2016 - 2017

Programarea Clientului Web

conf dr. ing. Simona Caraiman

asist drd.inf. Tiberius Dumitriu

mailto: tiberiusdumitriu@yahoo.com

Universitatea Tehnica "Gheorghe Asachi" din Iasi
Facultatea de Automatica si Calculatoare

Curs 1-2 - 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
  - URI, DNS, HTTP
- dezvoltarea unui site Web
  - responsabilitati client/server
  - web stacks – tehnologii client/server

Curs 3+ - preview

Web programming - client-side

- Browser-ul Web
  - arhitectura, exemple, extensibilitate
- tehnologii
  - HTML
  - CSS
  - Javascript
  - XML
  - AJAX
  - Web Workers

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)

Browser Web

Aplicatie software ce permite utilizatorilor sa localizeze si sa vizualizeze resurse Web

browser Web = client Web

Browser Web

In ordinea aparitiei (cele mai notabile):

- WorldWideWeb, February 26, 1991
- Mosaic, April 22, 1993
- Netscape Navigator and Netscape Communicator, October 13, 1994
- Internet Explorer (August 1995)
- Opera, 1996
- Mozilla Navigator, June 5, 2002
- Safari, January 7, 2003
- Mozilla Firefox, November 9, 2004
- Google Chrome, September 2, 2008
- Microsoft Edge, July 2015

Browser Web

Browsere specializate:

- Flock (social networking, blogging, photosharing, RSS newsreading)
- Image Xplorer (vizualizare, downloadare si listare imagini)
- Kirix Strata (data analytics)
- Songbird (browser cu capabilitati de audio streaming si built in media player)
- SpaceTime (search the web in 3D)
- Wyzo (media browser ce integreaza BitTorrent)
- Zac Browser (pentru copii cu autism)

Browser Web

Browsere pentru dispozitive mobile:

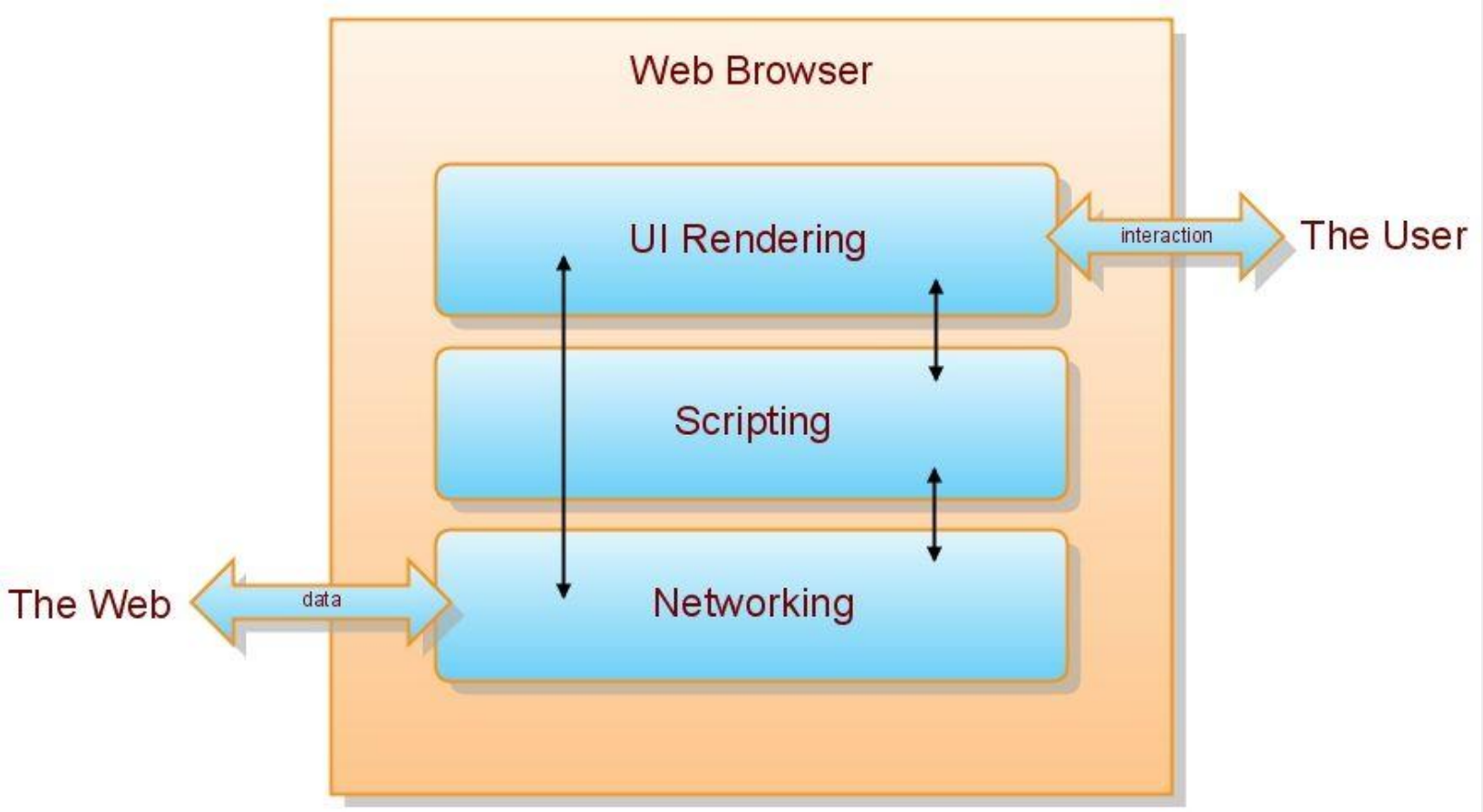
- Opera Mobile,
- Minimo,
- BlackBerryBrowser,
- Internet Explorer Mobile,
- Dolphin Browser
- Skyfire – faciliteaza accesul la Facebook
- Ninesky – grad de securitate sporit
- Boat Browser
- Maxthon
- AOSP – viteza mare de incarcare a datelor

Arhitectura browser-ului web

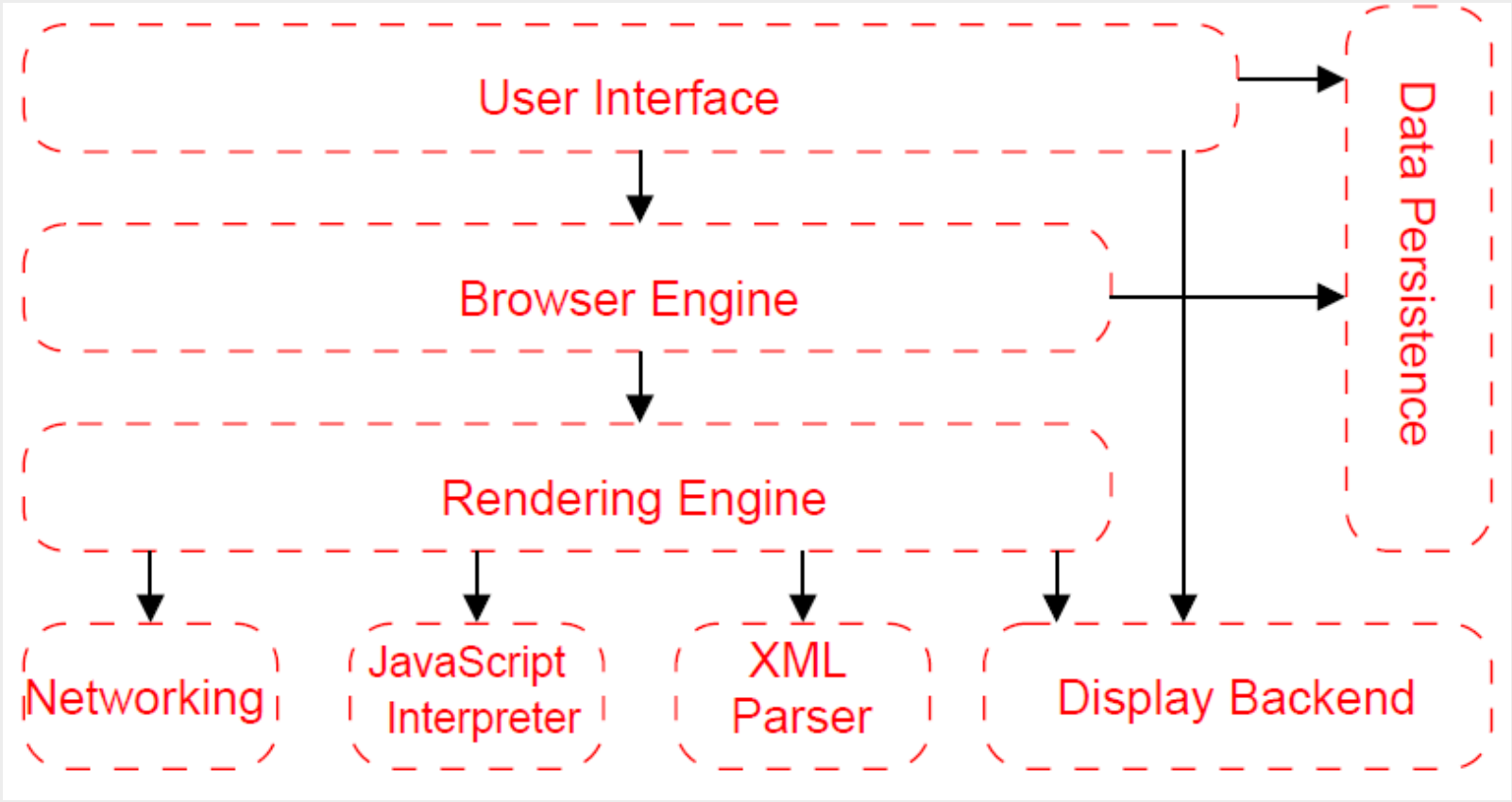
Un browser Web trebuie sa fie capabil sa:

