**Mønstre for informasjonsflyt**

**Hva er dette?**

I dette notatet beskrives forskjellige mønstre for informasjonsflyt, samt hvilke krav mønstrene setter for rutiner for personell når de jobber i IT-systemer, krav til integrasjoner, og krav til operasjoner/logikk ved behandling av informasjon. Disse kravene utledes fra sammenstillingen av mønstrenes natur med målet om å ivareta kvaliteten på informasjon når den flyter mellom prosesser og mellom IT-systemer. Med kvalitet på informasjon menes det her at informasjon er komplett, konsistent, tilgjengelig i tide, innehar riktig integritet *(«jeg vil vite hvor blodprøven kom fra, når den er tatt, og hvem som la inn verdiene for å kunne stole på informasjonen»*), og er oppdatert.

Dokumentet omtaler ikke tekniske standarder for grensesnitt eller dataoverføring mellom IT-komponenter. Ordet «grensesnitt» er i dokumentet brukt løst i betydningen at noe kan avgi eller motta data, og er her ikke ment som en teknisk term.

**Formål**

Dokumentet skal brukes i forbindelse med å vurdere hva som skal til for å ivareta kvalitet på informasjon som må flyte mellom prosesser, og mellom IT-systemer. Det er svært viktig å forstå konsekvensene for kvalitet på informasjon i valg som tas for applikasjonsarkitektur, integrasjonsarkitektur og datalagring. Det å ha en felles forståelse for disse konsekvensene vil gjøre det enklere og raskere å videreutvikle rutiner og løsningsbeskrivelser for IT-integrasjon.

**Anbefaling for foretrukne mønstre**

Det anbefales at foretrukne mønstre for informasjonsflyt skal styres av brukerbehov. Hvilke konkrete mekanismer og teknologier som bør brukes for integrasjon og lagring bør styres av Helse Sør-Østs arkitekturprinsipper. Det er imidlertid viktig å se at et foretrukket mønster ut i fra et brukerperspektiv kan gi den konsekvensen at en rekke regelsett må utvikles, og dette er ikke nødvendigvis en enkel oppgave å lage eller gjennomføre i praksis.

Videre følger beskrivelse av mønstre og krav.

Innhold

[Mekanismer og standarder for overføring av informasjon – krav om å bevare mening og kompletthet i informasjon 2](#_Toc430687679)

[Introduksjon til dimensjonene 2](#_Toc430687680)

[Én brukerflate – én kilde til informasjon 3](#_Toc430687681)

[Én brukerflate til redigering – flere til lesing – én kilde 4](#_Toc430687682)

[Én brukerflate for redigering - én autoritativ kilde - flere kopier 5](#_Toc430687683)

[Flere brukerflater til editering og lesing – én kilde 7](#_Toc430687684)

[Én autoritativ kilde – flere kopier av informasjon – redigering i flere brukerflater av potensielt samme informasjon 9](#_Toc430687685)

[Definerte delmengder av informasjon oppdateres i flere brukerflater, og bare der de oppstår 12](#_Toc430687686)

# Mekanismer og standarder for overføring av informasjon – krav om å bevare mening og kompletthet i informasjon

Casene under beskriver kun til en viss grad hvilke tekniske mekanismer som skal brukes til flyt av data mellom systemer. Det er for noen av casene flere mulige løsninger, som sende eller kringkaste endringsmeldinger med mottakende systemer som abonnenter, eller enkeltoppslag med hent og lagre etc. I praksis ser en at flere mekanismer nå er i produksjon i parallell mellom kilder til og kopier av data. Det tas ikke her stilling til om dette er uheldig eller ikke. Dog, en kan spørre seg om det for eksempel alltid er hensiktsmessig å sende med pasientdemografi i logistikkmeldinger hvis tjenester for å oppdatere demografiinformasjon uansett må kalles når pasient blir hentet opp i fagsystem, siden demografiinformasjonen kan være utdatert siden siste logistikkmelding. Beskrivelsene under antyder ikke at det er noe krav om å oppdatere en viss type informasjon når en annen type informasjon oppdateres. Dette kan heller potensielt føre til usikkerheter i kvaliteten på informasjonen, da oppdateringsfrekvens og -tidspunkter kan variere mellom forskjellige typer informasjon.

