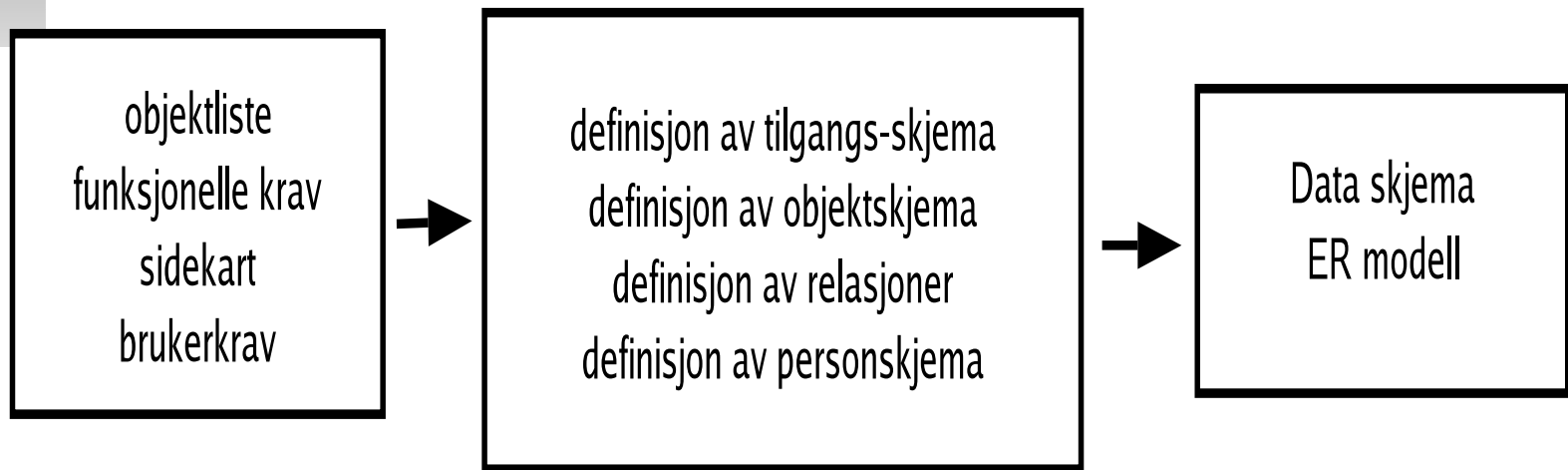


Lage ER modell av kravspesifikasjoner.



## 2 scenario:

- Ingen eksisterende database over hva applikasjonen skal inneholde. Datadesign'et utvikles samtidig med innholdsdatatabasen.
- Det eksisterer innholdsdatabase over hva applikasjonen skal inneholde. Datadesign'et må tilpasses helt/delvis.

# *tilpasning til eksisterende database*

---

Tilpasningsprosessen må ikke forkludre datadesignet.

- Dårlig strukturert innhold.
- Godt hypertekst design reflekterer godt datadesign.
- Dårlig struktur i innholdsdatabasen gir dårlig hypertekstdesign.

- **Hovedobjekter:**  
Hovedobjektene i kravspesifikasjonen. Et hovedobjekt kan representeres ved en eller flere entiteter. Ved flere entiteter får vi skjema av underobjekter. Utgjør hovedaktørene i ER-diagrammet. Utgjør ofte grensesnitt i web-applikasjon.
- **Relasjoner:**  
Utgjør semantiske relasjoner mellom hovedobjekter i dataordboken. Nyttig for design av linker.

- Tilgangsobjekter:  
Hjelpeobjekter som:
  - generaliserer, dvs.klassifiserer et hovedobjekt innen en mer generell kategori/klasse. F.eks, kan objekt 'Kunde' være en mer generell klassifiseringer for 'Privatkunde' og 'Bedriftskunde'.
  - spesialisierer, dvs.legger til entiteter til et hovedobjekt som dette objektet kan gjenkjennes ved.
  - grupperer flere hovedentiteter sammen for å fremheve fellestrekk (felles attributter).

- Brukerobjekter:
  - Plasserer brukere i systemet.
  - Objekter som inneholder bruker-relevante opplysninger.
  - Modellerer individuelt og/eller kollektivt.

Hvordan sette opp skjema av disse objektene?  
Proessen starter med enkelt skjema kun  
inneholdende hovedobjekter og så utvide etterhvert.

- Skjema av underobjekter.
- Skjema av relasjoner.
- Skjema av klassifiserings-objekter.
- Skjema av brukerobjekter.

