

Teknisk beskrivelse af Fælles Servicecenter Prototype

Formål	2
Indledning	3
FHIR SERVICE og FHIR DB	4
FHIR MODEL	4
Patient	5
Organization	5
Questionnaire	5
QuestionnaireResponse	6
CarePlan	6
Task	6
ReferralRequest	7
DeviceRequest	7
SupplyRequest	8
Condition	8
Overblik	8
PLAN SERVICE	10
OPGAVE SERVICE	12
PATIENT SERVICE	13
PLAN UI	13
Oprettelse af ny plan	14
Aktivitetsliste	15
Planstatus	17
Understøttelser af UseCases	20
Fælles service center som en del af RM TUD	22
Gemme og dele data jf. regionale og nationale retningslinjer	22
Login	23
Service sikkerhed	23
Behandlerrelation, samtykke og brugsret	23
Continuous deployment	24
Drift og skalering	24
Konklusion og anbefalinger	25
Konklusioner	26
Anbefalinger	26

Formål

Formålet med dette dokument er at dokumentere det tekniske setup i forhold til FSC prototypen.

Dokumentet beskriver følgende emner

- Implementerede services
- Implementerede model
- Brug af RM-TUD(Test, Udvikling og Driftsmiljø)
- Konklusioner og anbefalinger

Hovedformålet med FSC prototypen er at give en dybere indsigt i, hvilke krav der til et system, der skal understøtte FSC fremtidige arbejdsgange.

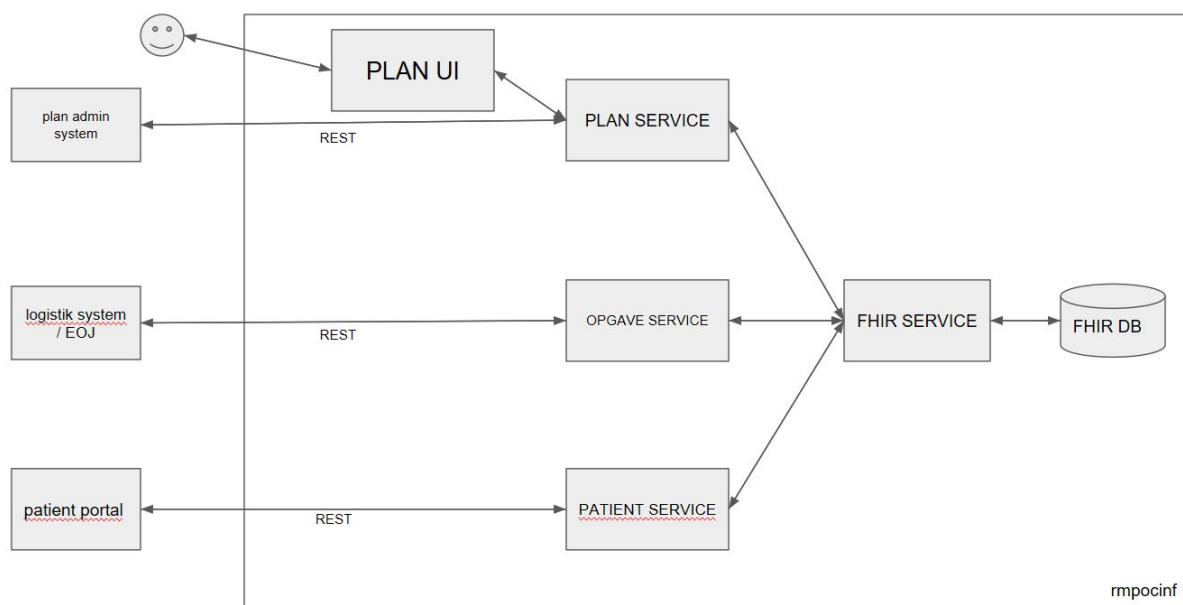
Indledning

Følgende er en teknisk beskrivelse af den prototype, der er udviklet (baseret på dokumentet “Beskrivelse af bestillingsflow og implementationsforslag”) og deployet på <http://rmpocinf.kvalitetsit.dk/fsc/> -miljøet (herefter omtalt som rmpocinf-miljøet eller RM-TUD).

Prototypen består af brugergrænseflader og REST-interfaces til at oprette, ændre og vise FHIR-ressourcer i et bestillings-flow (og opfølgning på denne bestilling).

FSC prototypen funktionalitet er designet som microservices. Microservices gør det lettere at implementere, drifte og automatisere processer. Det gør også man kan have flere leverandører der kan bidrage til det samlede system.

De forskellige services som prototypen består af, har deres kildekode i github <https://github.com/KvalitetsIT>. Samtlige services er pakket som docker images og herefter deployet på et kubernetes cluster på rmpocinf-miljøet. Følgende er en overordnet skitse af prototypen:



Komponenter inden i “rmpocinf” kassen illustrerer de implementerede komponenter, hvorimod komponenterne udenfor forestiller eksterne systemer, eller brugergrænseflader som ikke er implementeret i prototypen. Der er lavet REST-services som de eksterne systemer kan benytte. I demo-flowet kan der køres igennem POSTMAN eller lignende værktøj, der kan sammensætte GET og POST http-requests.

En kort gennemgang af de forskellige komponenter:

- FHIR DB: En mysql database der benyttes af FHIR service til at gemme FHIR ressourcer og relationer mellem disse.
- FHIR SERVICE: Service til læsning, skrivning og opdatering af FHIR-ressourcer
- PLAN SERVICE: Service der stiller håndtag til rådighed m.h.t. konfigurering af spørgeskema, servicekatalog, udrulning af en plan, aktiviteter på en patient og opfølgning på planen.
- OPGAVE SERVICE: Service der stiller håndtag til rådighed for at en kommune eller privat aktør kan hente en liste med opgaver og skifte status på disse
- PATIENT SERVICE: Service der stiller håndtag til rådighed for, at en patient kan se sine måleregimer
- PLAN UI: Brugergænseflade til besvarelse af spørgeskema, udrulning af plejeplan med aktiviteter, visning af status af brugerens plejeplan og aktiviteter under denne.

FHIR SERVICE og FHIR DB

Servicen er i dette projektet konfigureret til at håndtere FHIR resource-typerne:

- *Patient*
- *Questionnaire*
- *QuestionnaireResponse*
- *Careplan*
- *Task*
- *ReferralRequest*
- *DeviceRequest*
- *SupplyRequest*
- *Organization*
- *Condition*

Se nærmere beskrivelse her: <http://hl7.org/fhir/resourcelist.html>.

Servicen stiller ubegrænset skrive/læse-adgang til alle FHIR-ressourcer til rådighed.

