

A decorative graphic consisting of a thin yellow circle on the left and a horizontal bar with a yellow-to-white gradient on the right, framing the title text.

UD4. ADMINISTRACIÓ DE SERVIDORS D'APLICACIONS

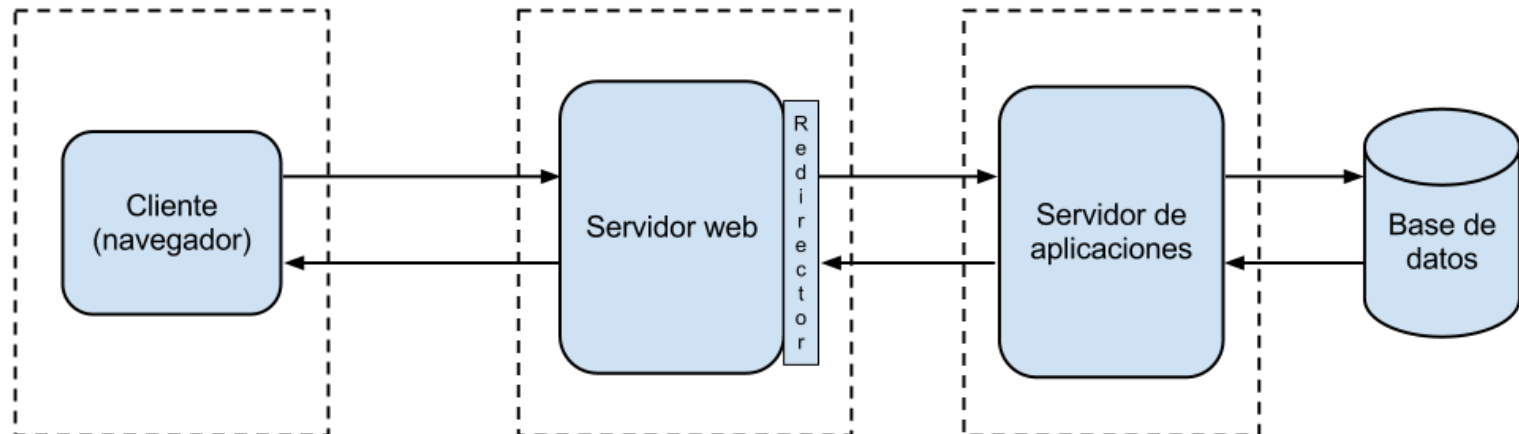
Desplegament d'aplicacions web
2on **DAW**

[0. ÍNDEX]

- INTRODUCCIÓ
- ARQUITECTURA DE SERVIDORS D'APLICACIONS
- ARQUITECTURA D'APLICACIONS WEB. STACK
- HTTP: PROTOCOL SENSE ESTAT (stateless)
- SESSIONS APLICACIÓ WEB

1. INTRODUCCIÓ

- El concepte de **servidor d'aplicacions** és posterior al concepte de **servidor web**.
- A l'aparèixer les primeres **tecnologies** de generació de contingut **web dinàmic** (CGI, PHP, ASP, JSP...) **apareix** el concepte de **servidor d'aplicacions web**.

