



Universitetet
i Stavanger

DET TEKNISK-NATURVITENSKAPELIGE FAKULTET

MASTEROPPGAVE

Studieprogram/spesialisering: Datateknikk	Vår.....semesteret, 2009 Åpen
Forfatter: Børje Stangeland (signatur forfatter)
Faglig ansvarlig: Reggie Davidrajuh. Det teknisk-naturvitenskapelige fakultet, Universitetet i Stavanger. Veileder(e): Mari Hjelstuen. Klinikk for blod og kreftsykdommer, Stavanger Universitetssjukehus.	
Tittel på masteroppgaven: Bruk av webtjenester for automatisk prosessering av DICOM medisinske bilder. Engelsk tittel: Application of Web Services for Automated Processing of DICOM Medical Images.	
Studiepoeng: 30	
Emneord: Webtjenester, DICOM, Matlab	Sidetall: + vedlegg/annet: Stavanger, dato/år

Bruk av webtjenester for automatisk prosessering av DICOM medisinske bilder

Børje Stangeland
408257

Universitetet i Stavanger
Juni 2009

Erindringsvers

Jeg har skrevet et sted,
hvor jeg daglig maa se,
det manende tankesprog:
T.T.T.

Naar man føler hvor lidet
man naar med sin flid,
er det nyttigt at mindes, at
Ting Tar Tid.

Piet Hein
1905-1996

I. SAMMENDRAG

Forskningsarbeid på digitale radiologiske bilder er ofte svært tidkrevende, ettersom det i dag ikke eksisterer gode verktøy til å lete frem ønskede bildeserier fra undersøkelser som er lagret i de stadig økende bildelagringsystemene. Forskningsarbeid knyttet til eksisterende bilder har ofte som hensikt å prosessere disse bildene i et program som kan lese og foredle informasjonen som er tilgjengelig. Matlab er et slikt program, det støtter lesing og viderebehandling av medisinske bilder, men har ikke støtte for de tjenestene som behøves for DICOM nettverkskommunikasjon.

Denne oppgaven har hatt som hensikt å lage en integrasjonsplattform som ikke er fundert i eksisterende løsninger og implementeringer, men som eksponerer kommunikasjonstjenester innen DICOM i en mer tjenesteorientert arkitektur. Dette gjør at applikasjoner kan utvikles på en mer agil måte ved bruk av ny teknologi som er kjent for de fleste utviklere i dag. Utviklerne kan benytte det rammeverket som er utviklet for DICOM kommunikasjon og det rammeverket som gir integrasjon mot Matlab, uten å ha stor kjennskap til tradisjonell DICOM kommunikasjon eller til COM basert kommunikasjon hos Matlab.

I denne oppgaven er det utarbeidet flere webtjenester og programmer som er benyttet i testing av konseptet. All programmering i oppgaven er utført ved bruk av Visual Studio 2008, C#.net.

Det er programmert en egen applikasjon for DICOM tjenesten SCP, webtjeneste grensesnitt mot DICOM som eksponerer tjenestene *C-MOVE*, *C-STORE* og *C-FIND*. Webgrensesnittet mot DICOM tar høyde for å bli satt opp i et generisk miljø i et nettverk som i det minste har komponenter som genererer medisinske bilder i DICOM format, og som har tjenester for kommunikasjon av disse bildene. Videre er det utviklet et webgrensesnitt som oppretter forespørsler til og fra et COM basert grensesnitt mot Matlab R2007b. Utformingen av Matlabgrensesnittet gir også brukerne av dette grensesnittet en enkel måte å legge til egne funksjoner utviklet i Matlab, direkte inn i rammeverket som tilgjengeliggjør disse funksjonene.

Testapplikasjonen StudyCandidates som er utviklet i oppgaven har som hovedformål å bruke de integrasjonsplattformene som er gjort tilgjengelig til å forenkle prosessen med å hente ut radiologiske bilder, samt automatisk kunne gjøre disse bildene tilgjengelig for viderebehandling i Matlab. Det er laget en applikasjon som foretar en slik uthenting av bilder basert på kriterier som er satt, og som automatisk instruerer Matlab til å prosessere disse bildene. De funksjonene som er gjort tilgjengelig i Matlab er i denne oppgaven bare demonstrasjonsfiltre, men kan tenkes til å kunne være filtre som kan detektere artefakter eller unormaliteter i medisinske bilder.

Målingene av responstider viser at tjenesteorientert arkitektur legger en forsinkelse på presentasjon av DICOM trafikk. Forsinkelsen vil ikke representere en lang forsinkelse opplevd av brukeren. Fordelene med å bruke denne type teknologi er at den på sikt kan forenkle fremtidig utvikling av tilsvarende løsninger. Testapplikasjonen som er utviklet representerer en bruk av webtjenestene som gir brukeren en markant tidsbesparelse i å få tilgjengeliggjort medisinske bilder for å få testet sine egne filtre, og som i seg selv kan stimulere til utvidet forskning og bruk av Matlab i medisinsk forskning.

II. INNHOLDSFORTEGNELSE

I. SAMMENDRAG	4
II. INNHOLDSFORTEGNELSE	5
III. FIGURLISTE	7
IV. TABELLER	7
V. FORKORTELSER	8
VI. FORORD	9
1. INNLEDNING	10
1.1. Bakgrunn	10
1.2. Mål med oppgaven	11
2. TEORI	13
2.1. DICOM.....	13
2.1.1. DICOM Standard	13
2.1.2. DICOM Tag	14
2.1.3. DICOM Kommunikasjon.....	16
2.1.4. DICOM Tjenester.....	16
2.1.5. DICOM SOP	17
2.1.6. DICOM Aktører	17
2.2. Webgrensesnitt	19
2.2.1. Webtjenester.....	19
2.2.2. Tjenesteorientert arkitektur (Service Oriented Architecture).....	20
3. EKSPERIMENTELT	21
3.1. Testmiljø.....	21
3.2. Utviklingsmiljø.....	21
3.3. Systemdesign.....	22
3.4. Responstider	24
4. UTVIKLING.....	25
4.1. Beskrivelse av løsning.....	25
4.2. Erfaringer fra utviklingsprosessen.....	28
4.3. DICOM Webtjenester.....	29
4.3.1. WS_GetNrOfPatients	30
4.3.2. WS_GetPersonStudies	30
4.3.3. WS_GetStudyDetails	31
4.3.4. WS_GetSerieDetails.....	32
4.3.5. WS_RetrieveStudy	32
4.3.6. WS_RetrieveSerie	33
4.3.7. WS_RetrieveImage	33
4.3.8. WS_GetStudiesCandidates.....	33
4.3.9. WS_SVGDEMO2_WS_AvailableFilter	34
4.3.10. WS_SVGDEMO2_WS_MatlabImageJob	34
4.3.11. WS_SVGDEMO2_WS_MatlabSerieJob	34
4.3.12. WS_StoreSerie	34
4.4. MATLAB Webtjenester	35
4.4.1. WS_AvailableFilter.....	36
4.4.2. WS_FilterMfile	37
4.4.3. WS_MatlabImageJob.....	37

4.4.4.	WS_MatlabSerieJob.....	37
4.4.5.	WS_OutStoreSerie	37
4.5.	MatlabWrapper.....	38
4.6.	APPLICATION SCP.....	39
5.	TESTRESULTATER.....	40
5.1.	StudyCandidates	40
5.2.	Matlab.....	45
5.2.1.	Matlab Filter 1. EdgeDetectionWriteBack.....	45
5.2.2.	Matlab Filter 2. SineScrollDemo	46
5.2.3.	Matlab Filter 3. MR_VolumeRender	46
5.2.4.	Matlab Filter 4. PossibleArtefactSerie	47
5.2.5.	Matlab Filter 5. MyEdgeFilterDemo.....	48
5.2.6.	Matlab Filter 6. PossibleArtefact	48
5.3.	Responstider	48
5.3.1.	Responstider for henting av bilder	48
5.3.2.	Interne responstider i henting av DICOM informasjon over webtjenestene.....	49
5.3.3.	Interne responstider i MatlabWrapper.....	49
6.	DISKUSJON	50
7.	KONKLUSJON	53
8.	REFERANSER / KILDEHENVISNINGER	54
9.	VEDLEGG	56
9.1.	WSDL for DICOM Webtjenester.....	56
9.2.	WSDL for Matlab Webtjenester.....	69
9.3.	Kildekode Application SCP	75
9.3.1.	ApplicationSCP	75
9.3.2.	Program	77
9.3.3.	iniAPI	77
9.3.4.	SCP (ini-fil).....	78
9.3.5.	StorageSCP.....	78
9.4.	Kildekode DICOM WS	82
9.4.1.	DICOMCONNECT.....	82
9.4.2.	WEBSERVICES	88
9.5.	Kildekode Matlab WS	91
9.5.1.	Service	91
9.5.2.	MatlabFilters	93
9.6.	Kildekode MatlabWrapper	94
9.6.1.	Program	94
9.6.2.	MatlabWrapper (inifil)	94
9.6.3.	MatlabWrapperClass	95
9.6.4.	IniApi	100
9.7.	Kildekode StudyCandidates	101
9.8.	M-Filer.....	110
9.8.1.	MatlabFilter 1: EdgeDetectionWriteBack.....	110
9.8.2.	MatlabFilter 2: SineScrollDemo	112
9.8.3.	MatlabFilter 3: MR_VolumeRender	113
9.8.4.	MatlabFilter 4: PossibleArtefactSerie	115
9.8.5.	MatlabFilter 5: MyEdgeFilterDemo.....	115
9.8.6.	MatlabFilter 6: PossibleArtefact	116
9.8.7.	Helperfunksjon for logging: Time.....	117
9.9.	Plakat til demonstrasjon	118

III. FIGURLISTE

Figur 1. Prinsippskisse av DICOM nettverk	14
Figur 2. Utdrag av tabell som viser DICOM tags.	15
Figur 3. DICOM kommunikasjon for C-STORE.....	18
Figur 4. Prinsipp skisse over webtjeneste med serialisering	19
Figur 5. SOA klient og webtjenestegrensesnitt	20
Figur 6. Skjematisk fremstilling av komponenter som inngår i utviklet løsning.....	22
Figur 7. Tankekart som viser nøkkelementer som er viktige i en tenkt applikasjon.....	23
Figur 8. Rafinering av tankekart fra figur 7.	23
Figur 9. Overordnet betraktning av systemet..	25
Figur 10. Eksempel på prosessflyt i applikasjonen.	26
Figur 11. Enterprise Service Bus.....	27
Figur 12. Skjerm bilde som viser de Webtjenester som er tilgjengelig fra DICOM_WS.....	29
Figur 13. Resultatdokument fra WS_GetPersonStudies spørring.....	31
Figur 14. Resultatdokument fra WS_GetStudyDetails spørring.....	31
Figur 15. Resultatdokument fra WS_GetSerieDetails spørring..	32
Figur 16. Resultatdokument fra WS_RetrieveStudy spørring..	32
Figur 17. Resultatdokument fra WS_RetrieveSerie spørring.....	33
Figur 18. Resultatdokument fra WS_RetrieveImage spørring.....	33
Figur 19. Webtjenester fra Matlab_WS.	35
Figur 20. XML Dokument som viser tilgjengelige funksjoner.....	36
Figur 21. Konfigureringsparametre for MatlabWrapper.....	38
Figur 22. Konfigureringsparametre for ApplicationSCP.....	39
Figur 23. StudyCandidates oppstart.	40
Figur 24. StudyCandidates Søk.....	41
Figur 25. StudyCandidates valgt undersøkelse.	41
Figur 26. StudyCandidates funksjonalitet studynivå.....	42
Figur 27. StudyCandidates valgt serie.....	43
Figur 28. StudyCandidates seriefilter.....	43
Figur 29. StudyCandidates imagefilter.....	44
Figur 30. Viser i testpacs at ny serie er opprettet..	45
Figur 31. Panel A viser originalbilde fra MR. Panel B viser bildet etter kantdeteksjon.....	46
Figur 32. Panel A viser originale MR bilder. Panel B viser det genererte 3D volum	47
Figur 33. Utdrag av Result log for PossibleArtefactSerie.....	48
Figur 34. Fordeling av tider mot responstid for MatlabWrapper.....	49

IV. TABELLER

Tabell 1. Lagring av DICOM objekter ved Stavanger Universitetssjukehus.....	11
Tabell 2. Målt responstid mellom kommersiell SCP og ApplicationSCP	48
Tabell 3. Interne responstider for henting av DICOM informasjon over webtjenestene.....	49

V. FORKORTELSER

FORKORTEELSE	BETYDNING
AETitle	Application Entity Title
ARC/NEMA	American College of Radiology / National Electrical Manufacturers Association
C-MOVE, C-STORE, C-FIND	Complex Move, Complex Store, Complex Find
COM, DCOM	Component Object Model, Distributed COM
CORBA	Common Object Request Broker Architecture
CR	Computed Radiography
CT	Computed Tomography
DICOM	Digital Imaging and Communications in Medicine
DIMSE	DICOM Message Service Element
EPJ	Electronic Patient Journal
ESB	Enterprise Service Bus
HL7	Health Level 7
I/O	Input/Output
UID: Study, Serie, Image	Unique identifier
KITH	Kompetansesenter for IT i helse- og sosialsektoren
MG	Mammography
Modalitet	Apparat som skaper bilder
MR	Magnetic Resonance
MTOM	Message Transmission Optimization Mechanism
N-FIND, N-GET, N-STORE	Normalisert Find, Normalisert Get, Normalisert Store
NM	Nukleær Medisinsk
PACS	Picture Archive and Communication System
PAS	Pasientsystem
PID	Personnummer
QA	Quality Assurance
QRP	Query Retrieve Provider
QRU	Query Retrieve User
RIS	Radiology Information System
RSNA	Radiological Society of North America
SAS	Serial Attached SCSI(Small Computer System Interface)
SCP	Service Class Provider
SCU	Service Class User
SDK	Software Development Kit
SOA	Service Oriented Architecture
SOAP	Simple Object Access Protocol
SOP Class	Service Object Pair
SOP Instance UID	Dette SOP
SP	Service Pack
SUS	Stavanger Universitetssjukehus
UL, US	Ultralyd, UltraSound
VR	Value Representation
WSDL	Web Services Description Language
WSE	Web Security Enhancement
XML	Extensible Markup Language

VI. FORORD

Denne oppgaven ble utført våren 2009 ved Stavanger Universitetssykehus under veiledning av medisinsk fysiker Mari Hjelstuen. Tusen takk til Mari, som har bidratt med tilbakemeldinger på testprodukt og i forbindelse med skriving av rapporten.

Videre rettes en takk til Reggie Davidrajuh, Universitetet i Stavanger for de tilbakemeldinger som han har kommet med.

Jeg vil også takke Helse Vest IKT for utlån av testmiljø som har gjort det mulig å utvikle programmene i rapporten.

Til slutt må jeg også takke min kone for korrekturlesing, og for å legge forholdene til rette for at jeg kunne prioritere arbeidet med denne oppgaven.

Takk

1. INNLEDNING

1.1. Bakgrunn

22. desember 1895 klarte Wilhelm Conrad Röntgen å foreta første eksponering av røntgenstråler mot et platesystem som visualiserte sin kones håndflate. Dette var i realiteten verdens første røntgenbilde.

Radiografi er læren om å framskaffe informasjon om menneskekroppen gjennom egnet bildemateriale i den hensikt å oppdage eller behandle/lindre sykdom og lidelser [1].

Utviklingen i dag er at disse eksponeringene er videreført i flere metoder for å generere slike bilder som går under fellesbetegnelsen røntgenbilder. Røntgenbilder blir i dag i hovedsak fremskaffet med digitale sensorer som detekterer utslag på en detektor. Maskinene og metodene for å gjøre dette er blant annet CT og CR for å nevne noen. Røntgenbildene blir som oftest lagret i et lagringssystem for radiologiske bilder som har betegnelsen PACS. I dette lagringssystemet lagres også digitale bilder fra annet medisinsk utstyr som ultralyd (UL) og MR.

Kompetansesenter for IT i helse- og sosialsektoren, KITH gjengir definisjonen for PACS som: *PACS står for "Picture Archive and Communication System" og er en betegnelse som brukes for digital røntgen. I et PACS-system blir røntgenbildene generert, lagret og distribuert digitalt og all behov for film og fremkalling er borte. PACS er blitt en nødvendighet for å kunne behandle, lagre og distribuere den store datamengden som dagens moderne røntgenutstyr genererer [2].* En presisering av denne formuleringen til KITH burde vært å omfatte også annet diagnostisk utstyr til bruk innen PACS og DICOM.

I Norge økte antall undersøkelser fra 641 pr tusen innbyggere i 1983 til 710 pr tusen innbyggere i 1993 og til 780 pr tusen innbygger i 2003. Denne sterke veksten ser ut til å fortsette. Ved siste års RSNA i USA (RSNA 2004) ble det lagt fram beregninger på at røntgen vil øke med over 200 % i de neste 10 årene [2].

Introduksjonen av flere typer digitale diagnostiske apparater har resultert i en eksponentiell vekst i antall mulige undersøkelser en kan gjøre, og antall bilder for hver undersøkelse har steget drastisk. Dette har blant annet sammenheng med mulighetene som digitalisering av radiologiske røntgenbilder har gitt. Modaliteter har i dag mulighet til å generere langt flere bilder innen en gitt tidsramme enn hva deres analoge ekvivalenter var i stand til å gjøre for bare noen år siden. Dette har gjort utslag i for eksempel CT- og MR-undersøkelser der det har blitt vanlig å utføre undersøkelser med langt flere snitt/ bilder for hver del av organet, som dermed har resultert i en massiv vekst i antall bilder pr serie og undersøkelse.

Tabell 1 viser økningen i antall bilder pr undersøkelse ved Stavanger Universitetssjukehus siden mai 2007 og frem til i dag. Viktig å merke seg at det i denne perioden ikke har vært anskaffelser av flere modaliteter, men at det er anskaffet noe nytt diagnostisk utstyr som erstatning for utrangert utstyr som gir disse utslagene.

Tabell 1. Lagring av DICOM objekter ved Stavanger Universitetssjuehus i perioden mai 2007 til mars 2009. Antall bilder pr undersøkelse (ratio Bilder/US) har i denne perioden økt fra ca 30 bilder pr undersøkelse til nærmere 50 bilder pr undersøkelse.

	Antall undersøkelser	Antall bilder	Ratio Bilder/US
mai.07	15341	475609	31
jun.07	2577	86192	33
jul.07	311	3899	13
aug.07	34599	1130167	33
sep.07	16464	498211	30
okt.07	16490	495702	30
nov.07	12069	392405	33
des.07	16324	510292	31
jan.08	10153	329723	32
feb.08	4982	165023	33
mar.08	16522	599226	36
apr.08	15672	592659	38
mai.08	19047	599106	31
jun.08	1469	87313	59
jul.08	35017	1231406	35
aug.08	16569	577913	35
sep.08	16296	648926	40
okt.08	13750	558314	41
nov.08	13402	537647	40
des.08	15103	667586	44
jan.09	13129	671341	51
feb.09	15371	775418	50
mar.09	13233	638910	48

Tabell 1 viser at det i gjennomsnitt ble eksponert cirka 30 bilder for hver undersøkelse i 2007, mens det i 2009 har steget til cirka 50 bilder i snitt. I juni/juli 2007, og i juni 2008 ble det foretatt oppgraderinger av lagringssystemet til Stavanger Universitetssjuehus. Dette fører til to uvanlig innslag i tabellen, da oppsamlet informasjon ble lagret i påfølgende måned. Dersom en normaliserer produksjon over foregående og etterfølgende måned vil en kunne få en riktigere vurdering av tabellen.

1.2. Mål med oppgaven

Denne oppgaven tar utgangspunkt i tanker fra forskningsmiljøet ved Stavanger Universitetssjuehus. Det er blitt observert at medisinsk personell, som radiologer og fysikere, som ønsker å gjøre et uttrekk av spesifikke serier i eksisterende bildemateriale fra bildelagringssystem, har en tidkrevende jobb med å ekstrahere ut disse bildene til bruk i forskning, kvalitetskontroll eller kvalitetsstudier.

Denne oppgaven har som mål å forenkle prosessen med å hente ut radiologiske bilder fra PACS, samt å se på muligheten for automatisk eksponering av disse bildene til videreprosessering i Matlab. Målsetningen med denne rapporten har vært å se på følgende problemstillinger:

- Vil det være mulig å løsrive DICOM kommunikasjon fra sin tradisjonelle punkt-til-punkt konfigurasjon inn i en mer tjenesteorientert arkitektur med de egenskapene det medbringer?
- Vil det være mulig å skape et eget tjenesteorientert grensesnitt mot et noe eldre COM grensesnitt representert i denne oppgaven som Matlab?
- Vil det ved å bruke en slik tjenesteorientert tankegang og implementering være mulig å skape en fornuftig integrasjon og automasjon for å løse en spesifikk utfordring?
- Vil summen av disse resultatene belyse om det på sikt kan være mulig å løsrive seg fra den konvensjonelle kommunikasjonsplattformen som DICOM har i industrien i dag, til å tilby mer tjenesteorienterte løsninger uten å måtte omdefinere en hel industri?

Oppgaven viser en tilnærming til løsning av disse problemstillingene i lys av en tjenesteorientert arkitektur. Det er forsøkt å skape integrasjonsplattformer som benytter konvensjonelle DICOM tjenester, men som gir disse tjenestene et skall av en tjenesteorientert tilnærming. Videre viser rapporten resultater fra målinger som sier noe om ytelsen til disse skallene.

2. TEORI

2.1. *DICOM*

Digital Imaging and Communications in Medicine, DICOM er en utbredt standard når det gjelder overføring av medisinske bilder mellom forskjellige aktører. I dag støtter de fleste leverandører denne standarden som nå er kommet til versjon 3. Eldre installasjoner av utstyr som ikke har støtte for en slik kommunikasjon oppgraderes ofte til å støtte det, eller det blir koplet opp maskiner som proprietært henter bilder og indirekte gir utstyret DICOM 3.0 kommunikasjon. Utstyr som driver med bildefangst (framegrabbers) blir mindre og mindre vanlig ettersom nytt utstyr har den nødvendige støtte for DICOM. Modaliteter som har DICOM 3.0 er konvensjonell røntgen apparatur innen radiologi som CT og CR, men også andre teknologier som MR og NM, samt annen diagnostisk apparatur innen blant annet orthomologi (øyebilder) og dental (tannhelse).

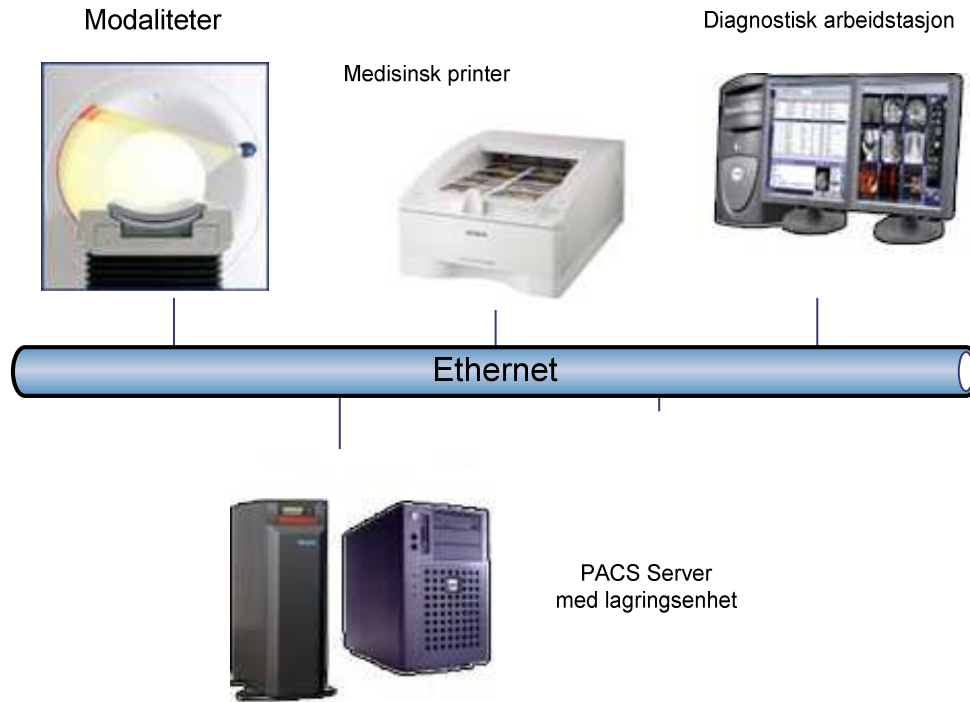
Følgende beskrivelse av DICOM med årstall er inspirert av informasjonsside hos RSNA [3].

2.1.1. *DICOM Standard*

DICOM er industristandard for overføring av medisinske bilder og annen medisinsk informasjon mellom datamaskiner i et medisinsk nettverk. DICOM standarden muliggjør kommunikasjon på tvers av leverandørskiller. Utveksling av medisinsk informasjon er nødvendig for samhandling.

DICOM er en standard som har utviklet seg i flere steg. Forgjengeren var ARC/NEMA som ofte blir referert som DICOM 1.0. Dette var et eksperiment mellom American College of Radiology og National Electrical Manufacturers Association, som klarte å få medisinsk apparatur til å utveksle bilder. De første versjonene av denne standarden ble presentert i henholdsvis 1985 og i 1988, før gjeldende standard, referert til som DICOM 3.0, ble etablert i 1994. Det har etter denne tid blitt gitt ut flere supplement til denne standarden.

Resultatet av dette arbeidet er at det i dag er mulig for utstyr som støtter DICOM 3.0 standarden å samhandle i utvekslingen av bilder. Dette kan blant annet være modaliteter som CT, MR og CR apparater, arbeidsstasjoner som vurderer disse bildene, arkiveringssystem som lagrer og videreformidler disse bildene, samt konvensjonelle filmskrivere som kan skrive ut røntgenfilm på et analogt format.



Figur 1. Prinsippskisse av DICOM nettverk. I figuren er det lagt til SCP og SCU noder representert som PACS lagringssystem, inn-enhet som modalitet, ut-enhet som medisinsk skriver og diagnostisk arbeidstasjon. Dette er de primære enhetene som inngår i et standard DICOM nettverk, og som oftest kommuniserer på et vanlig nettverk (Ethernet)

2.1.2. DICOM Tag

Som en følge av DICOM 3.0 standarden blir hvert enkelt bilde som blir generert fra modaliteter merket med en egen referansetabell av identifikatorer, *DICOM tags*. Disse tags gir eksplisitt informasjon om dette spesifikke bilde. Dette er blant annet informasjon om pasienten, arbeidsflytinformasjon som identifiserer undersøkelsen, og ikke minst hvordan dette bildet er generert. Modalitetsparametre, undersøkelsesparametre og annen vesentlig informasjon om undersøkelsen er representert i denne tabellen. Figur 2 viser et utdrag av informasjon fra en slik referansetabell.

Group - Element	Description	Type	Length	Value
0008 0016	SOP Class UID	UI	30	1.2.840.10008.5.1.4.1.1.1.2.1
0008 0018	SOP Instance UID	UI	50	1.2.840.113619.2.66.2218493546.870
0008 0020	Study Date	DA	10	20090508
0008 0021	Series Date	DA	10	20090508
0008 0022	Acquisition Date	DA	10	20090508
0008 0023	Content Date	DA	10	20090508
0008 0030	Study Time	TM	14	072746.000000
0008 0031	Series Time	TM	14	072815.000000
0008 0032	Acquisition Time	TM	14	072808.000000
0008 0033	Content Time	TM	14	072828.000000
0008 0050	Accession Number	SH	2	
0008 0060	Modality	CS	4	MG
0008 0064	Conversion Type	CS	4	WSD
0008 0068	Presentation Intent Type	CS	16	FOR PROCESSING
0008 0070	Manufacturer	LO	20	GE MEDICAL SYSTEMS

Figur 2. Utdrag av tabell som viser DICOM tags. Kolonnen til venstre viser gruppe-element organiseringen som sier hva verdien representerer. I neste kolonne vises en tekstlig forklaring og forståelse av gruppe-elementet. I Type kolonnen angis hvilken datatype som er representert, blant annet om det er et TM (TimeMinute), DA(Date), UI (unik identifikator) eller et fritekstfelt (SH). Nest siste kolonne viser hvor mange karakterer som kan benyttes i verdifeltet. Lengst til høyre vises den aktuelle verdien som er for dette gruppe-elementet.

Tabellen i figur 2 viser bare et lite utdrag av parametre som er tilgjengelig for hvert enkelt bilde. Over 1800 registrerte *DICOM tags* er tilgjengelig for benyttelse.

Som en ser av figur 2, er DICOM informasjonen gruppert etter type. Gruppe 0008 inneholder undersøkelsesspesifikk informasjon. Til sammenligning inneholder gruppe 0010 pasientrelaterte identifikatorer. Hver gruppe har et sett med element som identifiserer identifikatoren. Et eksempel på dette kan være gruppe 0008, element 0060 som beskriver hvilken type modalitet som har generert bildet. Av figur 2 ser man at dette bilde er av typen mammografibilde (MG).

For selve organiseringen av bildet, og hvordan dette lagres i et PACS system er det noen identifikatorer som er nødt til å være global unike. Disse har betegnelsen UID. I figur 2 kan man se UID for DICOM tag (0008,0018) som er SOP Instance UID. Dette er en global identifikator som identifiserer akkurat dette bilde. Andre tager som er av spesiell interesse for denne rapporten er:

(0020,000D) StudyInstanceUID som er undersøkelsesidentifikator.

(0020,000E) SerieInstanceUID som er serieidentifikator.

(0008,103E) Series Description, navnet på bildeserien.

Referansetabellen finner man i begynnelsen av hver enkelt DICOM fil (*.dcm). Denne første delen av den fysiske DICOM filen refereres ofte til som DICOM Header. Resterende del, DICOM Body er den faktiske bildedelen av bildet. Det er også mulig å benytte såkalte private elementer, Private Tags. Dette er Gruppe-Element som ikke er offisielt godkjent, men som er leverandør spesifikke og benyttes ofte for proprietær informasjon.

2.1.3. DICOM Kommunikasjon

DICOM kommunikasjon er typisk punkt til punkt kommunikasjon. En aktør som skal sende et bilde må åpne en kommunikasjonskanal definert med IP adresse og nettverksport. Videre er det en identifisering av aktørene ved bruk av en AETitle. Dette steget i DICOM kommunikasjon refereres ofte til som en *DICOM Negotiation*. Dette er en avklaring på at aktørene A og B har en felles kopling å kommunisere på.

Når dette fysiske laget er på plass skal aktørene gjensidig utveksle hvilke tjenester de tilbyr og utfører. Denne prosessen kalles i DICOM terminologi en *DICOM assosiering*. En viktig avklaring som foretas i denne assosieringen er hvordan aktørene betrakter bilder større en 8-bit. Dersom det benyttes 16-bit piksel verdier, må det avklares hvor signifikant den første byte skal være. Dersom det første byte skal være minst signifikant går dette under betegnelsen *Little Endian*. I konfigurasjon der den minst signifikante byte sendes sist, benytter DICOM terminologien *Big Endian*. Det er viktig å ha rett konfigurering mellom aktørene slik at de gjensidig kan tolke og gjengi innholdet i bildene på en korrekt måte, slik at gråskalarepresentasjonen av bildene bli korrekt.

En annen viktig avklaring som foretas i assosieringen er hvordan aktørene refererer til DICOM tag beskrivelsen. *Implicit VR* betyr at avsender benytter en egen referansetabell (*DataDictionary*) for oppslag av dataelementene og at beskrivelsen av DICOM element ikke blir sendt med i meldingen. Dersom avsender sender med *Explicit VR* betyr det at mottaker selv må lese i beskrivelsen av dataelementet som er inkludert i meldingen.

Resultatet av konfigureringen av *Endian* og *DataDictionary* refereres i DICOM terminologi til som *Transfer Syntax*.

Da er i prinsippet de nødvendige avklaringer som må til ferdigstilt og aktørene kan starte overføringen av bilder mellom seg. Bildene blir ofte overført en og en, der mottaker sender en *acknowledge*-melding mellom hver bildeforsendelse. Dette er en melding som informerer avsender om at bildene er mottatt. En stor ulempe med DICOM kommunikasjon er at denne *acknowledge*-meldingen fra applikasjonslaget er relativ tidkrevende, og foretas for hver enkel bildeoverføring. I den senere tid er det også blitt vanlig å utføre en *DICOM Storage Commit* melding når undersøkelsen er ferdig overført. Dette er en melding fra mottaker til avsender med informasjon om hvor mange bilder som er overført, og gir en kvalitetssikring på at alle avsendte bilder er mottatt.

2.1.4. DICOM Tjenester

Som nevnt tidligere foretas det en avklaring i DICOM assosiering av hvilke tjenester som de aktuelle aktørene har. På denne måten blir det klart hvilke forespørsler og bilder som kan komme til å oppstå.

Disse tjenestene er av to forskjellige kategorier, normaliserte (N) og komplekse (C) tjenester. Komplekse tjenester blir også referert til som *Composite*. Hovedgrunnen til at det finnes to forskjellige tjenestegrupper er av historisk art. Normaliserte tjenester er compatible med ARC/NEMA standard for bakoverkompatibilitet. En annen viktig forskjell er at komplekse tjenester har identifikatorer som pakker bildene inn i kolleksjoner av en undersøkelse, slik at disse igjen kan referere til felles pasientidentifikatorer. For å belyse dette nærmere kan det

nevnes at CR undersøkelser er av normalisert type. Det vil si at pasientidentifikatorer er inkludert i hvert enkelt bilde. I en kompleks undersøkelse som for eksempel MR undersøkelser, vil pasientidentifikatorene referere til et felles objekt, og vil ikke være fullstendig inkludert i alle bildeobjektene.

I DICOM terminologi refereres det til tjenestene med benevnelsen *DIMSE*, og som en følge av tjenestegrupperingen av normaliserte og komplekse tjenester vil det i mange tilfeller være to ekvivalenter av tjenestene. Dette gjør seg for eksempel gjeldene i lagringstjenesten. Dette blir i DICOM referert til som *DIMSE N-STORE* og *DIMSE C-STORE*. Tilsvarende for andre basis tjenester i DICOM har vi blant annet *C-FIND*, *N-FIND*, *C-STORE*, *N-GET* og *C-MOVE*.

2.1.5. DICOM SOP

SOP klasse er en identifikator som informerer aktørene om hvilken type objekt dette er. Med andre ord vil en SOP klasse beskrive for aktørene hvilken type bilder dette faktisk er, som for eksempel et 8-bit piksels CT bilde. Figur 2 viser gruppe-element (0008x0016) *SOP Class UID* som 1.2.840.10008.5.1.4.1.1.1.2.1. Dette er en unik identifikator for en gitt type mammografibilder. Gruppe-element (0008x0060) viser også at dette er et mammografibilde (MG).

Det er også mulig å referere til en *SOP Instance* som da vil være ett bestemt bilde av denne typen. Alle varianter av medisinske bilder kan typespesifiseres ut i fra identifikatoren som SOP klassen har. Disse SOP klassene utveksles i *DICOM Associate* prosessen slik at aktøren vet hvilke type objekter en kan forvente vil bli sendt.

2.1.6. DICOM Aktører

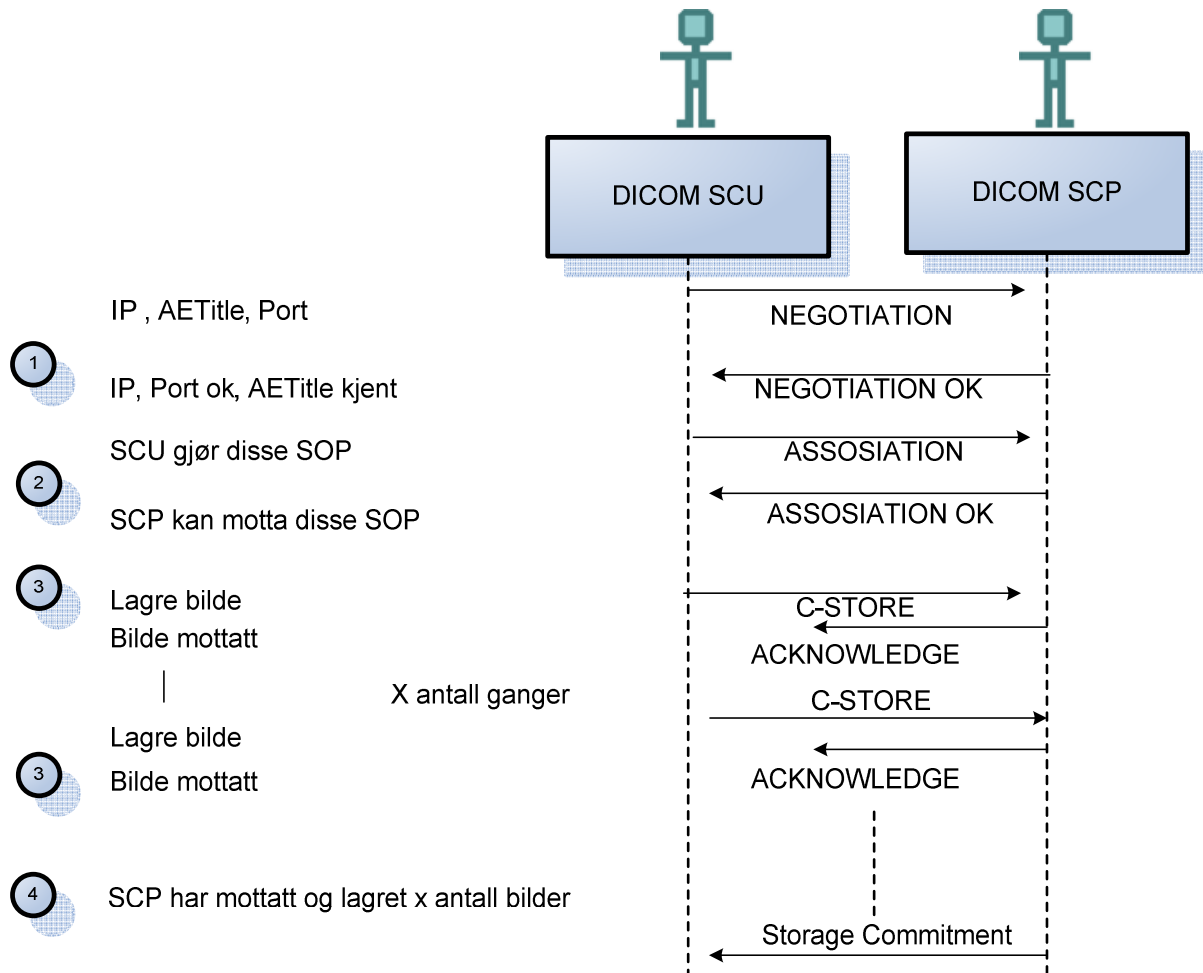
I prosessen med å etablere en DICOM kommunikasjon har vi som nevnt to aktører som skal utveksle bilder. Aktørens rolle bestemmes av hvilken vei DICOM bildeoverføring skal gå.

