

## Delrapport fra Elviraprojektet Nettbasert pasientinformasjonssystem

# Arkitektur og visualisering

Av:

Johan Gustav Bellika<sup>1</sup>, Gunnar Hartvigsen<sup>1 3</sup>, Leif Erik Loftesnes<sup>2</sup>,  
Thomas Strandenæs<sup>1</sup>

<sup>1</sup> Nye tjenester for helsenettet – Nasjonalt senter for Telemedisin,

<sup>2</sup> Framtidslaboratoriet - Nasjonalt senter for Telemedisin,

<sup>3</sup> Institutt for Informatikk, Mat.Nat. Fakultetet – Universitetet i Tromsø

Dato: 02.05.2001

<b>SAMMENDRAG .....</b>	<b>1</b>
<b>1 INTRODUKSJON.....</b>	<b>3</b>
<b>2 KJENTE SYSTEMER OG PROSJEKTER .....</b>	<b>4</b>
2.1 BOJ PROSJEKTET .....	4
2.2 BAPTIST HEALTH SYSTEM OF SOUTH FLORIDA .....	5
2.3 CAREWEB .....	6
2.4 HYGEIANET .....	7
<b>3 VISUALISERING.....</b>	<b>8</b>
3.1 INNLEDNING .....	8
3.2 LIVSLØPSFREMSTILLING .....	9
3.3 VIDERE ARBEID .....	10
3.3.1 Forslag til design av undersøkelse.....	10
3.3.2 Case.....	11
3.3.3 Forsøkspersoner.....	11
3.3.4 Spørreskjema.....	12
<b>4 ALTERNATIVE ARKITEKTURLØSNINGER .....</b>	<b>12</b>
4.1 FELLESKOMPONENTER OG TJENESTER FOR ALLE ALTERNATIVER .....	12
4.1.1 Loggtjeneste .....	12
4.1.2 Felles datamodell.....	13
4.1.3 Pasientinformasjonsindeks.....	13
<b>5 ELVIRA SOM DATAVAREHUS .....</b>	<b>14</b>
5.1.1 Dataarkitektur .....	15
5.1.2 Infrastruktur .....	15
5.1.3 Elvira som datavarehus.....	16
5.2 MULTIDATABASE-TILNÆRMING .....	16
5.2.1 Problemstillinger.....	17
5.3 MOBIL AGENT LØSNING (MELDINGSBASERT).....	17
5.3.1 Grunnkonsepter for mobile agenter .....	17
5.3.2 Innsamling og visualisering av pasientinformasjon.....	20
5.3.3 Problemstillinger.....	22
<b>6 DISKUSJON.....</b>	<b>23</b>
<b>7 KONKLUSJONER OG ANBEFALINGER.....</b>	<b>25</b>
<b>8 REFERANSER.....</b>	<b>27</b>

# Sammendrag

## Problembeskrivelse:

Visjonen i Elviraprojektet er at helsepersonell som er autorisert for innsyn skal ha tilgang til all informasjon som er relevant og nødvendig for problemstillingen de står ovenfor i møtet med pasienter, uavhengig av tid og sted og hvor pasientinformasjonen er fysisk lagret. Formålet med slik tilgang er bedret medisinsk behandling, omsorg og oppfølging av pasienter. Problemstillingen for denne rapporten er hvor pasientinformasjon skal lagres og hvordan informasjonen skal gjøres tilgjengelig for helsearbeiderne. Denne delrapporten omhandler alternative systemarkitekturer som muliggjør distribusjon av pasientinformasjon og hvordan visualisering av pasientinformasjon kan påvirke nytten av tilgang.

## Løsningsstrategier:

På overordnet plan finnes to hovedalternativer for å gjøre informasjon fra en mengde informasjonssystemer tilgjengelig for brukeren. Hovedalternativ 1 er å gjøre en integrasjon mellom systemene slik at f.eks alle pasientjournalssystemer er i stand til å aksessere, forstå og visualisere informasjon fra andre journalssystemer, røntgen, lab og andre systemer som inneholder pasientinformasjon. Dette alternativet har vi valgt å se bort fra på grunn av arbeidsomfanget forbundet med en slik realisering. Hovedalternativ 2 er å lage et felles visualiseringssystem for alle systemer som inneholder pasientinformasjon. Dette systemet kan integreres i dagens journalssystemer slik at det for brukeren fremstår som om all informasjon gjøres tilgjengelig via et system – journalssystemet.

Mange land og prosjektet jobber mot visjonen om å gjøre pasientinformasjon tilgjengelig uavhengig av hvor denne er lagret. Noen er kommet ganske langt og har allerede realisert løsninger ala det vi ser for oss i Elvira prosjektet. Av disse kan nevnes BoJ ved Dandryd sykehus, som startet opp i 1998, CareWeb prosjektet fra Boston, USA og HYGEIAnet på Kreta i Hellas. Alle disse har implementert systemer som delvis realiserer Elvira prosjektets visjon.

Vi har vurdert tre alternative systemarkitekturer for å avdekke problemstillingene en må fokusere på i et eventuelt oppfølgings-/realiseringsprosjekt. Ingen av disse alternativene er vurdert i en slik dybde at vi vil komme med en anbefaling om hvilken løsning en bør basere en reell løsning på.

Et hovedvalg for systemarkitektur er sentralisert kontra desentralisert lagring av pasientinformasjon. Slik vi ser det representerer en sentralisert løsning, der alle helseinstitusjoner sender data til en felles "datasentral", det beste alternativet med hensyn til sikkerhet, sårbarhet og tilgjengelighet. Felles lagring av pasientinformasjon på tvers av institusjonsgrenser er imidlertid problematisk juridisk. Et potensielt problem ved en sentralisert løsning er at et slikt system nødvendigvis vil fungere som et mellomledd mellom produsent og konsument av informasjon. Pasienter kan av denne grunn være restriktiv i forhold til at informasjon deles mellom helseinstitusjoner og redusere omfanget av informasjon som kan gjøres tilgjengelig for eksterne institusjoner. De desentraliserte alternativene har i denne sammenheng et større potensial fordi informasjon kun vil være tilgjengelig for produsenten og de helsearbeiderne pasienten gir innsynsrett til. De desentraliserte løsningene er problematisk med hensyn til å ivareta tilstrekkelig grad av sikkerhet (i små institusjoner), tilgjengelighet (driftsavbrudd, kabelbrudd etc.), og tilgang til systemressurser for å betjene eksterne institusjoners informasjonsbehov.

Slik vi vurderer det vil en desentralisert løsning være den beste med hensyn til mulig funksjonalitetsnivå. En mangler imidlertid kunnskap om hvordan en kan oppnå tilfredstillende grad av sikkerhet og tilgjengelighet ved slike alternativer.

En hypotese om enkel tilgang til informasjon er at når mengden på tilgjengelig informasjon øker, øker handlingsrommet for mottakeren av informasjonen og dermed tidsforbruket. Denne hypotesen tilsier at en ved å øke graden av enkelt tilgjengelig pasientinformasjon, øker tidsforbruket til å sette seg inn i pasientens sykehistorie. Nyttien av enkel tilgang til pasientinformasjon kan derfor i stor grad være avhengig av hvilken informasjon som er tilgjengelig og hvor enkelt det er for brukeren å vurdere hva som er relevant for problemstillingen. En bør gjennomføre et forskningsarbeid for å finne ut hvilken funksjonalitet i og utforming av brukergrensesnitt som tilfredstiller helsearbeidernes behov for informasjon og samtidig virker effektiviserende.

Anbefalinger/retninger for det videre arbeidet:

Alle alternativer til realisering av visjonen til Elviraprojektet må ivareta autentisering og autorisasjon av helsearbeidere som skal ha tilgang til pasientinformasjon. Disse funksjonene bør (må) være felles for alle pasientinformasjonssystemer i Norge. Selv om en velger å ikke realisere Elviraprojektets visjon, bør en etablere fellesløsninger for disse funksjonene på grunn av mobiliteten til helsearbeiderne. En lege kan ha mange forskjellige roller, i ulike institusjoner, i løpet av kort tid. En bør derfor starte prosjekter som realiserer fellesløsninger for infrastruktur, rutiner og systemer som understøtter autentisering og autorisering av tilgang til pasientinformasjon.

En bør, som nevnt ovenfor, starte forskningsprosjekt på hvordan et visualiseringssystem for pasientinformasjon best kan understøtte helsearbeidernes informasjonsbehov og samtidig ivareta effektivitet.

Distribuerte systemarkitekturer for tilgang til pasientinformasjon vil kunne gi det høyeste funksjonalitet og informasjonsnivå. En bør inngangsette prosjekt for å avdekke om og hvordan en kan oppnå tilfredsstillende grad av sikkerhet og tilgjengelighet til informasjon ved slike systemarkitekturer. Vi anbefaler derfor at en jobber videre med en risikoanalyse for ett eller flere desentraliserte systemarkitekturer.

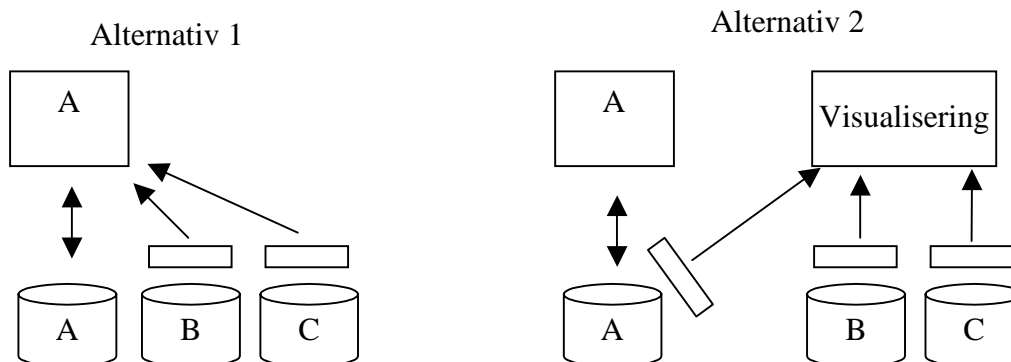
# 1 Introduksjon

Visjonen i Elviraprojektet er at helsepersonell som er autorisert for innsyn skal ha tilgang til all informasjon som er relevant og nødvendig for problemstillingen de står ovenfor i møtet med pasienter, uavhengig av tid og sted og hvor pasientinformasjonen er fysisk lagret. Formålet med slik tilgang er bedret medisinsk behandling, omsorg og oppfølging av pasienter.

Utviklingen av datasystemer som inneholder pasientinformasjon har vært eksplosiv og omfatter i dag hundrevis av datasystemer på legekontor, sykehus og andre helseinstitusjoner. Hovedregelen i dag er likevel at informasjonen utveksles manuelt, på papir og ved hjelp av postverket eller via faks. Utviklingen av et nasjonalt helsenett gir imidlertid muligheter for realisering av visjonen ovenfor.

Denne delrapporten fokuserer på noen systemarkitekturer som kan realisere visjonen om autorisert tilgang uavhengig av tid og sted og hvor pasientinformasjonen er fysisk lagret. Rapporten omhandler også kort problematikken rundt visualisering av pasientinformasjon og hvilken betydning dette har for helsearbeiderens evne til å raskt å finne informasjonen som er relevant for behandling, omsorg og oppfølging av pasienten.

På overordnet plan eksisterer det to hovedalternativer til å gjøre informasjonen fra alle disse informasjonssystemene tilgjengelig.



**Figur 1 Hovedalternativer for arkitektur og visualisering**

Alternativ 1 i figuren ovenfor viser hvordan informasjonssystem A (firkant) inneholder funksjonalitet for å lagre og visualisere informasjon fra sitt datalager. Informasjonssystem A kan også hente informasjon fra informasjonssystem B og C sine datalager gjennom integrasjonsprogramvare. Alternativ 2 kan visualisere informasjon fra system A, B og C gjennom integrasjonsprogramvare. Visualiseringskomponenten kan integreres i brukergrensesnittet til informasjonssystem A slik at det for brukeren fremstår som ett system. Alternativ 1 er med andre ord å gjøre all informasjon tilgjengelig via institusjonenes journalsystem. Dette innebærer at alle systemer som produserer pasientinformasjon må kunne levere informasjon til dette systemet. Journalsystemet må i dette tilfelle kunne visualisere denne informasjonen via journalsystemets brukergrensesnitt. Vi har valgt å ikke vurdere alternativ 1 nærmere i denne delrapporten fordi omfanget av integrasjonsarbeid som må utføres for å realisere dette alternativet, er enormt. På den annen side kan et slikt alternativ være mer fremtidsrettet enn alternativ 2. Det andre alternativet til realisering av visjonen er å lage et visualiseringssystem der all pasientinformasjon kan gjøres tilgjengelig for autorisert

personell. Dette alternativet synes mer realistisk fordi alle leverandørsystemene vil forholde seg til ett felles visualiseringssystem kontra mange. Denne delrapporten fokuserer kun på det siste alternativet.