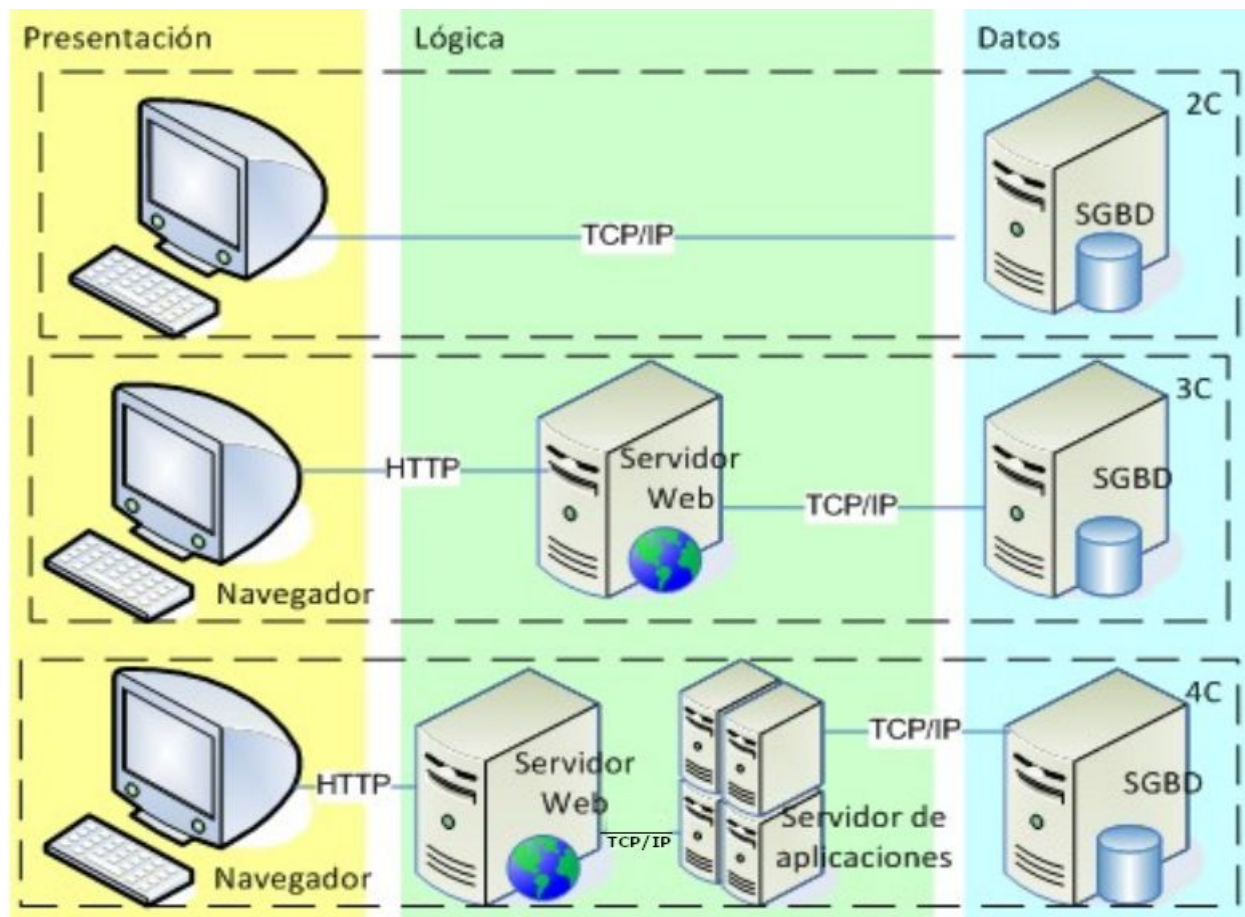


1. INTRODUCCIÓ

- **Actualment és difícil** distingir la **frontera** entre **servidor web** i servidor d'aplicacions. Podem destacar les següents **característiques**:
 - Sistemes **d'autenticació** (seguretat)
 - Gestió de **sessions d'usuari**.
 - Accés als **components** o **llibreries** de la plataforma utilitzada (**PHP, Java, .NET...**)
 - Gestió de les **connexions amb motors de bases de dades**)
 - En alguns casos implementen serveis com **clustering, load-balancing o fail-over**.
 - *Monitoratge del servei, gestió de processos, estadístiques...*

1. INTRODUCCIÓ

■ Evolució



2. SERVIDORS D'APLICACIONS

- Els servidors d'aplicacions estan molt lligats a la tecnologia utilitzada per al desenvolupament de l'aplicació.
 - **ASP.Net:** Aquesta tecnologia (.NET Framework), fa servir el mòdul *ASP.NET core* al costat del servidor **Internet Information Server (IIS)**.



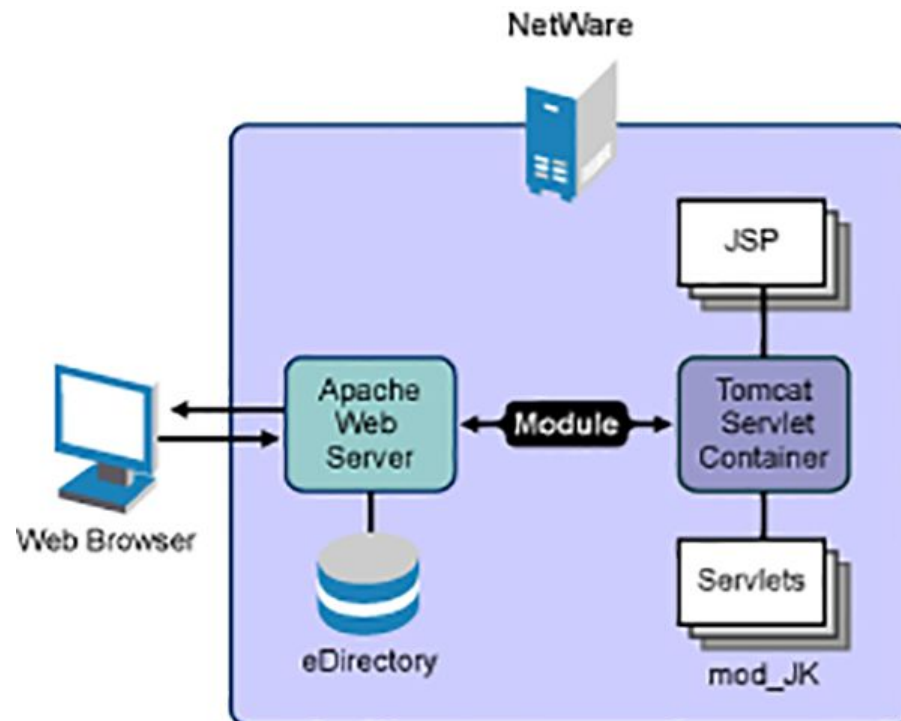
2. SERVIDORS D'APLICACIONS

- **JSP/Servlets:** La tecnologia Java compta amb diferents servidors d'aplicacions que no necessiten la interacció amb servidors webs com són:
 - **JBoss:** també anomenat **WildFly** es tracta d'un servidor d'aplicacions de **codi obert** basat en l'especificació Java EE (Java Enterprise Edition) i escrit completament en Java
 - **Oracle (BEA) Weblogic:** Servidor d'aplicacions basat en Java EE i també servidor web http. (licència propietària d'oracle)
 - **Websphere application server:** Servidor d'aplicacions d'**IBM** basat en Java EE.



2. SERVIDORS D'APLICACIONS

- **Apache tomcat:** integrat amb el servidor web Apache funciona com un contenidor de Servlets. Implementa les especificacions dels *Servlets* i de *Java Server Pages* (JSP).



