



Felles Registerløsning versjon 1.1

Integrasjonsgrensesnitt mot FRESH for fagsystemer i spesialisthelsetjenesten

Forfattere:

Arbeidsgruppe 4 i N-IKT HER pilotprosjekt
Jarle Vikshåland og Stig Karlsen på vegne av Norsk Helsenett

Dato: 19.06.2009

Versjon: 1.8

Status: Godkjent

Godkjent av Norsk Helsenett v/Pål Arve Sollie

Godkjent dato:19.06.2009

Versjonshistorikk:

Versjon	Dato	Forfatter	Kommentar
0.1-0.9	12.12.2007	Arbeidsgruppe 4 v/Jon Gupta og Jarle Vikshåland	Disposisjon, til diskusjon, til høring i prosjektgruppe og pilothelseforetak, oppdateringer etter høring, klar til utsendelse til leverandører
1.0-1.2	09.05.2008	Arbeidsgruppe 4 v/Jarle Vikshåland	Oppdatert etter tilbakemeldinger fra leverandører og intern gjennomgang. XSD for kommunikasjon med fagsystemene er definert, metoder presisert og enkelte metoder er tatt ut. Korrigert feil og uklarheter. Utdypet fagsystemsøifikke tilleggsplysninger etter diskusjon med Tieto Enator.
1.3	27.06.2008	Arbeidsgruppe 4 v/Jarle Vikshåland	Oppdatert i henhold til implementert Webservice i FRESH og siste feilrettinger, og avstemt med XSD versjon 1.5
1.4	26.08.2008	Arbeidsgruppe 4 v/Jarle Vikshåland	Oppdatert etter endringsbehov fremkommet i møte med UUS, Rikshospitalet og Siemens DocuLive. Organisasjonsbetegnelse er lagt inn i Organization elementet og input parametre til metoden inkluderer nå fagsystemkode for å muliggjøre innholdsbasert routing på integrasjonsplattformen. Samkjørt med XSD versjon 1.7
1.5	29.09.2008	Arbeidsgruppe 4 i N-IKT HER Pilotprosjekt v/Jarle Vikshåland	Ryddet. Siste versjon
1.6	20.03.2009	Acando v/Stig Karlsen	Oppdatert xml-kode i forhold til endringer i FR 1.0. XSD versjon 2.0
1.7	15.06.2009	Acando v/Jarle Vikshåland	Oppdatert figurer ihenhold til Felles Registerløsning versjon 1.1, endret navn på dokumentet og oppdatert beskrivelser i henhold til ny arkitektur med en felles løsning for RESH, Rekvirentregister og Adresseregisteret.
1.8	19.06.2006	Pål Arve Sollie	Endret dokumentnavn. Godkjent for produksjon

Referanser:

Nr	Dokument	Kommentar
[1]	http://www.kith.no/templates/kith_WebPage____1136.aspx	KITH Nettside: Identifikasjon av organisatoriske enheter innen helseinstitusjoner
[2]	http://www.kith.no/upload/1818/Retningslinjer_for_registereing_av_enheter_i_RESH_(2007_06_29).doc	KITH dokumentet: Retningslinjer for registrering i RESH
[3]	Dynamisk GUI arkiv	Dokumentsett som spesifiserer den dynamiske GUIen.
[4]	2008-09-29 N-IKT Spesifikasjon FRESH Utgående for meldingstjener og integrasjonsplattform - ver 0.8	Spesifiserer utgående integrasjonsgrensesnitt fra FRESH, primært for bruk av meldingsserver og integrasjonsplattform.

Innhold

1	Innledning	5
1.1	Formål	5
1.2	Forfattere	5
2	Integrasjon mellom FRESH og fagsystemene	5
2.1	Bakgrunn	5
2.2	Overordnet applikasjonsarkitektur	6
2.3	WebService Operasjoner	8
2.3.1	Overordnet beskrivelse	8
2.3.2	Prinsipper for Web Service metodene	8
2.3.3	Prinsippet om registrering av informasjon en og bare en gang	9
2.3.4	Eksterne føringer for arkitekturen	10
2.4	FRESH brukstilfeller	10
2.4.1	Brukstilfelle - Koble til fagsystem kommunikasjonspart	11
2.4.2	Brukstilfelle - Distribuer endringer	11
2.5	Detaljert beskrivelse av Web Service metoder i fagsystemene	12
2.5.1	Finn kommunikasjonspart - FindCommunicationParty	12
2.5.2	Opprett kommunikasjonspart - CreateCommunicationParty	13
2.5.3	Behandle kommunikasjonspart endringer - HandleCommunicationPartyUpdate ..	14
2.5.4	XSD Versjonshåndtering - GetXSDVersion	15
2.5.5	Fagsystemspsesifikke tilleggsopplysninger	16
2.6	Feilhåndtering og feilmeldinger	17
2.7	Sikkerhet	17
2.7.1	Sikkerhetsmodell i pilotperioden og fremover	17
2.7.2	Autentisering, pålogging og avlogging	17
3	XSD, XML og WDSLdokumentasjon	18
3.1	Hovedelementer i XSDen	18
3.1.1	CreateCommunicationPartyInputMsg og CommunicationParty – Kommunikasjonspart elementet	18
3.1.2	CommunicationPartyUpdate og UpdateCommunicationPartyInputMsg– kommunikasjonspart oppdatering	20
3.1.3	SearchForCommunicationPartyInputMsg – struktur for å søke i fagsystem	21
3.1.4	CommunicationPartySearchResultArray – Liste over rekvirenter funnet i fagsystemet	24
3.1.5	ParentType – definisjon av Parent delen av en kommunikasjonspart	25
3.1.6	ChildType – definisjon av Child delen av en kommunikasjonspart	25
3.1.7	OrganizationType definisjon	26
3.1.8	PersonType definisjon	27

3.1.9	ServiceType definisjon.....	29
3.1.10	GeographicAddressType definisjon.....	29
3.1.11	ElectronicAddressType definisjon.....	31
3.1.12	StatusMessage and StatusMessageType definisjon	32
3.1.13	ErrorMessageType	33
3.2	Detaljert dokumentasjon	34
3.2.1	XSD	34
3.3	Kardinalitet og sekvens.....	34
3.4	XSD schema for kommunikasjonspartinformasjon - CommuncationParty	34
3.5	Eksempel XML for Communication party	34
4	Begreper	36

Figurer

FIGUR 1	OVERORDNET ARKITEKTUR.....	6
FIGUR 2	OVERORDNET INTEGRASJONSKONSEPT MOT FAGSYSTEMENE OG ANSVARFORHOLD	7
FIGUR 3	TJENESTEGRENSESNIET MOT FAGSYSTEM MED FRESH PROSESSEN PÅ VENSTRESIDEN OG FAGSYSTEMTJENESTEN PÅ HØYRESIDEN	11
FIGUR 4	SKJERMBILDE EKSEMPEL FAGSYSTEMSPESIFIKKE TILLEGG	16
FIGUR 5	CREATECOMMUNICATIONPARTYINPUTMSG DEFINISJON	18
FIGUR 6	COMMUNICATIONPARTY DEFINISJON.....	19
FIGUR 7	UPDATECOMMUNICATIONPARTYINPUTMSG DEFINISJON.....	20
FIGUR 8	COMMUNICATIONPARTYUPDATE DEFINISJON.....	20
FIGUR 9	SEARCHFORCOMMUNICATIONPARTYINPUTMSG DEFINISJON	21
FIGUR 10	PERSONSEARCHCRITERIA TYPE DEFINISJON.....	22
FIGUR 11	ORGANIZATIONSEARCHCRITERIA TYPE DEFINISJON	23
FIGUR 13	COMMUNICATIONPARTYSEARCHRESULTARRAY DEFINISJON	24
FIGUR 14	COMMUNICATIONPARTYSEARCHTYPE DEFINISJON.....	24
FIGUR 15	PARENTTYPE DEFINISJON.....	25
FIGUR 16	CHILDTYPE DEFINISJON	26
FIGUR 17	ORGANIZATIONTYPE DEFINISJON.....	26
FIGUR 18	PERSONTYPE DEFINISJON	28
FIGUR 19	GEOGRAPHICADRESSTYPE DEFINISJON	30
FIGUR 20	ELECTRONICADRESSTYPE DEFINISJON.....	31
FIGUR 21	STATUSMESSAGE OG STATUSMESSAGETYPE DEFINISJON	32
FIGUR 22	ERRORMESSAGETYPE DEFINISJON	33

1 Innledning

1.1 Formål

Formålet med dette dokumentet er å beskrive et integrasjonsgrensesnitt, basert på Web Service metoder, som må implementeres i fagsystemene for å støtte distribusjon og vedlikehold av kommunikasjonspartinformasjon til det enkelte fagsystem.

Dette dokumentet beskriver første versjon av løsningen, og gjelder for systemer som har eller skal ha egen rekvirentdatabase. FRESH vil gjennom en Web Service spesifisert i dette dokumentet gis mulighet til å lese i og skrive til fagsystemene.

1.2 Forfattere

Dokumentet er utarbeidet av:

Jarle Vikshåland	: Helse Vest IKT / Norsk Helsenett
Jon Gupta	: Helse Sør-Øst RHF
Camilla Wadseth	: Helse Midt-Norge IT
Renard Nilsen	: Helse Nord IKT
Øystein Berg Sletteng	: Helse Nord IKT
Lars-Gunnar Hartveit	: Helse Vest IKT
Stig Karlsen	: Norsk Helsenett

2 Integrasjon mellom FRESH og fagsystemene

2.1 Bakgrunn

Det er et mål at kommunikasjonspartinformasjon skal vedlikeholdes ett sted av den som eier informasjonen, og distribueres fra en felles løsning til alle andre systemer som har bruk for kommunikasjonspartinformasjonen. En slik løsning vil sikre at alle systemene til enhver tid har oppdatert kommunikasjonspartinformasjon, deriblant fagsystemene i det enkelte helseforetak.

Dette dokumentet beskriver et Web Service integrasjonsgrensesnitt som må implementeres i fagsystemene for å støtte søk etter, opprettelse og vedlikehold av kommunikasjonsparter i fagsystemet

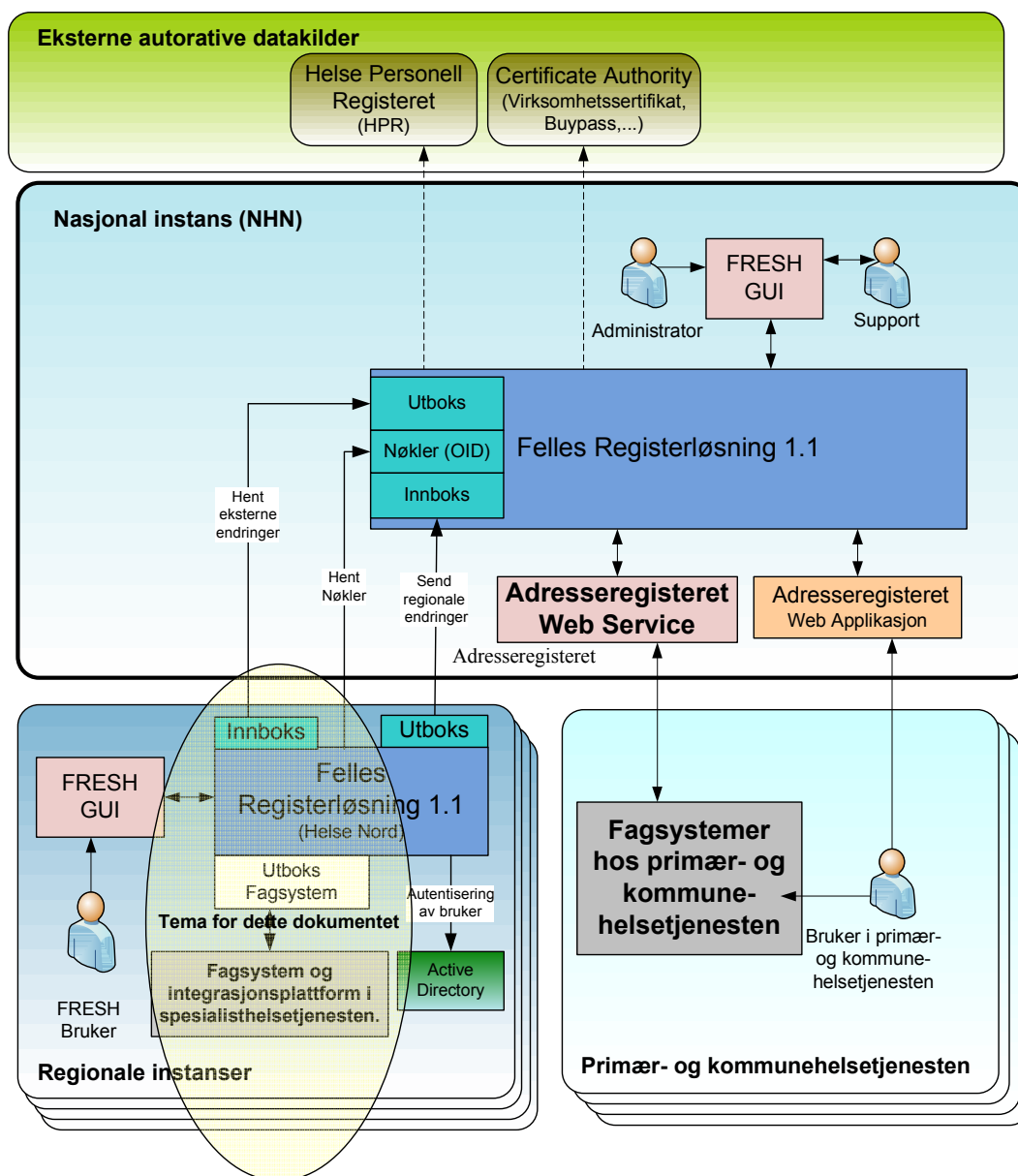
Web Service grensesnittet skal implementeres av systemer som har eller skal ha egen rekvirentdatabase. FRESH vil gjennom en Web Service spesifisert i dette dokumentet gis mulighet til å lese i og skrive til fagsystemene.

