

# ARQUITECTURA WEB



# ÍNDEX

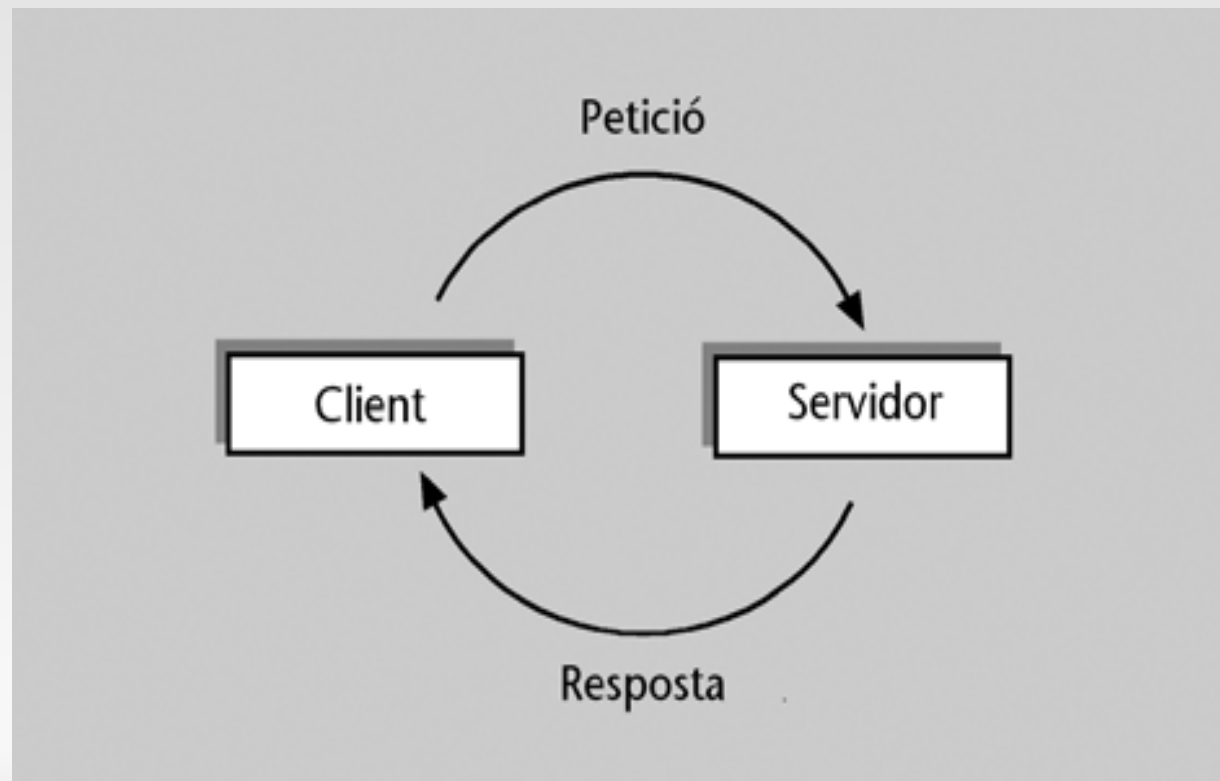
- **ARQUITECTURA CLIENT-SERVIDOR**
- **SERVIDORS WEB**
- **ARQUITECTURA APLICACIONS WEB**
- **SERVIDORS INTERMEDIARIS**
- **CONTINGUTS DISTRIBUÏTS**

# Client-Servidor

- El Web és un exemple d'arquitectura client-servidor.
- En el **model client-servidor** hi ha dos tipus de components:
  - **Clients:** fan peticions de servei. Normalment els clients inicien la comunicació amb el servidor.
  - **Servidors:** proveeixen serveis. Normalment els servidors esperen rebre peticions. Un cop han rebut una petició, la resolen i retornen el resultat al client.

