

# Oplæg til tekniske snitfladeændringer, FMK 1.6.0

Teknisk løsningsbeskrivelse

## Version

1.2, 2022-09-08

## Projektnavn

FMK

## Ansvarlig

FMK-teamet  
fmk@sundhedsdata.dk

Claus Hemberg Jørgensen  
+45 22785130  
chj@trifork.com

## Ændringshistorik

Version 1.0, 2021-09-01: Første version

Version 1.1, 2022-02-23: Tilføjet estimator, yderligere detaljering af enkelte afsnit

Version 1.2, 2022-09-08: Første version sendt til høring

<b>Indledning</b>	<b>4</b>
<b>FMK16-53 Modifikator udvides med rolle og FMK16-59 Forenkling af modifikator struktur</b>	<b>4</b>
Rolle	4
Modifikator tilretning	5
XML eksempel	6
<b>FMK16-54 Eksterne sources</b>	<b>7</b>
<b>FMK16-55 Optionelle tekst-datoer ved skrivninger</b>	<b>8</b>
Løsningsforslag og kompatibilitet	8
<b>FMK16-57 Generel response header</b>	<b>9</b>
XML eksempel - læsekald	10
XML-eksempel - opdaterende kald	11
Fejl/advarsler	12
<b>FMK16-58 Generisk fejlhåndtering</b>	<b>13</b>
Flag for angivelse af batch-operationer	15
<b>FMK16-60 Refaktorering af response skemaer</b>	<b>15</b>
<b>FMK16-62 Generisk pagineringsmetode</b>	<b>16</b>
<b>FMK16-63 Mulighed for at returnere, at der er flere (e)CPR-numre, der repræsenterer samme person</b>	<b>18</b>
<b>FMK16-64 Rækkefølge af elementer i bulkald</b>	<b>19</b>
<b>FMK16-66 Adresser</b>	<b>22</b>
<b>FMK16-67 Tillad tom FMKConfigurationList</b>	<b>23</b>
<b>FMK16-68 Både navn/adresse og flag retur ved navne- / adressebeskyttelse</b>	<b>24</b>
<b>FMK16-70 Tilføj oplysninger i adviser, der gør det lettere at afgøre om det er en avis man tidligere har behandlet.</b>	<b>25</b>
Kompatibilitet/udrulning	27
<b>FMK16-70 Tilføj versionId til alle entiteter, der kan opdateres</b>	<b>27</b>
<b>FMK16-76 Fælles elementer i request block</b>	<b>28</b>
<b>FMK16-77 Bløde valideringer som tilbud</b>	<b>29</b>
Løsning	29
<b>FMK16-86 Adgang til FMK konfiguration</b>	<b>30</b>

# Indledning

Den nye FMK 1.6.0 snitflade indeholder dels en række ændringer af klinisk karakter, f.eks. parenteral behandling, men også forslag rent tekniske ændringer i forhold til tidligere 1.4.x snitflader. Nærværende dokument beskriver de enkelte forslag til ændringer, bl.a. efter at have modtaget input fra afholdte mini-teknikermøder henover vinteren 2021/22 omkring disse tekniske ændringer.

Ud over punkterne i dette dokument, forefindes yderligere en række punkter fra FMK 1.6.0 bruttolisten, som p.t. stadig mangler afklaring/prioritering, men som evt. kan finde vej til nærværende dokument.

## FMK16-53 Modifikator udvides med rolle og FMK16-59 Forenkling af modifikator struktur

### Rolle

FMK modifikatorer blev oprettet ud fra visse standarder som engang gav mening, fx at kun læger havde et autorisationsnummer. Derfor har FMK aldrig haft nogen grund til at skulle gemme rollen på kald fra "AuthorisedHealthcareProfessional". Dette grundlag er der dog blevet lavet om på siden, så fx behandlerfarmaceuter nu også benytter autorisationsnummer. I nogle tilfælde kan en person også have flere roller og autorisationer, og dermed bør vi gemme, i hvilken egenskab, den pågældende person har foretaget handlingen som.

Da ikke alle anvender-systemer har samme adgang til stamdata som FMK, kan de ikke umiddelbart ud fra autorisationsnummer genkende, om der er tale om en læge eller en eventuel anden rolle som har lavet ændringen.

Derfor ønsker vi at ændre i modifikator opbygningen, således at rollen altid vil være til stede, uanset hvilken type modifikator der er snak om.

