Digitale Forløbsplaner

KvalitetsIT

Version 1.0.1

1. Indholdsfortegnelse

2. Formål og læsevejledning

Denne rapport belyser hvorvidt det eksisterende FLP system kan understøtte forventningerne til de fremtidige behov, som systemet er i dag og kommer med en række anbefalinger til eventuelle forbedringer i forbindelse med at KiAP og PLSP har påbegyndt et samarbejde om en modning af FLP som helhed.

Rapporten er yderst teknisk og dokumenterer konkrete iagttagelser og anbefalinger i forhold til den tekniske arkitektur, således at, først og fremmest, PLSP og KiAP kan beslutte, hvorvidt iagttagelserne vurderes relevante og økonomisk forsvarlige og hvorvidt anbefalingerne skal følges.

Opgaven baserer sig på kommissoriet for konsolidering af forløbsplaner, som bla. opstiller en række forventninger til fremtidig anvendelse af FLP:

- 75000 patientforløb pr. år, fordelt på KOL og diabetes
- Løsningen anvendes også til indberetning til RKKP
- Ved udgangen af 2020 vil der være 3 forløbsplaner i drift samt indberetning til RKKP for KOL og diabetes type.
- Den tekniske løsning er en mulig kandidat til at blive anvendt til andre opgaver. Disse opgaver er imidlertid ikke nøjere beskrevet eller aftalt.
- Der er dog en forventning til at den tekniske løsning om 3-5 år er god nok til at kunne håndtere ca. 10 områder, der i kompleksitet svarer til de nuværende forløbsplaner og RKKP indberetningerne (3 forløbsplaner, 2 RKKP indberetninger, 5 nye områder).
- Udgangspunktet er, at et eventuelt nyt teknisk set-up ikke må være dyrere end den nuværende løsning.

Baggrunden for udarbejdelse af rapporten er "*PLSP og PL-forum har peget på et behov for at vurdere og sikre sig, at den tekniske løsning kan honorere de krav som følger af de nye aktiviteter og initiativer, der kan forankres i arkitekturen*", samt at PLSP og KiAP har påbegyndt arbejdet en række tiltag for at modne løsningen og samarbejdet såsom: "API-baseret arkitektur", "kobling mellem driftsmiljø og komponenter" og "Sikring af skalering, optimering og kvalitet"

Rapporten er lavet på baggrund af oplysninger og dokumentation fra:

- Medcom
- KiAP
- PLSP

(KiAP og PLSP kaldes herefter "parterne" som fællesbetegnelse fremover)

Der er gennemført en række workshops, videomøder og individuelle samtaler med de relevante parter samt gennemgang af eksisterende og ny dokumentation. Alle parter har været både åbne og hjælpsomme i processen.

Rapporten er inddelt i følgende overordnede punkter

- Konklusion
- Næste skridt
- As-is arkitektur
- To-be arkitektur

De to arkitektur afsnit og er opdelt i as-is og to-be arkitektur. Disse afsnit er meget tekniske. De beskriver henholdsvis den eksisterende løsning og anbefalinger/opmærksomhedspunkter i forhold til en kommende arkitektur.

Afsnittene i to-be arkitekturen indeholder lister med konkrete anbefalinger.

Konklusionen og næste skridt indeholder dels de overordnede konklusioner og dels en række konkrete forslag til opgaver, der kan sættes i gang.

3. Konklusion

FLP består basalt set af en række services, der gør det muligt for lægepraksis at oprette og arbejde med forløbsplaner sammen med patienter. En forløbsplan består af en mængde relevant patientdata, brugergrænseflader til registrering af yderliger oplysninger der ikke kan hentes direkte fra lægepraksis og støtte til lægen i forhold til automatisk beregninger (f.eks. GOLD status). Data synkroniseres fra lægepraksissystemerne så brugerne undgår dobbelt registrering. Forløbsplanerne kan deles med patienterne via sundhedsmappen.

Helt overordnet kan det konkluderes at FLP, som det er i dag, kan anvendes fremover til håndtering af nye typer forløbsplaner, jvf kommissoriet og som kort beskrevet i foregående afsnit. Dvs løsningen kan skalere som ønsket og er overordnet set er sikkerhedsmæssig tilfredsstillende.

Men FLP står, som beskrevet i kommissoriet, overfor en række nye anvendelser og møder samtidig nye krav f.eks i forbindelse med driftsansvar kombineret med GDPR. Derfor må rapporten også forholde sig til de potentielle ændringer og modninger som skal til for at kunne understøtte dette. Rapporten beskriver derfor en række anbefalinger, som bør vurderes af PLSP, KiAP og deres interessenter.

De konkrete, tekniske, anbefalinger kan ses under henholdsvis løsnings-, sikkerheds-, informations- og driftsarkitektur i to-be arkitekturen.

I forhold til PLSP og KiAP's interessenter kan anbefalingerne overordnet koges ned til følgende:

- Der bør etableres klare aftaler om adgang til data både i forhold til hvordan data teknisk tilgås men også roller, ansvar og opfølgning i forbindelse med brug af data. Dette gælder både for de nuværende anvendelser af data og kommende anvendelser.
  Konsekvensen af eventuelle beslutninger omkring dette kan betyde opgaveflytninger mellem parterne, et behov for omskrivning af løsningerne etc. Generelt er det netop her, at de to parter har størst påvirkning af deres eksisterende løsninger og på kort sigt har sværest ved at blive enige. Det er dog samtidig her at grunden til eventuelle yderligere, fremtidige, anvendelser af data lægges.
  - KiAP, PLSP og deres interessenter bør prøve at afklare de potentielle fremtidige anvendelser FLP kan se ind i, da det er med til at sætte krav til FLP's arkitektur. Det korte spørgsmål er: hvad skal data anvendes til?

        - Er der f.eks. behov for at få etableret en fælles data delingsplatform (a la NSP's datadelings services), sådeles af de delte data kan anvendes af alle (som har ret til det) til alle (ønskede og godkendte) formål? En sådan platform gør, at man kan etablere nye anvendelser af data uafhængigt af de indberettende systemer og at disse anvendelser kan etableres i et flerleverandør setup, da der kun er afhængigheder til datadelingsplatformen.
    - KiAP, PLSP og deres interessenter bør afklare rollerne mellem parterne. Dette gælder især i forhold til dataadgang, herunder hvem "ejer" databasen, hvem "ejer" snitfladerne og er det den samme som "ejer" f.eks. snitfladerne, som også udvikler den tilhørende software?
        - På den ene side er der behov for en høj grad af styring dvs. hvilke data der tilgås af hvem og på den anden side er der behov for fleksibilitet i forhold til at udvikle og opsætte nye forløb.
- Der bør arbejdes på en ensretning, mellem parterne, omkring leverancer, tests, drift og supportmuligheder for at skabe ens forventninger og tekniske krav til parternes systemer.
    - I praksis er anbefalingen først at etablere user stories for IT og leverancekrav, ligesom man i dag arbejder med user stories for de kliniske behov. På den måde får parterne og deres interessenter en fælles forventning og beskrivelse af løsningen.
    - Anbefalingerne omkring ensretning må vurderes enkeltvis. Nogle er relativt enkle og faktisk er KiAP og PLSP allerede gået i gang med at løse nogle af dem. Andre er mere komplekse. Parterne må vurdere, hvilke de vil adressere, baseret på deres behov og krav.
    - KiAP og PLSP bør udarbejde de relevante user stories og få dem godkendt af deres interessenter. Når dette er på plads er de fleste opgaver relateret til dette punkt af rent teknisk karakter.
- Der er desuden en række anbefalinger i relation til sikkerhed, auditlogning etc, som alle har mere teknisk karakter og disse vil ikke blive behandlet yderligere i konklusionen.
    - KiAP og PLSP kan, for de fleste af anbefalingerne, umiddelbart vurdere hvilke de prioriterer og gør noget ved.

Nedenfor er indsat links til en tabelvisning af de forskellige anbefalinger og opmærksomhedspunkter:

- 5.1.6. Løsningsarkitekur - Anbefalinger
- 5.2.4. Sikkerhedsarkitektur - Anbefalinger
- 5.3.3. Informationsarkitektur - Anbefalinger
- 5.4.7. Driftarkitektur - Anbefalinger

3.1 Næste skridt

Hvad er så de næste skridt for FLP for at omsætte de relevante dele af rapporten til konkret handling? I praksis er de næste skridt drevet af de af anbefalingerne, som parterne vurderer giver mening og værdi, så FLP kan håndtere de behov som kommissoriet beskriver.

Det er derfor op til KiAP og PLSP og deres interessenter at identificere anbefalinger som giver konkrete forbedringer. F.eks. ser det ud til, at der allerede nu er gang i at ensrette og samle logs på tværs af systemer.

Der er heller ingen tvivl om, at roller og tekniske løsninger omkring adgang til data, er den største udfordring fordi det har potentiale til både at flytte opgaver og kræve større ændringer af de eksisterende løsninger, hvis man ikke passer på. Den opgave kan kun løses parterne imellem, men vi håber, at de anbefalinger om arkitekturændringer, opdeling af services og afkobling af services, kan være en inspiration for begge parter i det videre arbejde. I hvert fald er anbefalingerne i tråd med hvordan nyere IT systemer designes f.eks. FUT.

Til trods for ovenstående forbehold, vil vi foreslå en række next steps for FLP:

1. Fremtidig (forventet) anvendelse af data afklares
    - KiAP og PLSP samt deres interessenter bør etablere en fælles forventning til dette, da dette sætter krav til hvilke data der skal være til stede, samt hvilke anvendelser som skal understøttes. Dette påvirker i høj grad arkitekturen.
2. Roller og ansvar afklares
    - Dette punkt ligger i forlængelser af punkt 1 og KiAP og PLSP bør få dette afklaret i samarbejde med deres interessenter. Det er her der i dag er størst uenighed
3. Nye user stories defineres for leverance og vedligehold, så krav og behov kan aftales
4. Der defineres en ny løsningsarkitektur, der afvejer governance af dataadgang og hurtig implementering af ny funktionalitet
5. Gennemgang af anbefalinger.

Punkt 1 og 2 bør igangsættes hurtigst muligt, da det er grundlaget for en fælles opfattelse af kravene til FLP's data og anvendelsen af disse og dermed en forudsætning for at kunne fastlægge arkitekturen og fordele roller og ansvar i den samlede FLP leverance. Det anbefales at afholde en workshop med KiAP, PLSP og deres interessenter for at afklare:

- Kommende anvendelser af data
- Forudsætninger for anvendelsen af data
- Afvejning af et behov for en høj grad af styring af driften og behovet for fleksibilitet i forhold til at udvikle og opsætte nye forløb.
- Workshoppen vil kunne definere en række opgaver som parterne skal komme med oplæg til samt en tidsplan som parterne kan blive enige om

Punkt 3 kan påbegyndes, så snart parterne prioritere det. Punkt 3 kan køre parallelt med punkt 1 og 2, omend dele først kan afsluttes når punkt 2 afsluttet. Anbefalingen er, at parterne får aftalt og beskrevet user stories af typen "som dataansvarlig for FLP, vil jeg kunne se en samlet auditlog, således at jeg kan dokumentere overfor LPS systemejeren at GDPR overholdes" og "som IT supporter vil kunne tilgå en læges IT system, således at jeg kan yde den ønskede support". Begge user stories er blot eksempler, men kan være med til at beskrive de behov FLP har og som i dag ikke er beskrevne, da fokus har været 100% på slutbrugerne. De udarbejdede user stories bør godkendes af de respektive interessenter.

Punkt 4 kan påbegyndes efter punkt 1-3 er afsluttede, da disse sætter mål og rammer for løsningsarkitekturen. PLSP og KiAP kan umiddelbart selv udarbejde denne samt eventuel økonomi i forbindelse med dette og få det godkendt af deres interessenter.