Det som imidlertid er et krav, er at meningen til informasjonen må sikres harmonisert (semantisk interoperabilitet) uavhengig av hvilke tekniske mekanismer og standarder som brukes for å flytte data, enten det er manuelle eller automatiserte prosesser.

Semantisk interoperabilitet representerer ikke helheten tilstrekkelig. Semantisk interoperabilitet sikrer at det som skal være likt, er det. Imidlertid må det som skal være komplett også sikres komplett. Det finnes for eksempel nå logistikkoppdateringsmeldinger som sender andre delmengder av informasjon enn hent-tjenester. I tillegg kan filoverføringer fra eksterne systemer være med på å forkludre både semantikk og kompletthet. Det må derfor være et krav at nye løsninger, eller endringer i eksisterende må ivareta meningsinnhold og kompletthet i enda bedre grad enn for dagens løsninger. I tillegg anbefales det at det gjøres en vurdering av eksisterende portefølje, der en ser på om det bør gjennomføres en kvalifisering av informasjonsinnhold opp mot det som forventes av informasjon blant brukergrupper av IT-systemene der disse variasjonene av informasjon, grunnet forskjellige tekniske mekanismer for overføring av data, finnes.

# Introduksjon til dimensjonene

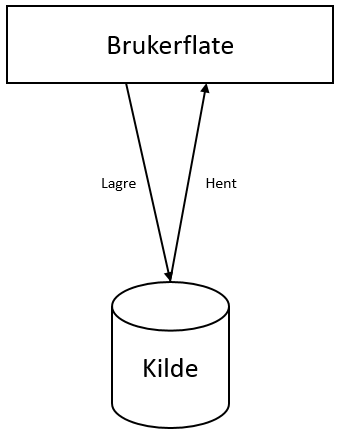
Dimensjonene som beskrives om informasjonsflyt er:

* Brukerflater til lesing og/eller redigering
* Kilder; der data lagres
* Kopier; en kilde som ikke er den autoritative

# Én brukerflate – én kilde til informasjon

**Kriterier:**

* Informasjon opprettes, endres og slettes via én brukerflate (typisk tilhørende et produkt eller spesialistsystem)
* Informasjonen kan kun ses i denne brukerflaten, og lagres ett sted.



Figur : Ingen integrasjon utover interne mekanismer

**Krav til rutiner:**

* Alle som skal opprette, endre, se eller slette informasjon, må gjøre det i dette produktets brukerflate

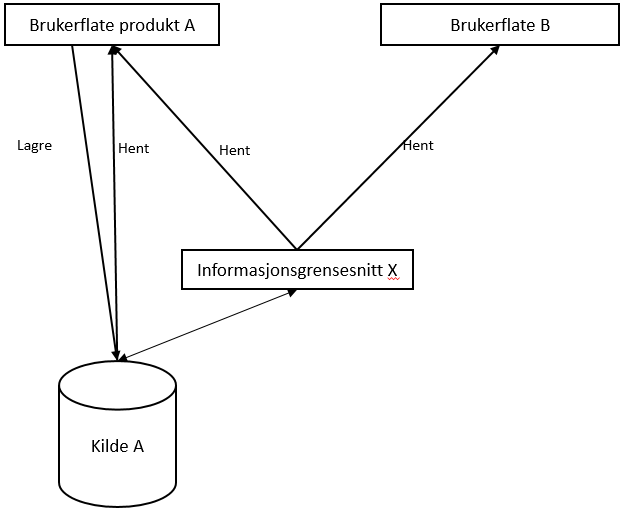
**Krav til operasjoner/logikk:**

* Det må finnes mekanismer som håndterer samtidighet. Med dette menes at det er en mulighet for at flere brukere vil prøve å editere den samme informasjonen samtidig. Dette må det da finnes regler for hvordan en skal løse, og disse reglene må implementeres i IT.