# *Modellering av underentitet-skjema*

---

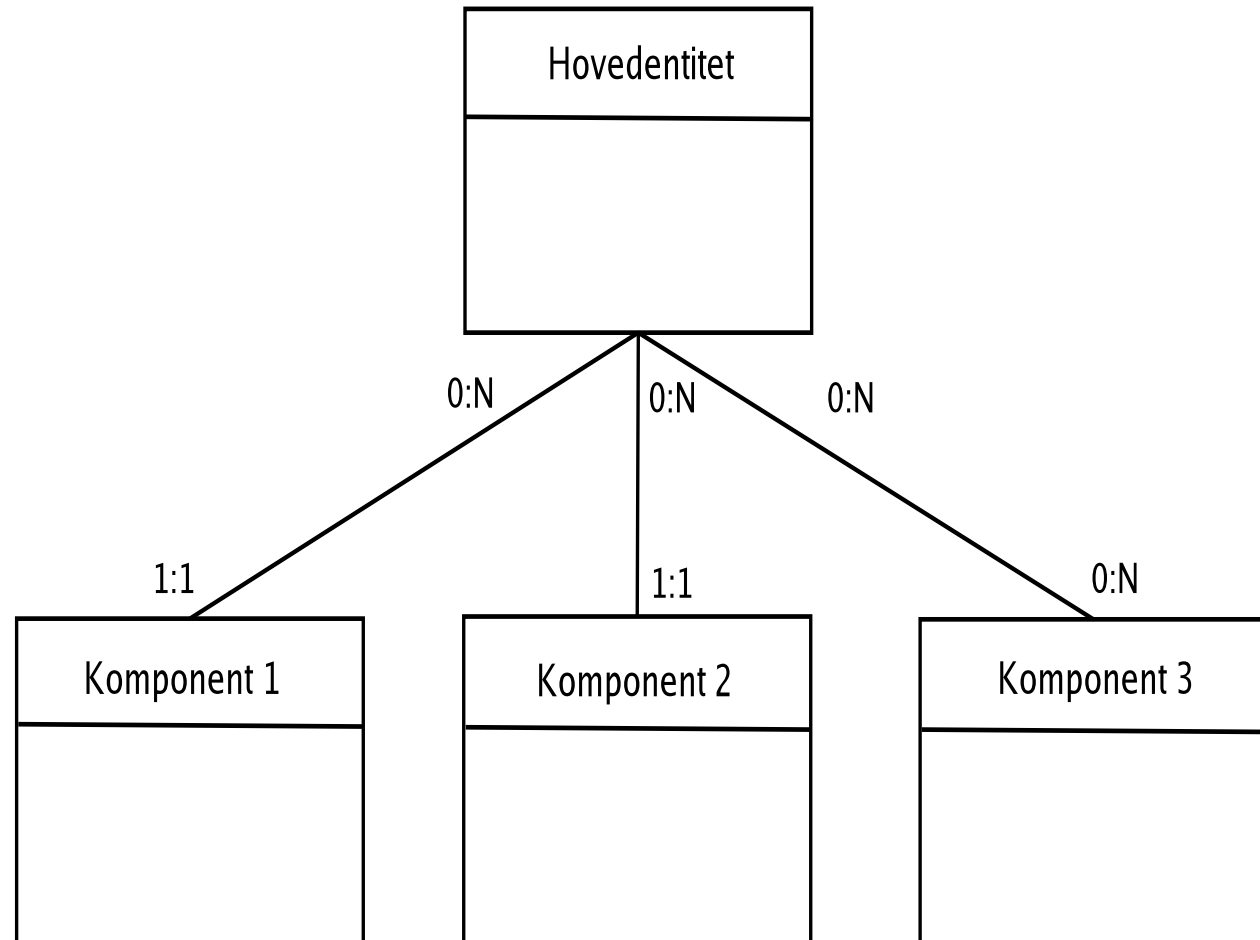
- Et hovedobjekt i kravspesifikasjonen tilsvarer en entitet(hovedentitet).
- Attributter er egenskaper ved entitetene.  
Signatur/ nøkler: En eller flere attributter som objektet gjenkjennes ved.
- Komplekse egenskaper kalles interne komponenter. Kobles til hovedobjektene via relasjonsentiteter.

## *2 typer interne komponenter*

---

- Svak entitet, kan ikke eksistere uavhengig av hovedobjektet det er tilknyttet og flere hovedobjekter kan ikke dele samme instans av den interne komponenten. Notasjon: (1:1).
- Kan eksistere uavhengig og være delt av flere hovedobjekt. Notasjon: (0:N).

# Underentitets-skjema



# *Design av relasjons-skjema*

---

- Relasjoner mellom hovedobjekter.
- I utg.punkt mange-til-mange.

# *Skjema av tilgangsentiteter*

---

Består av 2 typer entiteter:

- Kategorisende entiteter
- Spesialiserende entiteter.