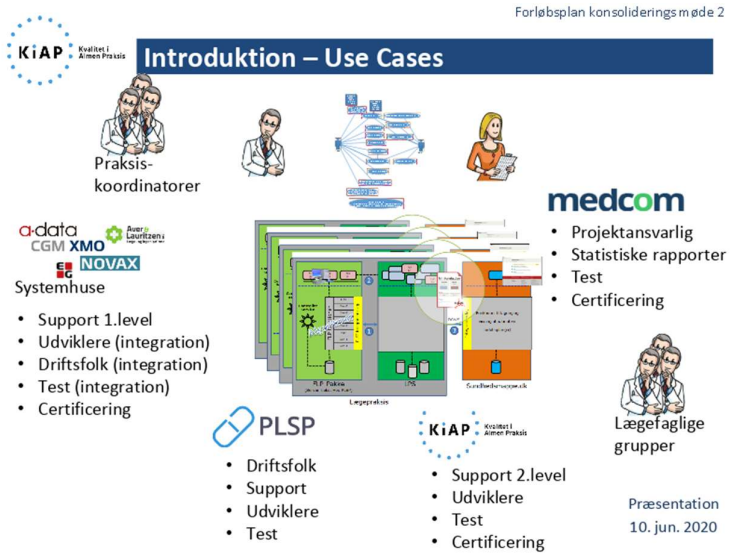
Punkt 5 kan PLSP og KiAP gennemføre når de kan prioritere det.

Til sidst vil KIT gerne sige tak til alle parter for et godt og konstruktivt samarbejde. Det har været en fornøjelse og vil gerne huske alle parter på, at FLP er et rigtig godt værktøj for slutbrugerne, ellers var det jo aldrig nået til hvor det er i dag.

4. As-is arkitektur

I denne del af rapporten beskrives as-is arkitekturen af Forløbsplaner.

På workshop 2 blev nedenstående tegning fremlagt. Her identificeres en række personaer, der supplerer de to brugertyper Læge og Patient, som der arbejdes med i forhold til projektbeskrivelsen.



Tegningen giver et overblik over rollerne for de involverede parter.

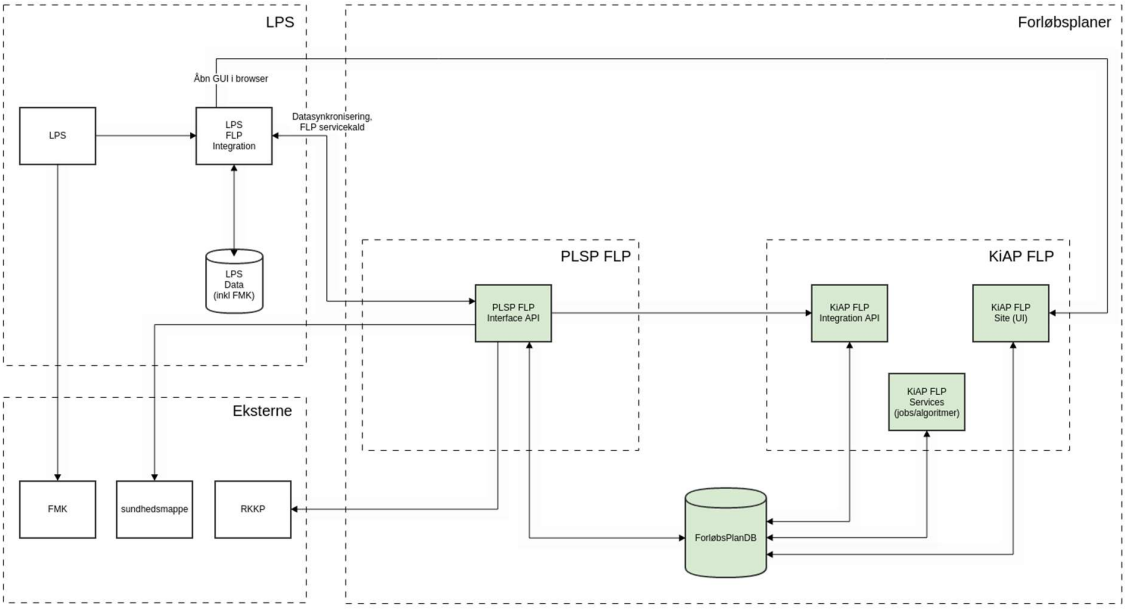
I de kommende afsnit gives et overblik over løsningsarkitekturen. Dette overblik har til formål at identificere de forskellige komponenter/services, som tilsammen udgør Forløbsplaner.

Med udgangspunkt i løsningsarkitekturen, vil rapporten fokussere på følgende:

- Sikkerhedsarkitektur: Hvorledes er interne/eksterne snitflader i løsningsarkitekturen sikret
- Informationsarkitektur: Hvor lever hvilke data? Hvilke komponenter har adgang til hvilke data? Hvilke komponenter er ansvarlige for vedligeholdelse af data (evt. gennem synkronisering)?
- Driftsarkitektur: Hvorledes driftes de forskellige dele af løsningsarkitekturen?

4.1. Løsningsarkitektur

Nedenstående tegning giver et overblik over de enkelte komponenter/services, der tilsammen udgør Forløbsplaner.



Komponenterne er i ovenstående tegning er delt ud på forskellige domæner:

- LPS (Lægepraksissystmer)
- Eksterne systemer
- Forløbsplaner
    - PLSP
    - KiAP

I dette arkitekturoverblik afspejler domæneinddelingen en logisk gruppering af komponenter/services ud fra et ansvars- og opgavemæssigt synspunkt. I det følgende gennemgås og motiveres de enkelte domæner og de tilhørende services/komponenter.

4.1.1. Eksterne systemer

Rundt om forløbsplaner findes en række eksterne systemer. Disse udgøres både af leverandører af data til forløbsplaner samt aftagere af data fra Forløbsplaner.

4.1.1.1. FMK

FMK indeholder masterdata i forhold til en patients medicinoplysninger. Forløbsplaner indeholder forretningslogik, der baserer sig på en patients medicinoplysninger. Hvis en forløbsplan giver anledning til ændringer i en patients medicinering, foretages ændringerne i medicinering af sundhedfagligt personale udenfor Forløbsplaner vha LPS.

4.1.1.2. Sundhedsmappe

Sundhedsmappe er patientens egen visning af sine forløbsplaner. Ved at tilgå https://forloebsplaner.sundhedsmappe.dk og logge ind med NemId, kan patienten se sin nuværende (replikerede) plan. Synkronisering af forløbsplanen til Sundhedsmappe sker efter

patientens ønske. Der foregår en replikering af data fra Forløbsplaner til Sundhedsmappe. Hvis der efterfølgende sker opdateringer af patientens data i Forløbsplaner, vil disse ændringer blive skubbet over til Sundhedsmappe. Efterfølgende mindre (minor) opdateringer af patientens plan replikeres uden videre over til Sundhedsmappe, mens der ved større (major) opdateringer først replikeres efter eksplicit godkendelse af patientens læge. Selve integrationen/datareplikeringen til Sundhedsmappe foretages af komponenten PLSP FLP Interface API.

4.1.1.3. RKKP

RKKP (regionernes kliniske kvalitetsudviklingsprogram) indeholder lovpligtige indberetninger. Forløbsplaner indberetter data til RKKP. Det er klinikken selv der uploader data til RKKP, dette gøres fra klinikkens udbakke, via en PLSP service. Indberetninger til RKKP sker i dedikerede skærmbilleder til RKKP indberetning i komponenten KIAP FLP Site (UI) og efterfølgende kald til PLSP FLP Interface API med formål at lægge RKKP indberetningen i en udbakke til senere leverance. PLSP FLP Interface API laver den faktiske integration til RKKP på baggrund af de godkendte indberetninger i udbakke.

4.1.2. LPS: Lægepraksissystemdomænet

Udgøres af den samlede mængde af lægepraksissystemer samt disses integrationer til Forløbsplaner. De enkelte leverandører af lægepraksissystemer har til opgave at integrere Forløbsplaner i lægepraksissystemets brugergrænseflade og workflow. Derudover har LPS til opgave at uploade/synkronisere data fra LPS databaserne til Forløbsplaner (databasen). Det konkrete datasæt, der skal uploades, er specificeret af Forløbsplaner og dækker laboratorieværdier, ydelser, diagnoser og medicin.

Deruodver har LPS domænet til ansvar at implementere integration mellem Forløbsplaner og FMK, således at aktuelle medicinoplysninger kan anvendes af Forløbsplaner. Da lægeprakisssystemerne i forvejen har implementeret integration til FMK foregår integrationen mellem FMK og Forløbsplaner i LPS domænet. De enkelte lægepraksissystemer i LPS domænet er ansvarlige for at synkronisere en patients medicinoplysninger med FMK inden oprettelsen af en forløbsplan for en given patient. Resten af Forløbsplaner er afhængig af, at denne opdatering foregår rettidigt. Synkroniseringen af FMK data fra LPS til FLP testes som en del af certificeringsprocessen.

4.1.3. PLSP FLP domænet

PLSP FLP (forløbsplaner) domænet udgør ansvarsmæssigt en forlænget og centraliseret serviceplatform for LPS domænet.

Alle servicekald til Forløbsplaner fra LPS går via PLSP FLP igennem kald til komponenten *PLSP FLP Interface API*. Denne komponent har til ansvar at stå for synkronisering af data med LPS domænet. Komponenten står også for at implementere sikkerhed og foretage passende auditloging i forhold til adgang til data. Som udgangspunkt kan LPS domænet kun oprette/redigere i data og ikke læse data fra Forløbsplaner. Dog er der mulighed for at lægen kan opdatere visse data (laboratorieværdier) i Forløbsplaner via brugergrænseflade. Disse opdateringer kan efterfølgende læses af LPS og derved synkroniseres tilbage i LPS domænet. Værdier fra LPS afvises, hvis de ikke overholder udfaldsrummet for koden, de er

registreret under. Afviste data logges. Enkelte værdier lukkes igennem selvom de ikke overholder ud udfaldsrummet, disse konverteres efterfølgende (se *KiAP FLP Services - jobs/algoritmer)*

Derudover fungerer PLSP domænet som integrationspunkt mellem Forløbsplaner og eksterne systemer, som aftager forløbsplandata. Replikering til eksterne systemer sker på foranledning af klinikken, der opretter og godkender afsendelserne. PLSP FLP henter derefter afsendelserne i dedikerede ”udbakker” og foretager den tekniske integration.

4.1.4. KiAP FLP domænet

KiAP - FLP (forløbsplaner) domænet udgøres af services og komponenter udviklet af KiAP.

KiAP FLP Integration API implementerer forretningsregler i forhold til forløbsplaner.

KiAP FLP Site (UI) tilbyder brugergrænseflader og formularer til det sundhedsfaglige personale til at understøtte arbejdet med forløbsplaner samt overblik over sårbare patientgrupper med henblik på opfølgning og forbedring af kvaliteten. Selve formularerne til de forskellige forløbsplaner er designet og kodet individuelt. Hvis en forløbsplan opdateres til at omfatte en ny patientværdi, så kræver det en opdatering og efterfølgende leverance af *KiAP FLP Site (UI)* med nye felter i brugergrænsefladen. Derudover tilbyder komponenten en formular til RKKP indberetning. Lægen skal ind i hver patients RKKP formular én gang årligt for at aktivere overførslen til RKKP. Det er aftalt at lægen fra sin RKKP patientliste skal kunne vælge at autoudfylde sine indberetninger ud fra allerede eksisterende data. Autoudfyldning kan kun foretages for patienter der opfylder en række krav. For patienter, der ikke opfylder disse, skal lægen fortsat udfylde dem manuelt, da der i disse tilfælde enten er mangelfulde data, eller de eksisterende data er for gamle jf. de krav der stillet af RKKP.