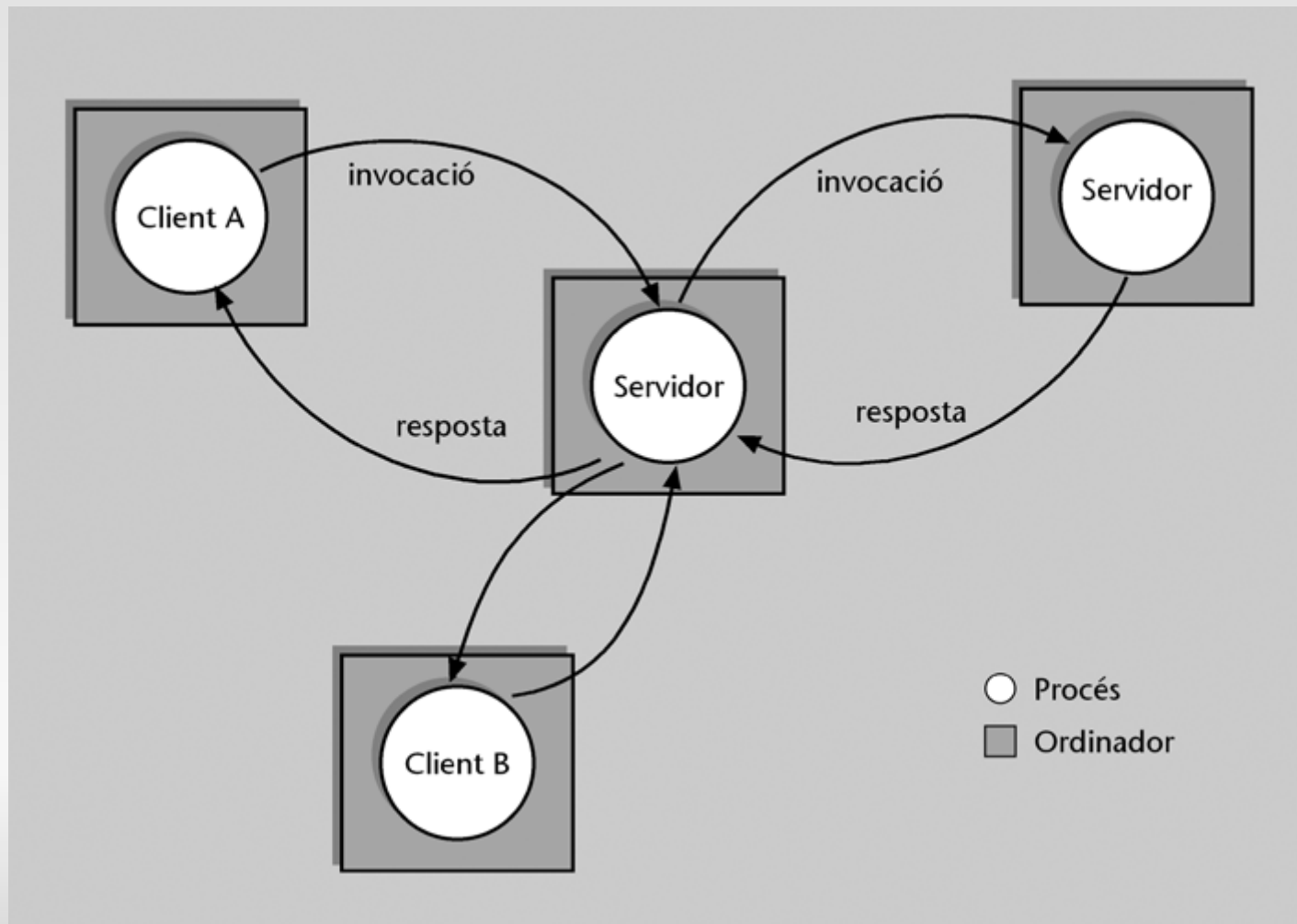
# Client-Servidor

- L'esquema general **client-servidor** és el següent:



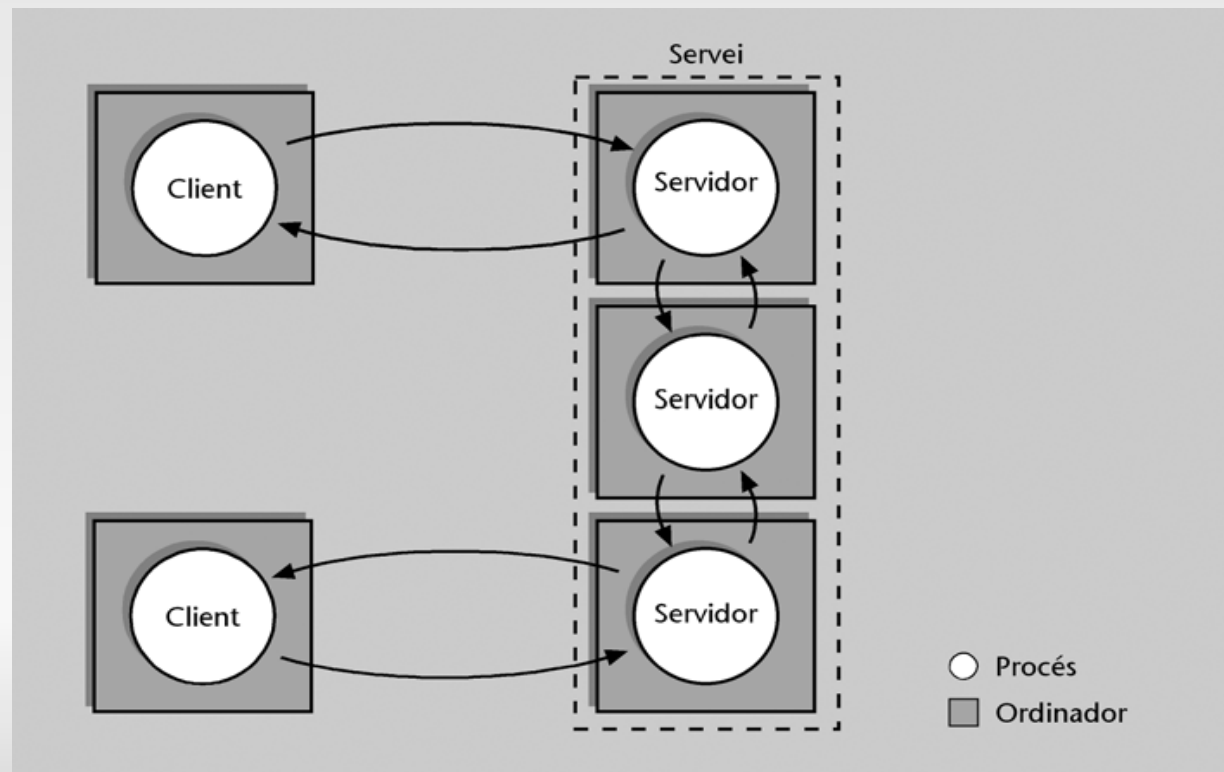
# Client-Servidor

- Un servidor també pot ser client d'altres servidors. (per exemple DNS).



# Client-Servidor

- Els **serveis** es poden implementar com a diferents processos servidors que s'executen en diferents ordinadors i que interactuen per a proporcionar un servei a processos clients



# Client-Servidor

- **La reproducció** s'usa per a incrementar el:
  - Rendiment
  - La disponibilitat
  - Millorar la tolerància a fallades.
- Per exemple, el web proporcionat a [www.google.com](http://www.google.com) està mapat a diferents servidors que tenen dades reproduïdes.