Det er tenkt at en fremtidig versjon av FRESH vil inkludere støtte for oppslag på kommunikasjonspartinformasjon i FRESH fra fagsystemene, slik at fagsystemene kan bruke FRESH som sitt kommunikasjonspart register.

Dokumentet består av en overordnet beskrivelse av problemstillingene, føringer for integrasjonsløsningen og en detaljert beskrivelse av tjenesten og tilhørende metoder.

2.2 Overordnet applikasjonsarkitektur

Dette dokumentet beskriver hvordan man skal sikre at kommunikasjonspartinformasjon (kalles også rekvirenter, adresseregister mm) i det enkelte fagsystem til enhver tid er oppdatert med nasjonalt oppdatert og korrekt informasjon. Dette gjøres ved at fagsystemene implementerer et Webservicegrensesnitt som brukes av FRESH for å opprette og oppdatere kommunikasjonspart informasjon.



Figur 1 Overordnet arkitektur

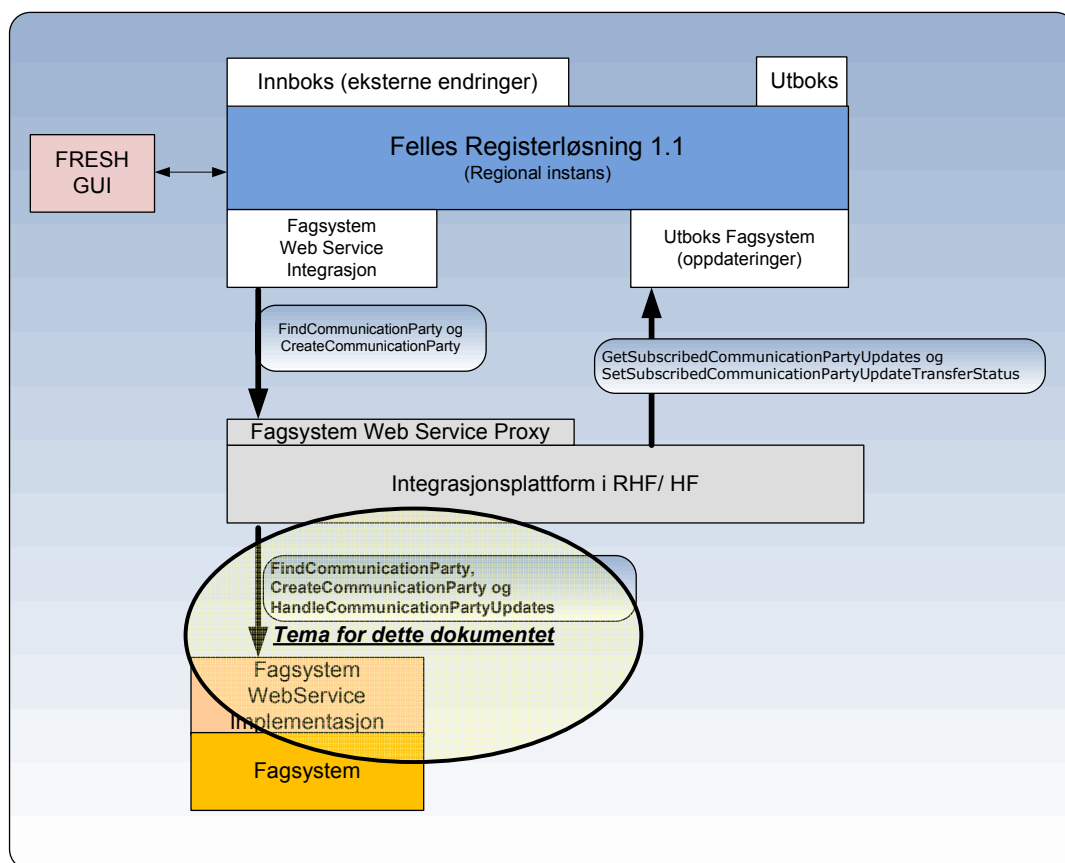
FRESH vil vedlikeholde en oversikt over hvilke rekvirenter som er registrert i det enkelte fagsystem, vha. en kobling mellom HER-id og lokal rekvirentkode, og

sikrer at oppdateringer av kommunikasjonspartinformasjon kan propageres automatisk når det skjer endringer i kommunikasjonspartinformasjon som er relevant for det enkelte fagsystem.

Det er etablert utvidelser i FRESH som ivaretar kravene om sentralisert vedlikehold av kommunikasjonspartopplysninger:

1. Brukergrensesnitt som muliggjør registrering og oppdatering av kommunikasjonspartopplysninger
2. Brukergrensesnitt som muliggjør kobling og distribusjon av kommunikasjonspartopplysninger fra FRESH til påkoblede fagsystem. Denne koblingen kalles også et abonnement.
3. Støtte for en prosess for distribusjon og oppfølging av endringer til enkelte fagsystem, samt synkronisering med A-HER.

Fagsystemene skal tilby en Web Service med tilhørende metoder som understøtter brukstilfellene og de krav til informasjonsflyt mellom løsningene som er beskrevet i dette dokumentet. FRESH spiller i dette tilfelle rollen som konsument av tjenesten der informasjonsbehovet er avtalt gjennom kontrakten som etableres mellom FRESH og fagsystemene.



Figur 2 Overordnet integrasjonskonsept mot fagsystemene og ansvarsforhold

Figuren ovenfor viser på et konseptuelt nivå integrasjonsløsningen mellom FRESH og fagsystemet. Fagsystemene har ansvar for realisering av Web Service grensesnittet somerbeskrevet i dette dokumentet.

2.3 Webservice Operasjoner

2.3.1 Overordnet beskrivelse

Web Service metodene som skal støttes er

- FindCommunicationParty – søkemetode som returnerer en eller flere kommunikasjonsparter som tilfredstiller søket, deriblant en unik identifikator (primærnøkkel eller tilsvarende) for hver kommunikasjonspart i fagsystemet.
- CreateCommunicationParty – metode for å opprette en ny kommunikasjonspart i fagsystemet. Fagsystemet returnerer en unik identifikator (primærnøkkel eller tilsvarende) for kommunikasjonsparten i fagsystemet.
- HandleCommunicationPartyUpdates – metode for å oppdatere eksisterende kommunikasjonsparter i fagsystemet. Baserer oppdateringen på den unike identifikatoren for kommunikasjonsparten i fagsystemet.
- GetXSDVersion – returnerer versjonnummeret angitt i XSD filen som inneholder datakontrakten, brukes for fremtidig versjonshåndtering og feilsøk.

2.3.2 Prinsipper for Web Service metodene

Vi ønsker å understreke at vi ikke ønsker å komme i en situasjon der FRESH må kjenne til interne "oppgaver" i fagsystemet for å kunne benytte Web Service som tilbys, eksemplifisert ved brukstilfellet opprette ny kommunikasjonspart i et fagsystem:

1. FRESH ønsker å opprette en kommunikasjonspart i fagsystemet.
2. FRESH sender den informasjonen som kreves av kontrakten mellom den og fagsystemet.
3. Informasjonen behandles av fagsystemet. iht. til den interne prosessen som fagsystemet har definert for opprettelsen av rekvirenter.
4. FRESH får tilbake et svar som definert i kontrakten mellom FRESH og fagsystemet.

Web Service metoden bør forholde seg til følgende prinsipper:

1. FRESH skal IKKE trenge å vite noe om prosessen fagsystemet implementerer for å få opprettet en kommunikasjonspart
2. FRESH skal IKKE trenge å vite noe om rekkefølgen man må utfører Web Service metodene i for at integrasjonen med fagsystemet skal virke. Det vil si at tjenesten skal ikke utformes som et API, men skjule den interne prosessen som må gjennomføres for at metodene skal være fullstendig løskoblet. Hvorvidt tjenesten er tilstandsløs eller ikke er avhengig av hvordan dette implementeres internt av leverandøren

-
3. FRESH skal IKKE trenge å vite noe om datastrukturen fagsystemet bruker for å få opprettet en kommunikasjonspart
 4. FRESH skal IKKE trenge å vite noe om interne funksjoner fagsystemet bruker for å få opprettet eller endret en kommunikasjonspart.
 5. Alle Web Service metodene er av Request-Response typen. FRESH vil vente på svar fra fagsystemet, der et svar vil enten være et gyldig eller en feilmelding fra fagsystemet. Dette betyr at fagsystemet må kunne gjennomføre nødvendige interne operasjoner og returnere et svar til FRESH innen rimelig tid.

Disse prinsippene gjelder for alle Web Service metodene som skal implementeres.

2.3.3 Prinsippet om registrering av informasjon en og bare en gang

Det er et vedtatt prinsipp at informasjon ikke skal registreres to ganger. Dersom informasjon finnes på et høyere nivå skal den arves på underliggende nivåer og ikke registreres en gang til. En del informasjon kan registreres både på Parent og på Child i CommunicationParty, men vi tilstreber å følge retningslinjene fra KITH for registrering av informasjon i RESH [2], her heter det i kapittel 5.1.3

Man skal aldri gjenta registrering av egenskap dersom denne egenskap har samme verdi som for en overliggende enhet. Det er nok å registrere for eksempel en adresse kun en gang, til den mest overordnede enhet på denne fysiske plassen. Vanligvis vil alle underliggende enheter til slike enheter ligge på samme sted som den overordnede enhet. Det kan likevel være tilfeller av at en underordnet (eierskapsmessig) enhet har en helt annen adresse enn overordnet enhet. En slik avvikende adresse vil kunne registreres eksplisitt til denne underliggende enheten og dermed overstyre den overordnede enhetens adresse.

Slike egenskaper skal registreres på høyest mulige nivå, og disse egenskapene gjelder også for underliggende enheter. De arves.

Dette prinsipp gjelder for egenskapene:

- Organisasjonsnummer
- Kommunenummer
- HER-ID
- Ansvarlig person
- Eierskap
- Adresse (alle typer)

For CommunicationParty definisjonen betyr dette at Child arver Parent adresseinformasjon, og at Child kan overstyre den arvede informasjonen.

I tilfellet der Child er en organisasjon kan et komplett sett med adresseinformasjon oppgis, mens i tilfellet det Child er en person kan bare de elektroniske adressene overstyres.

Adresseregisteret HER bryter med dette prinsippet og oppgir adresse for den enkelte kommunikasjonspart. Hvordan det enkelte fagsystem representerer informasjonen internt er opp til den enkelte leverandør.

For mer informasjon se KITHs nettside [1] og dokumentet Retningslinjer for registrering i RESH [2].