Servicen er derfor ikke ikke udstillet på internettet, men tilgås via de tre hjælpe-services (plan, opgave patient). Servicen er desuden beskyttet v.h.a. OIO REST WS-standarden, så kun de tre services kan kalde den.

Kildekoden til dette projekt findes under <https://github.com/KvalitetsIT/fhirpatientservice> og bygger docker-imaget [kvalitetsit/fhirservice](https://github.com/KvalitetsIT/fhirservice) . Komponenten er en indpakning af open source projektet HAPI (<http://hapifhir.io/>) som er en implementering af persistens-lag, REST-interface til skrivning / læsning / søgning af FHIR-ressourcer og data-transfer objekter til samtlige FHIR resource-typer. Prototypen er baseret på HAPI version 2.5 og DSTU3.

FHIR MODEL

FHIR-modellen består groft sagt af en masse FHIR ressourcer, der hver har (som minimum) en resource-type og et auto-genereret ID - ressourcerne peger på hinanden v.h.a. en Reference, som er en streng bestående af ressourceypen efterfulgt af ID - f.eks. er

referencen "Patient/43" en reference til patient-ressourcen med ID 43. Referencen kan i et mere udviklede setup være en sti til den server, hvor denne ressource findes (så man f.eks. har én server til Patienter / stamkort, én til organizations-hierarkiet o.s.v.) men i prototypen benytter vi kun een server og d.v.s. altid lokale referencer.

Der vil i det følgende blive gennemgået hvilke FHIR-ressourcer der er i spil og hvorledes de refererer til hinanden:

Patient

Felt	Værdi
identificer	CPR-nummer

Patienter gemmes i prototypen blot som tomme ressourcer, med en identificer som er patientens CPR-nummer. Det er muligt (og dette lægger FHIR også op til) at gemme alle patientens stamdata (navn, adresse, etc...) her - eller mere oplagt (for ikke at indføre et nyt sted hvor stamdata er duplikeret) lade alle patient-referencer i modellen være referencer til et allerede eksisterende system der håndterer patienter og stamdata.

Det vil også være muligt i FHIR modellen at benytte et OID i stedet for et CPR-nummer som identificer.

Organization

Felt	Værdi
identificer	CVR-nummer

Organisationer gemmes i prototypen blot som tomme ressourcer, med en identificer som er organisationens CVR-nummer. Det vil også være muligt i FHIR modellen at benytte et andet ID i stedet for CVR det kunne være SOR ID, EAN nummer eller lignende.

Organisationer anvendes i prototypen som dem, der skal udføre en konkret opgave (ReferralRequest, SupplyRequest) for patienten. En organisation kan f.eks. være en kommune eller en privat aktør. Det er muligt i modellen at gemme en masse af organisationens stamdata (navn, type, adresse, etc...). Mere oplagt (for ikke at indføre et nyt sted hvor organisationer er duplikeret) er det muligt at lade alle organisation-referencer i modellen være referencer, til et allerede eksisterende system der håndterer organisationer.

Questionnaire

Felt	Værdi
Title	Titel på spørgeskema, vises i

	brugergænsefladen.
Item	De enkelte spørgsmål i spørgeskemaet, p.t. understøttes typerne CHOICE og STRING.

Prototypen understøtter p.t. kun et enkelt Questionnaire, som er det spørgeskema som udfyldes i "PLAN UI". En oplagt udvidelse er flere spørgeskemaer som ifølge FHIR-modellen kan knyttes op på UseContext, Code, eller en organisation (via ContactDetail).

QuestionnaireResponse

Felt	Værdi
Title	Titel på det spørgeskema der er besvaret
Subject	Reference til Patient
BasedOn	Reference til CarePlan
Questionnaire	Reference til Questionnaire
Item	Besvarede spørgsmål (svarende til items i Questionnaire)

QuestionnaireResponse svarer til en besvarelse af et spørgeskema for en given patient. Besvarelsen sker i forbindelse med udrulning af en careplan.

CarePlan

Felt	Værdi
Status	status, p.t. altid "Active"
Description	Titel på CarePlan
Subject	Reference til Patient
SupportingInfo	Reference til QuestionnaireResponse
Activity	En liste, hvor hvert element er en reference til en Task

En CarePlan udrulles på en patient, baseret på en spørgeskema-besvarelse, med en række aktiviteter.

Task

Felt	Værdi
For	Reference til Patient
Status	Starter i REQUESTED, vil skifte når aktøren (Organisationen) går i gang med opgaven, og senere afslutter denne
BasedOn	Reference til CarePlan
Focus	Reference til enten SupplyRequest , ReferralRequest eller DeviceRequest
Description	Beskrivelse af opgaven
Code	Kode på opgaven

Opgave svarende til en aktivitet i en CarePlan, som skal udføres på en patient. "Focus"-feltet bestemmer typen på aktiviteten, dvs. om det er noget der skal udføres af patienten selv (DeviceRequest) eller af en kommune/privat aktør (ReferralRequest, SupplyRequest)

ReferralRequest

Felt	Værdi
BasedOn	Reference til CarePlan
Subject	Reference til Patient
Recipient	Reference til Organization

En ReferralRequest er ifølge FHIR-specifikationen brugt når:

".....is seeking another practitioner or organization to use their own expertise and/or authority to determine the specific action to take....."

Det kunne f.eks. være en anmodning om et rygestopforløb på en patient (subject), men hvor det er op til den angivne organisation (Recipient) at bestemme det konkrete indhold.

DeviceRequest

Felt	Værdi
Subject	Reference til Patient
BasedOn	Reference til CarePlan

En DeviceRequest er en anmodning om, at en patient skal bruge noget måleudstyr (Device) til at foretage nogle målinger. En DeviceRequest kan opfattes som et måleregime. DeviceRequest omhandler ikke levering af udstyret, eller de observationer der kommer ud af målingerne - kun instruktioner i hvordan målingerne ønskes foretaget. Der er p.t. ikke arbejdet med, hvordan måleregimet præcist skal angives i et DeviceRequest, men ressourcen indeholder felter til angivelse af tid/interval for målinger (Occurrence), årsag til målinger (v.h.a. koder og referencer), mulighed for tekst/beskrivelse (Note).

SupplyRequest

Felt	Værdi
Supplier	Reference til Organization
DeliverTo	Reference til Patient

SupplyRequest drejer sig om levering af noget konkret (fysisk) til patienten (deliverTo), leveret af en organisation (supplier). En DeviceRequest kan være det måleapparat som patienten skal foretage målinger med (jvf. DeviceRequest).

Man kan via felterne orderedItem angive præcist, hvad der skal leveres enten som tekst eller som referencer til Device eller Medication (FHIR-ressourcer der ikke er med i prototypen).

