

UD 1: Selecció d'arquitectures i eines de programació.

1. Evolució i característiques dels navegadors Web (I)

World Wide Web:

- Representa un univers d'informació accessible a través d'Internet.
- Conjunt de recursos interconnectats.

Components físics: hubs, repetidors, ponts, routers, ...

Protocols de comunicació: TCP, IP, HTTP, FTP, ...

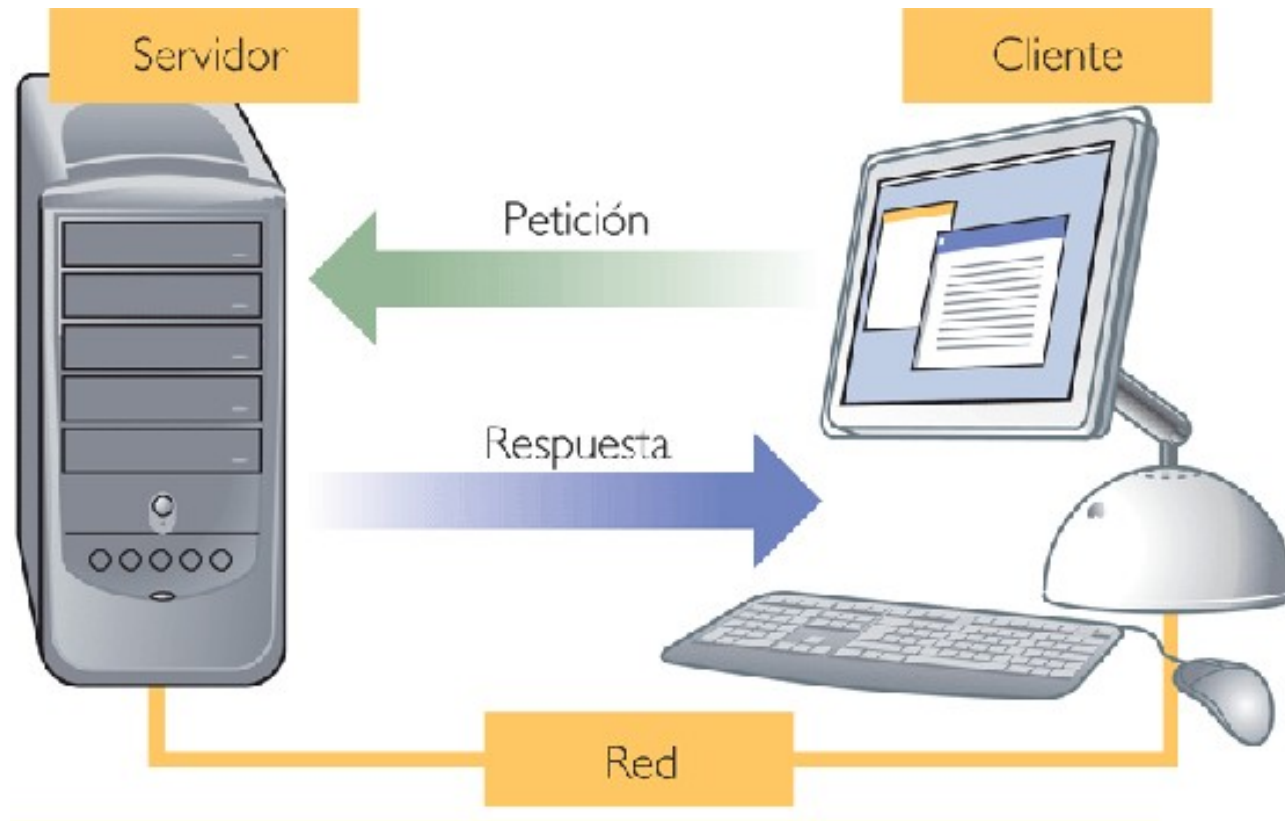
Programari per proveir i consumir aquests recursos: servidors i clients.

- Configuració arquitectònica habitual basada en client/servidor.

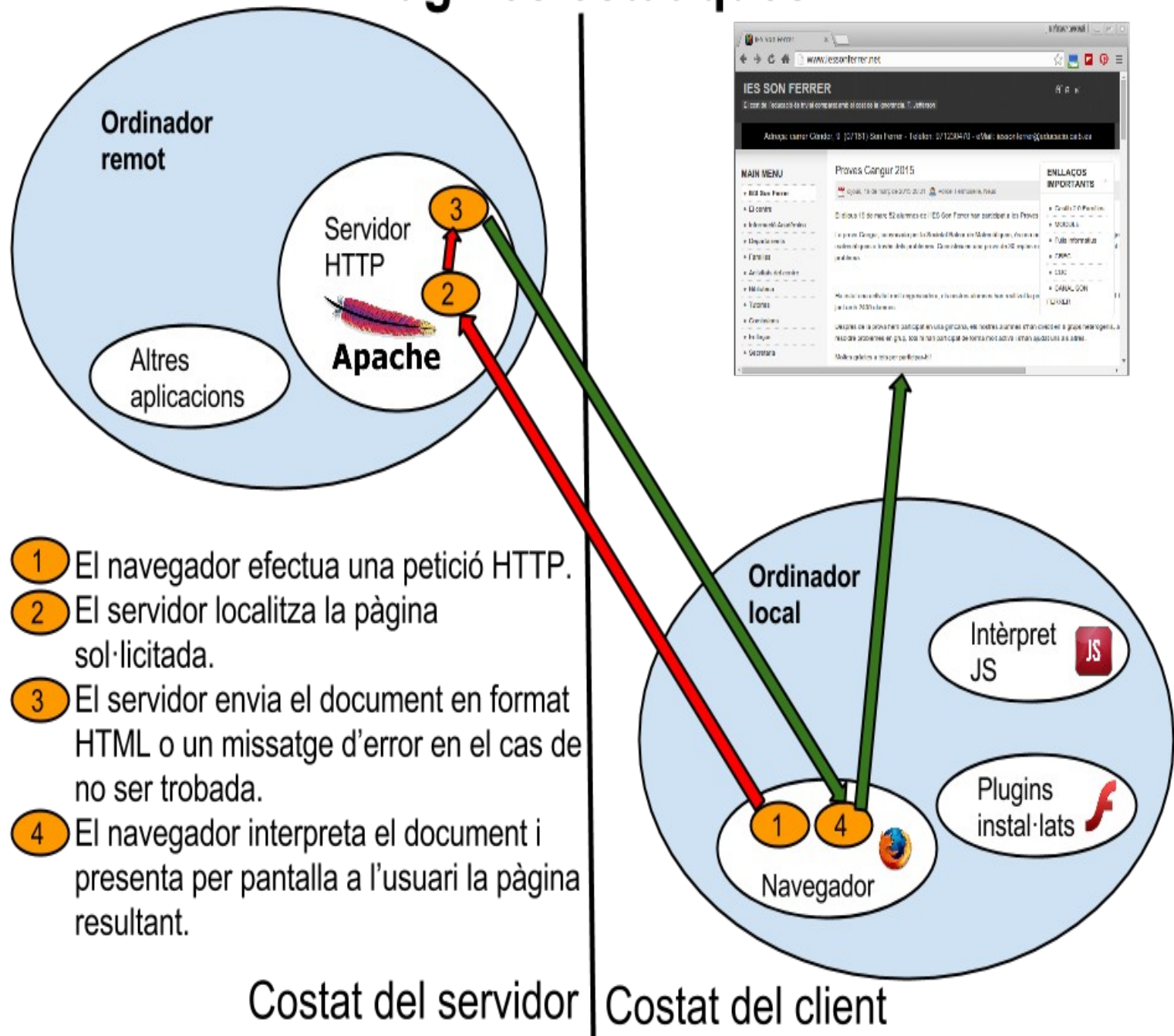
Client: Component consumidor de serveis.