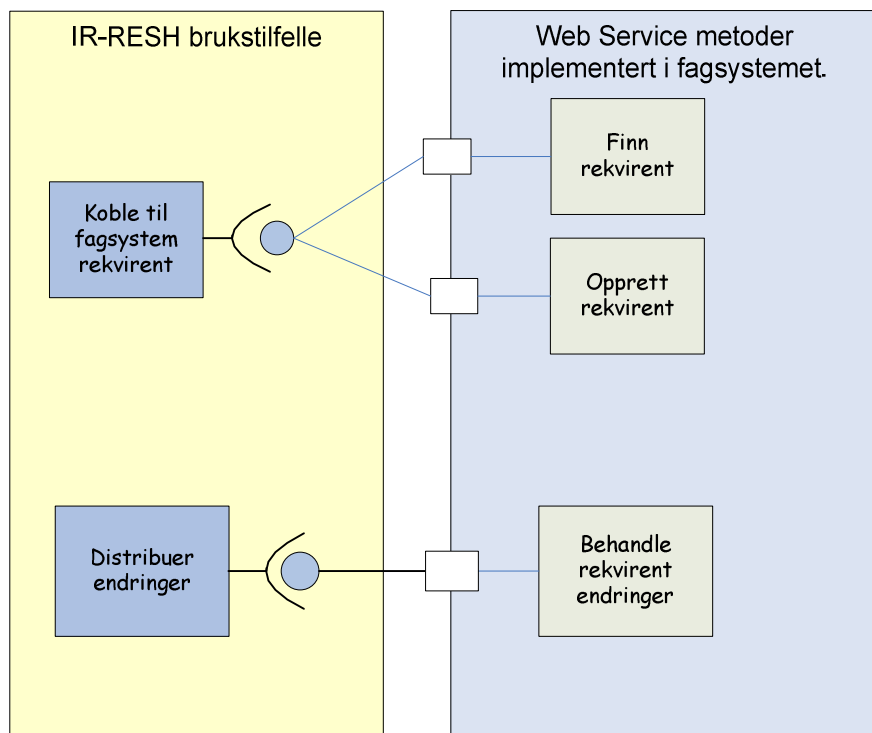
2.3.4 Eksterne føringer for arkitekturen

1. Adresseregisteret HER legger føringer på datamodell og distribusjonsarkitektur.
 - Datamodellen i A-HER skulle ikke endres og legger føringer for datakontrakten i Web Servicene
 - Distribusjon mellom regioner og informasjon fra primær- og kommunehelsetjenesten går gjennom A-HER
 - Primær- og kommunehelsetjenesten registrerer sine kommunikasjonsparter i A-HER
 - Primær- og kommunehelsetjenesten bruker A-HER for å hente ut informasjon om kommunikasjonsparter, og rek
2. Helseregionene og Helseforetakene har egne integrasjonsplattformer og – strategier.
 - Realiseringen av integrasjon mellom FRESH og fagsystemene vil derfor være ulik avhengig av hvilken region man er i.
 - Prosjektet legger opp til en løsning som vil kunne fungerer for alle.
3. Sikkerhetsproblematikk
 - Manglende plattform for sikker samhandling mellom regioner medfører at hver region skal ha en lokal kopi av FRESH.
 - Synkronisering går via A-HER også for data som har sin opprinnelse I FRESH
4. Mange fagsystemer, blant annet DIPS, DocuLive med flere, har egne rekvirentabeller og innholdet i fagsystemenes rekvirenttabellene skal kunne ivaretas
 - Prosjektet har gjort et valg om å støtte disse i første omgang.
 - FRESH brukergrensesnittet gir brukerstyrt kobling av kommunikasjonspart mot fagsystem, kalt abonnement
 - Fagsystemet mottar automatisk alle endringer det abonnerer på
 - Dette har lagt føring for integrasjonsløsningen.
5. Fagsystem som ikke har eget rekvirentregister, men gjør oppslag mot andre databaser eller systemer støttes eventuelt I fremtid versjon

2.4 FRESH brukstilfeller

N-IKT HER pilotprosjektet har identifisert brukstilfeller i FRESH som forutsetter samhandling med helseforetakenes fagsystem som en konsekvens av kravet om sentralisert vedlikehold av kommunikasjonspartopplysninger.

Brukstilfellene omfatter indirekte, gjennom bruk av Web Service metodene, både lese og skrive metoder mot fagsystemets rekvirenttabeller – se kapittel 3 for flere detaljer om XML for datautveksling.



Figur 3 Tjenestegrensesnittet mot fagsystem med FRESH prosessen på venstresiden og fagsystemtjenesten på høyresiden

2.4.1 Brukstilfelle - Koble til fagsystem kommunikasjonspart

Brukstilfellet initieres når en bruker med godkjenningsrettigheter søker etter tilsvarende fagsystemrekvirent med utgangspunkt i FRESH-rekvirenten som er valgt . For mer om rollebeskrivelser se " 2007-11-12 N-IKT Spesifikasjon brukeregrensesnitt felles rekvirentregister ver 1.0.doc".

Brukstilfellet er under arbeid.

Brukstilfellet forutsetter at man kan søke mot fagsystemet og opprette fagsystem rekvirenter.

2.4.2 Brukstilfelle - Distribuer endringer

Brukstilfellet håndterer distribusjon av godkjente endringer til fagsystemet. Distribusjonen til det enkelte fagsystem gjøres basert på listen over koblinger mellom rekvirenter i FRESH og tilsvarende rekvirenter i det enkelte fagsystem.

Følgende er endringer i kommunikasjonspartinformasjon

- Oppdateringer av helseforetakets kommunikasjonspartinformasjon gjennom bruk av FRESH brukergrensesnitt
- Oppdateringer på eksterne rekvirenter som distribueres fra Adresseregisteret HER

2.4.2.1 Implementering av distribusjon av endringer til fagsystemene

FRESH tilbyr et Webservice grensesnitt der to av operasjonene er implementert for å støtte distribusjon av endringer brukstilfellet *Distribuer endringer* sammen med *HandleCommunicationPartyUpdates*. Disse metodene er beskrevet i detalj i [4]. Fagsystemleverandøren trenger ikke kjenne til eller forholde seg til disse operasjonene, de vil normalt brukes av integrasjonsserveren. Metodene beskrives allikevel her for å vise helheten.

- *GetSubscribedCommunicationPartyUpdates* - operasjon som leverer den første av en eller flere nye endringer i kommunikasjonspartinformasjon for et spesifisert fagsystem. Endringene hentes en for en i rekkefølge basert på et sekvensnummer som angir rekkefølgen endringene er gjort i. Operasjonen og tilhørende datakontrakter er beskrevet i kapittel 3.3 *GetSubscribedCommunicationPartyUpdates* i [4].
- *SetSubscribedCommunicationPartyUpdateTransferStatus* - operasjon for å oppdatere status i forbindelse med overføring av kommunikasjonspartinformasjon til det enkelte fagsystem. hOperasjonen og tilhørende datakontrakter er beskrevet i kapittel 3.4 *SetSubscribedCommunicationPartyUpdateTransferStatus* i [4]

Brukstilfellet *Distribuer endringer* implementeres på integrasjonsserveren og er beskrevet i mer detalj i *2008-06-15 N-IKT Spesifikasjon FRESH Utgående integrasjonsgrensesnitt for meldingstjener - ver 0.7*.

2.5 Detaljert beskrivelse av Web Service metoder i fagsystemene

Delkapitlene som følger beskriver det prosjektet mener fagsystemleverandøren bør vurdere ivare tatt i sin Web Service leveranse. Det er laget en sjekklister over hva HER Pilotprosjektet mener bør vurderes håndtert i forbindelse med den enkelte Web Service metoden. I tillegg er dataelementene / datakontrakten knyttet til den enkelte metode beskrevet.

2.5.1 Finn kommunikasjonspart - *FindCommunicationParty*

"Finn kommunikasjonspart"-metoden vil bli benyttet i forbindelse med brukstilfellet "Koble til fagsystemrekvirent".

Metoden tar imot et sett med parametre som kommer fra bruker-konteksten i FRESH klient. Brukeren velger en kommunikasjonspart som brukes i søket og får returnert en liste over kommunikasjonsparter i datasystemet som tilfredsstiller søket. Listen er beskrevet i detalj senere i dokumentet, og i XSDen som beskriver datakontraktene.

Input:

findCommunicationPartyIn

- innpakking av *SearchForCommunicationPartyInputMsg* som er beskrevet i kapittel 3.1.3 *SearchForCommunicationPartyInputMsg – struktur for å søke i fagsystem*.

Output:

findCommunicationPartyOut

- innpakking av *CommunicationPartySearchResultArray* beskrevet i kapittel 3.1.4 *CommunicationPartySearchResultArray – Liste over rekvirenter funnet i fagsystemet*.

2.5.1.1 Sjekkliste for fagsystem implementasjon

Sjekklisten er et eksempel på funksjonalitet vi antar metoden bør ivareta internt i fagsystemet. Dette skjules for FRESH som forholder seg til et enkelt metodekall mot fagsystemet.

1. Pålogging av FRESH bruker til fagsystemet – fagsystemleverandøren bør ivareta krav til autentisering, autorisasjon og sporing i Web Service metoden
2. Kontroll av rekvirentopplysninger – fagsystemleverandøren bør kontrollere rekvirentopplysningene som sendes inn ved bruk av metoden
3. Oppslag organisasjonsenheter – Avhengig av datamodellen som ligger til grunn for fagsystemet, bør fagsystemleverandøren vurdere oppslag i interne enhetstabeller
4. Oppslag personer - Avhengig av datamodellen som ligger til grunn for fagsystemet, bør fagsystemleverandøren vurdere oppslag i interne persontabeller
5. Oppslag rekvirenter - Avhengig av datamodellen som ligger til grunn for fagsystemet, bør fagsystemleverandøren vurdere oppslag i interne rekvirenttabeller
6. Sammenstill og returner resultatsett – fagsystemleverandøren må sammenstille og returnere ønsket resultatsett til FRESH, se utkast til datakontrakt
7. Avlogging – sesjonen som er etablert bør kobles ned når metoden er gjennomført

2.5.2 Opprett kommunikasjonspart - CreateCommunicationParty

Opprett kommunikasjonspart metoden benyttes kun i forbindelse med brukstilfellet "Koble til fagsystem kommunikasjonspart" når bruker har funnet rekvirenten i FRESH, men ikke i valgt fagsystem.

Metoden tar imot en utfylt kommunikasjonspart og oppretter rekvirenten i fagsystemet. Fagsystemet må fore nødvendige kall og transformasjoner for å få opprettet rekvirenten i fagsystemet basert på mottatte verdier og returnerer en status, primærnøkkelen til opprettet kommunikasjonspart eller en eventuell feilmelding dersom en feil har inntruffet. Eventuelle fagsystemspesifikke tilleggsopplysninger håndteres i den fagsystemspesifikke utvidelsen av kommunikasjonspartdefinisjonen – se 2.5.5 *Fagsystemspesifikke tilleggsopplysninger*.

Metoden skal fra FRESH siden være et synkront kall for uavhengig om den underliggende implementasjonen i fagsystemet er synkron eller asynkron.

Input:

createCommunicationPartyIn

- Innpakking av CreateCommunicationPartyInputMsg elementet beskrevet i 3.1.1 *CreateCommunicationPartyInputMsg og CommunicationParty* – Kommunikasjonspart elementet.

Output:

createCommunicationPartyOut

- Innpakking av StatusMessage elementet beskrevet i 3.1.12 *StatusMessage and StatusMessageType definisjon*.

-
- Status 0 - OK
 - Fagsystemets interne primærnøkkel for rekvirenten returneres i returnvalue feltet
 - Trengs for opprettelse av rad i mapping tabell
 - Status -1
 - Feilmelding i ErrorMessage elementet
 - Feilsituasjoner kan være forsøk på å opprette allerede eksisterende rekvirenter, feil i input data eller interne feil i fagsystemet.

2.5.3 Behandle kommunikasjonspart endringer -

HandleCommunicationPartyUpdate

Metoden for behandling av endringer tar imot et parametersett som beskriver rekvirenten og foretar nødvendige kall og datatransformasjoner for å få endre kommunikasjonspartinformasjonen på den eksisterende rekvirenten i fagsystemet. Metoden returnerer en status eller en feilmelding dersom en feil har inntruffet.

Distribusjonsmodellen forutsetter at brukeren har etablert en kobling mellom FRESH rekvirenten og fagsystem rekvirenten, og at koblingen ligger lagret i FRESH.

Oppdatering av kommunikasjonspartinformasjon i fagsystemet gjøres i henhold til følgende prinsipper:

1. Oppdateringer kan bare gjøres på rekvirenter som allerede finnes i fagsystemet og som det finnes en kobling for.
 - a. Dersom en endring sendes på en kommunikasjonspart som ikke finnes i fagsystemet skal videre behandling av endringer skal avbrytes og det returneres en feilmelding.
2. Oppdateringene overføres til fagsystemet og håndteres av fagsystemet i samme rekkefølge som endringene er fortatt. Dette for å sikre at flere endringer på samme kommunikasjonspart blir gjennomført i riktig rekkefølge.
 - a. Ved feil vil videre distribusjon av endringer til fagsystemet stoppes inntil feilen er korrigert.
3. Dersom det oppstår en feil i behandlingen av en oppdatering skal fagsystemet avbryte behandlingen av resterende endringer og returnere en feilmelding.

Input

handleCommunicationPartyUpdatesIn

- Innpakking av CommunicationParty elementet som er spesifisert i 3.1.2 *CommunicationPartyUpdate* og *UpdateCommunicationPartyInputMsg* - kommunikasjonspart.

Merk at SystemSpecificInformation feltet i CommunicationParty vil være tomt for behandle endringer metoden. De verdier som er lagret i fagsystemet beholdes. Dette er nødvendig fordi oppdateringen er maskinell, og mange av oppdateringene oppdateringer vil komme via A-HER som ikke har støtte for fagsystemspsifikke tilleggsopplysninger.