2. SERVIDORS D'APLICACIONS

■ PHP

- La tecnologia **PHP** fa ús d'un **servidor web**, generalment **apache** o **Ngnix** i d'un **intèrpret de PHP**. Podem encontrar principalment 2 tipus d'execució de les aplicacions:
 - **Apache** amb l'activació d'un mòdul intern (**mod_php**) que s'encarregarà de la interpretació de les pàgines PHP.
 - **Apache / Ngnix** + servidor d'aplicacions (**phpFPM**): El servidor web s'encarregarà d'atendre totes les peticions i confiarà amb un tercer servidor per a la interpretació de les aplicacions escrites en **PHP**

2. SERVIDORS D'APLICACIONS

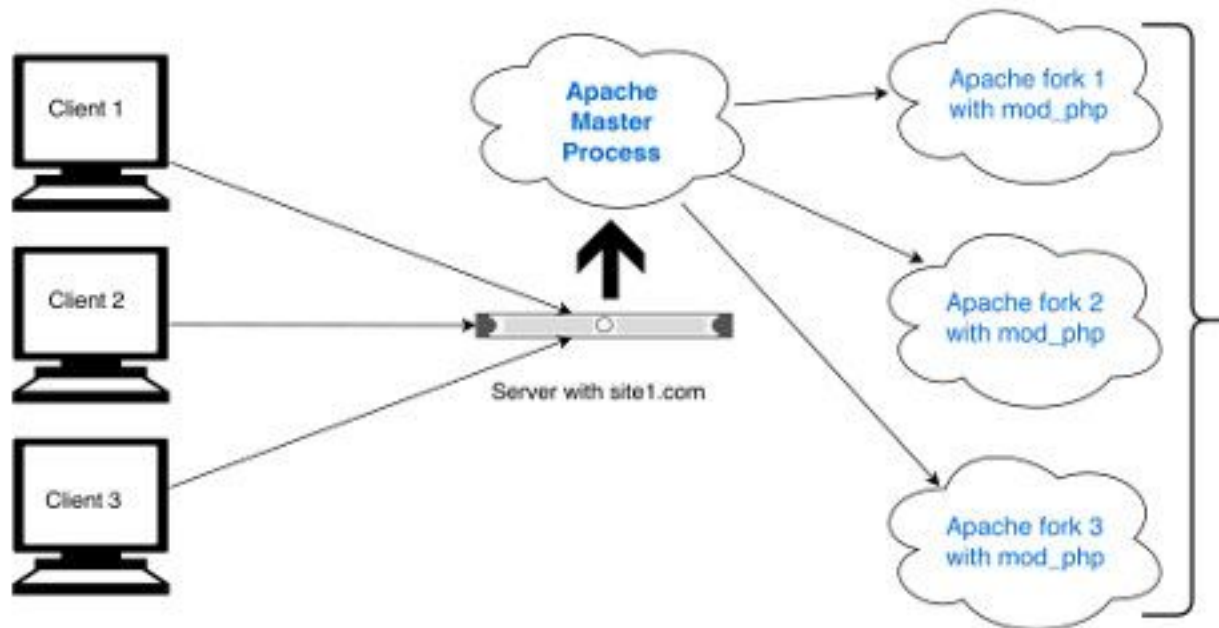
■ PHP

- **Apache** amb mòdul intern: es tracta del mètode més antic i ràpid, sempre que el nostre servidor **no atenga moltes** peticions **concurrents**.
 - El mòdul PHP es carrega a cada petició.
 - **No** podem **limitar** ni **restringir recursos** per a cada aplicació.
 - No podem aplicar diferents configuracions per a cada aplicació.
 - Cada aplicació pot comprometre la **seguretat** de les altres. *(Totes les aplicacions s'executaran amb el mateix usuari).*

2. SERVIDORS D'APLICACIONS

■ PHP

- **Apache** amb mòdul intern

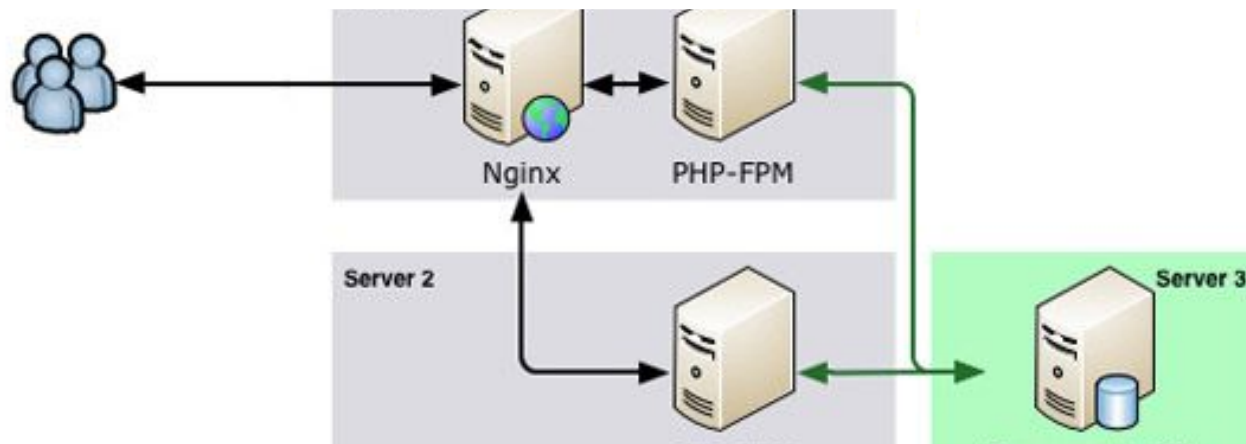


2. SERVIDORS D'APLICACIONS

■ PHP

○ Servidor web (Apache/nginx) + PHP-FPM

- El Servidor web i d'aplicacions són processos independents que poden estar al **mateix maquinari** o en **maquinaris diferents**.
(Comunicació a través de la xarxa)



