# Client-Server arkitektur

Client-Server er en arkitektur af et computernetværk, hvori mange klienter laver request til en central server(host).   
Klient siden indeholder et interface / gui, også kaldet front-end, hvorpå der kan laves request til serveren, og et interface til at fremvise den returnerede data.  
Serveren venter på requests fra klienten og svare på dem. Klienten har nødvendigvis ingen kendskab til hvordan serveren fungere, dette kaldes også back-end.  
Klienter er for det meste en arbejdsstation eller en personlig computer, hvorimod en server er en kraftigere computer der kan håndtere en masse trafik.  
Client-Server ariktekturen er effektiv i situationer hvor klienten og serveren har forskellige arbejdes fordelinger, fx På en hospitals computer, kan klient computeren køre en applikation hvori en bruger kan indtaste patient informationer, og server køre et andet program der står for at inddatere disse informationer i en database[[1]](#footnote-1).

## Fordele og Ulemper[[2]](#footnote-2)

### Fordele

Med Client-Server arkitekturen er der mange fordele som både brugerne og virksomheder kan udnytte.

* **Data deling**

Det er muligt for flere klienter at til gå den data der ligger på serveren på samme tid. Der kan tilføjes og læses data på samme tid.

* **Tilgængelighed**

Det er muligt for alle klienter at tilgå den data de ønsker, uden at tilslutte sig en specifik server igennem en terminal eller programmer som fx Teamviewer.  
Klienten kan også tilgå data fra alle steder, så længe de er tilsluttet internettet, hvilket giver en fordel for firmaer der har rejsende medarbejder, eller hjemme arbejde.

* **Forskellige systemer**

Om du er på en mac, pc eller andet. Er det fri mulighed for at tilgå serveren. Du skal blot have internetadgang, så kan man tilgå serveren. Dette gør at man ikke skal tage forbehold for hvilke systemer der skal bruge serveren.

* **Vedligeholdelse**

Med denne arkitektur kan man nemt opgradere, reparere, eller udskifte en server uden at det påvirker klient siden. Det er dog nødvendigt at have flere servere for at systemet skal fungere hvis man deaktivere én eller flere servere.

* **Sikkerhed**

Serveren har bedre kontrol over dataene, og at det er autoriserede bruger der kun har adgang til dataene.

### Ulemper

Selvom der er mange fordele ved Client-Server arkitekturen, har den også sine ulemper.

* **Server overload**

Hvis der er meget trafik på netværket, kan serveren blive overloaded. Hvilket betyder at den ikke kan følge med til alle de request der bliver kaldt fra klient siden, og derved får klienten en dårligere/langsommere oplevelse. Som når Skat, udgiver forskudsopgørelsen og hele den danske befolkning på samme tid vil tilgå deres data.

* **Downtime**

Hvis en vigtig server brænder sammen eller lign. Vil klientens request ikke blive behandlet og systemet går offline. (Dette kan løses ved at have backup servere)

## Client-Server med Budgetmanager

I vores system vil det give god mening at bruger client-server arkitekturen, da klienter skal kunne inddatere deres budgetter og hente informationer ud på samme tid.  
Derved har vi en front-end, klient side, og et back-end, server side. Som kan håndtere de nødvendige opgaver, der skal til for at systemet fungere bedst muligt.  
Ved at vi laver vores Budgetmanger i MVC, har vi udfyldt client-server arkitekturen, ved at brugeren ser deres ønskede views, og laver nogle requests der bliver behandlet af vores controllers. Dataene vil blive inddateret i en SQL database ved hjælp af nogle kald fra vores controller, her ved har vi en client-server arkitektur.

1. https://www.britannica.com/technology/client-server-architecture [↑](#footnote-ref-1)
2. http://clientserverarch.blogspot.dk/2013/03/advantages-and-disadvantages-of-client.html [↑](#footnote-ref-2)