Udover at tilbyde brugergrænseflader API’et beskrevet ovenfor, så sker der også en række data-transformationer fra LPS domænets rådata til patientens forløbsplan. *KiAP FLP Site (UI)* foretager f.eks. følgende beregninger:

- BMI ud fra nyeste værdier for vægt og højde.
- GOLD Score, som f.eks. anvendes til at vurdere, hvordan patientens medicinering skal være.
- FEV1 /FVC udfra lungestørrelsen og luftvejenes åbningsgrad.

KiAP FLP Services (jobs/algoritmer) kører periodisk og varetager følgende beregninger, validering og konvertering:

- Enkelte typer af værdier kan komme igennem valideringen, selvom de ikke overholder udfaldsrummet (PLSP service). Disse data konverteres automatisk. Det drejer sig om værdier der kommer fra laboratorierne og kan divergere f.eks. forhold til operatorer der ikke altid skrives på samme måde og at vigtig information skrives i “bemærkninger”. Konvertering foretages ved, at det oprindelige rådata sættes til “invalid” og nyt (konverteret) data oprettes. Fortolkningen foretages ud fra et ønske om at kunne bruge så meget af det rådata, der overføres fra LPS, som muligt. Følgende værdier laves der mønstergenkendelse/konvertering af:
    - DNK35302: Glomerulær filtrationshastighed (eGFR)
    - DNK35131: eGFR / 1,73m²(CKD-EPI) – LMV

- NPU19661: Albumin/kreatinin ratio
- *Urinprøver*. Der sker en transformation/konvertering af Urinprøver fra laboratorierne. Eksempelvis ved at "operatorer" tilrettes og tekst i bemærkningsfeltet fortolkes, så de rigtige flag/stausser kan sættes på værdien.

- Inklusionskriterier: Inklusionskriterier genberegnes, og der laves ekstra validering på, om de er overholdt. ForløbsPlaner sikrer sig, at der til enhver tid er evidens for, at en given patient er inkluderet i et givent forløb jf. det tilhørende inklusionskriterie. I tilfælde hvor en læge ved en fejl har registreret en diagnose eller andet der gør at en patient inkluderes i et forløb og derefter retter op på fejlen (sletter denne oplysning), er det vigtigt at patienten så ikke længere opfattes som inkluderet i det pågældende forløb. Dette sikres ved at lade denne del af forretningslogikken køre i ForløbsPlan Services.
- Beregner om noget skal sendes til sundhedsmappe eller RKKP. Eller om data skal slettes på sundhedsmappe.

Det er gjort meget ud af performance og for, at jobs skal afvikles så effektivt som muligt, så databaser ikke låses. Tilstand i databasen holder styr på, hvad der allerede er beregnet, så jobs kan nøjes med "delta-beregninger".

4.1.5. Ens regler for alle services

Services der udvikles af KiAP og PLSP implementeres i dag med forskellige teknologistakke og egen funktionalitet f.eks. i forhold til logning og dokumentation. Der ligger ikke et fælles regelsæt for hvornår en service er færdig og klar til drift.

4.2. Sikkerhedsarkitektur

Forløbsplaner har en række snitflader, som er eksponeret både eksternt og internt i Forløbsplaner. Dette afsnit analyserer, hvordan snitfladerne i Forløbsplaner er beskyttet. Vi starter med en kort beskrivelse af standarden Den Gode Webservice (DGWS) samt en præcisering af de sikkerhedsniveauer, der arbejdes med i denne standard, da denne standard anvendes flere steder i Forløbsplaner. Dernæst betragtes Forløbsplaners sikkerhedsarkitektur.

4.2.1. Den Gode Webservice (DGWS)

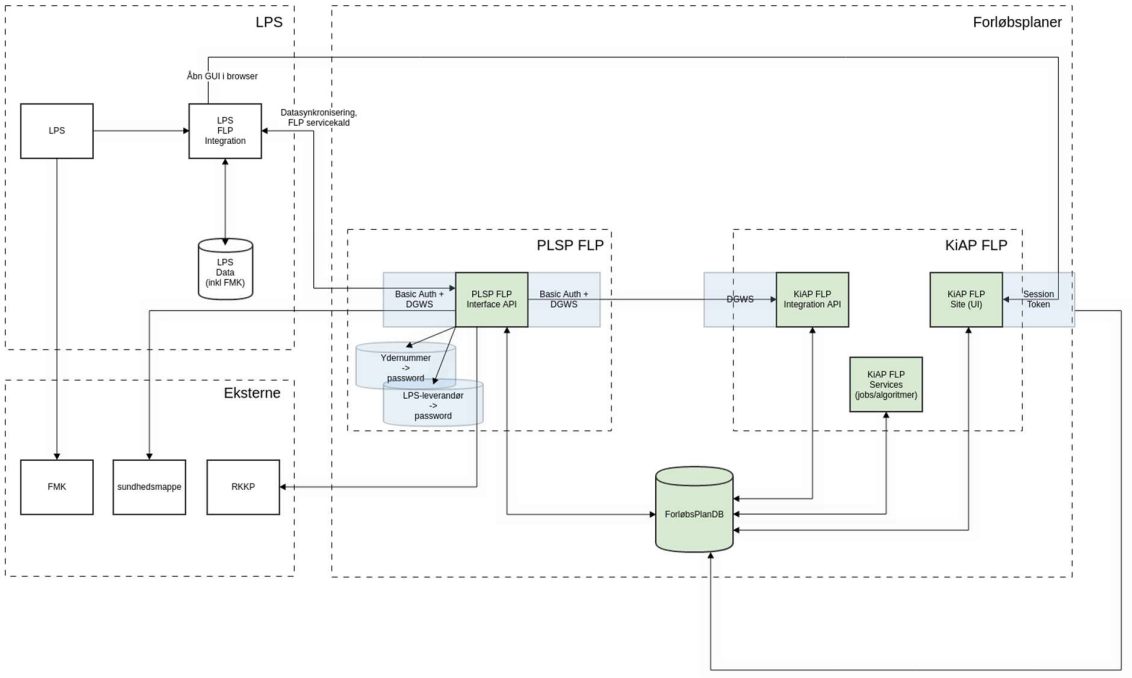
DGWS[1] er udviklet af Medcom for Sundhedsdatastyrelsen og anvendes bredt som en "fælles web-servicekuvert til brug for kommunikation mellem parterne i den danske sundhedssektor". DGWS som standard knytter sig til SOAP Webservices og specificerer, hvorledes akkreditivet (SOSI Idkort) indlejres som en header i en SOAP Envelope".

Akkreditivet SOSI Idkort kan tilvejebringes fra National Service Platform (NSP) Secure Token Service (SOSI-STS) på flere måder, men anvendelserne i dag baserer sig på certifikater:

[1] https://www.medcom.dk/standarder/webservice-standarder/den-gode-webservice

- VOCES/FOCES certifikat: Identificerer en organisation ved CVR nummer. Det resulterende SOSI Idkort har niveau 3. Der er således tale om en 'systembruger'.
- MOCES certifikat: Identificerer en (sundhedsfaglig) bruger ved deres CPR nummer samt brugerens tilknytning til en organisation identificeret ved CVR nummer. Det resulterende SOSI Idkort har niveau 4.

4.2.2. Snitfladerne i Forløbsplaner og sikkerhed



På ovenstående tegning er de snitflader, som er beskyttet af en eller anden form for sikkerhedsprotokol tegnet ind på komponentoverblikket. Detaljerne for sikkerhedsprotokollen for hver enkelt komponenter er beskrevet nedenfor.

4.2.2.1. PLSP FLP Interface API

Alle kald fra LPS domænet ind i Forløbsplaner domænet går gennem komponenten *PLSP FLP Interface API*.

Sikkerhedsprotokollen for servicekald til denne komponent er en blanding af:

- Den Gode Webservice (DGWS) samt
- Basic Authentication (brugernavn+password i en 'Logon' struktur som en del af serviceparametrene)

Organisationsidentifikatoren i DGWS er et CVR nummer, men i Forløbsplaner er organisationer/klinikker identificeret ved ydernummer.

PLSP FLP Interface API har løst dette problem ved at tilføje en 'Logon' struktur til alle requestdefinitioner i API'et. En 'Logon' struktur består af et brugernavn (et ydernummer) samt et password. Logon strukturen er således et slags 'klinik-login'.

Ved modtagelse af et kald vil *PLSP FLP Interface API* validere følende:

- SOSI IDkortet er gyldigt
- 'Logon' strukturen identificerer et gyldigt sæt af ydernummer + password

For at en praksis kan anvende Forløbsplaner, skal der derfor oprettes et 'klinik-login' i Forløbsplaner. Dette gøre i dag ved henvendelse til PLSP support. For at lette arbejdet er der også lavet en webservice på *PLSP FLP Interface API* til oprettelse af nye Forløbsplans logins (servicen *ItitializeService*). Her kan leverandører af LPS systemer oprette 'klinik-logins' for deres kunder. LPS systembrugerne identificerer sig vha brugernavn og password ('LPS-logins').

4.2.2.2. KiAP FLP Interface API

I forhold til Forløbsplaner er KiAP FLP Interface API en intern service, som kun har et anvendersystem, nemlig PLSP FLP Interface API.

Sikkerhedsprotokollen for denne komponent er:

- DGWS

Da *PLSP FLP Interface API* ved kald udefra modtager et SOSI Idkort anvendes dette i videre kald til *KiAP FLP Interface API*.

4.2.2.3. KiAP FLP Site (UI)

Denne komponent er en brugerrettet komponent og udbyder en brugergrænseflade. Der er derfor lavet en overførsel af sessionsid, når den sundhedsfaglige bruger åbner et konkret skærmbillede i sin browser. Sessionsid'et er indlejret i det link, som LPS henter ud via kald til *PLSP FLP Interface API*.

4.2.2.3.1. Sessionsoverførsel

KiAP FLP Site (UI) validerer det medsendte sessionsid ved opslag i ForløbsplanDB. Sessionsid'et anvendes til at etablere en session i KiAP FLP Site (UI) og er en engangnøgle, der fjernes fra databasen, når den er anvendt. Sessionsid'et er gyldigt i 8 timer.
Grundlaget til etableringen af SSO er niveau 4 SOSI Idkort.

4.3. Informationsarkitektur

Al data i Forløbsplaner lagres i den samme database, ForløbsPlanDB.

Dette betyder, at alle komponenter og services (både i PLSP og KiAP domænet) læser og skriver til denne fælles database.