Dette betyder, at FMK fremover skal gemme rollen, som brugeren kalder med, når det er en person som benytter "AuthorisedHealthcareProfessional" (fremover: "HealthcareProfessional"), altså en person som benytter sit autorisationsnummer i sine kald mod FMK, eller en evt. anden kilde der dokumenterer vedkommendes autorisation/rettigheder. Da de ikke i skemaet kan rapportere rollen, kan den læses ud fra rollen på ID-kortet. Fra og med 1.6.0 skal det også være muligt at angive sin rolle i modifikator elementet, og samtidig skal denne også returneres når modifikatoren hentes ud igen. Dette skal indsættes i 1.6.0 skemaerne.

For eksisterende data som allerede er registeret, kan FMK altid berige med en rolle ud fra autorisationsnummerets uddannelseskode, som findes i stamdata.

For at generalisere rolle-elementet og tilføje andre kilder til angivelse af autorisationer end autorisationsregisteret, f.eks. SEB, tilføjes desuden en source attribut til angivelse af kilden. Det eksisterende SpecialityCode element generaliseres ligeledes til <Information> med en typeangivelse, der gør at f.eks. en specialekode vil kunne fremsøges eksplicit. Der vil kunne tilføjes flere Information-elementer med forskellige type-angivelser, hvis nødvendigt.

## Modifikator tilretning

I dag har FMK mulighed for 3 typer aktører ifølge skemaet: 'sundhedsfaglige', 'patienten' og 'andre personer'. Dette er ikke altid fyldestgørende, da vi gennem tiden bl.a. har fået en større anvendelse af system-kald, som i dag "misbruger" 'andre personer' til at rapportere med.

Derfor ville det være oplagt at forsimple denne struktur yderligere således, at det både er nemmere at afgøre om det rent faktisk er en person eller et system som har lavet ændringer, men også for at åbne modellen tilstrækkelig op til, at alle modifikator-typer kan inkluderes uden at misbruge andre aktør-typer.

Den nuværende opsætning er som følger:

- Valget mellem
  - Sundhedsfaglig-element
    - Sundhedsfaglig med Authorisation (Navn, Auth, Speciale)
    - Organisation
    - Optionelt: Autoriseret af
  - Patient
  - Other
    - Person (Navn, + evt CPR)
    - Rolle
    - Optionelt: Organisation
    - Optionelt: Autoriseret af

Det foreslås derfor følgende opsætning af modifikatoren:

- Valg mellem modifikator-type.
  - Sundhedspersoner (læger, sygeplejersker, apoteksansatte m.v.): Navn, Id (f.eks. aut.kode eller cpr-nr.), Information (ved brug af autorisationskoder: uddannelseskode)
  - Patienten
  - Rollespecifikt navn (Ekstra navn(e) til rollen, benyttes i først omgang for system)
- Rolle
- Optionelt: Organisation

- Optionelt: Autoriseret af

Flytningen af rollen ud af Other vil også være en fordel i forhold til [#FMK16-53](#) således rollen altid kan bestemmes ud fra modifikatoren. Gennem validering kan det sikres at optionelle felter altid udfyldes, hvis det er relevant.

At organisation ikke er krævet kunne åbne op for at lægen stadig identificerer sig som læge, selv om de udfører handlinger i en privat kontekst, altså uden om deres arbejde i praksis/hospital. Dette kunne hjælpe med at stoppe misbruget af erstatnings-ydernummer.

## XML eksempel

Eksempel på en CreatedBy af læge, inkl. den anvendte rolle, fx ved oprettelse af lægemiddelordination:

```
<CreatedBy>
  <HealthcareProfessional>
    <Identifier source="Autorisation">12345</Identifier>
    <Name>Karl Karlsen</Name>
    <Information type="SpecialityCode">25</Information>
  </HealthcareProfessional>
  <Role>Læge</Role>
  <Organisation>
    <Name>Psykiatrien, Region Nordjylland - afdeling
Thy/Mors</Name>
    <AddressLine>Niels Bohrs Vej 30</AddressLine>
    <AddressLine>9220 Aalborg Øst</AddressLine>
    <TelephoneNumber>00000000</TelephoneNumber>
    <Type>Sygehus</Type>
    <Identifier source="SKS">7690</Identifier>
  </Organisation>
</CreatedBy>
```

Eksempel på ModifiedBy af patient, fx ved privatmarkering.

```
<ModifiedBy>
  <Role>Borger</Role>
</ModifiedBy>
```