# Én brukerflate til redigering – flere til lesing – én kilde

**Kriterier:**

* Informasjon opprettes, endres eller slettes i ett produkts brukerflate
* Informasjonen er også tilgjengelig for lesing i andre produkters brukerflater
* Informasjon lagres kun ett sted



Figur : Én brukerflate for editering, flere til lesing, én kilde

**Krav til rutiner:**

* Alle som skal opprette, endre, eller slette informasjon, må gjøre det i produkt As brukerflate
* Alle som skal se informasjonen må gjøre det i ett av produktenes brukerflater der informasjonen er tilgjengelig

**Krav til integrasjoner:**

* Kilde A må tilby et lesegrensesnitt X for å kunne lage en leseintegrasjon mot brukerflate B
* Produkt B må ha et grensesnitt for å ta imot informasjon fra lesegrensesnitt X
* Brukerflate A kan bruke lesegrensesnitt X eller ha interne grensesnitt for å hente informasjon fra kilde A.
* Hvis det er behov for logikk mellom kilde og brukerflate for lesing, må enten logikken dupliseres eller så må brukerflate A hente informasjon fra lesegrensesnitt X, eller så må logikken som brukes internt blir prosessert samme sted mellom kilde A og informasjonsgrensesnittet som internt mellom kilde og brukerflate A.

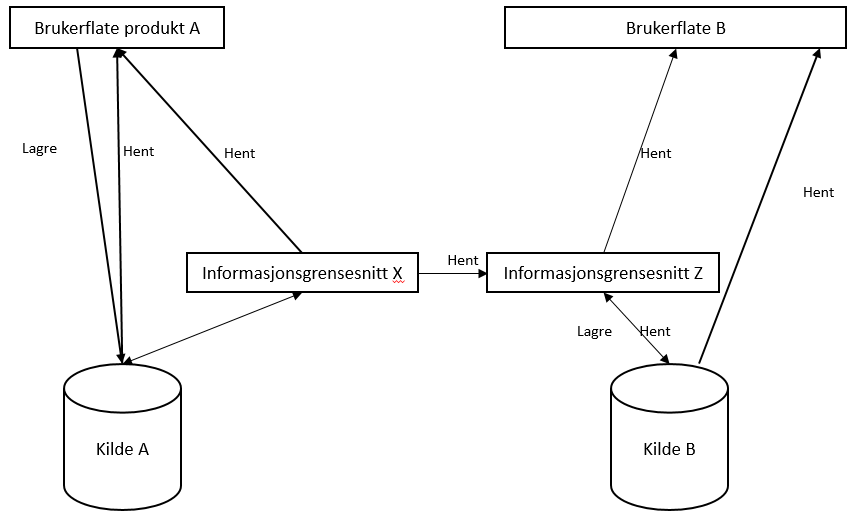
**Krav til rutiner og operasjoner/logikk ved mangler i integrasjoner:**

Ingen, da løsningen forutsetter integrasjonsmekanismer siden brukerflate B ikke har eget datalager

# Én brukerflate for redigering - én autoritativ kilde - flere kopier

**Kriterier:**

* Informasjonen opprettes, endres eller slettes via ett produkts brukerflate
* Informasjonen er også tilgjengelig for lesing i andre produkters brukerflater
* Informasjonen har en autoritativ kilde A
* Informasjonen lagres som kopier i database tilhørende brukerflate B



Figur : Autoritativ kilde, med kopier, redigering kun ett sted

**Krav til rutiner:**

* Alle som skal opprette, endre, eller slette informasjon, må gjøre det i produkt As brukerflate
* Alle som skal se informasjonen må gjøre det i ett av produktenes brukerflater der informasjonen er tilgjengelig

**Krav til operasjoner/logikk:**

* Gitt at integrasjoner som beskrevet under er på plass, så skal brukerflate B ikke kunne redigere data i kilde B som har kilde A som autoritativ kilde. Derfor må det finnes mekanismer som gjør at det er mulig å spore hvilke data som har hvilken autoritativ kilde i kilde B. På bakgrunn av dette må funksjonalitet i brukerflate B bygges deretter.
* Når brukere av brukerflate B vil se informasjon som har autoritativ kilde A, må det sikres at disse brukerne har tilgang til oppdatert informasjon når de ber om å lese informasjonen. Derfor må det sikres at den sist oppdaterte informasjonen hentes fra kilde A ved bruk av brukerflate B, eller sendes fra A til B når noe endres i A, og at brukerflate B derfor ikke får informasjon direkte fra kilde B uten at informasjonen er sikret oppdatert først.