Output:

handleCommunicationPartyUpdatesOut

- Innpakking av StatusMessage elementet beskrevet i 3.1.12 *StatusMessage and StatusMessageType definisjon*.

StatusMessage

- Status 0
 - Fagsystemets interne primærnøkkel for rekvirenten returneres i returnvalue feltet
 - Trengs for opprettelse av rad i mapping tabell
- Status -1
 - Feilmelding i ErrorMessage elementet
 - Feilsituasjoner kan være forsøk på å opprette allerede eksisterende rekvirenter, feil i input data eller interne feil i fagsystemet.

2.5.3.1 Sjekkliste for implementasjon av metoden

Sjekklisten er eksempel på funksjonalitet vi mener metoden bør ivareta internt og skjult for FRESH som bør kunne forholde seg til et enkelt metodekall mot fagsystemet.

1. Pålogging av FRESH bruker til fagsystemet – fagsystemleverandøren bør ivareta krav til autentisering, autorisasjon og sporing i Web Service metoden
2. Kontroll av rekvirentopplysninger – fagsystemleverandøren bør kontrollere rekvirentopplysningene som sendes inn ved bruk av metoden
3. Endre organisasjonsenheter – Avhengig av datamodellen som ligger til grunn for fagsystemet, bør fagsystemleverandøren vurdere endring av nødvendige interne enheter
4. Endre personer - Avhengig av datamodellen som ligger til grunn for fagsystemet, bør fagsystemleverandøren vurdere endring av personer i intern persontabell
5. Endre rekvirenter - Avhengig av datamodellen som ligger til grunn for fagsystemet, bør fagsystemleverandøren vurdere endring av rekvirenter i interne rekvirenttabell
6. Behandle feilsituasjoner – fagsystemleverandøren må sammenstille og returnere eventuelle feilbeskrivelser til FRESH, se utkast til datakontrakt
7. Avlogging – sesjonen som er etablert bør kobles ned når metoden er gjennomført

2.5.4 XSD Versjonshåndtering - GetXSDVersion

GetXSDVersion returnerer versjonnummeret angitt i XSD filen som inneholder datakontrakten, brukes for fremtidig versjonshåndtering og feilsøk.

Input

Ingen input parametre

Output:

getXSDVersionOut

- innpakking av Version elementet i XSDen
-

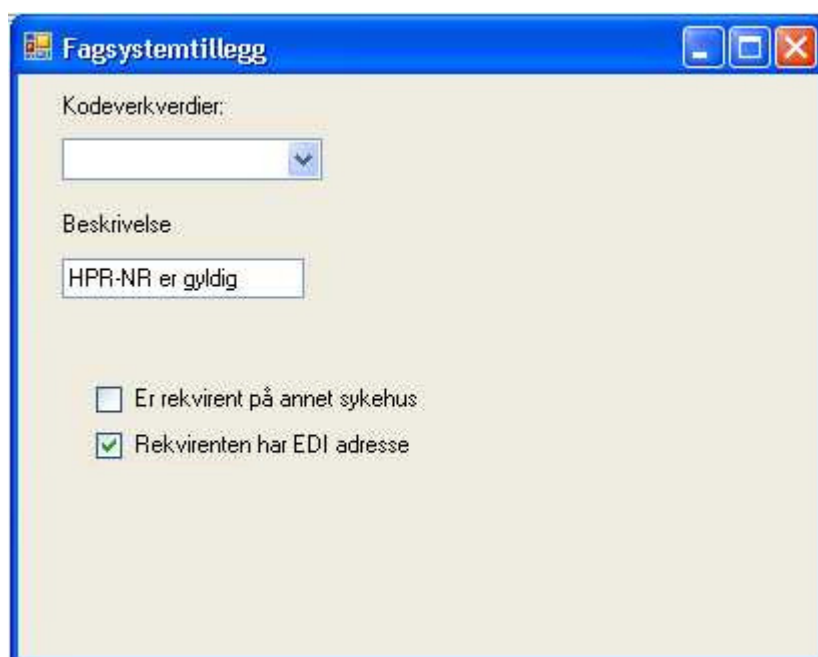
2.5.5 Fagsystemspesifikke tilleggsopplysninger

2.5.5.1 Beskrivelse

En del fagsystemer har behov for tilleggsopplysninger f.eks for å støtte kodeverk som brukes internt i systemet utover det som er definert i basismodellen for en kommunikasjonspart.

Fagsystemspesifikke tilleggsopplysninger følger med som en del av kommunikasjonspart informasjonen i feltet SystemSpecificInformation.

Fagsystemspesifikke tilleggsopplysninger vil, basert på informasjonen fra fagsystemet, bli vist i en dynamisk generert GUI, og brukeren vil kunne redigere fagsystemspesifikke felter gjennom denne GUIen.



Figur 4 Skjermbilde eksempel fagsystemspesifikke tillegg

Denne GUIen er beskrevet i eget vedlegg [3].

2.5.5.2 XAML for spesifikasjon

XAML brukes for å spesifisere og kommunisere de fagsystem spesifikke endringene.

Se <http://msdn.microsoft.com/en-us/library/ms752059.aspx> for mer detaljer om XAML.

2.5.5.3 Opprett kommunikasjonspart og distribuer kommunikasjonspart endringer

De fagsystem spesifikke endringene sendes bare fra FRESH over i forbindelse med opprettelse av ny kommunikasjonspart. Beskrivelsen av de fagsystemspesifikke tilleggene må kunne håndtere de ulike typene rekviere som fagsystemet eventuelt skiller på (et slikt eksempel er skillet på interne og eksterne rekviere i DIPS) og tilhørende fagsystemspesifikke kodeverk.

De fagsystemspesifikke tilleggene vil ikke bli sendt til fagsystemet i forbindelse med distribuer kommunikasjonspart endringer. Verdiene satt da rekvirenten ble opprettet, og eventuelle endringer gjort lokalt i fagsystemet på de fagsystemspesifikke tilleggene, beholdes som de er. SystemspecificInformation vil være tom.

2.5.5.4 Distribusjon av XSD

I første versjon av løsningen distribueres XSD for systemspesifikke tillegg ved hjelp av en URL som spesifiserer hvor FRESH kan hente XSDen.

2.6 Feilhåndtering og feilmeldinger

Feilmeldinger fra fagsystemene vil primært komme som følge av ustabiliteter, feil i konfigurasjoner etc. De fleste feil vil kreve manuelle inngrep for å kunne korrigeres, og det er ikke lagt opp til automatisert feilkorrigering i integrasjonen mellom FRESH og fagsystemene.

Det må beskrives hvilke feil som kan oppstå og hvordan disse skal håndteres av brukeren.

FRESH bruker per idag SOAP Exception for å håndtere feilmeldinger fra Adresseregisteret HER og vil brukes samme mekanisme mot fagsystemene.

2.7 Sikkerhet

2.7.1 Sikkerhetsmodell i pilotperioden og fremover

Dette dokumentet beskriver ikke en sikkerhetsmodell for integrasjonen med fagsystemene. FRESH støtter en enkel form for sikkerhet der man bruker en brukernavn og passord for autentisering. Det er ikke spesifisert ytterligere sikkerhetsmekanismer ut over dette.

Sikkerheten implementere gjennom hvordan man velger å bruke mekanismer i SOAP, mens fokus for dette dokumentet har vært innholdet i SOAP "konvoluttene" og ikke sikkerhetsmekanismene. En enkel gjennomgang sikkerhetsmekanismene i SOAP er gitt i <http://www.codeproject.com/KB/webservices/WS-Security.aspx#Implementation>, denne artikkelen dekker WSS 1.0.

I pilotperioden må man vurdere ulike sikkerhetsbehov og beslutte en sikkerhetsmodell dekker behovene til de ulike regionene, og som vil fungere som løsning for alle relevante fagsystem integrasjoner.

2.7.2 Autentisering, pålogging og avlogging

Det legges opp til en modell der fagsystemet håndterer pålogging og avlogging, og at dette gjøres for hver enkelt Web Service metode hver gang den kalles. FRESH skal ikke trenge å vite noe om pålogging og avlogging i det enkelte fagsystem.

Det er viktig huske på at integrasjonen representerer et system til system grensesnitt. Det er FRESH-serveren som kobler seg til fagsystemet, og sluttbrukeren er allerede autentisert og autorisert i FRESH og forholder seg bare til FRESH systemet.

3 XSD, XML og WSDLdokumentasjon

3.1 Hovedelementer i XSDen.

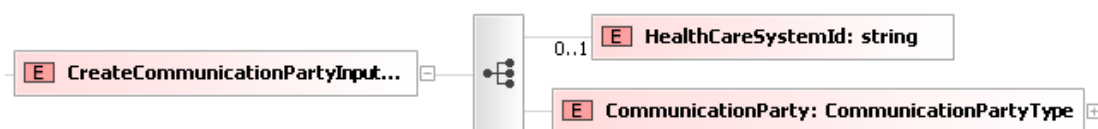
3.1.1 CreateCommunicationPartyInputMsg og CommunicationParty – Kommunikasjonspart elementet

Dette er definisjonen av en kommunikasjonspart i FRESH og beskriver informasjonen som skal utveksles med fagsystemet.

En kommunikasjonspart er definert som et forhold mellom en enhet kalt Parent, og enten en annen enhet eller en person, kalt Child, som er knyttet organisatorisk inn under Parent. Et eksempel kan være en lege (child) på et legekantor (parent) i spesialisthelsetjenesten, eller et helseforetak (parent) og en avdeling (child) på helseforetaket.

Kommunikasjonspart inneholder også adresseinformasjon (geografisk og elektronisk) som gjør det mulig å kommunisere med rekvirenten.

Figuren nedenfor viser en representasjon av input meldingen til CreateCommunicationParty operasjonen for å opprette en kommunikasjonspart (kommunikasjonspart) i fagsystemet.



Figur 5 CreateCommunicationPartyInputMsg definisjon

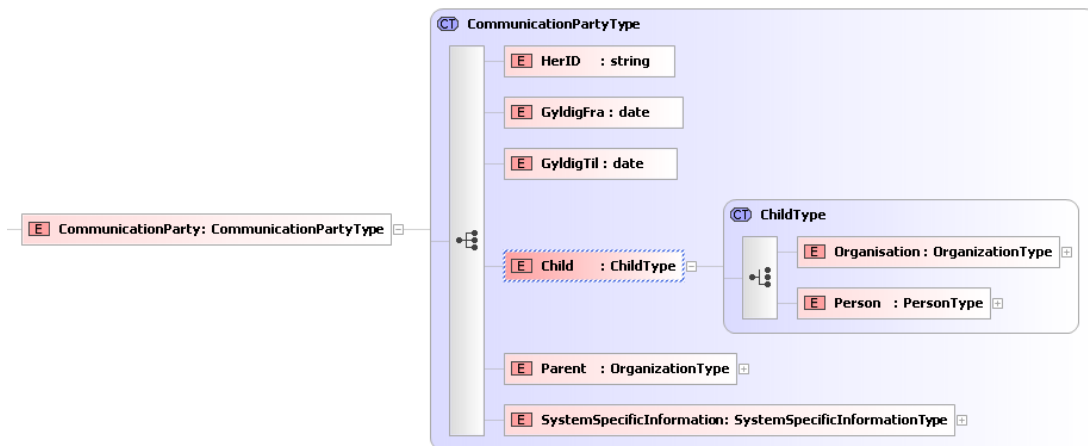
CreateCommunicationPartyInput inneholder to elementer

- HealthCareSystemId – elementet er en unik kode for fagsystemet og brukes av integrasjonsserveren for innholdsbasert routing i de foretak og/eller regioner der dette er ønskelig.
- CommunicationParty – beskrevet nedenfor.

