea a motoarelor de cautare este cea de cautare pe baza de concept. Aceasta implica utilizarea unei analize statistice pe paginile ce contin cuvintele cautate de utilizator, pentru a gasi si alte pagini ce pot fi de interes. Bineinteles informatia stocata este mai mare in acest caz si procesarea in cazul unei cautari necesita mai mult timp.
Alta directie este dezvoltarea motoarelor de cautare bazate pe intrebari in **limbaj "natural"**. In acest caz utilizatorul introduce o intrebare exact ca si cum l-ar adresa unei persoane. Un asemenea server de cautare este **askjeeves.com**, care cauta cuvinte cheie in intrebare si le aplica indexului pe care il are.

## 4.5 Cookies

Ofera posibilitatea utilizatorilor de a naviga mai usor pe web. Cei care creaza pagini web le utilizeaza ca sa ofere utilizatorilor o navigare mai usoara si sa aiba o informatie clara despre vizitatorii unui site. **Cookie** este un sir de text pe care serverul Web il scrie pe hard-diskul utilizatorului. Permit serverului Web sa stocheze informatie pe calculatorul utilizatorului pe care sa-l citeasca ori de cate ori utilizatorul se conecteaza la site-ul respectiv. Informatia este

stocata sub forma unei perechi de **nume-valoare**. De exemplu un site Web poate genera un identificator (ID) unic pentru fiecare vizitator si sa memoreze acest numar pe calculatorul utilizatorului in fisierul cookie. Practic acesta este un fisier text care contine valorile salvate de fiecare site, in general acesta fiind un ID unic pentru acel utilizator. Serverul Web poate ulterior sa citeasca aceasta valoare, si poate citi numai aceasta valoare nu si alte valori scrise de alte servere Web. Pe sistemele Windows aceste date se inscriu ori in fisierul **cookies.txt** sau pentru fiecare cookie exista un fisier separat intr-un subdirector **Cookies** (Windows NT). Functionarea sistemului este astfel:

- la introducerea unui URL in browser acesta trimite cererea de pagina web la adresa respectiva
- cand browserul trimite cererea verifica pe hard-disk in fisierul cookie daca are informatii legate de site-ul respectiv si daca da atunci trimite si perechea nume-valoare odata cu URL-ul
- serverul Web primeste cererea pentru pagina impreuna cu cookie-ul, si citeste informatia stocata in baza de date la ID-ul primit prin cookie.
- daca nu se primeste perechea nume-valoare, atunci serverul stie ca utilizatorul respectiv nu a vizitat pagina web, Serverul creaza un nou ID in baza de date si trimite inapoi perechea nume-valoare in header-ul paginii pe care il transmite. Calculatorul memoreaza aceasta informatie pe hard-disk.
- serverul Web poate schimba perechea nume-valoare sau sa mai adauge valori ori de cate ori se viziteaza site-ul

Browser-ul poate fi setat sa accepte sau nu cookies-uri.
Utilitatea cookie-urilor:

- serverele pot determina pe baza acestei informatii **cati utilizatori viziteaza site-ul**. Astfel pot afla cati utilizatori acceseaza site-ul, cati sunt noi si cei care revin si cat de des revine un utilizator. Acest lucru este realizat prinintermediul unei baze de date. Cand un utilizator viziteaza site-ul pentru prima oara i se creaza un ID in baza de date si i se trimite acesta prin cookie. Data viitoare cand utilizatorul acceseaza site-ul, se va incrementa contorul asociat cu acel ID si se va sti de cate ori a vizitat acel utilizator pagina web
- site-urile **pot stoca preferintele utilizatorilor** astfel incat site-ul sa arate diferit pentru fiecare utilizator. Unele site-uri ofera posibilitatea schimbarii continut/dispunere/culoare. Permite introducerea introducerea codului postal pentru a primi prognoza meteo pentru zona respectiva
- **site-urile e-commerce** pot implementa cosurile de cumparaturi in care un utilizator poate selecta ce doreste sa cumpere si oricand poate "iesi din magazin" pentru ca s-au

memorat articolele selectate. Toate informatiile sunt stocate in baza de date a site-ului cu ID-ul vizitatorului, si ID-ul este trimis ca cookie pe calculatorul utilizatorului. La urmatoarea vizita pe baza ID-ului site-ul va sti ce anume a selectat un anumit utilizator

In toate exemplele de mai sus baza de date stocheaza lucruri selectate de utilizator, pagini vizitate, informatii date in formulare etc. Toate aceste informatii sunt stocate in baza de date a site-ului, si tot ce este stocat pe calculatorul utilizatorului este un cookie ce contine ID-ul.

Probleme cu cookies:

- mai multi utilizatori lucreaza de la acelasi calculator (sali de calculatoare, Internet Cafe)
- stergerea cookies-urilor de pe calculator
- utilizatorii pot lucra de la mai multe calculatoare (de acasa, de la servici)

Discutiile pe tema cookies-urile au provocat controverse in mass-media din urmatoarele motive:

- site-ul web poate urmari nu numai articolele pe care le cumpara utilizatorul ci si paginile pe care le citeste, reclamele pe care le viziteaza. In momentul cumpararii utilizatorul introduce date personale cum ar fi numele, adresa etc. Astfel site-ul are foarte multe informatii despre utilizator. In functie de politica site-ului (privacy policy) care difera de la site la site, acesta poate sa transmita informatiile la o a treia parte care va trimite reclame prin mail, utilizatorul primind in casuta postala mailuri pe care nu le-a cerut si nu le doreste (junk mail).
- exista site-uri care pot crea cookie-uri pentru clientii lor pe calculatorul utilizatorului, care apoi pot aduna informatii despre ce site-uri viziteaza utilizatorul, ce preferinte are si astfel sa-i trimita reclame legate de interesele lui. Un asemenea serviciu este DoubleClick, serviciile lor fiind folosite de foarte multe site-uri, care pe pagina lor principala pun o cerere de cookie catre DoubleClick. Pe baza acestui cookie Doubleclick poate urmari utilizatorul pe mai multe site-uri, ce cautari face pe motoarele de cautare. Cand utilizatorul acceseaza un asemenea site, serverul va cere cookie-ul, dupa care trimite ID-ul la Doubleclick cerand toata informatia marketing despre utilizator. Pe baza acestor informatii utilizatorului ii apar pe pagina reclame legate de interesele lui.
  Ingrijorarea este legata de faptul ca aceasta culegere de informatii se intampla transparent fara stirea si acordul utilizatorului.