Servidor: Procés proveïdor del servei.

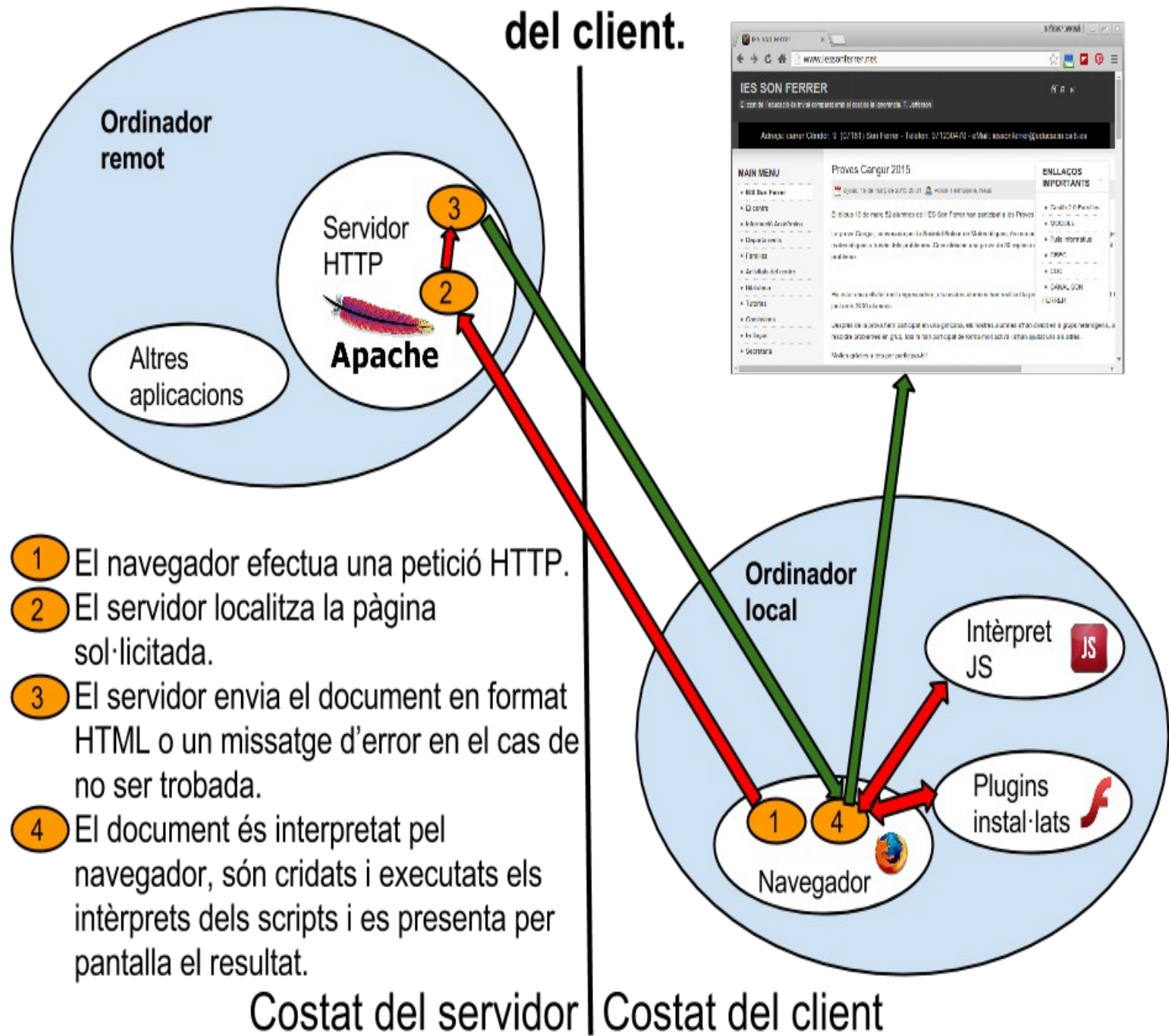
1. Evolució i característiques del navegadors Web (II)