I tilfælde af fuldmagtshavere/værger, angives dette i rollen, inkl. navn og identifier:

```
<ModifiedBy>
  <Role>Fuldmagtshaver</Role>
  <Person>
    <Identifier source="CPR">111111119</Identifier>
    <Name>Karl Karlsen</Name>
  </Person>
</ModifiedBy>
```

Eksempel på ModifiedBy af ikke-autoriseret sundhedsperson, her vist eksempel på Apoteker der identificeres vha. CPR-nummer.

```
<ModifiedBy>
  <HealthcareProfessional>
    <Identifier source="CPR">1212641820</Identifier>
    <Name>Ninna Nilsen</Name>
  </HealthcareProfessional>
  <Role>Apoteker</Role>
  <Organisation>
    <Name>Ørsted Apotek</Name>
    <AddressLine>Rougsøvej 89</AddressLine>
    <AddressLine>8950 Ørsted</AddressLine>
    <TelephoneNumber>00000000</TelephoneNumber>
    <Type>Apotek</Type>
    <Identifier
source="EAN-Lokationsnummer">5790000171033</Identifier>
  </Organisation>
</ModifiedBy>
```

Eksempel på OrderedBy af System, her vist et automatisk system omkring Indkøbskurv for et bosted.

```
<OrderedBy>
  <RoleSpecificName>
    <NameLine>Indkøbskurv</NameLine>
    <NameLine>Bostedet Mellemvang</NameLine>
    <NameLine>Recept-bestilling</NameLine>
  </RoleSpecificName>
  <Role>System</Role>
  <Organisation>
    <Name>Mellemvang</Name>
    <AddressLine>Sommervej 52</AddressLine>
    <AddressLine>7840 Højselv</AddressLine>
    <TelephoneNumber>00000000</TelephoneNumber>
    <Type>Bosted</Type>
    <Identifier source="SOR">308281020066042</Identifier>
  </Organisation>
</OrderedBy>
```

## FMK16-54 Eksterne sources

FMK accepterer i dag en række kendte sources såsom 'Medicinpriser', 'Magistrel' og 'Stærke vitaminer m.v.' for lægemidler, pakninger mm. Ud over dette accepteres 'Local' som en definition på data som FMK ikke kender stamdata på, men som måske kan genkendes af andre ud fra fx DrugId.

Dette princip udvides, så det bliver muligt at angive hvilken lokal datakilde, der er tale om.

Dette gøres ved at udvide brugen af 'Local' definitionen, således at Local kan suppleres med navnet på det datagrundlag som systemet har anvendt, fx localsource='MærkevareTaksten'. Dette vil tillade

at lokale systemer kan signalere et mere præcist datagrundlag, uden at FMK nødvendigvis kender til disse data.

Bemærk at source attributten stadig er udvidbar i xml-skemaet. Det vil sige at der vil også være mulighed for, at FMK på et senere tidspunkt åbner op for nye kendte sources. Dette kunne fx dreje sig om 'Mærkevaretaksten' og 'Nomeco'. Gøres dette kan FMK berige disse sources ud fra stamdata. Inden der åbnes for modtagelsen af nye sources, bør det undersøges om der er overlap, mellem de data der findes i den nye source, og data vi allerede kender fra fx 'Medicinpriser'.

Åbning for localsource attributten ville være generel, og data vil ikke blive valideret af FMK. Hvis en localsource 'forfremmes' til en kendt source, bør source="Local" og localsource="xxx" ikke længere anvendes. FMK vil efter en overgangsperiode validere mod dette, at de så ikke også blev brugt som en localsource source, men så længe FMK ikke har adgang til stamdata, vil der ikke blive valideret på indholdet.

Uanset om der åbnes for nye sources, vil FMK stadig fortsætte med at matche indrapporterede source op mod det stamdata den har til rådighed, således at åbenlys forkert brug af 'Local' på et DrugId som findes i 'Medicinpriser' kan fanges og rettes.

Håndtering af bagudkompatibilitet er simpel, idet der i tilfælde af registreringer med localsource angivelse, blot skal returnes source="Local" uden localsource på gamle snitflader. For registrering på gamle snitflader er der heller ingen problemer, idet localsource attributten ikke er krævet, og dermed vil registreringer fra 1.4.x kunne anvendes og mappes helt identisk i den nye snitflade.

## FMK16-55 Optionelle takst-datoer ved skrivninger

Når der i opdaterende kald refereres til elementer fra taksten, anvendes en sourcedate i elementerne til at angive, hvilken takstversion som den pågældende id er fra.