2. SERVIDORS D'APLICACIONS

■ PHP

○ Servidor web (Apache/nginx) + PHP-FPM

- El **servidor web** es comunica amb el servidor d'aplicacions mitjançant una versió millorada de la **tecnologia Fast CGI** (*Fast Common Gateway Interface*).
- Es tracta d'un **protocol estàndard** que **habilita la comunicació** entre 2 processos.
 - *Independència del llenguatge utilitzat.*
 - *S'utilitza un procés separat.*
 - *És possible l'execució en maquinari separat.*

2. SERVIDORS D'APLICACIONS

■ PHP

○ Servidor web (Apache/nginx) + **PHP-FPM**

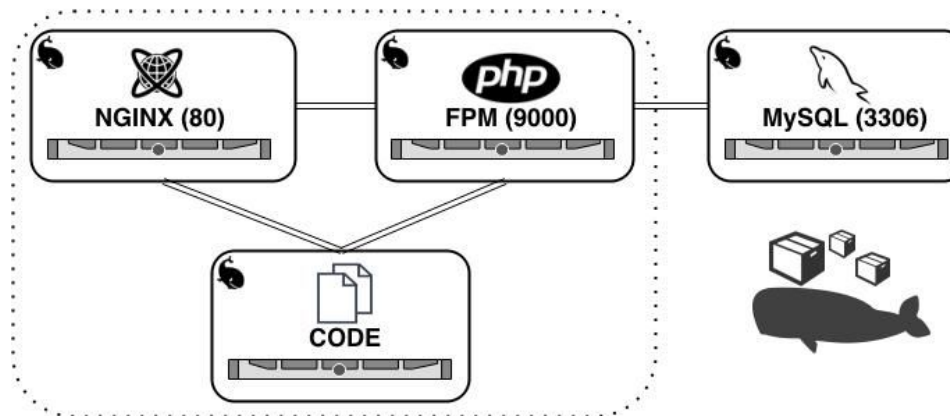
- El servici de **PHP-FPM** permet establir un o més processos persistents, amb quantitats específiques de recursos, que es mantenen a l'espera d'atendre peticions mitjançant un **socket**.
- Permet establir diferents configuracions
 - Ports d'escolta.
 - Tipus de socket Unix/TCP.
 - Usuaris d'execució.
 - Mòduls/Llibreries de PHP a carregar
 - Diferents dominis de seguretat.



2. SERVIDORS D'APLICACIONS

■ PHP

- **Servidor web (Apache/nginx) + PHP-FPM**
 - Permet mantindre **diferents fitxers de log** per a cada aplicació.
 - **Monitoratge i estadístiques** dels processos engegats.
 - Permet **aplicar configuracions i reiniciar el servei** de forma aïllada



2. SERVIDORS D'APLICACIONS

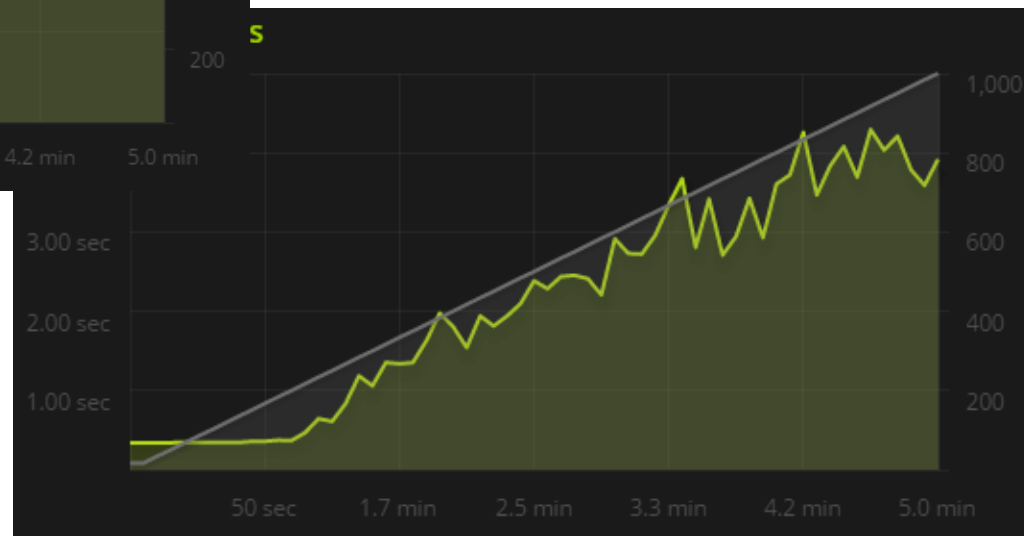
■ PHP

○ Servidor web (Apache/nginx) + PHP-FPM

PHP-FPM



mod-php



2. SERVIDORS D'APLICACIONS

■ Python

- Igual que l'anterior tenim diferents formes d'execució d'un script en **Python**:
 - Mitjançant un **mòdul intern a Apache**;
 - **libapache2-mod-wsgi** → Python 2
 - **libapache2-mod-wsgi-py3** → Python 3
 - Mitjançant un **servidor WSGI extern**
 - En aquest cas, el **servidor web** actua com a un Reverse proxy, que reenvia les peticions al servidor WSGI (**gunicorn**)



<https://www.digitalocean.com/community/tutorials/a-comparison-of-web-servers-for-python-based-web-applications>

2.1 ENTORN DE DESENVOLUPAMENT

- La majoria de frameworks i/o tecnologies de desenvolupament web, ens proporcionen un servidor d'aplicacions per a ***l'entorn de desenvolupament***.
 - L'interprete PHP → `php -S localhost:8001`
 - Symfony (PHP) → `Symfony Local Web Server`
 - Django (Python) → `Django Web Server`

"Aquests servidors fan ús de l'interprete del llenguatge disponible de forma global al sistema operatiu.