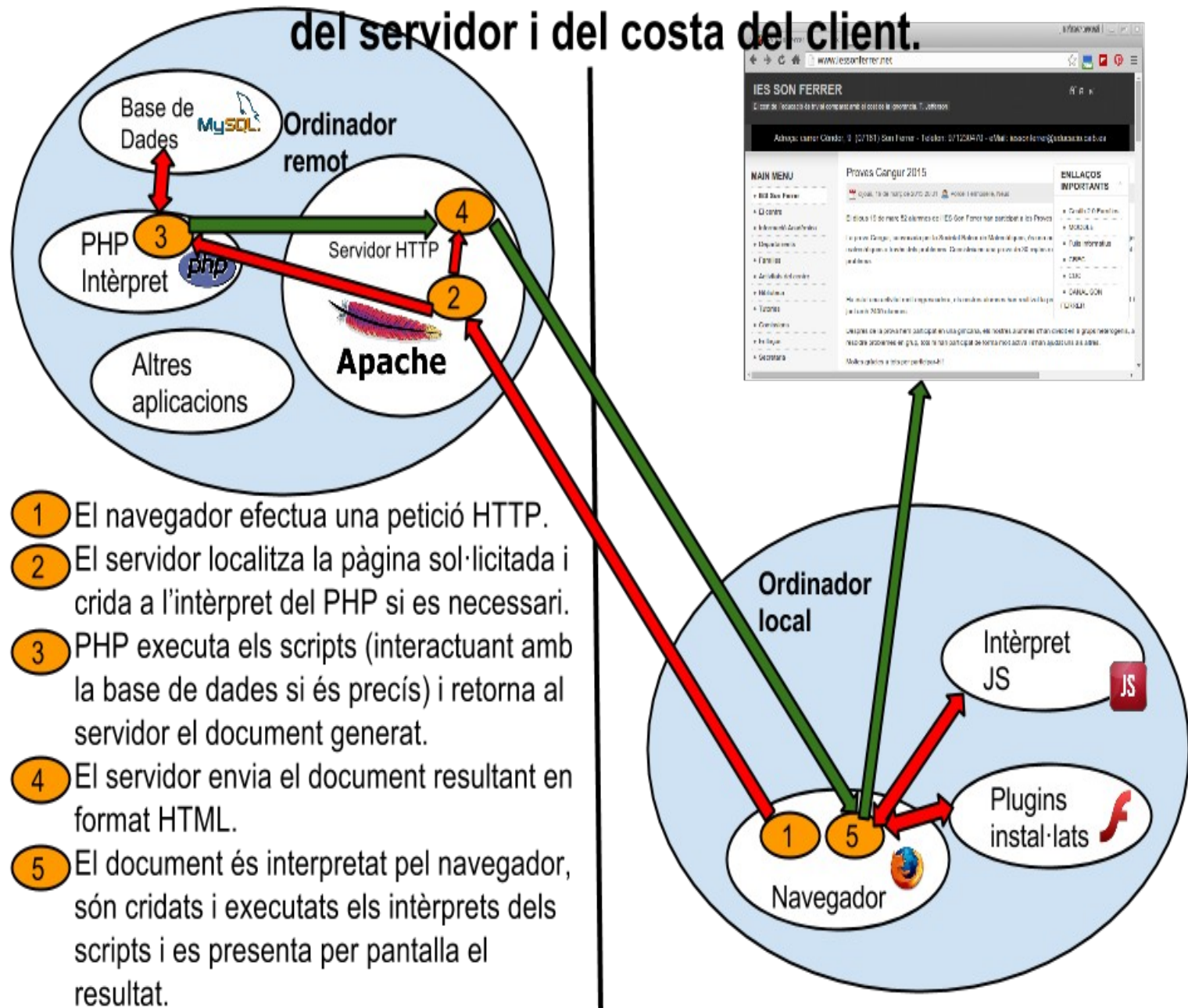
Pàgines estàtiques



Pàgines dinàmiques fent servir només aplicacions del costat del client.



Pàgines dinàmiques fent servir només aplicacions del costat del servidor i del costat del client.



1. Evolució i característiques del navegadors Web (VI)

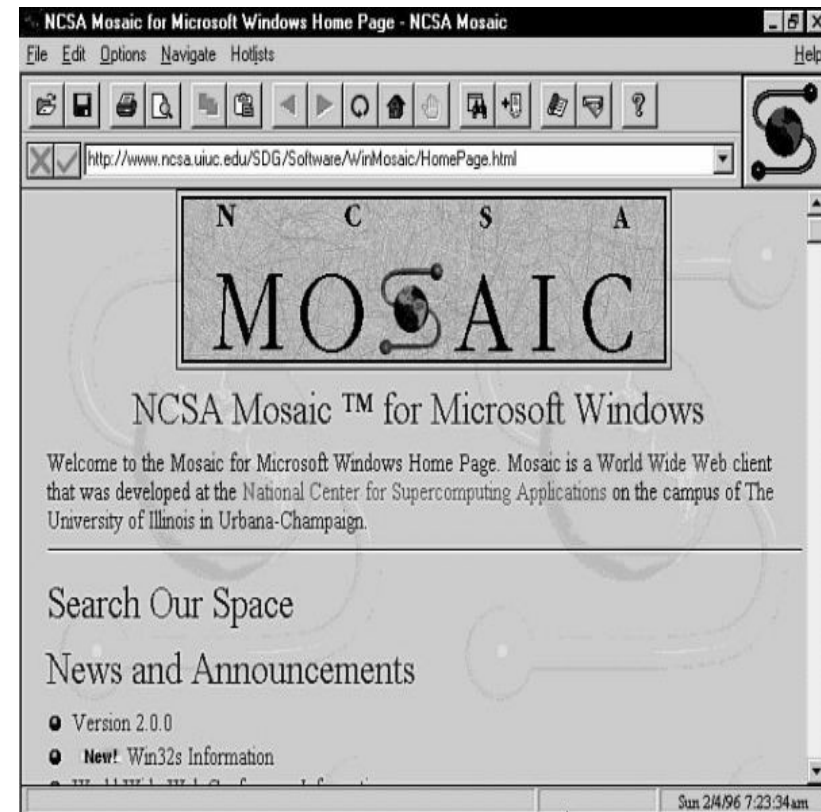
Navegador Web:

- Programari client per accedir a continguts dels servidors d'Internet. Normalment són gratuïts. Als recursos dels servidors s'accedeix mitjançant una URL (*Uniform Resource Locator*).
- Diferents tipus en funció de les seves capacitats, llenguatges suportats o facilitat de configuració.
- La majoria suporta DHTML (Dynamic HTML) per atorgar major funcionalitat.

1. Evolució i característiques del navegadors Web (VII)

Mosaic:

- Un dels primers navegadors i el primer amb capacitats gràfiques.
- Inicialment executat sobre UNIX, posteriorment a la resta de les plataformes.
- Base per a les primeres versions d'Internet Explorer i Mozilla.
- Va deixar de desenvolupar-se en 1997.