- acceseze informatii de la diferite adrese pe Web
- interpreteze acele informatii
- transforme informatiile intr-o prezentare atractiva vizual: interfata utilizator
- interpreteze script-urile asociate si evenimentele generate de utilizator precum si alte evenimente din contextul interfetei cu utilizatorul
- trimita informatii inapoi la adresele web specificate

Arhitectura generala a browser-ului



Browser Web – arhitectura de referinta



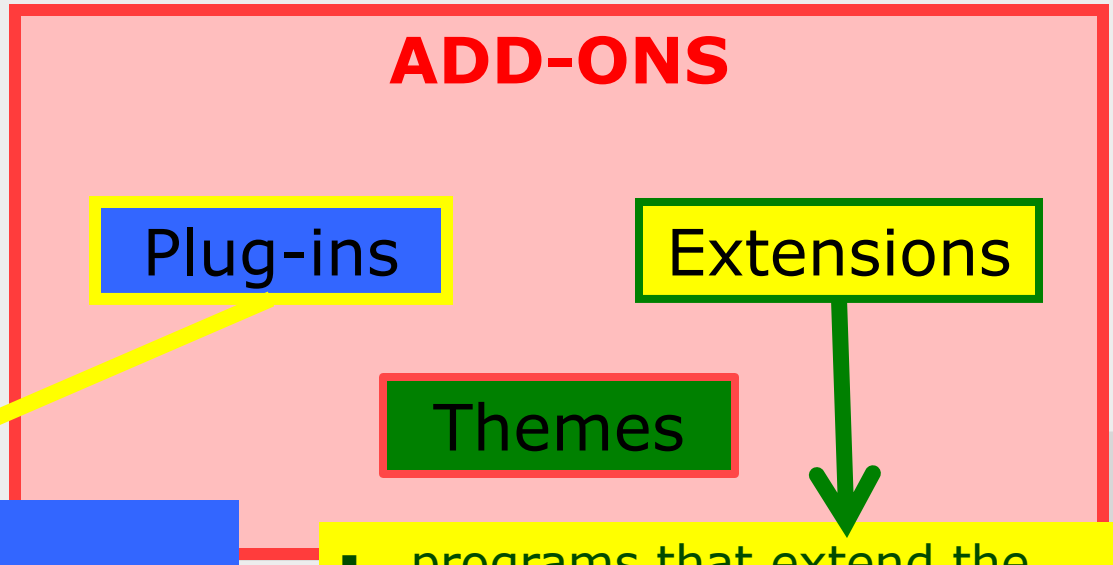
User Interface

- *Layer intre utilizator si motorul browser-ului*
- toolbar (address bar, back/forward, bookmarks menu, hystory, reading list, …)
- vizualizare progres incarcare pagini,
- preferinte
- printare
- …

Browser Engine

- Interfata de nivel inalt intre UI si motorul de randare

- nucleu (*kernel*)
- *plug-ins*
- *extensions*
- *add-ons*



- third-party libraries
- add functionality unavailable natively in the browser
- only affect the specific page in which they are placed
- ex.: Flash, Silverlight, Quicktime, Adobe Reader

- programs that extend the functionality of the browser
- they affect the browser itself + the pages
- Ex.: Firebug, Google Voice

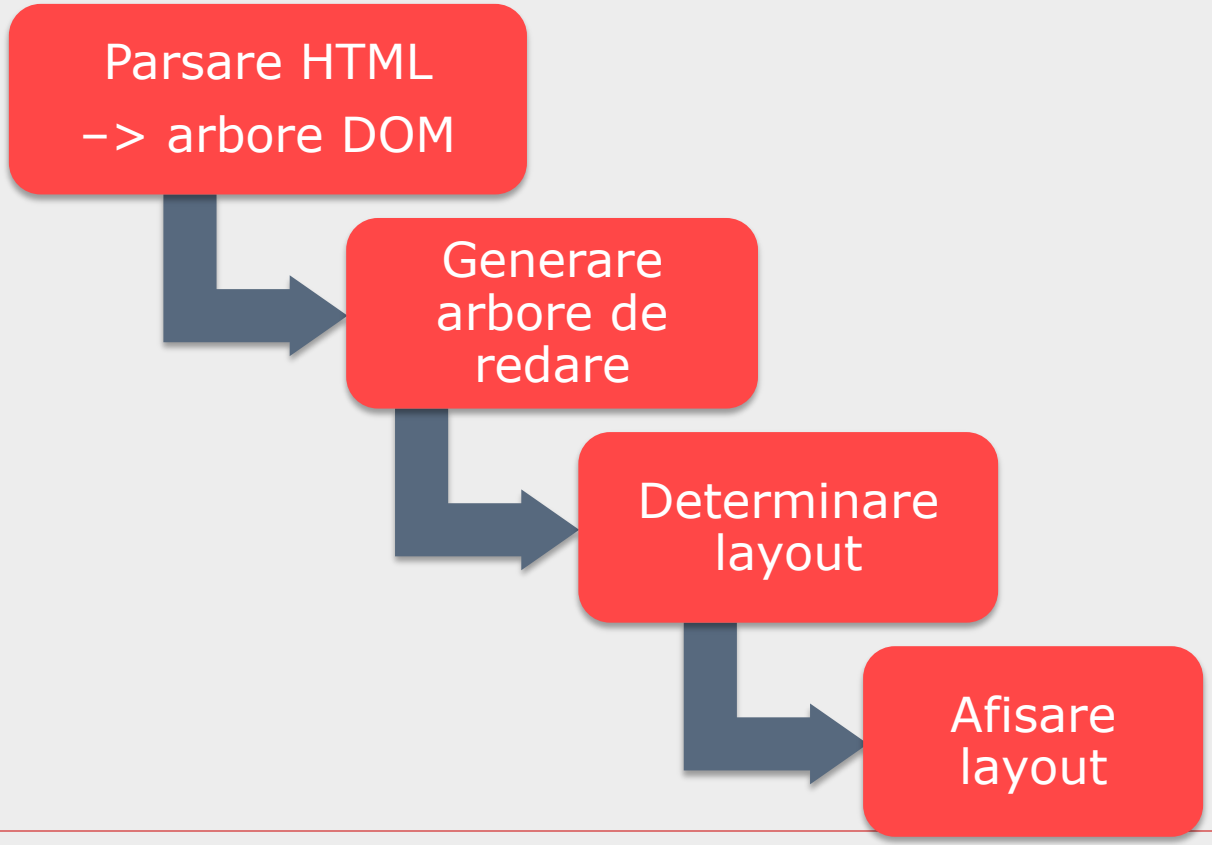
ent-s
ser

Rendering Engine

- Produce o reprezentare vizuala pentru un URI dat (redarea continutului solicitat)
  - default: HTML, XML, imagini
  - alte tipuri: plug-in (ex. PDF)
- Exemple:
  - Webkit (Safari, Chrome, Android, Blackberry, iOS)
  - Gecko (Firefox)
  - Presto (Opera, Opera Mobile, Opera Mini)
  - Trident (IE, IE Mobile)
  - EdgeHTML (Edge)

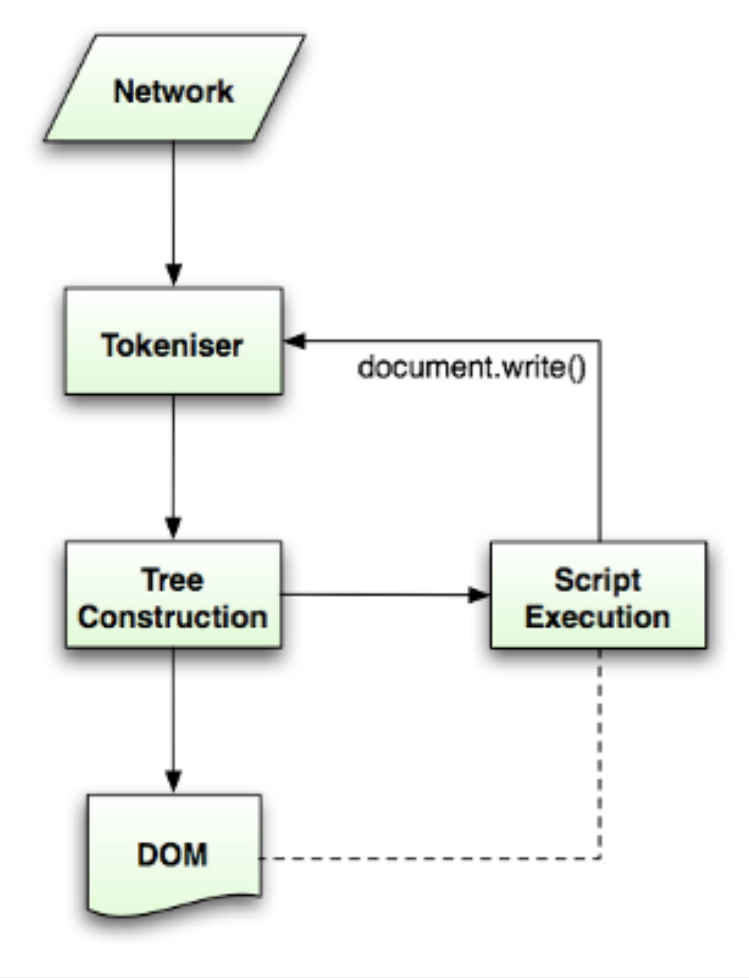
Rendering Engine

- **Basic flow**:



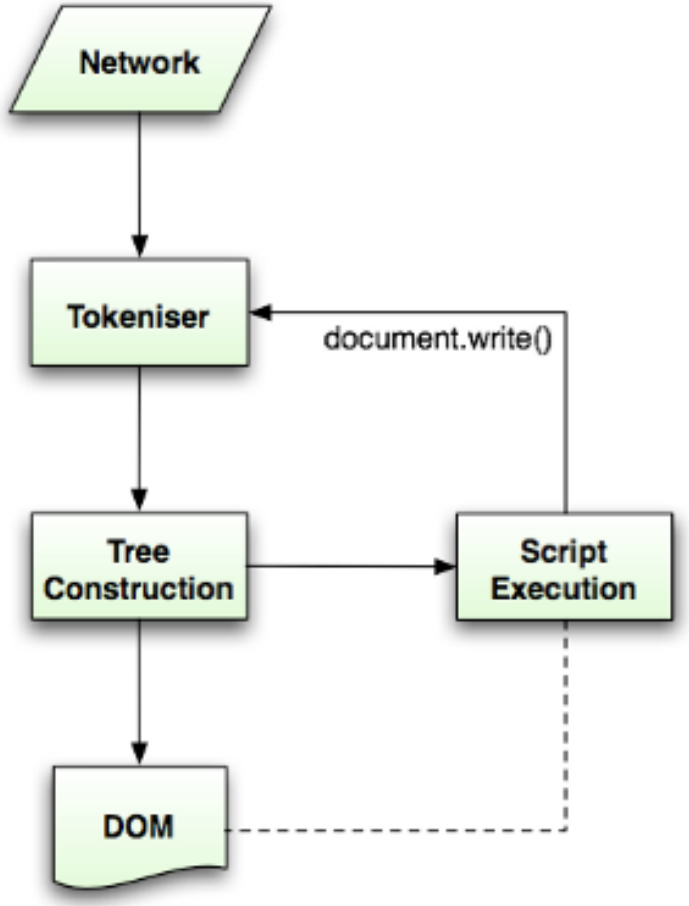
Rendering Engine

Parsare HTML
–> arbore DOM



(HTML5 spec.)

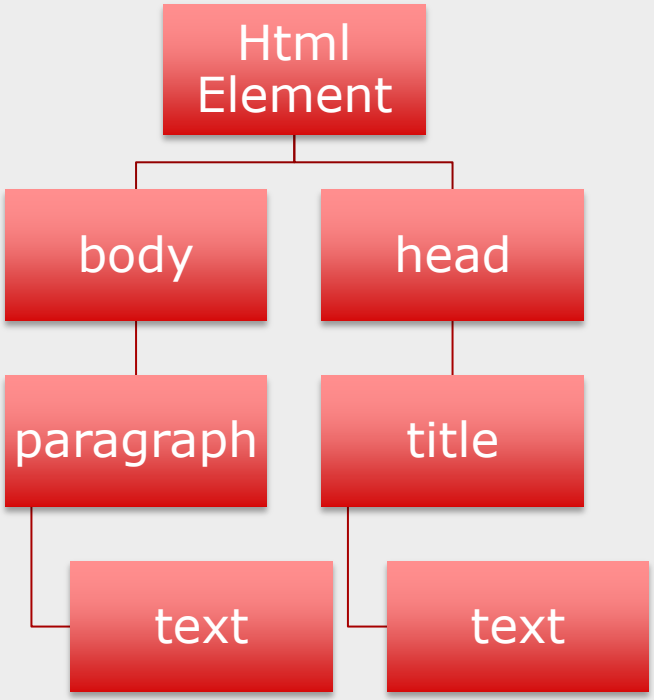
Rendering Engine



- **Tokenizare**
  - analiza lexicala
  - parsarea input-ului in token-uri
    - tag-uri de start/end, numele si valorile atributelor

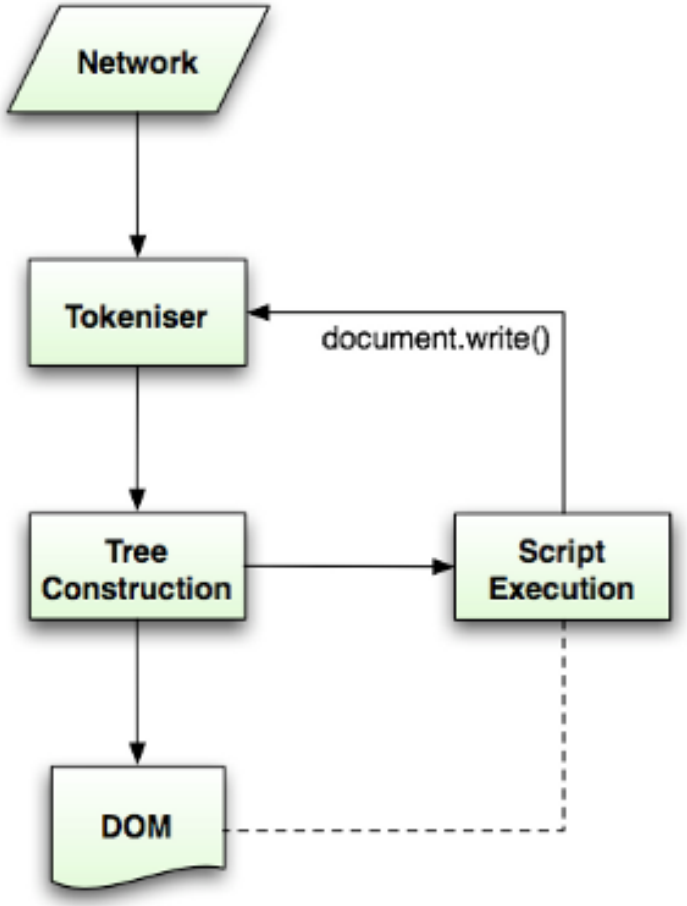
Rendering Engine

- **Constructia arborelui DOM**




```
<!DOCTYPE html>
<html>
 <head>
        <title>Example</title>
 </head>
 <body>
   <p>Hello, world!</p> </body>
</html>
```

Rendering Engine



- **Constructia arborelui**
  - la instantierea parserului se creeaza obiectul Document
  - elementul curent (produs de tokeniser)
    - adaugat in arbore
    - adaugat intr-o stiva de elemente deschise
      - corectarea nepotrivirilor si a tag-urilor ne-inchise

Rendering Engine

Parsare HTML
–> arbore DOM

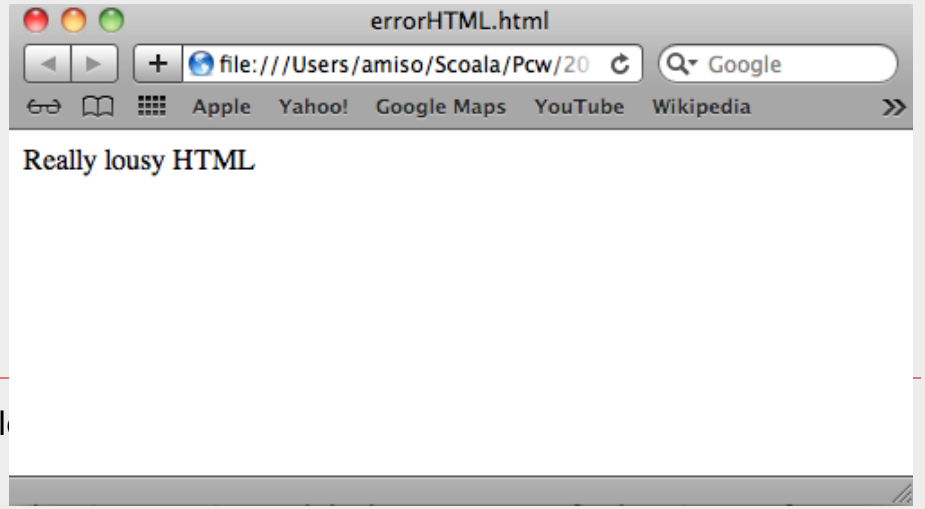
- La terminarea parsarii:
  - documentul este marcat ca fiind '*interactiv*'
  - se starteaza parsarea script-urilor in modul '*deferred*'
  - starea documentului este setata '*complete*'
  - se lanseaza un eveniment '*load*'

