Teknologi tilføjelser:

### HTTP

HTTP står for HyperText Transfer Protocol. Denne protokols funktion er anmodning om ressourcer fra “user agenten”. Dette forekommer ofte i form af HTML dokumenter, billeder og andre medier. En user agent(ofte webbrowseren) kan foretage requests e.g funktioner som “GET eller POST”. GET kendes også ved navnet “fetch”. HTTP foregår i applikationslaget i TCP/IP lag-modellen.

Et forløb for HTTP kunne være således. Klienten(user agent) initierer et request til serveren, som specifikt er et fetch request for en HTML dokument ressource. Serveren besvarer med en HTTP “Response” der indeholder et HTML dokumentet. herefter fortsætter processen ud fra modtagende ressource(r), hvis der kræves flere elementer, for at kunne konstruere HTML dokumentet.

HTTP kommunikerer primært vha. TCP, men kan teoretisk fungerer vha. andre protokoller da HTTP er åben for extension af nye funktioner.

HTTP Requests består primært af 5 elementer:

* **Methods** (GET, HEAD, POST, PUT, DELETE, TRACE, OPTIONS, CONNECT)
* **Path**
* **Version of Protocol**  (Versionen af HTTP der benyttes, HTTP/1.1 etc)
* **Headers**
* **Body(optional)**  (Indeholder ressourcer, hvis metoden kræver det)

HTTP Responses består primært af 5 elementer:

* **Version of Protocol** (Versionen af HTTP der benyttes, HTTP/1.1 etc)
* **Status Code**
* **status Message**
* **Headers**
* **Body (optional)** (Indeholder ressourcer, hvis request kræver det)

### Proxies

Proxies forekommer som et mellemled i kommunikationen imellem en user agent og en server. Proxy’en forwarder requesten fra user agenten, hvor der kan forekomme ændringer på requesten. Proxies varetager ofte opgaver såsom: caching, filtering, load balancing, logging og authentication.

### Sockets (udkast)

Sockets er en kombination af en IP adresse og en port, som giver applikationer muligheden for at tale med andre computere/enheder.

Pointen med sockets er at sikre kommunikationen der modtages på enheden kan differentieres. e.g en enhed kan modtage flere forskellige informationer på samme tid, (steam download samtidig med at der streames/browses på nettet). vha. socket opsætning kan programmet differentiere imellem modtaget data, da den kun lytter til data der modtages via en specificeret port.

### API

En API (Application Programming Interface) er en måde man kan stille funktionalitet til rådighed for tredjeparts brugere. Det kan også være en måde at stille data til rådighed. Lad os f.eks. sige at vi, på vores hjemmeside, gerne ville vise vejret. Så kunne vi via en vejr-API (hvis sådan er stillet til rådighed) “bare” binde vores hjemmeside op på denne API. Altså slipper vi for at tænke på hvordan funktionaliteten i vejr-appen er skruet sammen. Vi er kun interesseret i udbyttet.