1. Evolució i característiques del navegadors Web (VIII)

Netscape Navigator (Communicator):

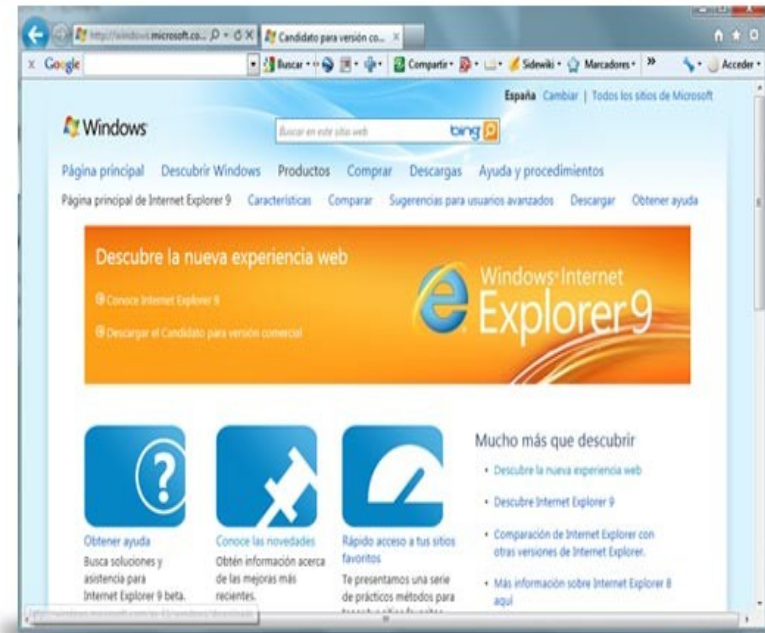


- Primer a incloure un mòdul per a l'execució de codi *script* (Javascript).
- “Perdedor” en la “guerra dels navegadors” enfront de Microsoft (domini d'aquest a la fi dels 90).
- Caraterístiques base per Mozilla Firefox.

1. Evolució i característiques del navegadors Web (IX)

Internet Explorer:

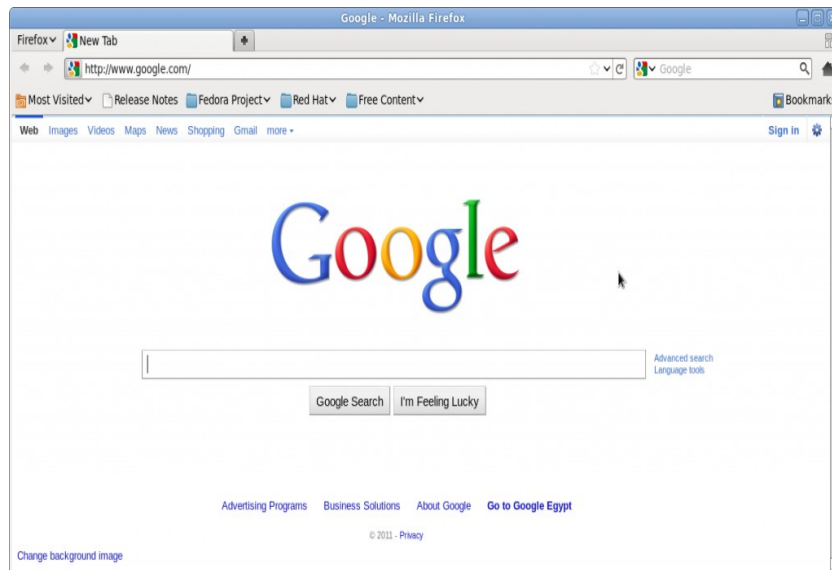
- Navegador de Microsoft.
- Quota de distribució i ús elevada gràcies a la seva integració en sistemes Windows.
- Descens en els últims anys davant Firefox o Chrome.
- A partir de versió 9.0 (finals 2011), suport per a estàndards web, personalització de navegació i seguretat.



1. Evolució i característiques del navegadors Web (X)

Mozilla Firefox:

- De codi obert, multiplataforma, de gran acceptació en la comunitat de desenvolupadors web.
- Gran varietat d'utilitats, extensions i eines per a la personalització i aparença del navegador.
- Va ser dels primers a incloure la navegació per pestanyes.
- En ser multiplataforma, ha retallat la quota de distribució que tenia Internet Explorer.



1. Evolució i característiques del navegadors Web (XI)

Google Chrome:

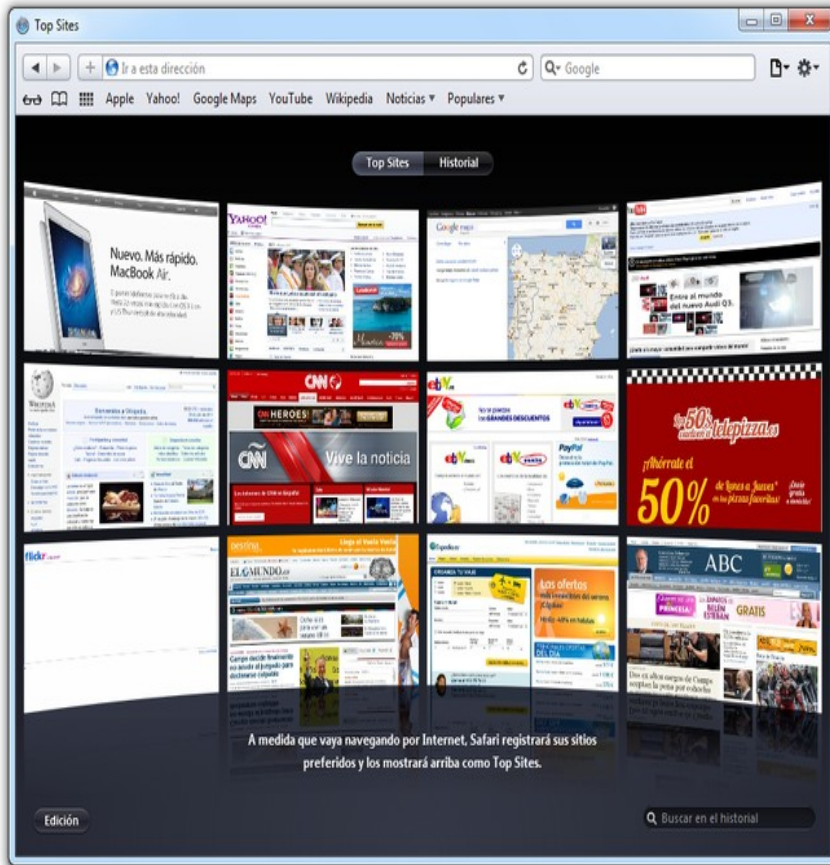
- Creat el 2008, és el navegador de Google compilat a partir de components de codi obert.
- Seguretat, velocitat i estabilitat són les seves característiques.
- La seva rapidesa i seguretat en els tests comparatius es deu al fet que segueix una arquitectura multiprocés on cada pestanya s'executa de forma independent.