Qualsevol configuració sobre l'interprete es farà en l'àmbit global"

3. ARQUITECTURA APLICACIONES WEB. STACK

- Anomenen "**Stack**" al conjunt de tecnologies triades com a eines per a la implementació de la solució d'un projecte.
- Generalment s'usen per a donar una descripció de l'arquitectura d'una manera ràpida. Podem destacar les següents:
 - LAMP (**L**inux - **A**pache - **M**ySQL - **P**HP)
 - LEMP (**L**inux - **E**ngine X - **M**ySQL - **P**HP)
 - MEAN (**M**ongo - **E**xpressJS - **A**ngular - **N**ode JS)

NGINX

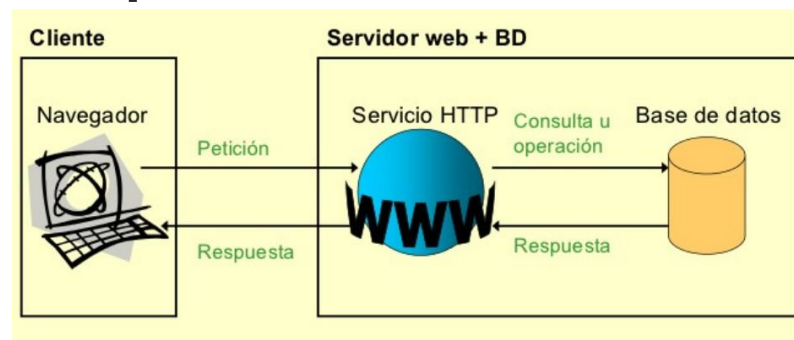
node JS

mongoDB



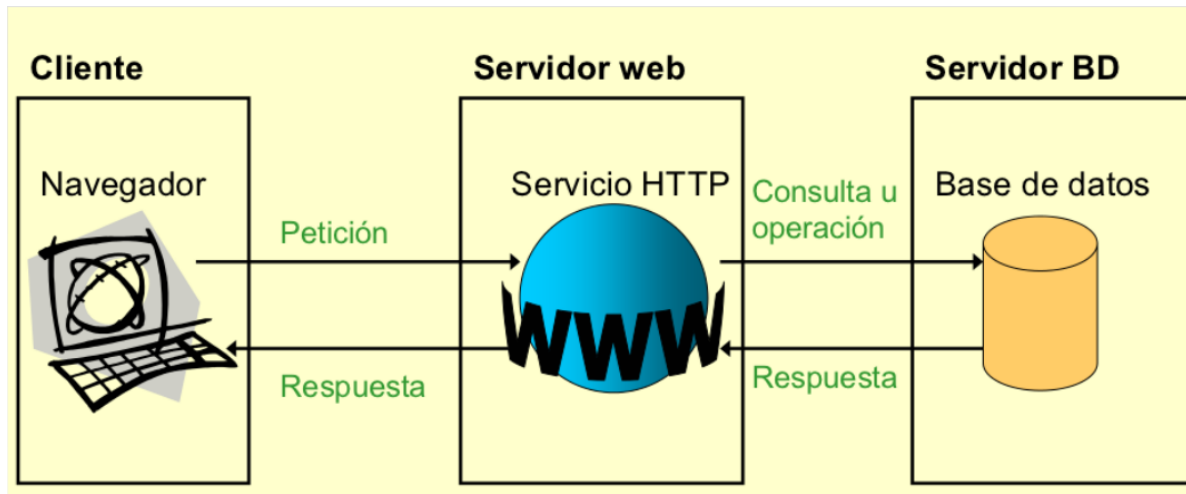
3. ARQUITECTURA APLICACIONES WEB. STACK

- Podem trobar diferents distribucions de l'**stack** en funció del nivell d'escalabilitat que necessitem
 - Servidor d'aplicacions i base de dades al **mateix maquinari**.
 - **Més ràpid.**
 - No podem **escalar** horitzontalment el servici.
 - **No** podem **compartir el SGBD** amb diferents aplicacions. (+ Gestió i manteniment)
 - **No independència de serveis.** (Seguretat / Failover)