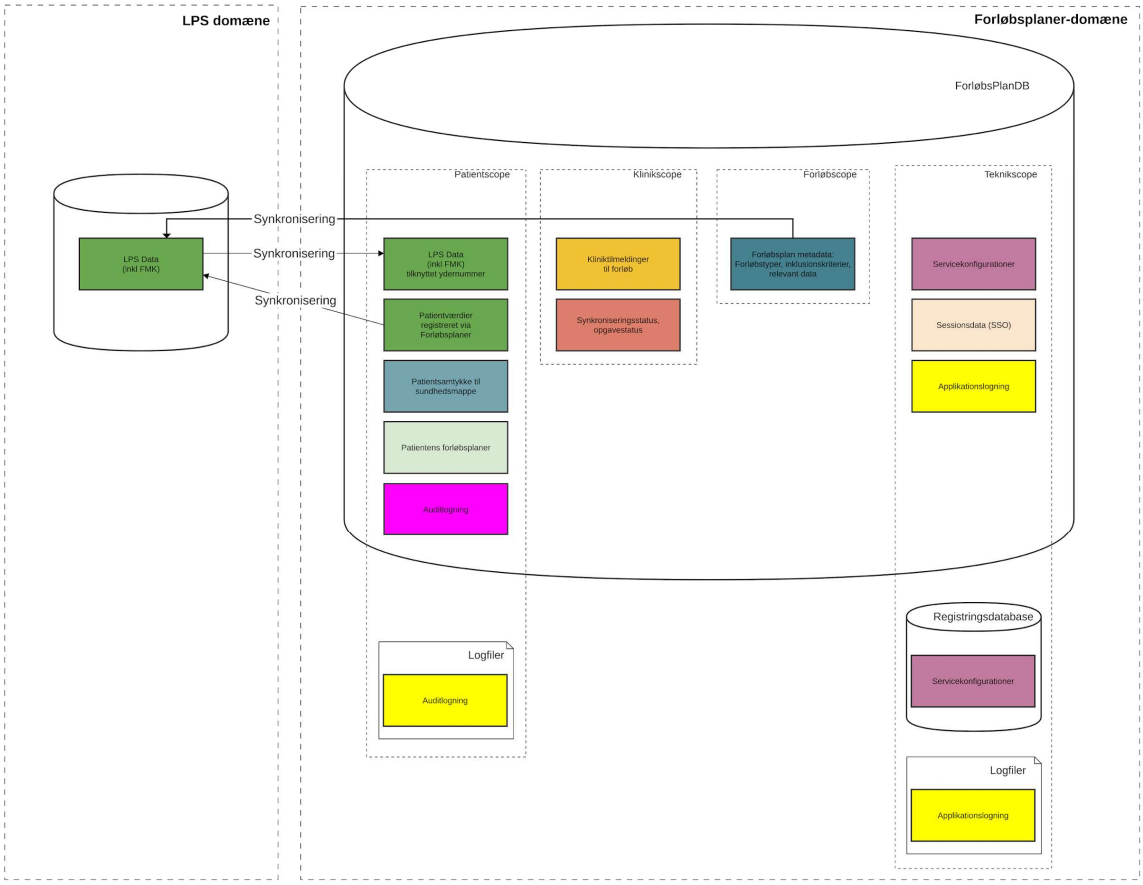
I forhold til Forløbsplaner giver det mening at analysere ForløbsPlanDB og identificere de forskellige dataområder (logisk gruppering af data), som der er i spil i Forløbsplaner, fordi forskellige domæner har forskellige krav f.eks. er dataansvar i forbindelse patientdata ikke det samme som for tekniske konfigurationer.

Nedenstående diagram giver et bud på en sådan gruppering. Grupperingerne er inddelt i kategorier, der beskriver, hvilket scope data har:

- Patientscope: Data for en specifik patient herunder auditlogning

- Klinikscope: Data for en specifik klinik (identificeret ved ydernummer)
- Forløbsscope: Data for specifikke forløbstyper
- Teknikscope: Diverse data, der skal understøtte integrationer og infrastruktur i Forløbsplaner herunder logning

En vigtig detaljer er, at patientdata vedligeholdes i (og ejes af) LPS domænet (dvs den enkelte klinik) og synkroniseres med Forløbsplaner-domænet. Dette er vist i diagrammet via synkroniseringspilene. Det er den enkelte klinik, som er dataansvarlig, og som via databehandleraftaler giver LPS, PLSP og KiAP ret til at behandle data for dem.



I de følgende afsnit kigger vi på hver enkelt dataområde i Forløbsplaner-domænet. Vi kommer med en beskrivelse af, hvad hver enkelt gruppering handler om. Dernæst nævnes de konkrete tabeller i ForløbsPlanDB, hvor data er persisteret.

Vi starter med det mest generelle scope i forhold til selve forløbsplanerne: Forløbsscope. Data i dette scope er overordnet modellering af forløbsplaner og relaterer sig ikke til specifikke klinikker og/eller patienter. Det er i dette scope, at definitionerne af forløbstyperne og deres indhold (indikation og relevante data). KiAP har den formelle governance i forhold til data.

I klinikscopet findes de dele af Forløbsplaner, der holder styr på relationerne mellem de forskellige forløbstyper og de konkrete klinikker. Data i dette scope relaterer sig til specifikke

klinikker (identificeret vha ydernummer). Data i dette scope tilhører klinikkerne (LPS domænet og PLSP som en central forlængelse heraf).

I patientscopet findes de konkrete patientdata - både rådata fra LPS systemerne og de data, der relaterer sig patientens konkrete forløbsplan(er). Data i dette scope relaterer sig til specifikke patienter og dennes tilknyttede klinik. Data i dette scope tilhører klinikkerne (LPS domænet og PLSP som en central forlængelse heraf). Patientdata er det mest følsomme data og opbevaring og adgang til data i dette scope er underlagt regler (GDPR, krav om auditlogning, etc).

I patientscopet findes desuden auditlogning. I Forløbsplaner i dag er der ikke fælles måder at auditlogge. Det gælder både måden, som auditloggen opbevares på:

- PLSP komponenter auditlogger til database
- KiAP komponenter auditlogger til filer

så vel som indholdet af auditloggen:

- Hvilke oplysninger identificerer brugeren.
    - KIAP: Ydernummer og initialer
    - PLSP: Ydernummer

Derudover findes ingen fælles retningslinjer for, hvornår der bør auditlogges.

Endelig findes teknikscopet. I dette scope findes diverse tekniske konfiguration af services f.eks. endpoints til eksterne services, logningskonfigurationer, applikationslogs.

4.3.1. Forløbsscope

4.3.1.1. Forløbsplan metadata

I Forløbsplaner opereres der med forløbstyperne f.eks. KOL, diabetes og hjerte.
Disse forløbstyper vedrører forskellige patientgrupper defineret vha inklusionskriterier.

Inklusionskriterierne er defineret som et søgeudtryk (udtrykt i SQL udtryk), som kan anvendes af LPS leverandørerne som dokumentation for at udvikle inklusionskriterierne i de konkrete LPS systemer. Hvert system hardcoder således, hvordan de patienter, der er relevante at inkludere i det konkrete forløb, udsøges.

Hvor inklusionskriterierne hjælper til at finde ud af *hvem*, der er relevante for et forløb, så indeholder Forløbsplaner også definitioner af *hvilke data*, der er relevante at overføre for at understøtte det konkrete forløb. Forløbsplan metadata er afbildet i ovenstående figur med med kassen *Forløbsplan metadata*.

4.3.1.2. Governance for forløb

MedCom uddelegerer opgaven med at prioritere nye forløbsplaner. Medcom nedsætter et lægefagligt udvalg, som specificerer krav til data, funktionalitet og evt. design med udgangspunkt i sundhedsfaglige krav. Outputtet fra det lægefaglige udvalg overdrages til relevante parter (f.eks. KiAP og PLSP).
Det er i as-is arkitekturen udokumenteret, hvordan data i dette scope vedligeholdes og hvem som står for det. Det antages at ændringer af eksisterende forløbstyper også sker via Lægefaglige Grupper.

4.3.1.3. Tabeller

I databasen dækker dette over følgende tabeller:

- Forloeb
- RelevanteDiagnoser
- RelevantMedicin
- RelevantePatientVaerdier
- RelevanteYdelser
- RelateredeTabeller

Forløbsplan metadata er således modellering af de konkrete forløbstyper, som er understøttet af Forløbsplaner og indeholder således hverken patient- eller yderspecifikke data.

4.3.2. Klinikscope

4.3.2.1. Kliniktilmeldinger

Det er op til den enkelte praksis (identificeret ved ydernummer) at tilmelde sig de ønskede forløbstyper. Deltagelse i forløbstyperne gemmes i Forløbsplaner og er repræsenteret i ovenstående figur ved den orange kasse *Kliniktilmeldinger til forløb*.

4.3.2.2. Tabeller

I ForløbsPlanDB er kliniktilmeldinger modelleret i følgende tabeller:

- PraksisTilmelding

4.3.3. Patientscope

4.3.3.1. Patient rådata

Masterdata for patientdata i Forløbsplaner befinder sig i LPS domænet. Det er i LPS domænet, at de sundhedsfaglige anvender deres respektive lægepraksissystemer til at arbejde med og vedligeholde data om patienten. Dette foregår dels gennem de datamodeller, flows og brugerflader, som de enkelte lægepraksissystemer implementerer og via integrationer til eksterne systemer f.eks. FMK. I ovenstående figur er patientdata repræsenteret i den grønne kasse *LPS Data (inkl FMK)* i LPS domænet.

Forløbsplaner indeholder funktionalitet til at synkronisere data fra LPS domænet ind i domænet for Forløbsplaner, hvor de lagres tilknyttet ydernummeret for den dataejende klinik. I Forløbsplaner er der mulighed for at lave registreringer af supplerende patientværdier via brugergrænsefladen. Disse værdier gemmes i Forløbsplaner-domænet og synkroniseres efterfølgende tilbage til LPS domænet. Det synkroniserede patientdata fra LPS domænet samt det i Forløbsplaner registrerede patientdata er repræsenteret ved de to grønne kasser *LPS data (inkl FMK) tilknyttet ydernummer* og *Patientværdier registreret i Forløbsplaner* i Forløbsplaner-domænet. I praksis gemmes både synkroniseret og inddateret data i de samme tabeller i databasen, men det giver mening at betragte disse to typer af data ud fra et informationsmæssig synspunkt.

4.3.3.1.1. Krav i forbindelse med adgang til patientdata

Fælles for data i patientscopet er, at det er er ejet af klinikkerne. Adgang til data skal ske gennem beskyttede snitflader med identifikation af den konkrete bruger (sundhedsfaglig, systemforvalter og support) og skal auditlogges.

Der findes ikke i Forløbsplaner i dag ensartede regler for adgang til patientdata og/eller en ensartet mådet at auditlogge.

4.3.3.1.2. Tabeller

De konkrete tabeller, der er dækket af de grønne *LPS data (inkl FMK) tilknyttet ydernummer* og *Patientværdier registreret i Forløbsplaner* Forløbsplaner-domænet er:

- Patient
- Diagnose
- Medicin
- Patientvaerdi
- Ydelse

Systemerne i LPS domænet har hver deres måde at repræsentere ovennævnte data. Systemerne indberetter via samme API. API'et gemmer data ensartet i FLP domænet. Tilknytning til dataejer (via ydernummer) sker i tabellen Patient. Den enkelte patient kan være repræsenteret fra flere ydernumre, hvis patienten f.eks er flyttet.

4.3.3.2. Patientens forløbsplaner

På baggrund af opfyldte inklusionskriterier og konkrete relevante patientdata, kan lægen (og patienten i fællesskab) oprette en konkrete forløbsplan for patienten.
Forløbsplanen består både af:

- Data kopieret fra patientens rådata (synkroniseret fra LPS domænet og evt. suppleret/rettet i Forløbsplaner)
- Forløbsspecifikke patientværdier (f.eks. "anbefalede mål" i Diabetes)
- Beregninger på baggrund af nyeste data (f.eks. BMI og GOLD score). Beregningen sker kun hvis data ikke er leveret fra LPS.

Patientens forløbsplandata er vist i ovenstående figur med kasse: *Patientens forløbsplaner*.

4.3.3.2.1. Tabeller

I ForløbsPlanDB er patientens forløbsplaner realiseret i følgende tabeller:

- FLP_KOL
- FLP_KOL_Medicin
- FLP_Diabetes
- FLP_Diabetes_Medicin
- FLP_Hjerte
- FLP_Hjerte_Medicin

4.3.3.3. Adgang til information

I foregående afsnit blev de forskellige dele af informationsarkitekturen inddelte i scopes. Denne inddeling motiveres af behovet for at specificere:

- Hvem har ansvaret for data?
- Juridiske aspeketer:

- Hvem må tilgå information?
- Hvilke regler findes i forhold til adgang?

Datamodellen er i dag realiseret ved én database, der indeholder information for alle scopes. Alle komponenter har adgang til databasen og kan både læse og skrive.

Dette vil blive behandlet yderligere i To-be arkitekturen.

4.4. Driftsarkitektur

I dag driftes samtlige centrale Forløbsplankomponenter hos en ekstern driftspartner. PLSP har det tekniske ansvar for driften. Det antages, at der eksisterer dækkende Databehandleraftaler.

