

# Serverteknologi webserver

Fagnr.: 16860



# Målpinde

1. Eleven kan installere og konfigurere en Web-server og tilhørende plugins i forhold til en given opgave.
2. Eleven kan konfigurere sikkerhed og brugerspecifikke tilladelser .
3. Eleven kan installere supplerende plugins på en Web-server, som f.eks. CGI og Java Servlets.
4. Eleven kan opsætte HTTP og HTTPS.
5. Eleven kan installere og konfigurere værktøjer til administration af hostede løsninger svarende til et web host panel suite, som bl.a. indeholder SQL implementering.
6. Eleven kan opsætte og anvende DNS, Mail server, DHCP, Web-server samt webmail.
7. Eleven kan anvende Pre- og Post-routing, samt opsætte NAT.
8. Eleven kan opsætte Firewalls via terminal.
9. Eleven kan forklare og opsætte traditionelle og transparente Proxy som f.eks. Squid, samt opsætte Firewalls, som f.eks. Iptables, FirewallD eller PFSense, sammen med Squid.

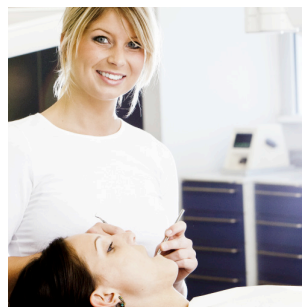
## Målpinde II

- Avanceret:  
Eleven kan, ud fra en case-opgave selvstændigt designe, planlægge, installere, konfigurere og dokumentere en avanceret Web-serverløsning, der omhandler de beskrevne mål, og kan herigennem demonstrere viden, færdigheder og kompetencer, der ligger ud over de i faget beskrevne mål og målniveauer.
- Ekspert:  
Eleven kan, ud fra en case-beskrivelse, indeholdende løsning af en kompleks arbejdsopgave, selvstændigt designe, planlægge, installere, konfigurere og dokumentere en samlet kompleks Web-serverløsning, der omhandler de beskrevne mål, og kan herigennem demonstrere viden, færdigheder og kompetencer, der ligger ud over de i faget beskrevne mål og målniveauer. Endvidere kan eleven begrunde de valgte løsninger og fremvise evner til at tilrettelægge og styre arbejdsprocessen.

# Webserver

## Definition

- Server software der kan besvare www-client requests.
- Kan som regel indeholde én til flere websites.
- Behandler indkommende forespørgsler via HTTP protokollen.



# Webserver

## Primære funktion

- Opbevare, bearbejde og levere web sider til klienter.
- Kommunikation med klienter foregår via HTTP.
- Leverede sider er primært HTML sider indeholdende billeder, stylesheets og scripts udover tekstindhold.

## Andre funktioner

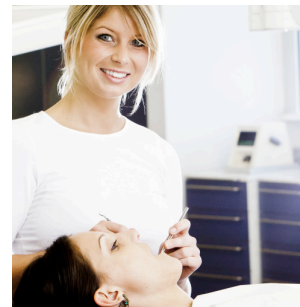
- Modtage indhold fra klienter





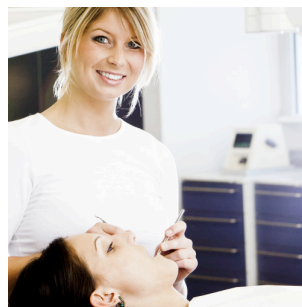
# User agents

- Browsers, web-crawlers ...
- Initierer kommunikation gennem en forespørgsel på en specifik ressource og serveren besvarer forespørgslen med vedkommende ressource eller en fejlmeddelelse.  
Ressourcen er typisk en fil lagret på serveren.