Det foreslås at fjerne kravet om sourcedate fra taksten, idet de ofte kun giver problemer, og klienter sætter alligevel som regel dags dato på.

## Løsningsforslag og kompatibilitet

Hovedreglen er, at hvis der ikke angives en sourcedate, antages det at der refereres til aktive data. Findes det pågældende id (drugid, indikationsid, m.v.) ikke i en version der er aktiv på kaldtidspunktet, fejler kaldet med ukendt id. Det giver flg. muligheder ved opdaterende kald:

- Der foretages et kald med fx et drugid fra taksten, ingen angivelse af sourcedate, og det pågældende drugid er aktivt iflg. taksten: opdateringen foretages, og der stemples en sourcedate på drugid'et der svarer til den aktuelle takstdato.
- Der foretages et kald med fx et drugid fra taksten, ingen angivelse af sourcedate, men det pågældende drugid er *ikke* aktivt iflg. taksten: opdateringen fejler med en besked om, at det pågældende id (her: drug id) ikke findes.
- Der foretages et kald med fx et drugid fra taksten, med en angivelse af en historisk sourcedate, hvor det pågældende drugid var aktivt iflg. taksten: opdateringen foretages, og den angivne sourcedate stemples på i databasen.

Da de opdaterende kald altid stempler en dato på i databasen, er der ingen databasemæssige ændringer, og dermed er løsningen også bagudkompatibel mht. læsning i den forstand, at en 1.6.0 opdatering uden source-dato vil kunne læses fra 1.4.6.

I de nuværende skemaer er date-attributten alle steder optional, da det ikke er for alle sources, at en dato muligvis giver mening, og derfor bliver kravet om en takstdato håndhævet af de respektive xml request mapperne. Dermed kræver ændringen ingen skema-ændringer, kun kodeændringer.

## FMK16-57 Generel response header

Da vi i de nuværende snitflader har en række tekniske elementer med i en lang række responses, elementer som egentlig ikke har en klinisk betydning, vil det være oplagt at få flyttet håndteringen af disse til et delt element.

Til dette er forslaget en ny generel response-header som så kan indeholde de, typisk tekniske, informationer som klient-systemet kan reagere på.

De foreslåede elementer er indtil videre:

1. Markering af at medicinkort er ugyldigt
2. Markering af at opdatering er sket på baggrund af forkert medicinkort-version, den nuværende 'VersionMismatchWarning' som forefindes på en lang række responses fra FMK.
3. Information omkring privatmarkering eller af andre årsager skjulte data. Dette felt er generaliseret i forhold til tidligere af hensyn til en evt. kommende MinSpærring integration. I tilfælde af udeladte data angives nu et element <HiddenData> med yderligere underelement der beskriver årsagen. P.t. vil kun <NegativeConsent> være implementeret, men altså med mulighed for udvidelser.
4. Status for bulk-opdateringer. (Afhænger af om der kan åbnes for partielle opdateringer i FMK/Receptmodul)
5. Information om paginering (se [FMK16-62](#))
6. Informationer omkring fejl / advarsler i forbindelse med fx valideringsfejl eller udvidede valideringer. (Se også [FMK16-58](#))
7. Advarsler vedrørende potentielle problemer eller ting, som brugeren bør være opmærksom på, i lighed med de eksisterende advarsler på dosisdispenserings-kortet. (Det er dog ikke



hensigtsmæssigt at flytte DD advarslerne til headeren, idet de typisk knytter sig til et specifikt element i selve DD kortet). Et eksempel på dette er advarslen til lægen om en igangværende ekspedition, ifm annullering af recepten.

8. Informationer omkring kendte PersonIdentifiers for den fremsøgte person, i tilfælde af patienten har skiftet CPR-nummer, eller eCPR som er blevet ændret) jvf [FMK16-63](#)
9. FMKConfigurationList jvf [FMK16-67](#)

Da listen helt sikkert ikke er udtømmende for hvilke værdier / informationer man kunne tænke sig kunne blive værdifulde i fremtiden, ville det være oplagt at man benytter sig af at lave skemaet for denne nye response-header åben for ændringer i fremtiden. Den mest effektive måde at opnå dette på, ville være at have muligheden for at benytte en vilkårlig mængde af enhver type element i fremtiden.