Noen DICOM aktører har alltid samme rolle, mens andre kan innta forskjellige roller alt etter hvilken kommunikasjon som skal foretas. I DICOM har de forskjellige rollene følgende benevnelser:

- **DICOM SCP** (Service Class Provider). En aktør som i øyeblikket tilbyr å lagre informasjon for andre. Med andre ord en aktør som kan motta bilder som blir sendt til han.
- **DICOM SCU** (Service Class User). Aktør som i øyeblikket har bilder som skal sendes til en annen aktør.
- **DICOM QRU** (Query Retrieve User). En aktør som kan åpne en kommunikasjon mot en annen som har bilder tilgjengelig, og som kan hente disse bildene til seg.
- **DICOM QRP** (Query Retrieve Provider). Aktør som har bilder og som kan svare på forespørsler, og som eventuelt kan sende dem til en annen aktør.

Aktørene har ofte en samling av elementære tjenester som behøves for å utøve rollen. Rollen *QRU-N* vil måtte utøve de elementære tjenestene *N-FIND* og *N-GET*, mens *QRU-C* vil utøve tjenesten *C-FIND* og *C-MOVE*.

Det finnes enda flere roller innen DICOM som er lagt til DICOM 3.0 som et supplement. Ytterligere informasjon om disse rollene kan blant annet leses på Wikipedia [4].



Figur 3. DICOM kommunikasjon for C-STORE. Punkt 1 viser til DICOM Negotiation hvor aktørene gjør avklaring og oppretter en fysisk kommunikasjonskanal. Punkt 2 overfører den SOP som aktørene kan gjøre. I punkt 3 starter selve bildeoverføringen. Dette kan fortsette x ganger. Når bildene er ferdig overført vil en storage commit melding bli overført. Storage Commit meldingen angir antall bilder som er verifisert lagret hos mottaker.

Figur 3 viser en systematisk fremstilling av den DICOM 3.0 kommunikasjonen som foretas av aktørene i en DICOM *C-STORE*. Punkt 1 er *DICOM Negotiation*. Ved hjelp av denne meldingen avklares og opprettes en kommunikasjonskanal mellom aktørene. I punkt 2 sendes meldinger som foretar en assosiering som identifiserer de forskjellige SOP klassene som henholdsvis SCP og SCU støtter. I punkt 3 starter bildeoverføringen. For hvert bilde som blir lagret kommer en *acknowledge* melding som sier at bildet er mottatt. Når en *timeout* hos mottaker inntreffer i punkt 5 sendes en *Storage Commitment* melding som informerer om hvor mange bilder som er lagret hos mottaker.

2.2. Webgrensesnitt

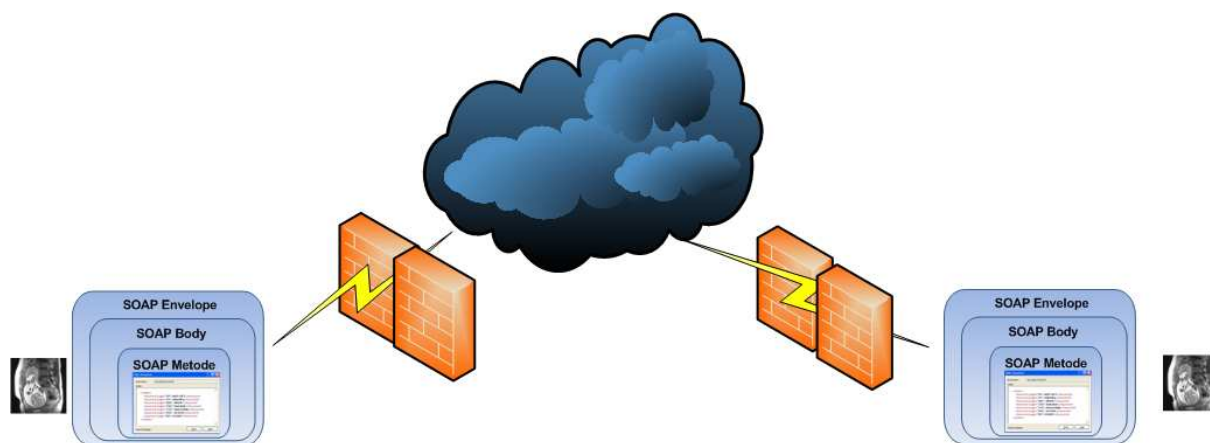
2.2.1. Webtjenester

Webtjenester er i dag blitt en industristandard for å utvikle programmer med tjenesteorientert arkitektur, mye på grunn av egenskapen til å tilby interoperabilitet mellom programmer som blir utviklet på forskjellige plattformer.

Grensesnittene for webtjenester blir ofte betraktet som en plattform for å eksponere intern applikasjonslogikk i fra en applikasjon til aktører som måtte ha nytte av funksjonaliteten som blir gjort tilgjengelig. En klar fordel som webtjenester har, er at det er relativt sett enkelt å utvide eksisterende applikasjoner, og dermed holde utviklingskostnadene lave. Webtjenester muliggjør eksponering av bare de delene av en applikasjonslogikk som en ønsker, og som strengt tatt er nødvendig for eksterne aktører å forholde seg til.

En annen fordel er at det kreves lite utvikling fra eksterne aktører for å benytte seg av nye webtjenester som blir gjort tilgjengelig. De fleste utviklingsverktøy og plattformer har i dag god støtte for å konsumere denne type tjenester.

En av ulempene med å benytte webtjenester er at de strengt tatt ikke er objektorientert, men denne effekten motvirkes ofte ved at man serialiserer objektene i meldingene som benyttes. Objektene, og deres egenskaper blir ekstrahert på en strukturert måte, for så å bli bygd opp igjen av mottaker av meldingen. En annen ulempe med å benytte webtjenester er at nettopp denne serialiseringen fører til forsinkelse i overføring av informasjon. Figur 4 viser en skjematisk fremstilling av webtjenester.

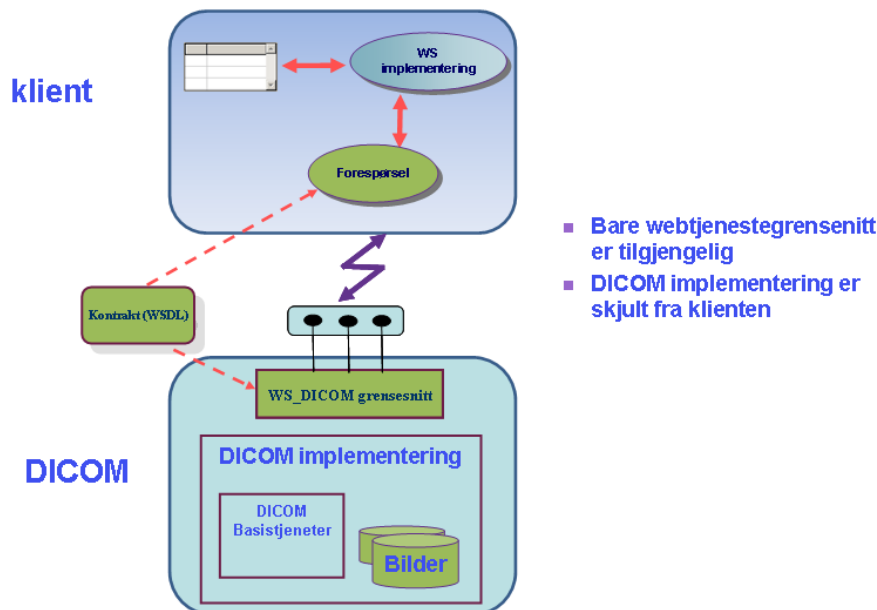


Figur 4. Prinsipp skisse over webtjeneste med serialisering. På venstre side sees et bilde som logisk blir delt opp i en tekstlig representasjon innen SOAP metoden. Videre legges en SOAP body logisk på objektet for å eventuelt kunne skille dette bilde fra andre som overføres. Som en siste innpakning legges innholdet i en SOAP Envelope som identifiserer mottaker av denne informasjonspakken. Denne informasjonspakken kan sendes gjennom en organisasjons brannmur om ønskelig. Dette er representert på skissen som brannmurene med lyn gjennom. Skyen i skissen representerer en intra-/internet kopling mellom aktørene. Den motsatte logikken gjøres på mottakersiden. Her mottas informasjonspakken over internet, for så å bli gitt til korrekt mottaker i henhold til SOAP envelope. Videre ekstraheres type-informasjonen ut av SOAP Body og metode for å kunne bygge bildet opp i sin opprinnelige form.

2.2.2. Tjenesteorientert arkitektur (Service Oriented Architecture)

Tjenesteorientert arkitektur (SOA) blir ofte definert som en samling av tjenester, og er i så måte ikke bundet opp mot webtjenester. Konseptet er benyttet tidligere, både i fra DCOM og fra CORBA, men har fått sin renessanse under webtjenester mye på grunn av webtjenestenes interoperabilitet.

Selve definisjonen på tjenesteorientert arkitektur er at aktører som har eller behøver en tjeneste utført, samhandler ved bruk av meldinger etter et definert mønster. Figur 5 viser en fremstilling av en SOA tankegang.



Figur 5. SOA klient og webtjenestegrensesnitt. På klientsiden kan en applikasjon motta en beskrivelse (WSDL) av webtjeneste hos en annen aktør som identifiserer tjenesten og bruk av denne. Når dette er kjent kan applikasjonen benytte seg av dette til bruk i egen applikasjon. På serversiden, her representert som DICOM, kan aktøren som tilbyr tjenestene maskere sin interne applikasjonslogikk og tilby et forenklet grensesnitt som ivaretar inn- og ut-parametrene for tjenesten. (Inspirasjon til figur er lånt fra Web Services: Principles and design, Papazoglou [5], kapittel 8.)

Den største fordelen med tjenesteorientert arkitektur er at programmene kan deles opp i mindre komplekse enheter som enklere lar seg håndtere og videreutvikle. Denne moduleringen gjør at tjenestene som representerer en verdi for andre kan benyttes direkte ved å sende meldinger til disse tjenestene, eller ved å abonnere på meldinger fra en tjeneste som kan gi mottakeren informasjon.

Når flere aktører er involvert i et nettverk som gjensidig tilbyr og konsumerer tjenester hos hverandre, refereres det ofte til begrepet Enterprise Service Bus (ESB). Dette er en logisk fremstilling som definerer et felles rammeverk som aktørene skal eksponere tjenestene på. Det finnes flere måter for de forskjellige aktørene å kommunisere på en slik ESB. Noen klarer å kommunisere direkte på ESB, mens det vanligste er å kommunisere ved hjelp av adaptere som gir basistjenestene et skall som simulerer et nødvendig format. Disse adapterne samles ofte i en *gateway*, som refereres til som en *Broker*.

3. EKSPERIMENTELT

3.1. Testmiljø

Testmiljøet som ble benyttet er i sin helhet installert i et virtuelt miljø som er en del av nettverket i Helse Vest. Maskinene som er benyttet i oppgaven er virtuelle servere som er instanser av en av Helse Vest sine VMWare servere.

Spesifikasjonen på denne serveren er som følger:

- DELL PowerEdge R900
- 4 x Quad Core Intel XEON X7350 cpu @ 2,93GHZ (Hyperthreading disabled)
- 64 GB RAM (16x4GB Dual Rank DIMMS) 667MHz FBD
- 2 x 73GB SAS 15K 2,5'' HD Hot Plug (Speilet)

De virtuelle test serverene svg-demo1, svg-demo2 og vir-app901 fikk tildelt følgende spesifikasjoner.

- 1CPU @ 2925 Mhz
- 1024 MB RAM
- 1 Gbit Nettverkskort
- OS: Windows 2003 Server, SP2
- IIS 6.0 installert

På denne måten var det enkelt å sette opp et miljø som la til rette for en kommunikasjon med flere servere innen testmiljøet, og som også hadde muligheter for kommunikasjon med produksjonsmiljø for enkelt å hente ut medisinske bilder.

3.2. Utviklingsmiljø

På disse serverene som også er testservere for systemet ble Visual Studio 2008 c#.net SP2 installert på maskinene svg-demo1 og svg-demo2. Her ble utviklingsarbeidet foretatt, slik at det ble mulig å gå stegvis igjennom kildekoden.

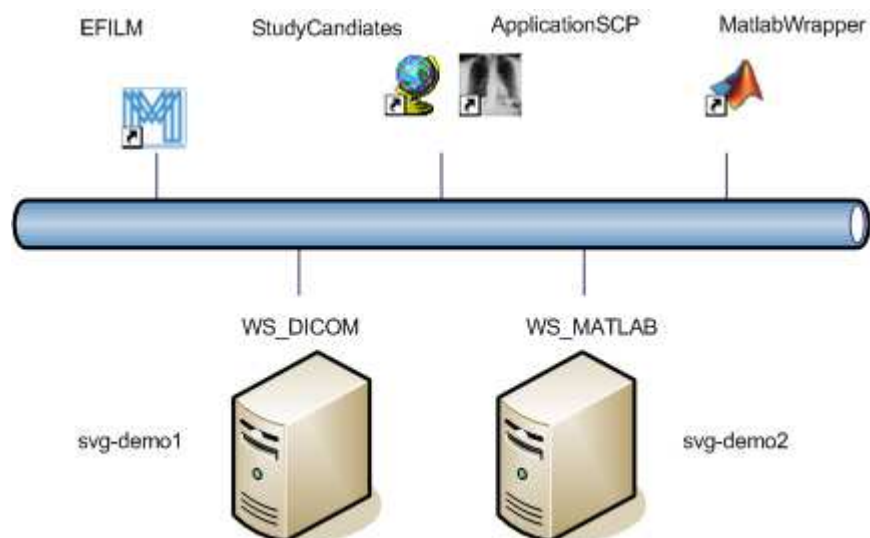
På maskinen svg-demo1 ble det lastet ned og installert en SDK fra ClearCanvas ®. SDK til ClearCanvas er en fleksibel .NET basert utviklingsplattform tilrettelagt for utvikling av DICOM i Visual Studio 2005. Til tross for at utviklingen i denne oppgaven ble utført ved bruk av Microsoft Visual Studio 2008, ble det ikke identifisert noen utfordringer ved bruk av klassene for basis DICOM trafikk. I begynnelsen av utviklingsarbeidet ble det benyttet en åpen kildekode med navn OpenDICOM-sharp, utviklet av Albert Gnandt.

EFILM Workstation 310T21 ble installert på serveren vir-app901 som et testpacs.

Matlab R2007b ble installert på serveren svg-demo2 for å utføre bildebehandlingen som systemet benytter.

Matlab ble installert på vanlig måte og det er ingen modifiseringer av Matlab miljøet. Dette inkluderer standard tilgjengelige pakker fra Matlab. Det er i oppgaven ikke lastet ned spesielle

verktøykasser (toolbox). Inspirasjonen til integrasjonen er i stor grad hentet fra arbeidet til Emanuele Ruffaldi, 1..2..3 ways of integrating MATLAB with the .NET [6].



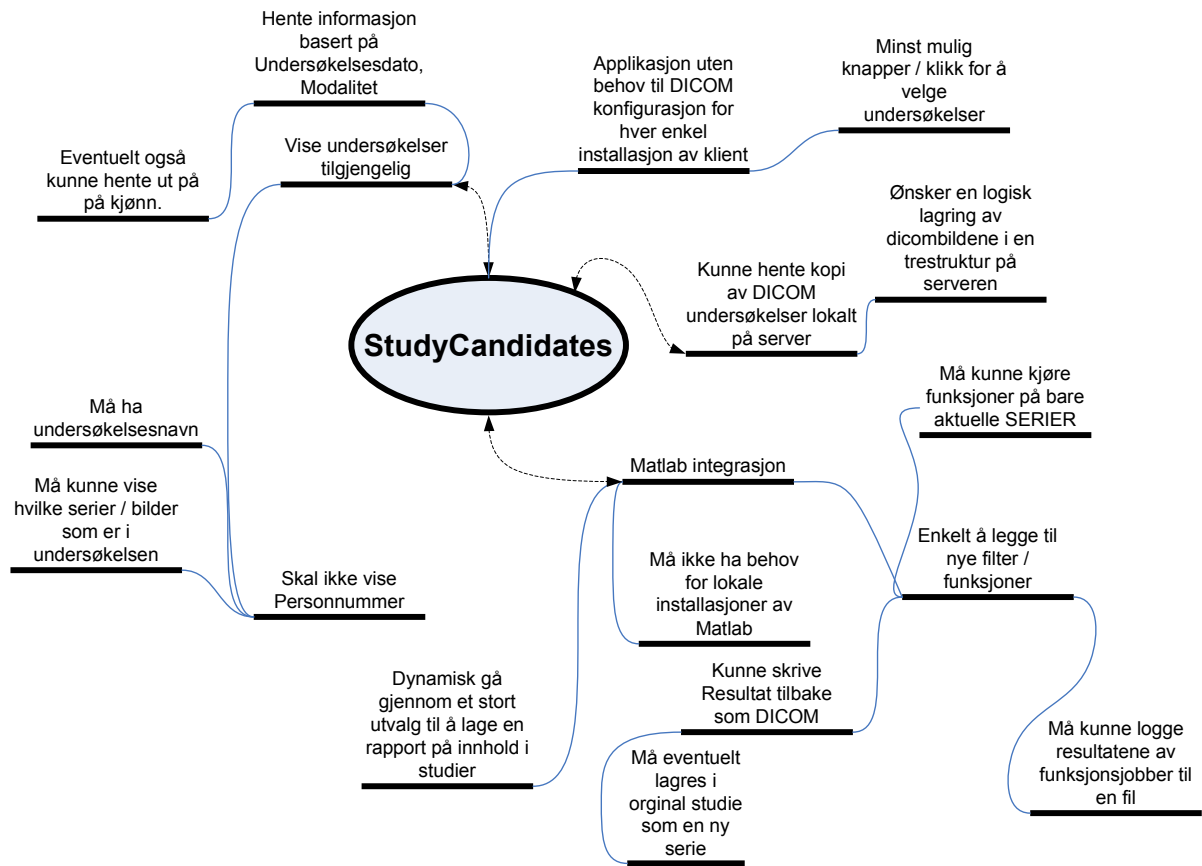
Figur 6. Skjematisk fremstilling av komponenter som inngår i utviklet løsning. Øvre del av figuren viser de applikasjonene som er tilgjengelig i løsningen. EFILM er et kommersielt produkt som i denne oppgaven representerer et PACS lagringssystem. StudyCandidates er utviklet testapplikasjon i oppgaven som benytter seg av de grensesnittene som er utviklet for å illustrere en mulig anvendelse av disse. ApplicationSCP er den modifiserte SCP klientapplikasjonen som mottar bilder for systemet. MatlabWrapper er den utviklede løsningen for å kople webtjenestegrensesnittet sammen med COM grensesnittet til Matlab. Den blå linjen representerer en tenkt intra-/internet kopling mellom aktørene. Nedre bit av figurene viser de to serverene som er kjerner i løsningen. WS_DICOM har webtjenestegrensesnittet og WS_Matlab har grensesnittet mot serveren som har Matlab tilgjengelig.

Figur 6 viser elementene som inngår i designet av løsningen og testmiljøet. Tanken er at en applikasjon tilsvarende StudyCandidates kan benytte seg av de webtjenestegrensesnittene som er tilgjengelig, sammen med støtteapplikasjonene som ApplicationSCP og MatlabWrapper. På denne måten kan det tilbys en forenklet og agil tilnærming til utviklere som ønsker å benytte seg av medisinske bilder i sin applikasjon, som også kan drive med postprosessering.

3.3. Systemdesign

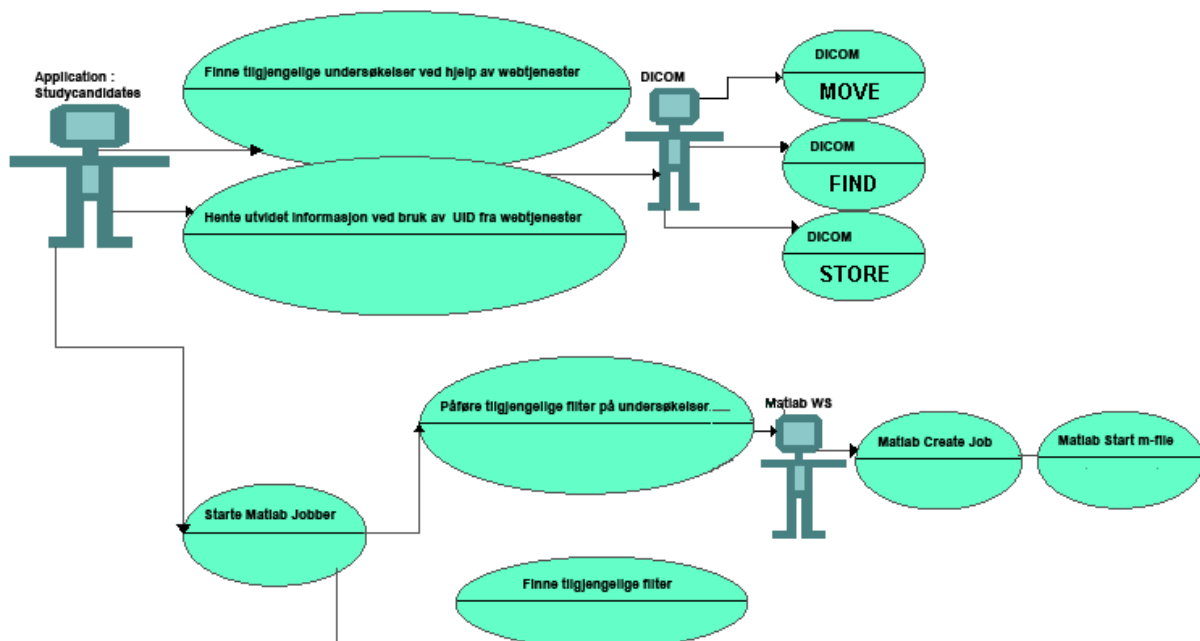
For å få en bedre forståelse av de utfordringer som er i en diagnostisk radiologisk avdeling har det blitt foretatt en del samtaler med personell som i sitt daglige virke benytter DICOM 3.0 og PACS. Dette har vært både radiologer, radiografer, fysikere og PACS administratorer fra ulike organisatoriske enheter i Helse Vest.

De tilbakemeldinger som er mottatt fra personell ved Stavanger Universitetssjukehus, og det tankearbeidet som disse tilbakemeldingene genererte, er forsøkt gjengitt som et tankekart i figur 7. Ideen til virkemåten for testapplikasjonen (StudyCandidates) som er utviklet i denne oppgaven oppsto i en tankeprosess i sammenheng med disse tilbakemeldingene. Adskillelsen av webtjenestene i to forskjellige grensesnitt oppsto også som en direkte følge av modellen som er skissert, og er å betrakte som den viktigste føringen for design av løsningen.



Figur 7. Tankekart som viser nøkkelementer som er viktige i en tenkt applikasjon som skal viderefremme bilder til et postprosesseringsystem i forskningsøyemed.

Dette tankekartet ble foredlet videre til en mer logisk skisse til bruk i designarbeidet med webtjenestegrensesnittene. Skissen i figur 8 viser arbeidsmodellen i en mer strukturert form.



Figur 8. Raffinering av tankekart fra figur 7. Her vises den segregeringen som er av tjenestene. Aktørene StudyCandidates, DICOM og MATLAB_WS vises sammen med de tjenestene som de skal ivareta.

3.4. **Responstider**

Ved innsamling av responstider ble det lagt vekt på å se om overbyggingen av DICOM kommunikasjon med webtjenestelag påvirker ytelsen av applikasjonen. Webtjenestelaget vil gi en forsinkelse i responstid i forhold til bare ren DICOM trafikk.

Responstidene som er registrert er målinger som blir generert i loggfiler som applikasjonene genererer. Disse tidene blir registrert når forskjellige metoder i systemet blir kjørt (se eksempel på dette i programkode for `myMenuStudy_ItemClicked` for `StudyCandidates` i kapittel 9.7). På denne måten er det mulig å spore den tidsbruk som er brukt på de forskjellige stegene i applikasjonslogikken. Metodikken som er benyttet er å vurdere disse verdiene etter kjøring av forskjellige deler av systemet gjentatte ganger. Det er i denne vurdering av tallmateriellet forkastet høyeste og laveste responstid for å minimere usikkerhet med tanke på eksterne påvirkningskilder. Dette kan være generell diskaktivitet, *caching* av resultat med videre som kan påvirke systemet og skape ekstremverdier i loggingen.

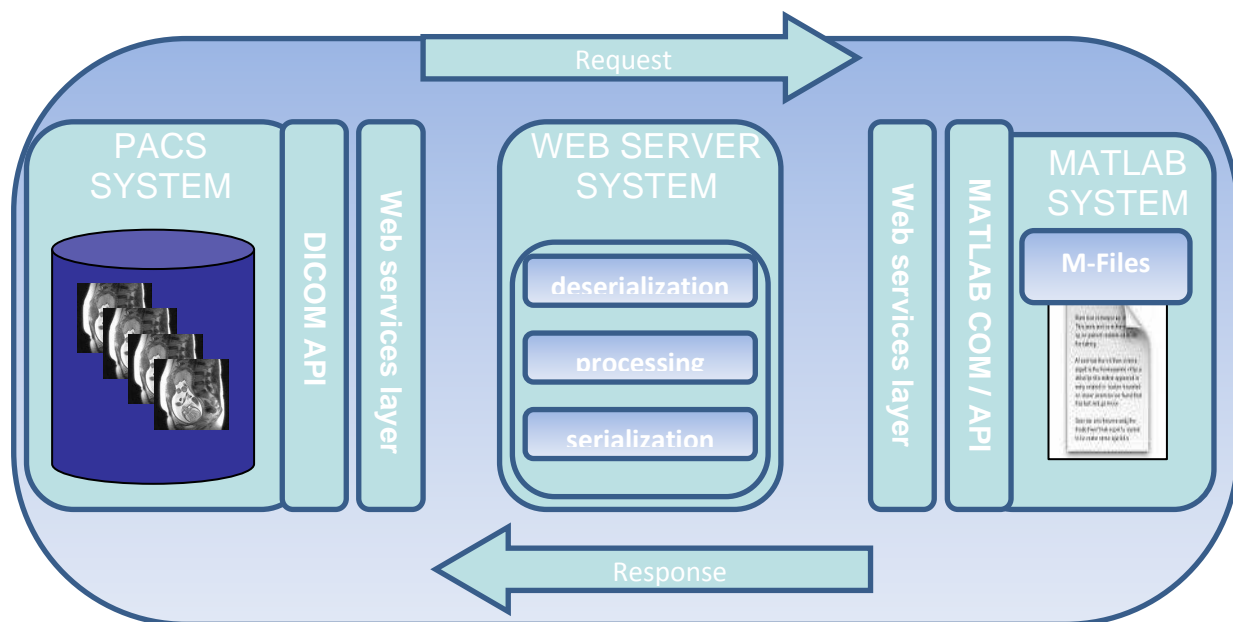
Under oppsamling av tidsmålingene relatert til flytting av bilder, ble det først foretatt en *C-MOVE* fra avsender til mottaker på aktuell undersøkelse. Dette for å prøve å begrense påvirkningen som *caching* av bilder kan ha. Bildene som ble mottatt i denne pre-flytting ble også slettet fra disk hos mottaker slik at skrivetid (I/O) hos mottaker ikke påvirket resultatene.

4. UTVIKLING

4.1. Beskrivelse av løsning

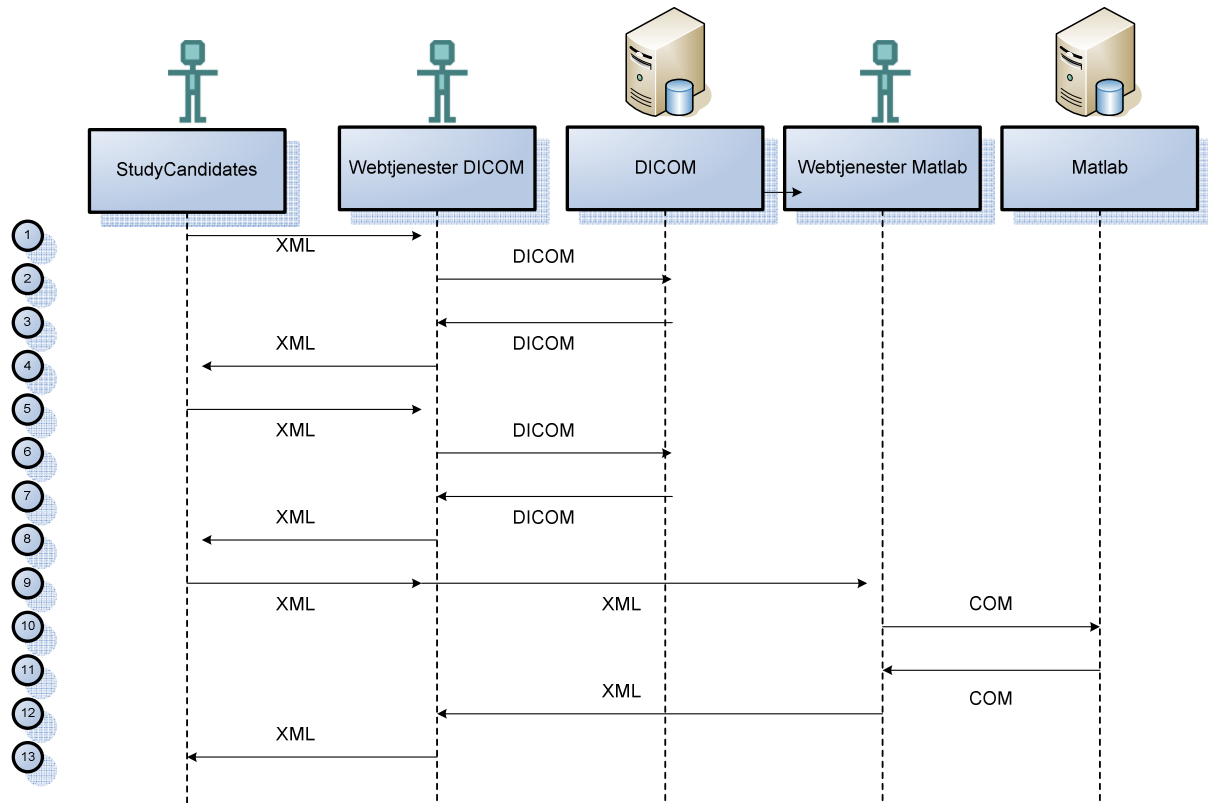
I denne oppgaven har det blitt programmert grensesnitt og en testapplikasjon som automatiserer prosessen med å hente ut spesifikke serier i angitte undersøkelser på en intuitiv måte, og som gjør disse automatisk tilgjengelig til videreprosessering i Matlab. For å muliggjøre dette kan programmet velge ut en spesifikk serie i flere undersøkelser, ekstrahere seriene ut av PACS, initiere en funksjon i Matlab, for videre å lese resultatet ut til en fil, eller tilbake til PACS selv. Det er i denne oppgaven ikke benyttet DICOM *Storage Commitment* meldinger.

Essensen av løsningen som er utviklet er et webgrensesnitt som maskerer DICOM trafikk som webtjenester, og som har mulighet til å generere prosesseringsjobber til Matlab. Disse prosesseringsjobbene er basert på egenproduserte Matlab funksjonsfiler (m-filer) som er gjort tilgjengelig i systemet. Resultatet av denne prosesseringen logges i en fil på en server, eller returnerer bildene og lagrer dem inn i aktuell undersøkelse som en ny postprosessert serie. Denne logiske prosessflyten er gjengitt i figur 9.



Figur 9. Overordnet betraktning av systemet. Til venstre i figuren ses representering av et standard PACS lagringssystem, som har sin implementering av DICOM 3.0 standard. I forkant av dette er det satt inn et webtjenestegrensesnitt som eksponerer disse tjenestene. Ved å benytte seg av vanlig SOAP meldinger og webtjenestebruk sendes det forespørsel og mottas svar mellom DICOM Webtjenestegrensesnitt og Matlab Webtjenestegrensesnitt. Forespørsler blir omdefinert til COM forespørsler på serveren, og standard Matlab installasjon benyttes for å utføre de jobbene som blir sendt. I denne oppgaven pakker SOAP meldingene inn et eget definert XML objekt.

En videre presisering av løsningen er beskrevet i figur 10.



- 1 Forespørsel etter tilgjengelige objekter ved hjelp av webtjeneste
- 2 Spørre etter objekter ved hjelp av DICOM
- 3 Motta svar på forespørsel som DICOM
- 4 Pakke identifikatorer inn i XML som returneres
- 5 Etterspør objekter med identifikatorer ved hjelp av webtjeneste
- 6 C-MOVE DICOM bildene basert på xmlforespørsel
- 7 Motta DICOM bildene med SCP tjeneste
- 8 Gi resultat som XML respons
- 9 Etterspør Matlab tjeneste, proxy fra DICOM grensesnitt til Matlab grensesnitt
- 10 Instansiere og eksekvere m-funksjon
- 11 Returnere resultat fra Matlab
- 12 Pakke inn resultat fra Matlab og returnere som XML til proxytjenesten
- 13 Returnere XML dokument med resultat

Figur 10. Eksempel på prosessflyt i applikasjonen. Her vises aktørene som samhandler for å utføre en oppgave. StudyCandidates gjør en forespørsel til webtjenester DICOM ved bruk av vanlig XML. Denne forespørsel blir pakket opp og omformet til en DICOM forespørsel mot DICOM server. Når svaret forligger sendes det tilbake til StudyCandidates som et XML svar. StudyCandidates foretar da en ny forespørsel med den informasjon som er mottatt fra webtjenesten, og en ny forespørsel blir sendt. Når de logiske DICOM operasjonene er ferdigstilt sendes en forespørsel til webtjenesten på Matlab webtjenestegrensesnitt, ved hjelp av proxytjeneste på DICOM webtjenestegrensesnitt, som starter en COM forsendelse til Matlab på den funksjonen som skal kjøres.

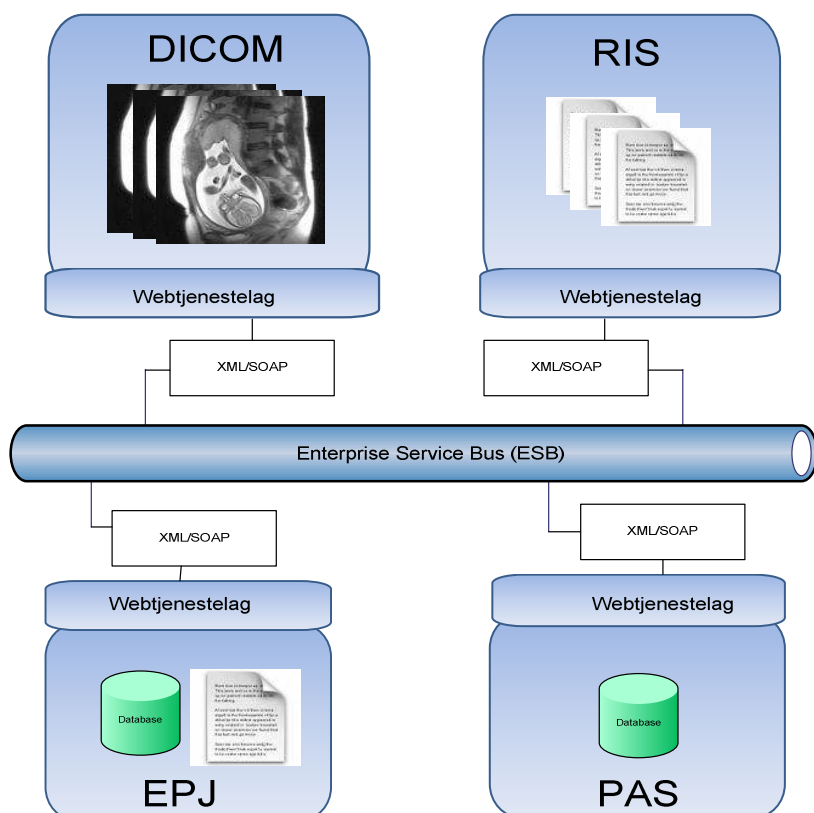
Av figur 10 kan en se hvordan meldingsflyten er implementert. Kommunikasjon fra klient (aktør) foregår direkte mot det ene grensesnittet. Dette grensesnittet eksponerer nødvendige webtjenester fra Matlab grensesnittet som proxytjenester. DICOM objektene pakkes inn i ett XML dokument og grensesnittet mot Matlab benytter COM objekt som er tilgjengelig fra Matlab.

Det som i praksis er gjort for å tilby funksjonalitet er følgende:

- Det er programmert en egen DICOM SCP tjeneste som kan motta bilder fra et system, og som lagrer dem i en trestruktur på eget filområde.
- Det er programmert et webtjenestegrensesnitt som eksponerer basis DICOM 3.0 tjenester som webtjenester.
- Det er programmert et annet webtjenestegrensesnitt som kan opprette og ta imot arbeidsordrer.
- Det er programmert en egen applikasjon for å overvåke køer som har kontroll med kommunikasjonen til og fra et COM grensesnitt.
- Det er programmert eksempel på program som benytter seg av disse nevnte webtjenester og applikasjoner for å vise hvordan en slik tjenesteorientert tilnærming kan benyttes.

Det er lagt fokus på å undersøke tidsestimater for tjenestene, og om integrasjonsplattformene kan ha en direkte nytteverdi for en uavhengig applikasjon.

Denne oppgaven prøver å gi DICOM tjenestene et rammeverk som på sikt kan videreutvikles sammen med andre medisinske komponenter til å kommunisere på en Enterprise Service Bus, se figur 11.



Figur 11. Enterprise Service Bus. Her sees en logisk kopling av aktuelle aktører på en ESB. I fremkant av tjenestene er en XML/SOAP webtjenestegrensesnitt som kan samhandle. På denne måten blir det mulig å utveksle informasjon, og gjensidig starte prosesser som blir eksponert på denne ESB.