Hvis man har et specielt ønske om hvor leveringen skal komme fra (altså forskelligt fra Supplier) kan dette angives i DeliverFrom, og en ønsket tidsplan for leverancen kan angives i Occurrence.

Condition

Felt	Værdi
Subject	Reference til Patient
Code	Beskrivelsen af Condition (Kode+tekst)

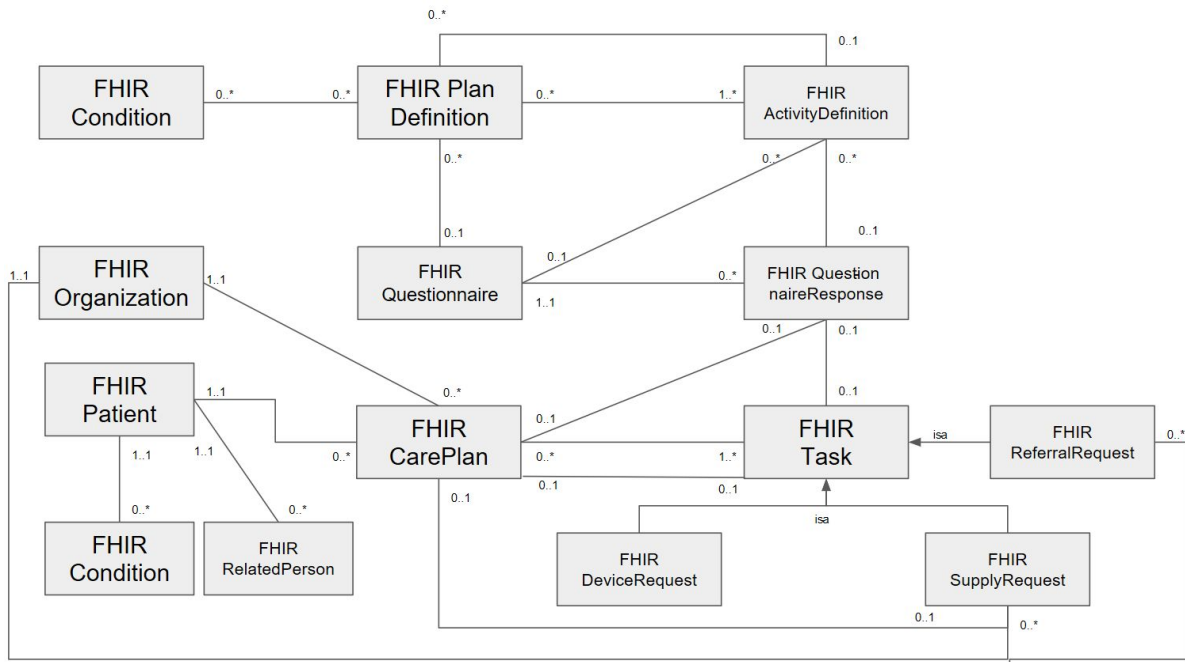
Condition er ifølge FHIR-standarden defineret som:

"...A clinical condition, problem, diagnosis, or other event, situation, issue, or clinical concept that has risen to a level of concern....."

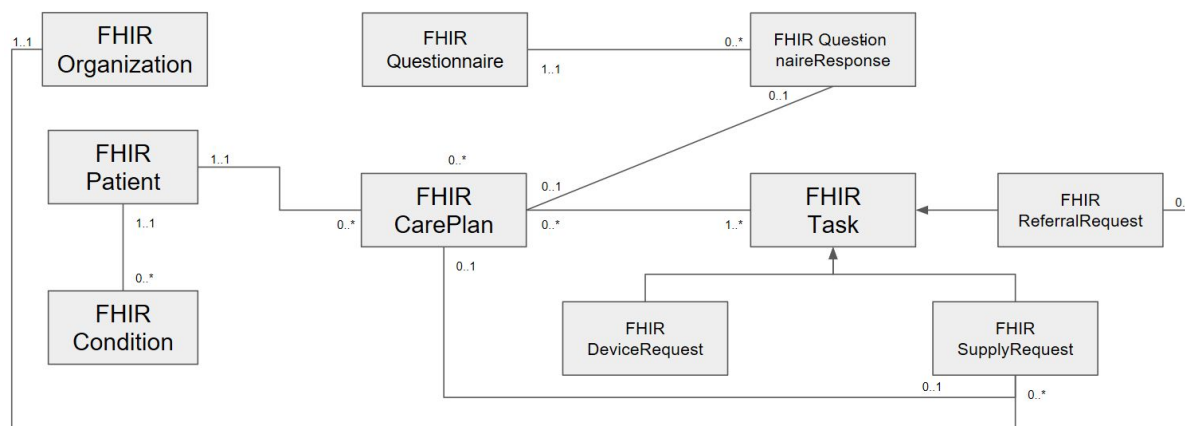
Der er lavet funktionalitet i prototypen til, at ved besvarelse af visse spørgsmål i spørgeskemaet (QuestionnaireResponse), at der oprette Conditions som tilknyttes patienten. Et eksempel på dette kunne være, at patienten svarer på, hvorvidt vedkommende er gangbesværet. Er svaret ja, bliver der oprettet en Condition med teksten "gangbesværet" tilknyttet patienten.

Overblik

Den oprindelige FHIR-model præsenteret i dokumentet "*Beskrivelse af bestillingsflow og implementationsforslag*" er som følger:



Den implementerede model er som følger:



Det som der er ikke er implementeret i prototypen er følgende:

- RelatedPerson (Pårørende) - dette er relateret til patientens stamdata, og derfor ude af scope som anden stamdata.
- Condition (i den oprindelige model fremkommer denne to steder - een gang tilknyttet Patient, og een gang tilknyttet PlanDefinition)- dette bliver udelukkende brugt som noget der bliver tilknyttet patienten, og bliver ikke brugt til at vælge en Plan Definition.
- PlanDefinition, ActivityDefinition og multiple "Questionnaire" (d.v.s. at det ikke altid er det samme spørgeskema man udfylder, men et forskelligt et afhængigt af brugskontekst) er ikke implementeret (der blev også nævnt i "*Beskrivelse af bestillingsflow og implementationsforslag*" at dette ikke blev en del af prototypen). Dette skulle på sigt være en mere avanceret form for "servicekatalog" som kunne afhænge af kommune (Organization), patientens tilstand (Condition). I prototypen er dette til dels erstattet af eet (konfigurerbart) skema og en CarePlan, hvor de mulige Tasks kan konfigureres.

PLAN SERVICE

Planservicen benyttes til konfigurering af spørgeskema, servicekatalog, udrulning af en plan, aktiviteter på en patient og opfølgning på planen.