4.4.1. Miljøer

Der opereres med en liste af miljøer for Forløbsplaner:

1. PLSP udviklingsmiljø: Anvendes af PLSP til at teste egne services. Ikke tilgængeligt for eksterne. PLSP kan teste med egen "test-LPS" for at emulere et rigtigt LSP system.
2. KiAP udviklingsmiljø: Anvendes af KiAP til at teste egne services. Ikke tilgængeligt for eksterne.
3. KiAP LPS test miljø: kan teste med "test-LPS" (XMO) for at emulere et rigtigt LPS. Ikke tilgængeligt for eksterne.
4. PLSP testmiljø: Driftes på A-data. Testmiljøet ligner ikke 1:1 produktionsmiljøet. Indeholder både PLSP og KiAP services. PLSP står for opdateringer af de enkelte services efter aftale med/bestilling fra KiAP. End-to-end test kan foretages af leverandører af LPS mod dette testmiljø. I praksis kan både PLSP og KiAP teste mod dette miljø med "test-LPS" (se punkterne 1 og 2).
5. Medcom testmiljø: Medcom har et fuldt testmiljø stående. Det forventes, at dette adskiller sig væsentlig i forhold til PLSPs miljøer. Medcom er ansvarlig for at vedligeholde og opdaterer dette. Miljøet indeholder både services fra KiAP og PLSP. Det er uklart hvornår opdateringer sker. Dette miljø bruges til at teste funktionaliteter eller andet ved nye installationer.
6. Godkendelsesserver: Godkendelsesserveren anvendes i forbindelse med certificering af LPS. Serveren står hos KiAP og det er KiAP der har ansvaret for at vedligeholde og opdatere den. MedCom har adgang via VPN.
7. Produktionsmiljø: Driftes i datacenter efter aftale med PLSP (i prakis driftes hos a-data). PLSP står for opdateringer af de enkelte services efter aftale med/bestilling fra KiAP.

4.4.2. Leverance- og installationsflow samt konfigurationsstyring

Leverenceprocessen for services til Forløbsplaner i dag er beskrevet og dokumenteret af Medcom.

Leveranceprocessen beskriver et i høj grad manuelt leverance- og installationsflow:

1. Leverancer fra KiAP leveres som en samlet pakke dvs. KiAP FLP Integration API, KiAP FLP Site og KiAP FLP Services leveres i en exe fil (eller som et link til en exe fil).
2. Konfigurationsændringer tilføjes til den centrale database
3. Installationen (eller dele af den) udføres manuelt på alle nodes i det givne miljø:
    a. Gammel installation fjernes
    b. Ny installation af leverance. Der udvælges én node til afvikling af KiAP FLP Services.

c. Konfigurationsændringer tilføjes til konfigurationsdatabasen på node (i praksis Windows registreringsdatabase)

4.4.2.1. Manuel installation

Installationen er som nævnt manuel. Opdatering af enkelt services på en enkelt maskine kan gøres hurtigt. En fuld produktionsopdatering af flere services kan dog i det nuværende setup tage flere timer at gennemføre, da opdateringen skal udføres på hver enkelt af produktions miljøets maskiner. En opdatering kan også inkludere windows opdateringer. Da Forløbsplaner typisk ikke anvendes udenfor almindelig kontortid er et langt servicevindue ikke som sådan noget problem. Men de mange (gentagelser) af manuelle skridt i produktion, gør, at f.eks. kritiske fejlrettelser også tager tid at rulle på.

Yderligere er der fra specielt fra KiAPs side et ønske om at levere oftere. Især i forbindelse med mindre ændringer på de brugerrettede applikationer, er det manuelle leverance- og installationsflow et stort overhead.

Som det er i dag, er konfigurationen og applikationen skilt ad. Det er således den samme deploymentpakke (exe-fil), der installeres i testmiljøer og senere i produktion. Denne opdeling er en forudsætning for at en effektiv automatisering på tværs af miljøer kan etableres.

4.4.3. Skalering

Alle komponenter i Forløbsplaner persisterer tilstand i databasen, hvorfor horisontal skalering af services i dag er opnåeligt.

4.4.4. QA og Dokumentation

I dag foregår det meste af testarbejdet i KiAP og PLSPs udviklingsmiljøer. Manuelle installationsprocedurer nedsætter antallet og hastigheden af deployments til de “officielle” testmiljøer. Der findes ikke tværgående tests på fælles testdata på tværs af leverandører, så den samlede leverance kan for nuværende ikke testes automatisk.

Der findes i dag dokumentation af Forløbsplaner. Dokumentationen er foretaget i (versionerede) PDF dokumenter, som deles efter behov. Dokumenterne indeholder både information vedr. arkitektur, use cases og konkrete snitfladebeskrivelser.

4.4.4.1. Fælles testdata og tests på tværs af leverandører

Der eksisterer i dag ikke fælles testdata eller testprotokoller.

4.4.5. Monitorering og drifts- og platformsrelaterede services

Komponenter/services i Forløbsplaner er forskellige i deres mulighed for monitorering:

- KiAP komponenter logger helbreds/liveness i applikationslogen
- PLSP komponenter udstiller endpoints målrette mod integration med Azures Microsoft Insight

Der er ikke adgang til logs og/eller helbredsoplysninger fra de respektive services udefra (set i forhold til miljøerne). Hvis en leverandør har brug for adgang til data (logs, database mv), så er det nødvendigt at etablere kontakt til driftsleverandøren og bede om disse.

4.4.6. Support

God support til slutbrugerne er altid centralt for it-systemer der er i drift. Supporten på Forløbsplaner i dag er fordelt ud på flere niveauer og flere organisationer:

1. First level support leveres af LPS leverandørerne: Dette er uden for scope af dette dokument
2. Second level support leveres af KiAP.
3. Third level support leveres af KiAP og PLSP i fællesskab (KiAP kontakter PLSP)

Der er i dag forskel på de forskellige organisationers mulighed for adgang til væsentlige oplysninger i forbindelse med support. Overordnet set foregår second level support på på følgende måder:

1. Adgang til Forløbsplaner via LPS: KiAP support[2] opnår denne adgang via TeamViewer i dag. Udfordringen kan være, at en supportmedarbejder på denne måde får adgang til informationer (selvom supportmedarbejderen kun kan læse data) uden at efterlade et logspor. Det er jf. parterne en del af de eksisterende databehandleraftaler
2. Adgang til Support-modulet: Der findes særlige supportsider i komponententen KIAP FLP Site (UI).
3. Adgang til Forløbsplaner via PLSP: Der er i dag ikke adgang til data (hverken database eller logs) for supporten udenfor PLSP. Hvis der ønskes supplerende information i forbindelse med support, sker dette ved, at information rekvireres fra PLSP. KiAP sender mail til udvalgte personer. Det kan f.eks. dreje sig om uddrag fra en bestemt logfil eller afvikling af et SQL script. Det antages at rekvirerede information overføres sikkert og slettes efter brug.

Der eksisterer i dag ikke en formel samarbejdsaftale på supportområdet mellem PLSP og KiAP.

[2] https://kiap.dk/kiap/support.php

5. To-be arkitektur

To-be arkitekturen kommer med konkrete vurderinger og anbefalinger i forhold til forskellige dele af as-is arkitekturen. Punkterne kan opfattes som en bruttoliste i forhold til vores vurderinger af as-is arkitekturen. Punkterne er gennemgået på workshoppen d. 12/8 2020.

Parterne kan gennemgå punkterne og designe konkrete løsninger, hvor de finder det nødvendigt.

Med afsæt i as-is arkitekturen, samt workshops, vil de følgende afsnit beskrive et bud på en to-be arkitektur for Forløbsplaner.

Som i afsnittet vedr. as-is arkitekturen, startes der med et overblik over den foreslåede løsningsarkitektur. I dette afsnit gives et overblik over de foreslåede services og overordnede strukturer. Med afsæt i den foreslåede løsningsarkitektur vil vi dernæst analysere denne med vægt på følgende emner:

- Sikkerhedsarkitektur: Hvorledes er interne/eksterne snitflader sikrede?
- Informationsarkitektur: Hvor lever data? Hvilke komponenter/services har adgang til hvilke data? Hvilke komponenter/services har ansvaret for vedligehold af data (evt. gennem synkronisering)?
- Driftsarkitektur: Hvorledes driftes de forskellige dele af løsningsarkitekturen?

Hvor det er relevant vil vi inddrage diskussioner/oplæg fra den afholdte workshop og motivere valget i to-be arkitekturforslaget.

5.1. Løsningsarkitektur for to-be

Flere af de udfordringer der er blevet identificeret i FLP i forbindelsen med arkitekturvurderingen omhandler adgang til data, f.eks.:

- Hvilke services kan tilgå FLP databasen?
- Hvem må udvikle services der tilgår data?
- Auditlogges der korrekt når data læses/skrives så sporbarhed er korrekt?
- Hvordan distribueres data mellem services?

Konkret skal det sikres at data altid tilgås til relevant formål, med den korrekte sikkerhedsmodel og at adgang logges korrekt. PLSP ønsker at styre denne process, da PLSP har et databehandleransvar overfor LPS.
KiAP efterspørger en struktur, hvor adgang til data er effektivt, både i forhold til, at det er let at udvikle ny funktionalitet og at fremsøgning af data performer.
På de afholdte workshops var der enighed om, at FLP både skal have styr på dataadgang, samtidigt med, at det skal være let at udvikle ny funktionalitet. Parterne er ligeledes enige om, at dette er en afvejning.

Den endelige afvejning og rollefordeling er dog ikke fundet.

Behovene er en klassisk afvejningen mellem “Security” og “Convenience”. Kort sagt er det svært at lave et system, hvor dataadgang, på et detaljeret niveau, er godkendt og

dokumenteret, samtidigt med det er let udvikle ny funktionalitet, specielt på tværs af leverandører.

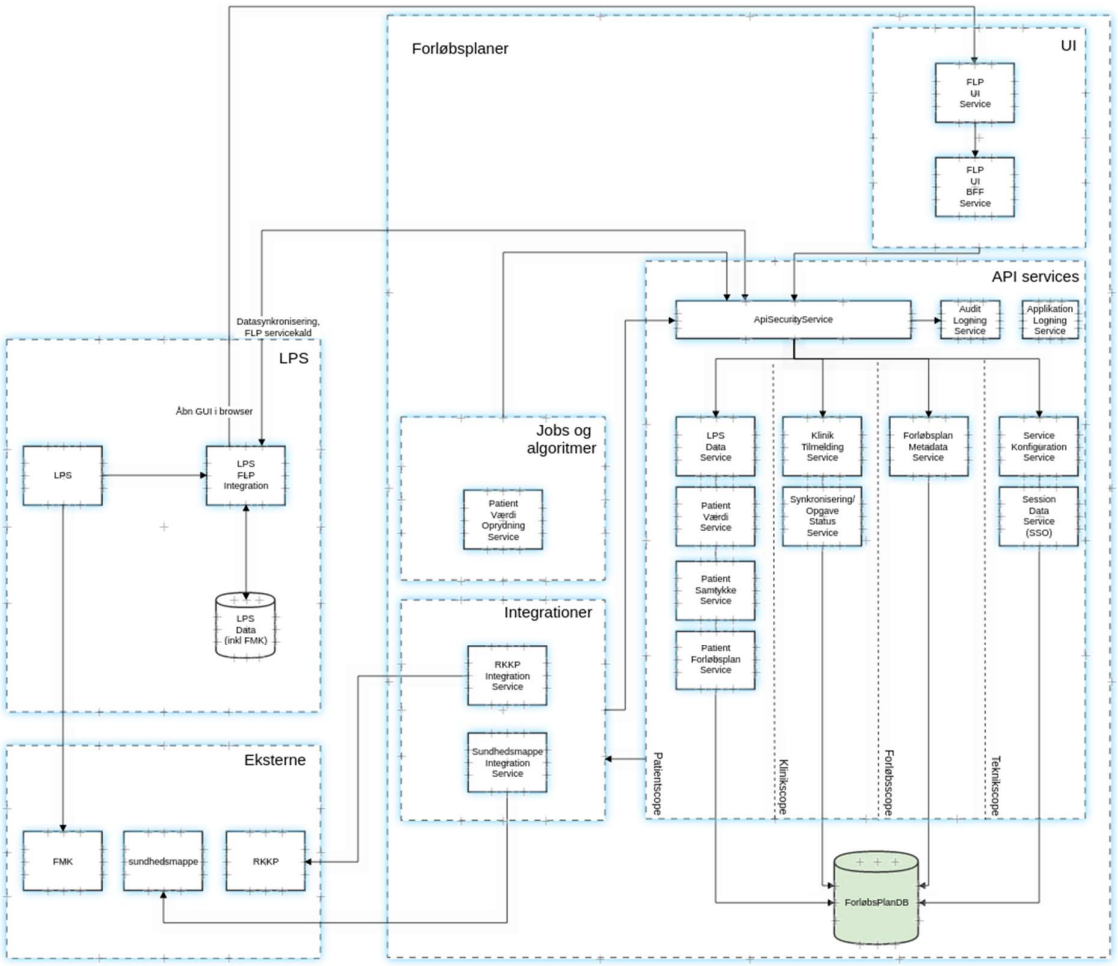
Indeværende afsnit beskriver en løsningsarkitektur, der overordnet afkobler de enkelte services i forhold til dataadgang. Dette gøres for at skabe balance mellem “Security” og “Convenience”. Det er ligeledes en løsning, der bør være implementerbar med mindre omskrivninger af eksisterende services.