4.2. Erfaringer fra utviklingsprosessen

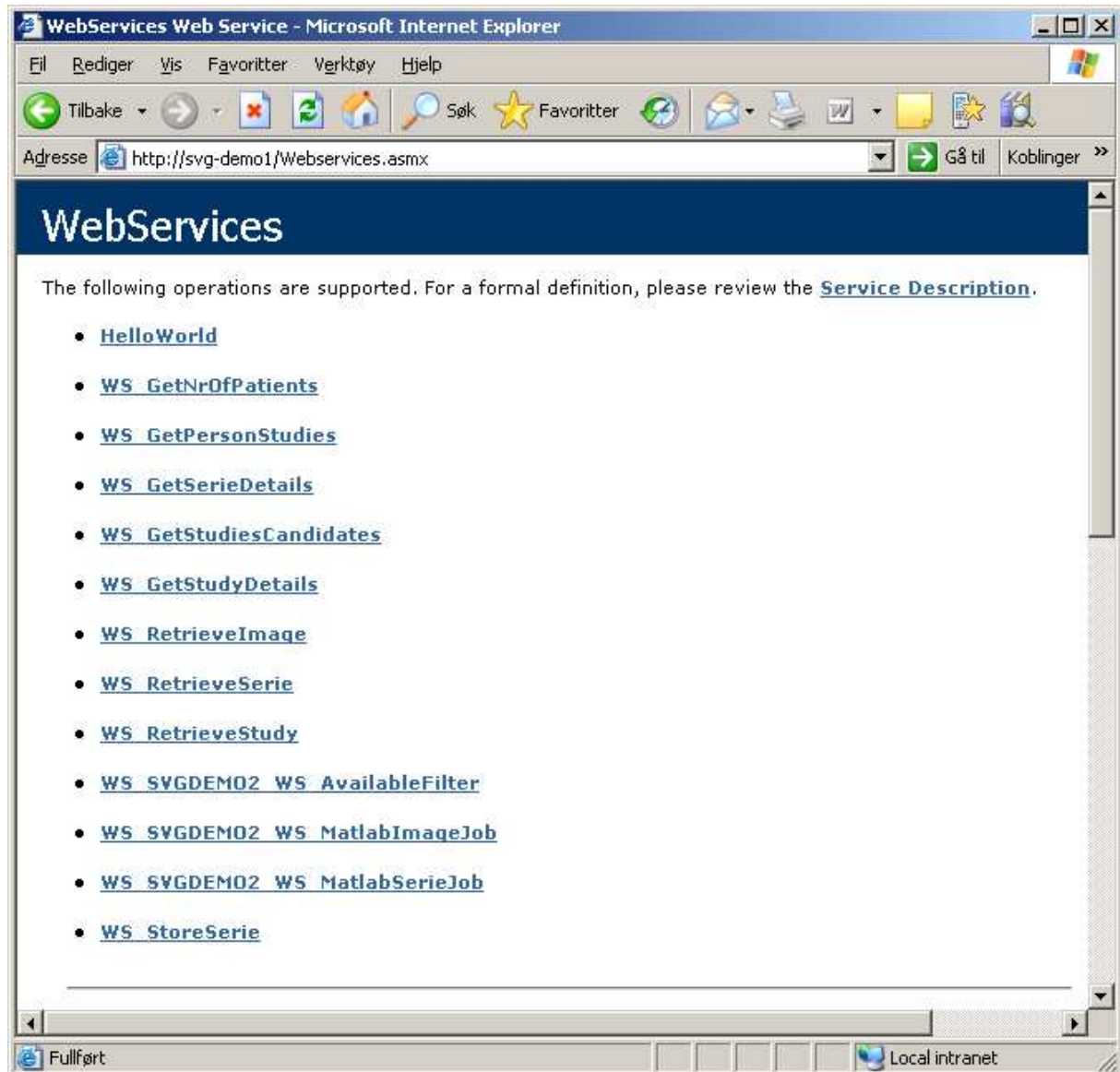
Som nevnt i kapittel 3 ble det i begynnelsen av utviklingsarbeidet benyttet en DICOM pakke fra Albert Gnanndt. Denne pakken ga ikke ønsket ytelse. OpenDICOM-sharp ble derfor forkastet og programmene ble endret til å benytte ClearCanvas SDK.

Under testing av responser ved bruk av webtjenestene oppsto det en *exception* fra Microsoft Visual Studio, *QuotaExceededException*. Standard konfigurasjon av webtjenester viste seg å ha en begrensning på hvor mye data som kunne bli sendt for hver overføring. Denne begrensningen ble korrigert for i web.config. Det viste seg også å være en standard *timeout* på selve eksekveringen av webmetoden på ett minutt. Dette er ikke formålstjenelig i DICOM trafikk da det ofte kan inntreffe reelle overføringstider på mer enn ett minutt. Det ble foretatt endringer i web.config også for denne innstillingen til å understøtte en prosessetid på opp til ti minutt.

Webtjenesten som oppretter arbeidsordrer til MatlabWrapper må ha tilgang til å skrive i angitt mappe for dette programmet. Dermed ble det eksplisitt gitt tilgang til *IIS_WPG* (Internet Information Server, Worker Process Group) gruppen slik at denne tjenesten kunne fullføre opprettelsen av filen lokalt.

4.3. DICOM Webtjenester

På serveren svg-demo1 er webtjenestene som ivaretar DICOM trafikk gjort tilgjengelig. Figur 12 viser de tjenestene som i hovedsak tilgjengeliggjør basis DICOM 3.0 tjenester som webtjenester. Webtjenestens definisjonsfil (WSDL) for disse tjenestene er tilgjengelig i vedlegg 9.1.



Figur 12. Skjerm bilde som viser de Webtjenester som er tilgjengelig fra DICOM_WS. Virkemåten til de forskjellige webtjenestene blir gjennomgått i kapittel 4.3.

Videre ser en av figur 12 at det finnes tre proxytjenester som refererer til server svg-demo2 og det webgrensesnittet som er der. Disse tjenestene har ingen annen logikk i seg en å motta forespørslers, samt å videreformidle disse til de reelle tjenestene på grensesnittet til svg-demo2. Dette er blitt gjort for å ha en løs kopling mellom grensesnittene som DICOM trafikken representerer og den koplingen som er til Matlab grensesnittet.

For å konfigurere webtjenestene til å kjøre mot riktig grensesnitt for Matlab webtjenester, må systemet konfigureres med riktige parametre. Disse parametrene settes i web.config for webgrensesnittet, under appSettings noden.

```
<appSettings>
  <add key="svg_demo2_WS.Service" value="http://svg-demo2/Matlab/Service.asmx"/>
</connectionStrings/>
```

Her settes også DICOM konfigurasjonsparametre, og for testsystemet er denne konfigurasjonen gjeldene.

```
<appSettings>
  <add key="LocalStudyRoot" value="C:\DicomFiles"/>
  <add key="LocalAETitle" value="DEMOAE"/>
  <add key="RemoteAETitle" value="MERGE"/>
  <add key="RemoteIPAdress" value="161.4.246.27"/>
  <add key="RemotePort" value="4006"/>
</appSettings>
```

Disse variablene blir lest inn i initieringen av webtjenestene av funksjonen *ReadWebConfigSetting*.

Denne funksjonen ligger i DicomConnect klassen (se kapittel 9.1.4).

4.3.1. WS_GetNrOfPatients

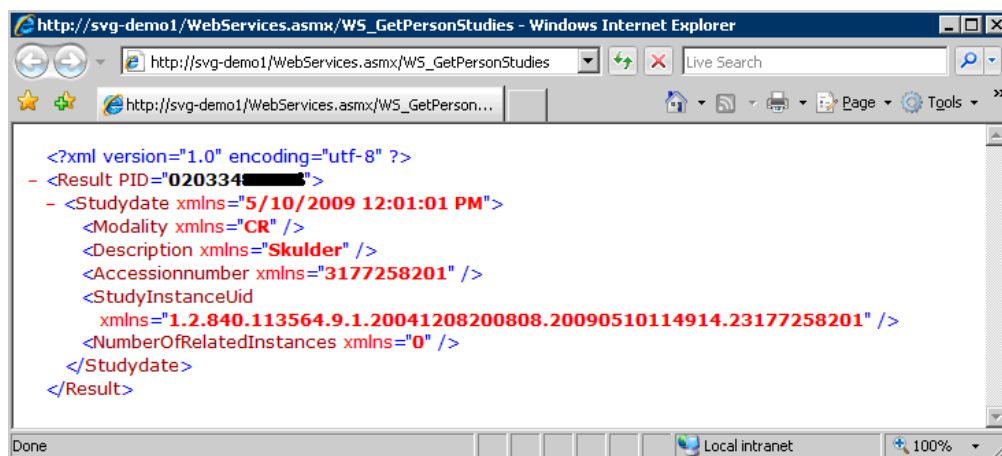
Denne webtjenesten maskerer DICOM tjenesten *N-FIND*, som opererer på PatientRoot nivå. Input parameter for tjenesten er PID (Personnummer). Resultatet av WS vil være at den returnerer en streng som angir hvor mange pasienter med samsvarende pasientid som er tilgjengelig i DICOM lagringsarkiv

Webtjenesten kan returnere med

Antall pasienter funnet	: # patients found.
En pasient funnet	: 1 patient.
Ingen pasienter funnet	: No patient found.

4.3.2. WS_GetPersonStudies

Denne webtjenesten maskerer DICOM tjenesten *C-FIND*, som opererer på StudyRoot nivå. Inputparameter er igjen PID, men resultatet av denne tjenesten vil være et *XmlDocument*. Tjenesten serialiserer resultatet fra DICOM spørringen og returnerer et svar på en strukturert form i et *XmlDocument* objekt. Figur 13 viser et slikt XML dokument.



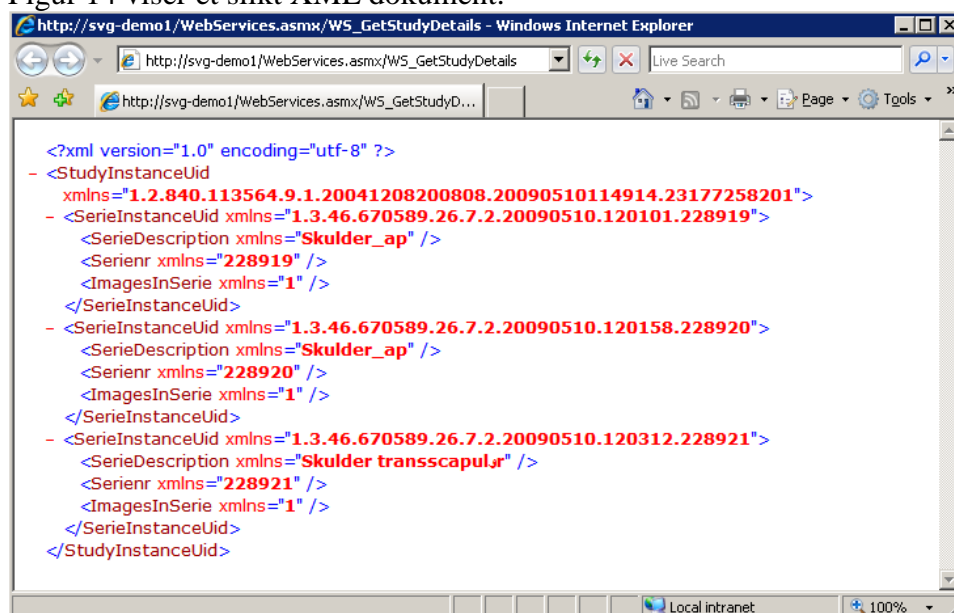
Figur 13. Resultatdokument fra WS_GetPersonStudies spørring. I dette skjermbildet kan det sees at det er foretatt en spørring etter PID "020334xxxxx". Resultatet viser at denne personen har en undersøkelse fra 05.10.2009. Dette er en CR skulder undersøkelse. StudyInstanceUID viser den unike identifikatoren for akkurat denne undersøkelse. Det vises også at antall relaterte undersøkelser tid denne er 0.

Figur 13 viser de attributtene som returneres. I den tjenesten som er generert returneres dato for undersøkelse, deretter spesifikke attributter for Modalitet, Beskrivelse og Accessionnummer. Siste attributt som returneres er *StudyInstanceUID*, som er den unike identifikatoren for akkurat denne undersøkelse. Dette er som nevnt tidligere den unike identifikatoren for dette bilde.

4.3.3. WS_GetStudyDetails

Denne webtjenesten maskerer også DICOM tjenesten *C-FIND*, men webtjenesten blir instansiert med andre inputparametre. Denne tar med seg nevnte *StudyInstanceUID* og finner relaterte serier i nettopp denne undersøkelse. Resultatet av denne tjenesten vil også være et *XmlDocument*.

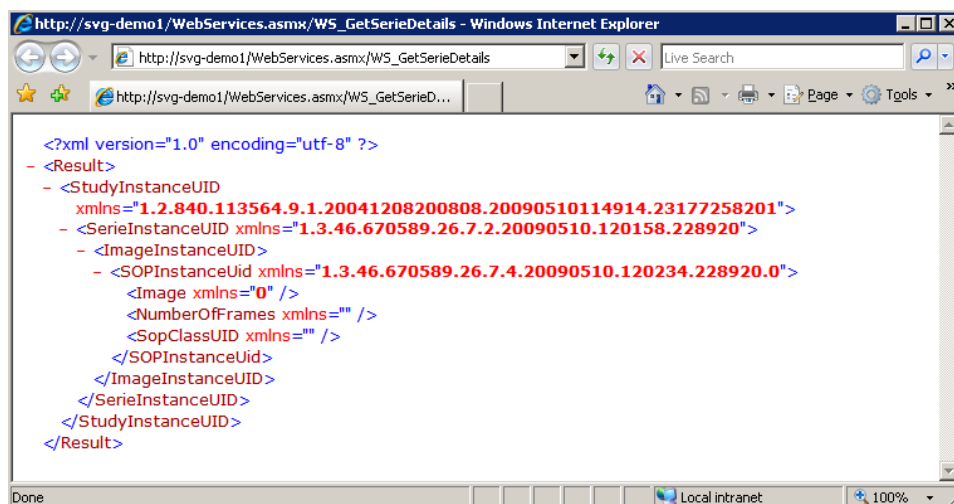
Figur 14 viser et slikt XML dokument.



Figur 14. Resultatdokument fra WS_GetStudyDetails spørring. Dette skjermbildet viser at angitt unik identifikator for undersøkelsen har tre tilknyttete serier. I hver enkel tilknyttet serie er det oppgitt den unike serieidentifikatorene for serien, sammen med tekstlig beskrivelse av hvilken serie som er utført. Videre vises serienummer, og antall bilder i hver enkelt serie.

4.3.4. WS_GetSerieDetails

Enda en webtjeneste som maskerer DICOM tjenesten *C-FIND*, men instansieres med *StudyInstanceUID* og med *SerieInstanceUID*. Denne webtjenesten returnerer et *XmlDocument* som finner relaterte bilder til denne serien. Figur 15 viser et slikt XML dokument.



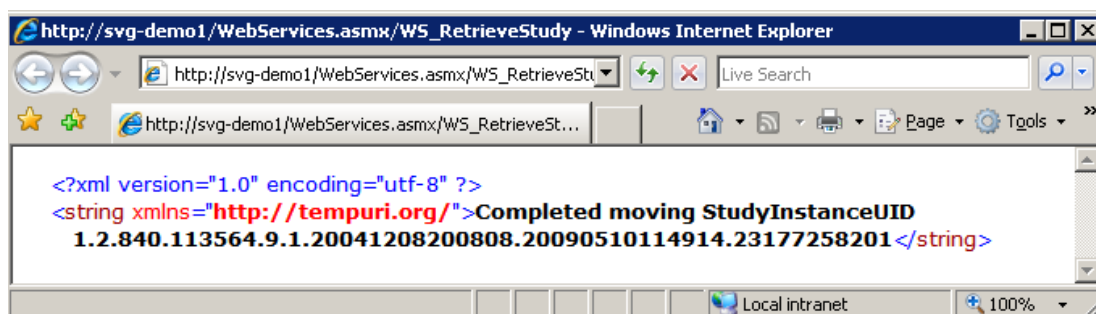
Figur 15. Resultatdokument fra WS_GetSerieDetails spørring. Skjermbildet viser de unike identifikatorene for undersøkelse, serie og bilde. I dette skjermbilde kan en se at denne serien har ett bilde (image 0). SOPClassUID for dette bildet er ikke satt (som angir type).

Av figur 15 ser en at denne serien i angitt undersøkelse har et bilde, som identifiseres med *SOPInstanceUID*. Dersom denne serien hadde flere bilder ville webtjenesten returnere flere XML noder, en for hvert bilde.

4.3.5. WS_RetrieveStudy

Webtjenesten WS_RetrieveStudy har en innparameter med StudyInstanceUID. Denne webtjenesten eksponerer egentlig en *DICOM C-MOVE*, men siden AETitle er hardkodet i webtjenesten, vil den i prinsippet være en *N-GET* tjeneste. Webtjenesten vil initiere en DICOM trafikk som flytter denne undersøkelse med alle serier og bilder til konfigurert DICOM aktør.

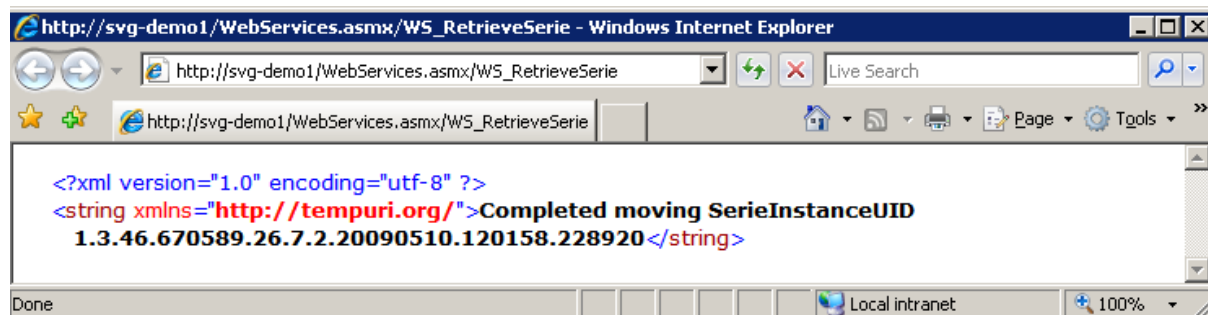
Resultatet fra en forespørsel til webtjenesten kan se ut som i figur 16.



Figur 16. Resultatdokument fra WS_RetrieveStudy spørring. Her vises en melding som informerer om at undersøkelse med angitt StudyInstanceUID er ferdig flyttet fra aktør A til aktør B.

4.3.6. WS_RetrieveSerie

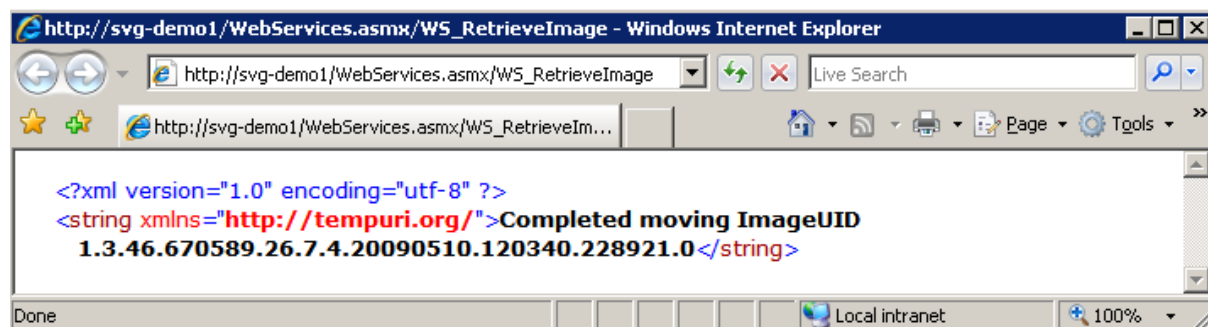
WS_RetrieveSerie fungerer på samme måte som WS_RetrieveStudy, med unntak av at denne tjenesten bare vil returnere akkurat denne serien og tilhørende bilder til konfigurert SCP. Figur 17 viser XML dokument fra WS_RetrieveSerie forespørsel.



Figur 17. Resultatdokument fra WS_RetrieveSerie spørring. Her kan en se at DICOM serie, med identifikator SerieInstanceUID er ferdig flyttet til konfigurert DICOM aktør.

4.3.7. WS_RetrieveImage

Denne webservicen fungerer som de to foregående, men denne plukker ut et unikt bilde og sender det til mottaker. Figur 18 viser informasjonsmeldingen som informerer at angitt bilde er kopiert til konfigurert DICOM aktør.



Figur 18. Resultatdokument fra WS_RetrieveImage spørring. Informasjonsmeldingen viser at bilde med angitt ImageUID er flyttet til konfigurert DICOM aktør.

4.3.8. WS_GetStudiesCandidates

Dette er en spesialtilpasset webtjeneste som fungerer annerledes enn vanlige DICOM basis tjenester, men som er bygget rundt DICOM *C-FIND*.

Dette er en aggregert tjeneste som har innparametrene *studyrange*, *modality*, *studydescription* og *sex*. Denne tjenesten vil finne alle undersøkelser som passer til angitte filter. Dersom et filter ikke er angitt, vil tjenesten se bort i fra det filteret.

Denne tjenesten kan benyttes for å angi parameterkriterier som anses som viktige for å hente ut et representativt utvalg.

- StudyRange settes for eksempel til å være 20080101-20081231 for å angi forrige år.

- Modality for å angi type modalitet, eksempelvis CR undersøkelser.
- StudyDescription for å angi type av undersøkelse, eksempelvis Skulder_ap.
- Sex for å angi om dette eksempelvis bare skal være kvinner (F=Female).

4.3.9. WS_SVGDEMO2_WS_AvailableFilter

Denne webtjenesten fungerer som en proxytjeneste for webtjenesten WS_AvailableFilter på serveren svg-demo2. Virkemåte er beskrevet i kapittel 4.4.

4.3.10. WS_SVGDEMO2_WS_MatlabImageJob

Denne webtjenesten fungerer som en proxytjeneste for webtjenesten WS_MatlabImageJob på serveren svg-demo2. Virkemåte er beskrevet i kapittel 4.4.

4.3.11. WS_SVGDEMO2_WS_MatlabSerieJob

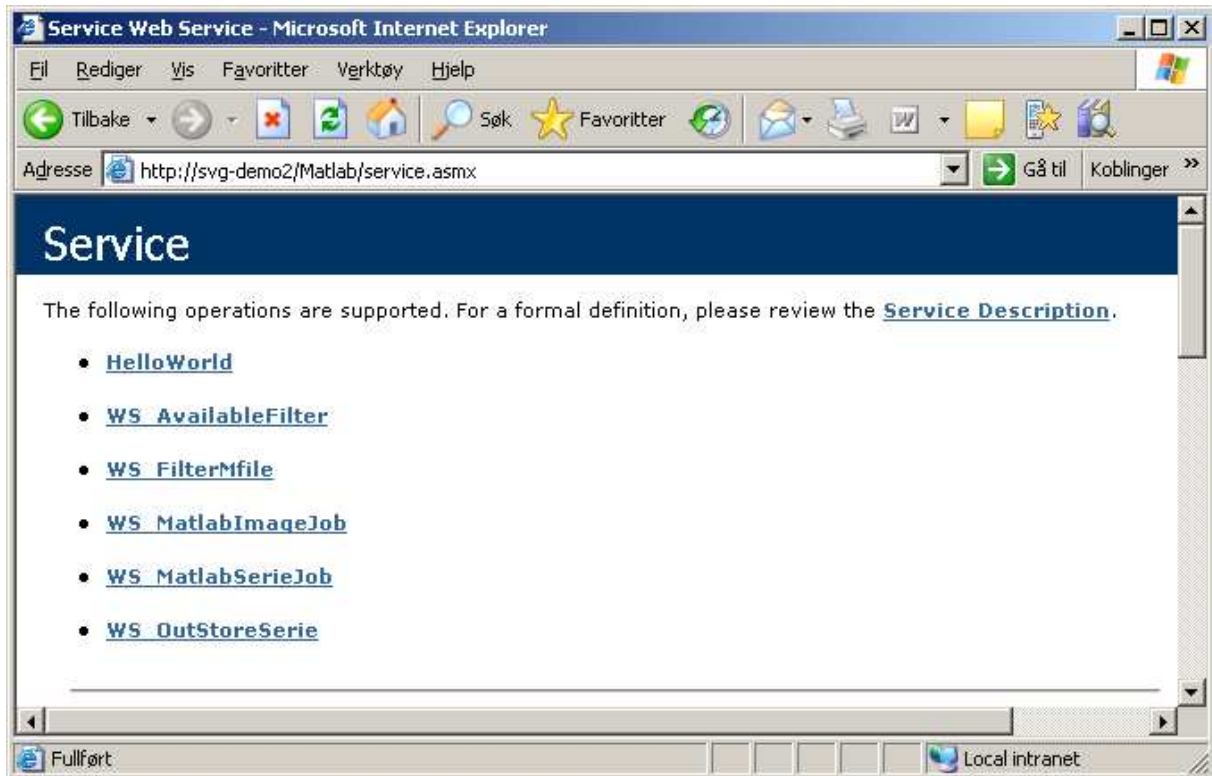
Denne webtjenesten fungerer som en proxytjeneste for webtjenesten WS_MatlabSerieJob på serveren svg-demo2. Virkemåte er beskrevet i kapittel 4.4.

4.3.12. WS_StoreSerie

Denne webtjenesten er også eksponert som proxytjeneste for webtjenesten WS_OutStoreSerie på serveren svg-demo2. Denne tjenesten har *StudyInstanceUID* og *SerieInstanceUID* som innparametre og identifiserer den spesifikke serien i angitt undersøkelse som skal sendes til mottaker. Denne tjenesten eksponerer DICOM tjenesten *C-STORE*. Webtjenesten vil først lese inn de bildene som er blitt gjort tilgjengelig i egen logisk trestruktur inn i en DICOM lagringspakke. Tjenesten vil så initiere en ny DICOM *C-STORE* mot mottaker, og sende disse bildene.

4.4. MATLAB Webtjenester

På serveren svg-demo2 er webtjenestene som ivaretar kommunikasjon med Matlab gjort tilgjengelig. Figur 19 viser disse. Webtjenestens definisjonsfil (WSDL) for disse tjenestene er tilgjengelig i vedlegg 9.2.



Figur 19. Webtjenester fra Matlab_WS. Skjerm bilde viser de webtjenestene som kommuniserer med Matlab. Disse blir gjennomgått i kapittel 4.4

4.4.1. WS_AvailableFilter

Dette er en webtjeneste som annonserer hvilke filtre som er tilgjengelig i systemet. En forespørsel mot denne tjenesten vil resultere i et XML dokument som vist i figur 20.



Figur 20. XML Dokument som viser tilgjengelige funksjoner. I dette XML dokumentet kan brukerne definere inn sine egne filtre som automatisk blir gjort tilgjengelig i webtjenestegrensesnittet. XML Root noden MatlabFilters vil ha de aktuelle beskrivelsene av filterne under seg. Hvert barn av disse vil ha konfigurering av nivå (Serie/image) som vil identifisere hvilket nivå disse filterne skal være tilgjengelige fra. Videre har disse barnnodene en m-file node som identifiserer navnet på Matlab funksjonen.

Dette er et XML dokument som blir generert ut i fra de filtre som er blitt definert inn av bruker i filen *MatlabFilters.xml*. I *MatlabFilters.xml* er tanken at brukerne eller administratorer skal legge inn de egendefinerte funksjonene som er utviklet i Matlab. Konfigureringen er basal, det opprettes xmlnoder som angir det navnet som skal eksponeres i applikasjonslogikken. Videre angis det om funksjonen skal initieres på bildenivå, eller om funksjonen skal peke på en mappe som angir hvilken serie det skal jobbes med.

4.4.2. WS_FilterMfile

Denne tjenesten er en støttetjeneste for programmet *MatlabWrapper*. Tjenesten har to innparametre, *FilterLevel* og *FilterDesc*. Disse to angir en unik xmltraversering for å finne konfigurert m-fil, som er den filen som skal kjøres av programmet *MatlabWrapper*.

4.4.3. WS_MatlabImageJob

Denne webtjenesten instansieres for å opprette en Matlab jobb. Tjenesten vil lese inn parametrene *StudyInstanceUID*, *SerieInstanceUID*, *ImageUID*, *FilterName* og *FilterLevel*. Disse verdiene blir lagret i en fil på *MatlabWrapper InBound* område på dobbel pipe separert måte.

4.4.4. WS_MatlabSerieJob

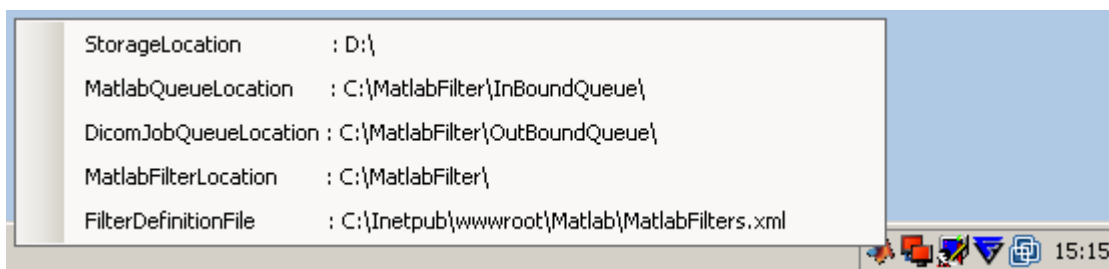
Denne webtjenesten instansieres for å opprette en Matlab jobb. Tjenesten vil lese inn parametrene *StudyInstanceUID*, *SerieInstanceUID*, *FilterName*, *FilterLevel*. Disse verdiene blir lagret i en fil på *MatlabWrapper InBound* område på dobbel pipe separert måte.

4.4.5. WS_OutStoreSerie

Denne tjenesten fungerer som en proxytjeneste for webtjenesten *WS_StoreSerie* på *svg-demo1*. (Kapittel 4.3.12)

4.5. *MatlabWrapper*

MatlabWrapper er et program med formål å foreta kommunikasjon til og fra Matlab over det COM grensesnittet som Matlab eksponerer. Videre overvåker dette programmet en kølogikk som identifiserer jobber til og fra Matlab, og som knytter webtjenestene til serveren sammen med Matlab. Årsaken til denne splittingen av applikasjonslogikk var at instansiering av dette COM grensesnittet tar veldig lang tid, og ville være en uhensiktsmessig forsinkelse i den totale løsningen. Tiden for å instansiere Matlab er nå bundet opp mot oppstart av MatlabWrapper, og skjer i prinsippet bare en gang siden løsningen holder COM grensesnittet åpent. Instansieringen til COM tar på svg-demo2 serveren mellom 5 og 8 sekunder.



Figur 21. Konfigureringsparametre for MatlabWrapper. Her vises konfigurasjonen som er lest fra inifil og presentert som en innholdsmeny for brukeren ved å høyreklikke på ikonet i startmenyen.

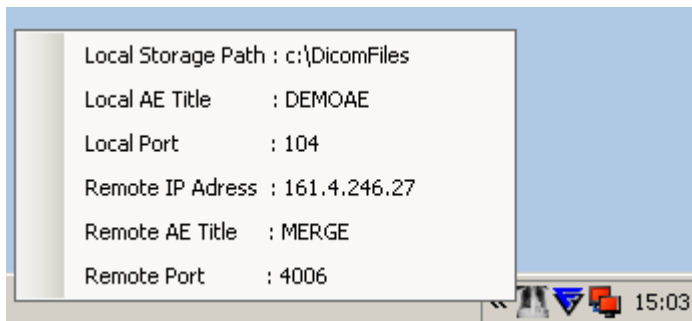
Figur 21 viser konfigurasjonen til MatlabWrapper programmet. Disse parametrene defineres i tilhørende inifil, og menyen vises ved høyreklikk på Matlabsymbolet.

Som nevnt overvåker MatlabWrapper køen MatlabQueueLocation. Når en ny jobb fra webtjenestene WS_MatlabImageJob eller WS_MatlabSerieJob blir opprettet, leser MatlabWrapper innholdet i jobben og kjører m-fil med angitte parametre.

Dersom en m-funksjon er programmert til å opprette nye serier, og lagrer en dicfil i DicomJobLocation vil MatlabWrapper lese denne jobben, og kalle webtjenesten WS_OutStoreSerie.

4.6. APPLICATION SCP

For å kunne motta DICOM filer fra en annen DICOM node var det hensiktsmessig å lage en egen SCP tjeneste på svg-demo1 serveren. Denne SCP har som oppgave å fungere som en fullverdig *DICOM Storage Provider*. Konfigurasjon av denne blir også lest fra en tilhørende inifil. På serveren vil en prosess kjøre applikasjonen, og konfigurasjonen kan vises ved å høyreklikke på ikonet nede på startmenyen, som vist i figur 22.



Figur 22. Konfigureringsparametre for ApplicationSCP. Her vises konfigurasjonen som er lest fra inifil og presentert som en innholdsmeny for brukeren ved å høyreklikke på ikonet i startmenyen.

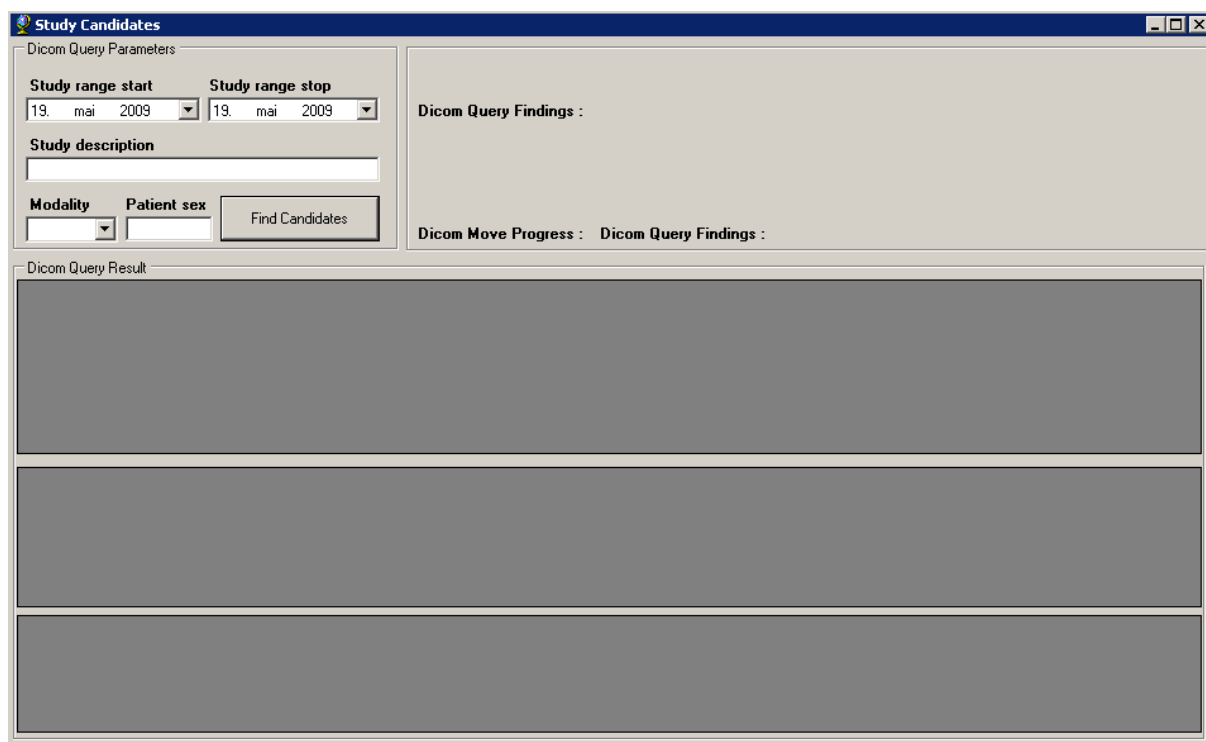
Applikasjonen som er konfigurert i testmiljøet lytter etter DICOM kommunikasjon på port 104. Når en *C-STORE* blir sendt på denne kommunikasjonsporten vil tilhørende bilder bli mottatt og lagret ned i en mappestruktur under Local Storage Path. Denne strukturen vil være:

Local Storage Path\StudyInstanceUID\SerieInstanceUID\ImageUID (SOPUID)

5. TESTRESULTATER

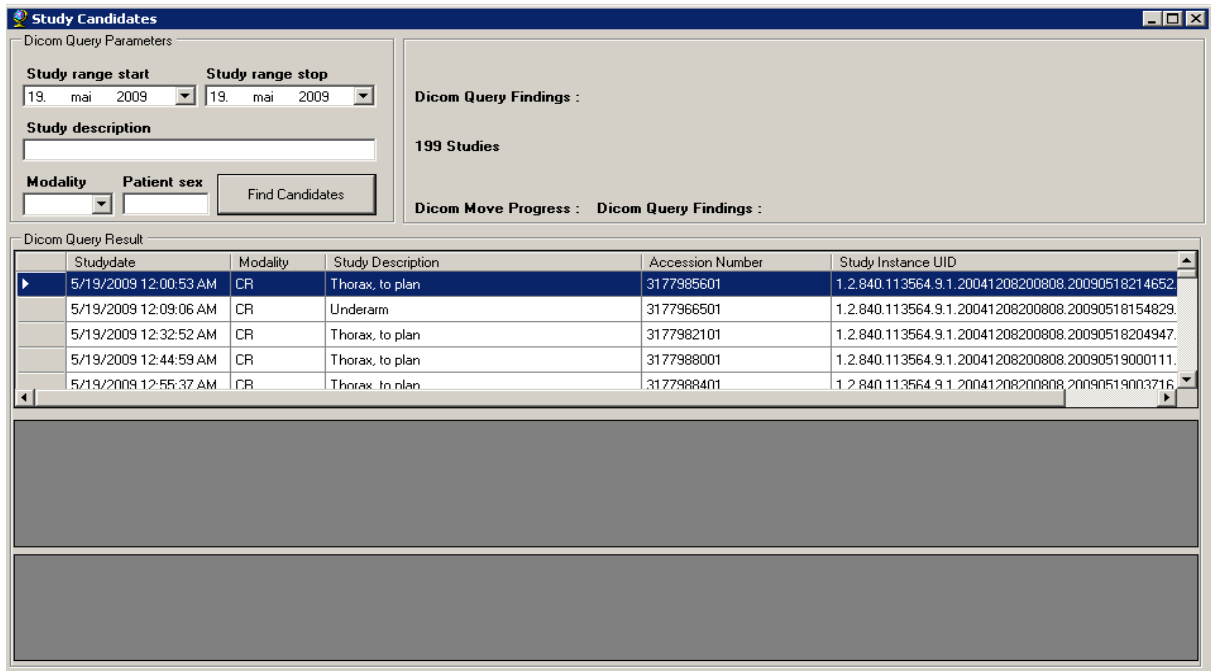
5.1. StudyCandidates

StudyCandidates er et program som er utviklet for å demonstrere bruken av webtjenestene og løsningene som er utviklet. Figur 23 til figur 29 viser skjermbilder som demonstrerer virkemåte til programmet.