Det finnes en mengde alternative systemarkitekturer som kan realisere visjonen i Elvira-prosjektet. Målet med denne delrapporten er å avdekke problemstillinger som en må arbeide videre med dersom en velger å realisere visjonen. Delrapporten vil se på tre alternative systemarkitekturer som kan realisere autorisert tilgang til pasientinformasjon. Vi vil også beskrive systemkomponenter som vil være nødvendige uavhengig av hvilken arkitekturløsning en velger, og derfor kan sees på som kandidater for oppfølgingsprosjekt til Elviraprojektet. De tre arkitekturalternativene er 1) sentralisert lagring av pasientinformasjon, 2) distribuert multibase tilnærming og 3) asynkron innsamling av pasientinformasjon ved hjelp av mobile agenter. Disse alternativene vil vi evaluere i forhold til skalerbarhet, ytelse, sikkerhet, teknologisk modenhet, sårbarhet, kostnader og funksjonalitet.

Oppsummert er konklusjonene i denne rapporten at et nettbasert system for tilgang til pasientinformasjon er teknologisk mulig å realisere ved hjelp av mange systemarkitekturer. To hovedalternativer er sentraliserte kontra desentraliserte systemarkitekturer. Problemet med en distribuert løsning er økt kompleksitet og derved økt sårbarhet og sikkerhetsproblemer. En sentralisert løsning vil være langt mer sikker, men vil i følge delrapporten om juridiskeproblestillinger ikke være ideell. Vår anbefaling til det videre arbeid er derfor at en jobber videre med å avdekke hvordan en kan oppnå tilstrekkelig grad av sikkerhet til å realisere en distribuert systemarkitektur for nettbasert tilgang til pasientinformasjon.

Visualisering av pasientinformasjon er et annet område en bør jobbe videre med. Nyten og bruken av et slikt system vil i stor grad avhenge av dette. Uansett valg av systemarkitektur bør en starte arbeidet med å realisere felles komponenter for autentisering, autorisasjon og logging av innsyn i pasientinformasjon.

## **2 Kjente systemer og prosjekter**

Mange land og prosjekter jobber mot visjonen om å gjøre pasientinformasjon tilgjengelig uavhengig av hvor denne er lagret. Noen er kommet ganske langt og har allerede realisert løsninger tilsvarende det vi ser for oss i Elvira prosjektet.

### **2.1 BoJ prosjektet**

BoJ er et akronym for Bild och Journal. Prosjektet har sitt utspring i innføringen av elektronisk journal ved Danderyds Sjukhus (DS) i Stockholm, Sverige i 1998 [1]. Innføringen av elektronisk journal ved DS hadde som mål å minske den manuelle håndteringen av journalinformasjon og øke tilgjengeligheten av denne. Formålet med innføringen var å øke sikkerheten og kvaliteten i arbeidet. Etter innføringen gjensto likevel følgende problemer:

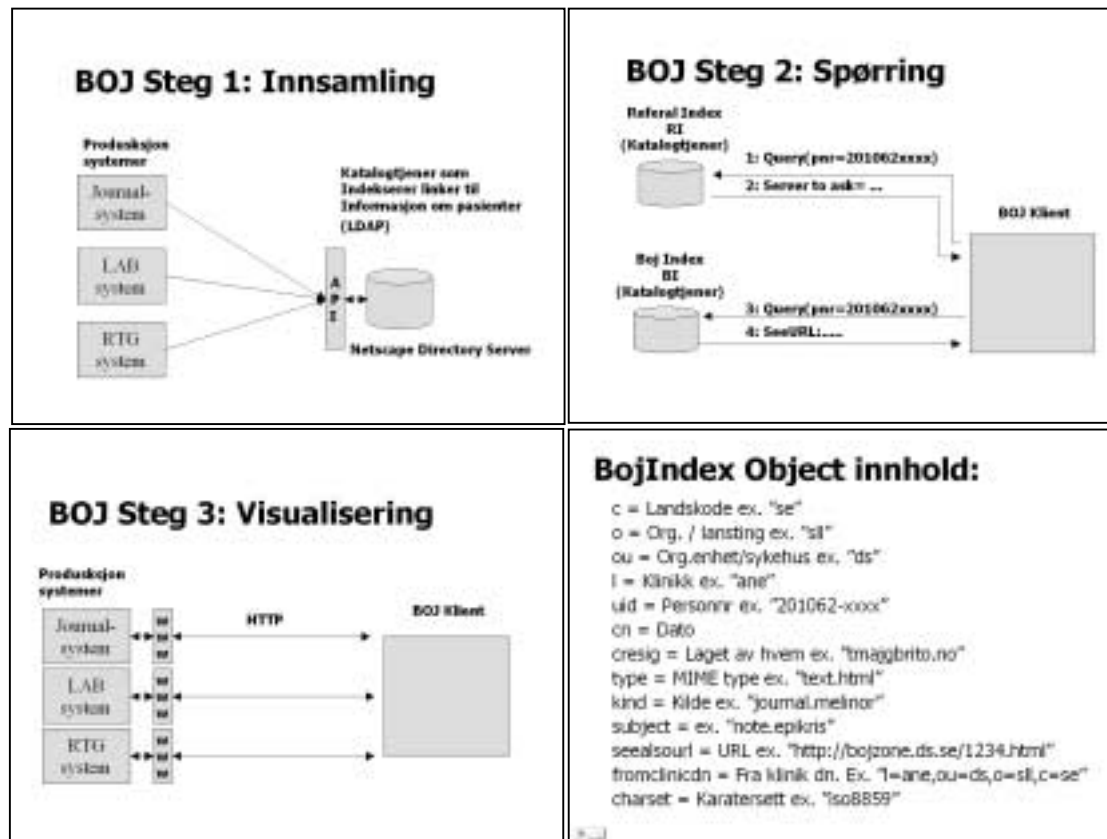
1. Visse typer data som laboratorieresultat, røntgenbilder, samt andre grafer og diagram kunne ikke presenteres i journalen. Dette minsket påtagelig nytten av en elektronisk journal.
2. Juridiske begrensinger i forhold til utveksling av informasjon mellom institusjoner førte til lokale datalager.
3. På tross av at flere sykehus benyttet elektroniske pasientjournaler kunne disse ikke nås fra DS.

BoJ prosjektet har gjennomgått 3 faser hvor den siste omfattet utveksling av pasientinformasjon over nettet i form av en "Virtuell pasientjournal". Nedenfor vil vi beskrive arkitekturen i dette systemet.

Pasientinformasjon gjøres tilgjengelig i den virtuelle pasientjournalen gjennom tre faser.

1. Indeksering av tilgjengelig pasientinformasjon.
2. Spørring mot katalogtjenere med indeksert informasjon.
3. Visualisering av pasientinformasjon.

Disse stegene kan illustreres med følgende figur:



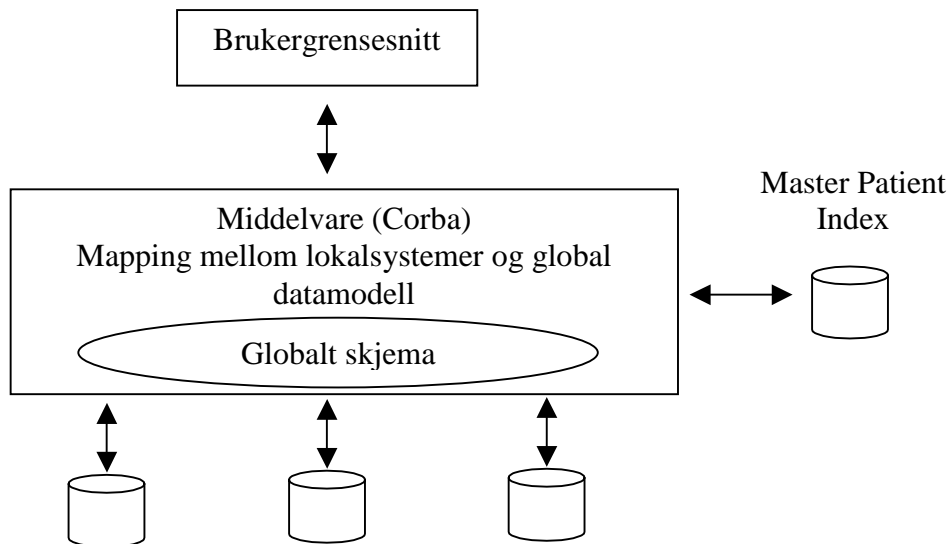
Figur 2 Indeksering, spørring og visualisering i BoJ

Før informasjon kan gjøres tilgjengelig, indekseres pasientinformasjon ved at produksjonssystemene eksporterer data til en katalogtjener som vist i steg 1. Når en bruker ønsker å aksessere informasjon om pasienten benyttes flere katalogtjenere til å finne hvilke systemer som har informasjon om pasienten som vist i steg 2. BoJ Index inneholder dataene vist nederst til høyre i figuren. Til slutt kan detaljinformasjonen overføres på brukerens forespørsel direkte fra produksjonssystemet, som vist i steg 3.

## 2.2 Baptist Health System of South Florida

Baptist Health System of South Florida (BHSSF) [2] betjener ca 2 millioner pasienter og medlemmer. En rekke helseinstitusjoner er tilknyttet dette nettverket. Kjernen i dette systemet er en indeks (Master Patient Index) som knytter sammen demografiske data og pasientjournaler i de enkelte institusjonene. Løsningen, som er basert på middelvare (CORBA), muliggjør fullstendig integrasjon mellom heterogene systemer og gir umiddelbar

tilgang til alle data om en pasient som kan være lagret i forskjellige databaser. Skjematisk kan løsningen beskrives med figur 3 nedenfor.



Figur 3 Skjematisk beskrivelse av BHSSF arkitekturen

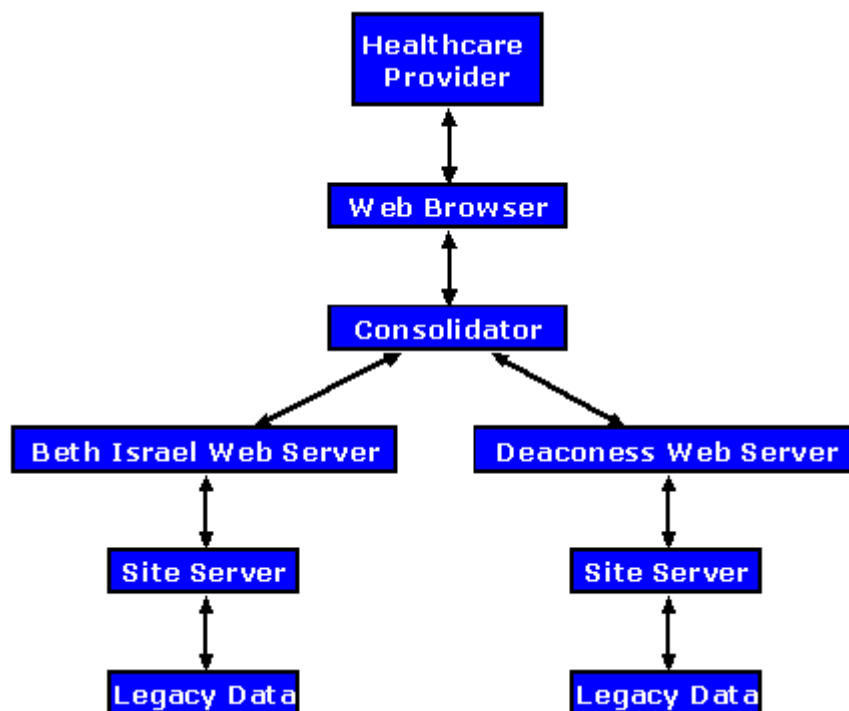
## 2.3 CareWeb

CareWeb-prosjektet har som målsetning å koble sammen de kliniske databasene til "The CareGroup Hospitals" via Internett. Forskningsgrupper ved Children's Hospital, MIT og Massachusetts General Hospital, har implementert en løsning som binder sammen Beth Israel og Deaconess sykehusene. CareWeb er en generell modell for hvordan en kan knytte sammen klinisk informasjon fra forskjellige helseinstitusjoner. Løsningen er også i stand til i formidle informasjon til og fra et nettverk av sykehus i Boston-området i USA. CareGroup Healthcare System i Boston, USA, består av 6 sykehus, med 2.500 tilbydere og 800.000 pasienter [3].