I det grafiske overblik over løsningsarkitekturen for to-be nedenfor har vi taget udgangspunkt i den tilsvarende for as-is. Det bør bemærkes, at de eksterne parters (her LPS og RKKP, FMK og Sundhedsmappe) anvendelse og samarbejde med Forløbsplaner er uændret.

Nedenstående skal ses som et forslag, som parterne kan benytte til at diskutere arkitektur, ansvar og delegering af ansvar. Det er vigtigt at sige, at det er parterne selv der skal komme med den endelige løsning.

Den foreslået arkitektur har følgende egenskaber

- Services opdeles i mere dedikerede og på sigt mindre services, hvilket gør dem lettere at teste og deploye.
- Der kan laves trinvis udrulning, hvor det kode der eksisterer i dag f.eks. blot deployes flere gange for at tilfredsstille arkitekturen. En mulighed er, at de eksisterende komponenter forbliver som de er, men det så kun er nye services der implementeres efter den nye arkitektur.
- Der bliver en ensrettet måde at tilgå data og audit logge.
- Der vil blive flere services at deploy. Det betyder et endnu større behov for automatisering.



Det første, der er værd at bemærke er, at de enkelte underdomæner under Forløbsplansdomænet ikke længere er forsynet med navn på leverandør. I to-be arkitekturen er fokus på services og deres ansvarsområde. Den konkrete leverandør af de enkelte services kan besluttes senere i en dialog mellem parterne. Leverandørforholdene kan også ændre sig over tid.
I stedet er fokus på ansvarsområder, snitflader og strukturer.

Overordnet er arkitekturens services inddelt i en række underområder/domæner. I to-be arkitekturen arbejdes der med følgende domæner:

- API services: Disse services tilbyder adgang til FLP forretningsdata bl.a. læse- og skriveoperationer. Der kan være flere leverandører af Api-services, men de vil være underlagt samme retningslinjer.
- Integrationer: Integrationer til eksterne systemer
- Jobs/algoritmer: Batchjobs, der arbejder på FLP forretningsdata f.eks. oprydning
- UI: De brugervendte applikationer med skærmbilleder og præsentationslogik

I det følgende beskrives de enkelte services i hvert af domænerne.

5.1.1. ApiSecurity Service

Som en vigtig del af dette serviceområde findes ApiSecurityService. Denne service har til opgave at verificere de akkreditiver, som præsenteres af anvenderne af de konkrete forretningsservices. Ved at arbejde med en ApiSecurityService opnås en ensartet verificering af sikkerhedsbilletter og tokens på tværs af hele arkitekturen. Udover at lette test- og vedligeholdelsesarbejdet i forhold til nuværende (og fremtidige) sikkerhedsprotokoller, vil denne tilgang gøre det muligt for de enkelte forretningsservices i domænet at fokusere på den del af forretningsområdet, som den enkelte service er sat i verden for at dække. ApiSecurityService har som en del af sin opgave at initiere en passende, ensartet og lovlig auditlogning af de enkelte kald. Auditlog håndteres via en dedikeret service.

5.1.2. Audit logning service

Til auditlogningsformål har to-be arkitekturen defineret en dedikeret service. Denne service kan enten udvikles og vedligeholdes som en del af FLP, men det kan også være en tredjeparts service, der findes til formålet.

5.1.3. Applikation logning service

For at fremhæve adskillelse af auditlogs og applikationslogs er der i løsningsoverblikket defineret en dedikeret service til applikationslogs, ApplikationLoggingService. Rent praktisk kan den konkrete service, der anvendes til at udfylde ApplikationLoggingService, muligvis dækkes af den samme service/produkt, som anvendes til AuditLoggingService.

Til håndtering af applikation/system logs implementeres der en dedikeret service. Servicen kan opsamle systemlogs f.eks. via standard out, syslog eller lignende.

5.1.4. Database access services

Den ovenfor skitserede arkitektur arbejder med flere services, end det var tilfældet i as-is. De konkrete forretningsservices er inddelt i klassifikationerne: Patientscope, klinikscope, forløbsscope og teknikscope med udgangspunkt i de scopes, der blev identificeret i as-is arkitekturens informationsmodel. I to-be arkitekturen anvendes disse scopes til at klassificere de enkelte services i forhold til deres krav til sikkerhed, auditlogning, governance etc.

I det følgende gennemgås de konkrete forretningsservices kort. Nedenstående skal ses som et oplæg til dialog mellem KiAP og PLSP. Det skal ikke læses at parterne fra starten skal lave denne opdeling. Det kan være et trinvist opdeling af allerede eksisterende services.

5.1.4.1. Patientscope

I patientscopet findes de forretningsservices, der opererer på patientrelateret data.

LPS Data Service: Denne service har til ansvar at udstille endpoints til LPS'erne til synkronisering af LPS data mellem LPS og FLP. LPS Data Service arbejder på de dele af FLP datamodellen, der i as-is blev beskrevet som *LPS Data.*

Patient Værdi Service: Denne service har til ansvar at arbejde med de patientværdier, der bliver oprettet i forbindelse med klinikerens arbejde med en patients forløbsplan. Patient Værdi Service har også til ansvar at stille services til rådighed for LPS'erne, således at data inddateret i FLP kan læses tilbage i LPS. Patient Værdi Service arbejder på de dele af FLP datamodellen, der i as-is blev beskrevet som *Patientværdier registreret via Forløbsplaner.*

Patient Samtykke Service: Denne service arbejder med data i forbindelse med patientsamtykke til dataudlevering f.eks. i forhold til Sundhedsmappe. Den arbejder således på de dele af FLP datamodellen, der i as-is blev beskrevet som *Patientsamtykke til sundhedsmappe.* I forbindelse med to-be arkitekturen kunne patientsamtykker generaliseres til også at dække andre integrationer.

Patient ForløbsPlan Service: Denne service udstiller snitflader til at arbejde med data i forhold til konkrete forløbsplaner for patienter. Den arbejder således på de dele af FLP datamodellen, der i as-is blev beskrevet som *Patientens forløbsplaner.*

5.1.4.2. Klinikscope

Klinik Tilmelding Service: Service der kan tilmelde en klinik. En klinik identificeres via et Ydernummer.

Synkroniseringsservice/opgavestatus Service: Synkronisering af data og opgaver med LPS'erne

5.1.4.3. Forløbsscope

Forløbsplan metadataservice: Service til at trække informationer omkring det konkrete forløb.

5.1.4.4. Teknikscope

Service Konfiguration Service: Service der udstiller servicekonfigurationer.

Session Data Service: Service til at gemme og hente oplysninger der skal bruges i forbindelse med sikker sessions overdragelse mellem LPS og FLP

5.1.5. Komponenter og services

I dette afsnit vil vi vurdere på den konkrete opdeling af services i Forløbsplaner i dag.
Et bemærkningspunkt er, at de services, der udgør Forløbsplaner i dag er forholdsvis store. Som det blev beskrevet i foregående afsnit, så har hver enkelt komponent i as-is arkitekturen flere ansvarsområder.

Nedenstående skal igen ses som et forslag, da det er parterne selv har den detaljeret indsigt og kan lave den korrekte opdeling af koden.

Det anbefales at udrulning af en ny arkitektur sker trinvist:

1. Eksisterende services bruges i videst muligt omfang.
2. Det planlægges hvilke services der giver mest værdi at opdele i mindre dele.

3. Services identificeret under punkt 2, deles op i mindre services ved at strukturer allerede eksisterende kode i mindre dele.
4. Når der skal implementeres nye services, f.eks. til at understøtte nye forløb, gøres dette ud fra to-be arkitekturen.

Den konkrete opdeling/afgrænsning af services i arkitekturen kan motiveres ud fra flere aspekter:

- IT tekniske: Skalering og opdatering
- Governance: Hvem har ansvaret for de enkelte delområder.
- Sammenhæng med informationsmodellen: Hvilke dele af informationsmodellen skal den enkelte service have adgang til?

5.1.6. Ens regler for alle services

Antallet af services i FLP stiger. Det giver derfor mening, at parterne sammen kravsætter attributterne for en servicen. Krav til services kan f.eks. være

- Hvordan og hvornår der auditlogges.
- Hvordan og hvornår der systemlogges
- Hvordan en service kan konfigureres
- Hvordan en service installeres
- Hvilke endpoints har en service i forhold til at den overvåges korrekt
- Dokumentationsniveau og krav

Det er op til parterne bestemme regler for eksisterende og nye services. Parterne bør sammen finde ud af, hvad der giver størst værdi og starte der. Parterne vil så løbende kunne udvide regelsættet.

Det anbefales som minimum at services ensrettes i forhold til auditlog og systemlog. Begge bør være til at implementere uden for store omkostninger.

5.1.7. Løsningsarkitekur - Anbefalinger

For nedenstående tabel i dette og kommende afsnit vil følgende kolonnerne fremgå:

1. Observation: En kort beskrivelse af emnet.
2. Vigtighed: En vurdering af vigtigheden for at få observationen løst.
3. Kompleksitet: En vurdering af observationens kompleksitet i forhold til løsningen.
4. Løsning: Et forslag til løsning.

For vigtighed og kompleksitet er det en vurdering, der er lavet ud fra nuværende viden. Det anbefales at parterne laver deres egen vurdering både af vigtighed og kompleksitet men også af selve løsningen.

For løsningsarkitekturen anbefales det, at der kigges på følgende områder

Observation	Vigtighed	Kompleksitet	Løsning
Ensartet adgang til data	Høj	Høj	Det er vigtigt at parterne får aftalt hvordan data tilgås.

			En løsning kan være, som er skitseret ovenfor. Her laves en ny service så alt adgang til data går igennem en central service til dette. Denne håndterer sikkerhed og logning. Services der tilgår data ligger bag denne service. Det kan betyde at eksisterende services opdeles eller deployes flere gange. Det anbefales at der startes småt og at nye services implementeres efter den nye arkitektur.
Ens regler for alle services	Høj	Mellem	Ens regler dokumenteres og efterleves

5.2. Sikkerhedsarkitektur

Sikkerhedsniveauet i forhold til sikkerhedsprotokoller er generelt godt i FLP. Der benyttes (danske) standarder i form af Den Gode Webservice ved kommunikation mellem services. DGWS benyttes også til national infrastruktur.