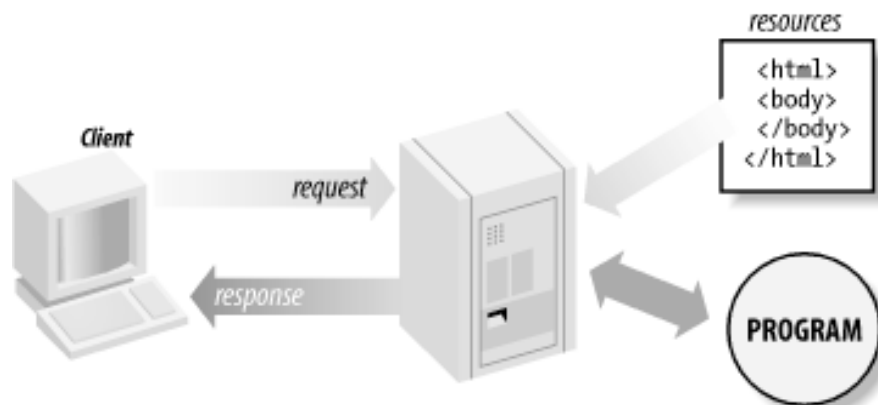


# Dynamik

- Mange servere understøtter server side scripting
  - ASP, PHP ...
  - HTML dokumenterne skabes dynamisk
  - Langsommere end statiske sider



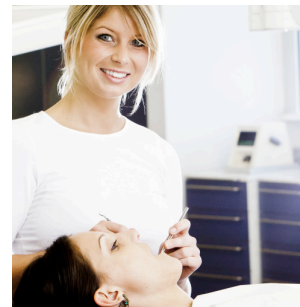
# Request/Response pattern





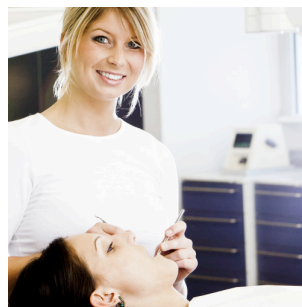
# HTTP

- Stateless
  - Serveren har ingen informationer om klienten og har således ingen viden om, at flere requests kan komme fra samme klient.
- Enhver interaktion mellem server og klient kræver en request/response udveksling.



# Historie

- 1989
  - Tim Berners-Lee foreslår et nyt projekt til sin arbejdsgiver CERN
- 1990
  - En webbrowser "WorldWideWeb"
  - Verdens første webserver "CERN httpd"
- 1994
  - Berners-Lee opretter World Wide Web Consortium (W3C)



# Path translation

En webserver kan mappe en URL til:

- En lokal filsystemsressource
  - Statisk
- Et internt eller eksternt program
  - Dynamisk



# Path translation

Flg. URL er et eksempel på en indtastning i en webbrowser:

`http://www.example.com/path/file.html`

Browseren oversætter til flg. HTTP request:

`GET /path/file.html`

`Host: www.example.com`

Webserveren vil lede efter den angivne path i sit root dir (på en Apache server ofte `/var/www` ) altså:

`/var/www/path/file.html`

Dernæst læser serveren filen og sender en respons til klienten indeholdende en beskrivelse af filen foruden selve filen

- eller en fejlmeddelelse.