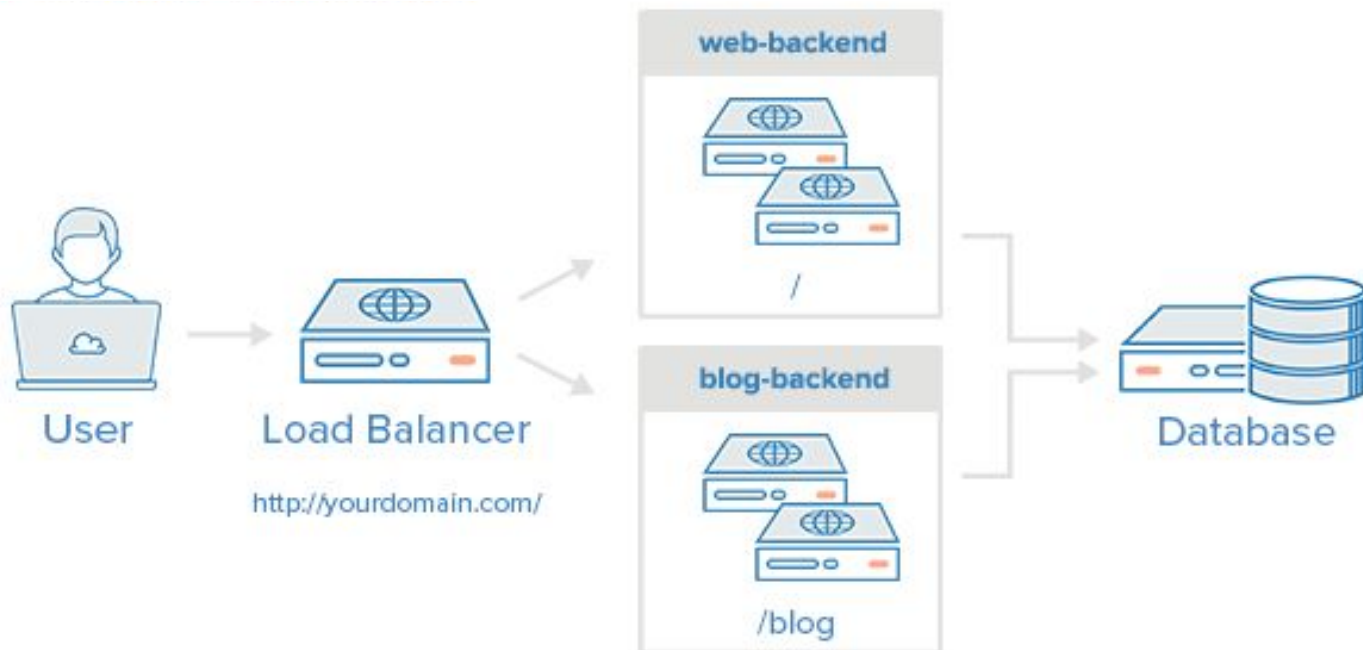
3. ARQUITECTURA APLICACIONES WEB. STACK

- Servidor d'aplicacions i bases de dades en **diferents hosts**:



3. ARQUITECTURA APLICACIONES WEB. STACK

- Servidor d'aplicacions i bases de dades en **diferents host:**



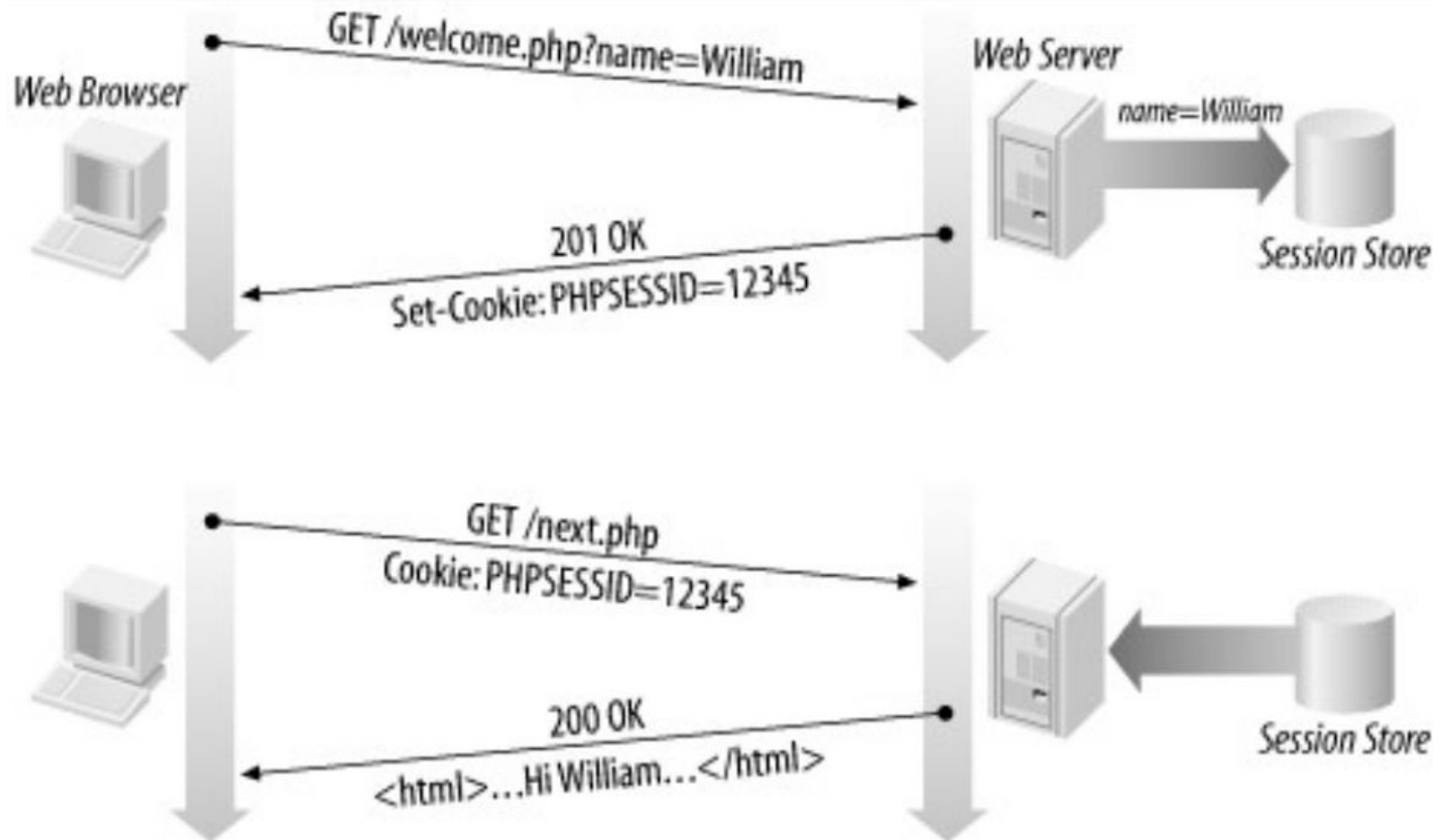
4. HTTP: PROTOCOL SENSE ESTAT

- El **protocol HTTP** no manté **l'estat** entre peticions.
- A vegades, en el **context d'una aplicació web** es fa necessari mantindre l'estat entre peticions HTTP que es produeixen entre el client (navegador) i el servidor web.
 - **Saber** si un **usuari** ha fet "**login**" a una aplicació.
 - Mantindre **informació** sobre accions realitzades per usuari.