**Krav til integrasjoner:**

Enten:

* Kilde A må tilby et hentegrensesnitt X for å kunne lage en henteintegrasjon mot brukerflate B og C databases integrasjonsgrensesnitt
* Produkt B database må ha et grensesnitt for å ta imot informasjon fra lesegrensesnitt X
* Produkt B må oppdateres fra Kilde A hver gang det kreves å lese informasjonen i brukerflate B
* Brukerflate A kan bruke lesegrensesnitt X eller ha interne grensesnitt for å hente informasjon fra kilde A. Hvis det er logikk i lesegrensesnitt X, må enten logikken dupliseres eller så må brukerflate A hente informasjon fra lesegrensesnitt X.

Eller

* A må tilby en abonnementsordning der oppdateringer på informasjonen fortløpende sendes ut ved endringer
* B må abonnere på denne endringsstrømmen, og sikre at endringer kommer fortløpende inn i kilde B
* Det må finnes en mekanisme som holder styr på rekkefølge på endringsmeldinger, som B eller integrasjonsplattformen må kunne håndtere, som å etterspørre mangler i datagrunnlag fra A.

**Krav til rutiner og operasjoner/logikk ved mangler i integrasjoner:**

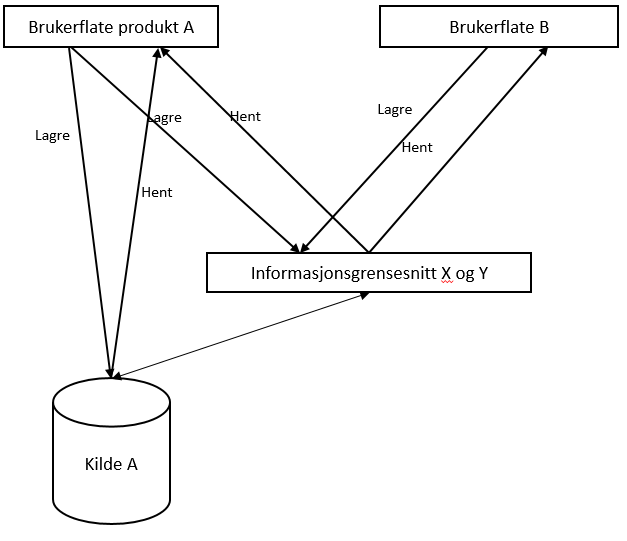
To siloer uten informasjonsflyt seg imellom. Kravene forutsetter at det er et faglig behov for å ha tilgang til informasjonen også i B, noe som er antatt behov i hele scenarioet for øvrig. Rutinene kompliseres betraktelig hvis integrasjoner ikke eksisterer, samt at annen logikk må eksistere i B.

* Rutiner:
  + Ved registrering i A, må det sikres at også det samme registreres i B, helst samtidig. Det må derfor eksistere rutiner som sikrer dobbelt(multippel)-registrering av informasjon. Det er potensielt forskjellige brukergrupper av A og B, og det må da sikres ivaretakelse av dobbelt (multippel) -registrering i rutinebeskrivelser på tvers av A og B brukergrupper, med klart plassert ansvar for hvem som gjør hva når, og at oppgaver blir gitt til riktig sted og tid, og at oppgavene blir ivaretatt i tide. I praksis er dette potensielt vanskelig å få gjennomført av god kvalitet. Det er i tillegg en risiko at kilde B i praksis oppstår som en autoritativ kilde for Bs brukere, og at dette forringer kvaliteten i informasjonen totalt sett.
* Logikk:
  + Det må være mulig å skrive inn informasjon som har oppstått i A i brukerflate B, noe som ikke skal gjøres ved tilstrekkelige integrasjonsmekanismer
  + Det er spesielt viktig å få harmonisert semantikken mellom A og B, da dette ikke har blitt sikret gjennom utvikling av integrasjoner. Ved utvikling av integrasjoner er semantikkharmonisering et krav.