Utgangspunktet for systemet var at CareGroup ville tilby bedre behandling til en lavere kostnad for organisasjonen. Løsningen ble et intranetsystem som kunne vise pasientjournaler fra geografisk spredte pasienter ved å sammenstille pasientjournaldata og gjøre disse tilgjengelig via en webleser.

Arkitekturen i CareWeb systemet er vist i figur 4 nedenfor som er hentet fra CareWeb-prosjektets hjemmesider [4]. De kliniske datasystemene ved Beth Israel Hospital og Deaconess var forskjellige. CareWeb utviklet en "site server" som jobbet bak sykehusenes webservere og opprettet en link mellom web serverne og de kliniske datasystemene ved hver institusjon. "Site serveren" tolker innkommende Health Level 7 (HL7) forespørsler om informasjon, oversetter dette til spørringer mot det lokale datasystemet ved hver institusjon og pakker informasjonen inn i et HL7 svar. For å muliggjøre spørringer mot flere sykehus utviklet CareWeb en "Consolidator" som behandler brukerforespørsler, sender disse til institusjonenes "site server" og behandler svarene fra denne.

En typisk arbeidsøkt starter når en helsearbeider ved hjelp av en standard web leser oppretter en spørring ved å oppgi pasientens identitet (id-nummer). Denne informasjonen sendes til "Consolidator" ved hjelp av html-formularer (forms). Consolidator genererer en HL7 forespørsel til både Beth Israel og Deaconess site servere. Disse site serverne returnerer demografiske data om pasienten (ved hjelp av HL7 meldinger), problemer, medisiner, allergier, notater og besøk. Consolidator tolker disse innkommende dataene og lager en enhetlig sammenstilling som sendes tilbake til helsearbeideren som en serie med web sider. Denne prosessen er illustrert i figur 4 nedenfor.



Figur 4 Arkitekturen i CareWeb systemet.

## 2.4 HYGEIANet

HYGEIANet er et regionalt helsenett som knytter sammen 6 distriktssykehus, 2 regionsykehus, 16 helsesentre og 6 lokale legekantor på Kreta i Hellas. Miljøet på Kreta har jobbet i flere år med problemstillingene for denne rapporten [5] [7][8]. De har vurdert to alternative metoder for å gjøre pasientinformasjon tilgjengelig for å understøtte "continuity of care": En meldingsbasert tilnærming og en samarbeidende (eng. "federated") løsning basert på distribuerte multidatabaser bundet sammen ved hjelp av CORBA mellomvare.

Katehakis m.fl. [6] konkluderer med at meldingsbaserte systemer er utmerket til inter-system kommunikasjon som gjør informasjon tilgjengelig for brukeren. Svakheten ved meldingsbaserte systemer er at en ikke oppnår systemintegrasjon. En skaper også en redundans som kan føre til inkonsistens mellom forskjellige instanser av samme informasjonsobjekt. Et meldingsbasert system har også skaleringsproblemer. Samarbeidende (mellomvarebaserte) løsninger, basert på en infrastruktur som muliggjør en mapping av data til og fra et felles definert skjema, er en bedre løsning dersom en har behov for systemintegrasjon. En slik løsning unngår redundans (og potensiell inkonsistens) ved at informasjonen er lagret én plass. "Continuity of care" forutsetter et samarbeid mellom ulike avdelinger eller helseinstitusjoner om å tilby gode helsetjenester. Et slikt samarbeid kan ha

bedre støtte av et system som oppnår systemintegrasjon kontra et alternativ som bare overfører data mellom ulike datasystemer.

## 3 Visualisering

### 3.1 Innledning

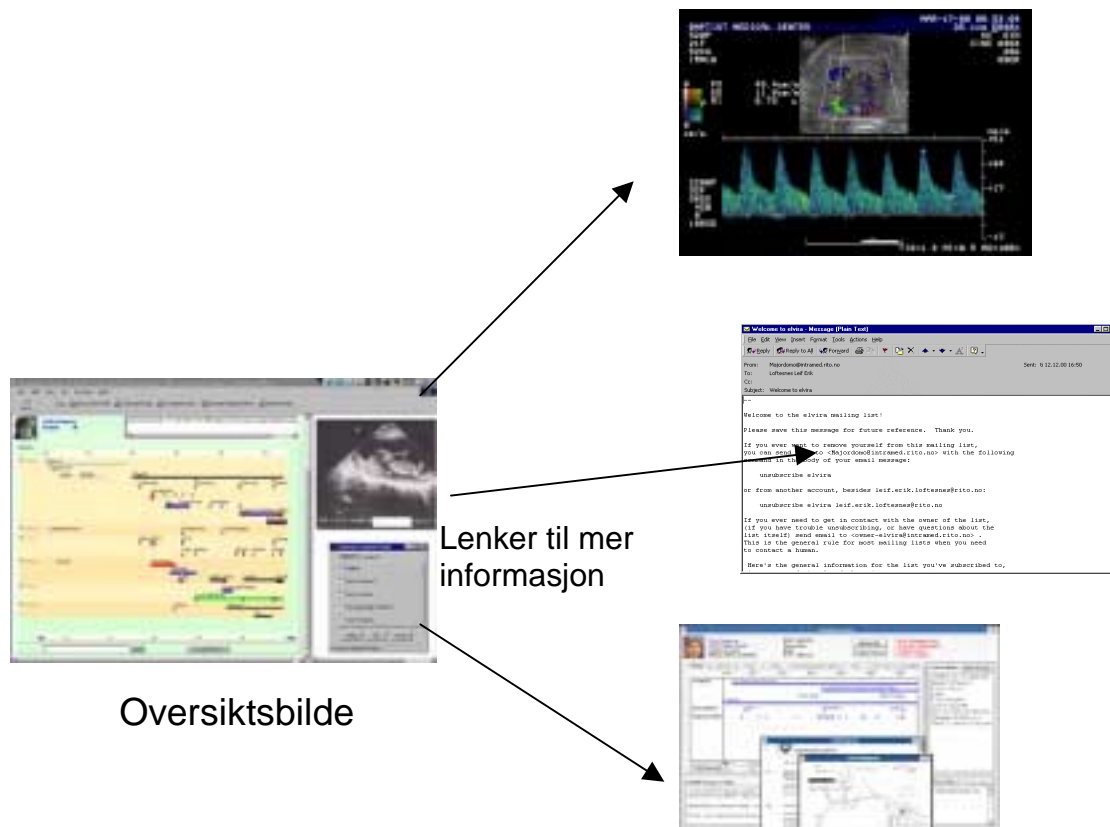
Allerede i dag inneholder journalsystemer mye informasjon om hver pasient. Enten journalen er elektronisk eller papirbasert kan den inneholde journalnotater, epikriser, røntgenbilder, prøvesvar og trygdeskjemaer. Det er vanskelig for journalbrukeren å skaffe seg oversikt over pasientens fulle sykehistorie, og relevant informasjon kan drukne blant alt det andre. Med ELVIRA vil informasjonsmengden øke ytterligere og det vil være helt nødvendig å fremstille denne informasjonen på en oversiktlig måte. Vi kan tenke oss en skala som strekker seg fra tilgang til all eksisterende informasjon om en pasient til informasjon som er begrenset til det som er relevant for den oppgaven som skal utføres.



Tilgang til fullstendig informasjon eller et sammendrag av alle tilgjengelig opplysninger stiller store krav til både systemets evne til å formidle opplysningene og brukerens ferdigheter til raskt å fremskaffe relevant informasjon for den oppgaven som skal løses. Den andre ytterligheten forutsetter derimot at presentasjonen av data er tilpasset spesielle oppgaver som skal utføres. Her er utfordringen å konstruere filtre og grensesnitt som viser nødvendig informasjon, og skjuler data som ikke er interessante.

Sannsynligvis vil en god løsning ligge mellom disse to ytterpunktene. I utgangspunktet bør så mye informasjon som mulig være tilgjengelig, men gruppert på en slik måte at det er mulig å skjule eller minimere for eksempel eldre røntgenundersøkelser dersom det dreier seg om en allergiundersøkelse. En fritt skalerbar tidsakse vil også være en form for brukerstyrt filter.

I dag har en lege svært kort tid til rådighet for hver pasient, i mange tilfeller ned mot 10-15 minutter. I løpet av denne tiden skal legen sette seg inn i pasientens sykehistorie, sette diagnose og foreslå behandling. Dersom legen skal bruke ELVIRA, må det derfor være mulig å raskt få oversikt over pasientens bakgrunn og eventuell tidligere behandling. Det er en klar forutsetning at ELVIRA er lett å lære for en førstegangsbruker. I dagens norske helsevesen er det mange vikarleger, og for at disse skal ha mulighet til å bruke ELVIRA må det legges vekt på et meget lettfattelig brukergrensesnitt som ikke krever mye opplæring.



**Figur 5 Hierarkisk organisering av pasientinformasjon**

ELVIRA vil samtidig kunne være et viktig verktøy for spesialisten i vurderingen av en pasient som det er vanskelig å sette diagnose på. Her er behovet at tilleggsinformasjon må være lett tilgjengelig, gjerne ved at brukeren beveger seg opp og ned i hierarkier i systemet. Slik tilleggsinformasjon kan være prøvesvar, røntgenbilder, tidligere journalnotater eller epikriser.

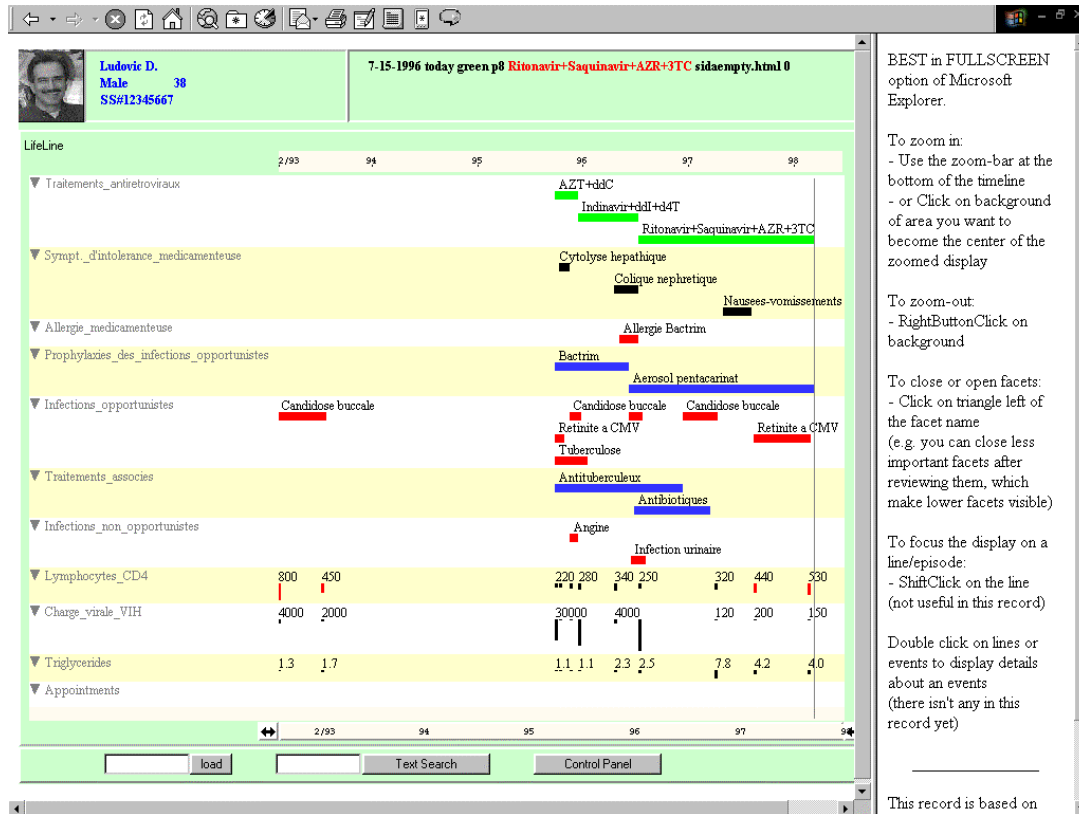
Figur 5 viser hvordan en spesialist kan ta utgangspunkt i informasjon om en pasient på toppnivå og bevege seg nedover til mer detaljerte opplysninger. Ray Simkus [9] påpeker hvor viktig det er at de dataene som vises på skjermen er relevante, spesielt for allmennlegen som må dele oppmerksomheten sin mellom pasienten og journalsystemet. Han kaller dette ”kognitiv belastning”, og han mener et journalsystem må ha innebygget filtre eller agenter som siler informasjon for brukeren.