Kildekoden til dette projekt findes under <https://github.com/KvalitetsIT/fsc-bestillingsservice> og bygger docker-imaget `kvalitetsit/fsc-bestillingsservice`. Imaget er deployet så servicen således det kan nås udefra på <http://rmpocinf.kvalitetsit.dk/fsc/planconfig/>

Servicen stiller nedenstående REST-håndtag til rådighed.

I følgende - og øvrige service-oversigter - vil første kolonne G/P angive om det er et POST- eller GET- request). GET-requests bruges primært til læse-operationer, hvorimod POST bruges til at opretter og opdatere.

P	/createCarePlan	<p>Input: JSON-struktur indeholdende:</p> <ul style="list-style-type: none"> • cpr - patientens cpr • allowReuse - boolean der indikerer hvorvidt aktiviteter fra tidligere careplan må genbruges • carePlan - FHIR CarePlan på JSON-format • questionnaireResponse - FHIR QuestionnaireResponse på JSON-format • activities - liste af koder på aktiviteter der skal udrulles <p>Handling: Vil knytte QuestionnaireResponse og CarePlan sammen og gemme disse i FHIR-databasen. De vil være knyttet op på patienten med det givne CPR. Hvis QuestionnaireResponse indeholder items med en kode der starter med "COND..." vil der blive oprettet Conditions på patienten svarende til disse. Baseret på indholdet af activities (koder) og serverens aktuelle konfiguration (se /getTaskConfiguration) vil der blive oprettet activities, tasks og requests. Hvis allowReuse er valgt, vil der ikke blive oprettet Tasks og Requests, hvis den aktuelle activity allerede er oprettet på en anden CarePlan.</p> <p>Denne metode anvendes af PLAN UI ved tryk på "Gem".</p>
G	/getConditions	<p>Parametre: cpr</p> <p>Handling: Vil hente alle FHIR Conditions på det givne cpr-nummer. Denne metode anvendes af planUI til at forudfylde felter i spørgeskema, hvis patienten allerede har en Condition med denne kode.</p> <p>Output: liste, hvor hver indgang indeholder en "condition" (FHIR Condition på JSON-format)</p>
G	/getCarePlans	<p>Parametre: cpr</p>

		<p>Handling: Henter alle CarePlan FHIR ressourcer på den givne patient. Bruges af PLAN UI's "plan status" side.</p> <p>Output: liste, hvor hver indgang indeholder en "careplan" (FHIR CarePlan på JSON-format)</p>
G	/getTasksForCarePlan	<p>Parametre: carePlanId</p> <p>Handling: Henter alle Tasks (med tilhørende SupplyRequests, DeviceRequests og ReferralRequests) tilhørende den givne carePlanId. Bruges af PLAN UI's "plan status" side.</p> <p>Output: Liste "taskAndReferralRequests", hvor hver indgang indeholder en "task" (FHIR Task på JSON-format) og en referralRequest (FHIR ReferralRequest på JSON-format) - derudover lignende lister "taskAndSupplyRequests" og "taskAndDeviceRequests".</p>
G	/getTaskConfiguration	<p>Parametre: ingen</p> <p>Handling: Henter servicens aktuelle konfiguration m.h.t. hvilke aktiviteter der er til rådighed, og hvilke forskellige requests disse fører til.</p> <p>Eksempel-output:</p> <pre> { "activities": [{ "code": "ACT002", "name": "Activity 2", "requests": [{ "type": "SUPPLY", "code": "SUP001", "name": "request2", "cvr": "123" }] }], } </pre> <p>Dette betyder, at der er mulighed for at udrulle en aktivitet med navn "Activity 2", kode "ACT002", som vil oprette en SupplyRequest med Kode "SUP001", navn "request2", som skal håndteres af Organization med cvr 123</p>
P	/updateTaskConfigurations	<p>Handling: Dette skal bruges til at opdatere servicens aktuelle konfiguration m.h.t. hvilke aktiviteter der er til rådighed, og hvilke forskellige requests disse fører til.</p> <p>Input: Form-data indeholdende feltet "file" med en fil indeholdende JSON på samme format som getTaskConfiguration leverer som output.</p>

G	/getQuestionnaire	<p>Parametre: Ingen</p> <p>Handling: Henter det ene FHIR Questionnaire objekt, der bruges af PLAN UI.</p>
P	/updateQuestionnaire	<p>Handling: Dette skal bruges til at opdatere det ene FHIR Questionnaire objekt, der bruges af PLAN UI.</p> <p>Input: Form-data indeholdende feltet "file" med en fil indeholdende JSON på samme format som getQuestionnaire leverer som output.</p>

OPGAVE SERVICE

Opgave servicen gør det muligt for, at en region, kommune eller privat aktør kan hente en liste med opgaver og skifte status på disse.

Kildekoden til dette projekt findes under <https://github.com/KvalitetsIT/fsc-opgaveservice> og bygger docker-imaget `kvalitetsit/fsc-opgaveservice`. Imaget er deployet således, at det kan nås udefra på <http://rmpocinf.kvalitetsit.dk/fsc/opgave/>

G	/getReferralRequests	<p>Parametre: cvr</p> <p>Handling: Vil hente alle "ledige" referralrequests (d.v.s. i status accepted,ready,received,requested jvf http://hl7.org/fhir/task.html#statemachine) på det givne cvr-nummer</p> <p>Output: liste, hvor hver indgang indeholder en "task" (FHIR Task på JSON-format) og den tilhørende "request" (FHIR ReferralRequest på JSON-format)</p>
G	/getSupplyRequests	<p>Parametre: cvr</p> <p>Handling: Vil hente alle "ledige" SupplyRequests (d.v.s. i status accepted, ready, received, requested jvf. http://hl7.org/fhir/task.html#statemachine) på det givne cvr-nummer</p> <p>Output: Liste, hvor hver indgang indeholder en "task" (FHIR Task på JSON-format) og den tilhørende "request" (FHIR SupplyRequest på JSON-format)</p>
P	/updateStatus	<p>Handling: Vil opdatere status på en given task, og evt. tilknytte en kommentar. De mulige status'er er: draft, requested, received accepted, rejected, ready, cancelled, in-progress, on-hold, failed, completed, entered-in-error (jvf. http://hl7.org/fhir/task.html#statemachine)</p> <p>Eksempel-input:</p> <pre>{ "taskId" : "1", "status" : "completed", "text" : "gennemført uden problemer" }</pre>

	}
--	---

PATIENT SERVICE

Patient servicen stiller håndtag til rådighed for, til fremsøgning af en patients måleregimer.