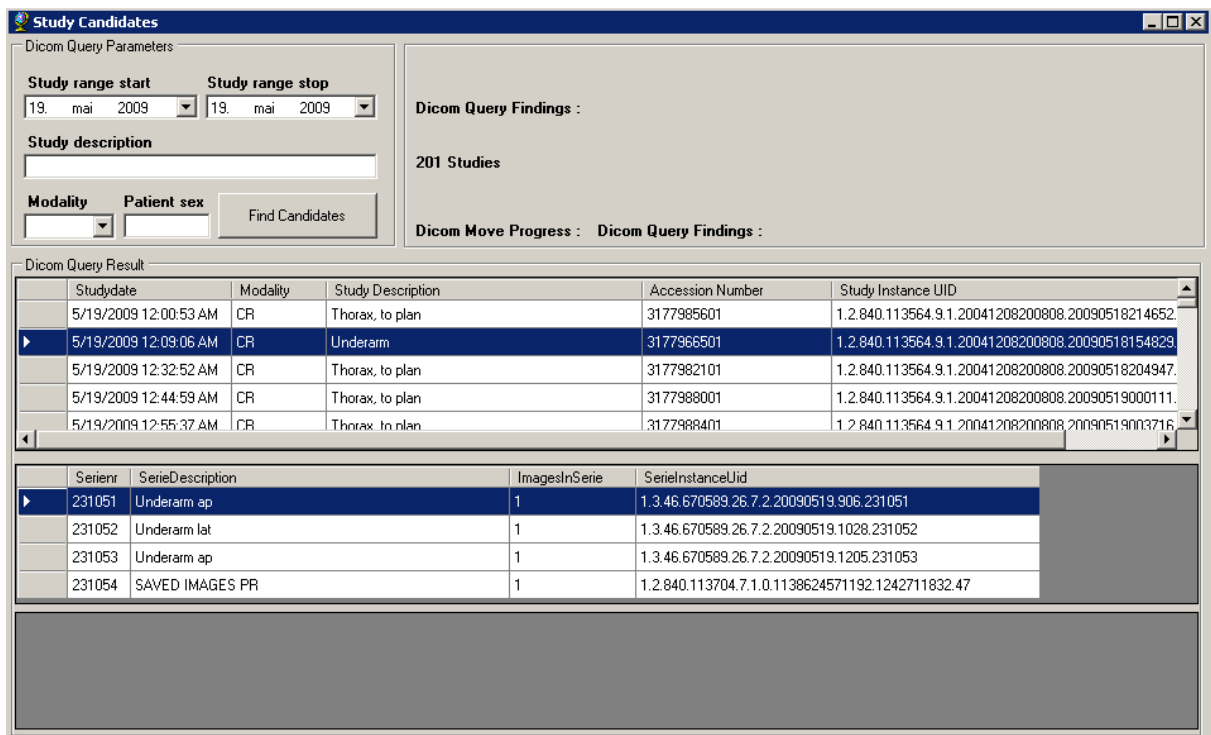
Figur 23. StudyCandidates oppstart. Under DICOM Query Parameters vises de filtreringer som er mulig å velge. Dette er undersøkelsesområde(StudyRange), beskrivelse (Study description), modalitet (Modality) og kjønn (Patient sex).

Figur 23 viser oppstartsbilde til programmet. Øverst i venstre hjørne er parametrene som begrenser søket. Dette er parametrene som gjenspeiler WS_StudyCandidates.



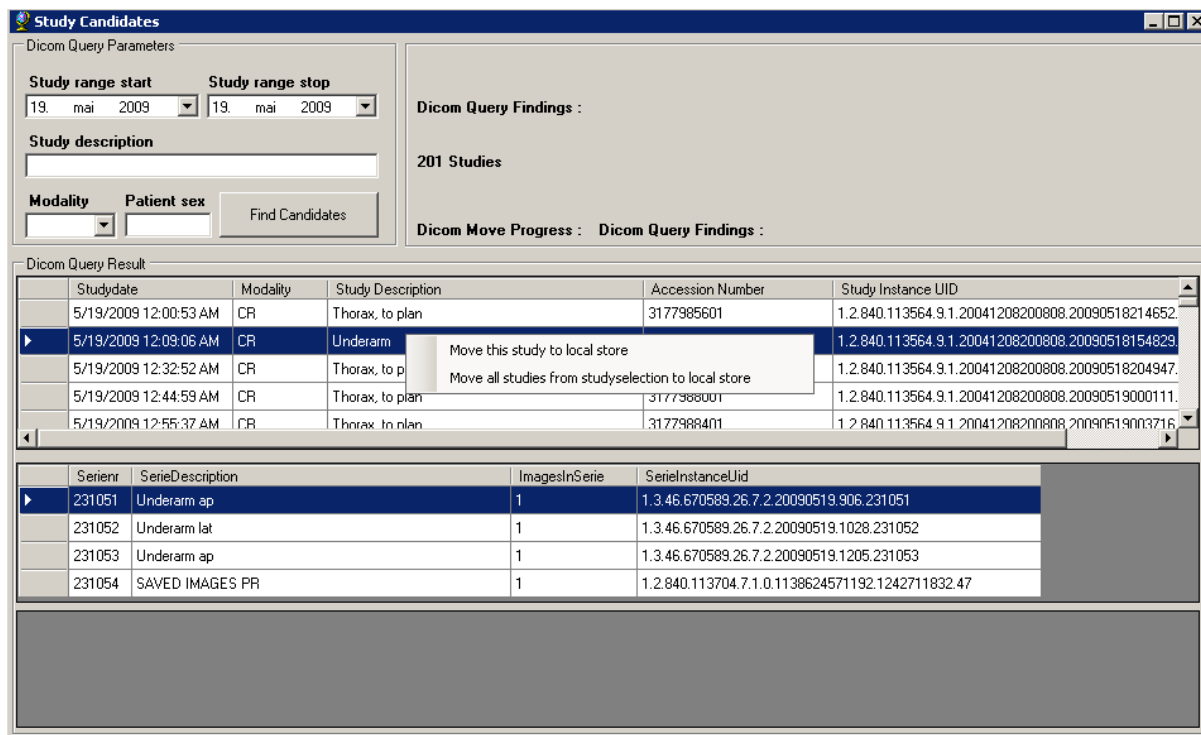
Figur 24. StudyCandidates Søk. Ved å foreta filtreringsvalg og klikke på Find Candidates foretas en spørring mot konfigurert DICOM bildearkiv. Skjermbildet viser en spørring etter alle undersøkelser utført 19.mai 2009.

I eksemplet som er vist ønsker man å se på undersøkelser tatt fra en bestemt dato, 19. mai 2009. Webtjenesten returnerer resultatet som et XML dokument som populerer datagrid for undersøkelsene, som vist i figur 24.



Figur 25. StudyCandidates valgt undersøkelse. Skjermbildet viser at valg av undersøkelse automatisk foretar en ny spørring til DICOM bildearkiv etter informasjon om denne undersøkelsen. Resultatet av denne spørringen blir vist i den midterste datagrid.

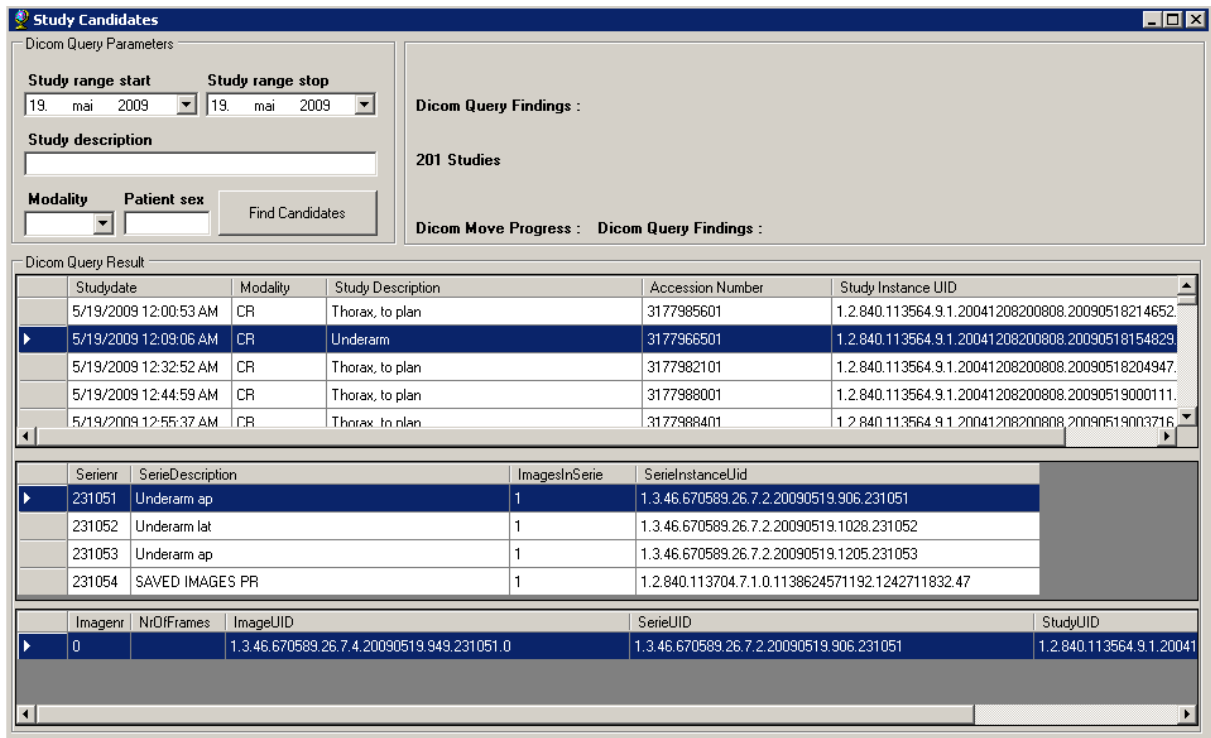
Figur 25 viser at ved å velge en undersøkelse blir webtjenesten WS_GetStudyDetails utført. Dette resultatet populerer neste datagrid med informasjon om den valgte undersøkelse.



Figur 26. StudyCandidates funksjonalitet studynivå. Ved å høyreklikke på en undersøkelse vil en kontekstmeny med tilgjengelige funksjoner bli generert og vist til brukeren.

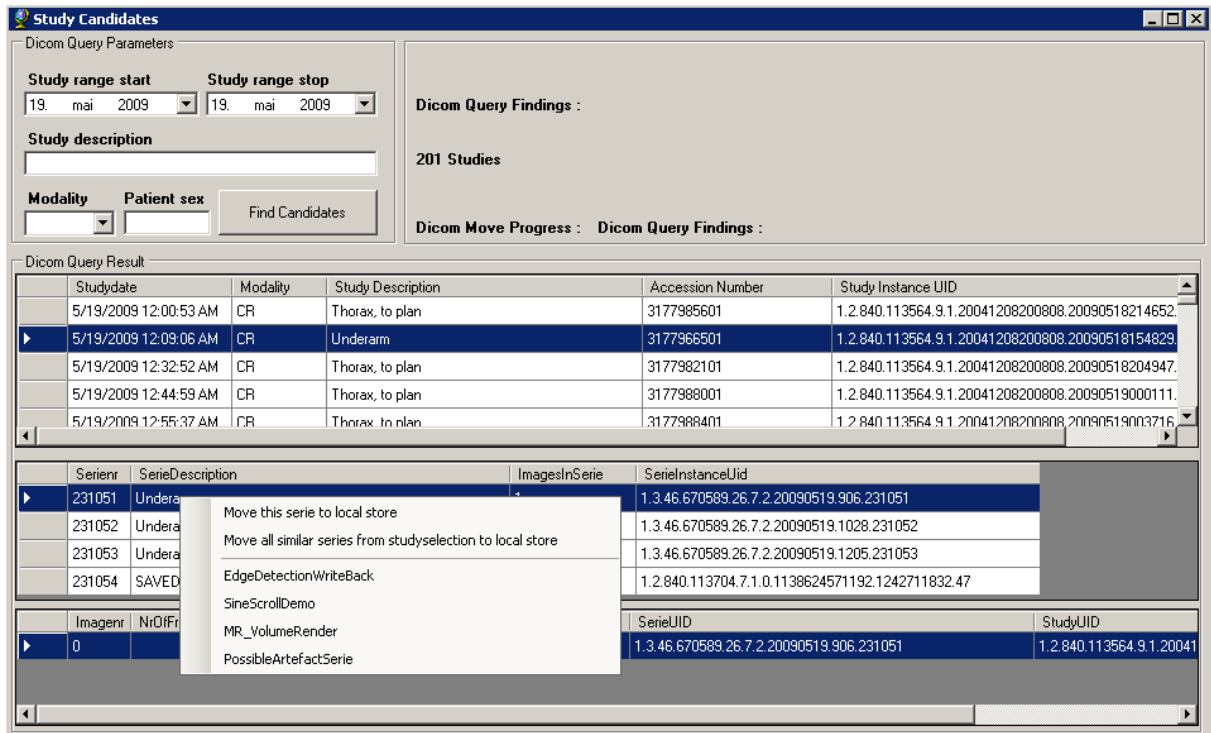
Ved å høyreklikke på valgt undersøkelse i datagrid, som vist i figur 26, vises en kontekstmeny med valgene:

- Move this study to local store.
 - Dette valget vil utføre WS_RetrieveStudy med valgt *StudyInstanceUID*.
- Move all studies from studyselction to local store.
 - Dette valget vil iterere gjennom alle undersøkelser og flytte dem til lokal maskin ved hjelp av WS_RetrieveStudy.



Figur 27. StudyCandidates valgt serie. Ved å velge en spesifikk serie i en undersøkelse vil det bli foretatt en spørring til konfigurert DICOM bildearkiv etter bildeinformasjon. Resultatet fra denne spørringen blir vist i nederste datagrid.

Figur 27 viser hva som skjer når en spesifikk serie er valgt. Når det blir foretatt et valg på serienivå, vil WS_GetSerieDetails bli utført. Dette resulterer i at datagrid for bildeinformasjon blir populert.



Figur 28. StudyCandidates seriefilter. Ved å høyreklikke på en serie vil en ny kontekstmeny med funksjoner på serienivå bli generert og vist til brukeren.

Figur 28 viser at ved å høyreklikke på serienivå vil en få følgende kontekstmeny.

- Move this serie to local store.
 - Flytte denne spesifikke serie ved å bruke WS_RetrieveSerie.
- Move all similar series from studyslection to local store.
 - Dette valget vil iterere gjennom alle studier, og de serier som har identisk seriebeskrivelse vil bli flyttet til lokalt område.
- EdgeDetectionWriteback.
 - Dette er et filter som mottar StudyInstanceUID og SerieInstanceUID, og genererer en ny postprosessert serie som blir lagret tilbake i PACS.
- SineScrollDemo.
 - Dette er et filter som viser alle bildene i en serie på svg-demo2 noden.
- MR_VolumeRender.
 - Funksjon som lager et 3D volum av tilgjengelige snitt.
- PossibleArtefactSerie.
 - Dette er et filter som er tenkt å loggføre funn i en egen resultatfil.

Som man kan se har kontekstmenyen blitt generert etter den konfigurasjonen som finnes av MatlabFilters.xml på svg-demo2. De filtre som er tilgjengelig på serienivå vil dukke opp som egne valg.

De forskjellige demonstrasjonsfiltrene er beskrevet i kapittel 5.2.

Studydate	Modality	Study Description	Accession Number	Study Instance UID
5/19/2009 12:00:53 AM	CR	Thorax, to plan	3177985601	1.2.840.113564.9.1.20041208200808.20090518214652
5/19/2009 12:09:06 AM	CR	Underarm	3177966501	1.2.840.113564.9.1.20041208200808.20090518154829
5/19/2009 12:32:52 AM	CR	Thorax, to plan	3177982101	1.2.840.113564.9.1.20041208200808.20090518204947
5/19/2009 12:44:59 AM	CR	Thorax, to plan	3177988001	1.2.840.113564.9.1.20041208200808.20090519000111
5/19/2009 12:55:37 AM	CR	Thorax, to plan	3177988401	1.2.840.113564.9.1.20041208200808.20090519003716

Serienr	SerieDescription	ImagesInSerie	SerieInstanceUID
231051	Underarm ap	1	1.3.46.670589.26.7.2.20090519.906.231051
231052	Underarm lat	1	1.3.46.670589.26.7.2.20090519.1028.231052
231053	Underarm ap	1	1.3.46.670589.26.7.2.20090519.1205.231053
231054	SAVED IMAGES PR	1	1.2.840.113704.7.1.0.1138624571192.1242711832.47

Imagernr	NrOfFrames	ImageUID	SerieUID	StudyUID
0		1.3.46.670589.26.7.4.20090519.906.231051	1.3.46.670589.26.7.2.20090519.906.231051	1.2.840.113564.9.1.20041208200808.20090518154829

Figur 29. StudyCandidates imagefilter. Ved å høyreklikke på bildenivå vil en kontekstmeny med tilgjengelige funksjoner på bildenivå bli generert og vist til brukeren. På skjermbildet kan man se tre forskjellige funksjoner som er tilgjengelig. Tilgjengelig funksjonalitet er å flytte bildene lokalt, eller å sende bildet til prosessering i Matlab med henholdsvis MyEdgeFilterDemo og PossibleArtefact.

Figur 29 viser de filtre som er tilgjengelig på bildenivå ved å høyreklikke på valgt bilde.

- Move this image to local store.
 - Dette valget vil eksekvere WS_RetrieveImage med rette parametre.
- MyEdgeFilterDemo.
 - Dette filteret vil vise en kantdeteksjon av valgt bilde.
- PossibleArtefact.
 - Dette filteret vil prosessere valgte bilde og loggføre om dette har et funn.

Felles for alle filtre er at bildene først vil bli hentet til lokal node, for så å opprette tilsvarende Matlab jobb ved å eksekvere tilhørende webtjenesten.

5.2. Matlab

Systemet har blant annet som formål å kunne tilby DICOM tjenester for Matlab. Matlab R2007b har ifølge Matlab sitt eget hjelpeprogram ingen støtte for DICOM SCP eller SCU tjenester. Systemet som er utviklet i denne oppgaven vil indirekte gi Matlab disse tjenestene.

Systemet har støtte for å initiere og kjøre Matlab funksjoner (m-filer) over det COM grensesnittet som tilbys. I denne oppgaven er det utarbeidet seks forskjellige demonstrasjonsfiltre, som har til hensikt å belyse noen av mulighetene. Filtrene er i sin helhet vedlagt denne rapporten. Nedenfor gis en kort beskrivelse av filtrene.

5.2.1. Matlab Filter 1. EdgeDetectionWriteBack

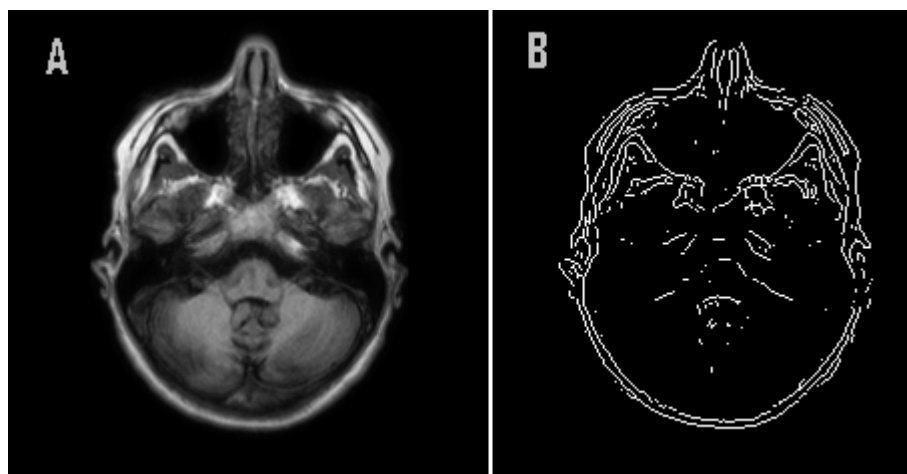
Denne funksjonen har til hensikt å demonstrere hvordan Matlab kan programmeres til å lese ut alle DICOM bildefiler. Det gjøres en bildemanipulasjon i form av en kantdetektering, og det genereres ut et nytt bilde i en ny serie, som har en annen unik *SerieInstanceUID*.

Figur 30 viser hvordan dette filteret har prosessert bildene i serien fra kolonne tre, og har laget en ny serie i kolonne to.

Series Number	Series Description	Series Date	Series Time	Modality	Images
103	Patient Aligned MPR	2009-05-12	11:50:50	MR	54
301	Created by Matlab	2009-05-12	11:52:04	MR	110
301	DWI HR	2009-05-12	11:52:04	MR	110

Figur 30. Viser i testpacs at ny serie er opprettet. I rad to vises den nye serien som er opprettet som en prosessert versjon av rad tre. Serienummerbeskrivelse, tidsangivelser, type og antall bilder er identisk. Seriebeskrivelsen i kolonne to er endret med den teksten som vi har oppdatert på bildene i serien.

Figur 31, panel A og B viser den påvirkningen som dette filteret har påført det originale bildet. Matlabfunksjonen leser inn informasjonen fra DICOM bildet, og finner grenseverdiene for de forskjellige tetthetene i bildet. Disse tetthetene blir markert i et nytt DICOM bilde som vises i panel B.



Figur 31. Panel A viser originalbilde fra MR undersøkelse av hjerne (aksialsnitt). Panel B viser bildet etter kantdeteksjon med filteret `EdgeDetectionWriteBack`.

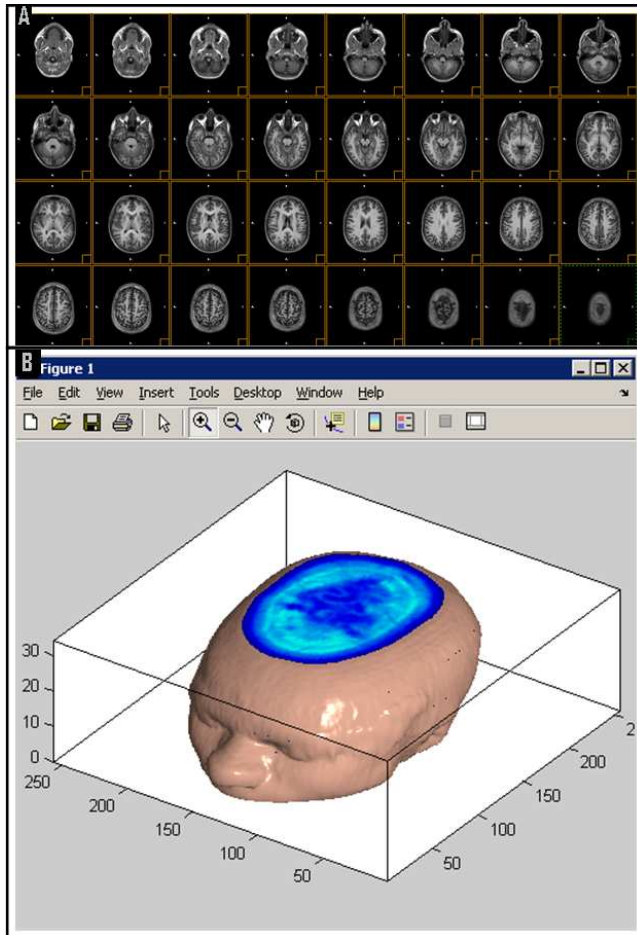
5.2.2. Matlab Filter 2. *SineScrollDemo*

Dette filteret har som hensikt å demonstrere mulighetene som er tilgjengelig i Matlab Image Toolbox. Ved å kjøre dette filteret vil bildene som er tilgjengelig i valgt mappe bli vist på serversiden. Bildene vises et og et av gangen.

5.2.3. Matlab Filter 3. *MR_VolumeRender*

`MR_VolumeRender` er en mer kompleks filtervariant. Dette filteret tar utgangspunkt i beskrivelse fra Matlab sitt hjelpeprogram, og er implementert som en dynamisk variant av dette [7]. Filteret leser bildene sekvensielt i angitt mappe. Bildene analyseres og en finner kanten av bildeinformasjonen. Denne kanten påføres et dekkende lag av oransje farge. Deretter leses innholdet i bildesnittet, og tettheten som representeres i bildet får påført en farge som samsvarer til denne tettheten. På denne måten blir det enklere å lokalisere forskjellig tetthetsmasser. Til slutt adderes alle bildene sammen, og det tegnes et volum med de snittene som er tilgjengelig.

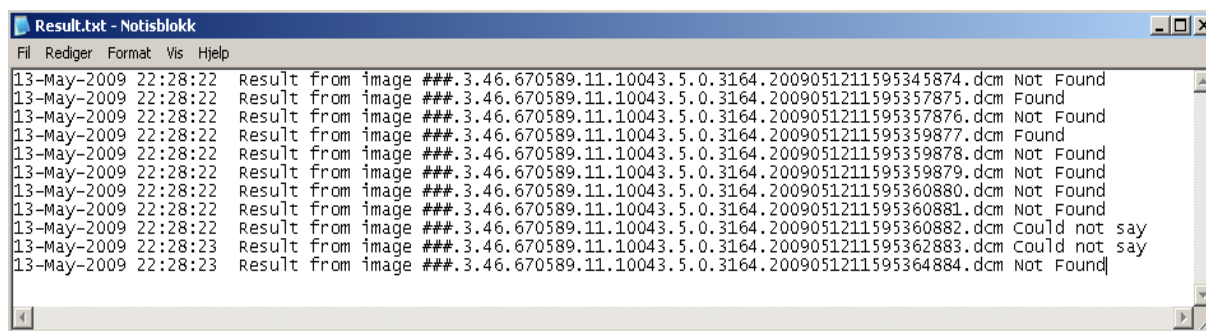
Figur 32, panel A viser de tilgjengelige bildene fra den originale MR bildeserien, mens Panel B viser det genererte 3D volumet.



Figur 32. Panel A viser originale MR bilder fra et aksialt snitt av hjerne. Panel B viser det genererte 3D volum av de originale MR-snittene i panel A ved å påføre MR_VolumeRender.

5.2.4. Matlab Filter 4. PossibleArtefactSerie

Dette filteret har til hensikt å belyse hvordan den utviklete funksjonen ArtInSerie.m i Matlab kan benyttes til å skanne store serier. Ved å benytte denne funksjonen kan Matlab analysere de enkelte snitt etter en algoritme, og skrive resultatet ut til en fil. Figur 33 viser eksempel på loggfil fra en ferdig analysert serie. Loggfilen har en linje for hvert bilde som blir analysert. I første del av linjen vises tidspunkt for analyse. Videre vises logiske identifikatorer til DICOM filen, og i siste del vises resultatet av prosesseringen. I dette demonstrasjonsfilteret er det bare satt opp en sannsynlighetsmodell som kan gi tre forskjellige utfall. Dette er "Not Found", "Found" og "Could not say". Ideen med denne modellen er at brukere av systemet kan legge inn sin logikk og analyse for å markere sine funn.



Figur 33. Utdrag av Result log for PossibleArtefactSerie. Hvert prosessert bilde vil ha sin egen linje i denne filen. Første del av en linje vil angi tidsangivelse for prosessering. Videre vil linjen inneholde StudyInstanceUID, SerieInstanceUID og ImageUID for bildet. Videre vil resultat av analysen for dette bildet bli lagt inn. Her er verdiene "Not Found", "Found" og "Could not say" tilgjengelig. Figuren viser bare et utsnitt av loggfilen. Visning av StudyInstanceUID og SerieInstanceUID er utelatt for å gjøre innholdet i loggfilen mer lesbart.

5.2.5. Matlab Filter 5. MyEdgeFilterDemo

Dette filteret utfører en kantdetektering av et enkelt bilde. Dette filteret er laget for å vise hvordan et enkelt bilde ser ut originalt, mens det etter 5 sekunders pause vises med påført edge funksjon fra Matlab.

5.2.6. Matlab Filter 6. PossibleArtefact

Dette filteret er som PossibleArtefactSerie. Forskjellen er at dette filteret kjører algoritmen på et enkelt snitt. Resultatet skrives inn i samme fil som PossibleArtefactSerie.

5.3. Responstider

5.3.1. Responstider for henting av bilder

Tabell 2 viser responstidene for å hente ut bilder fra produksjonssystemet for henholdsvis EFILM som er et kommersielt produkt, og ApplicationSCP utviklet i denne oppgaven.

Tabell 2. Målt responstid mellom kommersiell SCP og ApplicationSCP. Tabellen viser responstid i sekunder for å hente angitt antall bilder til lokalt lagringsområde. Type modalitet bildene er tatt på er angitt.

Type modalitet	CR	CR	MG	MG	UL	UL	MR	CT	CT	MR
Antall bilder	2	2	8	8	9	19	173	200	385	443
EFILM	3	3	7	8	4	7	9	13	31	17
ApplicationSCP	2	2	7	8	3	7	10	14	33	20

Som en ser av tabell 2 er det en noe bedre overføringshastighet ved å benytte utviklet løsning i forhold til EFILM som et kommersielt produkt. Dette gjør seg gjeldene om antall bilder som overføres er mindre enn 9 bilder. Ved større antall bilder i en overførsel ser en at overføringstiden til EFILM er noe bedre. Dette som en følge av at EFILM bare har en direkte DICOM kommunikasjon mot den andre parten, og ikke et webtjenestelag som påfører overførselen tidsforbruk.

5.3.2. Interne responstider i henting av DICOM informasjon over webtjenestene

Tabell 3 viser resultatet av målinger opp mot antall DICOM objekter som XML dokumentet returnerer for å generere et resultat for brukeren. Tabellen viser også forholdstallet mellom responstid og antall DICOM objekter.

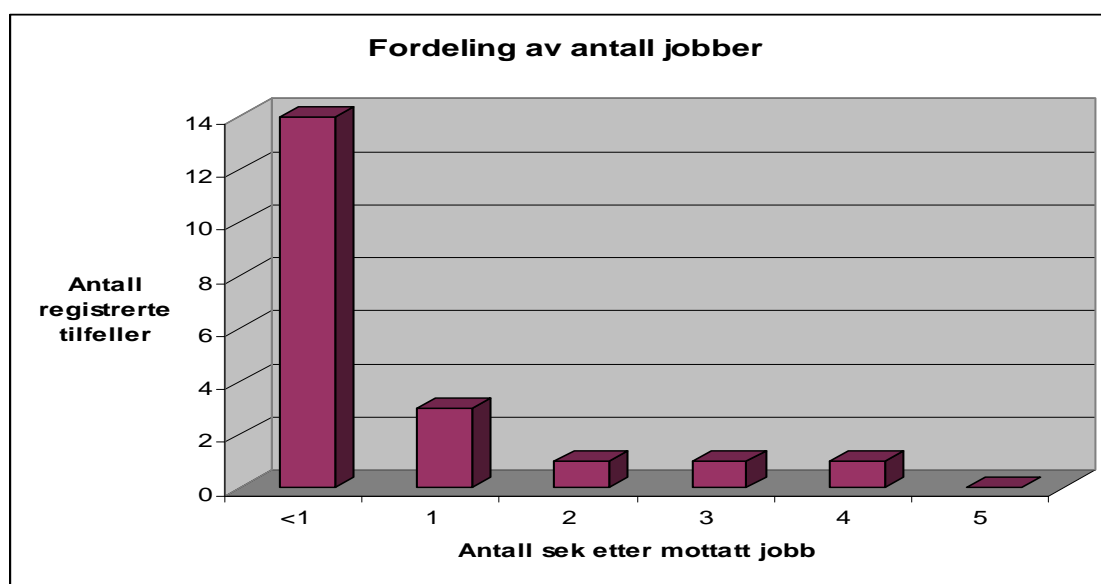
Tabell 3. Interne responstider for henting av DICOM informasjon over webtjenestene. Tabellen viser noen utvalgte opphentinger av bilder og den tiden dette behøver. Her kan en se at når antall objekter som skal legges inn i XML dokumentet øker, bli den totale ytelsen påvirket av kjøremiljøet slik at ytelsen pr bilde blir noe forverret. Det totale tidsforbruket for å hente slike store bildeinformasjonselementer er selv ved denne ytelsen marginal og burde ikke gi en bruker en opplevd forsinkelse i bruk av webtjenestene.

Antall DICOM objekt	0	0	5	10	15	36	51	84	279	489
mSek	172	348	219	172	172	219	375	1403	828	1250
mSek/objekt	0	0	0,023	0,058	0,087	0,164	0,136	0,06	0,337	0,391

Tabell 3 viser interne responstider for henting av DICOM informasjon over webtjenestene. Opphenting av et utvalg av undersøkelser gav disse tidene. Dersom en fordeler total tid for hver overførsel mot antall bilder ser en at responstiden øker noe for større undersøkelser med flere bilder. Dette er utslag av den *DICOM acknowledge* som utveksles mellom aktørene, men også på grunn av intern diskaktivitet (*cacheing/buffering*).

5.3.3. Interne responstider i MatlabWrapper

I målingen av responstidene til MatlabWrapper ble det sendt arbitrært filterjobber til grensesnittet. Ut av 20 jobber som ble sendt til MatlabWrapper, ble de fleste igangsatt etter en relativ kort tid. Median for igangsetting av disse jobbene lå på 0,3 sekunder. Figur 34 viser fordelingen av starttid som disse 20 jobbene hadde.



Figur 34. Fordeling av tider mot responstid for MatlabWrapper. Av 20 jobber som ble registrert i grensenettet ble størsteparten av jobbene igangsatt under et sekund. Noen av jobbene fikk en lengre igangsettingstid, noe som trolig skyldes at maskinen som har grensenettet hadde andre aktive prosesser i øyeblikket som krevde maskinressurser, og som gav en reel forsinkelse.

6. DISKUSJON

I utviklingen av løsninger som benytter medisinske bilder blir det stort sett benyttet konvensjonell DICOM 3.0 kommunikasjon i dag. Dette er en tidkrevende plattform å utvikle på, der utviklerne må ha god kjennskap til både DICOM 3.0 protokoll og til konfigurering av DICOM løsninger. Systemet som er utviklet og presentert i denne rapporten er ment å kunne belyse noen av de mulighetene som en tjenesteorientert løsning representerer, samt på sikt være en kandidat til et rammeverk som integrerer DICOM 3.0, medisinske bilder og kommunikasjon på en standardisert måte. Dette rammeverket kan da være en plattform for å kunne utvikle nye løsninger og produkter på en mer agil måte.

Webgrensesnittene og programmet StudyCandidates som er utviklet i denne oppgaven er laget spesifikt for å automatisere uthenting og videreprosessering av medisinske bilder som i dag er en svært tidkrevende jobb i flere forskningsprosjekter ved Stavanger Universitetssjukehus. Forskerne klarer fint å hente ut et utvalg av undersøkelser, men å hente spesifikke serier i disse undersøkelsene er en manuell og til dels tidkrevende jobb, som også representerer en risiko for feil. Bruk av en automatisert tjeneste ville minimert risikoen for denne typen feil.

I følgende diskusjon vil jeg prøve å se på en mulig presisering og videreutvikling av de to hovedkomponentene som inngår i den presenterte løsningen, og som skjematisk er presentert i figur 9 som webtjenestelag henholdsvis mot DICOM og mot Matlab.

Webtjenestene under DICOM kunne vært implementert mer entydig ved å benytte etablert DICOM navngiving for tjenestene. I stedet for WS_RetrieveStudy burde den vært implementert som eksempelvis WS_C_STORE. På denne måte ville bruken av tjenestene vært mer intuitive for andre utviklere.

I arbeidet med DICOM kommunikasjonen ble det prøvd ut OpenDICOM-sharp som SDK. Denne pakken ble forkastet til fordel for ClearCanvas. Det er meget mulig at konfigurering av parametre tilgjengelig i klassene til OpenDICOM-sharp ikke ble utført på en optimal måte og at denne implementeringen av DICOM har den nødvendige ytelsen som kreves.

Sammen med webgrensesnittet er det i løsningen implementert en egen SCP applikasjon, som er en noe modernisert utgave av testapplikasjon til ClearCanvas. Dette er strengt tatt ikke nødvendig. Det er tilgjengelig flere kommersielle applikasjoner med SCP støtte som eventuelt kan benyttes i løsningen. Forutsetningen er at det da blir implementert støtte for å generere en Matlab jobb etter at bildene er tilgjengelig til bruk.

I denne oppgaven er det bare forsøkt å benytte basis DICOM 3.0 tjenester. Det vil være mulig å implementere mer utvidede DICOM tjenester over samme grensesnittet. Dette kunne være tjenester som *DICOM Storage Commit* og *DICOM Modality Performed Procedure Step* for å nevne noen.

Videre i figur 9 kan en se presentert det webtjenestegrensesnittet som ivaretar Matlab kommunikasjonen. På Matlab grensesnittet er det implementert løsninger som initierer jobber til Matlab ved å bruke en Inbound/Outbound mappe struktur. Det kunne vært interessant å implementere dette som en *socket stream*. På denne måten kunne en kommunikasjon mellom

DICOM og Matlab COM bli uavhengig av en direkte fil-kommunikasjon. *Socket* er hyppig benyttet av andre aktører som er en del av den utvidete DICOM familien, og meldingstjenester som HL7 benytter ofte *socket* kommunikasjon.

MatlabWrapper har store muligheter. I dag er det bare implementert en løsning som kaller funksjonsfiler over COM grensesnittet. Dette grensesnittet har veldig mye mer å by på. Et eksempel er mulighet til å kommunisere arrayene til og fra Workbench hos Matlab. På denne måten kunne en ha integrert lengre inn i Matlab, og interne matriser med informasjon i Matlab kunne blitt tilgjengelig for utviklerne.