XML skemaet indeholder muligheden for følgende opsætning:

```
<sequence>
  <element name="MedicineCardInvalid" type="" minOccurs="0"/>
  <element name="VersionMismatchWarning" type="" minOccurs="0"/>
  ...
  <any minOccurs="0" maxOccurs="unbounded"/>
</sequence>
```

**Afklaring:** vi ved at flere klientsystemer har problemer med <any> elementer i xml skemaerne. Alternativt kan vi udelade dette. Det afhænger lidt af, i hvor høj grad man ser response-headeren som en del af snitfladen, eller om vi kan tillade os at tilføje nye elementer til sekvensen, uden det behøver at kræve en ny snitflade. Det betyder dog, at tilføjelse af nye elementer skal koordineres med klientsystemerne, hvilket evt. kan umuliggøre udvidelsen.

## XML eksempel - læsekald

Alt efter hvilken service der kaldes, kan antallet af informationer som returneres variere, 'VersionMismatchWarning' findes kun ved kald til opdatering af data på medicinkortet, derfor vil dette element aldrig optræde hvis man fx henter medicinkortet.

Eksempel på information i en response-header ved hentning af medicinkort.

```
<ResponseInfo>
  <MedicineCardInvalid/>
  <HiddenData>
    <NegativeConsent>
      <DrugMedication>
        <Identifier>498135138844</Identifier>
        <Version>1341404071666006001</Version>
      </DrugMedication>
    </NegativeConsent>
  </HiddenData>
  <PagingInfo>
    <Drugmedications>
      <PageSize>50</PageSize>
      <TotalSize>67</TotalSize>
    </Drugmedications>
  </PagingInfo>
</ResponseInfo>
```

```

        <LastReceived>
            <Field name="DrugmedicationIdentifier">8236765124124</Field>
        </LastReceived>
    </Drugmedications>
</PagingInfo>
<PersonIdentifierList>
    <PersonIdentifier>
        <Name>Ronny Andersen</Name>
        <Identifier source="CPR">1111110001</Identifier>
        <Validity>
            <Status>Changed</Status>
            <Since>2021-01-01</Since>
        </Validity>
    </PersonIdentifier>
    <PersonIdentifier>
        <Name>Rebecca Andersen</Name>
        <Identifier source="CPR">1111110002</Identifier>
        <Validity>
            <Status>Active</Status>
            <Since>2021-01-01</Since>
        </Validity>
    </PersonIdentifier>
    <PersonIdentifier>
        <Name>Rebecca Andersen</Name>
        <Identifier source="X-eCPR">1111118RA2</Identifier>
        <Validity>
            <Status>Active</Status>
            <Since>2021-05-11</Since>
        </Validity>
    </PersonIdentifier>
</PersonIdentifierList>
</ResponseInfo>

```

Der ses markeringer af at medicinkortet er ugyldigt, personen har privatmarkeret 1 ordinationer på medicinkortet, og der er paginering af medicinkortets ordinationer. Herudover kan det ses eksempler på at personen har fået CPR-skift, og også har et aktivt e-CPR, der er knyttet til et "rigtigt" CPR nummer.

## XML-eksempel - opdaterende kald

```

<ResponseInfo>
    <VersionMismatchInfo/>
    <PersonIdentifierList>
        <PersonIdentifier>
            <Name>Ronny Andersen</Name>
            <Identifier source="CPR">1111110001</Identifier>
            <Validity>
                <Status>Changed</Status>
                <Since>2021-01-01</Since>
            </Validity>
        </PersonIdentifier>
        <PersonIdentifier>
            <Name>Rebecca Andersen</Name>
            <Identifier source="CPR">1111110002</Identifier>
            <Validity>
                <Status>Active</Status>
                <Since>2021-01-01</Since>
            </Validity>
        </PersonIdentifier>
    </PersonIdentifierList>

```

```

        <Name>Rebecca Andersen</Name>
        <Identifier source="X-eCPR">1111118RA2</Identifier>
        <Validity>
            <Status>Active</Status>
            <Since>2021-05-11</Since>
        </Validity>
    </PersonIdentifier>
</PersonIdentifierList>
</ResponseInfo>

```

## Fejl/advarsler

For at tillade, at FMK klienter kan implementere en mere generel håndtering af både fejl og advarsler, skal xml strukturen for returnerede advarsler være på samme form som egentlige fejl der returneres i en SOAPFault. Kort fortalt skal skemaet se ud som følger (se flere detaljer vedr. strukturen i [FMK16-58 Generisk fejlhåndtering](#).):

```

<element name="FMKDetail">
    <complexType>
        <sequence>
            <element name="FaultCode" type="string" />
            <element name="FaultText" type="string" />
            <element name="FaultDetails" type="FaultDetailsType" />
        </sequence>
    </complexType>
</element>