Kildekoden til dette projekt findes under <https://github.com/KvalitetsIT/fsc-deviceuseservice> og bygger docker-imaget kvalitetsit/fsc-deviceuseservice. Servicen er deployet så det kan nås på <http://rmpocinf.kvalitetsit.dk/fsc/patient/>

G	/getDeviceRequests	<p>Parametre: cpr</p> <p>Handling: Vil hente alle devicerequests på det givne cpr-nummer</p> <p>Output: liste, hvor hver indgang indeholder en "task" (FHIR Task på JSON-format) og den tilhørende "request" (FHIR DeviceRequest på JSON-format)</p>
---	--------------------	---

PLAN UI

PLAN UI stiller Brugergrænseflade rådighed for besvarelse af spørgeskema, udrulning af plejeplan med aktiviteter, visning af status af brugerens plejeplan og aktiviteter under denne.

Formålet med PLAN UI er at vise og afprøve indsatsen for at implementere en dynamisk brugergrænseflader, der genereres ud fra FHIR ressourcer. Dette gælder især generering af brugergrænseflade baseret på FHIR *Questionnaire*. En sådan brugergrænseflade vil skulle benyttes i overgangsordninger, hvor integration ikke er sat op mellem brugerens "eget system" (EOJ, EPJ...) og FSC.

Kildekoden til dette projekt findes under <https://github.com/KvalitetsIT/fsc-questionnaireservice> og bygger docker-imaget kvalitetsit/fsc-questionnaireservice-webgui. Servicen er tilgængelig på <http://rmpocinf.kvalitetsit.dk/fsc/plan/>

Bruger og patient-kontekst er uden for scope for denne prototype. Alle sider er derfor uden sikkerhed / login. Patientkonteksten er hardkodet til CPR-nummer (2512489996). Senere afsnit beskriver hvordan sikkerhed/login vil kunne håndteres.

Plan UI består af to brugergrænseflade:

/questionnaire	Brugergrænseflade til at oprette en ny plan for en patient/borger. En plan består af et spørgeskema og en række aktiviteter.
/planstatus	Brugergrænseflade der gør det muligt at se aktiviteterne(Tasks) for en bestemt plan.

Oprettelse af ny plan

PlanUI bygger automatisk en brugergrænseflade ud fra en Questionnaire og en aktivitetsliste.

Prototypen indeholder en første version af et værktøj der automatisk genererer en brugergrænseflade ud fra en Questionnaire. Et Questionnaire er et spørgeskema der består af en række Items der definerer felt-typerne (tal, tekst, selectbox osv).

I prototypen er følgende elementer understøttet:

- Choice
- Text

Alle felter af typen "Choice" vil blive genereret som "select" bokse, dvs felter hvor der indgår en række valgmuligheder. I elementer af typen "choice" indgår der en "reference" som peger på et ValueSet der ligeledes er defineret i Json definitionen af Questionnaire. Felter med typen "Text" er tekstfelter.

Nedenstående er et eksempel på et FHIR Questionnaire i Json format.

```
{
  "linkId": "1.4",
  "code": [
    {
      "system": "http://loinc.org",
      "code": "9267-6"
    }
  ],
  "type": "choice",
  "options": {
    "reference": "#Udstyr"
  }
},
{
  "linkId": "2.1",
  "text": "Fysisk tilstand",
  "type": "string"
},
```

Brugergrænsefladen vil blive genereret ud fra ovenstående definition.

Nedenstående viser et eksempel på bestillings-spørgeskema genereret ud fra et FHIR Questionnaire.

Bestilling af Telemedicin - KOL

Udfyld oplysninger om patienten

Rygning
1-5 pr. dag ▼

Aktivitetsniveau
Mellem ▼

Egenmestring
Mellem ▼

Udstyr
Afhentes/udleveres ▼

Fysisk tilstand

Psykisk tilstand

Andet

Det er muligt at ændre det FHIR questionnaire der benyttes i prototypen via `updateQuestionnaire` - se *Plan Service*.

Brugergrænsefladen bygger på AngularJS og AngularJS Material. AngularJS¹ er et Open Source Javascript projekt. AngularJS Material er en reference implementation af Googles Material-Design specifikation².

Hvis der på sigt ønskes en alternativ UI teknologi, kan der laves en ny implementation af *AngularJSQuestionnaireBuilder* sammen med enkelte ændringer i JSP siden.

Aktivitetsliste

I brugergrænsefladen til oprettelse af en ny CarePlan, er det muligt at oprette en række aktiviteter. Aktiviteterne vil være de handlinger der indgår som en del af bestillingen. Brugergrænsefladen til oprettelse af aktiviteter er dynamisk. Aktiviteterne der udgør listen hentes fra *Plan service* via `/getTaskConfiguration` (listen kan ændres via `/updateTaskConfigurations`). Brugergrænsefladen oprettes ud fra denne liste.

¹ <https://angularjs.org/>

² <https://material.io/guidelines/material-design/introduction.html#>

Aktiviteter

Vælg aktiviteter

- Måling af blodtryk - morgen og aften
- Blodtryksapparat
- Måling af lungekapacitet - dagligt
- Spirometer
- Vægt
- Rygestopkursus
- IT-kursus

Tillad genbrug af eksisterende aktiviteter fra eksisterende planer

Den nederste checkboks gør det muligt at undgå oprettelse af allerede eksisterende aktiviteter. Har patienten eksempelvis allerede har aktiviteten "foretag blodtryksmåling" i forbindelse med en anden plan, så vil denne aktivitet ikke blive oprettet igen (hvis nederste checkbox er markeret). Brugeren vil (ved oprettelsen på planen) få vist en liste hvis der bliver genbrugt aktiviteter fra tidligere planer.

Planstatus

Planstatus gør det muligt at få et simpelt overblik over aktiviteterne i en plan. Øverst vælges den konkrete plan. Nederste vises aktiviteterne opdelt efter: Måleregime, Henvisning og Udstyr.

Status på behandlingsplaner

Vælg behandlingsplan ▼

Aktiviteter i behandlingsplan

Måleregime

Henvisning (Referral)

Udstyr

Måleregime(DeviceRequest) viser hvilke målinger der skal tages, hvilke sundhedsrelaterede spørgsmål der er relevante og hvor ofte de aftalte målinger skal tages. Som beskrevet tidligere er der ikke oprettet konkrete måleregimer som en del af prototypen, men det understøttet i modellen og brugergrænsefladen.

Henvisning(RefferalRequest) viser aktiviteter som skal udføres som en del af planen. Det kan eksempelvis være et rygestopkursus der håndteres af kommunen.

Udstyr(SupplyRequest) viser det udstyr som patienten er tildelt/skal tildeles, det kunne være en vægt, blodtryksmåler eller lignende. Dette inkluderer leverancen af konkret udstyr.