StudyCandidates, som er *front-end* og testapplikasjonen som er presentert i denne rapporten, har absolutt rom for endringer. Applikasjonen er basert på synkron kommunikasjon mot DICOM, og GUI vil være låst til en undersøkelse er overført om en benytter *WS_RetrieveStudy*. Synkron kommunikasjon vil i klinisk drift ikke være optimal, da brukere forventer å få en kontinuerlig tilbakemelding på statusforløp. SDK til ClearCanvas har egenskaper som muliggjør en asynkron kommunikasjon. En alternativ måte å løse dette på kunne vært å instansiere webtjenestene ved å bruke en egen *thread*.

Responstidene til systemet viser at påbygningen av webtjenester på DICOM protokollen ikke representerer en stor økning i overføringstider. Overføringstidene i tabell 2 viser at selv ved relativt mange bilder er tillegget som webtjenestene representerer marginale. For undersøkelser med 443 bilder vil denne økningen være fra 17 til 20 sekunder. Dette åpner for muligheten til å kunne tilby tjenester som kan instansieres direkte fra en diagnostisk arbeidsstasjon, med tilhørende resultat på en synkron måte. Den største begrensningen i en slik konfigurasjon er relatert til selve ytelsen til filteret. Dette er en av grunnene til at webtjenestegrensesnittet i oppgaven er fordel på to logiske servere. På denne måten kan en ha et grensesnitt som andre får tilgang til, mens Matlabgrensesnittet er tilgjengelig kun for systemet selv. Dette gjør at grensesnittet mot Matlab relativt enkelt kan settes ut som en asp løsning. En av utfordringene med en slik forming av løsningen er rettet mot den fysiske tilgangen til bildene. En interessant utvikling av kommunikasjonen kunne være å *stream* bildeobjektene i en webtjeneste, og ikke som en løsning der et felles filområde benyttes slik det er implementert i løsningen i dag. Microsoft har tilgjengelig et tillegg til webtjenesteplattformen sin, som har utvidete muligheter til å *stream* bildeobjekter som webtjenester. Dette tillegget heter WSE, Web Security Enhancement. En av de mest interessante egenskapene til dette tillegget er at det også kan implementeres med MTOM som kan lese store binærfiler og stream dem som webtjenester. Dette er utviklet for å skape en hurtigere utveksling av slik informasjon [8].

Responstidene til webtjenestene utviklet i denne oppgaven er ytelsemessig bedre enn målt utveksling ved hjelp av ren DICOM når det er snakk om overføring av et mindre antall bilder (tabell 2). Disse responstidene har en krysning i området 5-10 bilder, og overføringen har etter denne grensen bare en marginalt dårligere ytelse for overføring av flere bilder. Grunnen til dette antas å være at EFILM SCP annonserer støtte for flere tjenester og støtte for flere SOP klasser. Dermed vil selve *DICOM Associate* prosessen ta noe lengre tid. Dersom det er snakk om overføring av flere bilder vil det tillegget som webtjenesten representerer flytte den totale ytelsen til å bli noe dårligere. Reduksjon i ytelse er cirka 10 % ved overføring av store bildemengder. Denne reduksjonen er for en bruker tilnærmet neglisjerbar da en overføring på 10 sekund for en vanlig DICOM overføring vil bli overført på 11 sekund ved bruk av webtjenestene. Fordeles økningen i tid på alle bildene som blir hentet, vil den reduksjonen i ytelse som hvert enkelt bilde har, ligge på cirka 0,005 sekund.

For å understøtte sporbarhet i bruken av systemet, som er en nødvendighet i en klinisk arbeidssituasjon, kunne grensesnittene hatt egne databaser til lagring av informasjon om bruk og status til systemet. Det er valgt å ikke inkludere en slik database i denne løsningen for heller å holde fokus på grensesnittene. Videre kunne en løsning som er presentert i denne oppgaven blitt utvidet til å kunne kommunisere mot et radiologisk informasjonssystem(RIS) og et elektronisk Journal System (EPJ). Ved å legge til et utvidet webgrensenitt som kunne kommunisere mot disse produktene, kunne en ha oppnådd en løsning som skissert i figur 11. Dersom disse produktene kunne annonsere og konsumere hendelser i de respektive systemene på en ESB, kunne en ved hjelp av orkestrering automatisert forretningsprosessene og skapt en produktiv dynamikk mellom disse selvstendige komponentene.

StudyCandidates tar utgangspunkt som en ad-hoc løsning der brukerne som i denne oppgaven er tenkt å være forskere ved Stavanger Universitetssjukehus, skal generere et representativt utvalg som skal bearbeides av Matlab. Det trenger nødvendigvis ikke bare være slik. Det grensesnittet som er utviklet tar høyde for at det skal være mulig å utvikle en applikasjon som logisk kan bli plassert mellom modalitet og lagringsenhet. På denne måten kan en typisk QA (QualityAssurance) tjeneste bli tilgjengelig. Denne kunne eksempelvis mottat alle bilder fra en gitt type undersøkelse, plukke ut en spesifikk serie som igjen kunne blitt videreprosessert. Resultatet av en slik prosessering kan lagres som en ny serie til hjelp i vurdering av undersøkelsen hos radiolog.

Integrasjonsplattformen som er utviklet i denne oppgaven kan benyttes til å utvikle andre relaterte produkter. Et annet viktig område dette kunne benyttes på er kvalitetssikring av de undersøkelsene som er utført. Dersom det utvikles filtre som kan gjenkjenne spesifikke abnormaliteter, kan multiple undersøkelser bli skannet som en maskinell "Second opinion". Negative undersøkelser kunne da bli plukket ut for en ny vurdering, noe som vil redusere feilvurdering.

En mulig anvendelse av løsningene som er presentert i denne rapporten kunne være at filtre som ble utviklet av medisinsk personell i Matlab kunne bli testet ut på historisk data av et større omfang enn hva som er praktisk mulig i dag. Andre interessante filtre kunne være postprosesseringsfiltre til hjelpe for radiolog i vurdering av bildemassen. Som eksempel kan nevnes at ved CT-undersøkelser blir rådata påvirket av metalliske gjenstander (for eksempel hofteproteser i metall) noe som skaper artefakter og forstyrrelser i bildene. Disse artefaktene er til stor forstyrrelse i granskningen av bilder og kan kamuflere patologi. Dersom det var mulig å gjenkjenne disse artefaktene med et filter, kunne en se for seg en løsning der det automatisk ble generert en ny serie som tok bort eller kamuflerte disse artefaktene slik at radiolog kunne vurdere bildene uten å bli påvirket av disse artefaktene.

Den tjenesteorienterte løsningen som er presentert i denne oppgaven viser hvordan webtjenester kan brukes for automatisk prosessering av DICOM medisinske bilder. Løsningen rommer store muligheter for videreutvikling og tilpasning til ulike problemstillinger der DICOM medisinske bilder inngår.

7. KONKLUSJON

I innledningen ble det listet opp noen utfordringer som er forsøkt løst i denne oppgaven.

- Vil det være mulig å løsrive DICOM kommunikasjon fra sin tradisjonelle punkt-til-punkt konfigurasjon inn i en mer tjenesteorientert arkitektur med de egenskapene det medbringer?

Webtjenestegrensesnittet og tilhørende testapplikasjon (StudyCandidates) som er utviklet i denne oppgaven viser at det er mulig å eksponere basis DICOM tjenester som webtjenester. Grensesnittet kan videreutvikles til også å eksponere andre DICOM tjenester.

- Vil det være mulig å skape et eget tjenesteorientert grensesnitt mot et noe eldre COM grensesnitt representert i denne oppgaven som Matlab?

Det er mulig å generere et webtjenestegrensesnitt mot COM grensesnittet til Matlab 2007b. Det er viktig å merke seg at det bør lages en wrappertjeneste på serveren som har webmotoren til å holde kommunikasjonen mot Matlab oppe. Responstidene vil øke drastisk om webtjenestene skal instansiere grensesnittet hver gang, og løsningen ville da i prinsippet ikke kunne bli akseptert i en klinisk arbeidsflyt.

- Vil det ved å bruke en slik tjenesteorientert tankemåte og implementering være mulig å skape en fornuftig integrasjon og automasjon for å løse en spesifikk utfordring?

Ut av resultatene i punkt en og to er det også mulig å skape fungerende applikasjoner og tjenester ved å knytte disse teknologiene sammen. Webtjenester og tjenesteorientert arkitektur er et stort område, men det kan i denne rapporten ikke påpekes grunner til å ekskludere spørremeldinger eller abonneringer av meldinger i samspillet mellom DICOM og tjenesteorientert arkitektur.

- Vil summen av disse resultatene belyse om det på sikt kan være mulig å løsrive seg fra den konvensjonelle kommunikasjonsplattformen som DICOM har i industrien i dag, til å tilby mer tjenesteorienterte løsninger uten å måtte omdefinere en hel industri?

Løsningen som er utviklet i denne oppgaven og ytelsesmålingene til webtjenestene, viser at det er mulig å tilby et grensesnitt med webtjenester for DICOM 3.0. Utbredelsen av DICOM 3.0 gjør at en transisjon til ren tjenesteorientering ikke er reelt på nåværende tidspunkt.

8. REFERANSER / KILDEHENVISNINGER

- [1] Ukjent forfatter Informasjonsside Radiografforbundet, Fag og Yrke [online].
http://www.radiograf.no/modules/module_123/proxy.asp?D=2&C=12&I=79&mids=17
- [2] Ukjent forfatter Informasjonsside KITH, Forside >> Nasjonale oppgaver >> PACS [online]
http://www.kith.no/templates/kith_WebPage_549.aspx
- [3] Ukjent forfatter, Informasjonsside RSNA [online]
<http://www.rsna.org/Technology/DICOM/>
- [4] Ukjent forfatter, Wikipedia [online]
http://en.wikipedia.org/wiki/Digital_Imaging_and_Communications_in_Medicine
- [5] Michael P. Papazoglou, Web services: Principles and technology.
ISBN: 978-0-321-15555-9
- [6] Emanuele Ruffaldi, 1..2..3 ways of integrating MATLAB with the .NET [online]
<http://www.codeproject.com/KB/dotnet/matlabeng.aspx>
- [7] Matlab Documentation - Techniques for Visualizing Scalar Volume Data
<http://www.mathworks.com/access/helpdesk/help/techdoc/index.html?/access/helpdesk/help/techdoc/visualize/f5-3291.html>
- [8] SOAP Message Transmission Optimization Mechanism
<http://www.w3.org/TR/soap12-mtom/>

PROGRAMMERINGSLINKER INSPIRERT I UTVIKLINGEN

ClearCanvas hjemmeside[online]
<http://www.clearcanvas.ca/dnn/>

C# - Check if a file is in use file, check, use
http://www.experts-exchange.com/Programming/Languages/C_Sharp/Q_22952790.html

C# ContextMenuStrip Example
<http://dotnetperls.com/Content/ContextMenuStrip-Example.aspx>

CodeGuru Using FileSystemWatcher
http://www.codeguru.com/csharp/csharp/cs_network/article.php/c6043

CodeProject .NET COM+ Interop Component with Classic ASP (Part I). Free source code and programming help
http://www.codeproject.com/KB/vb-interop/Net_COM_Interop.aspx

DotNetTech - Article
<http://www.dotnettech.net/BCL/Article.aspx?ArticleID=27680&Url=http%3a%2f%2fforums.microsoft.com%2fMSDN%2fShowPost.aspx%3fPostID%3d2870647%26SiteID%3d1>

How to Add Menu Items to a ContextMenuStrip
<http://msdn.microsoft.com/en-us/library/ms229626.aspx>

HOW TO Use XmlDocument Elements When Passed to or Returned from WebMethods by Using Visual C# .NET

<http://support.microsoft.com/kb/330600>

kirupa.com - Reading XML Files Directly, Page 1

http://www.kirupa.com/net/reading_xml_directly_pg1.htm

Log4Net Tutorial in C# .net (How can I show log in a file) « Sadi02's Weblog

<http://sadi02.wordpress.com/2008/06/29/log4net-tutorial-in-c-net-how-can-i-show-log-in-a-file/>

MATLAB - Documentation

<http://www.mathworks.com/access/helpdesk/help/techdoc/index.html?/access/helpdesk/help/techdoc/ref/what.html&http://www.google.no/search?hl=no&q=matlab+read+files+in+directory&meta=&aq=5&oq=matlab+read>

Notifyicon with contextmenu and no form.

<http://social.msdn.microsoft.com/forums/en-US/Vsexpressvcs/thread/bc4556d5-3cb4-458b-96be-6f51c68700fd/>

Opening form instances in C#

http://articles.techrepublic.com.com/5100-10878_11-1050650.html

http://www.techrepublic.com.com/5100-10878_11-1050650.html

<http://www.dotnetjunkies.com/WebLog/razor/archive/2004/02/09/6810.aspx>

Release Note Report

http://www.mathworks.com/access/helpdesk/help/bugfixes_13.html

Store .NET application settings in custom XML .config files - Program - Web Technologies - Builder AU

<http://www.builderau.com.au/program/web/soa/Store-NET-application-settings-in-custom-XML-config-files/0.339024632.320265625.00.htm>

Test Run Automate Your ASP.NET Web Services Testing

<http://msdn.microsoft.com/en-us/magazine/cc163830.aspx>

9. VEDLEGG

9.1. WSDL for DICOM Webtjenester

```

<?xml version="1.0" encoding="utf-8" ?>
- <wsdl:definitions xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
xmlns:tm="http://microsoft.com/wsdl/mime/textMatching/"
xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/"
xmlns:mime="http://schemas.xmlsoap.org/wsdl/mime/"
xmlns:tns="http://tempuri.org/"
xmlns:s="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://schemas.xmlsoap.org/wsdl/soap12/"
xmlns:http="http://schemas.xmlsoap.org/wsdl/http/"
targetNamespace="http://tempuri.org/"
xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/">
- <wsdl:types>
- <s:schema elementFormDefault="qualified"
targetNamespace="http://tempuri.org/">
- <s:element name="HelloWorld">
<s:complexType />
</s:element>
- <s:element name="HelloWorldResponse">
- <s:complexType>
- <s:sequence>
<s:element minOccurs="0" maxOccurs="1"
name="HelloWorldResult" type="s:string" />
</s:sequence>
</s:complexType>
</s:element>
- <s:element name="WS_GetNrOfPatients">
- <s:complexType>
- <s:sequence>
<s:element minOccurs="0" maxOccurs="1" name="PID"
type="s:string" />
</s:sequence>
</s:complexType>
</s:element>
- <s:element name="WS_GetNrOfPatientsResponse">
- <s:complexType>
- <s:sequence>
<s:element minOccurs="0" maxOccurs="1"
name="WS_GetNrOfPatientsResult" type="s:string" />
</s:sequence>
</s:complexType>
</s:element>
- <s:element name="WS_GetPersonStudies">
- <s:complexType>
- <s:sequence>
<s:element minOccurs="0" maxOccurs="1" name="PID"
type="s:string" />
</s:sequence>
</s:complexType>
</s:element>
- <s:element name="WS_GetPersonStudiesResponse">
- <s:complexType>

```



```
- <s:sequence>
- <s:element minOccurs="0" maxOccurs="1"
name="WS_GetPersonStudiesResult">
- <s:complexType mixed="true">
- <s:sequence>
<s:any />
</s:sequence>
</s:complexType>
</s:element>
</s:sequence>
</s:complexType>
</s:element>
- <s:element name="WS_GetStudyDetails">
- <s:complexType>
- <s:sequence>
<s:element minOccurs="0" maxOccurs="1"
name="StudyInstanceUID" type="s:string" />
</s:sequence>
</s:complexType>
</s:element>
- <s:element name="WS_GetStudyDetailsResponse">
- <s:complexType>
- <s:sequence>
- <s:element minOccurs="0" maxOccurs="1"
name="WS_GetStudyDetailsResult">
- <s:complexType mixed="true">
- <s:sequence>
<s:any />
</s:sequence>
</s:complexType>
</s:element>
</s:sequence>
</s:complexType>
</s:element>
- <s:element name="WS_GetSerieDetails">
- <s:complexType>
- <s:sequence>
<s:element minOccurs="0" maxOccurs="1"
name="StudyInstanceUid" type="s:string" />
<s:element minOccurs="0" maxOccurs="1"
name="SerieInstanceUID" type="s:string" />
</s:sequence>
</s:complexType>
</s:element>
- <s:element name="WS_GetSerieDetailsResponse">
- <s:complexType>
- <s:sequence>
- <s:element minOccurs="0" maxOccurs="1"
name="WS_GetSerieDetailsResult">
- <s:complexType mixed="true">
- <s:sequence>
<s:any />
</s:sequence>
</s:complexType>
</s:element>
</s:sequence>
</s:complexType>
```

```
</s:element>
- <s:element name="WS_RetrieveStudy">
- <s:complexType>
- <s:sequence>
<s:element minOccurs="0" maxOccurs="1"
name="StudyInstanceUID" type="s:string" />
</s:sequence>
</s:complexType>
</s:element>
- <s:element name="WS_RetrieveStudyResponse">
- <s:complexType>
- <s:sequence>
<s:element minOccurs="0" maxOccurs="1"
name="WS_RetrieveStudyResult" type="s:string" />
</s:sequence>
</s:complexType>
</s:element>
- <s:element name="WS_RetrieveSerie">
- <s:complexType>
- <s:sequence>
<s:element minOccurs="0" maxOccurs="1"
name="SerieInstanceUID" type="s:string" />
</s:sequence>
</s:complexType>
</s:element>
- <s:element name="WS_RetrieveSerieResponse">
- <s:complexType>
- <s:sequence>
<s:element minOccurs="0" maxOccurs="1"
name="WS_RetrieveSerieResult" type="s:string" />
</s:sequence>
</s:complexType>
</s:element>
- <s:element name="WS_RetrieveImage">
- <s:complexType>
- <s:sequence>
<s:element minOccurs="0" maxOccurs="1" name="ImageUid"
type="s:string" />
</s:sequence>
</s:complexType>
</s:element>
- <s:element name="WS_RetrieveImageResponse">
- <s:complexType>
- <s:sequence>
<s:element minOccurs="0" maxOccurs="1"
name="WS_RetrieveImageResult" type="s:string" />
</s:sequence>
</s:complexType>
</s:element>
- <s:element name="WS_StoreSerie">
- <s:complexType>
- <s:sequence>
<s:element minOccurs="0" maxOccurs="1"
name="StudyInstanceUID" type="s:string" />
<s:element minOccurs="0" maxOccurs="1"
name="SerieInstanceUID" type="s:string" />
</s:sequence>
```

```

</s:complexType>
</s:element>
- <s:element name="WS_StoreSerieResponse">
- <s:complexType>
- <s:sequence>
<s:element minOccurs="0" maxOccurs="1"
name="WS_StoreSerieResult" type="s:string" />
</s:sequence>
</s:complexType>
</s:element>
- <s:element name="WS_GetStudiesCandidates">
- <s:complexType>
- <s:sequence>
<s:element minOccurs="0" maxOccurs="1"
name="studyRange" type="s:string" />
<s:element minOccurs="0" maxOccurs="1" name="Modality"
type="s:string" />
<s:element minOccurs="0" maxOccurs="1"
name="StudyDescription" type="s:string" />
<s:element minOccurs="0" maxOccurs="1" name="Sex"
type="s:string" />
</s:sequence>
</s:complexType>
</s:element>
- <s:element name="WS_GetStudiesCandidatesResponse">
- <s:complexType>
- <s:sequence>
- <s:element minOccurs="0" maxOccurs="1"
name="WS_GetStudiesCandidatesResult">
- <s:complexType mixed="true">
- <s:sequence>
<s:any />
</s:sequence>
</s:complexType>
</s:element>
</s:sequence>
</s:complexType>
</s:element>
- <s:element name="WS_SVGDEMO2_WS_MatlabSerieJob">
- <s:complexType>
- <s:sequence>
<s:element minOccurs="0" maxOccurs="1"
name="StudyInstanceUID" type="s:string" />
<s:element minOccurs="0" maxOccurs="1"
name="SerieInstanceUID" type="s:string" />
<s:element minOccurs="0" maxOccurs="1"
name="FilterName" type="s:string" />
<s:element minOccurs="0" maxOccurs="1" name="FilterLevel"
type="s:string" />
</s:sequence>
</s:complexType>
</s:element>
- <s:element name="WS_SVGDEMO2_WS_MatlabSerieJobResponse">
- <s:complexType>
- <s:sequence>
<s:element minOccurs="0" maxOccurs="1"
name="WS_SVGDEMO2_WS_MatlabSerieJobResult"

```

```

type="s:string" />
</s:sequence>
</s:complexType>
</s:element>
- <s:element name="WS_SVGDEMO2_WS_MatlabImageJob">
- <s:complexType>
- <s:sequence>
<s:element minOccurs="0" maxOccurs="1"
name="StudyInstanceUID" type="s:string" />
<s:element minOccurs="0" maxOccurs="1"
name="SerieInstanceUID" type="s:string" />
<s:element minOccurs="0" maxOccurs="1" name="ImageUid"
type="s:string" />
<s:element minOccurs="0" maxOccurs="1"
name="FilterName" type="s:string" />
<s:element minOccurs="0" maxOccurs="1" name="FilterLevel"
type="s:string" />
</s:sequence>
</s:complexType>
</s:element>
- <s:element name="WS_SVGDEMO2_WS_MatlabImageJobResponse">
- <s:complexType>
- <s:sequence>
<s:element minOccurs="0" maxOccurs="1"
name="WS_SVGDEMO2_WS_MatlabImageJobResult"
type="s:string" />
</s:sequence>
</s:complexType>
</s:element>
- <s:element name="WS_SVGDEMO2_WS_AvailableFilter">
<s:complexType />
</s:element>
- <s:element name="WS_SVGDEMO2_WS_AvailableFilterResponse">
- <s:complexType>
- <s:sequence>
- <s:element minOccurs="0" maxOccurs="1"
name="WS_SVGDEMO2_WS_AvailableFilterResult">
- <s:complexType mixed="true">
- <s:sequence>
<s:any />
</s:sequence>
</s:complexType>
</s:element>
</s:sequence>
</s:complexType>
</s:element>
</s:schema>
</wsdl:types>
- <wsdl:message name="HelloWorldSoapIn">
<wsdl:part name="parameters" element="tns:HelloWorld" />
</wsdl:message>
- <wsdl:message name="HelloWorldSoapOut">
<wsdl:part name="parameters" element="tns:HelloWorldResponse" />
</wsdl:message>
- <wsdl:message name="WS_GetNrOfPatientsSoapIn">
<wsdl:part name="parameters" element="tns:WS_GetNrOfPatients" />
</wsdl:message>

```

```
- <wsdl:message name="WS_GetNrOfPatientsSoapOut">
<wsdl:part name="parameters"
element="tns:WS_GetNrOfPatientsResponse" />
</wsdl:message>
- <wsdl:message name="WS_GetPersonStudiesSoapIn">
<wsdl:part name="parameters" element="tns:WS_GetPersonStudies" />
</wsdl:message>
- <wsdl:message name="WS_GetPersonStudiesSoapOut">
<wsdl:part name="parameters"
element="tns:WS_GetPersonStudiesResponse" />
</wsdl:message>
- <wsdl:message name="WS_GetStudyDetailsSoapIn">
<wsdl:part name="parameters" element="tns:WS_GetStudyDetails" />
</wsdl:message>
- <wsdl:message name="WS_GetStudyDetailsSoapOut">
<wsdl:part name="parameters"
element="tns:WS_GetStudyDetailsResponse" />
</wsdl:message>
- <wsdl:message name="WS_GetSerieDetailsSoapIn">
<wsdl:part name="parameters" element="tns:WS_GetSerieDetails" />
</wsdl:message>
- <wsdl:message name="WS_GetSerieDetailsSoapOut">
<wsdl:part name="parameters"
element="tns:WS_GetSerieDetailsResponse" />
</wsdl:message>
- <wsdl:message name="WS_RetrieveStudySoapIn">
<wsdl:part name="parameters" element="tns:WS_RetrieveStudy" />
</wsdl:message>
- <wsdl:message name="WS_RetrieveStudySoapOut">
<wsdl:part name="parameters"
element="tns:WS_RetrieveStudyResponse" />
</wsdl:message>
- <wsdl:message name="WS_RetrieveSerieSoapIn">
<wsdl:part name="parameters" element="tns:WS_RetrieveSerie" />
</wsdl:message>
- <wsdl:message name="WS_RetrieveSerieSoapOut">
<wsdl:part name="parameters"
element="tns:WS_RetrieveSerieResponse" />
</wsdl:message>
- <wsdl:message name="WS_RetrieveImageSoapIn">
<wsdl:part name="parameters" element="tns:WS_RetrieveImage" />
</wsdl:message>
- <wsdl:message name="WS_RetrieveImageSoapOut">
<wsdl:part name="parameters"
element="tns:WS_RetrieveImageResponse" />
</wsdl:message>
- <wsdl:message name="WS_StoreSerieSoapIn">
<wsdl:part name="parameters" element="tns:WS_StoreSerie" />
</wsdl:message>
- <wsdl:message name="WS_StoreSerieSoapOut">
<wsdl:part name="parameters" element="tns:WS_StoreSerieResponse" />
</wsdl:message>
- <wsdl:message name="WS_GetStudiesCandidatesSoapIn">
<wsdl:part name="parameters"
element="tns:WS_GetStudiesCandidates" />
</wsdl:message>
- <wsdl:message name="WS_GetStudiesCandidatesSoapOut">
```

```
<wsdl:part name="parameters"
element="tns:WS_GetStudiesCandidatesResponse" />
</wsdl:message>
- <wsdl:message name="WS_SVGDEMO2_WS_MatlabSerieJobSoapIn">
<wsdl:part name="parameters"
element="tns:WS_SVGDEMO2_WS_MatlabSerieJob" />
</wsdl:message>
- <wsdl:message name="WS_SVGDEMO2_WS_MatlabSerieJobSoapOut">
<wsdl:part name="parameters"
element="tns:WS_SVGDEMO2_WS_MatlabSerieJobResponse" />
</wsdl:message>
- <wsdl:message name="WS_SVGDEMO2_WS_MatlabImageJobSoapIn">
<wsdl:part name="parameters"
element="tns:WS_SVGDEMO2_WS_MatlabImageJob" />
</wsdl:message>
- <wsdl:message name="WS_SVGDEMO2_WS_MatlabImageJobSoapOut">
<wsdl:part name="parameters"
element="tns:WS_SVGDEMO2_WS_MatlabImageJobResponse" />
</wsdl:message>
- <wsdl:message name="WS_SVGDEMO2_WS_AvailableFilterSoapIn">
<wsdl:part name="parameters"
element="tns:WS_SVGDEMO2_WS_AvailableFilter" />
</wsdl:message>
- <wsdl:message name="WS_SVGDEMO2_WS_AvailableFilterSoapOut">
<wsdl:part name="parameters"
element="tns:WS_SVGDEMO2_WS_AvailableFilterResponse" />
</wsdl:message>
- <wsdl:portType name="WebServicesSoap">
- <wsdl:operation name="HelloWorld">
<wsdl:input message="tns:HelloWorldSoapIn" />
<wsdl:output message="tns:HelloWorldSoapOut" />
</wsdl:operation>
- <wsdl:operation name="WS_GetNrOfPatients">
<wsdl:input message="tns:WS_GetNrOfPatientsSoapIn" />
<wsdl:output message="tns:WS_GetNrOfPatientsSoapOut" />
</wsdl:operation>
- <wsdl:operation name="WS_GetPersonStudies">
<wsdl:input message="tns:WS_GetPersonStudiesSoapIn" />
<wsdl:output message="tns:WS_GetPersonStudiesSoapOut" />
</wsdl:operation>
- <wsdl:operation name="WS_GetStudyDetails">
<wsdl:input message="tns:WS_GetStudyDetailsSoapIn" />
<wsdl:output message="tns:WS_GetStudyDetailsSoapOut" />
</wsdl:operation>
- <wsdl:operation name="WS_GetSerieDetails">
<wsdl:input message="tns:WS_GetSerieDetailsSoapIn" />
<wsdl:output message="tns:WS_GetSerieDetailsSoapOut" />
</wsdl:operation>
- <wsdl:operation name="WS_RetrieveStudy">
<wsdl:input message="tns:WS_RetrieveStudySoapIn" />
<wsdl:output message="tns:WS_RetrieveStudySoapOut" />
</wsdl:operation>
- <wsdl:operation name="WS_RetrieveSerie">
<wsdl:input message="tns:WS_RetrieveSerieSoapIn" />
<wsdl:output message="tns:WS_RetrieveSerieSoapOut" />
</wsdl:operation>
- <wsdl:operation name="WS_RetrieveImage">
```

```

<wsdl:input message="tns:WS_RetrieveImageSoapIn" />
<wsdl:output message="tns:WS_RetrieveImageSoapOut" />
</wsdl:operation>
- <wsdl:operation name="WS_StoreSerie">
<wsdl:input message="tns:WS_StoreSerieSoapIn" />
<wsdl:output message="tns:WS_StoreSerieSoapOut" />
</wsdl:operation>
- <wsdl:operation name="WS_GetStudiesCandidates">
<wsdl:input message="tns:WS_GetStudiesCandidatesSoapIn" />
<wsdl:output message="tns:WS_GetStudiesCandidatesSoapOut" />
</wsdl:operation>
- <wsdl:operation name="WS_SVGDEMO2_WS_MatlabSerieJob">
<wsdl:input
message="tns:WS_SVGDEMO2_WS_MatlabSerieJobSoapIn" />
<wsdl:output
message="tns:WS_SVGDEMO2_WS_MatlabSerieJobSoapOut" />
</wsdl:operation>
- <wsdl:operation name="WS_SVGDEMO2_WS_MatlabImageJob">
<wsdl:input
message="tns:WS_SVGDEMO2_WS_MatlabImageJobSoapIn" />
<wsdl:output
message="tns:WS_SVGDEMO2_WS_MatlabImageJobSoapOut" />
</wsdl:operation>
- <wsdl:operation name="WS_SVGDEMO2_WS_AvailableFilter">
<wsdl:input
message="tns:WS_SVGDEMO2_WS_AvailableFilterSoapIn" />
<wsdl:output
message="tns:WS_SVGDEMO2_WS_AvailableFilterSoapOut" />
</wsdl:operation>
</wsdl:portType>
- <wsdl:binding name="WebServicesSoap" type="tns:WebServicesSoap">
<soap:binding transport="http://schemas.xmlsoap.org/soap/http" />
- <wsdl:operation name="HelloWorld">
<soap:operation soapAction="http://tempuri.org/HelloWorld"
style="document" />
- <wsdl:input>
<soap:body use="literal" />
</wsdl:input>
- <wsdl:output>
<soap:body use="literal" />
</wsdl:output>
</wsdl:operation>
- <wsdl:operation name="WS_GetNrOfPatients">
<soap:operation soapAction="http://tempuri.org/WS_GetNrOfPatients"
style="document" />
- <wsdl:input>
<soap:body use="literal" />
</wsdl:input>
- <wsdl:output>
<soap:body use="literal" />
</wsdl:output>
</wsdl:operation>
- <wsdl:operation name="WS_GetPersonStudies">
<soap:operation
soapAction="http://tempuri.org/WS_GetPersonStudies"
style="document" />
- <wsdl:input>

```



```
<soap:body use="literal" />
</wsdl:input>
- <wsdl:output>
<soap:body use="literal" />
</wsdl:output>
</wsdl:operation>
- <wsdl:operation name="WS_GetStudyDetails">
<soap:operation soapAction="http://tempuri.org/WS_GetStudyDetails"
style="document" />
- <wsdl:input>
<soap:body use="literal" />
</wsdl:input>
- <wsdl:output>
<soap:body use="literal" />
</wsdl:output>
</wsdl:operation>
- <wsdl:operation name="WS_GetSerieDetails">
<soap:operation soapAction="http://tempuri.org/WS_GetSerieDetails"
style="document" />
- <wsdl:input>
<soap:body use="literal" />
</wsdl:input>
- <wsdl:output>
<soap:body use="literal" />
</wsdl:output>
</wsdl:operation>
- <wsdl:operation name="WS_RetrieveStudy">
<soap:operation soapAction="http://tempuri.org/WS_RetrieveStudy"
style="document" />
- <wsdl:input>
<soap:body use="literal" />
</wsdl:input>
- <wsdl:output>
<soap:body use="literal" />
</wsdl:output>
</wsdl:operation>
- <wsdl:operation name="WS_RetrieveSerie">
<soap:operation soapAction="http://tempuri.org/WS_RetrieveSerie"
style="document" />
- <wsdl:input>
<soap:body use="literal" />
</wsdl:input>
- <wsdl:output>
<soap:body use="literal" />
</wsdl:output>
</wsdl:operation>
- <wsdl:operation name="WS_RetrieveImage">
<soap:operation soapAction="http://tempuri.org/WS_RetrieveImage"
style="document" />
- <wsdl:input>
<soap:body use="literal" />
</wsdl:input>
- <wsdl:output>
<soap:body use="literal" />
</wsdl:output>
</wsdl:operation>
- <wsdl:operation name="WS_StoreSerie">
```



```
<soap:operation soapAction="http://tempuri.org/WS_StoreSerie"
style="document" />
- <wsdl:input>
<soap:body use="literal" />
</wsdl:input>
- <wsdl:output>
<soap:body use="literal" />
</wsdl:output>
</wsdl:operation>
- <wsdl:operation name="WS_GetStudiesCandidates">
<soap:operation
soapAction="http://tempuri.org/WS_GetStudiesCandidates"
style="document" />
- <wsdl:input>
<soap:body use="literal" />
</wsdl:input>
- <wsdl:output>
<soap:body use="literal" />
</wsdl:output>
</wsdl:operation>
- <wsdl:operation name="WS_SVGDEMO2_WS_MatlabSerieJob">
<soap:operation
soapAction="http://tempuri.org/WS_SVGDEMO2_WS_MatlabSerieJob"
style="document" />
- <wsdl:input>
<soap:body use="literal" />
</wsdl:input>
- <wsdl:output>
<soap:body use="literal" />
</wsdl:output>
</wsdl:operation>
- <wsdl:operation name="WS_SVGDEMO2_WS_MatlabImageJob">
<soap:operation
soapAction="http://tempuri.org/WS_SVGDEMO2_WS_MatlabImageJob"
style="document" />
- <wsdl:input>
<soap:body use="literal" />
</wsdl:input>
- <wsdl:output>
<soap:body use="literal" />
</wsdl:output>
</wsdl:operation>
- <wsdl:operation name="WS_SVGDEMO2_WS_AvailableFilter">
<soap:operation
soapAction="http://tempuri.org/WS_SVGDEMO2_WS_AvailableFilter"
style="document" />
- <wsdl:input>
<soap:body use="literal" />
</wsdl:input>
- <wsdl:output>
<soap:body use="literal" />
</wsdl:output>
</wsdl:operation>
</wsdl:binding>
- <wsdl:binding name="WebServicesSoap12" type="tns:WebServicesSoap">
<soap12:binding transport="http://schemas.xmlsoap.org/soap/http" />
- <wsdl:operation name="HelloWorld">
```

```
<soap12:operation soapAction="http://tempuri.org/HelloWorld"
style="document" />
- <wsdl:input>
<soap12:body use="literal" />
</wsdl:input>
- <wsdl:output>
<soap12:body use="literal" />
</wsdl:output>
</wsdl:operation>
- <wsdl:operation name="WS_GetNrOfPatients">
<soap12:operation
soapAction="http://tempuri.org/WS_GetNrOfPatients"
style="document" />
- <wsdl:input>
<soap12:body use="literal" />
</wsdl:input>
- <wsdl:output>
<soap12:body use="literal" />
</wsdl:output>
</wsdl:operation>
- <wsdl:operation name="WS_GetPersonStudies">
<soap12:operation
soapAction="http://tempuri.org/WS_GetPersonStudies"
style="document" />
- <wsdl:input>
<soap12:body use="literal" />
</wsdl:input>
- <wsdl:output>
<soap12:body use="literal" />
</wsdl:output>
</wsdl:operation>
- <wsdl:operation name="WS_GetStudyDetails">
<soap12:operation
soapAction="http://tempuri.org/WS_GetStudyDetails"
style="document" />
- <wsdl:input>
<soap12:body use="literal" />
</wsdl:input>
- <wsdl:output>
<soap12:body use="literal" />
</wsdl:output>
</wsdl:operation>
- <wsdl:operation name="WS_GetSerieDetails">
<soap12:operation
soapAction="http://tempuri.org/WS_GetSerieDetails"
style="document" />
- <wsdl:input>
<soap12:body use="literal" />
</wsdl:input>
- <wsdl:output>
<soap12:body use="literal" />
</wsdl:output>
</wsdl:operation>
- <wsdl:operation name="WS_RetrieveStudy">
<soap12:operation soapAction="http://tempuri.org/WS_RetrieveStudy"
style="document" />
- <wsdl:input>
```

```
<soap12:body use="literal" />
</wsdl:input>
- <wsdl:output>
<soap12:body use="literal" />
</wsdl:output>
</wsdl:operation>
- <wsdl:operation name="WS_RetrieveSerie">
<soap12:operation soapAction="http://tempuri.org/WS_RetrieveSerie"
style="document" />
- <wsdl:input>
<soap12:body use="literal" />
</wsdl:input>
- <wsdl:output>
<soap12:body use="literal" />
</wsdl:output>
</wsdl:operation>
- <wsdl:operation name="WS_RetrieveImage">
<soap12:operation
soapAction="http://tempuri.org/WS_RetrieveImage"
style="document" />
- <wsdl:input>
<soap12:body use="literal" />
</wsdl:input>
- <wsdl:output>
<soap12:body use="literal" />
</wsdl:output>
</wsdl:operation>
- <wsdl:operation name="WS_StoreSerie">
<soap12:operation soapAction="http://tempuri.org/WS_StoreSerie"
style="document" />
- <wsdl:input>
<soap12:body use="literal" />
</wsdl:input>
- <wsdl:output>
<soap12:body use="literal" />
</wsdl:output>
</wsdl:operation>
- <wsdl:operation name="WS_GetStudiesCandidates">
<soap12:operation
soapAction="http://tempuri.org/WS_GetStudiesCandidates"
style="document" />
- <wsdl:input>
<soap12:body use="literal" />
</wsdl:input>
- <wsdl:output>
<soap12:body use="literal" />
</wsdl:output>
</wsdl:operation>
- <wsdl:operation name="WS_SVGDEMO2_WS_MatlabSerieJob">
<soap12:operation
soapAction="http://tempuri.org/WS_SVGDEMO2_WS_MatlabSerieJob"
style="document" />
- <wsdl:input>
<soap12:body use="literal" />
</wsdl:input>
- <wsdl:output>
<soap12:body use="literal" />
```

```
</wsdl:output>
</wsdl:operation>
- <wsdl:operation name="WS_SVGDEMO2_WS_MatlabImageJob">
<soap12:operation
soapAction="http://tempuri.org/WS_SVGDEMO2_WS_MatlabImageJob"
style="document" />
- <wsdl:input>
<soap12:body use="literal" />
</wsdl:input>
- <wsdl:output>
<soap12:body use="literal" />
</wsdl:output>
</wsdl:operation>
- <wsdl:operation name="WS_SVGDEMO2_WS_AvailableFilter">
<soap12:operation
soapAction="http://tempuri.org/WS_SVGDEMO2_WS_AvailableFilter"
style="document" />
- <wsdl:input>
<soap12:body use="literal" />
</wsdl:input>
- <wsdl:output>
<soap12:body use="literal" />
</wsdl:output>
</wsdl:operation>
</wsdl:binding>
- <wsdl:service name="WebServices">
- <wsdl:port name="WebServicesSoap" binding="tns:WebServicesSoap">
<soap:address location="http://svgdemo1.
ihelse.net/Webservices.asmx" />
</wsdl:port>
- <wsdl:port name="WebServicesSoap12"
binding="tns:WebServicesSoap12">
<soap12:address location="http://svgdemo1.
ihelse.net/Webservices.asmx" />
</wsdl:port>
</wsdl:service>
</wsdl:definitions>
```