```

Dermed vil et svar fra FMK indeholdende en advarsel kunne se således ud:

```

<ResponseInfo>
    <VersionMismatchInfo/>
    <Warnings>
        <FMKDetail>
            <FaultCode>101111</FaultCode>
            <FaultText>Handlingen vil medføre at xxxx</FaultText>
            <FaultDetails>
                <KeyValueSet>
                    <Key>ElementPath</Key>

<Value>CreateDrugMedicationRequest.DrugMedication[1]
                </Value>

                </KeyValueSet>
                <KeyValueSet>
                    <Key>MedicineCardVersion</Key>
                    <Value>999</Value>
                </KeyValueSet>
                <KeyValueSet>
                    <Key>PersonIdentifier</Key>
                    <Value>2603558084</Value>

```

```
        </KeyValueSet>
      </FaultDetails>
    </FMKDetail>
  </Warnings>
</ResponseInfo>
```

Til afklaring hos klientudviklerne: kunne det være relevant at lade de udvidede valideringer opføre sig anderledes i tilfælde af batch-operationer (UpdateMedicineCard), hvor det nogle gange alligevel ikke vil være muligt at præsentere en advarsel for en bruger? Vi behandler dermed udvidede valideringer ved batch-operationer som om de implicit er markeret til at blive ignoreret, og returnerer samtidig et flag i response-headeren, der angiver, at handlingen er udført som et batch-kald, og at der derfor ikke returneres advarsler fra bløde valideringer? En responseheader fra det samme batch-kald som det foregående eksempel, kunne dermed se således:

```
<ResponseInfo>
  <VersionMismatchInfo/>
  <BatchProcessed/>
</ResponseInfo>
```

## FMK16-58 Generisk fejlhåndtering

FMK håndterer fejlbeskeder i et standard format bestående af:

1. En fejlkode, eksempelvis "2"
2. En fejltekst, eksempelvis "CPR:111111118 (PersonIdentifier) findes ikke"
3. En liste af KeyValue elementer, eksempelvis "[{PersonIdentifierSource, CPR},{PersonIdentifier,111111118}]"

Dette format har været standard igennem længere tid, desværre har den ikke været defineret korrekt i FMKs WSDL filer. Her fremgår det forskelligt alt efter hvilken type (DGWS, IDWS, IDWS-Xua).

FMKs WSDL for DGWS snitflade har defineret indholdet i en soap-fejl til kun at være fejlkoden. WSDL for IDWS har defineret indholdet i en soap-fejl til kun at være Fejlkode+Fejlbesked. IDWS-Xua har defineret fejlkoden efter den standard FMK nu anvender med Fejlkode, Fejlbesked og en liste KeyValue elementer.

I FMK fra 1.6.0 snitfladen samles alle disse til samme struktur, og her genanvendes det format som IDWS-Xua WSDL filen har angivet, da den matcher med det format som FMK i dag benytter.

Dermed bringer vi os også i overensstemmelse med wsdl 1.1. specifikationen, hvoraf det fremgår, at der i SOAP Faults kun må være ét element under <detail>, hvorfor dette specificeres som et FMKDetail element vha. skemaerne.

Derfor vil det nu også være muligt for FMKs klient-systemer at autogenerere kode til mapning af fejl-beskeder fra FMK, og det kan derfor også nemmere lave en programmatisk håndtering af FMK-fejl. Desuden vil der til brug for warnings blive brugt samme FMKDetail element, hvilket åbner mulighed for en mere generisk implementering i klientsystemerne af såvel fejl som advarsler (se [FMK16-57 Generel response header](#)).

‘ElementPath’ elementet, er oprindeligt blevet indført i forbindelse med FMKs udvidede valideringer (også kaldet bløde valideringer), og det blev samtidig efterspurgt, om ikke denne kunne påføres til samtlige valideringer fra FMK. ‘ElementPath’ fremtræder i listen af KeyValue elementer, og ‘ElementPath’ er også efterfølgende blevet indført i en større del af alle FMKs hårde valideringsfejl. ‘ElementPath’ er indført der hvor det har været muligt at påpege fejl i forhold til det indsendte XML. Fejl som relaterer til andre elementer end XMLen, såsom ID-kortet, har FMK ikke mulighed for at definere en ‘ElementPath’ for.