Rendering Engine

Parsare HTML
–> arbore DOM

- **Toleranta la erori**
  - toleranta la erori nu face parte din specificatiile HTML
    - HTML5 defineste unele cerinte
  - in paginile HTML NU se emit erori de tip "*Invalid Syntax*"

```
<html>
  <mytag>
  </mytag>
  <div>
  <p>
  </div>
            Really lousy HTML
  </p>
</html>
```



Rendering Engine

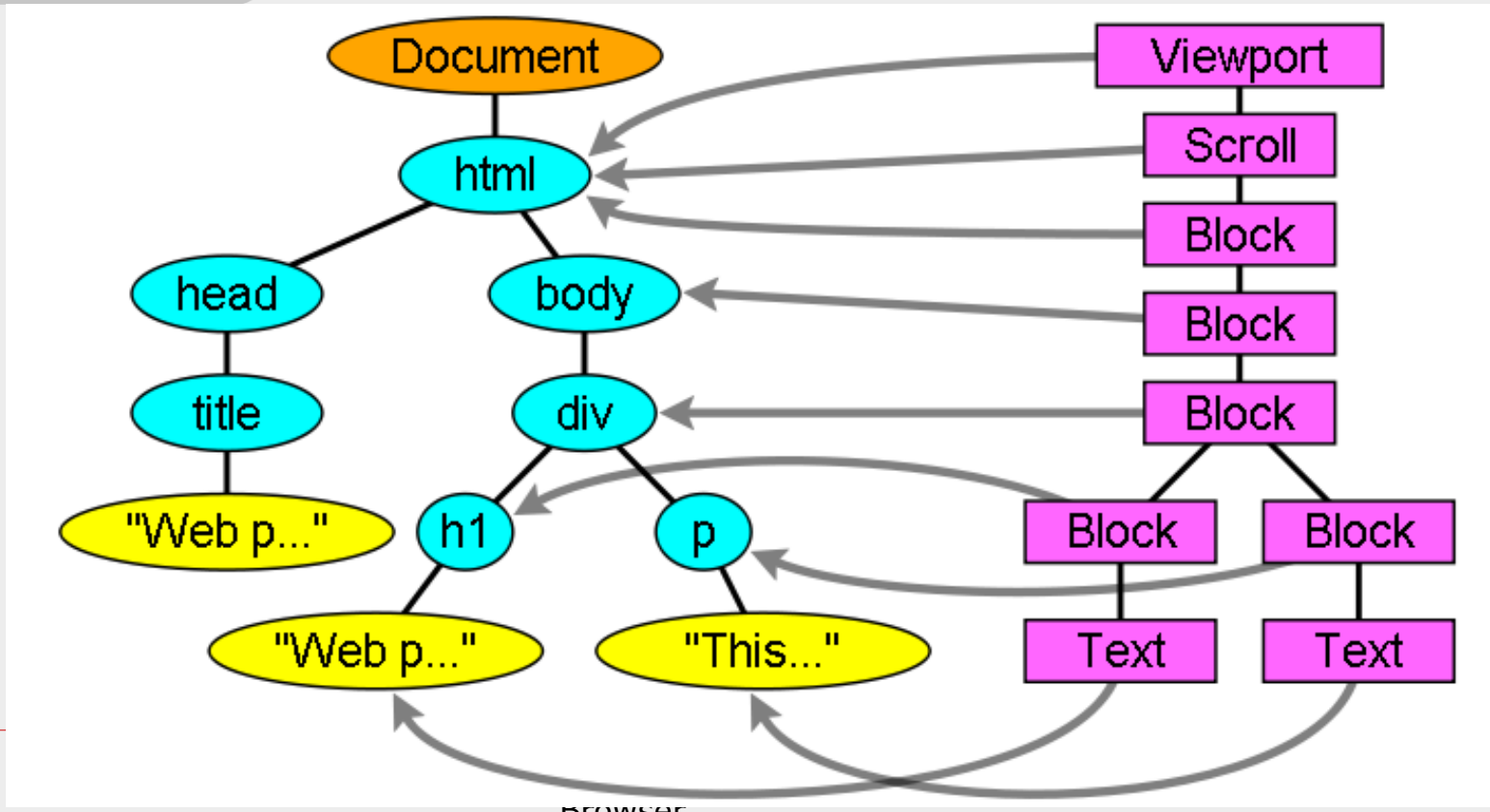
Generare arbore de redare

- Contine elementele vizuale in ordinea in care vor fi afisate
  - *frames* (Firefox), *renderers* (Webkit)
  - obiectele renderer corespund elementelor DOM (relatia nu este 1-la-1)
    - nu sunt introduse in arborele de redare elementele non-vizuale (head), elementele cu atributul display: none
    - elementele cu atributul visibility: hidden vor aparea in arbore

Rendering Engine

Generare arbore de redare

Relatia dintre arborele DOM si arborele de redare



Rendering Engine

Generare layout

- aranjamentul vizual al elementelor arborelui de redare
  - dependent de zona de afisare (uzual, un tab al browserului)
- se calculeaza pozitiile si dimensiunile elementelor

Rendering Engine

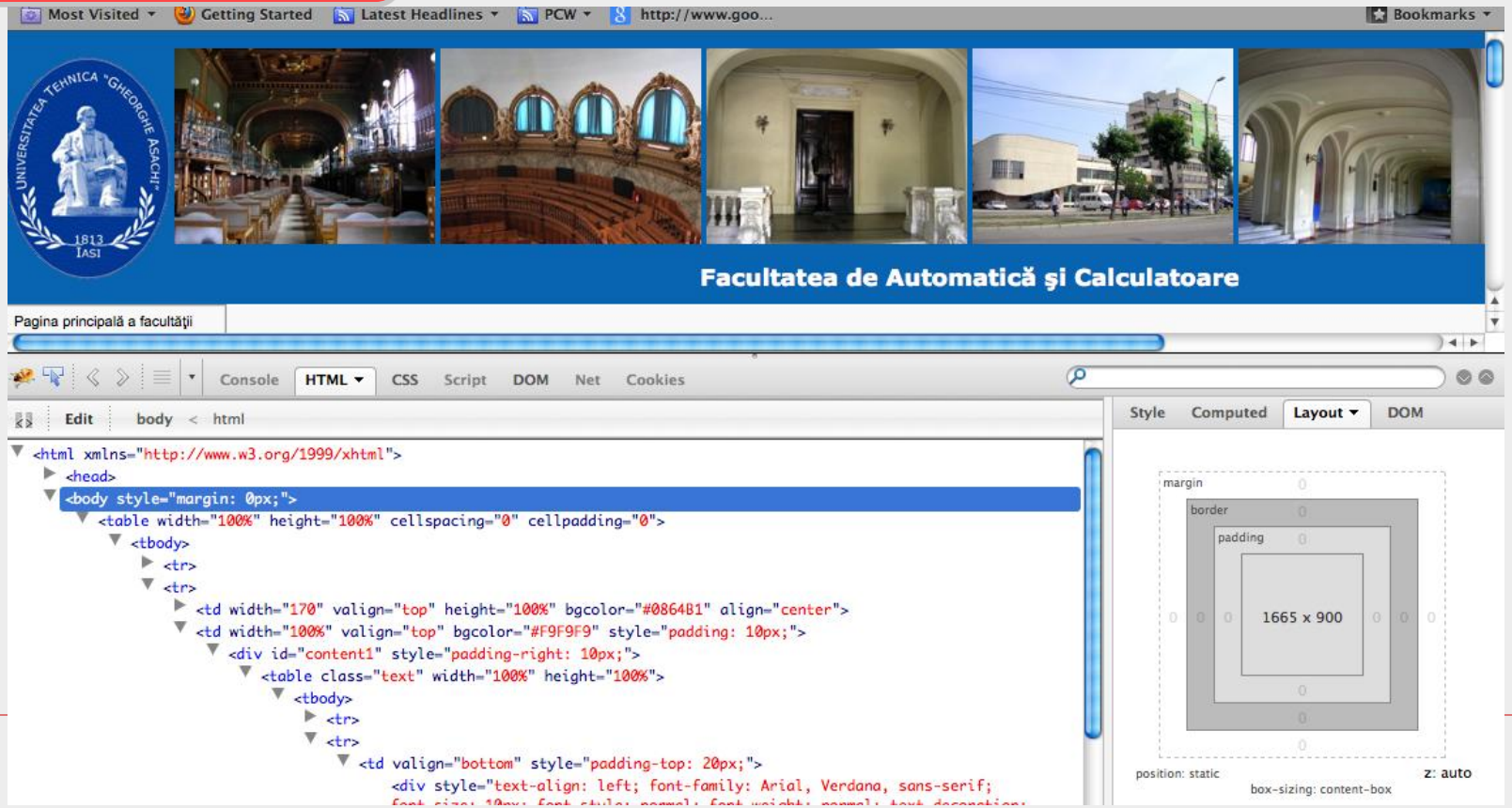
Afisare layout

- Aranjamentul vizual al elementelor arborelui de redare
  - dependent de zona de afisare (uzual, un tab al browserului)
  - proces recursiv
  - toate obiectele au o metoda “layout” (“reflow”)
- Layout
  - global vs. incremental
  - evitarea unui layout global in cazul modificarilor minore
    - systemul “dirty bit”
  - sincron vs. asincorn