# Servidors Web

- Un servidor web és un programa que atén les diverses peticions dels navegadors, proporcionant els recursos que sol·liciten mitjançant el protocol HTTP. Té un esquema de funcionament molt senzill que executa de manera infinita el bucle següent:
  1. Espera peticions al port TCP assignat (l'estàndard per a HTTP és el 80).
  2. Rep una petició.
  3. Busca el recurs a la cadena de petició.
  4. Envia el recurs per la mateixa connexió per on ha rebut la petició.
  5. Torna al punt 2.



# Servidors Web

- Un servidor web que seguís l'esquema anterior compliria els requisits bàsics dels servidors HTTP, encara que **només podria servir fitxers estàtics**.
- Un dels aspectes més importants del servidor web escollit és el nivell de suport que ens ofereix perquè serveixi **contingut dinàmic**.
- La majoria de servidors web ofereixen suport per a CGI (es pot recordar que els CGI són el mètode més antic i simple de generació de contingut dinàmic). Actualment també ofereixen suport per a alguns llenguatges de programació com PHP, JSP, ASP

# Arquitectura Aplicacions Web

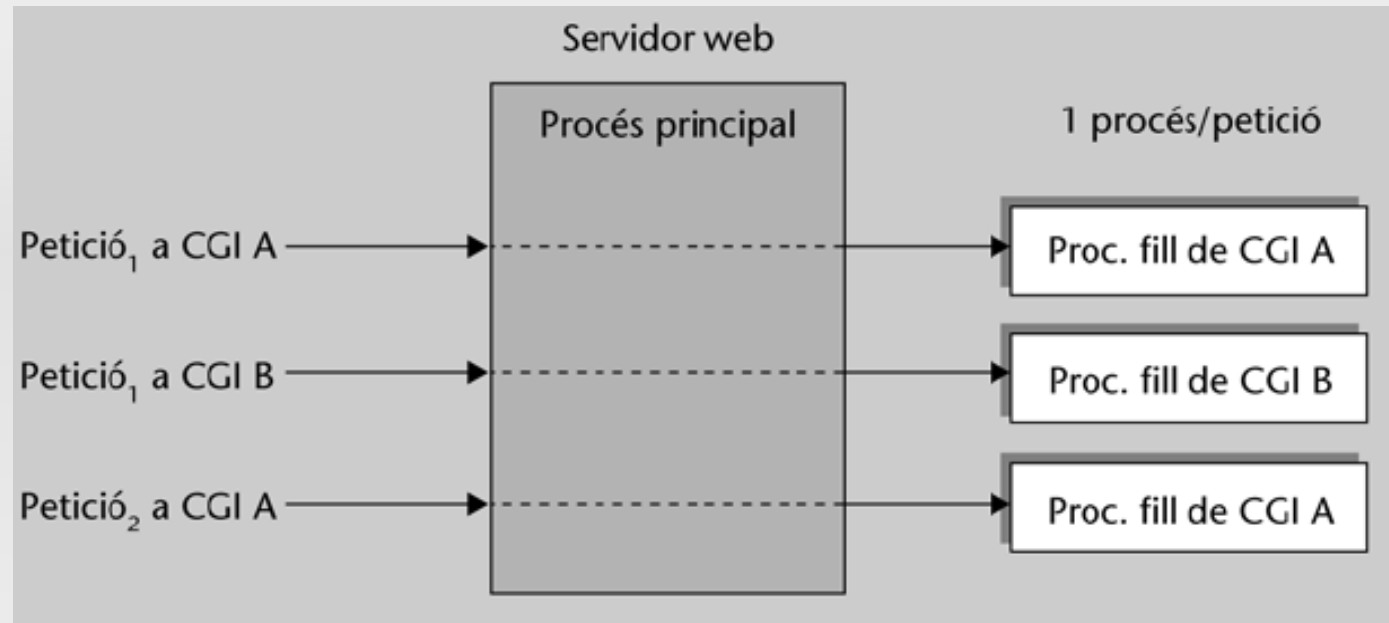
- Inicialment el web era simplement una col·lecció de **pàgines estàtiques**, documents, etc., que es podien consultar o baixar.
- L'evolució va ser un mètode per confeccionar **pàgines dinàmiques** que permetessin que el que es mostra fos dinàmic (generat o calculat a partir de les dades de la petició). Aquest mètode va ser el **CGI**.
- El **CGI** definia un mecanisme mitjançant el qual podíem **passar informació entre el servidor HTTP i programes externs**.