# Markedsandele

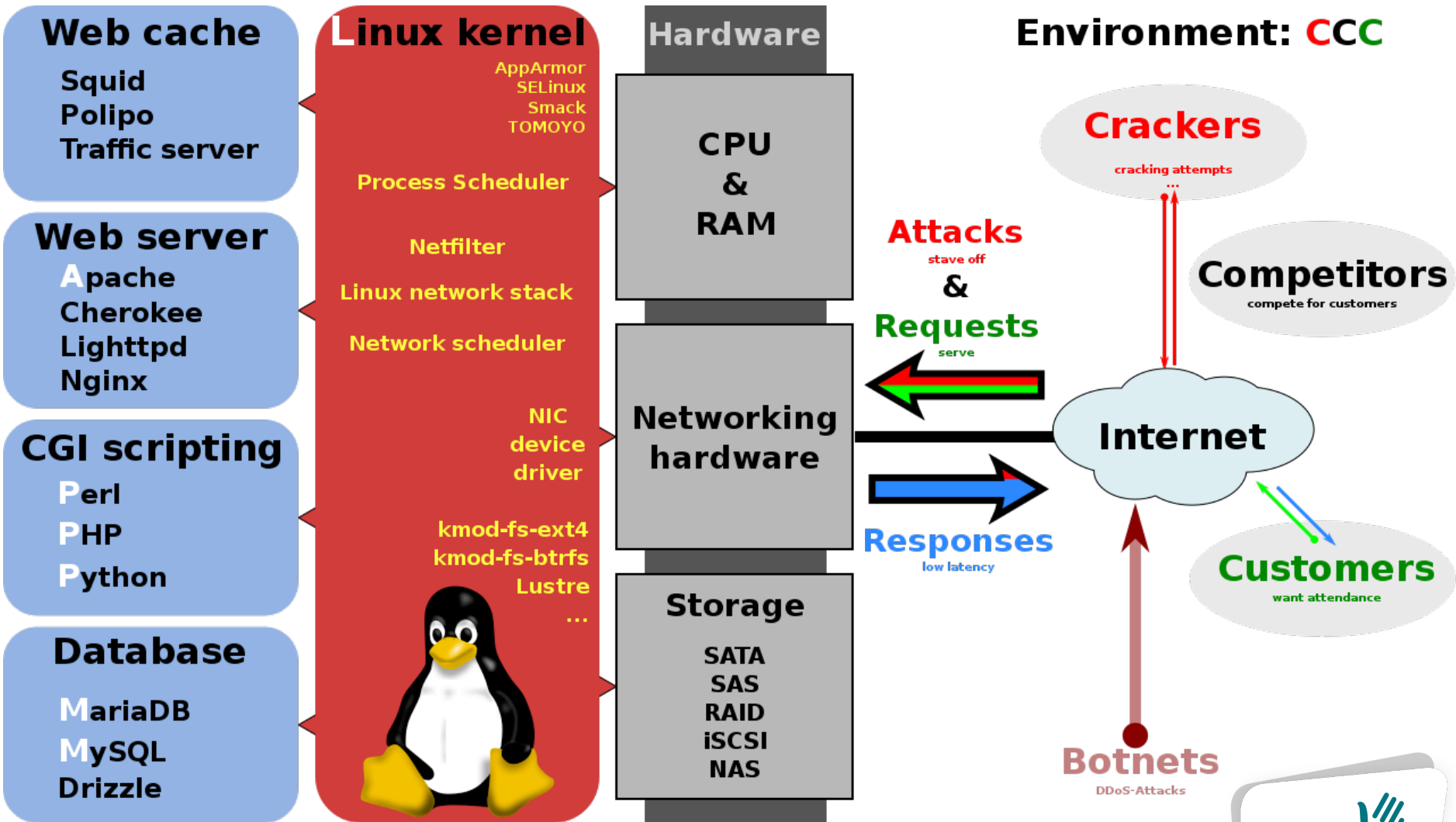
Februar 2019

Produkt	Vendor	Procent
Apache	Apache	44,3%
Nginx	NGINX, Inc.	41,0%
IIS	Microsoft	8,9%
LiteSpeed Web Server	LiteSpeed Technologies	3,9%
GWS	Google	0,9%
Øvrige		< 1,0%





# LAMP Stack



## Case - KAPOW!

Kampsportcenteret "Kapow" ønsker at få opsat en webserver.

Kapow tilbyder i øjeblikket træning i MBA og karate, men det er tanken løbende at udvide med nye aktiviteter efterhånden som centerets bygninger bliver færdiggjorte.

Webserveren skal til en start hoste domænerne "kapow.ninja", "kapow-mba.ninja" og "kapow-karate.ninja".

På hvert af disse domæner ønskes et særligt afsnit "intra", som skal password-beskyttes.

Efter rådslagning med deres webudvikler, som udvikler i PHP/MySQL har Kapow besluttet, at de ønsker løsningen opsat på en Linux/Ubuntu server. Udvikleren skal kunne tilgå de relevante områder på serveren via FTP.

I første omgang har man besluttet, at man ønsker at bibeholde nogle ældre tilmeldingsscripts, som er udviklet i Perl vha. nogle CGI-teknologier, så det er vigtigt, at disse kan eksekveres på serveren.

Det er tanke, at kommunikation mellem medlemmerne på sigt kommer til at foregå via www, så sikkerheden mht. persondata ligger Kapow stærkt på sinde.

# Opgave

- Gennemlæs case'en og opstil en kravsspecifikation på punktform, der kan løse case'en.
- Aflever kravsspecifikationen til godkendelse.
- Ud fra den godkendte kravsspecifikation opsættes en fungerende webserver. Der skal løbende udarbejdes en procesrapport på max. 5 sider indeholdende log over arbejdet.
- Aflevering:
  - Procesrapporten afleveres i pdf-format
  - Serveren fremvises og dens funktionalitet demonstreres