9.2. WSDL for Matlab Webtjenester

```

<?xml version="1.0" encoding="utf-8" ?>
- <wsdl:definitions xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
xmlns:tm="http://microsoft.com/wsdl/mime/textMatching/"
xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/"
xmlns:mime="http://schemas.xmlsoap.org/wsdl/mime/"
xmlns:tns="http://tempuri.org/"
xmlns:s="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://schemas.xmlsoap.org/wsdl/soap12/"
xmlns:http="http://schemas.xmlsoap.org/wsdl/http/"
targetNamespace="http://tempuri.org/"
xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/">
- <wsdl:types>
- <s:schema elementFormDefault="qualified"
targetNamespace="http://tempuri.org/">
- <s:element name="HelloWorld">
<s:complexType />
</s:element>
- <s:element name="HelloWorldResponse">
- <s:complexType>
- <s:sequence>
<s:element minOccurs="0" maxOccurs="1"
name="HelloWorldResult" type="s:string" />
</s:sequence>
</s:complexType>
</s:element>
- <s:element name="WS_MatlabImageJob">
- <s:complexType>
- <s:sequence>
<s:element minOccurs="0" maxOccurs="1"
name="StudyInstanceUID" type="s:string" />
<s:element minOccurs="0" maxOccurs="1"
name="SerieInstanceUID" type="s:string" />
<s:element minOccurs="0" maxOccurs="1" name="ImageUid"
type="s:string" />
<s:element minOccurs="0" maxOccurs="1"
name="FilterName" type="s:string" />
<s:element minOccurs="0" maxOccurs="1" name="FilterLevel"
type="s:string" />
</s:sequence>
</s:complexType>
</s:element>
- <s:element name="WS_MatlabImageJobResponse">
- <s:complexType>
- <s:sequence>
<s:element minOccurs="0" maxOccurs="1"
name="WS_MatlabImageJobResult" type="s:string" />
</s:sequence>
</s:complexType>
</s:element>
- <s:element name="WS_MatlabSerieJob">
- <s:complexType>
- <s:sequence>
<s:element minOccurs="0" maxOccurs="1"

```

```

name="StudyInstanceUID" type="s:string" />
<s:element minOccurs="0" maxOccurs="1"
name="SerieInstanceUID" type="s:string" />
<s:element minOccurs="0" maxOccurs="1"
name="FilterName" type="s:string" />
<s:element minOccurs="0" maxOccurs="1" name="FilterLevel"
type="s:string" />
</s:sequence>
</s:complexType>
</s:element>
- <s:element name="WS_MatlabSerieJobResponse">
- <s:complexType>
- <s:sequence>
<s:element minOccurs="0" maxOccurs="1"
name="WS_MatlabSerieJobResult" type="s:string" />
</s:sequence>
</s:complexType>
</s:element>
- <s:element name="WS_AvailableFilter">
<s:complexType />
</s:element>
- <s:element name="WS_AvailableFilterResponse">
- <s:complexType>
- <s:sequence>
- <s:element minOccurs="0" maxOccurs="1"
name="WS_AvailableFilterResult">
- <s:complexType mixed="true">
- <s:sequence>
<s:any />
</s:sequence>
</s:complexType>
</s:element>
</s:sequence>
</s:complexType>
</s:element>
- <s:element name="WS_FilterMfile">
- <s:complexType>
- <s:sequence>
<s:element minOccurs="0" maxOccurs="1" name="FilterLevel"
type="s:string" />
<s:element minOccurs="0" maxOccurs="1" name="FilterDesc"
type="s:string" />
</s:sequence>
</s:complexType>
</s:element>
- <s:element name="WS_FilterMfileResponse">
- <s:complexType>
- <s:sequence>
<s:element minOccurs="0" maxOccurs="1"
name="WS_FilterMfileResult" type="s:string" />
</s:sequence>
</s:complexType>
</s:element>
- <s:element name="WS_OutStoreSerie">
- <s:complexType>
- <s:sequence>
<s:element minOccurs="0" maxOccurs="1"

```

```
name="StudyInstanceUID" type="s:string" />
<s:element minOccurs="0" maxOccurs="1"
name="SerieInstanceUID" type="s:string" />
</s:sequence>
</s:complexType>
</s:element>
- <s:element name="WS_OutStoreSerieResponse">
- <s:complexType>
- <s:sequence>
<s:element minOccurs="0" maxOccurs="1"
name="WS_OutStoreSerieResult" type="s:string" />
</s:sequence>
</s:complexType>
</s:element>
</s:schema>
</wsdl:types>
- <wsdl:message name="HelloWorldSoapIn">
<wsdl:part name="parameters" element="tns:HelloWorld" />
</wsdl:message>
- <wsdl:message name="HelloWorldSoapOut">
<wsdl:part name="parameters" element="tns:HelloWorldResponse" />
</wsdl:message>
- <wsdl:message name="WS_MatlabImageJobSoapIn">
<wsdl:part name="parameters" element="tns:WS_MatlabImageJob" />
</wsdl:message>
- <wsdl:message name="WS_MatlabImageJobSoapOut">
<wsdl:part name="parameters"
element="tns:WS_MatlabImageJobResponse" />
</wsdl:message>
- <wsdl:message name="WS_MatlabSerieJobSoapIn">
<wsdl:part name="parameters" element="tns:WS_MatlabSerieJob" />
</wsdl:message>
- <wsdl:message name="WS_MatlabSerieJobSoapOut">
<wsdl:part name="parameters"
element="tns:WS_MatlabSerieJobResponse" />
</wsdl:message>
- <wsdl:message name="WS_AvailableFilterSoapIn">
<wsdl:part name="parameters" element="tns:WS_AvailableFilter" />
</wsdl:message>
- <wsdl:message name="WS_AvailableFilterSoapOut">
<wsdl:part name="parameters"
element="tns:WS_AvailableFilterResponse" />
</wsdl:message>
- <wsdl:message name="WS_FilterMfileSoapIn">
<wsdl:part name="parameters" element="tns:WS_FilterMfile" />
</wsdl:message>
- <wsdl:message name="WS_FilterMfileSoapOut">
<wsdl:part name="parameters" element="tns:WS_FilterMfileResponse" />
</wsdl:message>
- <wsdl:message name="WS_OutStoreSerieSoapIn">
<wsdl:part name="parameters" element="tns:WS_OutStoreSerie" />
</wsdl:message>
- <wsdl:message name="WS_OutStoreSerieSoapOut">
<wsdl:part name="parameters"
element="tns:WS_OutStoreSerieResponse" />
</wsdl:message>
- <wsdl:portType name="ServiceSoap">
```

```

- <wsdl:operation name="HelloWorld">
<wsdl:input message="tns:HelloWorldSoapIn" />
<wsdl:output message="tns:HelloWorldSoapOut" />
</wsdl:operation>
- <wsdl:operation name="WS_MatlabImageJob">
<wsdl:input message="tns:WS_MatlabImageJobSoapIn" />
<wsdl:output message="tns:WS_MatlabImageJobSoapOut" />
</wsdl:operation>
- <wsdl:operation name="WS_MatlabSerieJob">
<wsdl:input message="tns:WS_MatlabSerieJobSoapIn" />
<wsdl:output message="tns:WS_MatlabSerieJobSoapOut" />
</wsdl:operation>
- <wsdl:operation name="WS_AvailableFilter">
<wsdl:input message="tns:WS_AvailableFilterSoapIn" />
<wsdl:output message="tns:WS_AvailableFilterSoapOut" />
</wsdl:operation>
- <wsdl:operation name="WS_FilterMfile">
<wsdl:input message="tns:WS_FilterMfileSoapIn" />
<wsdl:output message="tns:WS_FilterMfileSoapOut" />
</wsdl:operation>
- <wsdl:operation name="WS_OutStoreSerie">
<wsdl:input message="tns:WS_OutStoreSerieSoapIn" />
<wsdl:output message="tns:WS_OutStoreSerieSoapOut" />
</wsdl:operation>
</wsdl:portType>
- <wsdl:binding name="ServiceSoap" type="tns:ServiceSoap">
<soap:binding transport="http://schemas.xmlsoap.org/soap/http" />
- <wsdl:operation name="HelloWorld">
<soap:operation soapAction="http://tempuri.org/HelloWorld"
style="document" />
- <wsdl:input>
<soap:body use="literal" />
</wsdl:input>
- <wsdl:output>
<soap:body use="literal" />
</wsdl:output>
</wsdl:operation>
- <wsdl:operation name="WS_MatlabImageJob">
<soap:operation
soapAction="http://tempuri.org/WS_MatlabImageJob"
style="document" />
- <wsdl:input>
<soap:body use="literal" />
</wsdl:input>
- <wsdl:output>
<soap:body use="literal" />
</wsdl:output>
</wsdl:operation>
- <wsdl:operation name="WS_MatlabSerieJob">
<soap:operation soapAction="http://tempuri.org/WS_MatlabSerieJob"
style="document" />
- <wsdl:input>
<soap:body use="literal" />
</wsdl:input>
- <wsdl:output>
<soap:body use="literal" />
</wsdl:output>

```



```
</wsdl:operation>
- <wsdl:operation name="WS_AvailableFilter">
<soap:operation soapAction="http://tempuri.org/WS_AvailableFilter"
style="document" />
- <wsdl:input>
<soap:body use="literal" />
</wsdl:input>
- <wsdl:output>
<soap:body use="literal" />
</wsdl:output>
</wsdl:operation>
- <wsdl:operation name="WS_FilterMfile">
<soap:operation soapAction="http://tempuri.org/WS_FilterMfile"
style="document" />
- <wsdl:input>
<soap:body use="literal" />
</wsdl:input>
- <wsdl:output>
<soap:body use="literal" />
</wsdl:output>
</wsdl:operation>
- <wsdl:operation name="WS_OutStoreSerie">
<soap:operation soapAction="http://tempuri.org/WS_OutStoreSerie"
style="document" />
- <wsdl:input>
<soap:body use="literal" />
</wsdl:input>
- <wsdl:output>
<soap:body use="literal" />
</wsdl:output>
</wsdl:operation>
</wsdl:binding>
- <wsdl:binding name="ServiceSoap12" type="tns:ServiceSoap">
<soap12:binding transport="http://schemas.xmlsoap.org/soap/http" />
- <wsdl:operation name="HelloWorld">
<soap12:operation soapAction="http://tempuri.org/HelloWorld"
style="document" />
- <wsdl:input>
<soap12:body use="literal" />
</wsdl:input>
- <wsdl:output>
<soap12:body use="literal" />
</wsdl:output>
</wsdl:operation>
- <wsdl:operation name="WS_MatlabImageJob">
<soap12:operation
soapAction="http://tempuri.org/WS_MatlabImageJob"
style="document" />
- <wsdl:input>
<soap12:body use="literal" />
</wsdl:input>
- <wsdl:output>
<soap12:body use="literal" />
</wsdl:output>
</wsdl:operation>
- <wsdl:operation name="WS_MatlabSerieJob">
<soap12:operation
```

```
soapAction="http://tempuri.org/WS_MatlabSerieJob"
style="document" />
- <wsdl:input>
<soap12:body use="literal" />
</wsdl:input>
- <wsdl:output>
<soap12:body use="literal" />
</wsdl:output>
</wsdl:operation>
- <wsdl:operation name="WS_AvailableFilter">
<soap12:operation
soapAction="http://tempuri.org/WS_AvailableFilter"
style="document" />
- <wsdl:input>
<soap12:body use="literal" />
</wsdl:input>
- <wsdl:output>
<soap12:body use="literal" />
</wsdl:output>
</wsdl:operation>
- <wsdl:operation name="WS_FilterMfile">
<soap12:operation soapAction="http://tempuri.org/WS_FilterMfile"
style="document" />
- <wsdl:input>
<soap12:body use="literal" />
</wsdl:input>
- <wsdl:output>
<soap12:body use="literal" />
</wsdl:output>
</wsdl:operation>
- <wsdl:operation name="WS_OutStoreSerie">
<soap12:operation soapAction="http://tempuri.org/WS_OutStoreSerie"
style="document" />
- <wsdl:input>
<soap12:body use="literal" />
</wsdl:input>
- <wsdl:output>
<soap12:body use="literal" />
</wsdl:output>
</wsdl:operation>
</wsdl:binding>
- <wsdl:service name="Service">
- <wsdl:port name="ServiceSoap" binding="tns:ServiceSoap">
<soap:address location="http://svgdemo2.
ihelse.net/Matlab/service.asmx" />
</wsdl:port>
- <wsdl:port name="ServiceSoap12" binding="tns:ServiceSoap12">
<soap12:address location="http://svgdemo2.
ihelse.net/Matlab/service.asmx" />
</wsdl:port>
</wsdl:service>
</wsdl:definitions>
```

9.3. Kildekode Application SCP

9.3.1. ApplicationSCP

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Collections;
using System.Windows.Forms;
using System.Drawing;
using ClearCanvas.Dicom;
using log4net;
using log4net.Config;
namespace ini
{
public class ApplicationSCP
{
string localAE;
string localPort;
string localStorageLocation;
string remoteIP;
string remoteAE;
string remotePort;
//protected static readonly
ILog Logger = LogManager.GetLogger(typeof(ApplicationSCP));
ini.iniApi myIni = new ini.iniApi();
public string AppPath()
{
string path;
path = System.IO.Path.GetDirectoryName(
System.Reflection.Assembly.GetExecutingAssembly().GetName().CodeBase);
return path.Substring(6);
}
public Hashtable GetINIValues()
{
string defaultValue = "???";
string iniFile = AppPath() + "\\scp.ini";
//List<String> MyIniFiles = new List<string>();
//Hs MyIniFiles = new HashSet<string>();
Hashtable iniTable = new Hashtable();
/*
* Try to read the content of a simple INI file
*/
List<string> categories = iniApi.GetCategories(iniFile);
foreach (string category in categories)
{
// Console.WriteLine(category);
/*
* Get the keys
*/
List<string> keys = iniApi.GetKeys(iniFile, category);
foreach (string key in keys)
{
/*
* Now output the content
*/
string content = iniApi.GetIniFileString(iniFile, category, key,
defaultValue);
//MyIniFiles.Add(category + "|" + key + "|" + content);
iniTable.Add(key, content);
// Console.WriteLine(string.Concat(" ", key, "\t", content));
}
}
}
}

```

```
return iniTable;
}
public void StartListener()
{
log4net.Config.XmlConfigurator.Configure();
Logger.Info("Application Logger Startet");
StorageScp.StorageLocation = localStorageLocation;
StorageScp.StartListening(localAE, Convert.ToInt16(localPort));
}
public void StartLogging()
{
}
public void PopulateMenu()
{
//ContextMenu = new ContextMenu();
ContextMenuStrip contextMenu1 = new ContextMenuStrip();
NotifyIcon notifyIcon1 = new NotifyIcon();
ToolStripMenuItem lStor = new ToolStripMenuItem("Local Storage Path : " +
localStorageLocation);
ToolStripMenuItem lAE = new ToolStripMenuItem("Local AE Title : " +
localAE);
ToolStripMenuItem lPort = new ToolStripMenuItem("Local Port : "
+ localPort);
ToolStripMenuItem rIP = new ToolStripMenuItem("Remote IP Address : " +
remoteIP);
ToolStripMenuItem rAE = new ToolStripMenuItem("Remote AE Title : " +
remoteAE);
ToolStripMenuItem rPort = new ToolStripMenuItem("Remote Port : " +
remotePort);
contextMenu1.Items.Add(lStor);
contextMenu1.Items.Add(lAE);
contextMenu1.Items.Add(lPort);
contextMenu1.Items.Add(rIP);
contextMenu1.Items.Add(rAE);
contextMenu1.Items.Add(rPort);
contextMenu1.Click += new EventHandler(contextMenu1_Click);
notifyIcon1.Icon = new Icon(AppPath() + "\\SystemTray.ico");
notifyIcon1.Text = "DICOM SCP SETTINGS";
notifyIcon1.ContextMenuStrip = contextMenu1;
notifyIcon1.Visible = true;
}
public void contextMenu1_Click(object Sender, EventArgs e)
{
if (MessageBox.Show("Do you want to stop the DICOM SCP?", "Confirm stop",
MessageBoxButtons.YesNo) == DialogResult.Yes)
{
//DialogResult.Yes was selected
StorageScp.StopListening(Convert.ToInt16(localPort));
Application.Exit();
}
}
public void iniSettings()
{
Hashtable myIniValues = new Hashtable();
myIniValues = GetINIValues();
localAE = myIniValues["LocalAE"].ToString();
localPort = myIniValues["LocalPort"].ToString();
remoteAE = myIniValues["RemoteAE"].ToString();
remoteIP = myIniValues["RemoteIp"].ToString();
remotePort = myIniValues["RemotePort"].ToString();
localStorageLocation = myIniValues["StorageLocation"].ToString();
}
}
}
```

9.3.2. Program

```
using System;
using System.Collections.Generic;
using System.Collections;
using System.Linq;
using System.Windows.Forms;
namespace ini
{
    static class Program
    {
        /// <summary>
        /// The main entry point for the application.
        /// </summary>
        [STAThread]
        static void Main()
        {
            Application.EnableVisualStyles();
            Application.SetCompatibleTextRenderingDefault(false);
            ini.ApplicationSCP MyFunctions = new ini.ApplicationSCP();
            MyFunctions.iniSettings();
            MyFunctions.PopulateMenu();
            MyFunctions.StartListener();
            Application.Run();
        }
    }
}
```

9.3.3. iniAPI

```
using System;
using System.Collections.Generic;
using System.Runtime.InteropServices;
using System.Linq;
using System.Text;
namespace ini
{
    public class iniApi
    {
        // Class uses old method of reading ini files.
        // Used in my program to give the users a common
        // configuration point.
        // Declaring the methods to read using native API
        [DllImport("KERNEL32.DLL", EntryPoint = "GetPrivateProfileStringW",
            SetLastError = true,
            CharSet = CharSet.Unicode, ExactSpelling = true,
            CallingConvention = CallingConvention.StdCall)]
        private static extern int GetPrivateProfileString(
            string lpAppName,
            string lpKeyName,
            string lpDefault,
            string lpReturnString,
            int nSize,
            string lpFilename);
        public static string GetIniFileString(string iniFile, string category, string key,
            string defaultValue)
        {
            //Gets the actual Value from the inifile, based on the Category and Key value
            string returnString = new string(' ', 1024);
            GetPrivateProfileString(category, key, defaultValue, returnString, 1024,
                iniFile);
            return returnString.Split('\0')[0];
        }
        public static List<string> GetKeys(string iniFile, string category)
        {
            // Generates a List collection of all Keys in inifile
            string returnString = new string(' ', 32768);
        }
    }
}
```

```

GetPrivateProfileString(category, null, null, returnString, 32768, iniFile);
List<string> result = new List<string>(returnString.Split('\0'));
result.RemoveRange(result.Count - 2, 2);
return result;
}
// Generates a List collection of all Categories in given iniFile
public static List<string> GetCategories(string iniFile)
{
string returnString = new string(' ', 65536);
GetPrivateProfileString(null, null, null, returnString, 65536, iniFile);
List<string> result = new List<string>(returnString.Split('\0'));
result.RemoveRange(result.Count - 2, 2);
return result;
}
}
}
}

```

9.3.4. SCP (ini-fil)

```

[MY SCP Settings]
StorageLocation=c:\DicomFiles
LocalAE=DEMOAE
LocalPort=104
[Dicom Communication]
Remotep=152.93.134.40
RemoteAE=sir-pacsFIR
RemotePort=104

```

9.3.5. StorageSCP

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.IO;
using System.Net;
using ClearCanvas.Dicom.Network;
using ClearCanvas.Common;
using log4net;
using log4net.Config;
namespace ClearCanvas.Dicom
{
class StorageScp : IDicomServerHandler
{
#region Private Members
private static bool _started = false;
private static String _staticStorageLocation;
private static ServerAssociationParameters _staticAssocParameters;
private ServerAssociationParameters _assocParameters;
#endregion
#region Constructors
protected static readonly ILog Logger = LogManager.GetLogger(typeof(StorageScp));
//public ILog Logger;
public void InitiateLogging()
{
log4net.Config.XmlConfigurator.Configure();
}
private StorageScp(ServerAssociationParameters assoc)
{
_assocParameters = assoc;
}
#endregion
#region Public Properties
public static bool Started
{
get { return _started; }
}
}
}

```

```

}
public static String StorageLocation
{
get { return _staticStorageLocation; }
set { _staticStorageLocation = value; }
}
#endregion
#region Private Methods
private static void AddPresentationContexts(ServerAssociationParameters assoc)
{
byte pcid = assoc.AddPresentationContext(SopClass.VerificationSopClass);
assoc.AddTransferSyntax(pcid, TransferSyntax.ExplicitVrLittleEndian);
assoc.AddTransferSyntax(pcid, TransferSyntax.ImplicitVrLittleEndian);
pcid = assoc.AddPresentationContext(SopClass.MrImageStorage);
assoc.AddTransferSyntax(pcid, TransferSyntax.ExplicitVrLittleEndian);
assoc.AddTransferSyntax(pcid, TransferSyntax.ImplicitVrLittleEndian);
pcid = assoc.AddPresentationContext(SopClass.CtImageStorage);
assoc.AddTransferSyntax(pcid, TransferSyntax.ExplicitVrLittleEndian);
assoc.AddTransferSyntax(pcid, TransferSyntax.ImplicitVrLittleEndian);
pcid = assoc.AddPresentationContext(SopClass.SecondaryCaptureImageStorage);
assoc.AddTransferSyntax(pcid, TransferSyntax.ExplicitVrLittleEndian);
assoc.AddTransferSyntax(pcid, TransferSyntax.ImplicitVrLittleEndian);
pcid = assoc.AddPresentationContext(SopClass.UltrasoundImageStorage);
assoc.AddTransferSyntax(pcid, TransferSyntax.ExplicitVrLittleEndian);
assoc.AddTransferSyntax(pcid, TransferSyntax.ImplicitVrLittleEndian);
pcid = assoc.AddPresentationContext(SopClass.UltrasoundImageStorageRetired);
assoc.AddTransferSyntax(pcid, TransferSyntax.ExplicitVrLittleEndian);
assoc.AddTransferSyntax(pcid, TransferSyntax.ImplicitVrLittleEndian);
pcid = assoc.AddPresentationContext(SopClass.UltrasoundMultiFrameImageStorage);
;
assoc.AddTransferSyntax(pcid, TransferSyntax.ExplicitVrLittleEndian);
assoc.AddTransferSyntax(pcid, TransferSyntax.ImplicitVrLittleEndian);
pcid = assoc.AddPresentationContext(SopClass.
UltrasoundMultiFrameImageStorageRetired);
assoc.AddTransferSyntax(pcid, TransferSyntax.ExplicitVrLittleEndian);
assoc.AddTransferSyntax(pcid, TransferSyntax.ImplicitVrLittleEndian);
pcid = assoc.AddPresentationContext(SopClass.NuclearMedicineImageStorage);
assoc.AddTransferSyntax(pcid, TransferSyntax.ExplicitVrLittleEndian);
assoc.AddTransferSyntax(pcid, TransferSyntax.ImplicitVrLittleEndian);
pcid = assoc.AddPresentationContext(SopClass.
DigitalIntraOralXRayImageStorageForPresentation);
assoc.AddTransferSyntax(pcid, TransferSyntax.ExplicitVrLittleEndian);
assoc.AddTransferSyntax(pcid, TransferSyntax.ImplicitVrLittleEndian);
pcid = assoc.AddPresentationContext(SopClass.
DigitalIntraOralXRayImageStorageForProcessing);
assoc.AddTransferSyntax(pcid, TransferSyntax.ExplicitVrLittleEndian);
assoc.AddTransferSyntax(pcid, TransferSyntax.ImplicitVrLittleEndian);
pcid = assoc.AddPresentationContext(SopClass.
DigitalMammographyXRayImageStorageForPresentation);
assoc.AddTransferSyntax(pcid, TransferSyntax.ExplicitVrLittleEndian);
assoc.AddTransferSyntax(pcid, TransferSyntax.ImplicitVrLittleEndian);
pcid = assoc.AddPresentationContext(SopClass.
DigitalMammographyXRayImageStorageForProcessing);
assoc.AddTransferSyntax(pcid, TransferSyntax.ExplicitVrLittleEndian);
assoc.AddTransferSyntax(pcid, TransferSyntax.ImplicitVrLittleEndian);
pcid = assoc.AddPresentationContext(SopClass.
DigitalXRayImageStorageForPresentation);
assoc.AddTransferSyntax(pcid, TransferSyntax.ExplicitVrLittleEndian);
assoc.AddTransferSyntax(pcid, TransferSyntax.ImplicitVrLittleEndian);
pcid = assoc.AddPresentationContext(SopClass.
DigitalXRayImageStorageForProcessing);
assoc.AddTransferSyntax(pcid, TransferSyntax.ExplicitVrLittleEndian);
assoc.AddTransferSyntax(pcid, TransferSyntax.ImplicitVrLittleEndian);
pcid = assoc.AddPresentationContext(SopClass.ComputedRadiographyImageStorage);
assoc.AddTransferSyntax(pcid, TransferSyntax.ExplicitVrLittleEndian);
assoc.AddTransferSyntax(pcid, TransferSyntax.ImplicitVrLittleEndian);
pcid = assoc.AddPresentationContext(SopClass.

```



```
GrayscaleSoftcopyPresentationStateStorageSopClass);
assoc.AddTransferSyntax(pcid, TransferSyntax.ExplicitVrLittleEndian);
assoc.AddTransferSyntax(pcid, TransferSyntax.ImplicitVrLittleEndian);
pcid = assoc.AddPresentationContext(SopClass.
KeyObjectSelectionDocumentStorage);
assoc.AddTransferSyntax(pcid, TransferSyntax.ExplicitVrLittleEndian);
assoc.AddTransferSyntax(pcid, TransferSyntax.ImplicitVrLittleEndian);
pcid = assoc.AddPresentationContext(SopClass.
OphthalmicPhotography16BitImageStorage);
assoc.AddTransferSyntax(pcid, TransferSyntax.ExplicitVrLittleEndian);
assoc.AddTransferSyntax(pcid, TransferSyntax.ImplicitVrLittleEndian);
pcid = assoc.AddPresentationContext(SopClass.
OphthalmicPhotography8BitImageStorage);
assoc.AddTransferSyntax(pcid, TransferSyntax.ExplicitVrLittleEndian);
assoc.AddTransferSyntax(pcid, TransferSyntax.ImplicitVrLittleEndian);
pcid = assoc.AddPresentationContext(SopClass.VideoEndoscopicImageStorage);
assoc.AddTransferSyntax(pcid, TransferSyntax.ExplicitVrLittleEndian);
assoc.AddTransferSyntax(pcid, TransferSyntax.ImplicitVrLittleEndian);
pcid = assoc.AddPresentationContext(SopClass.VideoMicroscopicImageStorage);
assoc.AddTransferSyntax(pcid, TransferSyntax.ExplicitVrLittleEndian);
assoc.AddTransferSyntax(pcid, TransferSyntax.ImplicitVrLittleEndian);
pcid = assoc.AddPresentationContext(SopClass.VideoPhotographicImageStorage);
assoc.AddTransferSyntax(pcid, TransferSyntax.ExplicitVrLittleEndian);
assoc.AddTransferSyntax(pcid, TransferSyntax.ImplicitVrLittleEndian);
pcid = assoc.AddPresentationContext(SopClass.VIEndoscopicImageStorage);
assoc.AddTransferSyntax(pcid, TransferSyntax.ExplicitVrLittleEndian);
assoc.AddTransferSyntax(pcid, TransferSyntax.ImplicitVrLittleEndian);
pcid = assoc.AddPresentationContext(SopClass.VIMicroscopicImageStorage);
assoc.AddTransferSyntax(pcid, TransferSyntax.ExplicitVrLittleEndian);
assoc.AddTransferSyntax(pcid, TransferSyntax.ImplicitVrLittleEndian);
pcid = assoc.AddPresentationContext(SopClass.VIPhotographicImageStorage);
assoc.AddTransferSyntax(pcid, TransferSyntax.ExplicitVrLittleEndian);
assoc.AddTransferSyntax(pcid, TransferSyntax.ImplicitVrLittleEndian);
pcid = assoc.AddPresentationContext(SopClass.
VISlideCoordinatesMicroscopicImageStorage);
assoc.AddTransferSyntax(pcid, TransferSyntax.ExplicitVrLittleEndian);
assoc.AddTransferSyntax(pcid, TransferSyntax.ImplicitVrLittleEndian);
pcid = assoc.AddPresentationContext(SopClass.
XRayAngiographicBiPlaneImageStorageRetired);
assoc.AddTransferSyntax(pcid, TransferSyntax.ExplicitVrLittleEndian);
assoc.AddTransferSyntax(pcid, TransferSyntax.ImplicitVrLittleEndian);
pcid = assoc.AddPresentationContext(SopClass.XRayAngiographicImageStorage);
assoc.AddTransferSyntax(pcid, TransferSyntax.ExplicitVrLittleEndian);
assoc.AddTransferSyntax(pcid, TransferSyntax.ImplicitVrLittleEndian);
pcid = assoc.AddPresentationContext(SopClass.
XRayRadiofluoroscopicImageStorage);
assoc.AddTransferSyntax(pcid, TransferSyntax.ExplicitVrLittleEndian);
assoc.AddTransferSyntax(pcid, TransferSyntax.ImplicitVrLittleEndian);
pcid = assoc.AddPresentationContext(SopClass.XRayRadiationDoseSrStorage);
assoc.AddTransferSyntax(pcid, TransferSyntax.ExplicitVrLittleEndian);
assoc.AddTransferSyntax(pcid, TransferSyntax.ImplicitVrLittleEndian);
pcid = assoc.AddPresentationContext(SopClass.ChestCadSrStorage);
assoc.AddTransferSyntax(pcid, TransferSyntax.ExplicitVrLittleEndian);
assoc.AddTransferSyntax(pcid, TransferSyntax.ImplicitVrLittleEndian);
pcid = assoc.AddPresentationContext(SopClass.XRay3dAngiographicImageStorage);
assoc.AddTransferSyntax(pcid, TransferSyntax.ExplicitVrLittleEndian);
assoc.AddTransferSyntax(pcid, TransferSyntax.ImplicitVrLittleEndian);
pcid = assoc.AddPresentationContext(SopClass.XRay3dCraniofacialImageStorage);
assoc.AddTransferSyntax(pcid, TransferSyntax.ExplicitVrLittleEndian);
assoc.AddTransferSyntax(pcid, TransferSyntax.ImplicitVrLittleEndian);
pcid = assoc.AddPresentationContext(SopClass.EncapsulatedCdaStorage);
assoc.AddTransferSyntax(pcid, TransferSyntax.ExplicitVrLittleEndian);
assoc.AddTransferSyntax(pcid, TransferSyntax.ImplicitVrLittleEndian);
pcid = assoc.AddPresentationContext(SopClass.OphthalmicTomographyImageStorage);
;
assoc.AddTransferSyntax(pcid, TransferSyntax.ExplicitVrLittleEndian);
assoc.AddTransferSyntax(pcid, TransferSyntax.ImplicitVrLittleEndian);
```



```

}
#endregion
#region Static Public Methods
public static void StartListening(string aeTitle, int port)
{
if (!_started)
return;
_staticAssocParameters = new ServerAssociationParameters(aeTitle, new
IPEndPoint(IPAddress.Any, port));
AddPresentationContexts(_staticAssocParameters);
DicomServer.StartListening(_staticAssocParameters,
delegate(DicomServer server, ServerAssociationParameters assoc)
{
return new StorageScp(assoc);
});
_started = true;
}
public static void StopListening(int port)
{
if (!_started)
{
DicomServer.StopListening(_staticAssocParameters);
_started = false;
}
}
#endregion
#region IDicomServerHandler Members
void IDicomServerHandler.OnReceiveAssociateRequest(DicomServer server,
ServerAssociationParameters association)
{
server.SendAssociateAccept(association);
}
void IDicomServerHandler.OnReceiveRequestMessage(DicomServer server,
ServerAssociationParameters association, byte presentationID, DicomMessage message)
{
if (message.CommandField == DicomCommandField.CEchoRequest)
{
server.SendCEchoResponse(presentationID, message.MessageId, DicomStatuses.
Success);
return;
}
String studyInstanceUid = null;
String seriesInstanceUid = null;
DicomUid sopInstanceUid;
bool ok = message.DataSet[DicomTags.SopInstanceUid].TryGetUid(0, out
sopInstanceUid);
if (ok) ok = message.DataSet[DicomTags.SeriesInstanceUid].TryGetString(0, out
seriesInstanceUid);
if (ok) ok = message.DataSet[DicomTags.StudyInstanceUid].TryGetString(0, out
studyInstanceUid);
if (!ok)
{
Logger.Error("Unable to retrieve UIDs from request message, sending
failure status.");
server.SendCStoreResponse(presentationID, message.MessageId,
sopInstanceUid.UID,
DicomStatuses.ProcessingFailure);
return;
}
if (!Directory.Exists(StorageScp.StorageLocation))
Directory.CreateDirectory(StorageScp.StorageLocation);
StringBuilder path = new StringBuilder();
path.AppendFormat("{0}{1}{2}{3}{4}", StorageScp.StorageLocation, Path.
DirectorySeparatorChar,
studyInstanceUid, Path.DirectorySeparatorChar, seriesInstanceUid);
Directory.CreateDirectory(path.ToString());
path.AppendFormat("{0}{1}.dcm", Path.DirectorySeparatorChar, sopInstanceUid.