# Arquitectura Aplicacions Web

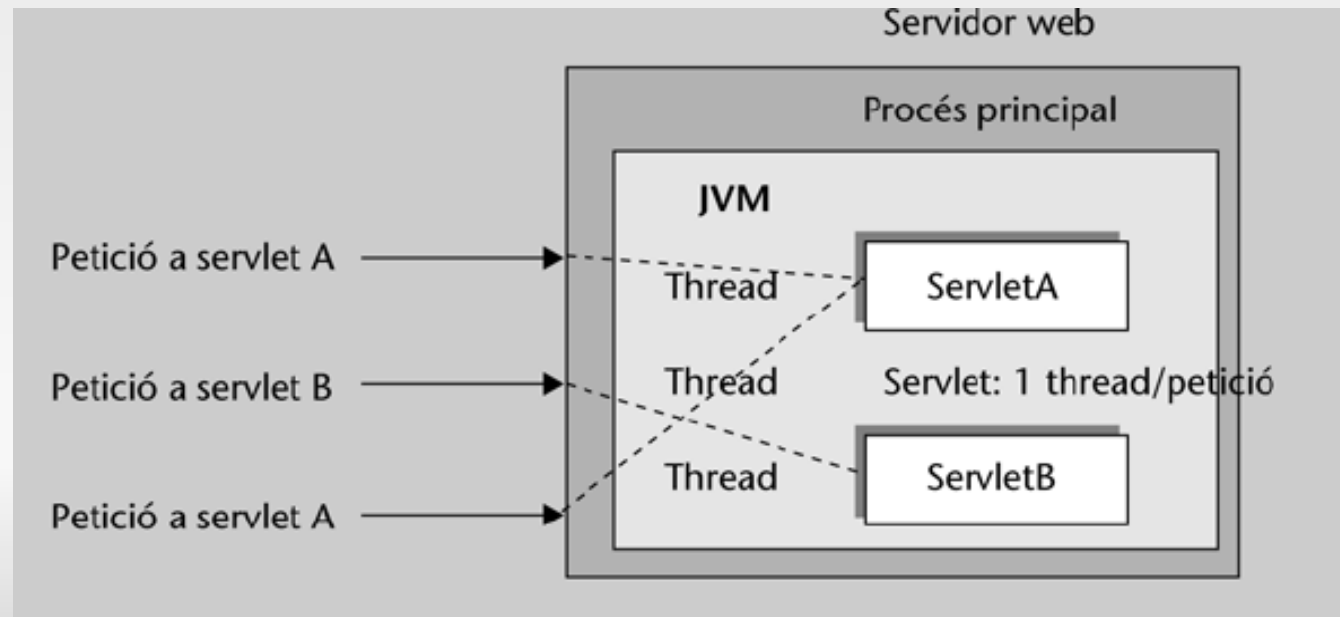
- El **CGI** = problemes de rendiment.  
1 petició = 1 procés en el servidor web.
- **Solucions:**
  - Sistemes d'execució de mòduls més integrats amb el servidor.
  - Interprets de llenguatges de programació integrats al servidor web.
  - Servlets, JSP, PHP, ASP , ASP.NET...

# Arquitectura Aplicacions Web

- Model CGI



- Model Servlet  
JAVA



# Arquitectura Aplicacions Web

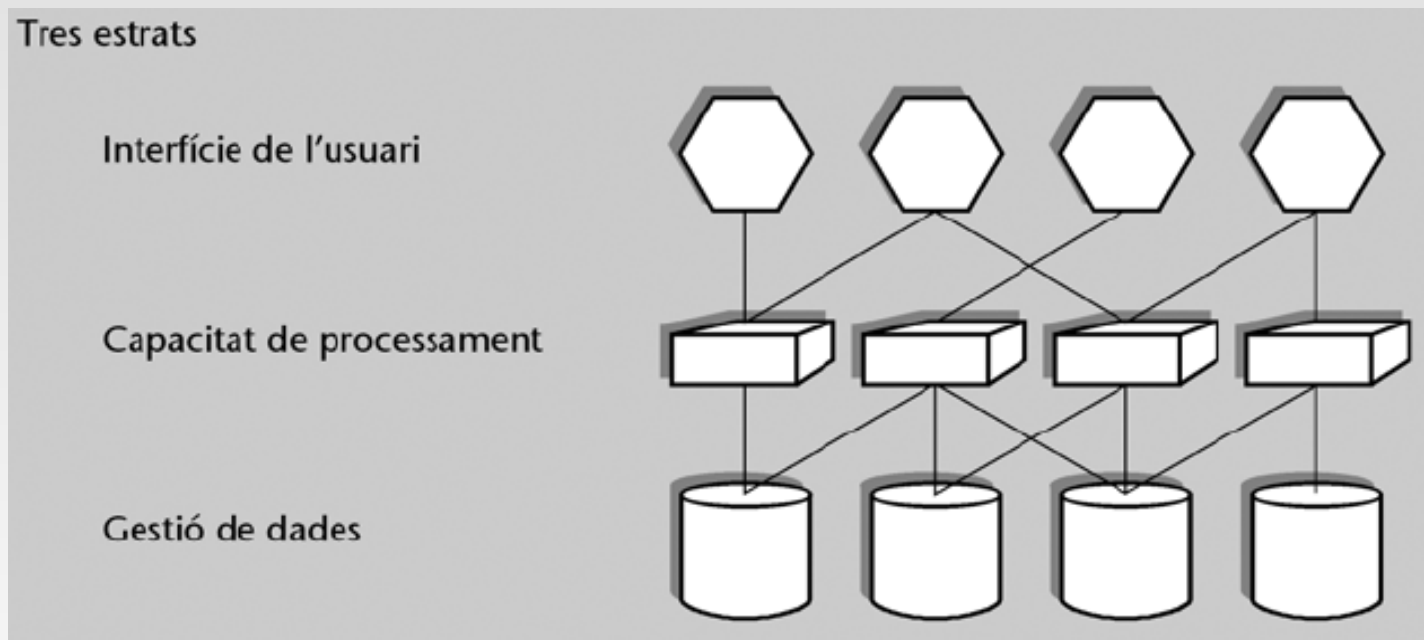
- **Arquitectures N capes** ( $N=1,2,3 \dots$ )

En les arquitectures multicapa, **la funcionalitat està distribuïda entre diverses plataformes o ordinadors.**

- La **interfície** resideix en l'ordinador de l'usuari.
- Els **serveis funcionals** (lògica de negoci) poden estar en un o més ordinadors
- **Les dades** o els sistemes propietaris estan en plataformes addicionals.