XML kode

```
<xs:element name="CreateCommunicationPartyInputMsg">
  <xs:complexType>
    <xs:sequence>
      <xs:element minOccurs="0" name="HealthCareSystemId" type="xs:string" />
      <xs:element name="CommunicationParty" type="CommunicationPartyType" />
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

Figuren nedenfor viser en representasjon av rekvirenten (CommunicationParty):



Figur 6 CommunicationParty definisjon

Rekvirenten består av følgende elementer

- HerID – kommunikasjonspart ID tildelt av Adresseregisteret HER og er en nasjonal identifikator som unikt identifiserer den enkelte kommunikasjonspart.
- GyldigFra – dato som angir når en kommunikasjonspart blir gyldig og kan foreta rekvisisjoner. En kommunikasjonspart skal ikke kunne foreta rekvisisjoner før denne dato.
- GyldigTil – dato som angir når en kommunikasjonspart ikke lenger er gyldig og ikke lenger kan foreta rekvisisjoner. En kommunikasjonspart skal ikke kunne foreta nye rekvisisjoner etter denne dato.
- Parent – Organisasjonen som er overordnet
- Child – Organisasjonen eller personen som hører organisatorisk inn under Parent
- SystemSpecificInformation – en del fagsystem har utvidelser av rekvirentbegrepet internt i sin modell. Alle de fagsystemspesifikke feltene håndteres i elementet SystemSpecificInformation.

XML kode

```

<xs:complexType xmlns:xs="http://www.w3.org/2001/XMLSchema" name="CommunicationPartyType">
  <xs:sequence>
    <xs:element name="HerID" type="xs:string"/>
    <xs:element name="GyldigFra" type="xs:date"/>
    <xs:element minOccurs="0" name="GyldigTil" type="xs:date"/>
    <xs:element name="Child" type="ChildType"/>
    <xs:element name="Parent" type="OrganizationType"/>
    <xs:element name="SystemSpecificInformation" type="SystemSpecificInformationType"/>
    <xs:element name="LocalSystemCommunicationPartyId" type="xs:string" />
  </xs:sequence>
</xs:complexType>
  
```

Endringer i forhold til FR 1.0:

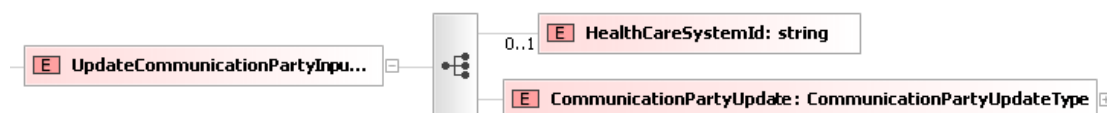
Endringer (merket gul bakgrunn)

- minOccurs="0" er lagt til for GyldigTil elementet
- nytt felt for lokal rekvirentkode

3.1.2 CommunicationPartyUpdate og UpdateCommunicationPartyInputMsg–kommunikasjonspart oppdatering

CommunicationPartyUpdate er en CommunicationPartyUpdateType som er en definisjonen av hvordan rekvirenter som er endret skal overføres til fagsystemet

Figuren nedenfor viser en representasjon av input meldingen til UpdateCommunicationPartyInputMsg operasjonen for å opprette en kommunikasjonspart (kommunikasjonspart) i fagsystemet.



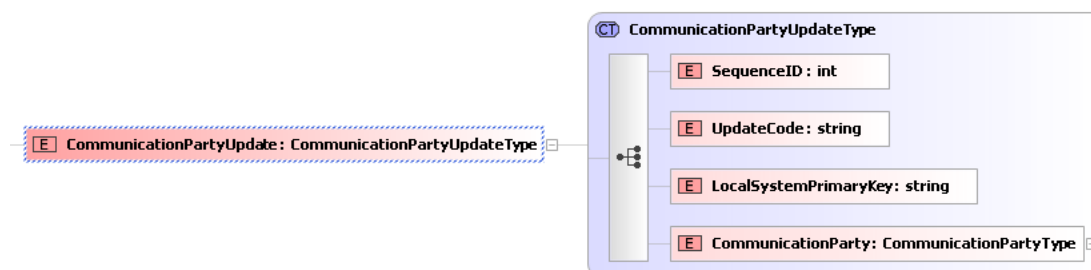
Figur 7 UpdateCommunicationPartyInputMsg definisjon

UpdateCommunicationPartyInputMsg inneholder to elementer

- HealthCareSystemId – elementet er en unik kode for fagsystemet og brukes av integrasjonsserveren for innholdsbasert routing i de foretak og/eller regioner der dette er ønskelig.
- CommunicationPartyUpdate – beskrevet nedenfor.

XML Kode

```
<xs:element xmlns:xs="http://www.w3.org/2001/XMLSchema" name="UpdateCommunicationPartyInputMsg">
  <xs:complexType>
    <xs:sequence>
      <xs:element minOccurs="0" name="HealthCareSystemId" type="xs:string"/>
      <xs:element name="CommunicationPartyUpdate" type="CommunicationPartyUpdateType"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
```



Figur 8 CommunicationPartyUpdate definisjon

CommunicationPartyUpdatetype består av følgende elementer

- SequenceID – teller som unikt identifiserer den enkelt endring. Brukes i forbindelse med sporing, samt sikre at endringer overføres i riktig rekkefølge.
 - UpdateCode – kode for typen oppdatering. Defineres nærmere – usikker på behov for å si noe om typen endring.
 - LocalSystemPrimaryKey – primærnøkkel i fagsystemet som unikt identifiserer rekvirenten i fagsystemet. Holdes av FRESH i mapping/koblingstabellen.
 - CommunicationParty – den oppdaterte kommunikasjonspartinformasjonen
-

XML kode

```
<xs:complexType name="CommunicationPartyUpdateType">
  <xs:sequence>
    <xs:element minOccurs="0" name="SequenceID" type="xs:int" />
    <xs:element minOccurs="0" name="UpdateCode" type="xs:string" />
    <xs:element minOccurs="0" name="LocalSystemPrimaryKey" type="xs:string" />
    <xs:element minOccurs="0" name="CommunicationParty" type="CommunicationPartyType" />
  </xs:sequence>
</xs:complexType>
```

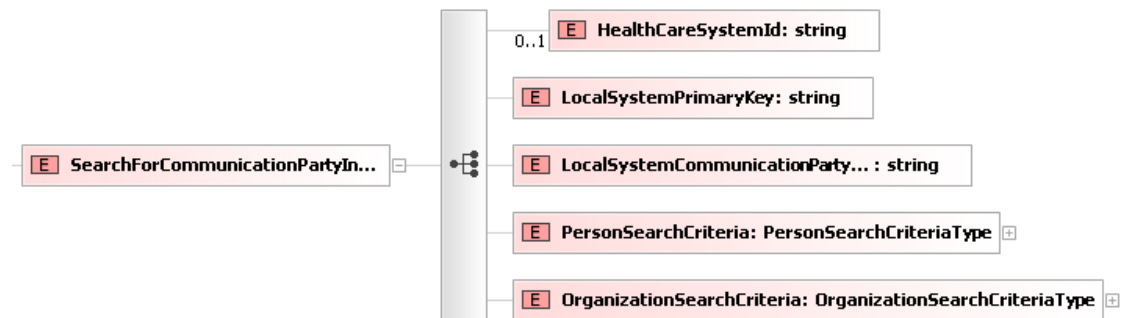
Endringer i forhold til FR 1.0:

Endringer (merket gul bakgrunn)

- minOccurs="0" er lagt til for alle elementer

3.1.3 SearchForCommunicationPartyInputMsg – struktur for å søke i fagsystem

SearchForCommunicationPartyInputMsg definerer strukturen som brukes for å gjøres søk mot fagsystemet .



Figur 9 SearchForCommunicationPartyInputMsg definisjon

SearchForCommunicationPartyInputMsg består av følgende elementer

- HealthCareSystemId – elementet er en unik kode for fagsystemet og brukes av integrasjonsserveren for innholdsbasert routing i de foretak og/eller regioner der dette er ønskelig.
- LocalSystemPrimaryKey – primærnøkkel i fagsystemet
- LocalSystemCommunicationPartyID – Id for rekvirenten i fagsystemet
- PersonSearchCriteria – elementer knyttet til en person som fornavn, etternavn og HPR-nummer. For detaljer se 3.1.3.1 PersonSearchCriteriaType nedenfor.
- OrganizationSearchCriteria – elementer knyttet til søk på organisasjonsinformasjon som navn, organisasjonsnummer og kommune. For detaljer se 3.1.3.2 OrganizationSearchCriteriaType nedenfor.

XML kode

```
<xs:element name="SearchForCommunicationPartyInputMsg">
  <xs:complexType>
    <xs:sequence>
      <xs:element minOccurs="0" name="HealthCareSystemId" type="xs:string"/>
      <xs:element minOccurs="0" name="LocalSystemPrimaryKey" type="xs:string" />
      <xs:element minOccurs="0" name="LocalSystemCommunicationPartyId" type="xs:string" />
```

```

<xs:element minOccurs="0" name="HerID" type="xs:string" />
<xs:element name="Active" type="xs:boolean" />
<xs:element name="PersonSearchCriteria" type="PersonSearchCriteriaType" />
<xs:element name="OrganizationSearchCriteria" type="OrganizationSearchCriteriaType" />
<xs:element name="ServiceSearchCriteria" type="ServiceSearchCriteriaType" />
</xs:sequence>
</xs:complexType>
</xs:element>

```

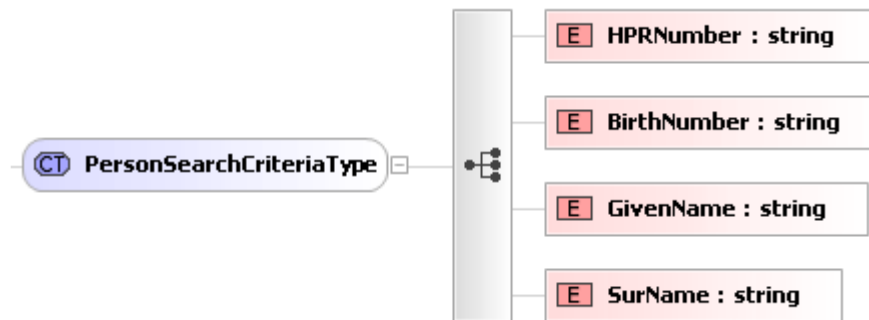
Endringer i forhold til FR 1.0:

Endringer (merket med gul bakgrunn)

- Element med HerId er lagt til
- Element med Active er lagt til. Om kommunikasjonsparten er aktiv eller ikke styres av datointervaller (gyldig_fra og gyldig_til)
- Element med ServiceSearchCriteria er lagt til

3.1.3.1 PersonSearchCriteriaType

PersonSearchCriteriaType spesifiserer de elementer man kan oppgi verdier for å søke i en persondelen av en kommunikasjonspart.



Figur 10 PersonSearchCriteriaType definisjon

Elementene er definert ut fra hva som er tilgjengelig i FRESH-brukergrensesnittet.

XML kode

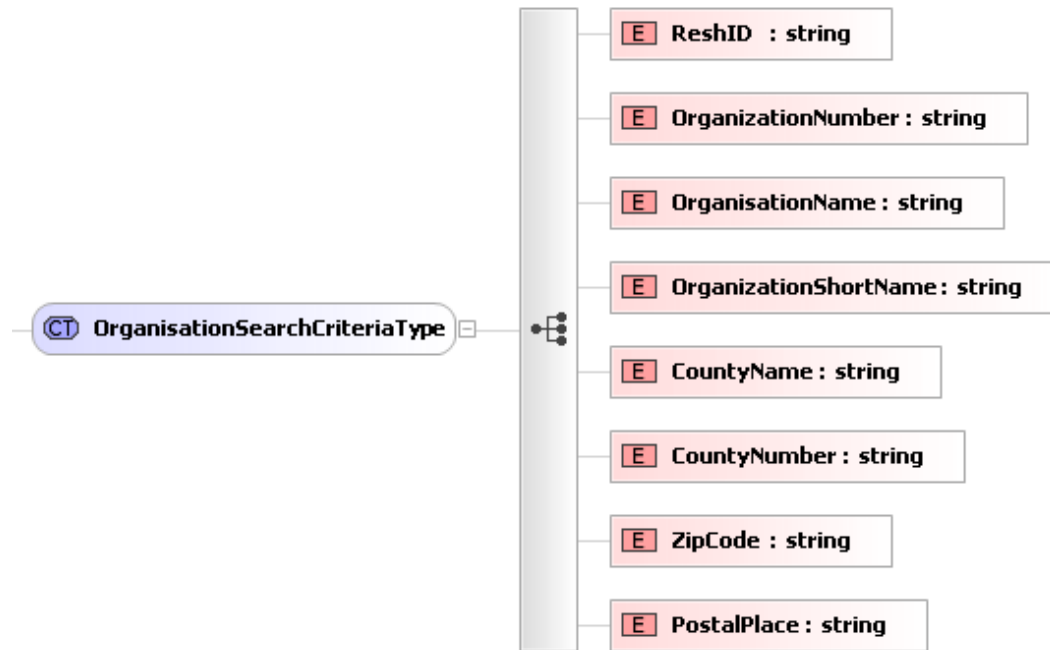
```

<xs:complexType xmlns:xs="http://www.w3.org/2001/XMLSchema" name="PersonSearchCriteriaType">
  <xs:sequence>
    <xs:element minOccurs="0" name="HPRNumber" type="xs:string"/>
    <xs:element minOccurs="0" name="BirthNumber" type="xs:string"/>
    <xs:element minOccurs="0" name="GivenName" type="xs:string"/>
    <xs:element minOccurs="0" name="SurName" type="xs:string"/>
  </xs:sequence>
</xs:complexType>

```

3.1.3.2 OrganizationSearchCriteriaType

OrganizationSearchCriteriaType spesifiserer de elementer man kan oppgi verdier for å søke i en organisasjonsdelen av en kommunikasjonspart.



Figur 11 OrganizationSearchCriteriaType definisjon

Elementene er definert ut fra hva som er tilgjengelig i FRESH-brukergrensesnittet.

XML kode