# Flere brukerflater til editering og lesing – én kilde

**Kriterier:**

* Informasjon kan opprettes, endres eller slettes via flere produkters brukerflater
* Informasjon er tilgjengelig for lesing i flere brukerflater
* Informasjon lagres ett sted



Figur : Flere brukerflater til redigering og lesing - én kilde

**Krav til rutiner:**

* Alle som skal se, opprette, endre, eller slette informasjon, kan gjøre det i produkt A og Bs brukerflater

**Krav til operasjoner/logikk:**

* Det må finnes mekanismer som håndterer samtidighet. Med dette menes at det er en mulighet for at brukere av forskjellige brukerflater vil prøve å editere den samme informasjonen samtidig. Dette må det da finnes regler for hvordan en skal løse, og disse reglene må implementeres i IT.

**Krav til integrasjoner:**

* Kilde A må tilby et lesegrensesnitt X for å kunne lage en leseintegrasjon mot brukerflate B
* Kilde A må tilby et lagregrensesnitt Y for å kunne lage en lagreintegrasjon mot brukerflate B
* Produkt B må ha et grensesnitt for å ta imot informasjon fra lesegrensesnitt X
* Produkt B må ha et grensesnitt for å avgi informasjon til lagregrensesnitt Y
* Brukerflate A kan bruke lesegrensesnitt X og lagregrensesnitt Y eller ha interne grensesnitt for å hente og avgi informasjon fra og til kilde A. Hvis det er logikk i grensesnitt X og Y, må enten logikken dupliseres eller så må brukerflate A hente informasjon fra grensesnitt X og Y.

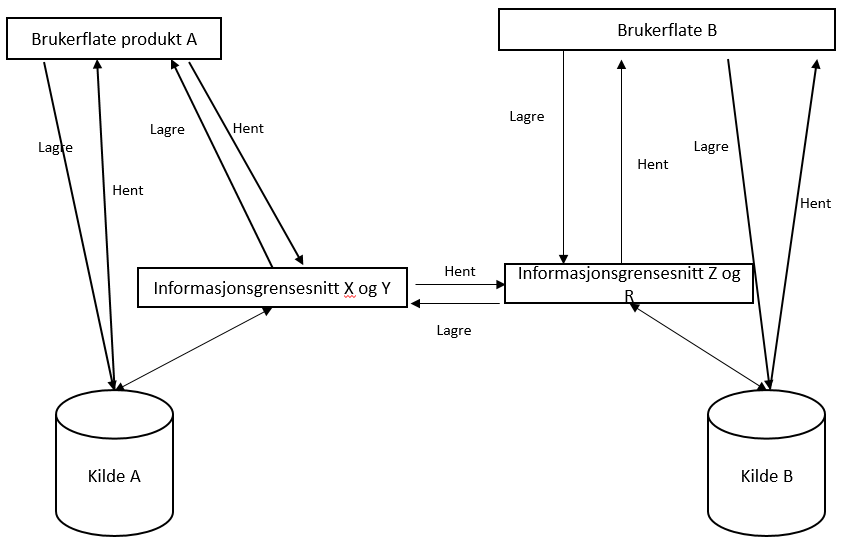
**Krav til rutiner og operasjoner/logikk ved mangler i integrasjoner:**

Ingen, da løsningen forutsetter integrasjonsmekanismer siden brukerflate B ikke har eget datalager