# Arquitectura Aplicacions Web

- **Arquitectures de tres capes**  
(three tier software architecture)



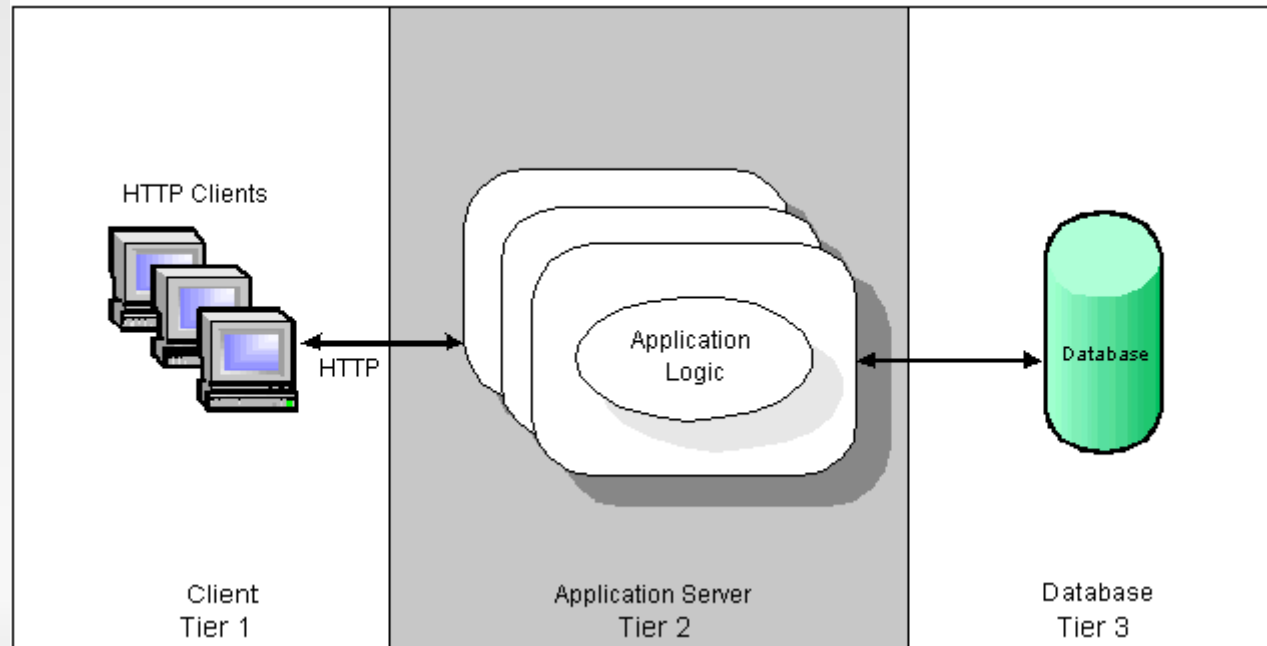
- Millora en el rendiment, la flexibilitat, el manteniment, la reusabilitat i l'escalabilitat.

# Arquitectura Aplicacions Web

- Un cas particular d'aplicacions client-servidor són les aplicacions que s'executen aprofitant l'arquitectura del web.
- Aquestes aplicacions es basen en el fet de tenir **tota la capacitat de processament en un servidor web** (o conjunt de servidors) als quals s'accedeix des d'un navegador web.
- L'ús d'aquestes arquitectures facilita el disseny de les aplicacions, ja que no cal implementar la comunicació entre el client i el servidor (s'utilitza el **protocol HTTP**)

# Arquitectura Aplicacions Web

- La facilitat i universalitat en l'accés a les aplicacions que proporciona aquesta arquitectura és la base dels serveis oferts a Internet.





# Servidors Intermediaris (Proxy)

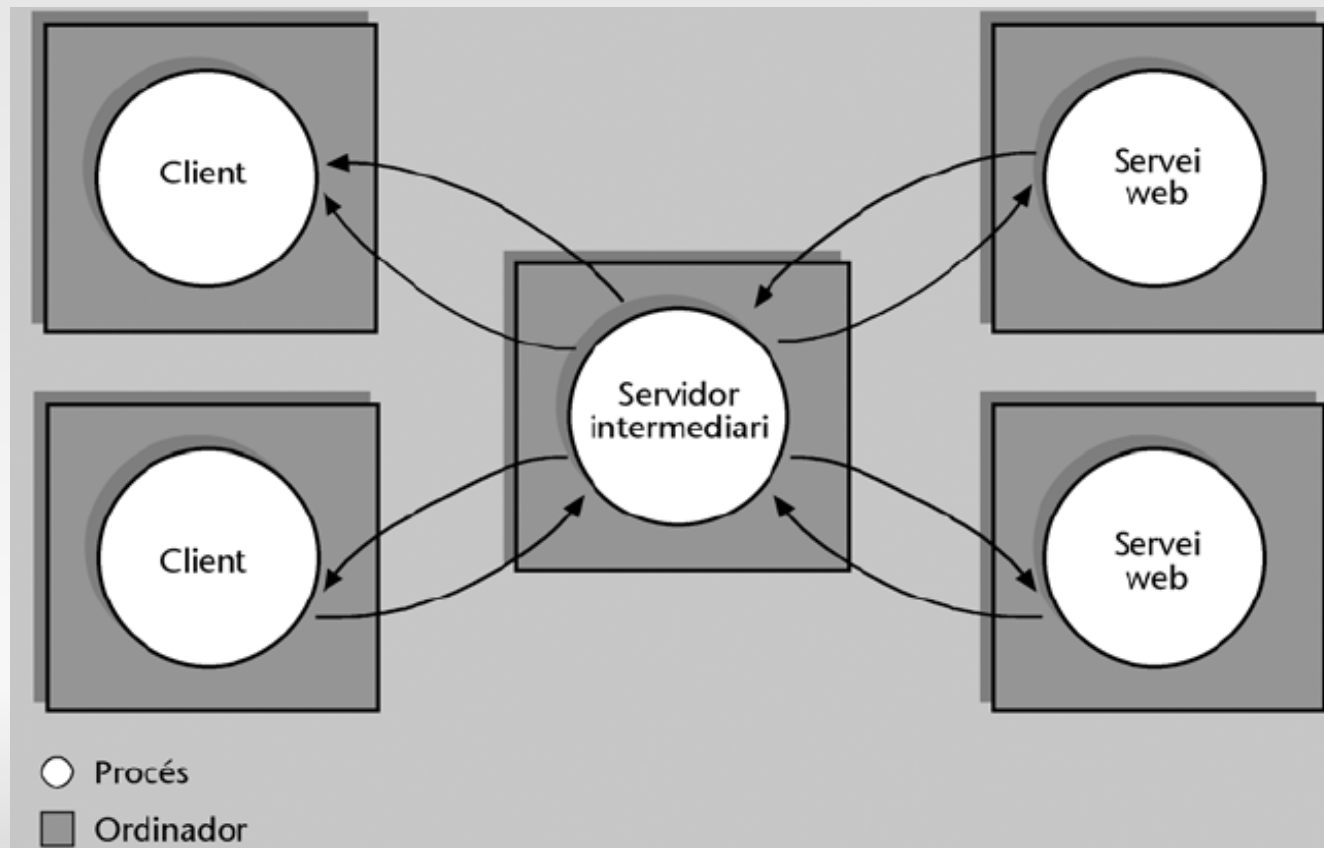
- El model client-servidor es dota d'un conjunt de mecanismes per a millorar la seva **disponibilitat, accessibilitat i temps de resposta**. Aquests mecanismes són les **memòries cau i els intermediaris**.
- Una memòria cau és un **magatzem d'objectes usats recentment** que actua com a medidora entre un client i un servidor.