```

<xs:complexType xmlns:xs="http://www.w3.org/2001/XMLSchema" name="OrganizationSearchCriteriaType">
  <xs:sequence>
    <xs:element minOccurs="0" name="ReshID" type="xs:string"/>
    <xs:element minOccurs="0" name="OrganizationNumber" type="xs:string"/>
    <xs:element minOccurs="0" name="OrganizationName" type="xs:string"/>
    <xs:element minOccurs="0" name="OrganizationShortName" type="xs:string"/>
    <xs:element minOccurs="0" name="CountyName" type="xs:string"/>
    <xs:element minOccurs="0" name="CountyNumber" type="xs:string"/>
    <xs:element minOccurs="0" name="ZipCode" type="xs:string"/>
    <xs:element minOccurs="0" name="PostalPlace" type="xs:string"/>
  </xs:sequence>
</xs:complexType>

```

3.1.3.3 ServiceSearchCriteriaType

ServiceSearchCriteriaType spesifiserer de elementer man kan oppgi verdier for å søke i en tjenstedelen av en kommunikasjonspart.

XML kode

```

<xs:complexType name="ServiceSearchCriteriaType">
  <xs:sequence>
    <xs:element name="ServiceCode" type="xs:string" minOccurs="0" />
    <xs:element name="ServiceDescription" type="xs:string" minOccurs="0" />
    <xs:element name="ServiceGroup" type="xs:string" minOccurs="0" />
    <xs:element name="ServiceGroupDescription" type="xs:string" minOccurs="0" />
  </xs:sequence>
</xs:complexType>

```

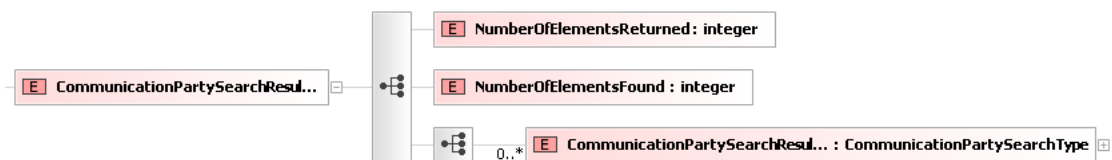
Endringer i forhold til FR 1.0:

Endringer (merket med gul bakgrunn)

- ComplexType ServiceSearchCriteriaType er lagt til
- Beskrivelser av elementene er gitt i kapittel som beskriver ServiceType

3.1.4 CommunicationPartySearchResultArray – Liste over rekvirenter funnet i fagsystemet

CommunicationPartySearchResultArray er et array av kommunikasjonspart informasjon som returneres fra fagsystemet etter at det er gjort et søk fra FRESH . Et søk kan resultere i 0 eller flere treff.



Figur 12 CommunicationPartySearchResultArray definisjon

CommunicationPartySearchResultArray består av følgende elementer

- NumberOfElementsReturned – antall returnerte elementer
- NumberOfElementsFound – antall treff som tilfredsstiller søkekriteriene
- CommunicationPartySearchResultElement – en CommunicationPartySearchType. Det returneres en sekvens med null eller flere av CommunicationPartySearchResultElement der antallet tilsvarer verdien i NumberOfElementsReturned.

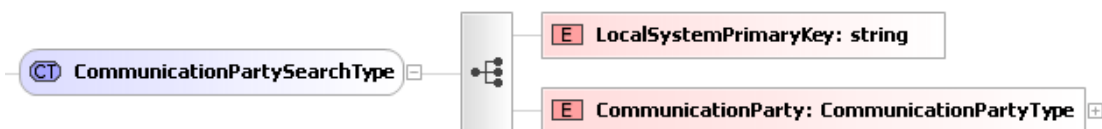
NumberOfElementsReturned og NumberOfElementsFound kan være forskjellige. Dermed kan antall returnerte elementer kan begrenses, og man unngår søk som potensielt sett returnerer alle kommunikasjonsparter i databasen. Brukeren må eventuelt spisse sitt søk dersom kommunikasjonsparten man leter etter ikke finnes.

XML kode

```
<xs:element xmlns:xs="http://www.w3.org/2001/XMLSchema"
name="CommunicationPartySearchResultArray">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="NumberOfElementsReturned" type="xs:integer"/>
      <xs:element name="NumberOfElementsFound" type="xs:integer"/>
      <xs:sequence>
        <xs:element minOccurs="0" maxOccurs="unbounded"
name="CommunicationPartySearchResultElement" type="CommunicationPartySearchType"/>
      </xs:sequence>
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

3.1.4.1 CommunicationPartySearchType

CommunicationPartySearchType definerer kommunikasjonspartinformasjonen som skal returneres av fagsystemet ved et søk



Figur 13 CommunicationPartySearchType definisjon

CommunicationPartySearchType består av følgende elementer

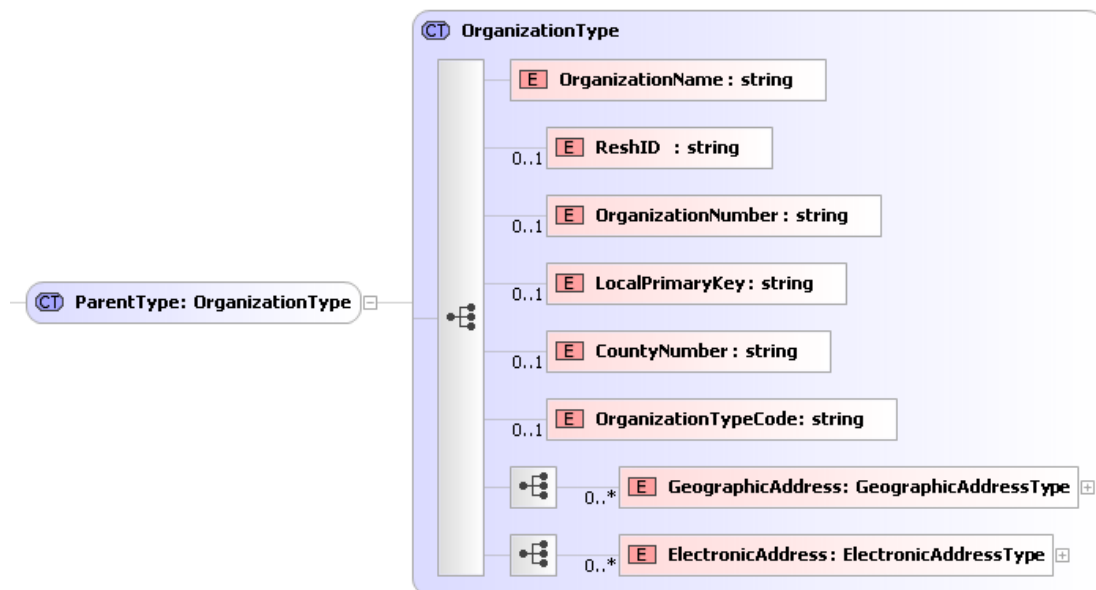
- LocalSystemPrimaryKey – primærnøkkelen i fagsystemet for rekvirenten. Denne brukes i FRESH for å kunne koble mellom FRESH kommunikasjonspart og fagsystemrekvirent.
- CommunicationParty – kommunikasjonspart informasjonen som definert av CommunicationPartyType

XML kode

```
<xs:complexType xmlns:xs="http://www.w3.org/2001/XMLSchema" name="CommunicationPartySearchType">
  <xs:sequence>
    <xs:element name="LocalSystemPrimaryKey" type="xs:string"/>
    <xs:element name="CommunicationParty" type="CommunicationPartyType"/>
  </xs:sequence>
</xs:complexType>
```

3.1.5 ParentType – definisjon av Parent delen av en kommunikasjonspart.

En ParentType definerer parent delen av en kommunikasjonspart.



Figur 14 ParentType definisjon

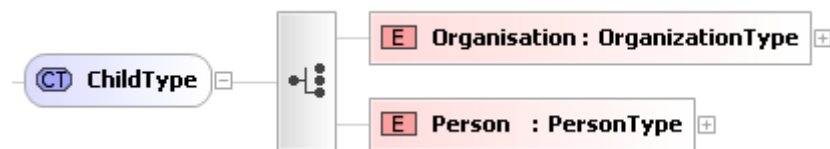
En parent er alltid en organisasjon: For nærmere beskrivelse se organisasjonsbeskrivelsen nedenfor.

XML kode for ParentType

```
<xs:complexType xmlns:xs="http://www.w3.org/2001/XMLSchema" name="ParentType">
  <xs:complexContent mixed="false">
    <xs:extension base="OrganizationType"/>
  </xs:complexContent>
</xs:complexType>
```

3.1.6 ChildType – definisjon av Child delen av en kommunikasjonspart

Child kan være enten en person eller en organisasjon, andre typer kan komme på sikt. Se definisjonene under.



Figur 15 ChildType definisjon

XML kode for ChildType

```
<xs:complexType name="ChildType">
  <xs:choice>
    <xs:element name="Organization" type="OrganizationType" />
    <xs:element name="Person" type="PersonType" />
    <xs:element name="Service" type="ServiceType" />
  </xs:choice>
</xs:complexType>
```

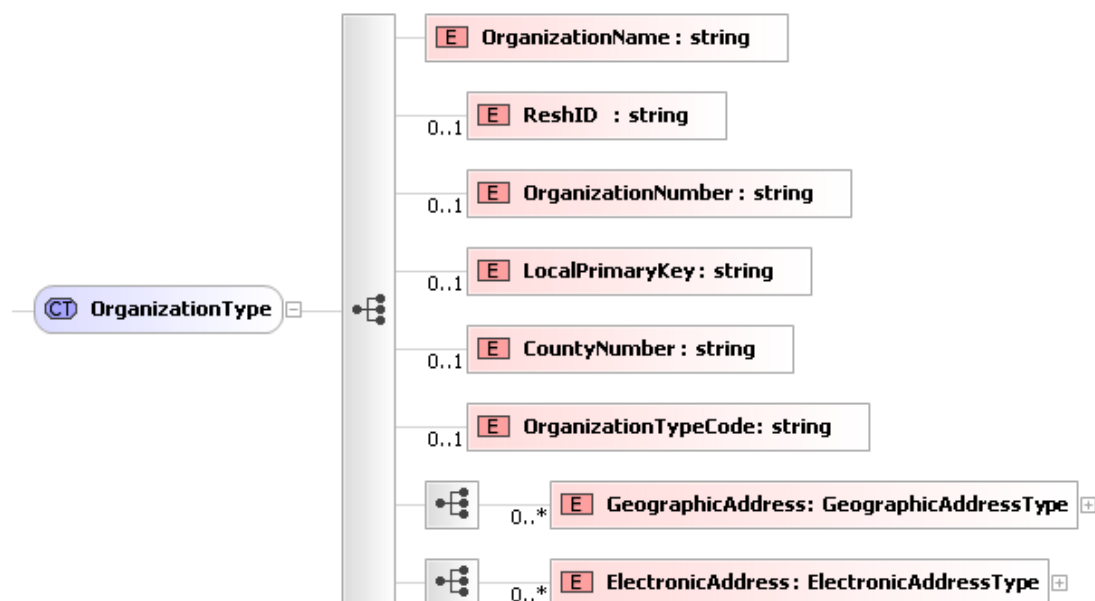
Endringer i forhold til FR 1.0:

Endringer (merket med gul bakgrunn)

- Element med Service er lagt til

3.1.7 OrganizationType definisjon

OrganizationType definerer representasjonen av et organisasjonselement i en kommunikasjonspart. Organisasjonen inngår i Parent og kan også inngå i Child elementet.



Figur 16 OrganizationType definisjon

OrganizationType består av følgende elementer

- OrganizationName –organisasjonens navn
- ReshID – organisasjonens RESH ID når det det er en organisasjon i spesialist heljestjenesten.