Rendering Engine

Afisare layout

Firebug

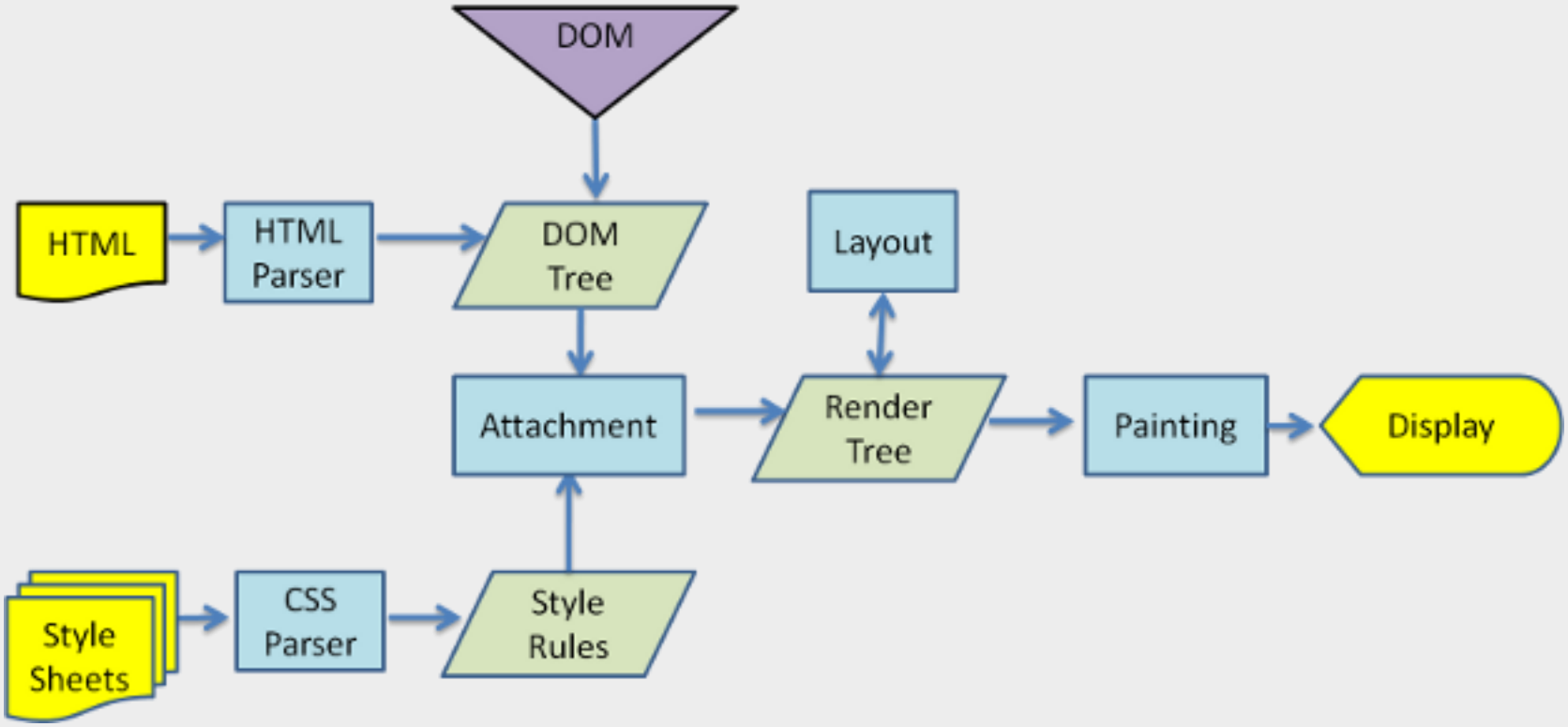


Rendering Engine

- modelul de procesare este sincron, *single-threaded*
- stilurile CSS nu modifica arborele DOM
  - procesarea poate avea loc independent de incarcarea fisierelor CSS
- programele JavaScript sunt executate imediat ce procesorul intalneste codul
  - implicit, procesul de *rendering* e oprit pana ce codul JavaScript este executat complet
  - HTML5
    - script-urile JavaScript pot fi executate asincron (intr-un thread separat)

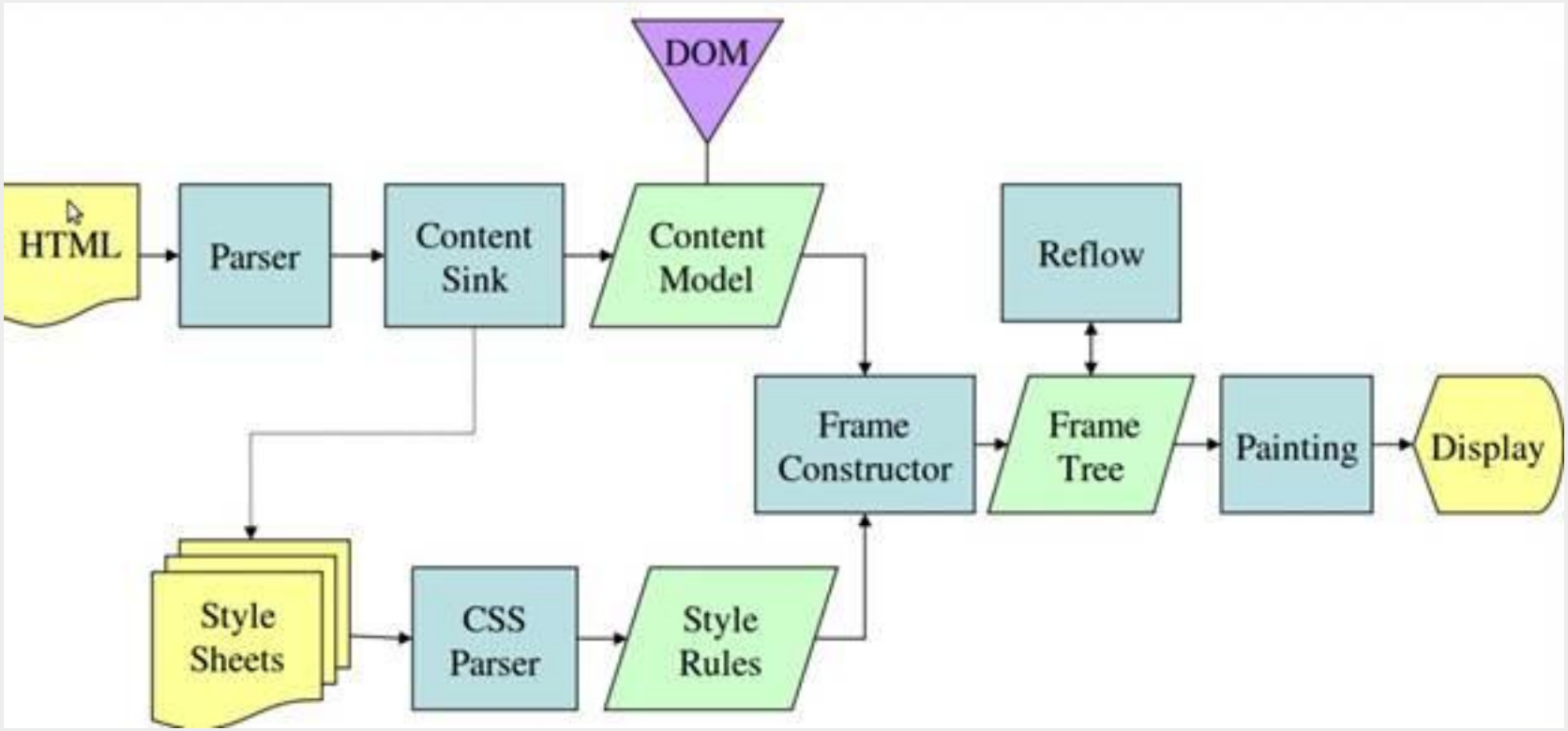
Rendering Engine

- Rendering workflow - Webkit



Rendering Engine

- Rendering workflow - Gecko



Display Backend

- afiseaza componentele de interfata
  - ferestre, scroll-bars, tipuri de campuri ale formularelor Web (butoane radio, textarea, liste de selectie)
- expune o interfata generica cross-platform
- utilizeaza metodele UI ale sistemului de operare

Networking

- responsabil cu transferurile de date de pe Internet (HTTP/HTTPS/SPDY)
- API independent de platforma
  - implementari specifice fiecarui sistem de operare

- Implementeaza protocoale de transfer de fisiere (HTTP, FTP)
- Realizeaza translarea intre diferite seturi de caractere
- Rezolva tipul MIME pentru fisiere
- Poate implementa un cache al resurselor recente

- număřul conexiunilor HTTP paralele este limitat (2-6)