### **3.2 Livsløpsfremstilling**

En måte å fremstille en pasients sykehistorie er å vise den som en tidslinje der hendelser er plottet inn. Ved hjelp av en grafisk fremstilt tidslinje kan brukeren enkelt få oversikt over og finne fram til enkelthendelser og deres plassering i tid. Slike hendelser kan være :

- Innleggelse og polikliniske besøk
- Journalnotater og epikriser
- Diagnoser
- Medisinering
- Røntgenbilder og analyser

En hendelse kan være tilordnet ett bestemt tidspunkt, for eksempel en operasjonsdato, eller en periode med start- og sluttdato. Farger kan også brukes for å vektlegge spesielle typer informasjon, for eksempel antibiotika-allergi eller kroniske sykdommer med betydning i en akuttsituasjon.



Figur 6 Livsløpfremstilling av sykehistorie

Et slikt system er utviklet ved University of Maryland [10]. Systemet kalles LifeLines, og er basert på en dynamisk tidslinje og flere grupper av hendelser. Figur 6 viser et skjermbilde fra LifeLines.

Pasientinformasjonen er delt opp i grupper som kan åpnes og lukkes. Det er også mulig å klikke på hendelser for å få vist ytterligere informasjon. LifeLines bruker også fargekoder for å skille ulike typer informasjon.

### 3.3 Videre arbeid

For å undersøke nytten av ELVIRA, kan det gjennomføres en studie av ulike grensesnitt til journalinformasjon. Under beskrives et forslag til utforming av en slik undersøkelse.

#### 3.3.1 Forslag til design av undersøkelse

For visualiseringen forutsettes det at aggregert informasjon på entydig dataformat er gjort tilgjengelig for programvaren. Dataformatet er basert på hendelser (legebesøk, røntgenundersøkelser eller lignende).

Vi har diskutert to hovedmuligheter for utforming av undersøkelsen:

- En nettbasert, visualisert journal sammenlignes med tradisjonell journal (mange ulike systemer, papir etc.) og det måles hvor raskt forsøkspersonene tilegner seg informasjon fra de ulike kildene.
- Nettbasert, visualisert informasjon vises som en demonstrasjon med realistiske kasus, og forsøkspersonene svarer på et spørreskjema etterpå. Relevante spørsmål i denne sammenhengen er for eksempel: ”Gjorde den visualiserte informasjonen det lettere å sette seg inn i pasientens sykehistorie ?” eller ”Vil tidslinjebasert informasjon bidra til at du får informasjon om pasienten du ellers ikke ville fått ?”

### 3.3.2 Case

I begge de to tilfelle må vi utarbeide realistiske kasuistikker. Disse kan enten utledes fra eksisterende systemer eller konstrueres. Kasuistikkene må legge vekt på situasjoner der historisk informasjon om pasienten er viktig for behandling.

Eksempler på dette

- Pasient som tidligere har hatt alvorlige allergiske reaksjoner på spesielle typer narkose kommer inn som (bevisstløs) akuttpasient.
- Pasient med nylig behandlet hjertesvikt som er innkalt til en rutinemessig operasjon
- Pasient med diabetes blir brakt bevisstløs til sykehus

Dette er situasjoner der pasienten glemmer å opplyse om viktig opplysninger, er ute av stand til å fortelle det (bevisstløshet) eller unnlater å gjøre det av frykt for konsekvenser (utsatt operasjon eller lignende).

Sykehistoriene må så omgjøres til hendelser. Hendelsen grupperes i kategorier som for eksempel:

- Legebesøk/innleggelse
- Allergier
- Notater (journal)
- Undersøkelser (lab, rtg, MR etc)
- Kroniske tilstander som diabetes eller lignende

Innenfor hver kategori kan hendelsene i tillegg ha fargekoder slik at en røntgenundersøkelse for eksempel er grønn, mens en blodprøve er rød.

En hendelse har kategori, et tidspunkt og en fargekode (underkategori).

### 3.3.3 Forsøkspersoner

Testen bør utføres på helsepersonell som er hovedbrukere av journalene, i hovedsak leger. Det bør gjøres et representativt utvalg av ulike sykehusleger med ulike spesialiteter og allmennpraktikere.

### 3.3.4 Spørreskjema

Hovedmålet med undersøkelsen er å finne ut om en tidslinjebasert sykehistorie har et bruksområde, det vil si om det gir brukerne bedre og raskere informasjon som igjen gir et bedre beslutningsgrunnlag.

Utformingen av spørreskjemaet vil avhenge av utformingen av testcasene, dvs om vi satser på en sammenligning av ”gammel” og ”ny” teknikk eller om vi baserer oss på bare ”ny” teknikk.

## 4 Alternative arkitekturløsninger

### 4.1 Felleskomponenter og tjenester for alle alternativer

Arkitekturalternativene, som vil bli skissert nedenfor, har noen felles behov. Enkelte av disse behovene kan realiseres som felles systemkomponenter eller tjenester i helsenettet mer eller mindre uavhengig av hvilket system som skal benytte disse. Disse felleskomponentene er nært knyttet til sikkerhet og behovet for standardisering. Se rapportene om sikkerhet og standardisering for en mer detaljert beskrivelse av behovene disse felleskomponentene skal løse.

Visjonen i Elvira er at informasjon skal kunne overføres mellom helseinstitusjoner. En felles forståelse av hvilke personer og funksjoner/roller som har tilgang til informasjonselementer er da nødvendig. Miljøet på Kreta, Hellas har jobbet med denne problemstillingen [7].

Autentisering og autorisasjon er derfor funksjonalitet som må være felles for alle systemene som skal gjøre informasjon tilgjengelig via helsenettet. Helsepersonell er mobile og kan være ansvarlig for behandling ved flere institusjoner. Ambulerende spesialister er en slik gruppe. En bør ha som ambisjon at helsepersonell skal kunne benytte samme metode for tilgang til systemer og informasjon uavhengig av hvilken institusjon helsearbeideren befinner seg i. En ”Public Key Infrastructure” (PKI) løsning synes å være aktuell teknologi for å realisere denne funksjonaliteten. En anbefaling for videre arbeid er derfor at en setter av ressurser til å realisere felleskomponenter og -tjenester som understøtter visjonen om at helsepersonell får tilgang til tjenester og den informasjon de er autorisert til uavhengig av hvilken institusjon vedkommende måtte befinne seg i (se rapport fra arbeidspakken for sikkerhet).

Logging av informasjonstilgang er også kandidat som en fellestjeneste levert av helsenettet. Denne er beskrevet nedenfor.

Som nevnt i innledningen ser en for seg et felles system for visualisering av pasientinformasjon. Et slikt system kan bare realiseres dersom man har en felles datamodell som informasjonen i leverandørsystemene kan konverteres/mappes til. Omfanget av en felles datamodell er avhengig av hvilken systemarkitektur som velges og i hvilken grad en velger å standardisere datatyper som skal visualiseres via ett visualiseringssystem. Dette er beskrevet nærmere nedenfor.

#### 4.1.1 Loggtjeneste

En effektiv begrensning av innsyn i pasientinformasjon er å gjøre tilgjengelig hvem som har hatt innsyn i informasjon. Ett forslag<sup>1</sup> er å opprette et system som kan generere rapporter om hvem som har hatt innsyn i journalen på samme måte som en i dag mottar melding ved innsyn i kredittopplysninger. I tilknytning til en slik loggtjeneste må en også ha rutiner som medfører

---

<sup>1</sup> Forslag fra Jan Størmer under Elvira-seminaret avholdt høsten år 2000

at innsyn uten legitimt formål blir oppdaget. En kan tenke seg en slik tjeneste realisert som en sentralisert loggtjeneste i helsenettet i tilknytning til et nettbasert pasientinformasjonssystem. En slik tjeneste vil fungere som en effektiv bremsekloss for illegitimt innsyn i journalinformasjon.

#### **4.1.2 Felles datamodell**

Visjonen om en nettbasert journal hvor informasjon fra mange systemer er tilgjengelig i ett system, forutsetter enighet om en felles datamodell. Omfanget av denne datamodellen avhenger i stor grad av hvordan informasjon fra de enkelte leverandørsystemene gjøres tilgjengelig for brukeren. De enkelte arkitekturalternativene nedenfor vil ha forskjellig behov for felles datamodell.

##### *Minimal felles datamodell*

Dersom det enkelte leverandørsystem selv er ansvarlig for å gjøre detaljinformasjonen tilgjengelig (og visualisere denne informasjonen), vil omfanget av felles datamodell være relativt liten og ikke omfatte format på alle informasjonstyper. Felles datamodell vil i dette tilfellet beskrive medisinske hendelser (tidspunkt, varighet, type hendelse o.l.) og hvordan informasjonselementet som representerer hendelsen linkes til originaldata. Brukeren av informasjonen får tilgang til original/detaljdata ved å følge linkene som leverandørsystemene vil være ansvarlige for å levere. En slik minimal datamodell kan ha uheldige effekter som at samme type data visualiseres forskjellig avhengig av systemleverandør. Slike forskjeller kan forvirre brukeren, men man oppnår likevel en enhetlig presentasjon av overordnede hendelser.

##### *Maksimal felles datamodell*

I motsatt ende av skalaen for omfang av felles datamodell finner vi arkitekturalternativer som forutsetter at leverandørsystemene leverer data på et standardisert format og overlater ansvaret for å gjøre denne informasjonen tilgjengelig for brukeren til et annet visualiseringssystem. Leverandørsystemene må da foreta en mapping fra sin interne datamodell til en felles og standardisert datamodell. En slik felles datamodell må omfatte beskrivelse av alle datatyper som skal kunne visualiseres gjennom systemet. Faren ved dette alternativet er at systemet som leverer data ikke har kontroll med hvordan informasjonen visualiseres. Det kan da oppstå situasjoner hvor data blir tolket feilaktig fordi feil oppstår under konvertering fra leverandørsystemets datamodell/format til felles datamodell/format.

Mellom disse ytterpunktene finnes en mengde alternative løsninger. Den mest realistiske løsningen er at man begynner en prosess fra å ha en minimal felles datamodell i utgangspunktet og gå mot en maksimal felles datamodell ettersom standardisering på informasjonselementnivå blir mer omfattende.

#### **4.1.3 Pasientinformasjonsindeks**

I tillegg til tjenestene/komponentene ovenfor er alle løsninger for nettbasert tilgang til pasientinformasjon avhengig av indekser som forenkler tilgang/søking etter pasientinformasjon. En forutsetning for at en skal kunne lage slike indekser er at en kan identifisere en pasient entydig. Denne problematikken er omtalt i rapporten "Relevante standarder for elektronisk journal". Omfanget av data det er nødvendig å lagre i felles indekser er forskjellig i de enkelte arkitektur-alternativene. Disse forskjellene vil bli beskrevet nedenfor.

## 5 Elvira som datavarehus

Et datavarehus er et databasesystem med *historiske* og *detaljerte* data over bedriftens/organisasjonens virksomhet, hvor målsettingen gjerne er å fremskaffe strategisk informasjon. Datavarehus muliggjør analyse av detaljdata og trender i bedriftens virksomhet.