```

```

UID);
DicomFile dicomFile = new DicomFile(message, path.ToString());
dicomFile.TransferSyntaxUid = TransferSyntax.ExplicitVrLittleEndianUid;
dicomFile.MediaStorageSopInstanceUid = sopInstanceUid.UID;
dicomFile.ImplementationClassUid = DicomImplementation.ClassUID.UID;
dicomFile.ImplementationVersionName = DicomImplementation.Version;
dicomFile.SourceApplicationEntityTitle = association.CallingAE;
dicomFile.MediaStorageSopClassUid = message.SopClass.Uid;
dicomFile.Save(DicomWriteOptions.None);
String patientName = dicomFile.DataSet[DicomTags.PatientsName].GetString(0,
"");
Logger.Info(string.Concat("Received SOP Instance: ",sopInstanceUid, " for
patient ", patientName));
server.SendCStoreResponse(presentationID, message.MessageId,
sopInstanceUid.UID,
DicomStatuses.Success);
}
void IDicomServerHandler.OnReceiveResponseMessage(DicomServer server,
ServerAssociationParameters association, byte presentationID, DicomMessage message)
{
Logger.Error("Unexpectedly received response mess on server.");
server.SendAssociateAbort(DicomAbortSource.ServiceUser, DicomAbortReason.
UnrecognizedPDU);
}
void IDicomServerHandler.OnReceiveReleaseRequest(DicomServer server,
ServerAssociationParameters association)
{
Logger.Info(string.Concat("Received association release request from ",
association.CallingAE));
}
void IDicomServerHandler.OnReceiveAbort(DicomServer server,
ServerAssociationParameters association, DicomAbortSource source, DicomAbortReason
reason)
{
Logger.Error("Unexpected association abort received.");
}
void IDicomServerHandler.OnNetworkError(DicomServer server,
ServerAssociationParameters association, Exception e)
{
Logger.Error(string.Concat("Unexpected network error over association from
",association.CallingAE),e);
}
void IDicomServerHandler.OnDimseTimeout(DicomServer server,
ServerAssociationParameters association)
{
Logger.Info("Received DIMSE Timeout, continuing listening for messages");
}
protected void LogAssociationStatistics(ServerAssociationParameters association)
{
}
}
#endregion
}
}

```

9.4. Kildekode DICOM WS

9.4.1. DICOMCONNECT

```

using System;
using System.Collections.Generic;
using System.Collections;
using System.Linq;
using System.Web;
using System.Runtime.Serialization.Formatters;
using System.Xml.Serialization;

```

```

using System.Xml;
using System.Text;
using log4net;
using ClearCanvas.Dicom;
using ClearCanvas.Common;
using ClearCanvas.Dicom.Network.Scu;
using ClearCanvas.Dicom.Iod.Iods;
public class DicomConnect
{
    // Developed by : BS
    // Date: April 09
    // Version 1
    // Description:
    // This class has all methods used to communicate with a DICOM Archive
    // The package from ClearCanvas is used to instantiate a DICOM object
    //
    // Setting global variables to the system, read from Web Config file
    string LocalAE = ReadWebConfigSetting("LocalAETitle", "");
    string LocalStudyRoot = ReadWebConfigSetting("LocalStudyRoot", "");
    string RemoteAETitle = ReadWebConfigSetting("RemoteAETitle", "");
    string RemoteIPAdress = ReadWebConfigSetting("RemotelPAdress", "");
    string RemotePort = ReadWebConfigSetting("RemotePort", "");
    public DicomConnect()
    {
        //
        // TODO: Add constructor logic here
        //
    }
    public static String ReadWebConfigSetting(String key, String defaultValue)
    {
        // Method to read from WebConfig file.
        // If key is not found in the webconfig the default value will be returned
        try
        {
            object setting = System.Configuration.ConfigurationSettings.AppSettings[key];
            if (setting != null)
            {
                if (setting.ToString() == "")
                {
                    setting = null;
                }
            }
            return (setting == null) ? defaultValue : (String)setting;
        }
        catch
        {
            return defaultValue;
        }
    }
    public XmlDocument StudyRoot(string PID)
    {
        // Method generates a XmlDocument containing given properties
        // Instanciates a ClearCanvas StudyRootFind object and
        // a ClearCanvas StudyQueryIod to be used in the communication
        StudyRootFindScu findstudy = new StudyRootFindScu();
        StudyQueryIod studyMessage = new StudyQueryIod();
        //Populating studymessage with "default" values needed
        studyMessage.SetCommonTags();
        // Setting the PatientID property to our PID
        studyMessage.PatientId = PID;
        // The StudyRootFind populates a IList object with our resultset
        // based on the properties set
        IList<StudyQueryIod> studyList = findstudy.Find(LocalAE, RemoteAETitle ,
        RemoteIPAdress, Convert.ToInt16(RemotePort), studyMessage);
        // Building my own XmlDocument for the WS
        XmlSerializer ser = new XmlSerializer(typeof(ArrayList));
        StringBuilder sb = new StringBuilder();
    }
}

```

```

XmlWriter writer = XmlWriter.Create(sb, null);
writer.WriteStartDocument();
writer.WriteStartElement("Result");
writer.WriteAttributeString("PID", PID);
foreach (StudyQuerylod st in studyList)
{
writer.WriteStartElement("Studydate", Convert.ToString(st.StudyDate));
writer.WriteStartElement("Modality", st.ModalitiesInStudy);
writer.WriteEndElement();
writer.WriteStartElement("Description", st.StudyDescription);
writer.WriteEndElement();
writer.WriteStartElement("Accessionnumber", st.AccessionNumber);
writer.WriteEndElement();
writer.WriteStartElement("StudyInstanceUid", st.StudyInstanceUid);
writer.WriteEndElement();
writer.WriteStartElement("NumberOfRelatedInstances", Convert.ToString(st.
NumberOfStudyRelatedInstances));
writer.WriteEndElement();
writer.WriteEndElement();
}
writer.WriteEndElement();
writer.WriteEndDocument();
writer.Flush();
XmlDocument xmlDocument = new XmlDocument();
xmlDocument.LoadXml(sb.ToString());
return xmlDocument;
}
public XmlDocument StudyDetails(string StudyInstanceUID)
{
// Method generates a XmlDocument containing given properties
// Instanciates a ClearCanvas StudyRootFind object and
// a ClearCanvas StudyQuerylod to be used in the communication
StudyRootFindScu findstudy = new StudyRootFindScu();
SeriesQuerylod studyMessage = new SeriesQuerylod();
studyMessage.SetCommonTags();
// Setting my property to identify a given Study
studyMessage.StudyInstanceUid = StudyInstanceUID;
IList<SeriesQuerylod> studyList = findstudy.Find(LocalAE , RemoteAETitle ,
RemoteIPAdress , Convert.ToInt16(RemotePort), studyMessage);
//Building a XmlDocument by parsing the result from the IList
XmlSerializer ser = new XmlSerializer(typeof(ArrayList));
StringBuilder sb = new StringBuilder();
XmlWriter writer = XmlWriter.Create(sb, null);
writer.WriteStartDocument();
writer.WriteStartElement("StudyInstanceUid", StudyInstanceUID);
//Iterating through the ClearCanvas Resultset
foreach (SeriesQuerylod sr in studyList)
{
writer.WriteStartElement("SerieInstanceUid", sr.SeriesInstanceUid);
writer.WriteStartElement("SerieDescription", sr.SeriesDescription);
writer.WriteEndElement();
writer.WriteStartElement("Serienr", sr.SeriesNumber);
writer.WriteEndElement();
writer.WriteStartElement("ImagesInSerie", Convert.ToString(sr.
NumberOfSeriesRelatedInstances));
writer.WriteEndElement();
writer.WriteEndElement();
}
writer.WriteEndElement();
writer.WriteEndDocument();
writer.Flush();
XmlDocument xmlDocument = new XmlDocument();
xmlDocument.LoadXml(sb.ToString());
return xmlDocument;
}
public XmlDocument SerieDetails(string StudyInstanceUID,string SerieInstanceUID)
{

```

```

// Method generates a XmlDocument containing given properties
// Instanciates a ClearCanvas StudyRootFind object and
// a ClearCanvas StudyQuerylod to be used in the communication
StudyRootFindScu findserie = new StudyRootFindScu();
ImageQuerylod serieMessage = new ImageQuerylod();
serieMessage.SetCommonTags();
// Setting my given properties, StudyInstanceUID and SerieInstanceUID
serieMessage.StudyInstanceUid = StudyInstanceUID;
serieMessage.SeriesInstanceUid = SerieInstanceUID;
IList<ImageQuerylod> serieList = findserie.Find(LocalAE, RemoteAETitle,
RemoteIPAdress, Convert.ToInt16(RemotePort), serieMessage);
//Building my XmlDocument
XmlSerializer ser = new XmlSerializer(typeof(ArrayList));
StringBuilder sb = new StringBuilder();
XmlWriter writer = XmlWriter.Create(sb, null);
writer.WriteStartDocument();
writer.WriteStartElement("Result");
writer.WriteStartElement("StudyInstanceUID", StudyInstanceUID);
writer.WriteStartElement("SerieInstanceUID", SerieInstanceUID);
writer.WriteStartElement("ImageInstanceUID");
//Iterating through the ClearCanvas Resultset
foreach (ImageQuerylod se in serieList)
{
writer.WriteStartElement("SOPInstanceUid", se.SopInstanceUid);
writer.WriteStartElement("Image", se.InstanceNumber);
writer.WriteEndElement();
writer.WriteStartElement("NumberOfFrames",se.NumberOfFrames);
writer.WriteEndElement();
writer.WriteStartElement("SopClassUID", se.SopClassUid);
writer.WriteEndElement();
writer.WriteEndElement();
}
writer.WriteEndElement();
writer.WriteEndElement();
writer.WriteEndElement();
writer.WriteEndElement();
writer.WriteEndElement();
writer.Flush();
XmlDocument xmlDoc = new XmlDocument();
xmlDoc.LoadXml(sb.ToString());
return xmlDoc;
}
public string GetNrofPatients(string PID)
{
// Method generates a string containing objects found information
// Instanciates a ClearCanvas StudyRootFind object and
// a ClearCanvas StudyQuerylod to be used in the communication
PatientRootFindScu findScu = new PatientRootFindScu();
PatientQuerylod queryMessage = new PatientQuerylod();
queryMessage.SetCommonTags();
//Setting the PID Property
queryMessage.PatientId = PID;
IList<PatientQuerylod> patientList = findScu.Find(LocalAE , RemoteAETitle ,
RemoteIPAdress , Convert.ToInt16(RemotePort), queryMessage);
// Checking the resultset to return appropriate nrofpaticnts
if (patientList.Count < 1)
{
return ("No patient found");
}
if (patientList.Count > 1)
{
return string.Concat(patientList.Count.ToString(), " patients found");
}
else
{
return ("Found 1 patient");
}
}

```

```

}
public string RetrieveStudy(string StudyInstanceUID)
{
// Method generates a string identifying if a studymove was successfull
// Instanciates a ClearCanvas MoveScuBase object to be used
// in the communication
MoveScuBase moveScu = new StudyRootMoveScu(LocalAE ,RemoteAETitle , RemoteIPAdress
, Convert.ToInt16(RemotePort), LocalAE);
moveScu.AddStudyInstanceUid(StudyInstanceUID);
moveScu.Move();
return string.Concat("Completed moving StudyInstanceUID ", StudyInstanceUID);
}
public string RetrieveSerie(string SerieInstanceUID)
{
// Method generates a string identifying if a seriemove was successfull
// Instanciates a ClearCanvas MoveScuBase object to be used
// in the communication
MoveScuBase moveScu = new StudyRootMoveScu(LocalAE , RemoteAETitle ,
RemoteIPAdress , Convert.ToInt16(RemotePort), LocalAE);
moveScu.AddSeriesInstanceUid(SerieInstanceUID);
moveScu.Move();
return string.Concat("Completed moving SerieInstanceUID ",SerieInstanceUID);
}
public string RetrieveImage(string ImageUid)
{
// Method generates a string identifying if a imagemove was successfull
// Instanciates a ClearCanvas MoveScuBase object to be used
// in the communication
MoveScuBase moveScu = new StudyRootMoveScu(LocalAE , RemoteAETitle ,
RemoteIPAdress , Convert.ToInt16(RemotePort), LocalAE);
moveScu.AddSopInstanceUid(ImageUid);
moveScu.Move();
return string.Concat("Completed moving ImageUID ", ImageUid );
}
public XmlDocument StudyRootCandidate(string studyRange, string modality, string
StudyDescription, string Sex)
{
StudyRootFindScu findstudy = new StudyRootFindScu();
StudyQueryIod studyMessage = new StudyQueryIod();
studyMessage.SetCommonTags();
studyMessage.ModalitiesInStudy = modality;
//StudyRange format = YYYYMMDD-YYYYMMDD
studyMessage.DicomAttributeCollection[DicomTags.StudyDate].SetStringValue
(studyRange);
//WILDCARDS ALLOWED
studyMessage.StudyDescription = StudyDescription;
//studyMessage.PatientId = PatientID;
studyMessage.PatientsSex = Sex;
IList<StudyQueryIod> studyList = findstudy.Find(LocalAE, RemoteAETitle,
RemoteIPAdress, Convert.ToInt16(RemotePort), studyMessage);
XmlSerializer ser = new XmlSerializer(typeof(ArrayList));
StringBuilder sb = new StringBuilder();
XmlWriter writer = XmlWriter.Create(sb, null);
writer.WriteStartDocument();
writer.WriteStartElement("Result");
foreach (StudyQueryIod st in studyList)
{
writer.WriteStartElement("Studydate", Convert.ToString(st.StudyDate));
// writer.WriteEndElement();
// writer.WriteStartElement("Study");
writer.WriteStartElement("Modality", st.ModalitiesInStudy);
writer.WriteEndElement();
writer.WriteStartElement("Description", st.StudyDescription);
writer.WriteEndElement();
writer.WriteStartElement("Accessionnumber", st.AccessionNumber);
writer.WriteEndElement();
writer.WriteStartElement("StudyInstanceUid", st.StudyInstanceUid);

```

```
writer.WriteEndElement();
writer.WriteStartElement("NumberOfRelatedInstances", Convert.ToString(st.
NumberOfStudyRelatedInstances));
writer.WriteEndElement();
writer.WriteEndElement();
}
writer.WriteEndElement();
writer.WriteEndDocument();
writer.Flush();
XmlDocument xmlDoc = new XmlDocument();
xmlDoc.LoadXml(sb.ToString());
return xmlDoc;
}
public string StoreSerie(string StudyInstanceUid, string SerieInstanceUID)
{
    // Method generates a string identifying if a store was successful
    // The method will iterate through all files in Root/PatientUID/SerieUid Folder,
    // adding the dcm files to the StoreBaseObject. When all images have been
    // added the method will issue a C-Store to the DICOM Archive
    string FullFolder = string.Concat(LocalStudyRoot, "\\", StudyInstanceUid, "\\",
SerieInstanceUID);
    string[] files = null;
    files = System.IO.Directory.GetFiles(FullFolder);
    // instancing a ClearCanvas StorageScu object
    StorageScu myStore = new StorageScu(LocalAE, RemoteAETitle, RemoteIPAdress,
Convert.ToInt16(RemotePort));
    //Iterate through all files, adding the dcm files
    foreach (string file in files)
    {
        if (file.EndsWith(".dcm"))
        {
            myStore.AddFile(file);
        }
    }
    //Issue a CStore request to the DICOM Archive
    myStore.Send();
    // GarbageCollect
    myStore.Dispose();
    return "Done";
}
}
```


9.4.2. WEBSERVICES

```

using System;
using System.Collections.Generic;
using System.Collections;
using System.Linq;
using System.Web;
using System.Web.Services;
using System.Xml.Serialization;
using System.Xml;
using System.Text;
/// <summary>
/// Summary description for WebServices
/// </summary>
[WebService(Namespace = "http://tempuri.org/")]
[WebServiceBinding(ConformsTo = WsiProfiles.BasicProfile1_1)]
// To allow this Web Service to be called from script, using ASP.NET AJAX, uncomment the
// following line.
// [System.Web.Script.Services.ScriptService]
public class WebServices : System.Web.Services.WebService {
// Developed by : BS
// Date: April 09
// Version 1
// Description:
// This class contains the webservices needed to communicate with a
// Dicom Node, having both a SCP and a SCU service
// Configuration of the node is set in the web.config file, under the application
// settings
// The Keys available are :
// key="LocalStudyRoot" Desc: The filearea to put the DICOM image files
// key="LocalAETitle" Desc: This local AETitle
// key="RemoteAETitle" Desc: The Remote AETitle
// key="RemoteIPAdress" Desc: The remote node IP adress
// key="RemotePort" Desc: The remote node listening port
// Instanciating a object from my local class
DicomConnect MyDicomConnection = new DicomConnect();
public WebServices () {
//Uncomment the following line if using designed components
//InitializeComponent();
}
[WebMethod]
public string HelloWorld() {
return "Hello World";
}
[WebMethod]
public string WS_GetNrOfPatients(string PID)
{
// WS Returning a string containing how many
// matches in the Dicom Archive for the given
// Patient Identifier
return MyDicomConnection.GetNrofPatients(PID);
}
[WebMethod]
public XmlDocument WS_GetPersonStudies(string PID)
{
// WS returning XmlDocument containing found persons
// and their related studies
XmlDocument IResult=new XmlDocument();
IResult = MyDicomConnection.StudyRoot(PID);
return IResult;
}
[WebMethod]
public XmlDocument WS_GetStudyDetails(string StudyInstanceUID)
{
//WS returning XmlDocument containing all series from a given

```



```

//StudyInstanceUID
XmlDocument IResult = new XmlDocument();
IResult = MyDicomConnection.StudyDetails(StudyInstanceUID);
return IResult;
}
[WebMethod]
public XmlDocument WS_GetSerieDetails(String StudyInstanceUid,string SerieInstanceUID)
{
// WS returning XmlDocument containing all images in a given
// StudyInstanceUid and SerieInstanceUId
XmlDocument IResult = new XmlDocument();
IResult = MyDicomConnection.SerieDetails(StudyInstanceUid, SerieInstanceUID);
return IResult;
}
[WebMethod]
public string WS_RetrieveStudy(string StudyInstanceUID)
{
// WS issues a move of a given study to the local cache
// Result of the move is given to the initiator
return MyDicomConnection.RetrieveStudy(StudyInstanceUID);
}
[WebMethod]
public string WS_RetrieveSerie(string SerieInstanceUID)
{
// WS issues a move of a given serie to the local cache
// Result of the move is given to the initiator
// NOTE: Not all DICOM Archives support seriesLevel move
// To bypass this issue one could issue a move on all images
// in wanted serie
return MyDicomConnection.RetrieveSerie(SerieInstanceUID);
}
[WebMethod]
public string WS_RetrieveImage(string ImageUid)
{
// WS issues a move of a given image identified by the ImageUid to the local cache
// Result of the move is given to the initiator
return MyDicomConnection.RetrieveImage(ImageUid);
}
[WebMethod]
public string WS_StoreSerie(string StudyInstanceUID, string SerieInstanceUID)
{
// WS stores all images in serie identified by the StudyInstanceUID and the
// SerieInstanceUID. The images must be found in correct folder seen from the
// RootFolder set in the initialisation of the entire class
return MyDicomConnection.StoreSerie(StudyInstanceUID, SerieInstanceUID);
}
[WebMethod]
public XmlDocument WS_GetStudiesCandidates(string studyRange, string Modality, string
StudyDescription, string Sex)
{
// WS finds the studies in the archive having properties set by :
// studyRange : Date range format yyymmdd-yyymmdd
// Modality : DICOM Modality type (MR,CT,CR etc)
// StudyDescription : The description from the Dicom Tags
// Sex : M=Male, F=Female
return MyDicomConnection.StudyRootCandidate(studyRange, Modality,
StudyDescription, Sex);
}
//[WebMethod]
//public string WS_RetrieveImageFilter(string StudyInstanceUID,string SerieInstanceUID
,string ImageUid,string Filtername)
//{
// // WS which combines two ws calls.
// // First the image given by uids will be moved to
// string TotalResult = string.Empty;
// TotalResult = "Starting";
// string DicomResult= MyDicomConnection.RetrieveImage(ImageUid);

```

```
// //TotalResult = DicomResult;
// //Call FilterJob on svg-demo2
// svg_demo2_WS.Service FilterWS = new svg_demo2_WS.Service();
// string AddFilterResult=FilterWS.WS_MatlabImageJob(StudyInstanceUID,
SerieInstanceUID, ImageUid, Filtername,"ImageLevel");
// TotalResult = string.Concat(TotalResult, AddFilterResult);
// return TotalResult;
//}
[WebMethod]
public string WS_SVGDEMO2_WS_MatlabSerieJob(string StudyInstanceUID, string
SerieInstanceUID, string FilterName, string FilterLevel)
{
// PROXY TO CALL WS ON SVG_DEMO2
string WSResult = string.Empty;
svg_demo2_WS.Service svgdemo2WS = new svg_demo2_WS.Service();
WSResult=svgdemo2WS.WS_MatlabSerieJob(StudyInstanceUID, SerieInstanceUID,
FilterName, FilterLevel);
return WSResult;
}
[WebMethod]
public string WS_SVGDEMO2_WS_MatlabImageJob(string StudyInstanceUID, string
SerieInstanceUID,string ImageUid, string FilterName, string FilterLevel)
{
// PROXY TO CALL WS ON SVG_DEMO2
string WSResult = string.Empty;
svg_demo2_WS.Service svgdemo2WS = new svg_demo2_WS.Service();
WSResult = svgdemo2WS.WS_MatlabImageJob(StudyInstanceUID, SerieInstanceUID,
ImageUid, FilterName, FilterLevel);
return WSResult;
}
[WebMethod]
public XmlDocument WS_SVGDEMO2_WS_AvailableFilter()
{
// PROXY TO CALL WS ON SVG_DEMO2
XmlDocument WSResult = new XmlDocument();
svg_demo2_WS.Service svgdemo2WS = new svg_demo2_WS.Service();
XmlNode myXMLNodeObject = svgdemo2WS.WS_AvailableFilter();
WSResult.AppendChild(WSResult.ImportNode(myXMLNodeObject, true));
return WSResult;
}
}
```

9.5. Kildekode Matlab WS

9.5.1. Service

```

using System;
using System.Collections.Generic;
using System.Collections;
using System.Linq;
using System.Web;
using System.Web.Services;
using System.Diagnostics;
using System.IO;
using System.Xml;
using System.Data;
[WebService(Namespace = "http://tempuri.org/")]
[WebServiceBinding(ConformsTo = WsiProfiles.BasicProfile1_1)]
// To allow this Web Service to be called from script, using ASP.NET AJAX, uncomment the
following line.
// [System.Web.Script.Services.ScriptService]
public class Service : System.Web.Services.WebService
{
    // Developed by : BS
    // Date: April 09
    // Version 1
    // Description:
    // This class contains the webservices needed to communicate with the MatlabWrapper
    program, and the
    // WebServersystem containing the DICOM Webservices located on the svg-demo1 server
    // NOTE : To be able to store files in the given path you have to explicite grant
    permissions to the
    // IIS Worker Process Group (IIS_WPG) on the InBoundQueue folder in the MatlabWrapper
    program
    public Service () {
        //
        //Uncomment the following line if using designed components
        //InitializeComponent();
    }
    [WebMethod]
    public string HelloWorld()
    {
        return "Hello World";
    }
    [WebMethod]
    public string WS_MatlabImageJob(string StudyInstanceUID, string SerieInstanceUID,
    string ImageUid, string FilterName, string FilterLevel)
    {
        //Webservice creating a local file in given path as a imageinstruction for
        MatlabWrapper program
        //Parameters given in WS is stored in the file created as double-pipe seperated
        fields
        string IJobPath = string.Empty;
        string IFilterLevel = string.Empty;
        IJobPath=@"C:\MatlabFilter\InBoundQueue\";
        IFilterLevel = FilterLevel;
        string ThisReq = Convert.ToString(DateTime.Now.Ticks);
        string FileName = string.Concat(IJobPath, ThisReq, ".MatJob");
        StreamWriter wr = new StreamWriter(FileName);
        string RequestText = string.Concat(ThisReq, "|", IFilterLevel, "|", FilterName,
        "|", StudyInstanceUID, "|", SerieInstanceUID, "|", ImageUid);
        wr.WriteLine(RequestText);
        wr.Close();
        return string.Concat(ThisReq, " is placed in queue");//ExitCode;
    }
    [WebMethod]

```

```

public string WS_MatlabSerieJob(string StudyInstanceUID, string SerieInstanceUID,
string FilterName, string FilterLevel)
{
//Webservice creating a local file in given path as a seriesinstruction for
MatlabWrapper program
//Parameters given in WS is stored in the file created as double-pipe seperated
fields
string IJobPath = string.Empty;
string IFilterLevel = string.Empty;
IJobPath = @"C:\MatlabFilter\InBoundQueue\";
IFilterLevel = FilterLevel;
string ThisReq = Convert.ToString(DateTime.Now.Ticks);
string FileName = string.Concat(IJobPath, ThisReq, ".MatJob");
StreamWriter wr = new StreamWriter(FileName);
string RequestText = string.Concat(ThisReq, "||",IFilterLevel, "||", FilterName, "|
|", StudyInstanceUID, "||", SerieInstanceUID);
wr.WriteLine(RequestText);
wr.Close();
return string.Concat(ThisReq, " is placed in queue");//ExitCode;
}
[WebMethod]
public XmlDocument WS_AvailableFilter()
{
// WS used give the initator a xmldocument of the available filters configured in
the system
string ConfFile = null;
ConfFile = @"C:\Inetpub\wwwroot\Matlab\MatlabFilters.xml";
XmlDocument MyFilters = new XmlDocument();
MyFilters.Load(ConfFile);
return MyFilters;
}
[WebMethod]
public string WS_FilterMfile(string FilterLevel,string FilterDesc)
{
// WS reports a string containing the associated m-file to the given description
// Filterlevel distinguish between image and serieslevel
string ConfFile = null;
ConfFile = @"C:\Inetpub\wwwroot\Matlab\MatlabFilters.xml";
XmlDocument MyFilters = new XmlDocument();
MyFilters.Load(ConfFile);
XmlElement XmlRoot = MyFilters.DocumentElement;
XmlNodeList nodes = XmlRoot.GetElementsByTagName(FilterDesc);
XmlNode node =nodes.Item(0);
string res=node[FilterLevel].ChildNodes.Item(0).InnerText.ToString();
return res;
}
[WebMethod]
public string WS_OutStoreSerie(string StudyInstanceUID, string SerieInstanceUID)
{
// PROXY TO CALL WS ON SVG_DEMO1
// This WS is a proxyhost for the svg-demo1 Webserver. Local request is rerouted
to the
// other webserver system. The result is that the Matlabwrapperprogram is unavare
of the other wsserver
string WSResult = string.Empty;
svg_demo1.WebServices svgdemo1WS = new svg_demo1.WebServices();
WSResult = svgdemo1WS.WS_StoreSerie(StudyInstanceUID, SerieInstanceUID);
return WSResult;
}
}

```

9.5.2. MatlabFilters

```
<?xml version="1.0" encoding="utf-8" ?>
<Configuration>
<MatlabFilters>
  <MyEdgeFilterDemo>
    <ImageLevel>
      <M-File>MyEdgeFilter</M-File>
    </ImageLevel>
  </MyEdgeFilterDemo>
  <PossibleArtefact>
    <ImageLevel>
      <M-File>ArtInIm</M-File>
    </ImageLevel>
  </PossibleArtefact>
  <EdgeDetectionWriteBack>
    <SeriesLevel>
      <M-File>EdgeWithDicomWB</M-File>
    </SeriesLevel>
  </EdgeDetectionWriteBack>
  <SineScrollDemo>
    <SeriesLevel>
      <M-File>SineScroll</M-File>
    </SeriesLevel>
  </SineScrollDemo>
  <MR_VolumeRender>
    <SeriesLevel>
      <M-File>MyVolFilterSerie</M-File>
    </SeriesLevel>
  </MR_VolumeRender>
  <PossibleArtefactSerie>
    <SeriesLevel>
      <M-File>ArtInSerie</M-File>
    </SeriesLevel>
  </PossibleArtefactSerie>
</MatlabFilters>
</Configuration>
```

9.6. Kildekode MatlabWrapper

9.6.1. Program

```

using System;
using System.Collections.Generic;
using System.Collections;
using System.Linq;
using System.Text;
using System.Windows.Forms;
using System.IO;
namespace MatlabWrapper
{
    ///Developed by: bs
    ///Develop time: march 2009
    ///Version : 0,5
    ///Description : Main program for Matlab and .net communication
    ///
    ///This is the main program, this instanciates the program, setting the
    ///appropriate parameters to the application and the notifyicon
    static class Program
    {
        [STAThread]
        static void Main()
        {
            Application.EnableVisualStyles();
            Application.SetCompatibleTextRenderingDefault(false);
            //Open connection to my functions class
            MatlabWrapper.MatlabWrapperClass MyFunctions = new MatlabWrapper.
            MatlabWrapperClass();
            MyFunctions.iniSettings();
            //MyFunctions.MatlabConnection();
            MyFunctions.StartWatchingOutBound();
            MyFunctions.StartWatchingInBound();
            MyFunctions.PopulateMenu();
            Application.Run();
        }
    }
}

```

9.6.2. MatlabWrapper (inifil)

```

[Matlab Wrapper Settings]
StorageLocation=D:\
MatlabJobQueueLocation=C:\MatlabFilter\InBoundQueue\
DicomJobQueueLocation=C:\MatlabFilter\OutBoundQueue\
MatlabFilterLocation=C:\MatlabFilter\
FilterDefinitionFile=C:\inetpub\wwwroot\Matlab\MatlabFilters.xml

```

9.6.3. MatlabWrapperClass

```

using System;
using System.Collections.Generic;
using System.Collections;
using System.Linq;
using System.Text;
using System.Runtime.InteropServices;
using System.Windows.Forms;
using System.Drawing;
using System.IO;
using System.Xml;
using log4net;
using log4net.Config;
namespace MatlabWrapper
{
    public class MatlabWrapperClass
    {
        // Developed by : BS
        // Date: April 09
        // Version 1
        // Description:
        // This class has got the MatlabWrapper program logic. Basic idea is to open a com
        automation object
        // and communicate with this, parsing in the appropriate m-file and params
        // Workflow is based on messages in and out using the InBoundQueue and the
        OutBoundQueue
        // This class has got a file delegate watching for new files in these folders
        //
        // Instanciating a new Matlab COM Automation Object to be used
        MApp.MLAppClass matlab = new MApp.MLAppClass();
        // Declaring a local class
        MatlabWrapper.iniApi myIni = new MatlabWrapper.iniApi();
        //Logger to be used
        ILog Logger = LogManager.GetLogger(typeof(MatlabWrapperClass));
        //class variables to be used
        string MatlabJobQueue;
        string DicomJobQueue;
        string localStorageLocation;
        string MatlabFilterLocation;
        string FilterDefinitionFile;
        public Hashtable GetINIValues()
        {
            //
            string defaultValue = "???";
            string iniFile = AppPath() + "\\MatlabWrapper.ini";
            Hashtable iniTable = new Hashtable();
            // Reading the content of the MatlabWrapper.ini
            List<string> categories = iniApi.GetCategories(iniFile);
            foreach (string category in categories)
            {
                /*
                * Get the keys
                */
                List<string> keys = iniApi.GetKeys(iniFile, category);
                foreach (string key in keys)
                {
                    /*
                    * Now storing the content in the List collection
                    */
                    string content = iniApi.GetIniFileString(iniFile, category, key,
                    defaultValue);
                    iniTable.Add(key, content);
                }
            }
        }
    }
}

```

```

return iniTable;
}
public string AppPath()
{
//Getting the application environment run area
string path;
path = System.IO.Path.GetDirectoryName(
System.Reflection.Assembly.GetExecutingAssembly().GetName().CodeBase);
return path.Substring(6);
}
public void PopulateMenu()
{
// instanciating a ContextMenu object
ContextMenuStrip contextMenu1 = new ContextMenuStrip();
//instanciating an Icon to be used by the Contextmenu
NotifyIcon notifyIcon1 = new NotifyIcon();
// The actual ToolStripMenuItems
// Populating the NotifyMenu for the application, fetching the values
ToolStripMenuItem IStor = new ToolStripMenuItem("StorageLocation
: " + localStorageLocation);
ToolStripMenuItem IQueue = new ToolStripMenuItem("MatlabQueueLocation : "
+ MatlabJobQueue);
ToolStripMenuItem IFilterIn = new ToolStripMenuItem("DicomJobQueueLocation : "
+ DicomJobQueue);
ToolStripMenuItem IFilterOut = new ToolStripMenuItem("MatlabFilterLocation
: " + MatlabFilterLocation);
ToolStripMenuItem IDefFile = new ToolStripMenuItem("FilterDefinitionFile
: " + FilterDefinitionFile);
//Adding the items to the ContextMenu object
contextMenu1.Items.Add(IStor);
contextMenu1.Items.Add(IQueue);
contextMenu1.Items.Add(IFilterIn);
contextMenu1.Items.Add(IFilterOut);
contextMenu1.Items.Add(IDefFile);
// assigning a custom eventhandler for my contextmenuobject
//
contextMenu1.Click += new EventHandler(contextMenu1_Click);
MatlabWrapperClass Path = new MatlabWrapperClass();
//Specifying some properties to this object
notifyIcon1.Icon = new Icon(Path.AppPath() + "\\Matlab.ico");
notifyIcon1.Text = "MATLAB WRAPPER SETTINGS";
notifyIcon1.ContextMenuStrip = contextMenu1;
notifyIcon1.Visible = true;
}
public void contextMenu1_Click(object Sender, EventArgs e)
{
//Click eventhandler for my ContextMenu
if (MessageBox.Show("Do you want to stop the Matlab interface?", "Confirm stop
", MessageBoxButtons.YesNo) == DialogResult.Yes)
{
//Calling needed garbagecollections/dispose of objects
StopWatching();
Application.Exit();
}
}
public void iniSettings()
{
//Reading logsettings
log4net.Config.XmlConfigurator.Configure();
// Getting my ini static variables
Hashtable myIniValues = new Hashtable();
myIniValues = GetINIValues();
//assigning these variables the values
MatlabJobQueue = myIniValues["MatlabJobQueueLocation"].ToString();
DicomJobQueue = myIniValues["DicomJobQueueLocation"].ToString();
localStorageLocation = myIniValues["StorageLocation"].ToString();
MatlabFilterLocation = myIniValues["MatlabFilterLocation"].ToString();
}

```



```

FilterDefinitionFile = myIniValues["FilterDefinitionFile"].ToString();
}
public void StartWatchingInBound()
{
// Creating a watcher object to control my InBoundQueue file area
FileSystemWatcher watcherIn = new FileSystemWatcher();
//Specify properties for this object
//The folder to watch
watcherIn.Path = MatlabJobQueue;
// Which filetypes to look for
watcherIn.Filter = "*.MatJob";
// Which fileproperties could interest me
watcherIn.NotifyFilter = NotifyFilters.FileName |
NotifyFilters.Attributes |
NotifyFilters.LastAccess |
NotifyFilters.LastWrite |
NotifyFilters.Security |
NotifyFilters.Size;
// The eventhandlers enabled in the watch object
// The only one I have interest in is the Creation of files in the folder
//watcher.Changed += new FileSystemEventHandler(OnFileEvent);
watcherIn.Created += new FileSystemEventHandler(OnFileInEvent);
//watcher.Deleted += new FileSystemEventHandler(OnFileEvent);
//watcher.Renamed += new RenamedEventHandler(OnRenameEvent);
//watcher.IncludeSubdirectories = true;
// Tell the object to start the logging
watcherIn.EnableRaisingEvents = true;
}
public void StartWatchingOutBound()
{
// Creating a watcher object to control my OutBoundQueue file area
// The logic of the Objectproperties can be read from the
// public void StartWatchingInBound() method
FileSystemWatcher watcherOut = new FileSystemWatcher();
watcherOut.Path = DicomJobQueue;
watcherOut.Filter = "*.DicJob";
watcherOut.NotifyFilter = NotifyFilters.FileName |
NotifyFilters.Attributes |
NotifyFilters.LastAccess |
NotifyFilters.LastWrite |
NotifyFilters.Security |
NotifyFilters.Size;
watcherOut.Created += new FileSystemEventHandler(OnFileOutEvent);
watcherOut.EnableRaisingEvents = true;
}
public void OnFileInEvent(object source, FileSystemEventArgs fsea)
{
// Method being fired by the watcher object looking in the InBoundQueue folder
//
//Time to start processing in Matlab
//call ReadJob
//Populating a array containing the instructions in the newly created file
Logger.Info(string.Concat("Inbound Message detected ", DateTime.Now.ToString()
, ":", DateTime.Now.Millisecond.ToString()));
string[] myMatJob = MatJob(fsea.FullPath);
// Executing a addpath command to Matlab. This is giving Matlab the path to
the m-files
try
{
matlab.Execute(string.Concat("addpath ", MatlabFilterLocation));
}
catch (COMException COM)
{
MessageBox.Show(COM.InnerException.ToString());
}
string DirFileName = string.Empty;
string DirName = string.Empty;

```

```

string IFilterLevel = string.Empty;
IFilterLevel = myMatJob[1];
// checking to see which type of job being processed
if (IFilterLevel == "ImageLevel")
{
DirFileName = string.Concat(localStoragelocation, myMatJob[3], "\\",
myMatJob[4], "\\", myMatJob[5], ".dcm");
string FilterDesc = string.Empty;
FilterDesc = myMatJob[2];
//Execute
string MatlabCommand = string.Concat(WS_GetMfile("ImageLevel", FilterDesc)
, "(" + DirFileName + ");");
Logger.Info(string.Concat("ImageJob executed ", DateTime.Now.ToString(), "
:", DateTime.Now.Millisecond.ToString()));
matlab.Execute(string.Concat(MatlabCommand));
}
else
{
// This is SerieLevel
DirName = string.Concat(localStoragelocation, myMatJob[3], "\\", myMatJob
[4]);
string FilterDesc = string.Empty;
FilterDesc = myMatJob[2];
//Check for paramarea
string MatlabCommand = string.Concat(WS_GetMfile("SeriesLevel",
FilterDesc), "(" + DirName + ");");
Logger.Info(string.Concat("SeriesJob executed ", DateTime.Now.ToString(), "
:", DateTime.Now.Millisecond.ToString()));
matlab.Execute(MatlabCommand);
}
Logger.Info(string.Concat("Inbound Job completed ", DateTime.Now.ToString(), "
:", DateTime.Now.Millisecond.ToString()));
}
public string WS_GetMfile(string FilterLevel,string FilterDesc)
{
//Getting the appropriate m-file to execute based on the filterdescription
WS_Matlab.ServiceSoapClient MyMatlabRequest = new WS_Matlab.ServiceSoapClient
();
return MyMatlabRequest.WS_FilterMfile(FilterLevel,FilterDesc);
}
public void OnFileOutEvent(object source, FileSystemEventArgs fsea)
{
//Time to start reading the result from Matlab
//File instruction created from m-file
Logger.Info(string.Concat("Outbound Message detected ", DateTime.Now.ToString
()), ":", DateTime.Now.Millisecond.ToString());
string[] myDicJob = DicJob(fsea.FullPath);
string IStudyUID = string.Empty;
string ISeriesUID = string.Empty;
IStudyUID = myDicJob[2];
ISeriesUID = myDicJob[3];
//Calling the local proxy ws (actually on the svg-demo1 server
// Telling the system to do a Dicom C-Store on the imagefiles
WS_Matlab.ServiceSoapClient DICOMJOB = new MatlabWrapper.WS_Matlab.
ServiceSoapClient();
string Result = string.Empty;
Result = DICOMJOB.WS_OutStoreSerie(IStudyUID, ISeriesUID);
//MessageBox.Show(Result);
Logger.Info(string.Concat("Outbound Job completed ", DateTime.Now.ToString(), "
:", DateTime.Now.Millisecond.ToString()));
}
public string[] MatJob(string JobID)
{
// Method to read the job in the InBoundQueue
// Being sure that the file is complete
while (IsFileLocked(JobID) == true)
{

```

```
Application.DoEvents();
}
//Reading the Job Content, returning the instructions in an array
StreamReader JobRead = new StreamReader(JobID);
string JobLine = string.Empty;
JobLine = JobRead.ReadLine();
string[] IMatJob = JobLine.Split(new string[] { "|" }, StringSplitOptions
.None);
JobRead.Close();
return IMatJob;
}
public string[] DicJob(string JobID)
{
// Method to read the job in the OutBoundQueue
// Being sure that the file is complete
while (IsFileLocked(JobID) == true)
{
Application.DoEvents();
}
//Reading the Job Content, returning the instructions in an array
StreamReader JobRead = new StreamReader(JobID);
string JobLine = string.Empty;
JobLine = JobRead.ReadLine();
string[] IDicJob = JobLine.Split(new string[] { "|" }, StringSplitOptions.
None);
JobRead.Close();
return IDicJob;
}
public bool IsFileLocked(string strFullFileName)
{
//Helper method to check if the file is completed (Closed by the originator)
bool blnReturn = false;
System.IO.FileStream fs;
try
{
fs = System.IO.File.Open(strFullFileName, System.IO.FileMode.OpenOrCreate,
System.IO.FileAccess.Read, System.IO.FileShare.None);
fs.Close();
}
catch (System.IO.IOException)
{
blnReturn = true;
}
return blnReturn;
}
public void StopWatching()
{
//Dispose of the watchers (Don't actually need)
//Termination of the program will destroy them
//(Application running with constant watching)
}
}
}
```