I 1.6.0 tilføjes ElementPath til alle hårde fejl, hvor det giver mening.

```
<wsdl:message name="FMKDetail">
  <wsdl:part name="fault" element="tns:FMKDetail"/>
</wsdl:message>

<element name="FMKDetail">
  <complexType>
    <sequence>
      <element name="FaultCode" type="string" />
      <element name="FaultText" type="string" />
      <element name="FaultDetails" type="FaultDetailsType" />
    </sequence>
  </complexType>
</element>
```

### Eksempel på svar med fejldetaljer

```
<soapenv:Fault>
  <faultcode>Server</faultcode>
  <faultstring>Medicinkortet 2603558084 findes ikke i version
999</faultstring>
  <detail>
    <FMKDetail>
      <FaultCode>3</FaultCode>
      <FaultText>Medicinkortet 2603558084 findes ikke i version
999</FaultText>
      <FaultDetails>
        <KeyValueSet>
          <Key>ElementPath</Key>
          <Value>
            CreateDrugMedicationRequest.DrugMedication[1]
          </Value>
        </KeyValueSet>
        <KeyValueSet>
          <Key>MedicineCardVersion</Key>
          <Value>999</Value>
        </KeyValueSet>
      </FaultDetails>
    </FMKDetail>
  </detail>
</soapenv:Fault>
```

```

    </KeyValueSet>
    <KeyValueSet>
      <Key>PersonIdentifier</Key>
      <Value>2603558084</Value>
    </KeyValueSet>
  </FaultDetails>
</FMKDetail>
</detail>
</soapenv:Fault>

```

## Flag for angivelse af batch-operationer

For klientsystemer der afvikler batches af FMK kald eksempelvis vha. køfunktionalitet, vil det være problematisk med udvidede valideringsfejl, da der ikke er nogen slutbruger til stede, der kan tage stilling til de udvidede validerings-fejlmeddelelser. For at undgå, at de bløde valideringer derved i realiteten kommer til at fungere som hårde valideringer, indføres mulighed for i request-headeren (se at angive, at en service kaldes som en del af en batch-operation. Derved ignoreres alle bløde valideringer uanset hvad der måtte fremgå af selve request'et.

## FMK16-60 Refaktorering af response skemaer

FMK har i de nuværende skemaer for 1.4.6 snitfladen 2 måder at starte effektueringen af en recept på.

Den første metode er igennem et normalt 'StartEffectuation' kald. Denne service kaldes med et 'StartEffectuationRequest' element, og svarer med et 'StartEffectuationResponse' element.

Den anden metode er igennem et 'AutomatedStartEffectuation' systemkald. Denne services kaldes også med et 'StartEffectuationRequest' element, men svarer med et 'AutomatedStartEffectuationResponse' element.

Forskellen mellem de 2 response elementer er, at indholdet i det automatiserede kald ikke indeholder detaljer omkring recepten eller bestilling, men blot ID på recept og bestilling. Dette er ikke specielt optimalt når de faktisk nemt kunne være det samme svar.

Forslaget er derfor at flette de 2 responses sammen, således at det bliver muligt både at returnere detaljerede informationer, hvis servicen er kaldt fra en bruger, eller blot identifiere, hvis det er et system der har foretaget kaldet. Dette er en rent teknisk oprydning som fjerner noget af den unødvendige kompleksitet i FMKs WSDL fil.

En opsætning kunne se ud som følgende eksempel.

```
<choice minOccurs="0" maxOccurs="unbounded">
  <element name="Prescription" type="PrescriptionType"/>
  <element name="PrescriptionAndOrderIdentifier"
type="PrescriptionAndOrderType"/>
</choice>
```

## FMK16-62 Generisk pagineringsmetode

I dag har FMK flere services som benytter sig af paginering i deres svar, dette for at lette byrden både på FMK, og på klient-systemerne under bearbejdning af XML'en. I mange tilfælde er det ikke nødvendigt at se en større mængde ældre data, som er gemt væk bag pagineringen, men i de tilfælde hvor det er, håndterer de eksisterende FMK services ikke pagineringen på en konsistent måde. Desuden 'støjer' paginerings-informationen i svaret, da det er noget rent teknisk, der optræder i sammenhæng med de kliniske data.

I 1.6.0. ændres dette, så paginerings-information flyttes op i request- og response-headerne (Se [FMK16-57](#)).