Til hver aktivitet vises en tilhørende status. I nedenstående eksempel er alle statusser sat til "received".

Aktiviteter i behandlingsplan

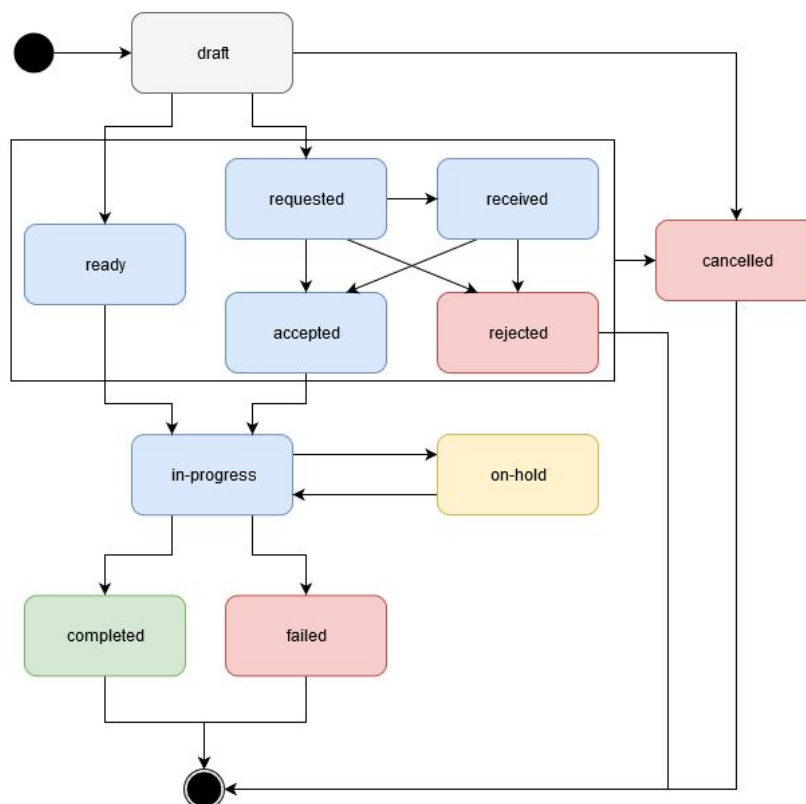
Måler regime

Måling af blodtryk - morgen og aften
received

Måling af lungekapacitet - dagligt
received

Vægt
received

Der benyttes standard FHIR statusser³ i prototypen. Nedenstående tegning viser oversigten over de statusser der som standard er moduleret i FHIR. Status for aktiviteter kan ændres via Opgave Service (/updateStatus).



FHIR lægger op til, at ovenstående statusser ikke nødvendigvis retvisende for alle arbejdsgange. Det vil således være muligt stadigvæk at være compliant med FHIR selvom

³ <https://www.hl7.org/fhir/task.html#statemachine>

ikke alle statusskift benyttes. Det vil desuden være muligt, selv at tilføje andre tilstande/statusser. Det kan overvejes om tilstandene skal "profileres" til dansk praksis.

Understøttelser af UseCases

Der vil i dette afsnit blive set på, hvilke dele af Prototypen der kan understøtte de usecases, der er beskrevet i dokumentet "Usecases" (version 1.12)

1. **UC1 - Administrer serviceydelser**

Dette er understøttet v.h.a. REST-kaldene "getTaskConfiguration" og "updateTaskConfiguration" hvor det er muligt at få vist (og derefter opdatere), de aktiviteter som der udbydes. Det er ikke muligt at angive pris i prototypen.

2. **UC2 - Rekvirer til telesundhed**

Denne usecases er til dels understøttet af brugergrænsefladen i "PLAN UI" (siden "questionnaire" som tillader udfyldelse af et spørgeskema og valg af nogle serviceydelser). Login og patientkontekst (d.v.s. hvilken sundhedsfaglig bruger er logget ind, og efterfølgende valg af patient) ligger udenfor prototypen, ligeledes er hentning/visning af patientens stamdata og hentning af diagnose.

3. **UC3 - Revider behandlingsplan**

4. **UC4 - Vis sundhedsfagliges behandlingsplan**

Disse 2 usecases er stort set samme brugergrænseflade, hvor hovedforskel er logon / patientvalg, hvilket er ude af scope for prototypen. Disse usecases kan understøttes af brugergrænsefladen i "PLAN UI" (siden "planstatus" som viser alle careplans på en patient og status på bekræftede tasks). Derudover er der lavet REST-håndtag "getDeviceRequests" på "PATIENT SERVICE", som henter en liste over patientens måleregimer.

5. **UC5 - Vis/løs kommunale opgaver**

6. **UC6 - Vis/løs regionale opgaver**

7. **UC7 - Vis privat aktørs opgaver**

Disse 3 usecases er basalt det samme - hente en liste over opgaver (ReferralRequests / SupplyRequests) tilknyttet en Organization (i prototypen regnes der bare med, at Organisation er defineret ved CVR-nummer, men kunne også være kommune / SOR / regionskode) - og derefter skifte status på disse opgaver når de udføres. Dette understøttes af REST-håndtag "getReferralRequests" "getSupplyRequests" og "updateStatus" i Opgave Service

Der var tidligt i projektet diskussion om hvorvidt eventuelle usecases m.h.t. afregning var relevante, men interviews med slutbrugere førte til konklusionen at dette var uden for scope. Der er heller ikke implementeret noget i prototypen m.h.t. afregning. FHIR-standarden tilbyder ikke direkte nogen form for "pris/afregning"-felter på nogen som helst af de

involverede ressourcer i prototypen. FHIR har en hel “Financial”-kategori af ressourcer:

	Support	Billing	Payment	General
Financial	<ul style="list-style-type: none"> Coverage 2 EligibilityRequest 2 EligibilityResponse 2 EnrollmentRequest 0 EnrollmentResponse 0 	<ul style="list-style-type: none"> Claim 2 ClaimResponse 2 	<ul style="list-style-type: none"> PaymentNotice 2 PaymentReconciliation 2 	<ul style="list-style-type: none"> ExplanationOfBenefit 2 Contract 1 Account 2 ChargeItem 0

En stor del af ressourcerne modellerer forskellige sundhedsforsikringer f.eks. afgørelse af hvorvidt sundhedsforsikringer dækker forskellige behandlinger, og hvem der skal betale for ydelserne. Ikke alle dele af modellen vil være relevant i en dansk kontekst. Desuden er FHIR’s egen vurdering af modenheden (Maturity Rating) på disse ressourcer ikke så høj.