Data vil være samlet inn over lengre tid, gjerne fra flere ulike operasjonelle transaksjonssystemer eller Internett/Intranett. En av de kritiske suksessfaktorene i et datavarehus er evnen til å lagre og gjenfinne alle spor etter kunden. Informasjonen anvendes i neste omgang som beslutningsgrunnlag, både strategisk og operasjonelt.

Overført til Elvira betyr dette at alle pasientrelaterte data tilgjengeliggjøres via en datavarehus-løsning.

Konstruksjonen av et datavarehus kan foregå i tre faser (jfr Tabell 1):

1. Data-arkitektur
2. Teknisk arkitektur
3. Infrastruktur

En solid datavarehusarkitektur inkluderer alle tre fasene.

Røed har i Tabell 1 listet opp et rammeverk for datavarehus. Rammeverket bygger på rammeverket til John A. Zachman (Zachman International).

**Tabell 1 Rammeverk for datavarehus**

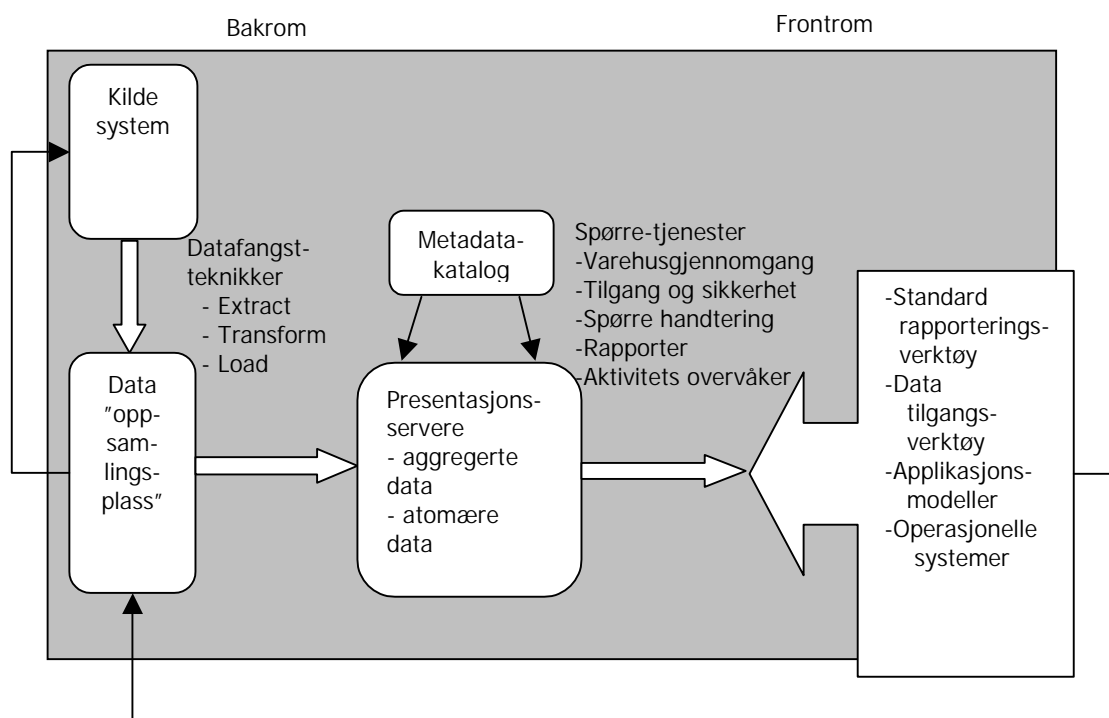
Detaljnivå	Dataarkitektur	Teknisk arkitektur		Infrastruktur
		"Bakrommet"	Sluttbrukere	
Forretningskrav	Hvilken informasjon trenger vi for å fatte bedre beslutninger?	Hvordan får vi tak i dataene, omformet disse og tilgjengeliggjort dem?	Hvilke spørsmål vil vi ha svar på? Hvordan måles disse? Hvordan skal dataene analyseres?	Hvilke hardware behøves? Hva slags kapasitet trenger vi? Hva har vi i dag?
Arkitekturmodeller og dokumenter	Overordnet datamodell (entiteter) ev. dimensjonsmodell	Hvilken ekstraheringskapasitet behøves? Hvor er datakildene? Til hvilken lokasjon skal dataene flyttes?	I hvilken form ønsker brukerne rapportene? Hva slags type rapporter ønskes, og hvordan skal disse prioriteres?	Har vi tilstrekkelig diskkapasitet og prosesseringskraft? Hvem er ansvarlig for infrastrukturen?
Detaljerte modeller og spesifikasjoner	Detaljerte logiske modeller, stjerneskjemamodeller, database-diagrammer	Hvilke produkter støtter våre behov? Hvordan vil vi integrere dem? Hvilke standarder vil vi følge?	Detaljerte rapportspesifikasjoner. Hvem skal ha rapportene? Hvor ofte? Hvordan distribuerer vi dem?	Hvordan klarer vi å støtte rapportspesifikasjonene? Hvilke utilities finnes?
Implementasjon	Opprett database, indekser, backuprutiner etc. Dokumenter.	Utvikle ekstraheringer og loader. Automatiser prosessen. Dokumenter.	Implementer rapporteringsmiljø, utvikle de første rapportene, lær opp brukerne. Dokumenter.	Installer og test nye infrastrukturkomponenter. Sjekk all kommunikasjon. Dokumenter.

(Kilde: Røed, D. "en introduksjon til datavarehusarkitektur." [www.datavarehus.net](http://www.datavarehus.net))

I dataarkitektur angis innholdet i datavarehuset (*hva*). I teknisk arkitektur angis *hvordan* data skal fremskaffes og presenteres. Infrastruktur spesifiserer egnet maskinvare og programvareplattform.

### 5.1.1 Dataarkitektur

ELVIRA skal i første omgang integrere flere pasientjournalssystemer og pasientinformasjonssystemer. På sikt vil også annen informasjon av medisinskfaglig, administrativ og økonomisk art kunne integreres. På denne måten vil vi kunne utnytte styrken i datavarehuskonseptet. Figur 7 illustrerer en mulig høynivå teknisk arkitektur for et datavarehus. Det overordnede målet er at arkitekturen tilfredsstiller de krav som forretningssiden fremsetter. Utgangspunktet for datavarehuset er ett eller flere kildesystem. Dataene blir hentet ut fra datakildene, restrukturert, mv., av dataoppsamlingsplass-modulen (eng., "data staging area"). Deretter lastes dataene inn i datavarehusdatabasen. Fra denne hentes informasjonen ut ved hjelp av ulike spørre- og analyseverktøy. Metadatakatalogen fungerer som datavarehusets innholdsfortegnelse, og gir oversikt over hele prosessen, inkludert hvor, når, definisjoner, tilgjengelighet, etc.



Figur 7 Eksempel på høynivå teknisk arkitektur (basert på modell fra Ralph Kimball)

### 5.1.2 Infrastruktur

Datavarehuskonseptet stiller store krav til infrastrukturen. Datavarehuset skal behandle og integrere data lagret i mange forskjellige datasystemer, gjerne spredt over et stort geografisk område. For å betjene et stort antall brukere vil man ofte benytte flere datamaskiner som kjører i parallell. Kravet til lagringskapasitet er også ofte meget store, typisk målt i terabyte.

Siden veksten i datamengde ofte er progressiv, stilles det store krav til skalerbarhet i datavarehus. Kravene til beregningskraft øker ofte proporsjonalt med databasen.

Etterhvert som man ser nytteverdien av datavarehuset, søkes også flere bruksområder. Man får nye brukergrupper, noe som også setter høyere krav til maskinytelsen. Mindre datavarehus med datamengder opp til 40 GB, vil kunne realiseres med en datamaskin med 2-4 prosessorer. På grunn av en forventet rask vekst i datamengden vil diskene være eksterne. Systemet kan skales ved å øke antall prosessorer og eksterne lagringsmedia.

For datavarehus med datamengder opp mot 200 GB, vil løsningen over, men med flere prosessorer, fortsatt kunne være aktuell. Overstiger lagringsmengden 200 GB, vil dette som regel kreve flere flerprocessor datamaskiner. NCR leverer f.eks. løsninger opp til 512 fire-prosessor maskiner med mer enn 100 Terabyte data.

### **5.1.3 Elvira som datavarehus**

En datavarehusløsning kan på mange måter løse de problemer som Elvira står overfor med hensyn på integrering av data. Men datavarehus-konseptet løser kun i liten grad den største utfordringen som Elvira-prosjektet står overfor – utforming av brukerdiallog og integrasjon av de ulike datakilder, kombinert med krav til sikkerhet og brukergruppetilpassede løsninger.

Den store fordelen med Elvira som datavarehus er at institusjonene som produserer informasjon eksporterer data til en fast formidler. Informasjonsprodusenene kan på denne måten ha en helsenett tilknytning der det kun er mulig å åpne nettverksforbindelser fra innsiden og ut. En eliminerer muligheten til oppkobling fra helsenettet med formål å finne pasientinformasjon i institusjonens datasystem. Dette representerer fordeler for institusjonen i form av reduserte sikkerhetstrusler og redusert behov for maskinpark for å betjene eksterne institusjoners informasjonsbehov.

## **5.2 Multidatabase-tilnærming**

I et multidatabase-system (MDBS) gjør hver enkelt deltakende database (her et helse-informasjonsystem) sine data eller et subsett av dataene tilgjengelig utenifra. Dataene som tilgjengeliggjøres er de dataene som kan deles med andre. Strukturen eller datamodellen til de deltakende databasene vil være ulik fra system til system, fordi ulike systemer er i bruk i det norske helsevesenet - også forskjellige versjoner av de samme systemene. De deltakende databasene – eller fragmentene som eksporteres fra hver enkelt database, kan dermed sies å være heterogene. Heterogenitet gjør integrasjon av data vanskeligere – det blir vanskelig å danne en enhetlig modell for dataene avhengig av hvor ulikt dataene er representert. De ulike systemene er ikke designet med tanke på integrasjon av data med andre systemer, og designet av en felles datamodell må gjøres fra bunnen av og opp ("bottom-up"). En måte å integrere fragmentene på, er å først oversette hvert enkelt fragment ned til et felles mellomformat, ofte kalt en kanonisk representasjon. For hver enkelt type fragment må det lages en slik oversetting. Deretter kan hver enkelt fragments kanoniske representasjon integreres til en overordnet eller felles datamodell [11]. Det vanskelige er imidlertid å lage en kanonisk representasjon.

En felles datamodell, også kalt globalt konseptuelt skjema, er ønskelig hvis det skal være mulig å utføre globale spørringer (søk) mot en MDBS. En slik spørring kan være "Hent datoer, måleverdier og institusjonsnavn for måling, for blodtrykksmålinger utført i løpet av siste fem år for gitt person". For å utføre slike spørringer er det hensiktsmessig å konstruere et lag som mottar spørringen og omgjør den til del-spørringer på hvert fragment. Laget utfører disse delspørringene og returnerer svarene ferdig integrert til brukerprogrammet, som sendte forespørselen på vegne av brukeren. En sentral forutsetning for et slikt multidatabase-lag er at det finnes en katalog som inneholder opplysninger om plasseringen av hvert enkelt fragment, og strukturen til fragmentene. Denne katalogen bør enten selv være distribuert eller repliseres, for å unngå konsentrasjon av belastning og redusere sårbarhet. Av samme grunn bør multidatabase-laget heller ikke være en sentral tjeneste, men finnes tilgjengelig i flere instanser.

### 5.2.1 Problemstillinger

En vanskeliggjørende faktor ved realisering av et multidatabasesystem, er at de ulike deltakende systemene gjerne bruker forskjellig teknologi både i forhold til databasesystem/lagringsteknologi, operativsystem, nettverk osv. Håndtering av denne typen teknologisk heterogenitet regnes generelt sett for å være en av de største utfordringene ved utvikling av distribuerte systemer. Ulike mellomvareteknologier ("middleware") er utviklet og utvikles for å håndtere slik heterogenitet, mest kjent av disse er kanskje CORBA (Common Object Request Broker Architecture, OM Group). Gatewayprodukter er også tilgjengelige for å håndtere ulikheter i bla. databasesystemer.

