* Adressering af MAC og IP adresse
  + MAC adressen bliver brændt ind på chippen fra fabrikanten af.
  + IP adressen bliver tildelt netværksenheden af DHCP serveren.
* Netværkspakker
  + Også kaldet PDU’er er lidt ligesom et brev eller en pakke, den har i Afsender og modtager på sig i form af IP adressen. MAC adressen bruges til at fortælle hvor pakken skal hen næste gang for at komme til dens destination. Når vi pusher til github bliver der sendt en PDU fra vores computer til Githubs server.
* Protokoller
  + Der bliver selvfølgelig anvendt en række protokoller for at facilitere netværks traffikken, selvom vi ikke har direkte kontakt med disse protokoller kan det være gavnligt at kende lidt til dem, især HTTP da det er den protokol vi har en chance for reelt set at komme i kontakt med igennem en HttpContext.
* WAN
  + Wide Area Network, i bund og grund et andet term for internettet som vi kender det, eftersom vores produkt er hostet og tilgængeligt på internettet er det klart at vi arbejder med WAN
* HTTP
  + Hyper Text Transfer Protocol, er den lag 7 protokol som oftest anvendes til at kommunikerer med webservers. Hyper text kommer fra at det ikke blot er text som kan sendes med HTTP, men også filer. Beskeder der sendes med HTTP kaldes også et request, og et requests indeholder ikke kun data, men også en header der beskriver dataen, og andet indformation omkring kommunikationen, sådan en header kan se ud som følgende:
* Date: Mon, 23 May 2005 22:38:34 GMT
* Content-Type: text/html; charset=UTF-8
* Content-Length: 138
* Last-Modified: Wed, 08 Jan 2003 23:11:55 GMT
* Server: Apache/1.3.3.7 (Unix) (Red-Hat/Linux)
* ETag: "3f80f-1b6-3e1cb03b"
* Accept-Ranges: bytes
* Connection: close

Her kan det ses at headeren indeholder information omkring type af indhold der bliver sendt, længden og meget andet. Denne header anvendes af parterne i kommunikationen for at kunne behandle dataen korrekt.

Der er flere forskellige former for HTTP requests, typisk arbejder man med GET requests som henter information fra en server, eksempelvis når du indlæser en hjemmeside og POST requests som sender data til en server, eksempelvis når du logger ind på en side, så bliver der sendt en POST request med dine informationer. Der er dog også andre former for requests så som PUT og DELETE.

Under motorhjelmen anvender HTTP igen lag 4 protokollen TCP (Transfer Control Protocol) for at oprette forbindelsen og sende dataen med, HTTP handler altså mere om hvordan dataen bliver pakket så den er ”spiselig” af server/client end den handler om hvordan dataen bliver overført

Sikkerhed

**Problemstillinger:**

Vi har haft snakket om at vi gerne vil holde det hele på en database på trods af at vi benytter flere api’er, her har vi haft snakket om at vi skal passe på med hvor vores connection string ligger henne. Da vi sagtens kunne ligge den direkte ind i visual studio, men så vi lden også blive lagt op på github, hvilket ikke ville være så godt. Desuden har vi også haft snakket om at når vi skal lave login system så skal vores passwords hashes eventuelt med salt for at sikre passwordene endnu mere. Her vil vi nok benytte .NetCore som har noget login funktionalitet vi kan benytte.

## Virtualisering:

Man kunne måske teste projektet af i virtualisering, så man har et sikkert miljø at teste tingene af i. Desuden kunne man måske hoste hjemmesiden i en docker container på en ubuntu vm, på den måde vil man udnytte det som linux er lavet til, nemlig at være server. På den måde kunne man også teste lidt ting af uden det ville påvirke ens fysiske maskine. Man kunne også kører tests herpå.

# Docker

Vi havde snakket om brugen af docker til vores projekt og at det er relevant fordi vi kører en form for microservices. Derfor ville det være oplagt at bruge docker til at hoste hjemmesiden i en container og API’en i en anden container. Derved skal vi dog bruge multicontainer og som snakkede om i teknologi er det ikke noget man bare lige gør, da det opper kompleksiteten en god del.