- OrganizationNumber – organisasjonens organisasjonsnummer som registrert i Brønøysundregisteret. Underavdelinger kan mangle organisasjonsnummer.
- LocalPrimaryKey – primærnøkkel i fagsystemet
- CountyNumber – kommunenummer
- OrganizationTypeCode – tilsvarer organisasjonsbetegnelse internt i FRESH og Adresseregisteret HER. Tilsvarer kodeverk 8624.
- GeographicAddress - geografiske adresseer
- ElectronicAddress – elektroniske adresse

XML kode for OrganizationType

```
<xs:complexType xmlns:xs="http://www.w3.org/2001/XMLSchema" name="OrganizationType">
  <xs:sequence>
    <xs:element name="OrganizationName" type="xs:string"/>
    <xs:element minOccurs="0" name="ReshID" type="xs:string"/>
    <xs:element minOccurs="0" name="OrganizationNumber" type="xs:string"/>
    <xs:element minOccurs="0" name="LocalPrimaryKey" type="xs:string"/>
    <xs:element minOccurs="0" name="CountyNumber" type="xs:string"/>
    <xs:element minOccurs="0" name="OrganizationTypeCode" type="xs:string"/>
    <xs:element minOccurs="0" name="DerivedOrganizationID" type="xs:string"/>
    <xs:sequence>
      <xs:element minOccurs="0" maxOccurs="unbounded" name="GeographicAddress"
type="GeographicAddressType"/>
    </xs:sequence>
    <xs:sequence>
      <xs:element minOccurs="0" maxOccurs="unbounded" name="ElectronicAddress"
type="ElectronicAddressType"/>
    </xs:sequence>
  </xs:sequence>
</xs:complexType>
```

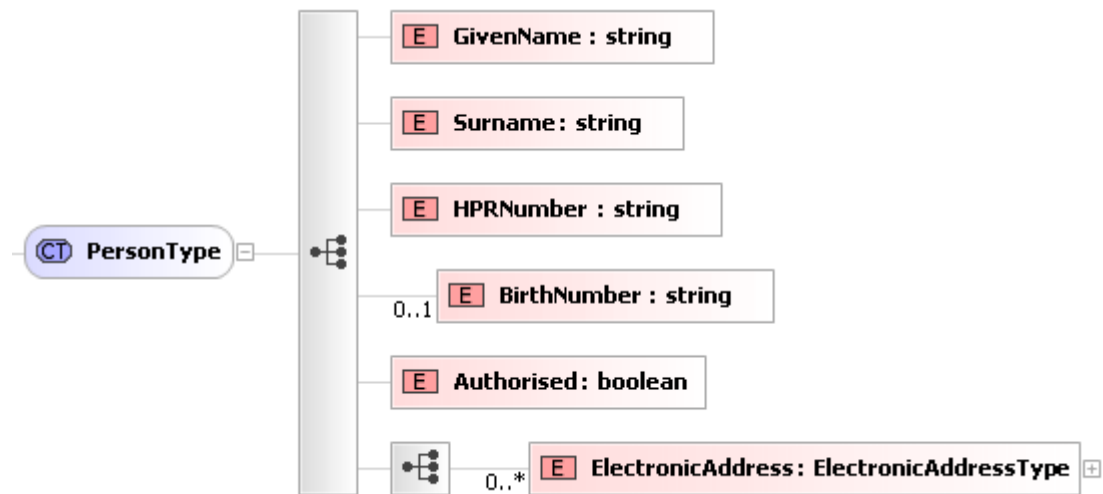
Endringer i forhold til FR 1.0:

Endringer (merket med gul bakgrunn)

- Element med DerivedOrganizationID er lagt til. Dette feltet brukes i de tilfeller hvor man ikke har tilgjengelig data for noen av de andre elementene til å identifisere en organisasjon, f.eks i primærhelsetjenesten.

3.1.8 PersonType definisjon

PersonType definerer representasjonen av en person og brukes i Child elementet når dette er en person.



Figur 17 PersonType definisjon

PersonType består av følgende elementer

- GivenName – fornavn
- SurName – etternavn
- HPRNumber – HPR nummer som registrert i Helsepersonellregisteret.
- BirthNumber – fødselsnummer
- Authorised – autorisert eller ikke
- ElectronicAddress – elektroniske adresser som epost, telefon mm

XML kode for PersonType

```
<xs:complexType xmlns:xs="http://www.w3.org/2001/XMLSchema" name="PersonType">
  <xs:sequence>
    <xs:element name="GivenName" type="xs:string"/>
    <xs:element name="Surname" type="xs:string"/>
    <xs:element name="HPRNumber" type="xs:string"/>
    <xs:element minOccurs="0" name="BirthNumber" type="xs:string"/>
    <xs:element name="Authorised" type="xs:boolean"/>
    <xs:sequence>
      <xs:element minOccurs="0" maxOccurs="unbounded" name="profession" type="xs:string" />
    </xs:sequence>
    <xs:sequence>
      <xs:element minOccurs="0" maxOccurs="unbounded" name="speciality" type="xs:string" />
    </xs:sequence>
    <xs:element minOccurs="0" name="title" type="xs:string" />
    <xs:sequence>
      <xs:element minOccurs="0" maxOccurs="unbounded" name="ElectronicAddress"
type="ElectronicAddressType"/>
    </xs:sequence>
  </xs:sequence>
</xs:complexType>
```

Endringer i forhold til FR 1.0:

Endringer (merket med gul bakgrunn)

- Sekvens med Profession er lagt til. Denne informasjonen kommer fra HPR-registeret og inneholder yrke knyttet opp mot en person.

-
- Sekvens med Speciality er lagt til. Denne informasjonen kommer fra HPR-registeret og inneholder medisinsk spesialitet knyttet opp mot en person.
 - Element med Title. Dette feltet knytter en tittel opp mot en personrekvirent.

3.1.9 ServiceType definisjon

ServiceType definerer representasjonen av en tjeneste og brukes i Child elementet når dette er en tjeneste.

ServiceType består av følgende elementer

- ServiceCode – Tjenestekode gitt av KiTH's kodeverk, feks 'R01'
- ServiceCodeDescription – Beskrivelse av tjenestekode
- ServiceGroup – Tjenestegruppe gitt av KiTH's kodeverk, feks '8655'
- ServiceGroupDescription – Beskrivelse av tjenestegruppe
- ElectronicAddress – elektroniske adresser som epost, telefon mm

XML kode for ServiceType

```
<xs:complexType name="ServiceType">
  <xs:sequence>
    <xs:element name="ServiceCode" type="xs:string" />
    <xs:element minOccurs="0" name="ServiceCodeDescription" type="xs:string" />
    <xs:element name="ServiceGroup" type="xs:string" />
    <xs:element minOccurs="0" name="ServiceGroupDescription" type="xs:string" />
  </xs:sequence>
  <xs:element minOccurs="0" maxOccurs="unbounded" name="ElectronicAddress"
type="ElectronicAddressType" />
</xs:sequence>
</xs:complexType>
```

Endringer i forhold til FR 1.0:

Endringer (merket med gul bakgrunn)

- ComplexType for tjeneste er lagt til

3.1.10 GeographicAddressType definisjon

GeographicAddressType definerer representasjonen av en geografisk adresse og brukes i OrganisationType elementene.

Bare organisasjoner har geografiske adresser.



Figur 18 GeographicAdressType definisjon

XML kode for Geografisk adresse

```
<xs:complexType xmlns:xs="http://www.w3.org/2001/XMLSchema"
name="GeographicAddressType">
  <xs:sequence>
    <xs:element name="Streetname" type="xs:string"/>
    <xs:element name="Streetnumber" type="xs:string"/>
    <xs:element name="ZipCode" type="xs:string"/>
    <xs:element name="PostalArea" type="xs:string"/>
    <xs:element name="POBoxNumber" type="xs:string"/>
    <xs:element name="AddressType" type="GeographicAddressCodeType"/>
  </xs:sequence>
</xs:complexType>
```

3.1.10.1 Koder for geografiske adressetyper

Det beskrevne koderverket er brukt i Adresseregisteret HER, og FRESH har valgt å følge samme standard.

H	Bostedsadresse	1
HP	Folkeregisteradresse	2
HV	Ferieadresse	3
INV	Faktureringsadresse	4
PST	Postadresse	5
RES	Besøksadresse	6
TMP	Midlertidig adresse	7
WP	Arbeidsadresse	8

XML kode

```
<xs:complexType xmlns:xs="http://www.w3.org/2001/XMLSchema" name="GeographicAddressCodeType">
  <xs:sequence>
    <xs:element name="Code">
      <xs:simpleType>
        <xs:restriction base="xs:string">
          <xs:enumeration value="H"/>
          <xs:enumeration value="HP"/>
          <xs:enumeration value="HV"/>
          <xs:enumeration value="INV"/>
          <xs:enumeration value="PST"/>
          <xs:enumeration value="RES"/>
          <xs:enumeration value="TMP"/>
        </xs:restriction>
      </xs:simpleType>
    </xs:element>
    <xs:element name="CodeDescription">
```

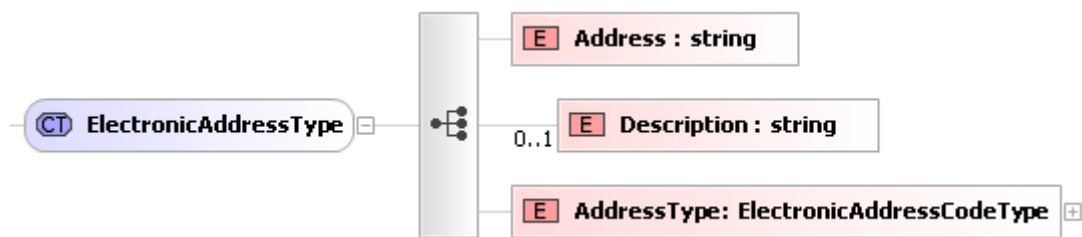
```

<xs:simpleType>
  <xs:restriction base="xs:string">
    <xs:enumeration value="H - 1 - Bostedsadresse" />
    <xs:enumeration value="HP - 2 - Folkeregisteradresse" />
    <xs:enumeration value="HV - 3 - Ferieadresse" />
    <xs:enumeration value="INV - 4 - Faktureringsadresse" />
    <xs:enumeration value="PST - 5 - Postadresse" />
    <xs:enumeration value="RES - 6 - Besoeksadresse" />
    <xs:enumeration value="TMP - 7 - Midlertidig adresse" />
  </xs:restriction>
</xs:simpleType>
</xs:element>
</xs:sequence>
</xs:complexType>

```

3.1.11 ElectronicAddressType definisjon

ElectronicAddressType definerer representasjonen av en elektronisk adresse og brukes i OrganizationType og PersonType elementene. Både organisasjoner og personer kan ha elektroniske adresser.



Figur 19 ElectronicAddressType definisjon

XML kode

```

<xs:complexType xmlns:xs="http://www.w3.org/2001/XMLSchema" name="ElectronicAddressType">
  <xs:sequence>
    <xs:element name="Address" type="xs:string"/>
    <xs:element minOccurs="0" name="Description" type="xs:string"/>
    <xs:element name="AddressType" type="ElectronicAddressCodeType"/>
  </xs:sequence>
</xs:complexType>

```

3.1.11.1 Koder for elektroniske adresse typer

Det beskrevne koderverket brukes i Adresseregisteret HER, og FRESH har valgt å følge samme standard.

E_BRV	Brev eller annen utskrift på et fysisk medium
E_DIV	Annet
E_DS	Digital sertifikat
E EDI	EDI (Standardisert elektronisk melding)
E_EPO	Epost-adresse
E_FAX	Faxnummer
E_TLF	Telefonnummer
E_URL	Hjemmeside

XML kode

```

<xs:complexType xmlns:xs="http://www.w3.org/2001/XMLSchema" name="ElectronicAddressCodeType">
  <xs:sequence>
    <xs:element name="Code">
      <xs:simpleType>
        <xs:restriction base="xs:string">
          <xs:enumeration value="E_BRV" />
          <xs:enumeration value="E_DIV" />
          <xs:enumeration value="E_DS" />
          <xs:enumeration value="E EDI" />
          <xs:enumeration value="E_EPO" />
        </xs:restriction>
      </xs:simpleType>
    </xs:element>
  </xs:sequence>
</xs:complexType>

```

```

        <xs:enumeration value="E_FAX"/>
        <xs:enumeration value="E_TLF"/>
        <xs:enumeration value="E_URL"/>
        <xs:enumeration value="E_MLD" />
        <xs:enumeration value="E_LOK" />
    </xs:restriction>
</xs:simpleType>
</xs:element>
<xs:element name="CodeDescription">
    <xs:simpleType>
        <xs:restriction base="xs:string">
            <xs:enumeration value="E_BRV - Brev eller utskrift paa annet fysisk medium"/>
            <xs:enumeration value="E_DIV - Annet"/>
            <xs:enumeration value="E_DS - Digitalt sertifikat"/>
            <xs:enumeration value="E EDI - EDI (Standard elektronisk melding)"/>
            <xs:enumeration value="E_EPO - EPost-adresse"/>
            <xs:enumeration value="E_FAX - Fax"/>
            <xs:enumeration value="E_TLF - Telefon"/>
            <xs:enumeration value="E_URL - Hjemmeside"/>
            <xs:enumeration value="E_MLD - Meldingsinformasjon" />
            <xs:enumeration value="E_LOK - Lokal Elektronisk Adresse" />
        </xs:restriction>
    </xs:simpleType>
</xs:element>
</xs:sequence>
</xs:complexType>

```

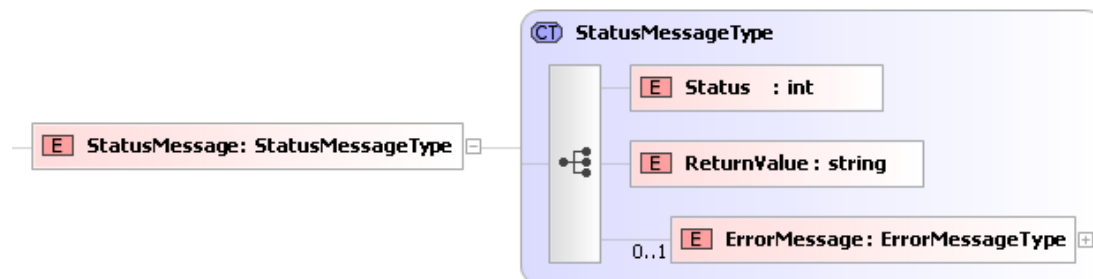
Endringer i forhold til FR 1.0:

Endringer (merket med gul bakgrunn)

- Element med type E_MLD lagt til. En midlertidig løsning ifm meldingstyper inntill SystemSpecificInformation feltet implementeres
- Element med type E_LOK lagt til. Brukes av integrasjonsplattform for ruting av melding til mottaker (lokal sluttadresse for en kommunikasjonspart)

3.1.12 StatusMessage and StatusMessageType definisjon

StatusMessage inneholder returverdien fra fagsystemet, og angir om metoden var en suksess eller feilet.