DGWS har 5 sikkerhedsniveauer. Det højeste sikkerhedsniveau er niveau 5, hvor der skal anvendes digital signatur for hele SOAP-kuverten, og hvor brugeren afkræves password ved hvert nyt HTTP-kald. De 5 sikkerhedsniveauer er:

- 5 Digital_Signatur - Hele SOAP-meddelelsen OCES-signeret (uafviselig)
- 4 Medarbejder_ID signatur - SOSI-id-kort signeret med OCES-medarbejdercertifikat
- 3 System_ID signatur - SOSI-id-kort signeret med OCES-virksomhedscertifikat
- 2 Username_Password - Brugerkontrolleret i eget register
- 1 No_ID - Brugeridentifikation ikke nødvendig

FLP benytter niveau 3 og 4.

Generelt, så bør alle læseoperationer af patientrelateret data kun ske, med et niveau 4 SOSI Idkort, da det er nødvendigt at kunne binde læseoperationer op på en konkret bruger, for at kunne foretage en passende auditlogning. Skrivning/opdatering er ok med niveau 3 SOSI Idkort.

Der er et arbejde i gang, i flere projekter i Danmark, med at kigge på mere moderne sikkerhedsstandarder f.eks. IDWS, her bør FLP naturligvis følge med.

Følgende afsnit beskriver nogle af de områder der bør adresseres i to-be arkitekturen.

5.2.1. Brugeroplysninger

Da 'klinik-login' er knyttet til ydernummeret, så er det ikke tilstrækkeligt i forhold til at identificere en konkret bruger. I dokumentationen er specificeret, hvilke services, der kræver hvilket niveau af SOSI Idkort[3] .

Helt konkret, så findes f.eks. en funktion på PLSP FLP Interface API *GetPatientData* som anvendes til at hente de data, der er oprettet/rettet i FLP via brugergrænsefladen, som i den nuværende løsning kun kræver niveau 3. Da der er tale om læsning af patientdata bør niveauet være på 4.

Alle snitflader bør gennemgås for at bekræfte at sikkerhedsniveauet er korrekt.

5.2.2. Validering af SOSI idkort

Både PLSP service og *KiAP FLP Interface API* validerer, at det medsendte SOSI Idkort er gyldigt og korrekt.

[3] Det er tilgangen som anvendes på Sundhedsdatastyrelsens Nationale Serviceplatform (NSP) i dag.

5.2.2.1. Validering af sammenhæng mellem CVR nummer og ydernummer

Et CVR nummer kan dække over et eller flere ydernumre, så det er ikke umiddelbart muligt at lave en entydig mapning fra CVR til ydernummer. Det kunne måske give mening, at kunne validere, om der er en sammenhæng mellem et CVR nummer og et ydernummer, men der findes ikke en autoritativ kilde i Forløbsplaner i dag, hvor det vil give mening for *PLSP FLP Interface API* at foretage et sådant tjek.

5.2.2.2. Sessionsoverførsel

Det skal overvejes, om sessionsoverførslen tilbyder det nødvendige niveau af sikkerhed og kontekst: Sessionsoplysningerne, der udpeges med sessionid ligger i databasen. For nuværende ligger der i databasen oplysninger om:

- Patientkonteksten (CPR nummer)
- Brugerkonteksten (ydernummer, PL enhed).

Det bør ændres, så disse oplysninger er fyldestgørende, da det er det grundlag, som der skal auditlogges på baggrund af. Ydernummeret udpeger ikke en person, men en organisation. Det kan f.eks. gøres, ved at cache yderligere oplysninger fra den Assertion der allerede er tilvejebragt.

Det kunne yderligere overvejes, om de browserbaserede brugervendte dele af Forløbsplaner burde implementere en mere standardiseret sikkerhedsmodel f.eks. ved standard SAML implementation op i mod en ekstern Identity Provider f.eks. SEB[4] eller NemId.

5.2.3. Anvendelse af DGWS til REST services

Alle kald mellem services er i dag SOAP med DGWS som sikkerhedsprotokol.

I klyngevisning er snitflader implementeret som REST sammen med dele af DGWS. Der er således lavet en proprietær protokol. Klyngevisning er ikke en del af vurderingen og observationen fremgår derfor ikke i nedenstående tabel. Det er dog anbefalingen, at hvis der i fremtiden skal benyttes REST i FLP, at der benyttes en standardiseret og sikker protokol. Her kan parterne eksempelvis kigges på IDWS REST.

5.2.4. Sikkerhedsarkitektur - Anbefalinger

For sikkerhedsarkitekturen anbefales det der kigges på følgende områder

Observation	Vigtighed	Kompleksitet	Løsning
Passende niveauer for læsning af data	Høj	Lav	*GetPatientData* er i dag niveau 3. Det det er læsning bør den være niveau 4. Generelt gennemgåes snitfladerne for at fastsætte om sikkerhedsniveauet er korrekt

[4] https://services.nsi.dk/seb

Validering af sammenhæng CVR og ydernummer	Høj	Lav	Undersøge om der er en mulighed for at verificere sammenhæng mellem cvr og ydernummer
Manglende brugerdata ved Sessionsoverføsel	Høj	Lav	Brugerkontekst skal kunne identificeres (ikke blot ydernummer). Det gøres ved at holde flere oplysninger om den konkrete bruger

5.3. Informationsarkitektur

5.3.1. Governance af forløbstyper

I dag anvendes konfiguration i Forløbsplaner udelukkende til tekniske setups:

- Credentials til kontakt med databasen
- URL'er til andre services
- Kører servicen i test eller prod mode

Ændringer eller tilføjelser af forløb, kræver i dag nye releases. Det bør overvejes om dele eller hele konfigurationen kan gøres uden at der skal releases en nye versioner af FLP services. Det kan give mening at overveje muligheden for at vedligeholde data i forløbsscopet ikke som en del af datamodellen for Forløbsplaner, men som konfiguration. Herigennem bliver det også muligt at arbejde med versionering af forløbstyper samt et egentlig release flow af disse. For eksempel kunne de elementer, der i informationsmodellen bliver omtalt som *Forløbsplan metadata: Forløbstyper, inklusionskriterier, relevant data*, opfattes som konfiguration. På denne måde kunne governance af det lægefaglige indhold i Forløbsplaner afkobles fra software leverancerne og kunne vedligeholdes uafhængig af disse. På denne måde vil man opnå en øget afkobling af det lægefaglige indhold og software leverancerne, og der kunne opstilles en helt særskilt model i forhold til governance og dataansvar, og der vil kunne aftales releases af nye versioner af Forløbsplan Metadata.

Der er i dag de kliniske grupper, der styrer det lægefaglige indhold omkring de enkelte forløb. Der er øjeblikket dog ikke use cases, der beskriver processen omkring administration af forløb. Eksempler kunne være:

- Oprettelse af ny forløbstype
- Ændring af eksisterende forløbstype

Det kunne give mening at udvide use case-beskrivelserne med arbejdsgange omkring forløbshåndtering.

5.3.2. Adgang til information

5.3.2.1. Logningsstrategi

Der bør defineres konkrete krav til auditlogning for komponenterne i Forløbsplaner. Disse krav skal gælde for alle komponenter, der arbejder med data i patient scope.

Der bør laves en opdeling mellem auditlogs og andre logs i Forløbsplaner. De to logtyper benyttes i forskellige usecases og af forskellige personer. Desuden er opbevaringskravene forskellige mellem logtyperne.

Der bør defineres user stories for adgang til disse logs. Dette kunne for eksempel være følgende:

- Som en udvikler ønsker jeg adgang til applikationsloggen for komponent x, så jeg kan fejlsøge.
- Som en auditør ønsker jeg adgang til auditloggen for ydernummer y, så jeg kan verificere, hvilke patienter, der er arbejdet med i Forløbsplaner i perioden a til b.

5.3.2.1.1. Auditlogs

Formålet med at have auditlogs er bl.a. at overholde gældende lovgivning. Så når en kliniker tilgår patientdata enten ved en læsning eller skrivning skal der auditlogges. Audit logs skal gemmes i den lovgivningsdefineret periode.

En auditlog bør som minimum indeholde

- Borgerens/Patients ident
- Hændelsestidspunkt
- Kliniker ident
- Organisations ident / navn
- Aktivitet udført

Det kan overvejes om der også skal indføres "på vegne af", hvis dette skønnes nødvendigt.

Som beskrevet under løsningsarkitektur foreslås det, at der indføres en central service til auditlogning.

5.3.2.1.2. Systemlogs

Systemlogs udtrykker typisk systemets nuværende og tidligere tilstand. Formålet med system logs er f.eks. at få indblik i brugen af systemet og gør det nemmere at fejlsøge.

Det skal løbende verificeres, at system logs ikke indeholder brugerdata.

Systemlogs er centrale for kunne drifte et system korrekt. Det er derfor vigtigt at relevante parter har adgang til systemlogs.

Det anbefales at der indføres en central service til opsamling og udstilling af systemlogs i to-be arkitekturen. Det gør, at systemlogs lettere kan ensrettes og tilgås på tværs af services.

Det anbefales at loglevel for systemlogs kan ændres via konfiguration, så det ikke kræver en ny release.

Der findes både kommercielle løsninger (f.eks. Microsoft application insights) og open source løsninger (f.eks. Grafana og Grafana Loki) til at opsamle og udstille systemlogs. Inden et system vælges, foreslås det at udarbejdes use cases der kan være med til at afdække behovene, så det rigtige produkt vælges.

5.3.2.2. Databaseadgang

Det er muligt på databaseniveau at begrænse databasebrugere til bestemte tabeller via rettigheder. På sigt kan det overvejes at indføre individuelle databasebrugere pr. service. Databasebrugerne kan begrænses via rettigheder, så de kun kan fremsøge de data der er aftalt.

5.3.3. Informationsarkitektur - Anbefalinger

For informationsarkitekturen anbefales det der kigges på følgende områder

Observation	Vigtighed	Kompleksitet	Løsning

Opsamling af auditlogs på tværs af leverandører	Høj	Lav	Nye central service til opsamling og udstilling af auditlogs.
Opsamling af systemlogs på tværs af leverandører	Høj	Lav	Nye central service til opsamling og udstilling af systemlogs.
Governance og konfiguration af forløb	Lav	Høj	Mulighed for at styre oprettelse og ændring af forløb via konfiguration (konfigurationsparamtre).

5.4. Driftsarkitektur

5.4.1. Miljøer

Som beskrevet i as-is arkitekturen eksisterer der en række miljøer både hos KiAP, PLSP og et enkelt hos MedCom.

Det anbefales at parterne beskriver det optimale antal miljøer og deres karakteristika og formål.

Det vil anbefales at deployment kode deles mellem miljøerne, så deployment kan blive ensrettet.

I øjeblikket er der driftsmæssige forskelle mellem miljøerne hosted af PLSP, Medcom og KiAP. Det kan give usikkerhed i forhold til test, QA osv. Hvis installation, testdata og integrationer er forskellige, er det svært at afgøre om nye versioner virker korrekt på et bestemt miljø. Det anbefales, at der arbejdes med en øget automatisering i forhold til installation og deployment, så det bliver lettere at ensrette miljøerne. Desuden at der defineres nogle minimale sets af testdata, der kan deles mellem parterne, så det bliver lettere at teste.

Det anbefales, at der som minimum arbejdes med følgende miljøer (inspirationen og miljøbeskrivelser er hentet fra Sundhedsdatastyrelsens NSP miljøer[5)]:

- PRODUKTION
- PRE-PROD: Testmiljø, der til enhver tid ligner produktion så meget som muligt i forhold til deployede services, konfigurationer samt (brugbart!) testdata.
- CERTIFICERING: Testmiljø, der kan anvendes til certificeringsprocessen (af Medcom).
- TEST1: Et dynamisk testmiljø i den forstand, at det primært anvendes af applikationsudviklere [fra PLSP og KiAP], der kan have behov for hurtig response og ny version. Selv om TEST1 kan være mere ustabilt end TEST2, anbefaler vi udviklere at anvende TEST1, for at få fundet fejl i det miljø, så der er muligt at holde TEST2 mere stabilt. Ny funktionalitet er også altid først tilgængeligt i TEST1.