# Én autoritativ kilde – flere kopier av informasjon – redigering i flere brukerflater av potensielt samme informasjon

**Kriterier:**

* Informasjon kan leses, opprettes, endres eller slettes via flere produkters brukerflater
* Kilde A er autoritativ kilde
  + Potensielt kan en her tenke at både A og B er likeverdige kilder, men casen beskriver en besluttet autoritativ kilde. Dette, fordi et potensielt system C også vil kunne ha behov for, for eksempel, å lese informasjonen, og det blir da et krav om at C skal bruke A. Hvilke konkrete IT-løsninger som skal representere A og B må tas ut i fra et strategisk videreutviklingsperspektiv og arkitekturprinsipper.
* Informasjon lagres som kopi i kilde B tilhørende brukerflate B
* Et scenario her er også at delmengder av informasjon oppstår i B, men at det er lov til å endre disse i A



Figur : Én autoritativ kilde, flere kopier og editering i flere brukerflater

**Krav til rutiner:**

* Alle som skal se, opprette, endre, eller slette informasjon, kan gjøre det i produkt A og Bs brukerflater

**Krav til operasjoner/logikk:**

* Det må finnes mekanismer som håndterer samtidighet på tvers av kilder og produkter. Med dette menes at det er en mulighet for at brukere av forskjellige brukerflater vil prøve å editere den samme informasjonen samtidig. Dette må det da finnes regler for hvordan en skal løse, og disse reglene må implementeres i IT.
* Det må finnes mekanismer og regler for konsistens. Potensielt vil forskjellige brukere ville avgi forskjellige verdier som har et forretningsmessig krav om å være like for å sikre konsistens. Dette må løses som implementerte regler i IT, eller at det tas case til case. Hvis det ønskes at dette løses ved implementering av regler i IT, vil dette regelverket antakelig være vanskelig å lage i mange tilfeller. Hvis det siste er velges, må det lages rutiner som sikrer at potensielle konsistenskonflikter løses, og at det finnes tydelig plassert ansvar for hvem som skal sørge for konsistenssikring.
* Når brukere av brukerflate B vil se informasjon som har autoritativ kilde A, må det sikres at disse brukerne har tilgang til oppdatert informasjon når de ber om å lese informasjonen. Derfor må det sikres at den sist oppdaterte informasjonen hentes fra kilde A ved bruk av brukerflate B, eller sendes fra A til B når noe endres i A, og at brukerflate B derfor ikke får informasjon direkte fra kilde B uten at oppdatering er gjennomført først.
* Når brukere av brukerflate B gjør endringer på informasjon, må dette tilflyte kilde A umiddelbart for å sikre konsistens og kompletthet i autoritativ kilde

**Krav til integrasjoner:**

* Kilde A må tilby et lesegrensesnitt X for å kunne lage en leseintegrasjon mot brukerflate B
* Kilde A må tilby et lagregrensesnitt Y for å kunne lage en lagreintegrasjon mot brukerflate B
* Produkt B må ha et grensesnitt for å ta imot informasjon fra lesegrensesnitt X
* Produkt B må ha et grensesnitt for å avgi informasjon til lagregrensesnitt Y, og må pushe endringer dit når disse oppstår i brukerflate B
* Brukerflate A kan bruke lesegrensesnitt X og lagregrensesnitt Y eller ha interne grensesnitt for å hente og avgi informasjon fra og til kilde A. Hvis det er logikk i grensesnitt X og Y, må enten logikken dupliseres i lagre og hent til kilde A, eller så må brukerflate A hente informasjon fra grensesnitt X og Y.
* Brukerflate B kan bruke lesegrensesnitt Z og lagregrensesnitt R eller ha interne grensesnitt for å hente og avgi informasjon fra og til kilde B. Hvis det er logikk i grensesnitt X og Y må denne logikken dupliseres i grensesnitt Z og R, eller mellom henting og lagring til kilde B.