Figur 20 StatusMessage og StatusMessageType definisjon

StatusMessageType består av følgende elementer

- Status - angir om metoden var suksess eller feilet.
 - 0 – suksess
 - -1 – feilet
- ReturnValue – returverdien fra metoden
 - Verdien er fagsystemets primærnøkkel ved Opprett kommunikasjonspart

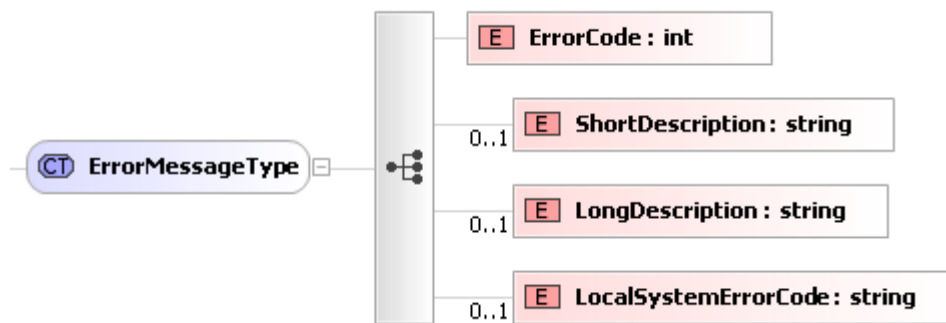
- Verdien settes til NULL for de andre metodene
- ErrorMessage – beskriver feilen dersom metoden feilet og Status = -1

XML kode

```
<xs:complexType xmlns:xs="http://www.w3.org/2001/XMLSchema" name="StatusMessageType">
  <xs:sequence>
    <xs:element name="Status" type="xs:int"/>
    <xs:element name="ErrorID" type="xs:string"/>
    <xs:element minOccurs="0" name="ErrorMessage" type="ErrorMessageType"/>
  </xs:sequence>
</xs:complexType>
```

3.1.13 ErrorMessageType

ErrorMessageType inneholder feilmeldinger fra fagsystemet når metoden har feilet.



Figur 21 ErrorMessageType definisjon

ErrorMessageType består av følgende elementer

- ErrorCode – kode som beskriver feilen. Denne er definert av FRESH og beskriver ulike feilsituasjoner for de enkelte metodekall.
- ShortDescription – kort beskrivelse av feilen fra fagsystemet.
- LongDescription – lang beskrivelse. Inneholder supplerende feilbeskrivelsestekst fra fagsystemet.
- LocalSystemErrorCode – feilkoden i fagsystemet.

LongDescription og LocalSystemErrorCode er primært ment for å lette utvikling og testing, fagsystemet bør vurdere om disse skal kunne skrus av.

XML kode

```
<xs:complexType xmlns:xs="http://www.w3.org/2001/XMLSchema" name="ErrorMessageType">
  <xs:sequence>
    <xs:element name="ErrorCode" type="xs:int"/>
    <xs:element minOccurs="0" name="ShortDescription" type="xs:string"/>
    <xs:element minOccurs="0" name="LongDescription" type="xs:string"/>
    <xs:element minOccurs="0" name="LocalSystemErrorCode" type="xs:string"/>
  </xs:sequence>
</xs:complexType>
```

3.2 Detaljert dokumentasjon

3.2.1 XSD

Det finnes en fullstendig HTML dokumentasjon for XSDene i vedlegg [Xyz], samt en XSD i

3.3 Kardinalitet og sekvens

De fleste elementene er mandatory med en kardinalitet på 1. Dette er default og vises ikke i figurene nedenfor eller i XSD koden. Man skal derfor anta at der det ikke er spesifisert noen kardinalitet er max og min lik 1 og elementet derfor påkrevd. Elementer som har annen kardinalitet er angitt i figurer og XSD koden, dette er typisk 0..unbounded og 1..unbounded.

Det er brukt sekvens i alle XSDer blant annet for å sikre at generert XML blir enkelt lesbar gjennom at man slipper å få elementene i en potensielt tilfeldig rekkefølge.

3.4 XSD schema for kommunikasjonspartinformasjon - CommuncationParty

Elementet for kommunikasjonspartinformasjon er definert i elementet Communication party. Elementet er bygd opp av andre typer definert innenfor samme xsd, typene er ordnet i økende grad av detaljering for å lette manuell lesing.

XSD filen er tilgjengelig i [3].

3.5 Eksempel XML for Communication party

TODO – nytt oppdatert eksempel fra testing der alle elementer er satt.

```
<CommunicationParty xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="../IRRESHCommunicationParty ver 1.4.xsd">
```

```
  <HerID>173</HerID>
  <GyldigFra>2011-07-05</GyldigFra>
  <GyldigTil>1988-03-13</GyldigTil>

  <Child>
    <Person>
      <GivenName>Ola</GivenName>
      <Surname>Nordmann</Surname>
      <HPRNumber>123456789</HPRNumber>
      <BirthNumber>01010122334</BirthNumber>
      <Authorised>true</Authorised>
      <ElectronicAddress>
        <Address>ola.nordmann@hfx.nhn.no</Address>
        <Description>string</Description>
        <AddressType>
          <Code>E_EDI</Code>
          <CodeDescription>E_EDI - EDI (Standard elektronisk
melding)</CodeDescription>
        </AddressType>
      </ElectronicAddress>
    </Person>
  </Child>
```

```

<Parent>
  <OrganizationName>HFX</OrganizationName>
  <ReshID>4637920</ReshID>
  <OrganizationNumber>93277068</OrganizationNumber>
  <LocalPrimaryKey>449871</LocalPrimaryKey>
  <CountyNumber>Kommunennummer - 4 siffer</CountyNumber>
  <OrganizationTypeCode>3</OrganizationTypeCode >
  <GeographicAddress>
    <Streetname>Lillegrensen</Streetname>
    <Streetnumber>5</Streetnumber>
    <ZipCode>0261</ZipCode>
    <PostalArea>Oslo</PostalArea>
    <POBoxNumber></POBoxNumber>
    <AddressType>
      <Code>H</Code>
      <CodeDescription>PST - 5 - Postadresse</CodeDescription>
    </AddressType>
  </GeographicAddress>
  <ElectronicAddress>
    <Address>post@hfx.nhn.no</Address>
    <AddressType>
      <Code>E_ED1</Code>
      <CodeDescription>E_ED1 - epostadresse</CodeDescription>
    </AddressType>
  </ElectronicAddress>
</Parent>

<SystemSpecificInformation>
  <GUICode>string</GUICode>
</SystemSpecificInformation>

</CommunicationParty> <SystemSpecificInformation>
  <GUICode>
  <![CDATA[

<Window x:Class="HER_POC_GUI.Window1"
  xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
  xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
  Title="HER GUI POC" Height="300" Width="300">
  <Grid>
    <TextBox Height="23" Margin="19,50,139,0"
      Name="textBox1" VerticalAlignment="Top">
      Text her
    </TextBox>
    <Label Height="28" Margin="19,23,139,0"
      Name="label1" VerticalAlignment="Top">
      Skriv noe her:
    </Label>
    <ListBox Margin="19,78,85,0"
      Name="listBox1" Height="52" VerticalAlignment="Top">
      <ListBoxItem Name="item1" Content="Epikrise1"/>
      <ListBoxItem Name="item2" Content="Svar1" />
    </ListBox>
    <Button Height="23" HorizontalAlignment="Left"
      Margin="19,0,0,54" Name="button1"
      VerticalAlignment="Bottom" Width="75"
      Click="Clicked">
      Lagre
    </Button>
    <x:Code>
      void Clicked(object sender, RoutedEventArgs e)
      {
        MessageBox.Show("Lagring var vellykket!");
      }
    </x:Code>
  </Grid>
</Window>

```

```

</x:Code>
<RadioButton Height="16" Margin="19,0,139,109"
    Name="rblSykehus" VerticalAlignment="Bottom"
    GroupName="sykehusid">
    Hammerfest
</RadioButton>
<RadioButton Height="16"
    Margin="19,0,139,89" Name="radioButton1"
    VerticalAlignment="Bottom"
    GroupName="sykehusid">
    UNN
</RadioButton>
</Grid>
</Window>
]]>
</GUICode>
</SystemSpecificInformation>
</CommunicationParty>

```

4 Begreper

Begrep	Definisjon
Adresseregisteret	Et nasjonalt register som forvaltes av Norsk Helsenett og inneholder adresseopplysninger om alle potensielle avsendere og mottakere av informasjon i helsesektoren. Adresseregisteret skal betjene alle aktører i helsesektoren
Brukergrensesnitt	Brukergrensesnitt er et grensesnitt som sluttbruker benytter for å gjennomføre et brukstilfelle. Brukergrensesnitt er ofte ensbetydende med skjermbildet, men kan være andre annet f.eks. rattet i en bil
Brukstilfelle	Et brukstilfelle er en stegvis beskrivelse av hva systemet gjennomfører for å oppnå ønsket mål
Enhet	En enhet inngår som en formell del av en organisasjon
HER	HelseEnhetsRegister blir fortsatt brukt i forbindelse med prosjekter i spesialisthelsetjenesten som har som målsetting å ta i bruk Adresseregisteret
HPR register	Statens kontor for autorisasjon av helsepersonell autoriserer og tildeler HPR nummeret
FRESH	Felles Rekvirent og Enhetsregister for SpesialistHelsetjenesten. FRESH betjener alle helseforetak innenfor spesialisthelsetjenesten. FRESH betegner både applikasjonen og tilhørende database. FRESH vil fungere både for vedlikehold av OK2007 kodeverket, og for vedlikehold av kommunikasjonspartinformasjon og adresseinformasjon fra Adresseregisteret
Kommunikasjonspart	En avsender eller mottaker av informasjon i helsesektoren Kommunikasjonsparter får tildelt en nasjonal entydig identifikator – HERID
Organisasjon	En organisasjon er en samling av enheter
Person	En person i denne sammenheng er et menneske som er entydig identifiserbart og registrert som autorisert helsepersonell. Personer kan opptre som rekvirenter dersom de har nødvendige autorisasjoner. Denne avgrensningen er gjort for å sikre god datakvalitet gjennom en streng tilnærming til hvilke personer og opplysninger

	skal ligge i FRESH, samt påvente av Folkeregister som en nasjonal oppslagstjeneste for helsesektoren
Pseudokode	Kvasi-kildekode (ikke maskinlesbar) som angir strukturen i en utvalgt del av et større programsystem, gir føringer for hvordan en funksjon bør implementeres
Rekvirent	En rekvirent er enten en enhet eller person som har autorisasjon til å rekvirere informasjon, medisinsk behandling eller tjenester. Organisasjonsenheter og autorisert helsepersonell kan opptre som rekvirenter. En person er ikke en rekvirent før den er autorisert og har en identifiserbar relasjon til en enhet. En rekvirent er en side av en <i>kommunikasjonspart</i> og er den part som gjør en bestilling/henvisning/rekvisisjon. Kommunikasjonspart og rekvirent brukes om hverandre i dette dokumentet.
RiS diagram	Et RiS (Roller!Samspill) diagram tilsvarer et arbeidsflytdiagram
Web Service	En tjeneste i en tjenesteorientert arkitektur (SOA). En tjeneste består av en eller flere metoder (operasjoner)
Web Service metode	En metode er en funksjon som mottar en datastruktur, utfører noe programlogikk og returnerer en datastruktur som er spesifisert i datakontrakten mellom konsument og den som publiserer Web Service grensensittet. Metoden skal skjule de bakenforliggende prosesser som må gjennomføres for å tilfredsstille konsumentens behov. Konsumenten skal ikke trenge å vite noe om prosessene i løsningen(e) som tilbyr tjenesten.