4.1 GESTIÓ DE SESSIONS D'UNA APLICACIÓ WEB

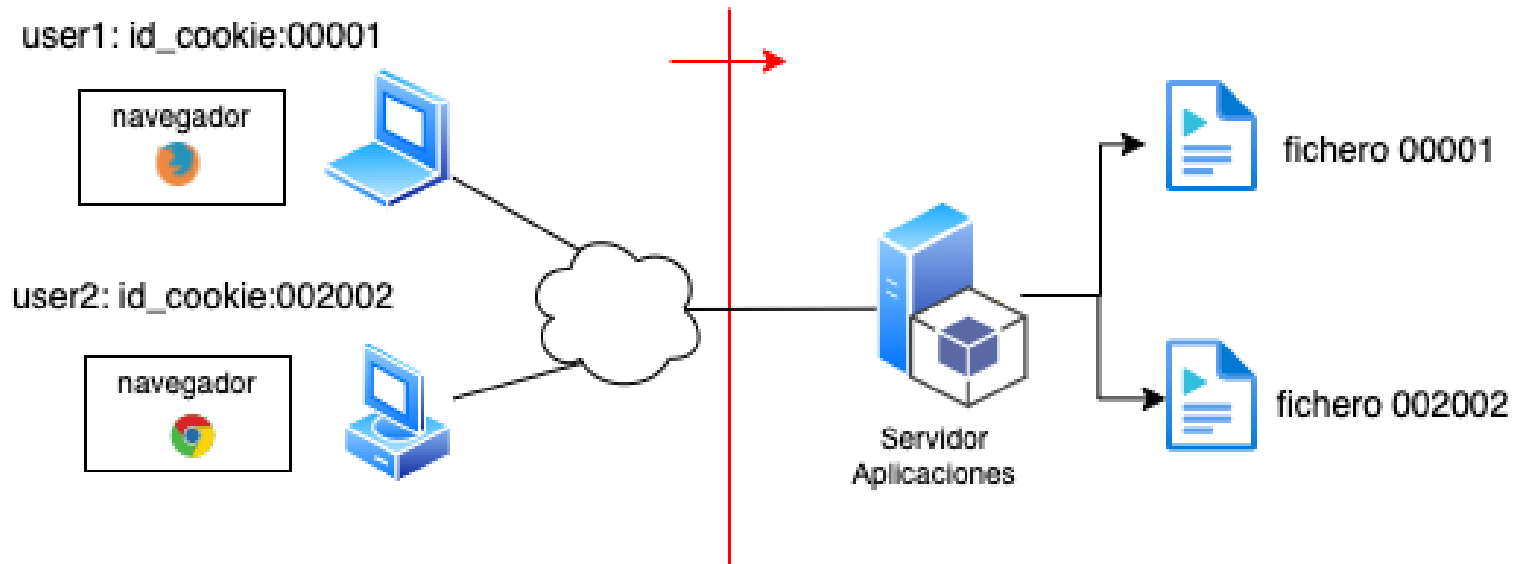
- Les **sessions** són un **mecanisme que utilitzen** els servidors web per guardar informació sobre l'usuari i la seua activitat.
- Una sessió **permet establir** una **comunicació** entre el **client** i el **servidor mantenint un estat** (conjunt de variables).
 - S'aconsegueix assignant a cada client un **ID de sessió** a l'iniciar la comunicació.
 - Aquest **ID** ha de circular entre el client i el servidor en totes les peticions HTTP.
 - El més habitual és **utilitzar una galeta**.