**Krav til rutiner og operasjoner/logikk ved mangler i integrasjoner:**

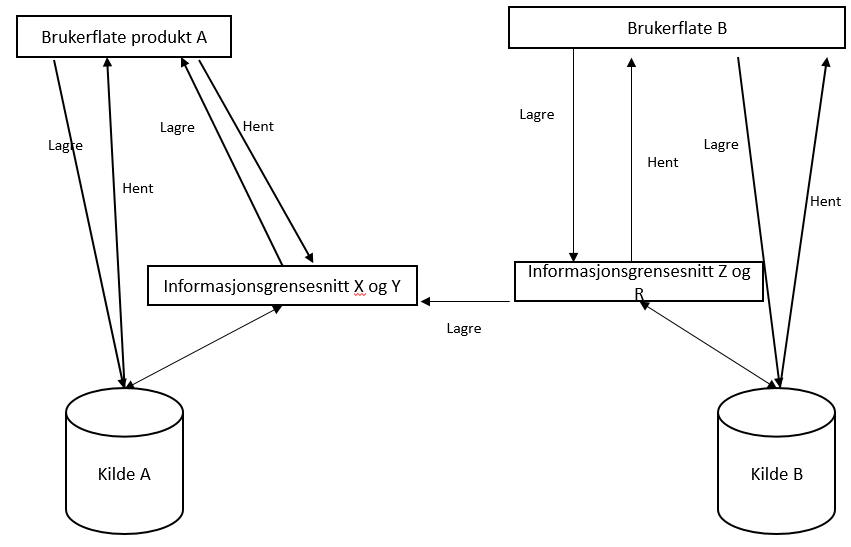
To siloer uten informasjonsflyt seg imellom. Kravene forutsetter at det er et faglig behov for å ha tilgang til, samt mulighet for editering av, informasjonen i B og vice versa, noe som er antatt behov i hele scenarioet for øvrig. Rutinene kompliseres betraktelig hvis integrasjoner ikke eksisterer, samt at annen logikk må eksistere i A og B.

* Rutiner:
  + Ved registrering i A, må det sikres at også det samme registreres i B, helst samtidig. Det må derfor eksistere rutiner som sikrer dobbelt(multippel)-registrering av informasjon. Det er potensielt forskjellige brukergrupper av A og B, og det må da sikres ivaretakelse av dobbelt (multippel) -registrering i rutinebeskrivelser på tvers av A og B brukergrupper, med klart plassert ansvar for hvem som gjør hva når, og at oppgaver blir gitt til riktig sted og tid, og at oppgavene blir ivaretatt i tide.
  + Ved registrering i B, må det sikres at også det samme registreres i A, helst samtidig. Det må derfor eksistere rutiner som sikrer dobbelt(multippel)-registrering av informasjon. Det er potensielt forskjellige brukergrupper av A og B, og det må da sikres ivaretakelse av dobbelt (multippel) -registrering i rutinebeskrivelser på tvers av A og B brukergrupper, med klart plassert ansvar for hvem som gjør hva når, og at oppgaver blir gitt til riktig sted og tid, og at oppgavene blir ivaretatt i tide.
  + Det må være manuelle rutiner for sjekk av konsistens i henhold til regelverk på tvers av A og B.
  + I praksis er dette potensielt vanskelig å få gjennomført av god kvalitet. Det er i tillegg en risiko at kilde B i praksis oppstår som en autoritativ kilde for Bs brukere, og at dette forringer kvaliteten i informasjonen totalt sett.
* Logikk:
  + Det er spesielt viktig å få harmonisert semantikken mellom A og B, da dette ikke har blitt sikret gjennom utvikling av integrasjoner. Ved utvikling av integrasjoner er semantikkharmonisering et krav.

# Definerte delmengder av informasjon oppdateres i flere brukerflater, og bare der de oppstår

**Kriterier:**

* Det er ikke overlapp mellom delmengder av informasjon som registreres i forskjellige brukerflater
* Distinkte delmengder av informasjonen kan opprettes, endres eller slettes i flere produkters brukerflater
* Det er behov for og finnes en komplett autoritativ kilde A
* Brukerflate A kan lese det komplette informasjonssettet
* Informasjon fra lokale systemer er tilgjengelig for gjenbruk fra kilde A og kan kun editeres i lokale systemer



Figur : Definerte delmengder oppdateres i flere brukerflater.