1. Evolució i característiques del navegadors Web (XII)

Safari:

- Navegador per defecte del sistema Apple.
- També incorpora la navegació per pestanyes, corrector ortogràfic en formularis, emmagatzematge d'adreces favorites (“marcadors”), bloquejador de finestres emergents, suport per a motors de recerca personalitzat o un gestor de descàrregues propi.



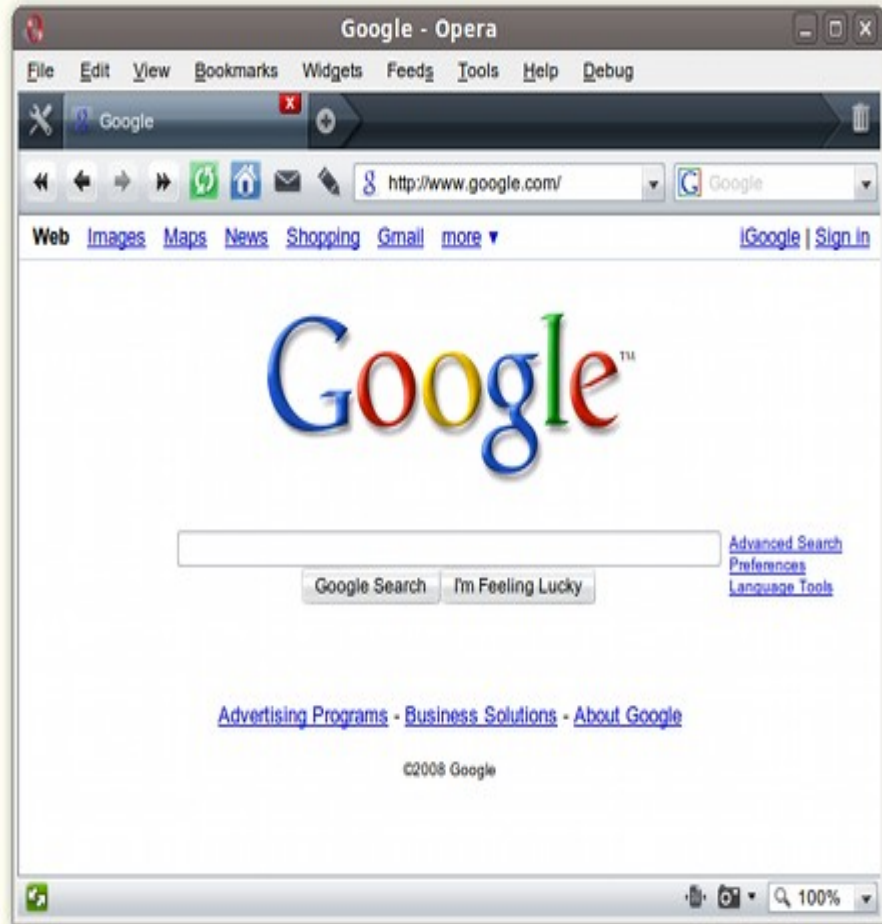
1. Evolució i característiques del navegadors Web (XIII)

Dolphin Browser:

- Popular en les plataformes de smartphones i tablets.
- Específic per Android.
- Un dels primers a incloure suport per a navegació multitàctil.
- Motor de renderitzat de pàgines similar a Chrome o Safari.