# Servidors Intermediaris (Proxy)

- **Exemple:** memòries cau locals dels navegadors.
  - Els navegadors web mantenen en el sistema de fitxers local del client una memòria cau amb les pàgines web o els recursos visitats recentment.
  - Aquests, abans de presentar la pàgina de la memòria cau a l'usuari, comproven al servidor original, utilitzant una petició HTTP especial, si la pàgina està actualitzada.

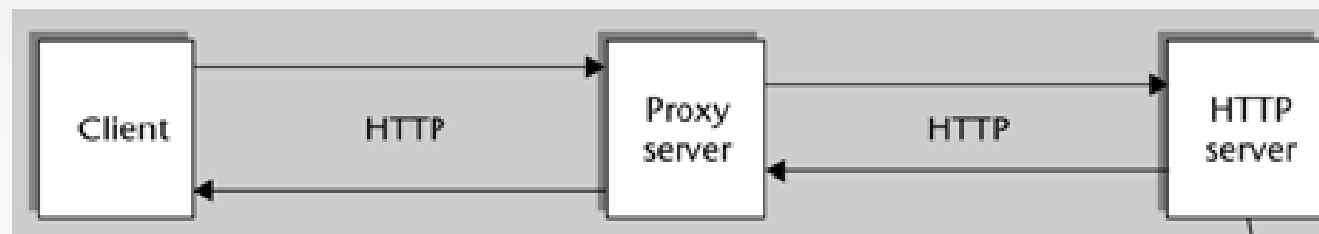
# Servidors Intermediaris (Proxy)

- Cada client pot tenir la seva memòria cau o aquestes poden estar situades en un **servidor intermediari (proxy)** que pot ser compartit per diversos clients.



# Servidors Intermediaris (Proxy)

- El model HTTP no solament considera la connexió directa d'un client amb el servidor, sinó també la connexió per mitjà de **servidors intermediaris**, com és el cas dels proxies.
- Un servidor intermediari (proxy) accepta peticions (HTTP) de clients o altres intermediaris, i genera al seu torn peticions cap a altres mediadors o cap al servidor web de destinació. Actua com a servidor del client i com a client del servidor.



# Servidors Intermediaris (Proxy)

- Aprofitant que tant la petició com el resultat (la pàgina web) han de passar pel servidor intermediari, aquest es pot aprofitar per a fer algunes funcions:
  - **Control d'accés a continguts:** el servidor intermediari consulta si la pàgina sol·licitada és o no permesa per la política de l'organització.
  - **Control de seguretat** (punt de sortida únic a Internet)
  - Funció cau (**proxy-cache**)
  - Adaptar els continguts (formats imatges, clients mòbils)

# Servidors Intermediaris (Proxy)

- **FORWARD PROXY**

- Un forward proxy és un servidor intermediari que **se situa entre el client i els servidors que han d'atendre la petició.**
  - Un ús típic de forward proxy és el de proveir d'accés a Internet els clients d'una xarxa interna que per motius de seguretat se n'aïlla, i només permetre l'accés per mitjà de l'intermediari, més fàcil de protegir.
  - Un efecte de l'ús de servidor intermediari és que, per als servidors, l'accés als recursos semblarà que provingui del nostre sistema intermediari, no de l'adreça real del client.