Globale spørringer er ressurskrevende å utføre. Utføringen kan blant annet medføre betydelig kommunikasjonsoverhead ved at store mengder data kan måtte overføres. Responstiden for globale spørringer kan bli høyere enn det brukere vanligvis er rede til å akseptere for interaktivt bruk. Globale spørringer medfører også ekstra belastninger for de deltakende systemene. Hvorvidt de deltakende systemene kan håndtere den ekstrabelastningen det vil være å motta slike spørringer uten at dette går ut over den ordinære produksjonen, vil variere fra systeminstallasjon til systeminstallasjon. Det er rimelig å anta at slik ekstra belastning ikke kan aksepteres for en (muligens høy) andel av de deltakende systemene, og at informasjonen fra disse systemene må eksporteres til et annet databasesystem for å kunne være tilgjengelig i en multidatabase-løsning som beskrives her. For slik eksport av data fra produksjonssystem vil datavarehusteknikker kunne benyttes. Dette medfører i så fall et ekstra teknologisk og driftsmessig nivå, og øker kompleksiteten.

### 5.3 Mobil agent løsning (meldingsbasert)

Dette arkitekturalternativet er basert på forskningsresultater (konsepter, modeller og programvare) utviklet innenfor Virtual Secretary prosjektet ved Universitetet i Tromsø, institutt for informatikk [12] [13]. Resultatene fra dette prosjektet vil benyttes og programvaren videreutvikles innenfor DiPato prosjektet ved Nasjonalt Senter for Telemedisin [14]. Programvaren implementert innenfor Virtual Secretary prosjektet har demonstrert at det er mulig å implementere et system som støtter asynkron utførelse av oppdrag på fjerne maskiner på vegne av en bruker. Med førstehånds kjennskap til mulighetene dette systemet gir, vil vi her beskrive en arkitektur basert på resultatene fra Virtual Secretary prosjektet.

Innledningsvis vil vi gå gjennom en del basale konsepter for å gjøre beskrivelsen av arkitektur mer forståelig for personer som ikke har førstehånds kjennskap til Virtual Secretary prosjektet og programvaresystemene utviklet innenfor dette prosjektet.

#### 5.3.1 Grunnkonsepter for mobile agenter

Visjonen bak arkitekturen er at en bruker kan få utført oppdrag på fjerne datamaskiner uten selv å være i aktiv dialog med disse datamaskinene. Når resultatene av oppdraget er klare skal disse kommuniseres til brukeren der denne måtte befinne seg via den datamaskinen brukeren for øyeblikket benytter. Analogien er en sekretær som utfører oppdrag uavhengig og på vegne av brukeren. Brukeren i dette systemet kan være mobil og benytte en mengde forskjellige datamaskiner. Entitetene som utfører oppdrag på vegne av en bruker kalles agenter og har innenfor dette konseptet evnen til å hoppe mellom datamaskiner.

##### *Agenthierarkiet*

Oppdrag som utføres på vegne av en bruker kalles agentoppdrag. Et oppdrag realiseres som regel av flere agenter. Agentene er organisert i et hierarki som består av tre typer agenter.

Disse typene er "Homebase agent", "User agent" og "Mission agent". Rent teknisk realiseres en instans av en agent som en eller flere prosesser på en datamaskin.

Homebaseagenten er en representant for brukeren når denne ikke er tilgjengelig og kontrollerer initiering av useragenter, men kan også benytte missionagenter. Homebaseagenten fungerer analogt til en sekretær som koordinerer oppdrag, beskjeder og lignende ved brukers fravær.

Når brukeren blir tilgjengelig (logger seg på en datamaskin) vil homebaseagenten sende en useragent til datamaskinen brukeren benytter. Useragenten er utstyrt med et brukergrensesnitt for å ta imot nye oppdrag og formidle resultater av agentoppdrag til brukeren. Useragenten utfører et oppdrag ved å sende en eller flere missionagenter til fjerne maskiner. Missionagentene som utfører et agentoppdrag er underlagt kontroll av useragenten som igangsatte oppdraget. En missionagent opererer med brukers identitet og rettigheter på fjerne maskiner. En missionagent kan ikke ha flere rettigheter enn brukeren eventuelt ville ha om denne benyttet det fjerne systemet direkte. Når missionagenten(e) har utført sitt oppdrag og levert eventuelle resultater til useragenten, har denne ansvaret for formidling av resultatene til brukeren. Dersom brukeren har logget seg av før et oppdrag er utført, overføres kontrollen over oppdrag til homebaseagenten. Når brukeren blir tilgjengelig igjen (potensielt på en annen datamaskin) vil homebaseagenten overføre kontrollen over agentoppdrag tilbake til useragenten der brukeren befinner seg. Useragenten er alltid underlagt homebaseagentens kontroll. Homebase kan beordre useragenten til å avslutte sin virksomhet på en datamaskin og overlate kontroll over oppdrag til homebaseagenten eller flytte sin virksomhet til en annen datamaskin, typisk dit brukeren har flyttet.

Missionagentene er arbeiderne i systemet og har som regel ikke noe brukergrensesnitt for kommunikasjon med brukere. Missionagenter kan benytte andre missionagenter til å utføre deloppdrag på vegne av seg og utøver kontroll over disse.

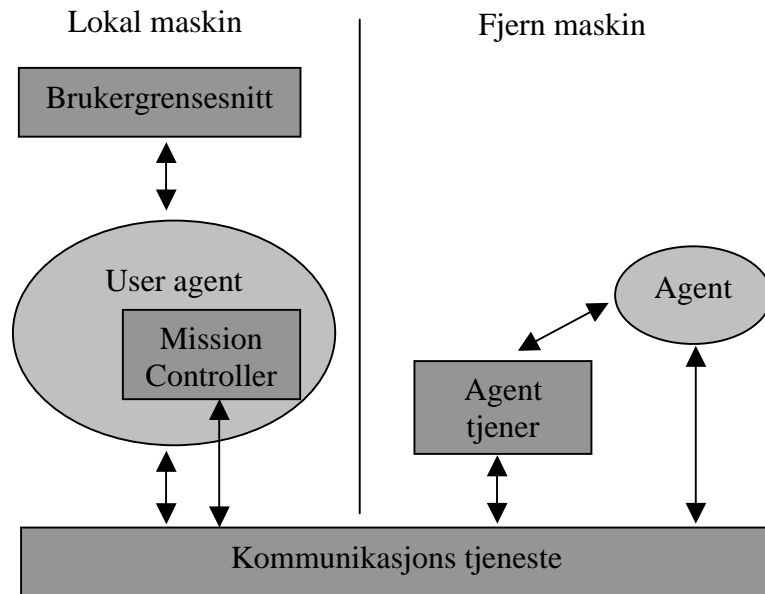
#### *Konstruksjon og programkontroll av agenter*

Agenter eller programprosesser opprettes som resultat av en forhandling mellom en foreldreagent (via system tjenester/komponenter) på en maskin og en agenttjener på en fjern maskin. Dersom foreldreagent og agenttjeneren på fjern maskin blir enige om at en agent kan startes opp på den fjerne maskinen, konstruerer agenttjeneren på den fjerne maskinen agenten (en eller flere prosesser). Agenten konstrueres på basis av programvare tilgjengelig på den fjerne maskinen. Programvaren (objekt/program kode) som agenten konstrueres fra må installeres av en bruker (systemeier) og godkjennes for bruk før agenttjener er i stand til å konstruere en agenttype. Agenten er underlagt kontroll av agenttjeneren på den fjerne maskinen og foreldreagenten. Se figur 8 nedenfor.

Det eneste unntaket fra denne regelen er homebaseagenten som initieres av brukeren selv.

### Systemkomponenter

Hovedkomponentene i systemet er agenttjeneren som er installert på maskiner som skal støtte mobile agenter og en "Mission controller" komponent som hver agent som kan initiere agentoppdrag har. Mission controller komponenten er ansvarlig for å initiere agentoppdrag, forhandle med agenttjener på fjerne maskiner og handtere agentoppdrag-resultater. Skjematisk kan disse komponentene visualiseres med figur 8.



**Figur 8** Systemkomponenter for implementasjon av mobile agenter

Brukeren kommuniserer med useragent via et brukergrensesnitt. Useragent benytter mission controller komponenten til å initiere agentoppdrag på en fjern maskin. Hvis mission controller og agenttjener blir enige om at et agentoppdrag kan utføres på den fjerne maskinen, oppretter agenttjeneren agent prosessen(e). Agenten er nå under kontroll av både agenttjeneren, mission controller/ useragent. Mission controller /useragenten, agenttjener og agent kommuniserer via en felles kommunikasjonstjeneste. Dersom agenten skal besøke flere maskiner som del av sitt oppdrag, forhandler denne direkte med agenttjener på fjern maskin om lov til å migrere til den nye maskinen. Dersom agenten trenger/ønsker å utføre deler av sitt oppdrag ved hjelp av andre agenter, vil agenten kunne initiere slike oppdrag via en egen mission controller.

Agentoppdrag kan utføres etter to metoder: I parallell eller i sekvens. Dersom et oppdrag utføres i parallell vil mission controller forsøke å opprette flere samtidige agenter, en agent pr. maskin som omfattes av oppdraget. Dersom oppdraget utføres i sekvens, hopper den "samme" agenten fra maskin til maskin inntil alle datamaskiner som skal dekkes av oppdraget er besøkt. Eventuelle data som agenten genererer eller finner under et oppdrag kan rapporteres tilbake direkte, eller bringes med ettersom agenten flytter fra maskin til maskin.

### Sikkerhetsaspekter

Alle agenter opererer med brukerens identitet og rettigheter på maskinene som besøkes. En agent kan således ha mange rettigheter på en maskin eid av brukeren og et veldig begrenset sett av rettigheter (eller ingen) på andre.

### 5.3.2 Innsamling og visualisering av pasientinformasjon

Dette arkitekturalternativet er basert på at mobile agenter besøker alle systemer som har informasjon om en pasient og samler inn denne informasjon for senere visualisering for brukeren. Den mobile agenten kan samle inn metainformasjon og linker til informasjon eller all informasjon som kan visualiseres. I tilfellet hvor kun metainformasjon og linker samles inn, må detaljinformasjon kunne overføres direkte til brukeren på et senere tidspunkt ved f.eks å benytte URL'er/linker som leverandørsystemene kjenner. Dette er en mulighet som vi ikke vil forfølge videre i beskrivelsen av dette arkitekturalternativet, selv om dette alternativet er en like reell løsning som innsamling av alle data. En kombinasjon av disse alternativene er også mulig.

#### *Forutsetninger*

Som beskrevet ovenfor er dette arkitekturalternativet basert på at det blir "lovlig" å opprette en forbindelse fra en datamaskin tilknyttet helsenettet til agenttjeneren/agenten innenfor en institusjons datanett. Dette betyr at en må åpne for å opprette forbindelser fra helsenettet, gjennom institusjonens brannmurløsning, og inn til en datamaskin som kjører en agenttjener. En slik løsning er i dag ikke lovlig i henhold til datatilsynets retningslinjer. Det eneste gjenværende alternativ dersom en ikke kan gjøre dette er at leverandørsystemene eksporterer data til et felles "datavarehus", som beskrevet ovenfor.

En annen forutsetning er at alle brukere som skal ha tilgang til pasientinformasjon via helsenettet benytter samme autentiserings- og autoriseringsteknologi, og at systemene som skal levere pasientinformasjonen har implementert støtte for disse funksjonene. Hver helsearbeider må da være utstyrt med smartkort eller lignende som kan oppbevare den private nøkkelen på en sikker måte. Helsenettet må da inneholde tjenester for publisering av offentlige nøkler og verifisering av gyldigheten til disse nøklene. Se rapport fra arbeidspakken "Sikkerhetsaspekter".

For å redusere antall datamaskiner som må besøkes for å samle inn pasientinformasjonen, forutsetter vi at det vil være mulig sett fra et sikkerhetsperspektiv å bygge opp en indeks over hvor pasientdata befinner seg. Disse indeksene trenger i minimal form kun å inneholde adressen til datamaskinene som inneholder pasientopplysningene og en entydig identifikator for pasienten. Denne indeksen må sees på som sensitiv på lik linje med pasientinformasjonen og må beskyttes på lik linje med annen pasientinformasjon.