5 Se https://www.nspop.dk/pages/releaseview.action?pageId=8915610

- TEST2: Et mere stabilt miljø, hvor primært brugere [gennem LSP leverandører] tester nye klient-applikationer, der gør brug af ny ikke-releaset funktionalitet. Miljøet giver således mulighed for at teste og afprøve klient funktionalitet [ny forløbsplansfunktionalitet] mod nye snitflader. Miljøet anvendes typiske også af klientudviklere op mod aflevering af nye releasen til brugerne.

Det er centralt, at miljøerne ligner hinanden så meget som muligt, så QA og leveranceflow også kan ensrettes så meget som muligt på tværs af miljøer.

Som minimum anbefales det at introduceres et staging (pre-prod) miljø, hvor komponenter og versioner af disse er ens med produktionsmiljøet. Formålet med Staging miljøer er:

- Altid at have en kørende miljø der ligner produktion
- Det gør det lettere for LPS'erne at have et stabilt miljø de kan bruge når integrationer skal rettes og testes.
- Det er muligt at genskabe eventuelle fejl på et miljø der ligner produktion så meget som muligt.

Udover ovenstående miljøer kan det være relevant, at have lokale udviklingsmiljøer.

5.4.2. Installation og teknologistakke

Det anbefales at arbejde med automatisering af leverance- og installationsprocessen. Det anbefales yderligere at splitte de nuværende leverancepakker op i flere mindre dele: Én leverancepakke pr komponent/service. På denne måde vil det være muligt at arbejde med forskellige processer for forskellige komponenter. F.eks. vil en komponent, der består af rene brugerflader muligvis kunne leveres hurtigere (og med færre godkendelser) end for en komponent, der indeholder megen forretningslogik. I arbejdet med automatiseringen vil det være relevant at foretage en ensretning af de forskellige miljøer, således at det automatiserede flow kan sættes op på samtlige miljøer.

Teknologistakkene er i dag forskellige for KiAP og PLSP komponenter. I forhold til automatisering af leverance- og installationsprocessen kan dette repræsentere en udfordring. Det bør overvejes, om der skal ske en ensartning af teknologistakkene. Det kan enten ske ved at omskrive dele af applikationerne eller ved en standardisering i måden leverancer bliver pakketeret og leveret.
Her anbefales den sidste løsning, da den er effektiv og har de mindste omkostninger. Det kan f.eks. gøres muligt ved at benytte container teknologi for at ensrette rammerne for en leverance. Der har, på de afholdte workshops, været talt om at anvende container teknologien Docker.

Det anbefales at udbrede Docker til de øvrige komponenter i Forløbsplaner. Ved at levere komponenterne i Forløbsplaner som Docker images, bliver forskellene i teknologistakken uvæsentlig. I forhold til leverance og installation skal der kun tages stilling til, hvordan Docker images skal håndteres – hvorvidt teknologien inden i Docker imaget er PHP eller .Net er således ikke en bekymring for leverance og drift.

Ved introduktionen af docker, bør det overvejes at benytte tooling til at håndtere idriftssættelse af dockerimages f.eks. automatiseringsscripts eller en egentlig Orkestreringsplatorm (som f.eks. Kubernetes).
Det skal også overvejes om de pakkede images kan pushes via et dockerregistry, så filerne ikke manuelt skal overføres ved en ny leverance.

Det anbefales yderligere, at KiAP får mulighed for at initiere deployment af egne services - som minimum på testmiljøerne. En forudsætning for dette er automatiseringen.

5.4.3. Automatisering og Konfigurationsstyring

Parterne kan på sigt arbejde, for at opnå en endnu bedre styring af konfigurationer og deployment. Det gøres for at opnå en endnu bedre compliance og sporbarhed i forhold til konfigurationen af Forløbsplaner henover miljøer og tid.
For systemer, der som Forløbsplaner består af flere forskellige services, er det en fordel at betragte konfigurationen af de forskellige miljøer som en slags selvstændig komponent i systemet på linje med de kørende services. Konfiguration dækker i denne sammenhæng som minimum:

- Hvilke versioner af de forskellige komponenter ønskes?
- Hvordan er hver enkelt komponent konfigureret?
- Hvilke maskiner består dette konkrete miljø af?

Konfiguration kan med fordel opfattes på linje med den kildekode, der ligger bag de udviklede forretningsservices. Ligesom alt andet kildekode, bør konfigurationer også være under versionskontrol.

Koblet med automatisering er konceptet med “konfiguration som kode” et stærkt koncept, der kan anvendes til:

- Dokumentation af, hvornår en ny version af en given service er kommet på et givent miljø.
- Dokumentation af, hvordan hver enkelt komponent er konfigureret i et givent miljø.
- Tilbagerulning af ændringer til et givent miljø.
- Implementation af flows for de enkelte miljøer (f.eks. godkendelsesprocedurer i forbindelse med idriftsættelse af komponenter eller konfigurationsændringer og dokumentation af, at disse flows er afviklet).

5.4.4. QA og dokumentation

Det bør overvejes, om der kan opsættes automatiserede tests, der gennemfører en række af det mest centrale use cases i systemet. I forbindelse med automatisering af deployment kan der opbygges QA trin, der kan udføres automatisk efter endt installation. Hvis det ikke er muligt at afvikle disse i produktion, så som minimum i testmiljøerne.

Som en del af certificeringsprocessen findes der prædefineret testdata i form af FNUX filer for en række kendte testpatienter. Dette datagrundlag vil med fordel kunne bruges som basis for den automatiserede QA proces i testmiljøerne og til at skabe et kendt datagrundlag på tværs af testmiljøer. Ydelser er ikke en del af FNUX og skal derfor testes på anden vis.

5.4.4.1. Standardisering af dokumentation

Dokumentation bør i Forløbsplaner være en del af en leverance på samme niveau som de kørende komponenter. Dokumentationen bør være standardiseret i krav til indhold og bør som minimum omfatte[6]:

1. Drifts- og installationsvejledning: Hvordan skal komponenten konfigureres og installeres? Hvordan kan komponenten monitoreres? Hvordan/hvad logges (applikationslogs, SLA logs, anvendelseslogs, auditlogs)?
2. Design- og arkitekturbeskrivelser: Overordnet beskrivelse af komponenten
3. Guide til anvendere: Hvordan er denne komponent tænkt anvendt? Hvad er betingelserne? Særlige punkter?
4. Snitfladebeskrivelser: Hvilke services udbydes og hvordan kaldes disse?
5. Testvejledning og testdata: Hvordan kan komponenten testes? Forslag til testdata og afvikling af tests.

Det bør overvejes om leveranceprocessen også skal omfatte versioneret dokumentation (der følger komponentens versioneringsstrategi). Ethvert miljø bør indeholde et site med den til miljøet tilhørende dokumentation. Der er i forbindelse med arbejdet med Klyngevisningsprojektet observeret, at definitionen af snitfladen mellem PLSP KV Interface API og KiAP KV UI baserer sig på OpenAPI (Swagger). Dette er en god og udbredt måde at beskrive REST snitflader på. Det kan overvejes, om der i fremtidige projekter, der kræver samarbejde mellem KiAP og PLSP kunne anlægges en “dokumentation-først” i stedet for den nuværende: Dokumentationen af snitfladerne genereres ud fra den færdige service, hvilket kan forsinke det samlede projekt.

5.4.4.2. Fælles testdata og tests på tværs af leverandører

Formålet med at indføre fælles testdata og test på tværs af leverandører er, at højne kvaliteten på leverancer og fange fejl så tidligt som muligt. Datagrundlag og testmetodik kan bruges som en fælles kontrakt i forhold til nye releases.

Det er et stort arbejde, hvis der skal udarbejdes et fælles datagrundlag og dækkende tests for hele FLP. Dette arbejde skal kun sættes i gang, hvis parterne er klar til bruge en del tid på etableringen, den efterfølgende vedligehold og få det integreret i release arbejdet.

Der anbefales følgende (mindre) skridt som bør sættes i gang:

- Der etableres en baseline. En måde at bootstrappe testdata, så udviklere og testere kan få testscenarier på samme datagrundlag hver gang(ikke bare “soft delete”). I dag er man nødt til at skabe nye patienter hele tiden, som gør det sværere f.eks. at automatisere tests.
- Hvis muligt fremskaffes der anonymiseret eksempler fra LPS’erne som kan integreres i baseline data.
- De manuelle tests der foretages i dag bør dokumenteres og deles mellem alle parter. Det kan give en fælles reference i forhold til test.

[6] Indholdsfortegnelse er lavet med udgangspunkt i dokumentationskravene på National Serviceplatform (NSP)

5.4.5. Monitorering og driftsrelaterede services

Monitorering og alarmering kan supplere f.eks. systemlogs til at få et korrekt billede af systemets nuværende og tidligere tilstand.

Monitering kan f.eks. være svartider og tilgængelighed af interne og eksterne services. Alarmering vil være at relevante parter notificeres, når en service har en bestemt tilstand. Det kan være hvis mængden af fejlkoder stiger eller en service ikke svarer.

Komponenterne i Forløbsplaner bør standardiseres i forhold til mulighederne for monitorering. Typisk kan dette ske ved, at hver komponent udstiller et endpoint med helbredsoplysninger, der derefter kan scrapes af monitoreringskomponenter som er en del af infrastrukturen.

Det anbefales at indføre en central service til opsamling monitoreringsdata og alarmering. Det gør at monitorering og alarmering lettere kan ensrettes på tværs af services.

Der findes både kommercielle løsninger (f.eks. Microsoft application insights) og open source løsninger (f.eks. Prometheus stakken) til at opsamle og udstille monitoreringsdata og foretage alarmering. Inden et system vælges, bør der udarbejdes usecases der kan være med til at afdække behovene, så det rigtige produkt vælges.

Det skal overvejes, hvordan adgangen til monitoreringsoplysninger skal sættes op, og hvem der har adgang til disse oplysninger (på hvilke miljøer). Som minimum bør enhver leverandør have adgang til helbredsoplyninger for de komponenter, som leverandøren er ansvarlig for, og have mulighed for at opsætte alarmering for egne komponenter.

5.4.6. Support

Det skal overvejes, om adgangen til f.eks. systemlogoplysninger kan foregå gennem en snitflade til PLSPs platform (API eller brugerrettet webgrænseflade). Eksterne interessenter (f.eks. supportmedarbejdere) skal kunne få adgang til en sådan snitflade ved passende login (2 faktor).

5.4.7. Driftarkitektur - Anbefalinger

For driftsarkitekturen anbefales det der kigges på følgende områder

Observation	Vigtighed	Kompleksitet	Løsning
Ensartet monitorering og alarmering på tværs af leverandører	Mellem	Lav	Behov for monitorering og alarmering skal afklares. Derefter kan en evt. central services til opsamlings af monitorering konstrueres.
Fælles testdata og tests på tværs af	Mellem	Mellem	Der etableres et fælles minimalt dataset til let at kunne bootstrappe et realistisk dataset. pga. af

leverandører			kompleksiteten etableres der ikke et komplet dataset der understøtter alle kliniske usecases.
Der anvendes forskellige teknologistakke og det giver udfordringer i driften	Høj	Lav	Der bruges docker til pakke services/applikationer, så teknologistakke bliver lettere at håndtere for driften.
Installation foregår manuelt og har potentiale for at resultere i fejl	Høj	Mellem	Større brug af automatisering
Der er i dag forskelle mellem test-, certificerings- og driftsmiljøerne	Mellem	Mellem	En øget ensartethed mellem miljøer vil gøre det lettere at foretage QA. At ensrette miljøerne vil kræve en større brug af automatisering
Mangel på stabilit testmiljø for LPS leverandører	Mellem	Lav	Der introduceres et staging miljø
Det skal undersøges om den nuværende remote support giver det nødvendige audit spor og i øvrigt overholder fremtidige krav til databehandleraftaler	Lav	Lav	Ifølge parterne er dette dækket af eksisterende databehandleraftaler