1. Evolució i característiques del navegadors Web (XIV)

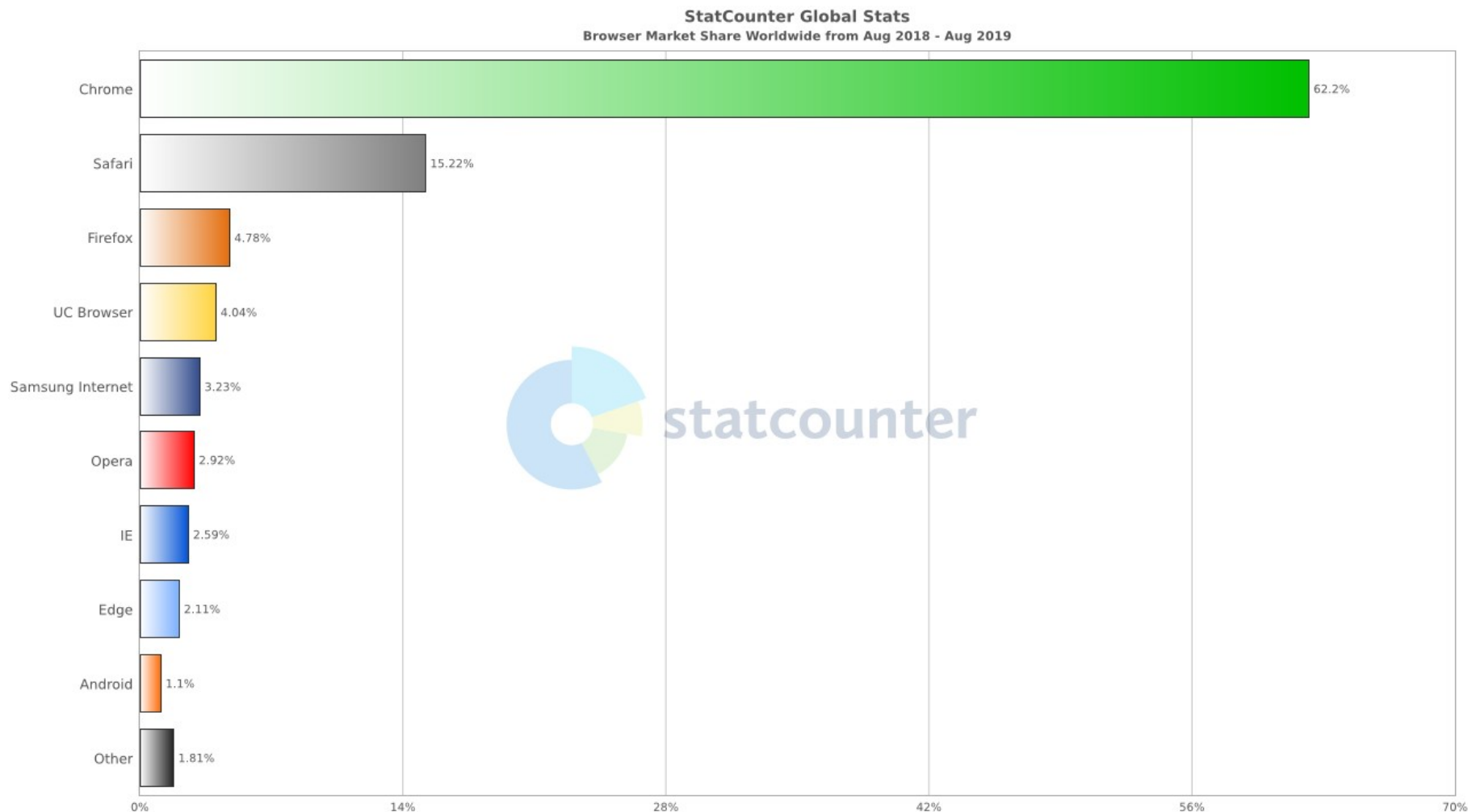


Opera:

- Velocitat, seguretat, suport d'estàndards (especialment CSS).
- Multiplataforma i mida reduïda.
- Les darreres versions d'Òpera fan servir el motor WebKit, el mateix que Chrome i Safari.

1. Evolució i característiques del navegadors Web (XV)

Estadística d'ús de navegadors a nivell mundial:



Font: <http://gs.statcounter.com/>

1. Evolució i característiques del navegadors Web (XVI)

Criteris per diferenciar els navegadors:

- **Plataforma d'execució:**

- No tots els navegadors es poden usar en qualsevol ordinador.
- Safari és exclusiu d'Apple. Va fer versions per a Windows fins l'estiu de l'any 2012.

- **Característiques del navegador.**

- La majoria afegeixen funcionalitats associades a l'experiència de l'usuari: administració de marcadors, gestors de descàrrega, emmagatzematge segur de contrasenyes i dades de formulari, correcció ortogràfica o definició d'eines de recerca.

1. Evolució i característiques del navegadors Web (XVII)

Criteris per diferenciar els navegadors (continuació):

- **Personalització de la interfície.**
 - Suport per a la navegació per pestanyes, bloquejadors de finestres emergents, integració amb visualitzadors de formats de fitxers (PDF), opcions de zoom o funcions avançades de recerca de text.
- **Suport de tecnologies Web.**
 - Nivell de suport de tecnologies CSS, Java, llenguatges de scripting del client (JavaScript), RSS o Atom, XHTML...
- **Llicència de programari.**
 - De codi lliure, com Mozilla (licència GNU GPL) i Google Chrome (licència BSD).
 - Propietaris, com a Internet Explorer o Safari.
 - Excepte rares excepcions (OmniWeb) tots són gratuïts.

2. Arquitectura d'execució (I)

La interacció usuari-navegador és, bàsicament, la següent:

- A) L'usuari indica l'adreça del recurs al que vol accedir.
- B) El navegador visualitza el recurs en la pantalla de l'usuari.

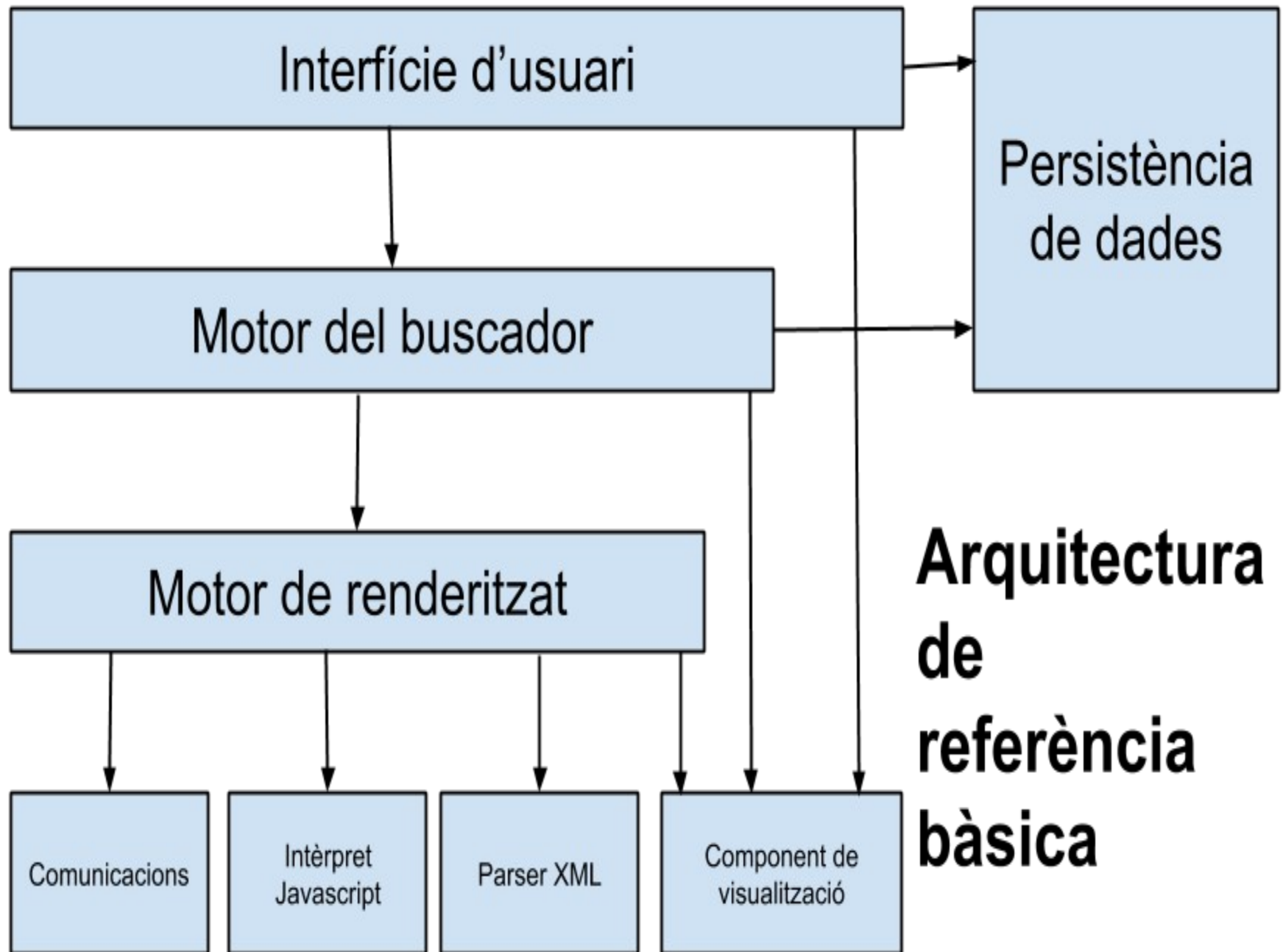
Cada navegador web té una forma d'interpretar aquesta interacció i pot centrar-se més en:

- Oferir una resposta ràpida.
- Mostrar una resposta més fidel al contingut del recurs.
- Prioritzar els aspectes de seguretat en les comunicacions.

...

2. Arquitectura d'execució (II)

Cada navegador, per tant, està format per una sèrie d'elements i components que conformen la seva arquitectura, encara que tots parteixen d'una **arquitectura de referència** bàsica i comuna a tots els navegadors.



**Arquitectura
de
referència
bàsica**

2. Arquitectura d'execució (IV)

Arquitectura de referència: **Subsistema d'Interfície d'Usuari.**

- Capa que actua d'interfície entre l'usuari i el motor del cercador.
- Visualitza barres d'eines.
- Visualitza el procés de càrrega.
- Gestiona les descàrregues de forma intel·ligent.
- Plasma les preferències de configuració d'usuari o impressió.
- Pot comunicar-se amb el Sistema Operatiu, en alguns casos, per al maneig de sessions d'usuari o l'emmagatzematge de preferències.

2. Arquitectura d'execució (V)

Arquitectura de referència: **Subsistema del Motor del Buscador.**

- Capa que actua d'interfície d'alt nivell per al motor de renderitzat.
- També anomenat **Motor de Navegació.**
- Carrega l'adreça determinada (URL).
- Suporta els mecanismes bàsics de navegació (pàgina anterior o següent, recarrega de la pàgina, etc.).
- Gestiona les alertes de Javascript.
- Consulta i administra les preferències d'execució del motor de renderitzat.

2. Arquitectura d'execució (VI)

Arquitectura de referència: **Subsistema de Renderitzat.**

- Encarregat de produir una representació visual del recurs obtingut.
- Interpreta el codi de la pàgina Web.
- En funció de les tecnologies suportades serà capaç de mostrar documents HTML o XML, fulls d'estil CSS, imatges i fins i tot contingut embegut (àudio/vídeo).
- Estableix les dimensions exactes de cada element a mostrar i la posició d'aquests.
- Els motors de renderitzat més coneguts són:
 - Gecko (Mozilla i Galeon)
 - Trident (Internet Explorer)
 - WebKit (Chrome, Safari i Epiphany)
 - Prest (Opera)
 - Tasman (Internet Explorer per a Mac)

2. Arquitectura d'execució (VII)

Arquitectura de referència: **Subsistema de Comunicacions.**

- Implementa els protocols de transferència de fitxers i documents (HTTP, FTP, ...).
- Identifica la codificació de les dades obtingudes en funció del seu tipus (text, àudio, vídeo, ...) codificat en estàndard MIME (*Multipurpose Internet Mail Extensions*).
- Pot emmagatzemar una cau (“*caché*”) d'elements accedits recentment.

2. Arquitectura d'execució (VIII)

Arquitectura de referència: **Intèrpret de JavaScript.**

- Analitza i executa el codi JavaScript intercalat en HTML.
- Pot ser configurat, i fins i tot deshabilitat des del motor de navegació o el motor de renderitzat.
- Cada navegador té els seus propis mòduls d'interpretació, per la qual cosa és possible que existeixin subsistemes intèrprets d'altres llenguatges, com applets de Java, Ajax o ActionScript.

2. Arquitectura d'execució (IX)

Arquitectura de referència: **Parser XML**.

- Permet carregar en memòria una representació en arbre (arbre DOM, Document Object Model) de la pàgina.
- L'accés als diferents elements d'una pàgina per part del navegador és molt més ràpid.

NOTA: Un “*parser*” és un analitzador sintàctic.

2. Arquitectura d'execució (X)

Arquitectura de referència: **Component de Visualització.**

- Ofereix funcionalitats relacionades amb la visualització dels continguts d'un document HTML en una pàgina web.
- Ofereix primitives de dibuix i posicionament en una finestra, un conjunt de components visuals predefinits (*widgets*) i un conjunt de fonts tipogràfiques.
- Està relacionat amb les llibreries de visualització del Sistema Operatiu.

2. Arquitectura d'execució (XI)

Arquitectura de referència: **Subsistema de persistència de dades.**

- Funciona com a magatzem de diferents tipus de dades per als principals subsistemes del navegador.
- Solen estar relacionats amb l'emmagatzematge d'historials de navegació i manteniment de sessions d'usuari en disc.
- Inclou les preferències de configuració del navegador o la llista de marcadors.
- A baix nivell, aquest sistema administra també els certificats de seguretat i les *cookies*.

2. Arquitectura d'execució (XII)

Estructura d'una URI:

Esquema:Part jeràrquica?Sol·licitud#Fragment

Esquema://màquina/directori/arxiu#fragment

- Esquema: identifica el protocol (normalment http). En sabeu un altre?
- Part jeràrquica (màquina i directori): Conté la informació del domini o IP a accedir i també la ruta cap a l'arxiu en el servidor.
- Sol·licitud: variables que passam cap al recurs.
- Fragment: Permet indicar una subdirecció dins el recurs.

Les URLs són un subconjunt de les URI. Cercau-ne les diferències.