#### *Initiering av agentoppdrag for innsamling av pasientinformasjon*

Med unntak av akutt situasjoner, tilfeller hvor pasienten tar kontakt med helsevesenet uanmeldt og lignende, vet en i god tid på forhånd at et behov for pasientinformasjon vil oppstå. Denne forutsigbarheten åpner for innsamling til og sammenstilling av pasientinformasjonen lokalt på datamaskinen brukeren benytter, før informasjonen gjøres tilgjengelig for en autorisert bruker. Ved øyeblikkelig behov for informasjon må en nødvendigvis vente til et agentoppdrag for innsamling og sammenstilling av pasientinformasjon er fullført før brukeren kan få tilgang til informasjonen. Hva tidsforbruket til å utføre en slikt oppgave er, er vanskelig å anslå uten først å implementere deler av systemet og utføre responstidseksperimenter. Responstiden vil nødvendigvis også avhenge av tilgjengelige maskinressurser der pasientinformasjonen ligger lagret.

I de tilfeller der en har rikelig med tid (minutter) til innsamling av informasjon vil responstiden for brukeren være uavhengig av nettbelastningen og gi optimal responstid. Dette er en stor fordel i forhold til forbindelsesorienterte alternativer. Responstiden vil være optimal

fordi informasjonen vil være tilgjengelig lokalt på maskinen som brukeren benytter fordi user-agenten har hentet informasjonen til datamaskinen som brukeren benytter. En kan for eksempel tenke seg at resepsjonen på en legevakt har ansvar for å initiere innsamling av pasientinformasjon, men at kun legen som mottar pasienten får tilgang til pasientinformasjonen. Dette kan gjøres ved at en benytter mottakende leges offentlige nøkkel til kryptering av pasientinformasjonen som samles inn. Dette vil igjen bety at kun denne legen kan dekryptere pasientinformasjon som er samlet inn ved at legen benytter sin personlige nøkkel som ligger lagret på et smartkort. Dersom en ikke vet identiteten til personen som skal motta informasjon og behandle pasienten, men vet "identiteten" til avdelingen kan en for eksempel benytte avdelingens "offentlige nøkkel" ved kryptering og avdelingens "private nøkkel" til dekryptering av pasientinformasjonen. Det er også mulig å begrense innsynet i innsamlet informasjon basert på helsearbeiderens rolle eller hvilket formål som benyttes som begrunnelse for innsynet i pasientinformasjonen.

I de tilfeller det eksisterer et avtalt møte med pasienten kan informasjon fra en avtalebok benyttes til å initiere innsamlingen automatisk og når ressursutnyttelsen på maskiner og helsenett er lav.

#### *Innsamling av pasientinformasjon*

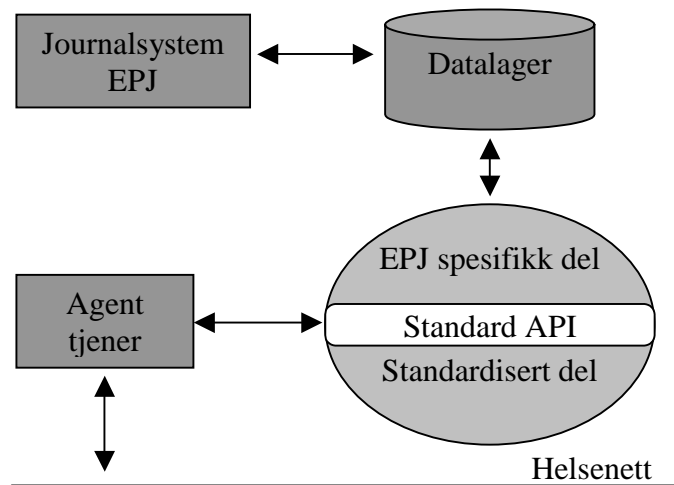
Ved initiering av informasjonsinnsamling må to betingelser være oppfylt, a) at en har tilgang til pasientens unike identifikator og b) at en har tilgang til mottakerens identitet eller avdeling (for oppslag i katalog over offentlige krypteringsnøkler). Pasientens unike identifikator benyttes til oppslag i indekser for hvor informasjon er tilgjengelig. Mottakerens identitet benyttes til å verifisere at denne har autorisasjon til å se informasjonen i indeksen. En slik innsamling må som nevnt i rapporten "Relevante standarder for nettsentisk journal", være basert på et besluttet tiltak. Det besluttede tiltaket vil i dette tilfelle bety initiering av et agentoppdrag for innsamling av data.

Et agentoppdrag for innsamling av pasientinformasjon vil først besøke datamaskinen som inneholder indeksen over hvor informasjon er tilgjengelig. Her vil komplett liste over datamaskiner som må besøkes for innsamling av informasjon være tilgjengelig. Når agenten har denne informasjonen kan den velge å samle inn informasjonen i parallell eller sekvensielt avhengig av responstidskrav.

#### *Ekstrakt av pasientinformasjon fra leverandørsystem*

Ved innsamling av pasientinformasjon vil agenten måtte forholde seg til en rekke forskjellige informasjonssystemer. Informasjonen i disse systemene må mappes fra leverandørsystemets datamodell til en felles datamodell. Dette kan gjøres ved å definere et standard applikasjonsgrensesnitt (API) (ala middelve løsnings ovenfor) som gjør konverteringen mulig og/eller definisjon av et felles dataformat for informasjonen som utveksles. Agenten må derfor konstrueres av programvare spesifikk for det lokale systemet og en standardisert agentspesifikk del. Leverandørsystemet må være i stand til å identifisere pasienten ved hjelp av den unike pasientidentifikatoren som agenten har med seg. Leverandørsystemet må også være i stand til å verifisere mottakerens autorisasjon og kun levere informasjon i henhold til denne autorisasjonen. Informasjonen som overleveres til agenten må krypteres med mottakerens offentlige nøkkel før denne blir brakt videre av agenten. Figur 9 nedenfor viser konfigurasjonen for en installasjon av et journalsystem. En agent for innsamling av pasientinformasjon vil forhandle med agent tjener om lov til å utføre et oppdrag på denne installasjonen. Om agenten får lov til å utføre sitt oppdrag kan være avhengig av om installasjonen støtter denne typen agentoppdrag, om det finnes lokale maskinressurser

tilgjengelig, om mottakerens identitet tilsier at oppdraget er lovlig osv. Dersom en oppnår enighet kan agenten konstrueres og kommunikasjon mellom agent- og leverandørsystem initieres for innsamling av informasjon.



Figur 9 Systemkomponenter for innsamling av pasientinformasjon fra journalsystem

#### *Visualisering av informasjon til sluttbrukeren*

Når agenten har samlet inn pasientinformasjon fra alle tilgjengelige systemer og kryptert denne informasjonen med mottakerens offentlig nøkkel, kan informasjonen overføres til datamaskinen som mottakeren benytter. Agenten kan også ta med seg visualiseringssoftware for spesielle datatyper dersom dette skulle være nødvendig. Med tilgang til funksjoner for dekryptering av innsamlet pasientinformasjon kan agenten sammenstille informasjonen. Dette krever at brukeren har gjort sitt smartkort tilgjengelig og autentiserer seg ovenfor dette med pinkode eller lignende. Avhengig av formålet brukeren oppgir for innsyn i informasjonen kan denne visualiseres for brukeren. Dette forutsetter naturligvis at det finnes regler og systemstøtte for å styre tilgang til informasjon i henhold til det formål som brukeren oppgir.

En kan også tenke seg at sluttbruker og mottaker av informasjonen er pasienten selv. Dette forutsetter naturligvis at pasienten er i stand til å identifisere seg på lik linje med helsearbeidere. En ny dimensjon i sikkerhetsproblematikken er imidlertid om en bruker på Internett får lov til å sende oppdrag inn i et helsenett.

### **5.3.3 Problemstillinger**

Dette arkitekturalternativet er basert på eksperimentell teknologi. Tilgang på kompetanse er derfor et problem i forhold til realisering av dette alternativet. En har heller ikke erfaring fra eksisterende systemer som kan gi grunnlag for reelle vurderinger av fordeler og ulemper med dette alternativet. Arkitekturalternativet er imidlertid spennende fordi det er asynkront og åpner for pasientens innsyn i egen journal.

Innsamling av informasjon i dette arkitekturalternativet forutsetter at en har tilgang til produksjonssystemene som spenner fra små legekontor til store sykehus. Fra et sårbarhets-synspunkt er det sannsynlig at pasientinformasjon kan være utilgjengelig av en rekke årsaker. Denne sannsynligheten er større sammenlignet med en sentralisert løsning. For å redusere konsekvensene av denne utilgjengeligheten kan en velge å legge mer detaljert informasjon inn

i indeksene for å ha muligheten til å oppdage mangler og muliggjøre tilgang til manglende informasjon på annen måte.

Som nevnt ovenfor utnytter dette alternativet tiden fra en vet at en får behov for pasientinformasjon til en faktisk skal benytte informasjonen. Ved øyeblikkelig behov for informasjon vil det nødvendigvis ta noe tid før denne er tilgjengelig for brukeren. Denne responstiden er i dag umulig å anslå og vil generelt være avhengig av hvilke maskinressurser som er tilgjengelige hos informasjonsleverandørene. I dette alternativet har en imidlertid muligheten til å fordele utnyttelsen av disse ressursene til perioder med lav belastning av produksjonssystemene for innsamling av informasjon til planlagte konsultasjoner med pasienten.

Ytelsesmessig vil responstiden ved planlagte konsultasjoner være optimal. Sikkerhetsmessig åpner dette alternativet for at pasienten kan ha tilgang til informasjonen om seg selv, og hvem som har hatt tilgang til denne, uten at en oppretter direkte forbindelser fra Internett til institusjonene som har pasientinformasjonen.

## 6 Diskusjon

Gjennomgangen av eksisterende systemer og prosjekter, samt de alternativene vi har vurdert, viser mange mulige alternativer til arkitektur for nettbasert tilgang til pasientinformasjon. Teknologisk sett kan en derfor konkludere med at et slikt system er teknisk mulig.

Innledningsvis gjorde vi et skille mellom alternativer for nettbasert tilgang til pasientinformasjon som forutsetter fullstendig integrasjon mellom helseinformasjonssystemer og alternativer som kun visualiserer pasientinformasjon. En fullstendig integrasjon kan ha et stort potensiale, men vil realistisk sett ta svært lang tid å utvikle, samt kreve store ressurser på grunn av behovet for "fullstendig" standardisering og integrasjon. På relativt kort sikt er derfor alternativet med kun å løse formidling og visualisering av pasientinformasjon det mest realistiske.

En av de sentrale problemstillingene for systemarkitektur i Elvira prosjektet er valget mellom en sentralisert lagring av pasientinformasjon kontra distribuert lagring av pasientinformasjon som kan overføres. Begge disse hovedalternativene har ulemper. En distribuert løsning vil ha større sannsynlighet for at informasjon er utilgjengelig enn en sentralisert løsning. Årsaker til slik utilgjengelighet kan være alt fra lokal driftsans til kabelbrudd på grunn av snøras. En distribuert løsning sikrer at leverandør/producent av informasjon har fullstendig kontroll, men også ansvar, i forhold til hvilken informasjon som overføres til annet helsepersonell. Leverandøren av informasjon vil også være ansvarlig for at informasjonen er tilgjengelig for andre institusjoner. Et slikt ansvar vil medføre kostnader til investering i maskin og programvare og konsulenttenester for å sikre tilfredstillende grad av sikkerhet. En må likevel anta at en liten helseinstitusjon ikke kan ivareta samme grad av sikkerhet som en stor driftsorganisasjon. Dette er en konsekvens av at små helseinstitusjoner sannsynligvis ikke vil ha tilgang til kompetanse på IKT og sikkerhet på samme måte som en større driftsorganisasjon.