Hvis man skal se på noget relevant for denne prototype, kunne det være noget funktionalitet, som ved afslutning (d.v.s statusskift til “completed”) på en SupplyRequest/ReferralRequest, oprettede et “ChargeItem” resource (baseret på koden af SupplyRequest/ReferralRequest). En “ChargeItem” kan forstås som en del af datagrundlaget for en fakturering, med felter til at angive patienten, diverse koder, hvem der har udført aktiviteten, hvem der har bedt om det o.s.v.

Fælles service center som en del af RM TUD

Dette afsnit beskriver hvordan Regions Midts TUD (Test, Udvikling og Drifts) miljø kan benyttes som platform for den funktionalitet der skal implementeres i FSC.

For en nærmere forståelse af koncepterne bag TUD refereres til evalueringsrapporten for [RMPOCINF].

Services der er udviklet som en del af FSC prototypen er deployet som en del af TUD. TUD er en skalerbar driftsplatform, der stiller en række standardservices til rådighed, som kan hjælpe med bl.a. dele, gemme og konvertere data og desuden leve op til diverse sikkerhedsstandard i forhold til f.eks. 2-faktor login og service sikkerhed(OIO IDWS).

TUD er et økosystem der muliggøre et reel flerleverandørstrategi, hvor forskellige leverandører er ansvarlige for at udvikler forskellige services på TUD. Det kunne også være de services FSC skal benytte.

TUD stiller i dag en række generelle services til rådighed, i forhold til

- Deling af data
- Login
- Samtykke håndtering
- Sikkerhed
- Leverance automatisering
- Drift og skalering

Ved at tænke services til Fælles Service Center ind i kontekst af TUD, vil umiddelbart give en række fordele i forhold til:

- Gemme og dele data jf. regionale og nationale retningslinjer
- Login er givet som en standard komponent og vil ikke skulle implementeres i FSC. Desuden vil samtykke og eventuelle enforcement points (tjek af samtykke) kunne håndteres uden for funktionaliteten der implementeres i FSC.
- Service sikkerhed håndteres af TUD.
- Behandlerrelation og brugsret.
- Nye releases og rettelser vil kunne lægges hurtigt og sikkert i drift via faste leverance pipelines
- Drift og skalering i forhold til antallet af brugere håndteres af TUD.

Følgende afsnit vil uddybe ovenstående punkter.

Gemme og dele data jf. regionale og nationale retningslinjer

I prototypen gemmes data i en FHIR database. FHIR er bl.a. valgt, da det er en operationel åben international standard. FHIR vil umiddelbart gøre det lettere at dele data med andre systemer, der kan "snakke" FHIR.

De nationale retningslinjer for datadeling definerer en model, hvor data skal gemmes i nationale/regionale XDS repositories som CDA dokumenter. Det gør, at data kan deles via disse centrale repositories.

Fokus for prototypen til FSC er ikke at lave integration med nationale/regionale XDS repositories. Det er KIT's anbefaling, at dette heller ikke laves i regi af FSC, da det er en opgave, der kan løses i det omkringliggende økosystem (TUD). En løsning kunne være at lave en service der løbende synkroniserer data og mapper FHIR data til CDA.

Login

Der er prototypen ikke etableret en login funktionalitet. Login for såvel borgere/patienter og klinikere bør bygge på en 2-faktor løsning. I TUD ligger der mekanismer til standardiseret at sikre login. Det gør, at alle løsninger der kører på TUD kan køre med samme høje sikkerhedsmodel. Det vil også gælde FSC. FSC vil således ikke skulle benytte ressourcer på at etablere egen login løsning.

I TUD er der etableret login, hvor der logges ind via standard IdP'er (identity provider) med et 2-faktor setup. Det understøttes bl.a. via en standard service til at SAML-enable applikationer. Læs eventuelt mere i [RMPOCINF]

Service sikkerhed

Prototypen til FSC benytter service sikkerhed der baserer sig på OIOXML IDWS REST standarden. Der benyttes kort sagt certifikater til at sikre services.

Service sikkerheden er ikke implementeret som en del af FSC-prototypen, men findes som standard funktionalitet i TUD-miljøet. Der behøves således ikke laves ny funktionalitet til overholde sikkerhedsstandarden.

Behandlerrelation, samtykke og brugsret

For at kunne styre adgang til applikationer og data er der tre overordnede områder der skal håndteres:

1. **Behandlerrelation.** Indikerer hvorvidt en sundhedsfaglig har patienten/borgeren i behandling. Hvis der eksisterer en behandlerrelation er adgang til udvalgte data lovlig (med mindre samtykker beskriver andet).
2. **Samtykke.** Patienten/borgerens samtykke beskriver, hvilke personer der har rettigheder til at se hvilke data. Det skal være muligt for at spærre/åbne (negativt/positivt samtykke) for dataadgang for udvalgte personer/afdelinger.
3. **Brugsret.** Brugsretten bestemmer, hvilke grupper der har lov til at benytte hvilke applikationer.

Alle tre områder skal håndhæves som en del af de services der udvikles til FSC. Hvis FSC kører som en del af TUD, vil det være oplagt at implementere behandler-relation og brugsret som en del af TUD (og nationale services). FSC vil således ikke skulle implementere egen løsning.

I forhold til samtykke vil håndhævingen kunne ske både i nationale services (XDS) men også som en del af TUD.

For at kunne håndtere Behandlerrelation og samtykke korrekt, vil det kræve, at en kontekst i form af et *Token* benyttes ved kald til regionale og nationale services. Dette kan også håndteres uden for FSC, hvis det køres i regi af TUD.

Continuous deployment

Continuous deployment dækker over en automatisk process, der kontinuerligt tager det nyeste skrevne software, og deployer denne til et miljø, og derved minimerer tiden fra koden skrives, til den kommer ud til brugerne. Nye releases og rettelser vil kunne lægges hurtigt og sikkert i drift, via faste automatiske leverance procedurer.

Fordelene ved at benytte Continuous deployment er, at slutbrugerne hurtigere får ny funktionalitet med høj kvalitet i produktion. Continuous deployment er ligeledes en del af TUD. Continuous deployment bliver brugt som en del af prototypen.