**Krav til rutiner:**

* Alle som skal se, opprette, endre, eller slette informasjon, kan gjøre det i A og Bs brukerflater kun for egen informasjonsmengde
* Hvis brukere av brukerflate A oppdager feil i informasjon som har oppstått i brukerflate B, må det sikres at verdiene blir endret fra brukerflate B. Det må sikres tydelige rutinebeskrivelser for hvordan dette skal gjennomføres, og at ansvar for registrering av informasjon på tvers av IT-systemgrenser er tydelig plassert og harmonisert for å sikre at oppgaven blir gjennomført.

**Krav til operasjoner/logikk**

* Brukerflate A skal ikke kunne editere data i kilde A som har oppstått i brukerflate B. Derfor må det finnes mekanismer som gjør at det er mulig å spore hvilke data som har hvilken kilde i kilde A. På bakgrunn av dette må funksjonalitet i brukerflate A bygges deretter.
* Det må sikres at kilde A fortløpende mottar endringer i informasjon fra andre kilder. Det vil si at endringer som oppstår på informasjonen via brukerfalte B, må tilflyte kilde A samtidig som det lagres i kilde B

**Krav til integrasjoner:**

* Kilde A må tilby et lesegrensesnitt X for å kunne lage en leseintegrasjon mot brukerflate B
* Kilde A må tilby et lagregrensesnitt Y for å kunne lage en lagreintegrasjon mot brukerflate B
* Produkt B må ha et grensesnitt for å ta imot informasjon fra lesegrensesnitt X
* Produkt B må ha et grensesnitt for å avgi informasjon til lagregrensesnitt Y, og må pushe endringer dit når disse oppstår i brukerflate B
* Brukerflate A kan bruke lesegrensesnitt X og lagregrensesnitt Y eller ha interne grensesnitt for å hente og avgi informasjon fra og til kilde A. Hvis det er logikk i grensesnitt X og Y, må enten logikken dupliseres i lagre og hent til kilde A, eller så må brukerflate A hente informasjon fra grensesnitt X og Y.
* Brukerflate B kan bruke lesegrensesnitt Z og lagregrensesnitt R eller ha interne grensesnitt for å hente og avgi informasjon fra og til kilde B. Hvis det er logikk i grensesnitt X og Y må denne logikken dupliseres i grensesnitt Z og R, eller mellom henting og lagring til kilde B.

**Krav til rutiner og operasjoner/logikk ved mangler i integrasjoner:**

To siloer uten informasjonsflyt seg imellom. Kravene forutsetter at det er et faglig behov for å ha tilgang til informasjonen også i A, noe som er antatt behov i hele scenarioet for øvrig. Rutinene kompliseres betraktelig hvis integrasjoner ikke eksisterer, samt at annen logikk må eksistere i B.

* Rutiner:
  + Ved registrering i B, må det sikres at også det samme registreres i A, helst samtidig. Det må derfor eksistere rutiner som sikrer dobbelt(multippel)-registrering av informasjon. Det er potensielt forskjellige brukergrupper av A og B, og det må da sikres ivaretakelse av dobbelt (multippel) -registrering i rutinebeskrivelser på tvers av A og B brukergrupper, med klart plassert ansvar for hvem som gjør hva når, og at oppgaver blir gitt til riktig sted og tid, og at oppgavene blir ivaretatt i tide. I praksis er dette potensielt vanskelig å få gjennomført av god kvalitet.
  + Det som skal oppstå i B, skal ikke oppstå eller endres i A uten at det er oppstått eller endret i B, selv om det teknisk mulig i A.
* Logikk:
  + Det må være mulig å skrive inn informasjon som har oppstått i B i brukerflate A, noe som ikke skal gjøres ved tilstrekkelige integrasjonsmekanismer
  + Det er spesielt viktig å få harmonisert semantikken mellom A og B, da dette ikke har blitt sikret gjennom utvikling av integrasjoner. Ved utvikling av integrasjoner er semantikkharmonisering et krav.

+ flere mønstre