3. Llenguatges i tecnologies de programació en entorn client (I)

Són aquells que s'executen en el navegador Web.

El llenguatge client principal és l'HTML (*HyperText Markup Language*), i les seves variacions DHTML i XHTML.

Amb la finalitat de millorar la interactivitat amb l'usuari s'inclouen els llenguatges d'*script Javascript* o *VBScript*.

3. Llenguatges i tecnologies de programació en entorn client (II)

Altres llenguatges són *ActionScript* (per crear contingut Flash).

AJAX (com a tecnologia d'extensió a Javascript per a comunicació asíncrona).

Els applets de Java i les CSS són altres tecnologies de l'entorn client.

3. Llenguatges i tecnologies de programació en entorn client (III)

XML (eXtensible Markup Language):

- És llenguatge (o metallenguatge) d'etiquetatge amb unes regles molt estrictes de codificació.
- S'usa per a l'intercanvi d'una gran quantitat de dades.
- Pot adjuntar un DTD (Definició de Tipus de Document) per a la validació d'un arxiu de dades (vàlid i ben format).
- El seu objectiu principal és descriure dades per a la seva transferència eficient i no mostrar-los.
- Els navegadors actuals solen mostrar els arxius XML de forma esquemàtica.

3. Llenguatges i tecnologies de programació en entorn client (IV)

HTML (HyperText Markup Language):

- És una particularització del llenguatge SGML (Standard Generalized Markup Language), un sistema per a l'organització i etiquetatge de documents (ISO 1986).
- És el llenguatge de marques de text més usat en la Web. Creat en 1989 per Tim Berners Lee a partir del concepte d'hipertext i el llenguatge de marques SGML.
- No és un llenguatge de programació. Es basa en l'ús d'un sistema d'etiquetes tancat aplicat a un document de text.

3. Llenguatges i tecnologies de programació en entorn client (V)

HTML (HyperText Markup Language, continuació):

- No necessita ser compilat, sinó interpretat pel navegador a mesura que s'avança en el document.
- Amb ell es pot fer: organitzar text i objectes, crear llistes i taules i, òbviament, permetre els hipervíncles (essència de la Web).
- La seva evolució a donat lloc a llenguatges derivats que veurem a continuació.

3. Llenguatges i tecnologies de programació en entorn client (VI)

XHTML (eXtensible HTML):

- No és més que una adaptació d'HTML al XML.
- Pretén un etiquetatge més estricte que l'HTML.

3. Llenguatges i tecnologies de programació en entorn client (VII)

DHTML (Dynamic HTML).

- Consisteix en una forma d'aportar interactivitat a les pàgines web.
- Permet la integració d'HTML amb llenguatges d'*scripting*, fulls d'estil personalitzats i la identificació dels continguts d'una pàgina web en format d'arbre (DOM).
- Permet augmentar la funcionalitat i interactivitat d'una pàgina web: crear efectes, animacions, jocs, ... i crear un autèntic entramat de capes.

3. Llenguatges i tecnologies de programació en entorn client (VIII)

CSS (Cascade Style Sheets).

- Els fulls d'estil serveixen per separar el format que es vol donar a la pàgina de l'estructura d'aquesta i altres instruccions.

3. Llenguatges i tecnologies de programació en entorn client (IX)

JavaScript.

- Llenguatge de programació de scripting (interpretat pel navegador) i embegut en un document HTML.
- Es defineix com orientat a objectes, feblement tipat i amb característiques dinàmiques.
- Permet millores en la interfície de l'usuari i la creació de pàgines dinàmiques.
- Sintaxi similar a C, encara que adopta noms i convencions pròpies de Java (encara que no té cap relació amb aquest últim llenguatge).
- Actualment existeixen dos estàndards alineats: ECMAScript i ISO/IEC 16262.
- Tots els navegadors moderns ho interpreten.

3. Llenguatges i tecnologies de programació en entorn client (X)

Applets de Java.

- Són petits components (objectes independents) que s'insereixen en una pàgina per incloure funcionalitats complexes.
- Són fragments de codi Java que s'executen en el client i es beneficien de la potència i flexibilitat d'aquest llenguatge.
- Els applets es programen en Java i s'envien al client precompilats.
- Són, per tant, menys depenents del navegador que Javascript i, fins i tot, independents del Sistema Operatiu.
- Són més lents de processar i no tenen accés a cap altre component de la pàgina.

3. Llenguatges i tecnologies de programació en entorn client (XI)

Ajax (Asynchronous JavaScript And XML).

- Conjunt de tècniques i mètodes de desenvolupament web per a la creació d'aplicacions web interactives.
- Amb Ajax es manté una comunicació asíncrona amb el servidor en segon pla, al contrari que amb HTML, que perd la comunicació amb el servidor quan acaba de carregar-se la pàgina.
- Es pot, per tant, realitzar canvis sobre les pàgines del client sense que es necessiti recarregar-les. Això implica un augment en la velocitat i en la interactivitat.
- El fonament d'Ajax es troba en l'ús d'un objecte específic de Javascript denominat XMLHttpRequest i acceptat per la majoria dels navegadors actuals.
- És una combinació, doncs de 4 tecnologies existents: XHTML/HTML i CSS, DOM, Javascript i XML.

3. Llenguatges i tecnologies de programació en entorn client (XII)

Adobe Flash i ActionScript.

- Flash és una tecnologia d'animació que utilitza ActionScript com a llenguatge principal.
- Es fa servir per crear gràfics i animacions.
- Actualment està en clar declivi davant altres tecnologies i llenguatges com HTML 5.
- El seu ús ha permès crear aplicacions interactives de gran complexitat i visualment molt atractives, permetent augmentar el grau d'interactivitat de l'usuari amb la pàgina web.
- En ser animacions d'índole vectorial el consum del processador (i de bateria de dispositius mòbils) és més elevat. A més és programari propietari.

4. Integració de codi amb les etiquetes HTML

En el mateix document HTML:

```
<script type="text/javascript">  
    alert("Hello world!");  
</script>
```

En elements HTML:

```
<input type="button" value="Clica" onclick="alert('Hello world!');" />
```

En un arxiu extern:

```
// arxiu missatge.js en la carpeta js i després insertar-ho  
alert("Hello world!");
```

```
<script type="text/javascript" src="js/missatge.js"></script>
```