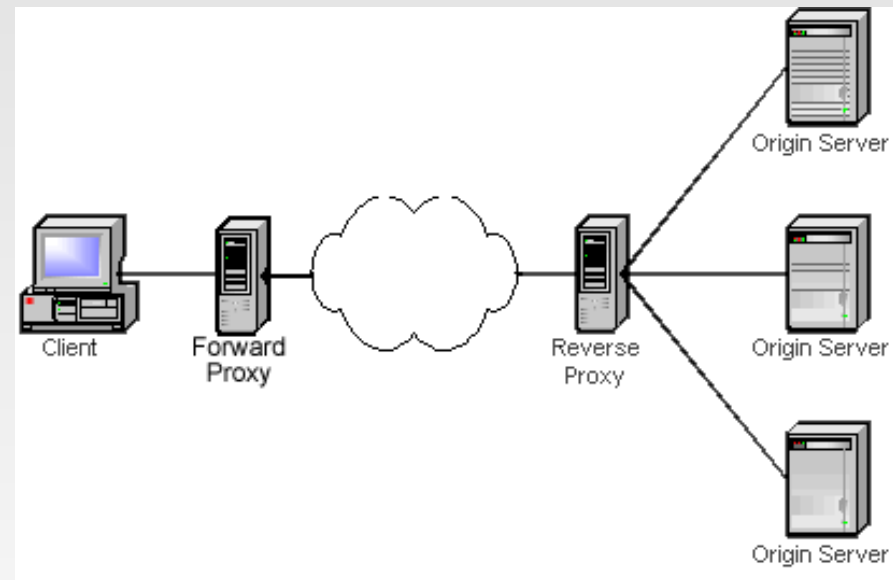
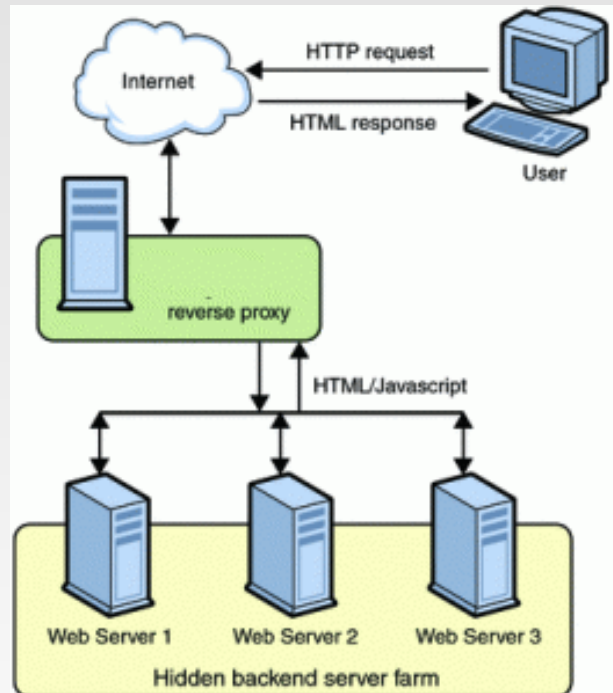
# Servidors Intermediaris (Proxy)

- **REVERSE PROXY**

- Un reverse proxy se situa davant d'un servidor concret i únicament controla l'accés a aquest servidor. Per als clients el reverse proxy apareixerà com un servidor web normal i no es necessita cap mena de configuració específica.
  - El client fa les peticions al servidor reverse proxy, que decideix on reexpedir aquestes peticions i una vegada resoltes, torna el resultat com si el reverse proxy fos la procedència del contingut.

# Servidores Intermediarios (Proxy)

## ■ EXAMPLES





# Servidors Intermediaris (Proxy)

- **REVERSE PROXY – EXEMPLES D'ÚS**
- **Filtrar i controlar l'accés** dels usuaris d'Internet a un servidor que volem que estigui molt aïllat.
- **Balanceig de càrrega** entre servidors o proveir mecanismes de memòria cau per a servidors més lents.
- **Unificar les adreces URL de diversos servidors** sota un únic espai de noms d'URL: el del servidor proxy.

# Continguts distribuïts

- Una altra millora que ajuda a combatre **el mal del “W-W-Wait”** són els sistemes de distribució de documents
- Les aplicacions que ofereixen continguts a Internet s'enfronten al repte de l'escala: **un únic servidor davant milions de persones** que eventualment poden demanar els seus serveis tots alhora: el proveïdor d'informació ha de posar tants recursos com audiència pugui tenir.
- Pot fer falta disposar de diversos servidors per a poder **repartir la càrrega**.

# Continguts distribuïts

- **Repartir peticions entre diverses màquines:**
  - **Mirrors:** un programa que readreça la petició HTTP a la millor rèplica.
  - El **DNS** torna diverses adreces IP pel mateix nom.
  - **Readreçament a nivell de transport (L4 switch):** un encaminador mira els paquets IP de connexions TCP cap a un servidor web (port 80) i les redirigeix a la màquina interna menys carregada
  - **Readreçament a nivell d'aplicació (L7 switch):** un encaminador que mira les connexions HTTP i pot decidir amb quina rèplica contactar en funció de l'URL sol·licitat. Molt complex.
  - Enviar totes les peticions a un **servidor intermediari invers** que respongui o amb contingut guardat en la memòria o passi la petició a un o diversos servidors interns.

# Continguts distribuïts

- **Xarxes de distribució de continguts (CDN – content delivery networks)**
  - Han sorgit empreses que s'han dedicat a instal·lar màquines a molts llocs del món i algorismes per a decidir quina màquina és la més adequada per a atendre les peticions, segons la ubicació del client i la càrrega de la xarxa.
  - Aquestes empreses venen aquest “**servidor web distribuït**” a diversos clients que paguen per a poder disposar d'un sistema de **servei web de gran capacitat i que pot respondre a demandes de continguts extraordinàries**.

# Continguts distribuïts

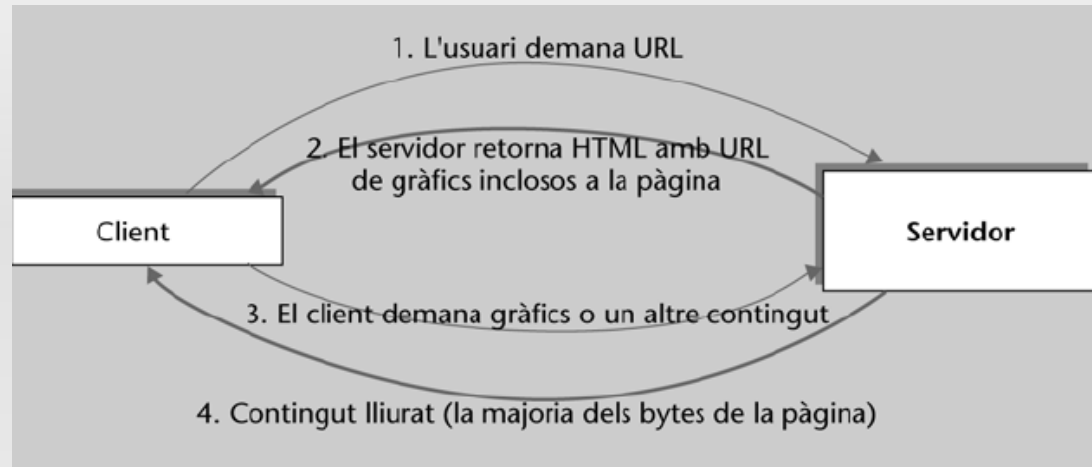
- **Xarxes de distribució de continguts (CDN – content delivery networks)**
  - És com un servei de “logística”: la **CDN** s'encarrega de mantenir còpies a prop dels clients als seus propis magatzems (no en un punt central sinó als extrems de la xarxa). Per això, ha de disposar d'un gran nombre de punts de servei en multitud de proveïdors d'Internet.
  - Es poden trobar diverses empreses que ofereixen aquest servei: **Akamai** és la més important
  - <http://spanish.akamai.com/enes/html/technology/nui/news/sites.html>