Javascript Interpreter

- interpreteaza si executa programele JavaScript
- Ex.:
  - SpiderMonkey (Mozilla)
  - V8 (Chrome)
  - JavaScriptCore (Safari)
  - Carakan (Opera)
  - JScript (IE<9), Chakra (IE9)

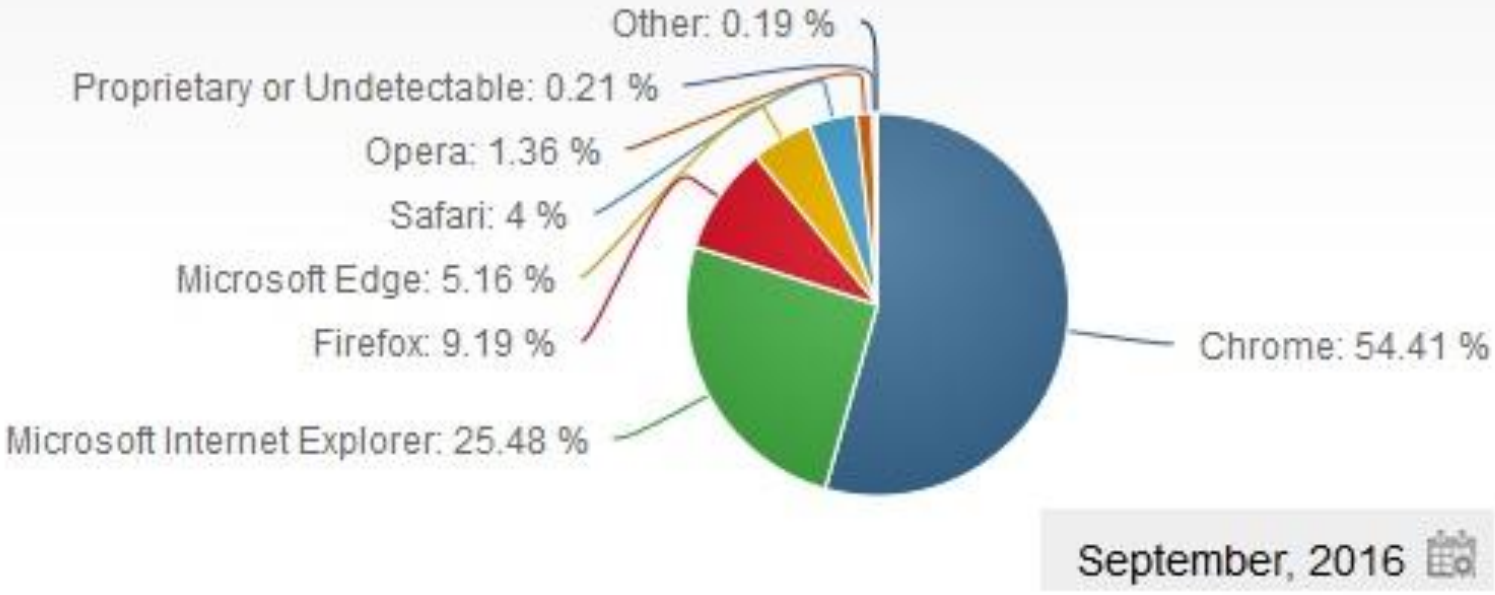
XML Parser

- procesarea documentelor XML intr-un arbore DOM
  - validarea datelor – via DTD
  - uzual, suport pt.
    - XML namespaces
    - XPath
    - XSLT
    - transferuri asincrone de date XML via XMLHttpRequest
    - alte limbaje XML (ex. MatML, SVG)
- majoritatea browserelor folosesc un parser existent
  - Ex.:
    - Expat (Mozilla, Safari)
    - MSXML (IE)

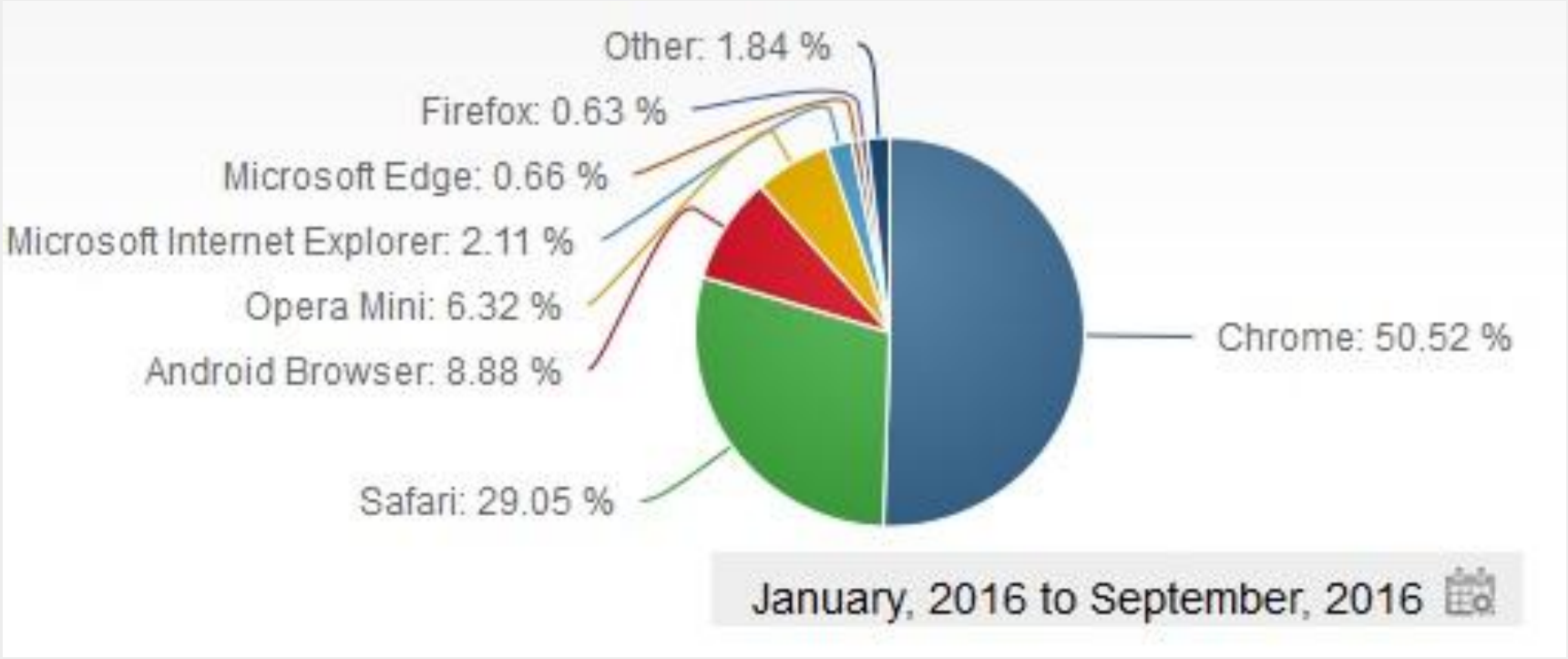
Data Persistence

- stocarea pe disc a diferitelor informatii asociate cu sesiunea de browsing
  - date de nivel inalt (*bookmarks*, *toolbar settings*)
  - date de nivel coborat (*cookies*, *certificate de securitate*, *cache*)
  - local storage (HTML5 – 'web database')