9.6.4. IniApi

```

using System;
using System.Collections.Generic;
using System.Collections;
using System.Linq;
using System.Text;
using System.Runtime.InteropServices;
namespace MatlabWrapper
{
    public class iniApi
    {
        // Class uses old method of reading ini files.
        // Used in my program to give the users a common
        // configuration point.
        // Declaring the methods to read using native API
        [DllImport("KERNEL32.DLL", EntryPoint = "GetPrivateProfileStringW",
        SetLastError = true,
        CharSet = CharSet.Unicode, ExactSpelling = true,
        CallingConvention = CallingConvention.StdCall)]
        private static extern int GetPrivateProfileString(
        string lpAppName,
        string lpKeyName,
        string lpDefault,
        string lpReturnString,
        int nSize,
        string lpFilename);
        public static string GetIniFileString(string iniFile, string category, string key,
        string defaultValue)
        {
            //Gets the actual Value from the inifile, based on the Category and Key value
            string returnString = new string(' ', 1024);
            GetPrivateProfileString(category, key, defaultValue, returnString, 1024,
            iniFile);
            return returnString.Split('\0')[0];
        }
        public static List<string> GetKeys(string iniFile, string category)
        {
            // Generates a List collection of all Keys in inifile
            string returnString = new string(' ', 32768);
            GetPrivateProfileString(category, null, null, returnString, 32768, iniFile);
            List<string> result = new List<string>(returnString.Split('\0'));
            result.RemoveRange(result.Count - 2, 2);
            return result;
        }
        // Generates a List collection of all Categories in given inifile
        public static List<string> GetCategories(string iniFile)
        {
            string returnString = new string(' ', 65536);
            GetPrivateProfileString(null, null, null, returnString, 65536, iniFile);
            List<string> result = new List<string>(returnString.Split('\0'));
            result.RemoveRange(result.Count - 2, 2);
            return result;
        }
    }
}

```

9.7. Kildekode StudyCandidates

```

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Windows.Forms;
using System.Xml;
using System.Xml.XPath;
using System.IO;
using System.Collections;
using log4net;
using log4net.Config;
// Developed by : BS
// Date: April 09
// Version 1
// Description:
// StudyCandidates is a prototype application to verify webservices generated
// for the Master Thesis.
// Basicly it uses the available webservices from the svg-demo1 server
// Services from svg-demo2 is not referenced, but are used from the proxy
// services from svg-demo1
//
namespace StudyCandidates
{
public partial class Main : Form
{
// global params for gridposition
int yCoord;
int ySerieCoord;
int yImageCoord;
int Imagenr = 0;
// using Log4Net to provide logging in app
ILog Logger = LogManager.GetLogger(typeof(Main));
// Generates a global identification to my webservices
WS_DICOM.WebServicesSoapClient MyMatlabRequest = new StudyCandidates.WS_DICOM.
WebServicesSoapClient();
public Main()
{
// initalizing logger
InitializeComponent();
log4net.Config.XmlConfigurator.Configure();
}
public void CreateDicomStudiesGrid(XmlDocument MyRawResult)
{
//Generates the first grid showing available studies
DataSet dstudy = new DataSet();
DataTable DicomTable = dstudy.Tables.Add();
DicomTable.Columns.Add("Studydate", typeof(string));
DicomTable.Columns.Add("Modality", typeof(string));
DicomTable.Columns.Add("Study Description", typeof(string));
DicomTable.Columns.Add("Accession Number", typeof(string));
DicomTable.Columns.Add("Study Instance UID", typeof(string));
XmlDocument doc = new XmlDocument();
// loading the queried result
doc.Load("temp.xml");
//ds.ReadXml(doc.InnerXml);
XmlElement XmlRoot = MyRawResult.DocumentElement;
XmlNodeList XmlItemNodes = XmlRoot.GetElementsByTagName("Studydate");
//Looping the result from ws
foreach (XmlNode node in XmlItemNodes)
{

```

```

DicomTable.Rows.Add(node.NamespaceURI, node.ChildNodes[0].NamespaceURI,
node.ChildNodes[1].NamespaceURI, node.ChildNodes[2].NamespaceURI, node.ChildNodes[3].
NamespaceURI);
}
dataGridView1.DataSource = dstudy.Tables[0];
dataGridView1.Columns[0].Width = 130;
dataGridView1.Columns[1].Width = 75;
dataGridView1.Columns[2].Width = 250;
dataGridView1.Columns[3].Width = 150;
dataGridView1.Columns[4].Width = 350;
dataGridView1.Refresh();
Cursor.Current = Cursors.Default;
}
public void CreateDicomSeriesGrid(XmlDocument MyRawResult)
{
//Creates the series grid, based on the ws result
DataSet dserie = new DataSet();
DataTable DicomTableSerie = dserie.Tables.Add();
DicomTableSerie.Columns.Add("Seriernr", typeof(string));
DicomTableSerie.Columns.Add("SerieDescription", typeof(string));
DicomTableSerie.Columns.Add("ImagesInSerie", typeof(string));
DicomTableSerie.Columns.Add("SerieInstanceUid", typeof(string));
XmlDocument doc = new XmlDocument();
doc.Load("tempserie.xml");
//ds.ReadXml(doc.InnerXml);
XmlElement XmlRoot = MyRawResult.DocumentElement;
XmlNodeList XmlItemNodes = XmlRoot.GetElementsByTagName("SerieInstanceUid");
foreach (XmlNode node in XmlItemNodes)
{
DicomTableSerie.Rows.Add(node.ChildNodes[1].NamespaceURI, node.ChildNodes
[0].NamespaceURI, node.ChildNodes[2].NamespaceURI, node.NamespaceURI);
}
dataGridView2.DataSource = dserie.Tables[0];
dataGridView2.Columns[0].Width = 50;
dataGridView2.Columns[1].Width = 300;
dataGridView2.Columns[2].Width = 100;
dataGridView2.Columns[3].Width = 320;
dataGridView2.Refresh();
Cursor.Current = Cursors.Default;
}
public void CreateDicomImagesGrid(XmlDocument MyRawResult)
{
//Creates the grid showing the image results
DataSet dimages = new DataSet();
DataTable DicomTableImages = dimages.Tables.Add();
DicomTableImages.Columns.Add("Imagenr", typeof(string));
DicomTableImages.Columns.Add("NrOfFrames", typeof(string));
DicomTableImages.Columns.Add("ImageUID", typeof(string));
//DicomTableImages.Columns.Add("SOP Class identifier", typeof(string));
DicomTableImages.Columns.Add("SerieUID", typeof(string));
DicomTableImages.Columns.Add("StudyUID", typeof(string));
XmlDocument doc = new XmlDocument();
doc.Load("tempimages.xml");
//ds.ReadXml(doc.InnerXml);
XmlElement XmlRoot = MyRawResult.DocumentElement;
XmlNodeList XmlItemNodes = XmlRoot.GetElementsByTagName("SOPInstanceUid");
foreach (XmlNode node in XmlItemNodes)
{
DicomTableImages.Rows.Add(node.ChildNodes[0].NamespaceURI, node.ChildNodes
[1].NamespaceURI, node.NamespaceURI, node.ParentNode.NamespaceURI, node.ParentNode.
ParentNode.NamespaceURI);
}
dataGridView3.DataSource = dimages.Tables[0];
dataGridView3.Columns[0].Width = 50;
dataGridView3.Columns[1].Width = 75;
dataGridView3.Columns[2].Width = 320;
dataGridView3.Columns[3].Width = 320;

```

```

dataGridView3.Columns[4].Width = 320;
dataGridView3.Refresh();
Cursor.Current = Cursors.Default;
}
private void button1_Click_1(object sender, EventArgs e)
{
// starts to query for objects
Logger.Info(string.Concat("Query Started ", DateTime.Now.ToString(), ":",
DateTime.Now.Millisecond.ToString()));
Cursor.Current = Cursors.WaitCursor;
//Clearing datagrid
DataSet ds = new DataSet();
dataGridView1.DataSource = ds;
dataGridView2.DataSource = ds;
dataGridView3.DataSource = ds;
WS_DICOM.WebServicesSoapClient MyRequest = new StudyCandidates.WS_DICOM.
WebServicesSoapClient();
//WebServicesSoapClient sdf = new TestWS.MyServiceReferense.
WebServicesSoapClient();
XmlDocument MyXML = new XmlDocument();
string DateRangeFrom = dateTimePicker1.Value.ToString("yyyyMMdd");
string DateRangeTo = dateTimePicker2.Value.ToString("yyyyMMdd");
string DateRange = string.Concat(DateRangeFrom, "-", DateRangeTo);
string QueryModality = comboBox1.Text.ToString();
string StudyDescription = textBox1.Text;
string Sex = textBox2.Text;
// the actual ws result
XmlNode myXMLNodeObject = MyRequest.WS_GetStudiesCandidates(DateRange,
QueryModality, StudyDescription, Sex);
Logger.Info(string.Concat("Query Result retrieved ", DateTime.Now.ToString(),
":", DateTime.Now.Millisecond.ToString()));
MyXML.AppendChild(MyXML.ImportNode(myXMLNodeObject, true));
// trying to save file
StreamWriter wr = new StreamWriter("temp.xml");
string RequestText = MyXML.InnerXml;
wr.WriteLine(RequestText);
wr.Close();
CreateDicomStudiesGrid(MyXML);
label7.Text = string.Concat(dataGridView1.RowCount.ToString(), " Studies");
Logger.Info(string.Concat(dataGridView1.RowCount.ToString(), "Studies found"))
;
Logger.Info(string.Concat("Query Completed ", DateTime.Now.ToString(), ":",
DateTime.Now.Millisecond.ToString()));
}
private void dataGridView1_CellClick(object sender, DataGridViewCellEventArgs e)
{
// Generating Context menu
//contextMenuStrip1.Items.Add("Move to Local Store");
Logger.Info(string.Concat("Query For Series in Study start ", DateTime.Now.
ToString(), ":", DateTime.Now.Millisecond.ToString()));
ContextMenuStrip myMenu = new ContextMenuStrip();
myMenu.Items.Add("Move this study to local store");
myMenu.Items.Add("Move all studies from studyselection to local store");
dataGridView1.ContextMenuStrip = myMenu;
// adding a handle to the click event
myMenu.ItemClicked += new ToolStripItemClickedEventHandler
(myMenuStudy_ItemClicked);
DataSet ds = new DataSet();
dataGridView2.DataSource = ds;
dataGridView3.DataSource = ds;
Application.UseWaitCursor = true;
this.Cursor = Cursors.WaitCursor;
Cursor.Current = Cursors.WaitCursor;
// dataGridView3.Dispose();
yCoord = dataGridView1.CurrentCellAddress.Y; // You can get X if you need it.
WS_DICOM.WebServicesSoapClient MyRequest = new StudyCandidates.WS_DICOM.
WebServicesSoapClient();

```



```

XmlDocument MyXML = new XmlDocument();
XmlNode myXmlNodeObject = MyRequest.WS_GetStudyDetails(dataGridView1.Rows
[yCoord].Cells[4].Value.ToString());
MyXML.AppendChild(MyXML.ImportNode(myXmlNodeObject, true));
// trying to save file
StreamWriter wr = new StreamWriter("tempserie.xml");
string RequestText = MyXML.InnerXml;
wr.WriteLine(RequestText);
wr.Close();
//Populates grid
CreateDicomSeriesGrid(MyXML);
Application.UseWaitCursor = false;
this.Cursor = Cursors.Default;
Cursor.Current = Cursors.Default;
Logger.Info(string.Concat(dataGridView2.RowCount.ToString(), " Series found"));
Logger.Info(string.Concat("Query For Series in Study end ", DateTime.Now.
ToString(), ":", DateTime.Now.Millisecond.ToString()));
}
private void WS_GetStudyDetailsGrid(string StudyInstanceUID)
{
WS_DICOM.WebServicesSoapClient MyRequest = new StudyCandidates.WS_DICOM.
WebServicesSoapClient();
//.WebServicesSoapClient sdf = new TestWS.MyServiceReferense.
WebServicesSoapClient();
XmlDocument MyXML = new XmlDocument();
XmlNode myXmlNodeObject = MyRequest.WS_GetStudyDetails(StudyInstanceUID);
MyXML.AppendChild(MyXML.ImportNode(myXmlNodeObject, true));
StreamWriter wr = new StreamWriter("tempstudy.xml");
string RequestText = MyXML.InnerXml;
wr.WriteLine(RequestText);
wr.Close();
Application.UseWaitCursor = false;
this.Cursor = Cursors.Default;
Cursor.Current = Cursors.Default;
}
private ArrayList WS_GetImagesFromGivenSerieDesc(string StudyInstanceUID, string
SerieDescription)
{
WS_DICOM.WebServicesSoapClient MyRequest = new StudyCandidates.WS_DICOM.
WebServicesSoapClient();
//.WebServicesSoapClient sdf = new TestWS.MyServiceReferense.
WebServicesSoapClient();
XmlDocument MyXML = new XmlDocument();
XmlNode myXmlNodeObject = MyRequest.WS_GetStudyDetails(StudyInstanceUID);
MyXML.AppendChild(MyXML.ImportNode(myXmlNodeObject, true));
StreamWriter wr = new StreamWriter("tempStudySerie.xml");
string RequestText = MyXML.InnerXml;
wr.WriteLine(RequestText);
wr.Close();
XmlDocument doc = new XmlDocument();
doc.Load("tempStudySerie.xml");
ArrayList MyStudyWSerie = new ArrayList();
XmlElement XmlRoot = doc.DocumentElement;
XmlNodeList XmlItemNodes = XmlRoot.GetElementsByTagName("SerieInstanceUid");
foreach (XmlNode node in XmlItemNodes)
{
if (node.ChildNodes[0].NamespaceURI == SerieDescription)
{
//get all imageuids
//MyStudyWSerie.Add(string.Concat(StudyInstanceUID,"||",node.
NamespaceURI));
string Serieuid = node.NamespaceURI;
XmlDocument MyXMLSerie = new XmlDocument();
XmlNode myXmlNodeObjectSerie = MyRequest.WS_GetSerieDetails
(StudyInstanceUID, Serieuid);
MyXMLSerie.AppendChild(MyXMLSerie.ImportNode(myXmlNodeObjectSerie,
true));
}
}
}

```



```

StreamWriter wrSerie = new StreamWriter("tempStudySerieImage.xml");
string RequestTextImage = MyXMLSerie.InnerXml;
wrSerie.WriteLine(RequestTextImage);
wrSerie.Close();
XmlDocument docImage = new XmlDocument();
docImage.Load("tempStudySerieImage.xml");
//ds.ReadXml(doc.InnerXml);
XmlElement XmlRootImage = docImage.DocumentElement;
XmlNodeList XmlNodeNodesImages = XmlRootImage.GetElementsByTagName(
"SOPInstanceUid");
foreach (XmlNode nodeImage in XmlNodeNodesImages)
{
// this.label9.Text = Imagenr.ToString();
MyStudyWSerie.Add(string.Concat(StudyInstanceUID,"||", Serieuid,
"||",nodeImage.NamespaceURI));
// Imagenr = Imagenr + 1;
// this.Refresh();
// Application.DoEvents();
}
}
return MyStudyWSerie;
}
private void dataGridView2_CellClick(object sender, DataGridViewCellEventArgs e)
{
Logger.Info(string.Concat("Query For images in serie start ", DateTime.Now.
ToString(), ":", DateTime.Now.Millisecond.ToString()));
// Generating Context menu for series
//contextMenuStrip1.Items.Add("Move to Local Store");
//Retrieve Available filters from Matlab
ArrayList AvailableFilters = new ArrayList();
AvailableFilters = WS_GetAvailableFilters("SeriesLevel");
// Generating Context menu for series
//contextMenuStrip1.Items.Add("Move to Local Store");
ContextMenuStrip myMenuSerie = new ContextMenuStrip();
myMenuSerie.Items.Add("Move this serie to local store");
myMenuSerie.Items.Add("Move all similar series from studyselection to local
store");
ToolStripSeparator sep = new ToolStripSeparator();
myMenuSerie.Items.Add(sep);
foreach (string filter in AvailableFilters)
{
myMenuSerie.Items.Add(filter);
}
dataGridView2.ContextMenuStrip = myMenuSerie;
myMenuSerie.ItemClicked += new ToolStripItemClickedEventHandler
(myMenuSerie_ItemClicked);
//
Application.UseWaitCursor = true;
this.Cursor = Cursors.WaitCursor;
Cursor.Current = Cursors.WaitCursor;
//Clearing Imagegrid
DataSet ds = new DataSet();
dataGridView3.DataSource = ds;
ySerieCoord = dataGridView2.CurrentCellAddress.Y; // You can get X if you need
it.
WS_DICOM.WebServicesSoapClient MyRequest = new StudyCandidates.WS_DICOM.
WebServicesSoapClient();
XmlDocument MyXML = new XmlDocument();
XmlNode myXMLNodeObject = MyRequest.WS_GetSerieDetails(dataGridView1.Rows
[yCoord].Cells[4].Value.ToString(), dataGridView2.Rows[ySerieCoord].Cells[3].Value.
ToString());
MyXML.AppendChild(MyXML.ImportNode(myXMLNodeObject, true));
// trying to save file
StreamWriter wr = new StreamWriter("tempimages.xml");
string RequestText = MyXML.InnerXml;
wr.WriteLine(RequestText);

```

```

wr.Close();
CreateDicomImagesGrid(MyXML);
Application.UseWaitCursor = false;
this.Cursor = Cursors.Default;
Cursor.Current = Cursors.Default;
Logger.Info(string.Concat(dataGridView3.RowCount.ToString(), " images found"));
Logger.Info(string.Concat("Query For Series in Study start ", DateTime.Now.
ToString(), ":", DateTime.Now.Millisecond.ToString()));
}
private void dataGridView3_CellClick(object sender, DataGridViewCellEventArgs e)
{
//Retrieve Available filters from Matlab
ArrayList AvailableFilters = new ArrayList();
AvailableFilters = WS_GetAvailableFilters("ImageLevel");
// Generating Context menu for series
//contextMenuStrip1.Items.Add("Move to Local Store");
ContextMenuStrip myMenuImages = new ContextMenuStrip();
myMenuImages.Items.Add("Move this image to local store");
ToolStripSeparator sep = new ToolStripSeparator();
myMenuImages.Items.Add(sep);
foreach (string filter in AvailableFilters)
{
myMenuImages.Items.Add(filter);
}
//myMenuImages.Items.Add("Move all images to local store");
dataGridView3.ContextMenuStrip = myMenuImages;
myMenuImages.ItemClicked += new ToolStripItemClickedEventHandler
(myMenuImages_ItemClicked);
//
Application.UseWaitCursor = true;
this.Cursor = Cursors.WaitCursor;
Cursor.Current = Cursors.WaitCursor;
yImageCoord = dataGridView3.CurrentCellAddress.Y; // You can get X if you need
it.
Application.UseWaitCursor = false;
this.Cursor = Cursors.Default;
Cursor.Current = Cursors.Default;
}
private ArrayList WS_GetAvailableFilters(string QueryLevel)
{
// get a list of all available filters from ws
XmlDocument MyXML = new XmlDocument();
XmlNode myXMLNodeObject = MyMatlabRequest.WS_SVGDEMO2_WS_AvailableFilter();
MyXML.AppendChild(MyXML.ImportNode(myXMLNodeObject, true));
XmlElement XmlRoot = MyXML.DocumentElement;
XmlNode node = XmlRoot.SelectSingleNode("MatlabFilters");
ArrayList Filters = new ArrayList();
for (int i = 0; i < node.ChildNodes.Count; i++)
{
if (node.ChildNodes[i].ChildNodes[0].Name == QueryLevel)
{
Filters.Add(node.ChildNodes[i].Name);
}
}
//foreach (XmlNode node in XmlItemNodes)
//{
// while (node.ChildNodesnode.LastChild=true)
// Filters.Add(node.ChildNodes[0].Name);
//}
return Filters;
}
private void myMenuStudy_ItemClicked(object sender, ToolStripItemClickedEventArgs
e)
{
Imagenr = 0;
if (e.ClickedItem.Text == "Move this study to local store")
{

```

```

Logger.Info(string.Concat("Move study start ", DateTime.Now.ToString(), ":
", DateTime.Now.Millisecond.ToString()));
MoveStudy(dataGridView1.Rows[yCoord].Cells[4].Value.ToString());
//MessageBox.Show("Flytter denne studien");
Logger.Info(string.Concat("Move study end ", DateTime.Now.ToString(), ":",
DateTime.Now.Millisecond.ToString()));
}
else
{
ArrayList myRow = new ArrayList();
foreach (System.Windows.Forms.DataGridViewRow row in dataGridView1.Rows)
{
//myRow.Add(row.Cells[4].Value.ToString());
MoveStudy(row.Cells[4].Value.ToString());
}
// myRow=dataGridView1.Rows[yCoord].Cells[4].Value.ToString();
//MessageB
MessageBox.Show("Alle studier flyttet");
}
}
private void myMenuSeries_ItemClicked(object sender, ToolStripItemClickedEventArgs
e)
{
//need to check if this is filterjob
if (e.ClickedItem.Text == "Move this serie to local store")
{
// Not a filterjob
MoveSerie(this.dataGridView2.Rows[ySerieCoord].Cells[3].Value.ToString());
//MessageBox.Show("Flytter denne studien");
}
else if (e.ClickedItem.Text == "Move all similar series from studyselction to
local store")
{
MoveRelatedSeries("NONE");
}
else
{
if (MessageBox.Show("Apply on all related?", "Apply filter on all similar
series from studyselction", MessageBoxButtons.YesNo) == DialogResult.No)
{
//No just on this
MoveSerie(this.dataGridView2.Rows[ySerieCoord].Cells[3].Value.ToString
());
//MessageBox.Show("Flytter denne studien");
//APPLY FILTER
WS_PostMatlabSerieJob(e.ClickedItem.Text, this.dataGridView1.Rows
[yCoord].Cells[4].Value.ToString(), this.dataGridView2.Rows[ySerieCoord].Cells[3].
Value.ToString());
}
else
{
//Apply on all
MoveRelatedSeries(e.ClickedItem.Text);
}
}
}
private void myMenuImages_ItemClicked(object sender, ToolStripItemClickedEventArgs
e)
{
Imagenr = 0;
if (e.ClickedItem.Text == "Move this image to local store")
{
MoveImage(this.dataGridView3.Rows[yImageCoord].Cells[2].Value.ToString());
//MessageBox.Show("Flytter denne studien");
}
else
{

```

```

//FilterJOB
//Get FilterName
string IFiltername = e.ClickedItem.Text;
string Istudyuid = this.dataGridView3.Rows[yImageCoord].Cells[4].Value.
ToString();
string Iserieuid = this.dataGridView3.Rows[yImageCoord].Cells[3].Value.
ToString();
string IimageUID = this.dataGridView3.Rows[yImageCoord].Cells[2].Value.
ToString();
//Move image to location
MoveImage(this.dataGridView3.Rows[yImageCoord].Cells[2].Value.ToString());
//Post a Matjob to Matlab
WS_PostMatlabImageJob(IFiltername, Istudyuid, Iserieuid, IimageUID);
}
}
private void MoveStudy(string studyUID)
{
WS_DICOM.WebServicesSoapClient MyRequest = new StudyCandidates.WS_DICOM.
WebServicesSoapClient();
MyRequest.WS_RetrieveStudy(studyUID);
}
private void MoveSerie(string serieUID)
{
//WS_DICOM.WebServicesSoapClient MyRequest = new StudyCandidates.WS_DICOM.
WebServicesSoapClient();
//MyRequest.WS_RetrieveSerie(serieUID);
//ClearCanvas has a bug in retrieve series. The retrieval of all images in one
serie simulates this
foreach (System.Windows.Forms.DataGridViewRow row in dataGridView3.Rows)
{
MoveImage(row.Cells[2].Value.ToString());
}
}
private void MoveRelatedSeries(string Filter)
{
ArrayList myStudyUIDList = new ArrayList();
string MyRelatedSeriesDesc = string.Empty;
MyRelatedSeriesDesc = this.dataGridView2.Rows[ySerieCoord].Cells[1].
Value.ToString();
foreach (System.Windows.Forms.DataGridViewRow row in dataGridView1.
Rows)
{
myStudyUIDList.Add(row.Cells[4].Value.ToString());
}
//Now I have all studyuids
//Building Study/Serie UID
ArrayList tempResult = new ArrayList();
ArrayList Result = new ArrayList();
foreach (string studyuid in myStudyUIDList)
{
tempResult = WS_GetImagesFromGivenSerieDesc(studyuid,
MyRelatedSeriesDesc);
if (tempResult.Count > 0)
{
foreach (string studyseriesuid in tempResult)
Result.Add(studyseriesuid);
}
}
//Now Result should have Studyuid||serie||images for all related
series
// adding a distinct resultseriesett
ArrayList ResultSerie = new ArrayList();
foreach (string studyserie in Result)
{
string[] tempserie = studyserie.Split(new string[] { "||" },
StringSplitOptions.None);
//Move All images in list

```

```

MoveImage(tempserie[2]);
if(ResultSerie.Contains(string.Concat(tempserie[0],"||",tempserie
[1])))
{
}
else
{
ResultSerie.Add(string.Concat(tempserie[0],"||",tempserie[1]));
}
}
//Filterjob?
if (Filter == "NONE")
{
}
else
{
//ApplyFilter
//Loop through all distinct series
foreach (string tmpStudyString in ResultSerie)
{
string[] myStudyString = tmpStudyString.Split(new string
[] { "||" }, StringSplitOptions.None);
WS_PostMatlabSerieJob(Filter, myStudyString[0],
myStudyString[1]);
}
}
private void MoveImage(string imageUID)
{
WS_DICOM.WebServicesSoapClient MyRequest = new StudyCandidates.WS_DICOM.
WebServicesSoapClient();
MyRequest.WS_RetrievalImage(imageUID);
}
private void WS_PostMatlabSerieJob(string Filtername, string StudyInstanceUID,
string SerieInstanceUID)
{
string IFiltername = Filtername;
string IStudyInstanceUID = StudyInstanceUID;
string ISerieInstanceUID = SerieInstanceUID;
//Open a WS conn
WS_DICOM.WebServicesSoapClient MyPostMatlabRequest = new WS_DICOM.
WebServicesSoapClient();
string myMatlabPost = MyPostMatlabRequest.WS_SVGDEMO2_WS_MatlabSerieJob
(IStudyInstanceUID, ISerieInstanceUID, IFiltername, "SerieLevel");
}
private void WS_PostMatlabImageJob(string Filtername, string StudyInstanceUID,
string SerieInstanceUID, string ImageUID)
{
string IFiltername = Filtername;
string IStudyInstanceUID = StudyInstanceUID;
string ISerieInstanceUID = SerieInstanceUID;
string IImageUID = ImageUID;
//Open a WS conn
WS_DICOM.WebServicesSoapClient MyPostMatlabRequest = new WS_DICOM.
WebServicesSoapClient();
string myMatlabPost = MyPostMatlabRequest.WS_SVGDEMO2_WS_MatlabImageJob
(IStudyInstanceUID,ISerieInstanceUID, IImageUID, IFiltername, "ImageLevel");
}
}
}
}

```

9.8. M-Filer

9.8.1. MatlabFilter 1: EdgeDetectionWriteBack

```
function EdgeWithDicomWB(myFileDir)
% EdgeWithDicomWB Function reading DICOM files from given folder (myFileDir)
% Developed by BS
% April 2009
%

%Cleanup of environment
close all;

% Split inbound parameter into logical array
InBound=regexp(myFileDir,'\','split');

% Get correct StudyUID from to use later on
StudyUID=char(InBound(2));

% Reading files in folder into variable
files=dir(myFileDir);

%Read Metadata from first DicomFile to hold my Dicom tags
info = dicominfo(strcat(myFileDir,'\',files(3).name));
% Create a new DICOM SerieUID
serieuid = dicomuid;
% Put this into my info objects
info.SeriesInstanceUID=serieuid;
% Custom series description
info.SeriesDescription='Created by Matlab';

%Create directory for new serie, name = new serieuid
NewSeriesFolder=serieuid;
workdir=cd;
cd(myFileDir);
cd('.');
mkdir(NewSeriesFolder);
cd (workdir);

% Looping the files, starting to iterate at 3 because of . and ..
i=3;

% Writing out new Dicom Files
while (i <= length(files))
    %Reading original image
    oldImage=dicomread(strcat(myFileDir,'\',files(i).name));
    %Creating a new "edge"enhanced file
    newImage=edge(oldImage);
```

```
%Getting new uid for this image
imageuid = dicomuid;
info.ImageInstanceUID=imageuid;
%Move to correct folder
cd(myFileDir);
cd('.');
FullFileName=strcat(NewSeriesFolder,'\,imageuid,.dcm');
% The actual creation of a new DICOM file
dicomwrite(newImage,FullFileName,info,'CreateMode', 'copy');
cd(workdir);
i=i+1;
end

%Need to create file to initiate postback
CStoreReqFile=strcat('c:\MatlabFilter\OutBoundQueue\',serieuid, '.DicJob');
CStoreReq = fopen(CStoreReqFile,'w');
fprintf(CStoreReq, date);
fprintf(CStoreReq, ' ');
fprintf(CStoreReq, Time);
fprintf(CStoreReq, '|');
fprintf(CStoreReq, 'Post this serie back to store');
fprintf(CStoreReq, '|');
fprintf(CStoreReq, StudyUID);
fprintf(CStoreReq, '|');
fprintf(CStoreReq, serieuid);
fclose(CStoreReq);

close all;
return;
```

9.8.2. MatlabFilter 2: SineScrollDemo

```
function SineScroll(myFileDir)
% SineScroll Function displays DICOM images sequentially
% from given myFileDir path
%
%Environment Cleanup;
close all;

%Reading files into variable
fil=dir(myFileDir);
% Iterating the variable, displaying images
% starts at 3 because of . og ..

i=3;
while (i < length(fil))
    % Getting file with folder (Fullpath)
    FullName=strcat(myFileDir,'\',fil(i).name);
    % Displaying the image
    imshow(FullName,'DisplayRange',[])

    % Setting timedelay between images
    if (length(fil)>10)
        pause(0.1);
    else
        pause(5);
    end
    i=i+1;
end
close all;
end
```


9.8.3. MatlabFilter 3: MR_VolumeRender

```
function MyVolFilterSerie(myFileDir)
%Reads MR slices, and generates a 3D

%clear all;
close all;
%Read directory into var
fil=dir(myFileDir);

% starts at 3 because of . og ..

i=3;

%Read files into array
while (i < length(fil))
    FullName=strcat(myFileDir,'\',fil(i).name);
    [dicomVolume(:, :, i+1)]= dicomread(FullName);

    i=i+1;

end

% squeeze MR info for size
Ds=squeeze(dicomVolume);

% Reads info for graph
image_num = 8;
image(Ds(:, :, image_num))
axis image

x = xlim;
y = ylim;

% Do a contour to give the object a nice apperance
contourslice(Ds,[],[],image_num)
axis ij
xlim(x)
ylim(y)
daspect([1,1,1])
colormap('default')

phandles = contourslice(Ds,[],[],[1,12,19,27],8);
view(3); axis tight
set(phandles,'LineWidth',2)

% interpolate to make this nice
Ds = smooth3(Ds);
```

```
hiso = patch(isosurface(Ds,5),...
            'FaceColor',[1,.75,.65],...
            'EdgeColor','none');

hcap = patch(isocaps(Ds,5),...
            'FaceColor','interp',...
            'EdgeColor','none');
%colormap(map)

view(45,30)
axis tight
daspect([1,1,.4])

% Show the volume
lightangle(305,30);
set(gcf,'Renderer','zbuffer'); lighting phong
isonormals(Ds,hiso)
set(hcap,'AmbientStrength',.6)
set(hiso,'SpecularColorReflectance',0,'SpecularExponent',50)
view(215,30);
%close all;
end
```

9.8.4. MatlabFilter 4: PossibleArtefactSerie

```
function ArtInSerie(myFileDir)
%MYEDGEFILTER Summary of this function goes here
% Detailed explanation goes here
%clear all;
close all;

fil=dir(myFileDir);

% starter på 3 pga . og ..

i=3;
while (i < length(fil))
    FullName=strcat(myFileDir,'\',fil(i).name);
    ArtInIm(FullName)
    pause(0.1);
    i=i+1;
end
close all;
end
```

9.8.5. MatlabFilter 5: MyEdgeFilterDemo

```
function newImage = MyEdgeFilter(myFileName)
%MYEDGEFILTER Summary of this function goes here
% Detailed explanation goes here
%clear all;
close all;

oldImage=dicomread(myFileName);
newImage=edge(oldImage);
imshow(oldImage,'DisplayRange',[]);
pause(5);
close all;
imshow(newImage,'DisplayRange',[]);
pause(5);
close all;
end
```

9.8.6. MatlabFilter 6: PossibleArtefact

```
function ArtInIm(myFileName)

LogFile = fopen('c:\MatlabFilter\Result.txt','a');
logtime=Time;
logCR='\r';

% Random token generator
rand_nr = rand;

% Inserting probability for the different results
if (rand_nr < 0.89),
    result = 'Not Found';
elseif (rand_nr >= 0.89 && rand_nr < 0.95),
    result = 'Found';
elseif (rand_nr >= 0.95),
    result = 'Could not say';
end

logtext=' Result from image ';
fprintf(LogFile, date);
fprintf(LogFile, logtime);
fprintf(LogFile, ' ');
fprintf(LogFile, logtext);
fprintf(LogFile, myFileName);
fprintf(LogFile, ' ');
fprintf(LogFile, result);
fprintf(LogFile,logCR);

fclose(LogFile);
end
```

9.8.7. Helperfunksjon for logging: Time

```
function Time=Time()
logtime=fix(clock);

loghr=num2str(logtime(4));
if length(loghr)==1
loghr=strcat('0',loghr);
end

logmin=num2str(logtime(5));
if length(logmin)==1
logmin=strcat('0',logmin);
end

logsec=num2str(logtime(6));
if length(logsec)==1
logsec=strcat('0',logsec);
end
Time=strcat(loghr,':',logmin,':',logsec);

end
```

9.9. Plakat til demonstrasjon

Automatisk deteksjon av artefakter og funn i DICOM bilder ved å bruke egne filter - prototype basert på webtjenester

Børje Stangeland

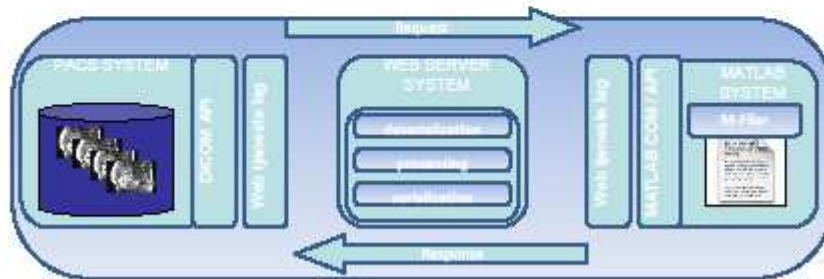
BAKGRUNN

I Norge økte antall undersøkelser fra 641 pr tusen Innbyggere i 1963 til 710 pr tusen Innbyggere i 1993 og til 780 pr tusen Innbyggere i 2003. Denne sterke veksten ser ut til å fortsette. Ved siste års RSNA i USA (RSNA 2004) ble det lagt fram beregninger på at røntgen vil øke med over 200 % i de neste 10 årene. utfordringen er etter hvert å kunne lete frem ønskede utvalg av bilder i forbindelse med forskning eller kvalitets sikring på en enkel måte.

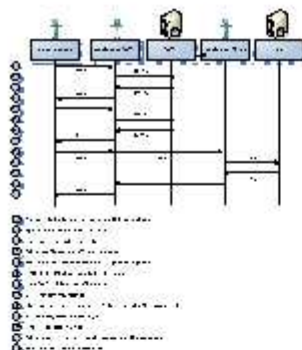
PROBLEMSTILLING

Dette prosjektet forsøker å lage en integrasjonsplattform som ikke er fundert i eksisterende løsninger og implementeringer, men som eksponerer kommunikasjonstjenester Innen DICOM i en mer tjenesteorientert arkitektur.

Prosjektet har også som mål å undersøke om det er mulig å løsrive DICOM kommunikasjon fra sin tradisjonelle punkt-til-punkt konfigurasjon inn i en mer tjenesteorientert arkitektur med de egenskapene det medbringer.



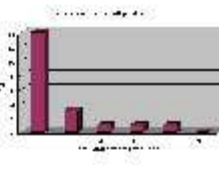
Figur 1: Overordnet struktur i løsningen.



Figur 2: Meldingsflyt intern i løsningen.



Figur 3: Responsider for henting av bilder, EPIFI og MYAPP.



Figur 4: Responsider for innføring av Matlab funksjon over webtjenester.



Figur 5: Eksempler på bilder før og etter Matlab-filtrering. Til venstre ett MR-akt uten og med kantfilter. Til høyre en serie MR-bilder filtrert og satt sammen til en 3D-model.

LØSNING og KONKLUSJON

Webtjenestegrensesnitt og tilhørende prototype på applikasjon viser at det er mulig å eksponere basis DICOM tjenester som webtjenester. Den overordnede struktur er illustrert i figur 1. Figur 2 illustrerer prosessflyt i applikasjonen. Det er ingen prinsipielle hindringer for å utvide dette grensesnittet til også å eksponere andre DICOM tjenester.

Måleresultatene viser at tidstillegget som webtjenesten og grensesnittet yter er marginalt og neglisjerbart for brukere. Disse resultatene vises i figur 3.

Det demonstreres et webtjenestegrensesnitt mot COM grensesnittet til Matlab 2007b. Figur 4 viser registrert tidsforsinkelse mellom opprettet Matlab jobb og den reelle iverksettingen av jobben over COM grensesnittet. Det er viktig å merke seg at det bør skapes en wrapptjeneste på serveren som har webmotoren til å holde kommunikasjonen mot Matlab oppe. Responsidene vil være for dårlige om webtjenestene skal instansiere grensesnittet hver gang. Eksempel på Matlab-filtrerte bilder vises i figur 5.

Ut av resultatene er det også mulig å skape fungerende applikasjoner og tjenester ved å knytte disse teknologiene sammen. Webtjenester og tjenesteorientert arkitektur er et stort område, men det kan i dette prosjektet ikke identifiseres noen prinsipielle grunner for å ekskludere spørremeldinger eller abonnement på meldinger i samsplitet mellom DICOM og tjenesteorientert arkitektur.

Summen av disse resultatene belyser at det på sikt kan være mulig å løsrive seg fra den konvensjonelle kommunikasjonsplattformen som DICOM har i dag, til å tilby mer tjenesteorienterte løsninger uten å måtte omdefinere en hel industri.