Ved en sentralisert løsning der informasjon kontinuerlig eksporteres fra produksjonssystemene til et datavarehus vil helsearbeiderne som produserer informasjonen måtte stole på at organisasjonen som oppevarer denne informasjonen gjør dette på en sikker og forsvarlig måte. En kan likevel ikke se bort fra at medarbeidere innefor en slik driftsorganisasjon kan få

tilgang til pasientinformasjon uten at slik tilgang logges. Konsekvensene av en illegitim inntrengning utenfra og inn i et sentralisert system som inneholder pasientinformasjon er også vesentlig større enn for en av informasjonsprodusentene, fordi det sentraliserte systemet vil inneholde informasjon om langt flere pasienter og mye mer data om hver pasient. Sannsynligheten for at slikt kan skje er derimot mindre sett i forhold til små helseinstitusjoner. Personer som vil tappe en slik sentralisert løsning for informasjon trenger nødvendigvis bare penetrere et system kontra mange systemer ved en distribuert løsning. Et lignende forhold eksisterer også med hensyn på sårbarhet og tilgjengelighet. Konsekvensen av et driftsavbrudd (eller kabelbrudd) i en sentralisert løsning er vesentlig større enn ved driftstans ved et legekantor. Sannsynligheten for at slikt kan skje er derimot mindre enn for hver enkelt informasjonsleverandør. Et annet moment som potensielt taler mot en sentralisert løsning er om mobile helsearbeidere kan ha tilgang til egen produsert informasjon fra andre institusjoner. Dette er ikke nødvendigvis et stort problem.

Et potensielt problem med en sentralisert løsning er om høyst sensitiv informasjon om seksuelle forhold, provoserte aborter og lignende overhode kan formidles via en sentralisert løsning. Det er meget sannsynlig at pasienter vil motsette seg at slik informasjon gjøres tilgjengelig via en sentralisert løsning fordi pasientene vil anta at driftspersonale potensielt har tilgang til slik informasjon. Det er også mulig at pasientene vil være tilbakeholden med å tillate at langt mindre sensitiv informasjon blir tilgjengelig av samme grunn. Ved etterfølgende komplikasjoner som behandles ved andre helseinstitusjoner vil informasjon om helsetjenester potensielt ikke være tilgjengelig og derfor utgjøre en risiko for pasienten. En distribuert løsning vil kunne tilby større grad av konfidensialitet fordi en ikke har en mellominstans som formidler informasjonen på vegne av en annen. En har muligheten til å kryptere slik informasjon og gi pasienten "nøkkelen" til å "låse opp" innholdet. Dette innebærer imidlertid også en risiko i tilfeller hvor pasienten er bevisstløs eller rett og slett har glemt nøkkelen. En sentralisert løsning truer på denne måten nytten av tilgang, fordi usikkerheten med hensyn til innsyn kan medføre at tilgjengelig informasjon er uinteressant og irrelevant.

Slik den distribuerte løsningen er presentert hittil, er det antatt at søk eller spørringer skal utføres direkte mot de lokale produksjonssystemene som inngår i løsningen. Slike direkte spørringer vil imidlertid øke belastningen på de lokale produksjonssystemene, og det vil være viktig å sørge for at systemene er skalert slik at de kan håndtere den økte belastningen uten at den lokale produksjonen påvirkes negativt (ved uakseptabel responstid o.l.). I tillegg til å skalere for ekstra belastning utenfra kan man se for seg at lokal bruk prioriteres foran eksterne spørringer, at det finnes en øvre grense for hvor store ressurser eksterne spørringer totalt kan legge beslag på til enhver tid, o.l. Et annet alternativ for å beskytte ytelsen i det lokale produksjonssystemet kan være at dataene det skal utføres eksterne spørringer mot, eksporteres til et annet system og spørres mot her. Spørringer utføres med andre ord ikke direkte mot produksjonssystemet, men mot en kopi. Det vil selvsagt være av stor viktighet at denne kopien oppfriskes tilstrekkelig ofte. Problemstillingene med oppfrisking av data vil være de samme som ved en sentralisert løsning (datavarehus), og en slik løsning vil selvsagt kreve ytterligere ressurser både maskin, menneske og programvaremessig.

Løsninger hvor spørringer gjøres mot en kopi (replika) av data fra et produksjonssystem i motsetning til direkte mot produksjonssystemet, er fordelaktige ved at integriteten i produksjonssystemene da kan sikres bedre. Ingen behøver da å ha direkte tilgang til det lokale produksjonssystemet utenfra, og risikoen for at produksjonsdata endres av utenforstående blir redusert. Videre kan oppdatering av kopien (som eksterne spørringer skal utføres mot) kun initieres fra det lokale produksjonssystemet og ut – ikke andre veien. Dette

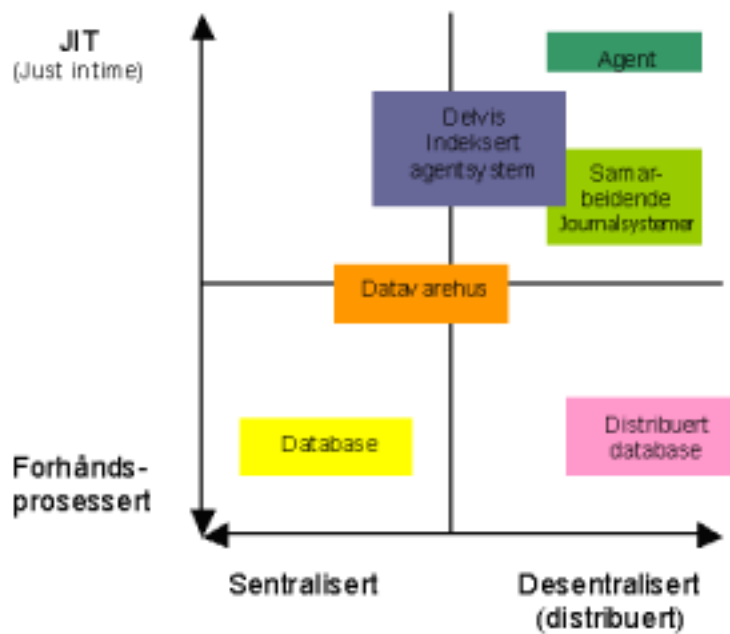
reduserer ytterligere risikoen for at data i de lokale produksjonssystemene kan bli endret av uvedkommende brukere som har ekstern tilgang til dataene. I en sentralisert løsning (datavarehus) kan man tenke seg at all oppdatering av det sentrale lagret initieres slik fra hvert lokalt produksjonssystem.

Valg av systemarkitektur bør være basert på hvilken funksjonalitet, informasjonsomfang og sikkerhetsnivå som er mulig å oppnå. Dersom et pasientinformasjonssystem bare gir tilgang til deler av all relevant pasientinformasjon og/eller for mye irrelevant informasjonen, vil nytten av et slikt system reduseres og på sikt kunne medføre at systemet ikke benyttes. Om et slikt system vil være nyttig avhenger derfor av om systemet drukner brukeren med irrelevant og/eller sultefører brukeren på relevant informasjon. Mye arbeid må legges ned for å sikre at disse problemstillingene ivaretas på en korrekt måte.

Dersom en sammenligner BoJ løsningens behov for indeksinformasjon kontra agentløsningens indeksbehov ligger det et element av sentraliserings-/desentraliseringsproblematikken i behovet for indeksinformasjon. BoJ's indeksinformasjonsbehov har en konfidensialitetsside som ikke er like sterk i agentløsningen. Dersom en sentralisert løsning er problematisk juridisk, kan dette få konsekvenser for løsninger som forutsetter indeksering av pasientinformasjon. Ved valg av systemarkitektur bør en derfor vurdere om slik informasjon medfører at løsningen kommer i strid med gjeldende lovverk.

## **7 Konklusjoner og anbefalinger**

Valget mellom en sentralisert kontra en desentralisert systemarkitektur er vanskelig og har mange parametre. En sentralisert løsning har ulemper i forhold til konfidensialitet som truer nytten av hele systemet. Denne ulempen ligger i at et sentralisert system vil fungere som et mellomledd mellom produsent av informasjon og den som har behov for denne. Om en sentralisert løsning er i henhold til lovverket er også usikkert (jfr. rapport om juridiske aspekter). Problemet med en distribuert løsninger er økt kompleksitet og derved økt sårbarhet og sikkerhetsproblemer. En sentralisert løsning vil være langt mer sikker. Tilgang til pasientdata kan enkelt reguleres, f.eks. ved autorisasjonsnivåer. Det er også langt enklere å etablere en speilserver (feiltoleranse) hvor alle data vil være tilgjengelige. Det distribuerte alternativet er derfor mer usikker fordi en i dag ikke vet nok om risiko og trusler ved en desentralisert løsning. Vi anbefaler derfor at en jobber videre med en risikoanalyse for en eller flere desentraliserte systemarkitekturer.



**Figur 10** Mulig modeller for Elvira realisering

Mulige arkitekturer til Elvira kan i hovedsak realiseres/skisseres langs to akser: JIT vs. forhåndsprosessert og sentralisert vs. desentralisert (se Figur 10). Langs disse aksene kan vi plassere database, distribuert database, datavarehus, samarbeidende journalsystemer og agenter.

Av disse modellene representerer (sentralisert) database og agent-basert system to ytterligheter. I den første vil alle data være lagret i en sentral database mens agenttilnærmingen innebærer at data først blir tilgjengelig etter et eksplisitt søk i databaser lokalisert på fremmede datamaskiner. Et alternativ til en ren agenttilnærming vil være å delvis indeksere pasientopplysningene, f.eks. at en på forhånd er kjent i hvilke kommuner det er lagret data om pasienten. Et annet alternativ er samarbeidende databaser, hvor ideen er at det for hver enkelt pasient legges eksplisitte linker i journalsystemene til andre journalsystemer i de tilfeller hvor data lagres flere steder.

Visualisering av pasientinformasjon er et annet område en bør jobbe videre med som beskrevet i seksjon 3.3. En bør starte arbeidet med å realisere felleskomponentene for sikkerhet og PKI løsning for helsenettet.

## 8 Referanser

- [1] <http://www.sll.se/ds/itds/boj.htm>
- [2] <http://www.baptisthealth.net>
- [3] <http://www.caregroup.org>
- [4] <http://informatics.caregroup.org/people/jhalamka/careweb.htm>
- [5] D. G. Katehakis, S. Sfakianakis, M. Tsiknakis, S. C. Orphanoudakis, An Infrastructure for Integrated Electronic Health Record Services: The Role of XML, MEDNET 2000, 5th World Congress on the Internet in Medicine, Brussels, Belgium, November 23-26, 2000, pp. 189-190.
- [6] D. Katehakis, S. Kostomanolakis, M. Tsiknakis, S. Orphanoudakis: "Evaluating Alternative Approaches for Integrating Clinical Information Systems: Messaging vs. Federating". Accepted for presentation, and to be included in the programme for the conference TEHRE 2000, Your Eye on the Future: What EHR and e-Health will provide, London, UK, November 12-15, 2000.
- [7] G. Potamias, M. Tsiknakis, D. G. Katehakis, E. Karabela, V. Moustakis, and S. C. Orphanoudakis, Role-Based Access to Patients Clinical Data: The InterCare Approach in the Region of Crete, MIE 2000, Hannover, Germany, August 27- September 1, 2000, pp. 1074-1079.
- [8] D. Katehakis, P. Lelis, E. Karabela, M. Tsiknakis, S. Orphanoudakis: An Environment for the Creation of an Integrated Electronic Health Record in HYGEIAnet, the Regional Health Telematics Network of Crete. TEPR 2000, Your Connection to Electronic Healthcare, San Francisco, CA, May 9-11, 2000, Vol. 1, pp. 89-98.
- [9] S. Ray. User Interfaces for Computer-based patient records.
- [10] C. Plaisant, B. Milash, A. Rose, S. Widoff, B. Shneiderman. (1996). Life Lines: Visualizing personal histories. Proc. of the ACM CHI '96 Conference (Vancouver, BC, Canada, April 13-18, 1996), pp. 221-227.
- [11] Ö. Tamer, "Principles of distributed multidatabase systems", second edition
- [12] <http://gunnarnt.cs.uit.no/vise/>
- [13] G. Hartvigsen, S. Johansen, A. Helme, R. A. Widding, J.G. Bellika & W. Cao: The Virtual Secretary Architecture for Secure Software Agents. In Proceedings of the First International Conference on the Practical Application of Intelligent Agents and Multi-Agent Technology, PAAM 96
- [14] J.G. Bellika, G. Hartvigsen. Integration of Electronic Patient Records – The DiPato Approach. In proceedings of Conference on the Creation of a European Electronic Health Record (TEHRE-2000). (Nov 12–15, London).