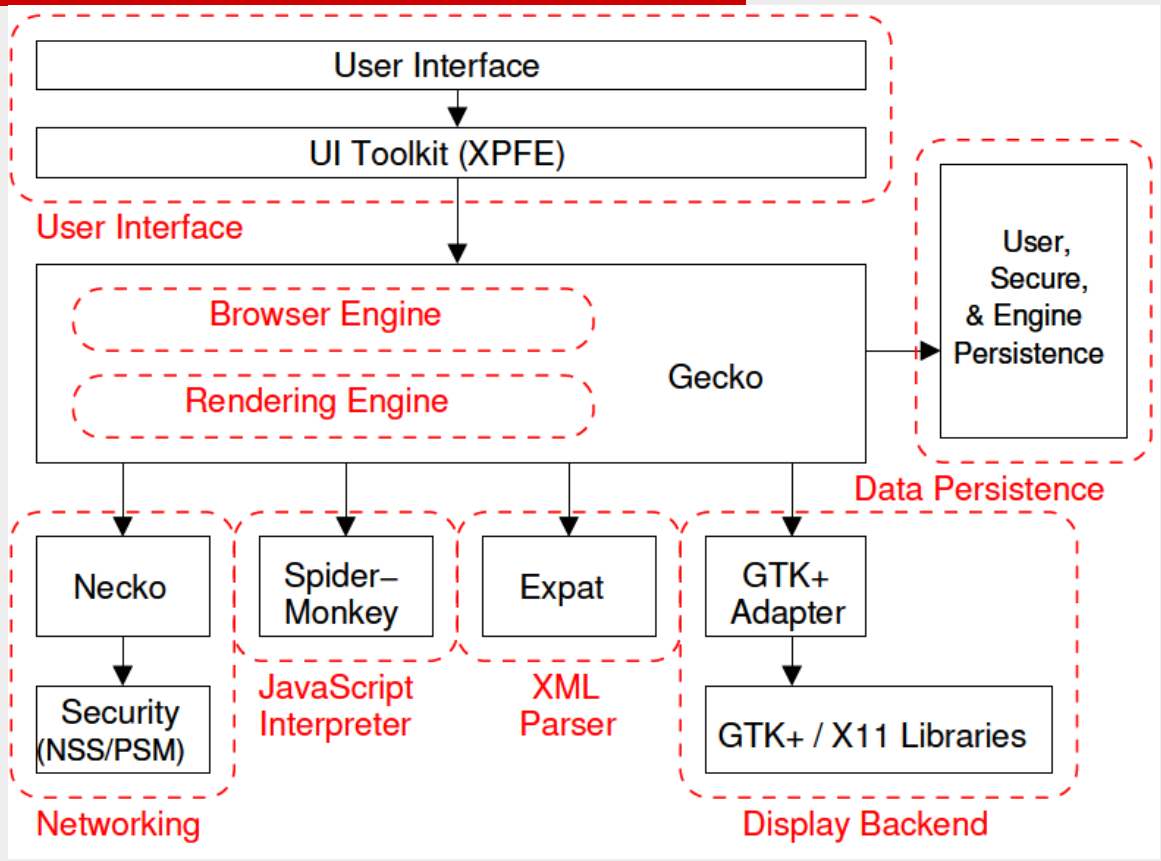
Web Browser


Other: 0.19 %
Proprietary or Undetectable: 0.21 %
Opera: 1.36 %
Safari: 4 %
Microsoft Edge: 5.16 %
Firefox: 9.19 %
Microsoft Internet Explorer: 25.48 %
Chrome: 54.41 %
September, 2016

Web Browser


Other: 1.84 %
Firefox: 0.63 %
Microsoft Edge: 0.66 %
Microsoft Internet Explorer: 2.11 %
Opera Mini: 6.32 %
Android Browser: 8.88 %
Safari: 29.05 %
Chrome: 50.52 %
January, 2016 to September, 2016

Browser Web - Firefox



Arhitectura Mozilla - http://web.uvic.ca/~hitchner/assign1.pdf

Browser Web - Firefox

- Versiune curenta 41.0.2

- Standarde: HTML4 (partial HTML5), XML, XHTML, SVG 1.1 (partial), CSS (cu extensii), ECMAScript (JavaScript), DOM, MathML, DTD, XSLT, XPath, si imagini PNG (animated) cu transparenta alpha, client-side storage, elementul *canvas*.
- 4 componente de baza:
  - **XUL (*XML User Interface Language*)** – dialect XML utilizat pentru construirea interfetelor utilizator
  - **JavaScript**
  - **RDF (*Resource Description Framework*)** – dialect XML pentru stocarea datelor
  - **XPCOM (*Cross Platform Component Object Model*)** – sistem de descoperire si administrare a obiectelor

Browser Web - Firefox

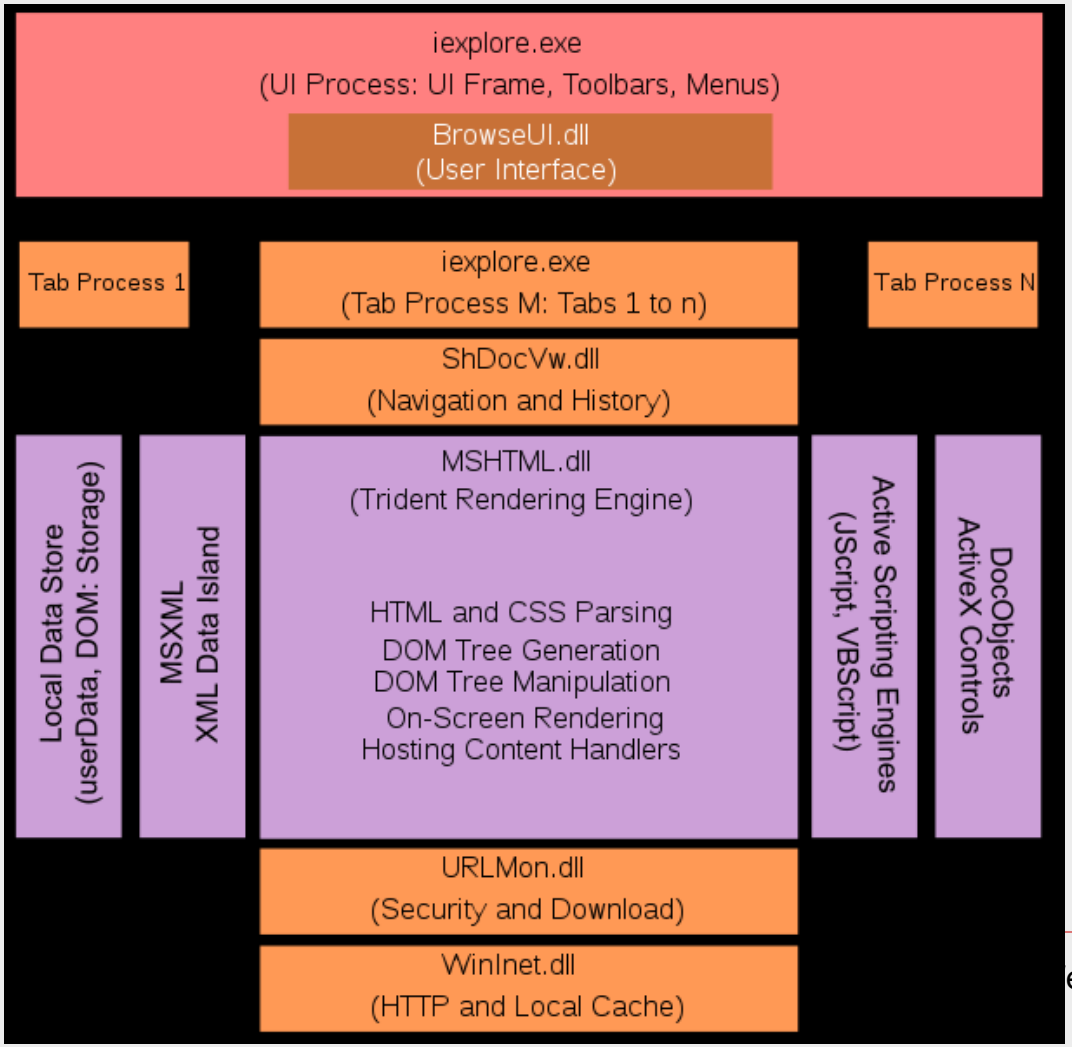
- User Interface
  - 2 subsisteme permitand reutilizarea in alte aplicatii din suita Mozilla (client mail)
- Rendering Engine
  - Desenarea interfetei cu utlizatorul si a continutului Web
  - Continutul este afisat pe baza unui set de structuri de date independente de platforma (primitive geometrice, definitii de culoare, specificatii pentru font-uri, …)
  - Foloseste *placeholders* pentru continutul anticipat; sunt inlocuite cu continut cand acesta devine disponibil => randare rapida a paginilor
- Data Persistance
  - 2 sub-componente
  - Browser persistance
    - mozStorage – database API peste sqlite
    - Setari pentru extensii si componente Firefox
  - User data
    - Cookies, DOM Storage, Flash Local Storage

Browser Web - Firefox

- Parser XML
  - Foloseste un parser existent – Expat
  - Cea mai reutilizabila componenta din arhitectura
  - parsare XML, XUL, SVG si a altor dialecte XML
  - suport si pentru XHTML, SVG, MathML and XLink

- Display Backend
  - Strans cuplat cu SO
  - Furnizeaza o interfata cross-platform utilizata de componenta User Interface pentru functionalitatea de desenare si management ferestre a diferitelor SO suportate de Firefox
  - 2 sub-componente:
    - Adaptor grafic specific SO
    - Librarii grafice SO
      - Linux: librariile GTK+ /X11
      - Mac OSX: Cocoa framework
      - Windows: WINAPI, MFC

Browser Web - Internet Explorer



Arhitectura IE v8

Browser Web - Internet Explorer

- Foloseste o arhitectura bazata pe componente (COM – *Component Object Model*)

1. **`WinInet.dll`**
   - comunicatie in retea folosind protocoalele HTTP si FTP
2. **`URLMon.dll`**
   - manipularea tipurilor MIME si download-ul continutului Web
3. **`MSHTML.dll`**
   - gazduieste motorul de randare Trident (IE4) responsabil cu afisarea paginilor si manipularea DOM a paginilor Web
   - parseaza fisierul HTML/CSS si creeaza o reprezentare interna arborescenta DOM
   - furnizeaza API pentru inspectia la rulare si modificarea arborelui DOM
   - arborele DOM este procesat mai departe de un layout engine care afiseaza reprezentarea interna pe ecran

Browser Web - Internet Explorer

4. **IEFrame.dll**
   - contine interfata utilizator si fereastra IE (IE 7+)
5. **ShDocVw.dll**
   - functionalitati de navigare, cache local, istoric vizitare
6. **BrowseUI.dll**
   - responsabil cu interfata utilizator a browser-ului, incluzand browser chrome (meniuri, toolbar-uri)