## ***Kategoriserende entitet***

---

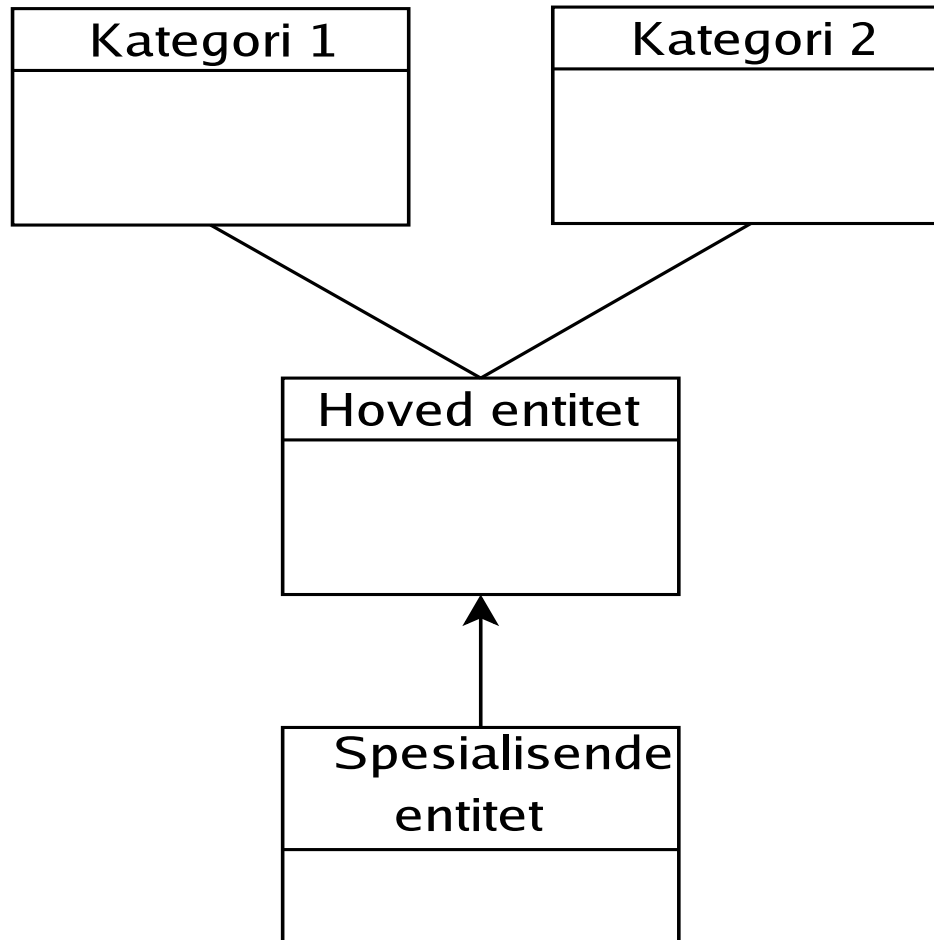
- Forbundet via relasjonsentitet til en hovedentitet.
- Har status som hovedentitet i seg selv.
- Hovedentitet kan tilhøre flere kategoriserende entiteter.
- Flere hovedentiteter kan tilhøre samme kategoriserende entitet.

## *Spesialiserende entitet*

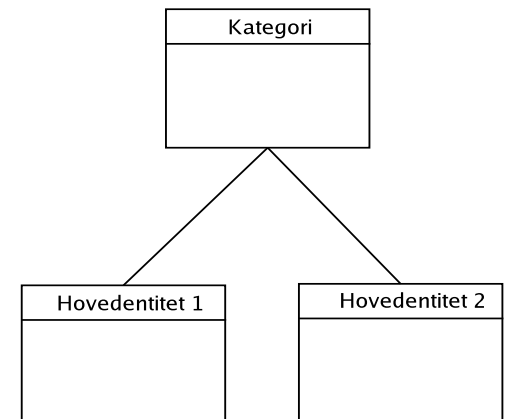
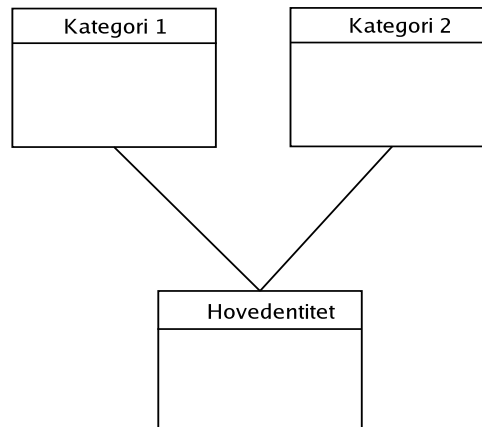
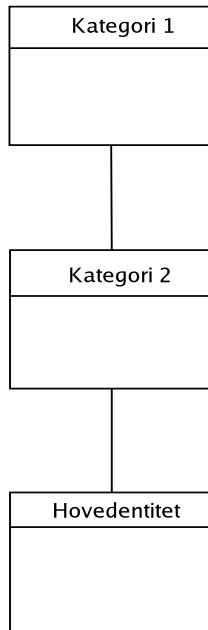
---

- Underentitet forbundet via en ISA relasjon til en hovedentitet.
- Instanser av disse har egenskaper som ikke finnes i det generelle tilfellet.
- Gruppere hovedentiteter med felles egenskaper.

# Kategori-skjema



# Kategoriseringstyper



# *Skjema av personentiteter*

---

- Brukerprofiler, informasjon om individuelle brukere. Personlige og/eller applikasjons-relevante.
- Brukergrupper, informasjon om grupper av brukere ut fra homogene krav.
- Personorienterte relasjoner som forbinder brukere med hovedentiteter angående f.eks, administrative rettigheter, tilgang på tjenester, osv...