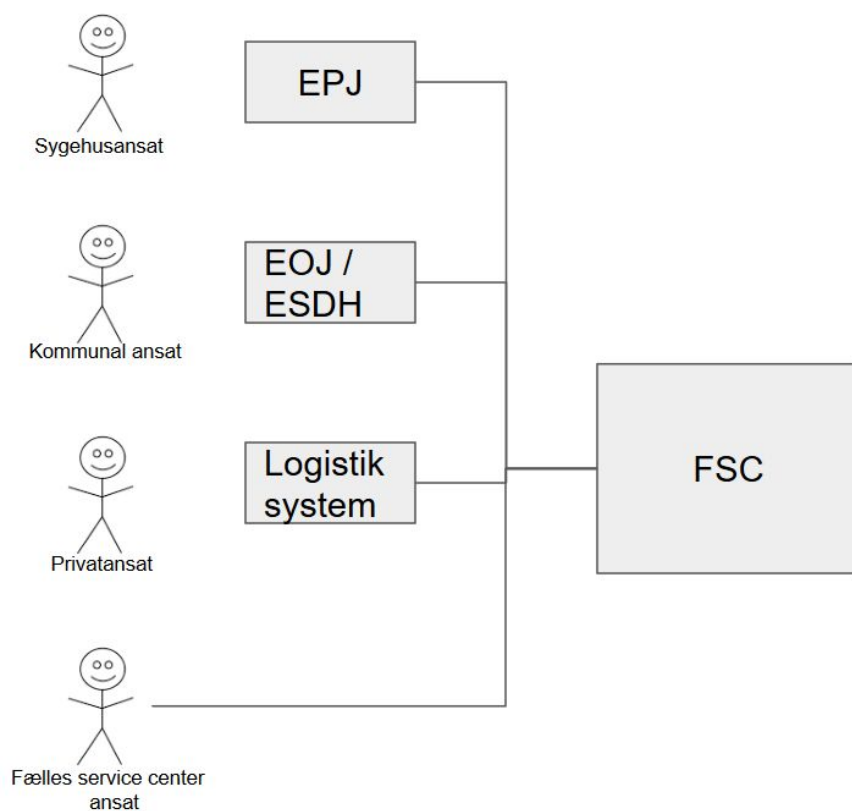
Drift og skalering

Prototypen benytter TUD til at drifte de services der er udviklet. Driftsmiljøet giver en skalerbar drift, det betyder, at services let kan skaleres ved replikere service-instaserne. Flere instanser vil også betyde en øget tilgængelighed.

Alle områderne er beskrevet nærmere i [RMPOCINF]

Konklusion og anbefalinger

Prototypen baserer sig på at kommune, sygehus og private aktører hovedsageligt skal benytte egne systemer til bestilling, opgave håndtering osv. FSC skal ikke være endnu et system, brugerne skal registrere og se data i. Systemet der skal benyttes af FSC skal kun indeholde de informationer, der skal deles mellem aktører. Systemet skal tilbyde åbne snitflader til at arbejde på disse oplysninger fra egne systemer. En forudsætning er, at der skal etableres integrationer fra en række systemer til Fælles Service Center.



Der skal være værktøjer (brugergrænseflader) til FSC medarbejdere der skal understøtte disses arbejdsopgaver. I prototypen er der ikke lagt op til at FSC skal indeholde samlede behandlingsplaner, blot de dele af behandlingsplanerne der er relevant for fælles service center. Det vurderes at den samlede behandlingsplan for patienten eksisterer i et andet system.

I forhold til prototypen er der arbejdet i 2 spor:

1. Definition og beskrivelse af usecases for aktører der har forbindelse med FSC.
2. Implementering af en konkret prototype

Definition og beskrivelse af usecases og hvilke konklusioner er beskrevet i et selvstændigt dokument [Usecases].

I prototypen er der arbejdet med følgende:

1. En datamodel der beskriver kerne usecases for bestilling. Modellen baserer sig udelukkende på FHIR ressourcer.
2. At benytte TUD økosystemet til nemt at drifte og udstille sikrede services.
3. Oprettelse af genererede brugergrænseflade ud fra FHIR ressourcer.

FHIR ressourcer benyttet af tre hovedgrunde:

1. For at undersøge om en åben standardiseret model kan understøtte FSC' usecases.
2. For at gøre det lettere at eksponere data standardiseret. Standardiseret data vil gøre det lettere at dele mellem forskellige systemer.
3. For at gøre det nemmere senere at mappe data til CDA og få dem gemt i et XDS.

TUD økosystem er benyttet for at vise, at det er nemt og hurtigt at få services sikret og i drift via regionens allerede etablerede økosystem. Det gør at prototypen til FSC kan fokusere på at implementere og understøtte konkrete usecases (funktionalitet).

Der er lavet brugergrænseflade for at vise, at det er let at generere dynamiske brugergrænseflader ud fra FHIR ressourcer. I den endelige version af systemet til FSC vil de fleste aktører arbejde i eget system, som integrerer til FSC. Det kan da være nødvendigt i en overgangsordning at etablere ekstra brugergrænseflader indtil alle integrationer er etableret.

Konklusioner

- FHIR modellen kan indeholde de oplysninger der er nødvendige for at understøtte de definerede usecases.
- FHIR modellen er operationel i selve data-delingen. Det giver således mening at eksponere FHIR direkte via service snitflader.
- RM TUD viser sig effektivt til at hoste og sikre FSC services. Det skyldes
 - a. Der skal ikke implementeres sikkerhed direkte i FSC. OIO IDWS REST og SAML Service Provider er allerede implementeret. Det kan gøre, at systemet til FSC kan fokusere på kernefunktionaliteten der skal understøtte de konkrete arbejdsgange.
 - b. TUD Driftmiljøet er allerede til rådighed. Det gør, at der ikke skal etableres nye driftmiljøer, databehandler aftale etc.
 - c. Standard services som konvertering til CDA (del af kravene for den nationale reference arkitektur), kan håndteres uden for systemet til TUD.

Anbefalinger

- Det anbefales at der laves et system til FSC som bygger på en åben standard, som f.eks. FHIR. De usecases der er implementeret i Prototypen kan understøttes af FHIR.
- Det skal afklares hvordan afregning skal implementeres. Afklaringen bør laves som et konkret afprøvningsprojekt. Det kunne være ved at benytte afregningsdelen af FHIR.

- Det skal afklares hvordan afregning håndteres organisatorisk mellem kommune, region og private aktører (på sigt måske også praksislæger). Dette er ikke en opgave der ligger i FSC alene, men i samarbejdsaftalerne.
- Services der udvikles som en del af FSC, bør tænkes ind i en regional og national kontekst. Dette kunne ske ved at brug af TUD og benytte de standardservices der eksisterer der. Her tænkes i første omgang især i forhold til: Drift, sikkerhed og datadeling.
- TUD kan på sigt bruges til at mappe FHIR data til CDA data og gemme i de nationale arkiver.
- TUD kan på sigt håndtere samtykke via nationale (eller regionale services).
- Det skal undersøges, hvilke brugergrænseflader der er behov for i slutversionen og hvilke der er nødvendige i en overgangsfase inden alle integrationer er etableret.