4.1 GESTIÓ DE SESSIONS D'UNA APLICACIÓ WEB



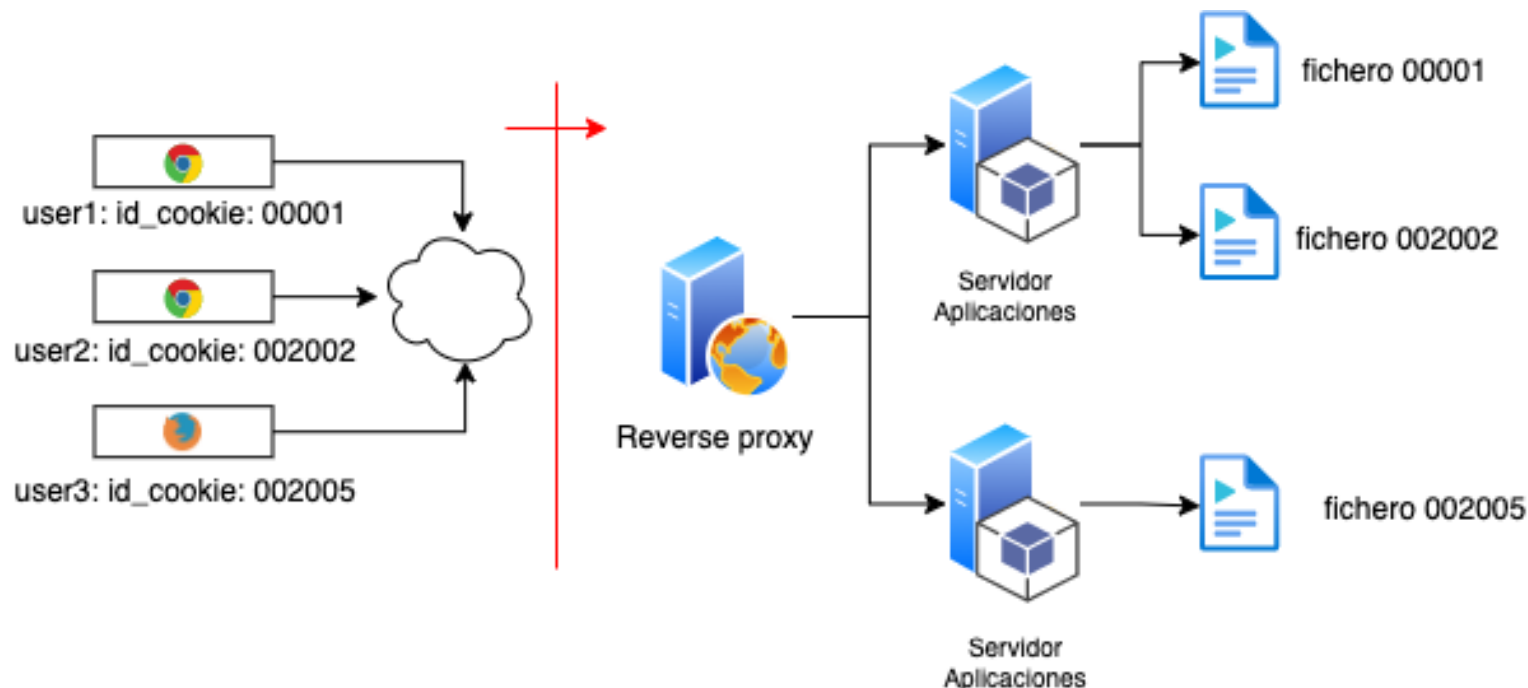
4.1 GESTIÓ DE SESSIONS D'UNA APLICACIÓ WEB

- El servidor d'aplicacions associarà cada **id de sessió** amb un arxiu en el servidor on **emmagatzemarà** la informació.
 - La seua lectura **penalitzarà** el **rendiment** del servidor.



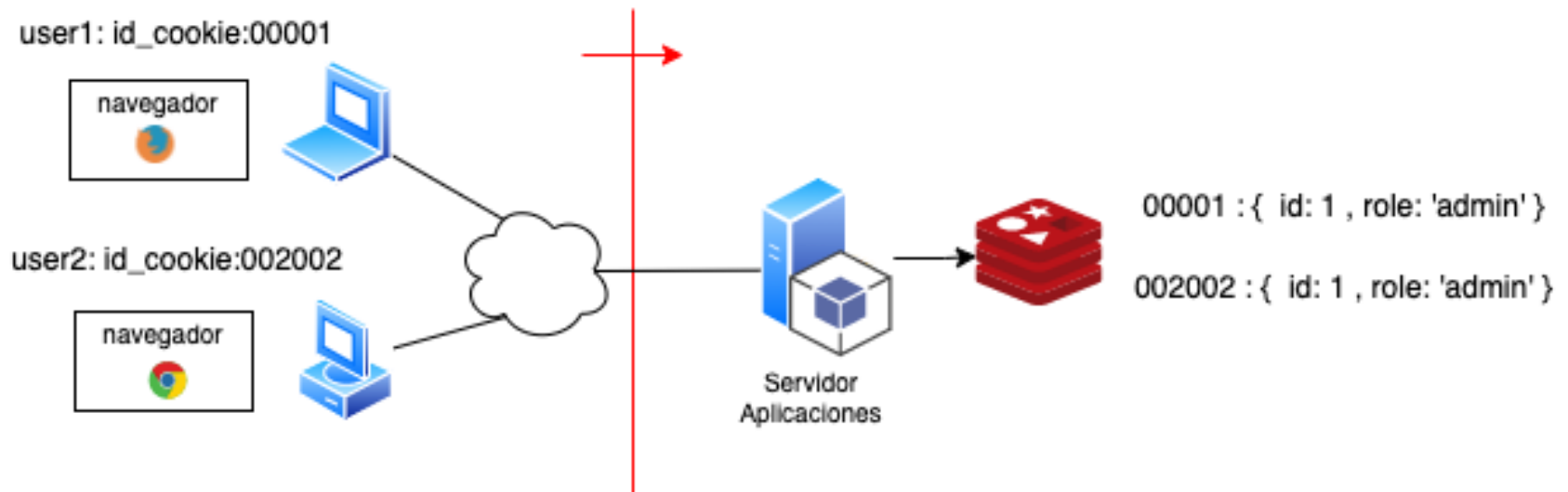
4.1 GESTIÓ DE SESSIONS D'UNA APLICACIÓ WEB

- Els **arxius** que guarden la informació de les sessions han de **gestionar-se** (esborrar quan no són necessàries) i tindre's en compte quan necessitem escalar el servei.



4.1 GESTIÓ DE SESSIONS D'UNA APLICACIÓ WEB

- Podríem decidir guardar-les a un sistema de base de dades en memòria (Redis, memcache,...)
 - Al fitxer **php.ini** podem configurar diferents paràmetres sobre el comportament del nostre servidor amb les sessions.
 - També es pot fer en l'àmbit d'aplicació



[4.1 GESTIÓ DE SESSIONS D'UNA APLICACIÓ WEB]

- Això és tot... de moment :-)