# Continguts distribuïts

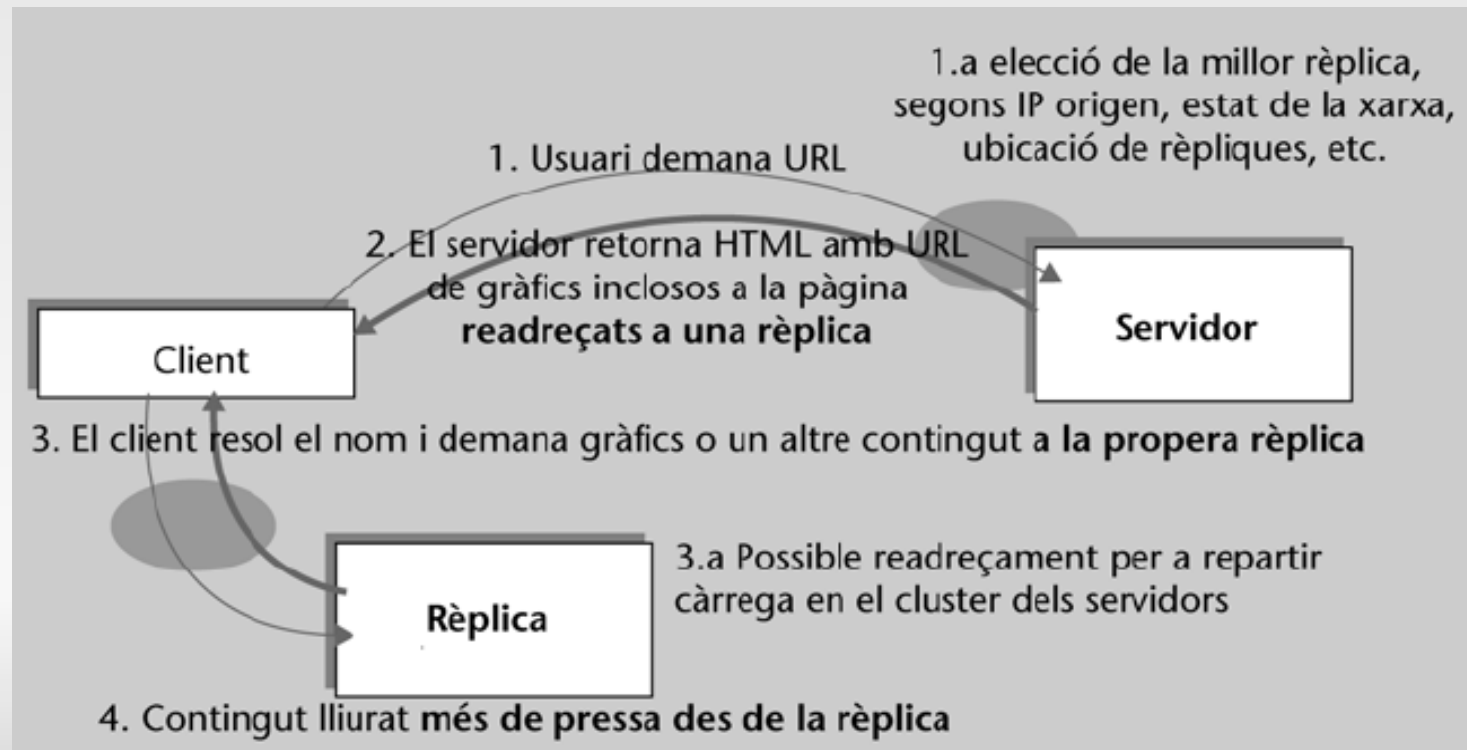
- **Xarxes de distribució de continguts (CDN – content delivery networks)**
- La CDN pot actuar sobre el següent:
  - El **DNS**: quan es resol un nom, el servidor DNS respon, segons la ubicació de l'adreça IP del client;
  - El **servidor web**: quan es demana una pàgina web, es reescriuen els URL interns segons la ubicació de l'adreça IP del client;

# Continguts distribuïts

- Petició normal



- Petició CDN



# Continguts distribuïts

- L'usuari navega a <http://www.adobe.com> i el navegador resol el nom en DNS (192.150.14.120):
  1. L'usuari demana l'URL a la IP 192.150.14.120.
    - 1.a El servidor web decideix la millor rèplica, segons l'IP d'origen, estat de la xarxa, ubicació de les rèpliques
  2. El servidor torna l'HTML amb l'URL dels gràfics inclosos a la pàgina redirigits a una rèplica.  

  3. El client resol el nom i demana gràfics o un altre contingut a la rèplica més propera.
  4. Contingut lliurat més de pressa des d'una rèplica.



# Continguts distribuïts