Browser Web - Internet Explorer

- Extensibilitate
  - expune un set de interfete COM care permit extinderea functionalitatii browser-ului
  - *extensibilitate browser* (toolbars, menu items, BHO – Browser Helper Objects)
  - BHOs – access nerestrictionat la DOM si modelul de evenimente, la sistemul de fisiere, registri si alte componente OS
  - *extensibilitate continut* – adaugare suport pentru formate non-native de continut
    - ActiveX Controls – manipulatori de continut care afiseaza continutul incorporat in pagina HTML (Adobe Flash, MS Silverlight)
    - Active Documents – continutul nu este incorporat in HTML (MS Word, PDF, XPS)
  - componentele add-on ruleaza cu aceleasi privilegii ca si browser-ul, pot fi instalate local sau direct de un site Web => probleme de securitate

Browser Web – Microsoft Edge

- Windows 10 – 2015
- built around web standards
  - user agent string:

Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/42.0.2311.135 Safari/537.36 Edge/12.<OS Build #>

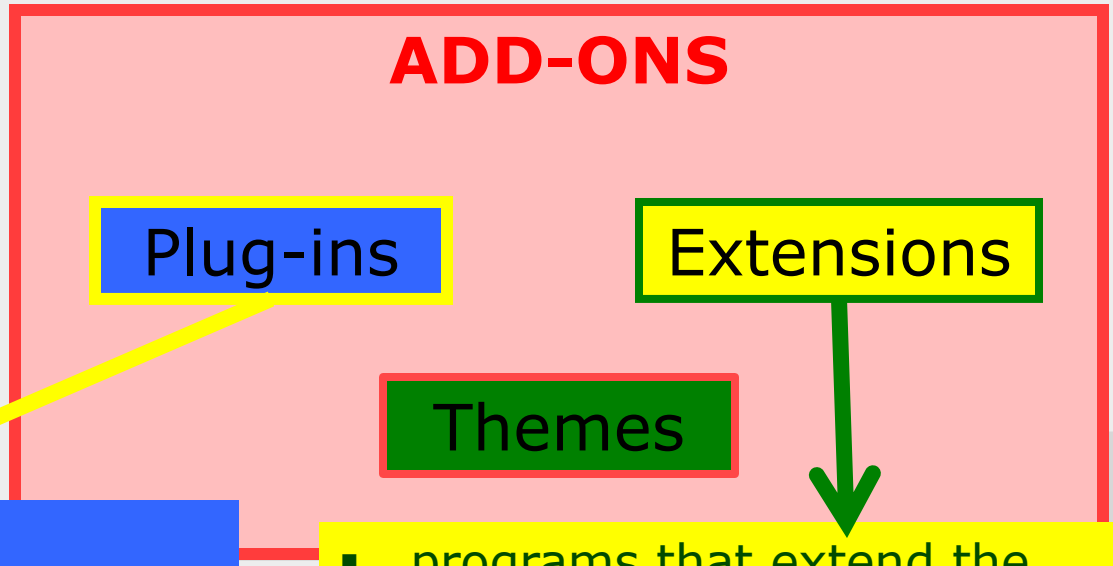
- support for asm.js
- integrates with Microsoft online platforms:
  - Cortana (digital assistant)
  - Web page annotations – stored and shared using OneDrive

Browser Web – Microsoft Edge

- Extensibility:
  - No support for third-party installable plug-ins
  - No support for legacy technologies:
    - ActiveX, BHOs
    - IE loaded automatically for sites that don't work without legacy technologies
  - integrated Flash player
  - integrated PDF Reader

Add-ons - plug-ins, extensii

Add-ons



- third-party libraries
- add functionality unavailable natively in the browser
- only affect the specific page in which they are placed
- ex.: Flash, Silverlight, Quicktime, Adobe Reader

- programs that extend the functionality of the browser
- they affect the browser itself + the pages
- Ex.: Firebug, Google Voice

Plug-ins vs. extensii

- **Plug-ins**
  - biblioteci shared
  - ajuta browser-ul sa afiseze continut specific (ex. fisiere multimedia)
  - sunt incluse (‘embedded’) intr-o pagina web
- **Extensii**
  - adauga noi functionalitati browser-ului
  - pot fi vazute ca parte integranta din browser
  - pot include un plug-in

Plug-ins vs. aplicatii helper

- **Plug-ins**
  - ruleaza intr-o pagina web
  - ruleaza instructiuni masina native cu aceleasi privilegii ca si procesul gazda
  - sunt ‘scriptable’ si pot accesa obiecte intr-un browser
- **Aplicatie helper**
  - aplicatie separata, de sine-statatoare ce poate fi lansata din browser
  - ruleaza separat de browser
  - nu interactioneaza cu browser-ul sau cu web-ul

Plug-ins

Plug-ins – ce pot face?

- register one or more MIME types
- draw into a part of a browser window
- receive keyboard and mouse events
- obtain data from the network using URLs
- post data to URLs
- add hyperlinks or hotspots that link to new URLs
- draw into sections on an HTML page
- communicate with JavaScript/DOM from native code

Plug-ins

- Plugin-uri instalate intr-un browser

  about:plugins

  - MIME type(s), description, file extensions, and the current state (enabled or disabled)
  - JavaScript

    navigator.mimeTypes

    enabledPlugin

Plug-ins - lifecycle

- controlat complet de pagina web care il apeleaza
  - cautarea unui plug-in pt. MIME type-ul ce trebuie tratat
  - incarcarea in memorie a codului plugin-ului
  - initializarea plugin-ului
  - crearea unei noi instante a plugin-ului
  - stergerea instantei plugin-ului (la parasirea paginii web)
  - stergerea din memorie a codului plugin-ului (cand ultima instanta a plugin-ului este distrusa)

Plug-ins - dezvoltare

- NPAPI (Netscape Plugin API)
  - Firefox, Chrome
- ActiveX
  - Internet Explorer

Plug-ins - dezvoltare

- NPAPI (Netscape Plugin API)
  - cross-platform plugin architecture
  - suportat de majoritatea browser-elor
  - impune ca fiecare plugin sa implementeze si sa expuna un set de functii pentru
    - Initializare, creare, distrugere, pozitionare
  - suporta:
    - scripting, printing, plugin-uri full screen, windowless, content streaming

Plug-ins - NPAPI

- functii specifice plugin-ului
  - prefixul “NPP”
- functii specifice browser-ului
  - prefixul “NPN”

- NP_Initialize
- NPP_New
- NPP_Destroy
- NP_Shutdown
- NPN_Write

Plug-ins - dezvoltare

Alternative pentru NPAPI

- Pepper API / PPAPI
- Google Native Client / NaCl
- Google Native Messaging
- Mozilla js-ctypes
- Web Sockets + external app

http://www.firebreath.org/display/documentation/FireBreath+2.0%3A+Browser+Plugins+in+a+post-NPAPI+world

Plug-ins

- Windowed
  - afisate in propria fereastra nativa in pagina web
  - opace
  - top HTML section
- Windowless
  - opace/transparente
  - pot fi invocate in sectiunile HTML

Plug-ins

- Default plug-in
  - invocat atunci cand nu exista un plug-in care sa trateze MIME type-ul referit in HTML
  - permite utilizatorilor sa gaseasca si sa instaleze un plugin pt MIME type-ul respectiv

Plug-ins in HTML

- Display modes:
  - *embedded* in a web page and *visible*
  - *embedded* in a web page and *hidden*
  - displayed as a full page in its *own window*
- <object>
- <embed>

```
<object data="newave.avi"
        type="video/avi" width="320"
height="200"
    autostart="true" loop="true">
</object>
```

```
<embed src="audiplay.aiff"
type="audio/x-aiff" hidden="true">
```

Plug-in free browsing

2015:

- Chrome – no support for plugins
- Edge – plugin-free

Plug-in free browsing

Plug-ins disadvantages:

- consume additional system resources
- expose additional attack surface to security risks
- not designed for touch
- don't benefit from any changes coming with newer browser versions
- difficult to predict/control their support across different browsers and operating systems
  - based on proprietary technologies
  - written with variable code quality

Plug-in free browsing

Plug-ins **alternative**:

- Standards-based technologies (i.e. HTML5)
  - **Avantaje**
    - *Consumers*:
      - better performing browsing experiences, longer battery life on mobile devices, and increased security, privacy, and reliability
    - *Developers*:
      - wider site interoperability across browsers and devices, better forward-compatibility, and overall easier site maintenance
    - *Modern Web:*
      - promoting and populating a more open, accessible, and vendor-neutral web

Plug-in free browsing

Plug-ins **alternative**:

- Standards-based technologies (i.e. HTML5)

Capability	Example of plug-in based technology	Standards-based equivalent technologies
Video and audio	Flash, Apple QuickTime, Silverlight	HTML5 video and audio
Graphics	Flash, Apple QuickTime, Silverlight, Java applets	HTML5 canvas , Scalable Vector Graphics (SVG), Cascading Style Sheets, Level 3 (CSS3) Transitions and Animations, CSS Transforms, WebGL
Offline storage	Flash, Java applets, Google Gears	Web storage, File API, IndexedDB, Application cache API
Network communication, resource sharing, file upload	Flash, Java applets	HTML5 Web Messaging, Cross-origin resource sharing (CORS)