Det vil betyde at hvis en service understøtter paginering (af en eller flere lister i svaret), og der er flere elementer end der kan rummes på en side, vil svaret indeholde en response-header. Herunder eksempel på svar fra GetEffectuationsOnDrugMedication servicen:

```
<Paging>
  <Effectuations>
    <PageSize>25</PageSize>
    <TotalSize>67</TotalSize>
    <LastReceived>
      <Field name="Identifier">8236765124124</Field>
    </LastReceived>
  </Effectuations>
</Paging>
```

Response headeren Paging optræder kun hvis der er noget i selve svaret, der rent faktisk er pagineret. Ligeledes optræder en elementtype kun hvis denne elementtype er pagineret. Det vil sige hvis der kun var 10 recepter i alt ville Prescriptions elementet ikke optræde i response headeren.

Elementerne under LastReceived er alle de felter svaret er sorteret efter (uanset om der sorteres i stigende eller faldende orden) samt værdien fra den sidste der var med det aktuelle svar.

Hvis klienten ønsker at se side 2, medsendes et Paging-element i en request-header til det efterfølgende kald til den samme service. Denne request-header er opbygget efter samme skabelon som response-headeren, dog uden TotalSize.

Klienten skal blot sende præcis det samme som blev modtaget i LastReceived for at modtage næste side.

Hvis LastReceived ikke angives, returneres side 1 i forhold til den anvendte sorteringsrækkefølge.

Alle services har en default for pagesize, men klienten kan angive en anden sidestørrelse hvis ønsket. Dog indenfor et rimeligt interval. eks. 5 - 100.

Eksempel på request header:

```

<Paging>
  <Effectuations>
    <PageSize>25</PageSize>
    <LastReceived>
      <Field name="Identifier">8236765124124</Field>
    </LastReceived>
  </Effectuations>
</Paging>

```

Paginering understøttes kun på øverste niveau. Det vil sige at i eksempelvis Hent Medicinkort er det ikke muligt at få pagineret effektueringerne på de enkelte lægemiddelordinationer. Her vil det altid kun være muligt at modtage et bestemt antal i forhold til den anvendte sortering. Hvis man har brug for hente de resterende effektueringer, må man anvende Hent lægemiddelordination med paginering af effektueringerne. Det er vurderet som værende for kompliceret at implementere paginering på flere underniveauer, og blev også på minitekniker-møderne vurderet som værende overflødigt og overkompliceret.

Servicen "Hent lægemiddelordination" returnerer et svar der kan være pagineret både mht. recepter og LMO-effektueringer, d.v.s. at Paging-elementet i response-headeren vil kunne se således ud:

```

<Paging>
  <Effectuations>
    <PageSize>25</PageSize>
    <TotalSize>67</TotalSize>
    <LastReceived>
      <Field name="Identifier">8236765124124</Field>
    </LastReceived>
  </Effectuations>
  <Prescriptions>
    <PageSize>25</PageSize>
    <TotalSize>14</TotalSize>
    <LastReceived>
      <Field name="Identifier">479836437773</Field>
    </LastReceived>
  </Prescriptions>
</Paging>

```

FMK klienterne har nu flere muligheder for at hente de resterende af de elementer, der er angivet som paginerede:

- 1) Hvis man kun vil hente alle recepter, skal "Hent lægemiddelordination" kaldes igen men med flg. header:

```

<Paging>
  <Prescriptions>
    <PageSize>25</PageSize>
    <TotalSize>14</TotalSize>
    <LastReceived>
      <Field name="Identifier">479836437773</Field>
    </LastReceived>
  </Prescriptions>
</Paging>

```

- 2) Hvis man kun vil hente alle effektueringer, kan man gøre som i 1) men med flg. header:

```

<Paging>
  <Effectuations>

```



```

        <PageSize>25</PageSize>
        <LastReceived>
            <Field name="Identifier">8236765124124</Field>
        </LastReceived>
    </Effectuations>
</Paging>

```

Ulempen ved dette, er dog at hele LMO'en bliver hentet én gang til. En anden mulighed er at kalde GetDrugMedicationEffectuations med samme header.

- 3) Endelig kan både recepter og effektueringer hentes fortsat pagineret ved at kombinere de to elementer ved fortsat kald til GetDrugMedications (igen hentes hele LMO'en):

```

<Paging>
    <Prescriptions>
        <PageSize>25</PageSize>
        <TotalSize>14</TotalSize>
        <LastReceived>
            <Field name="Identifier">479836437773</Field>
        </LastReceived>
    </Prescriptions>
    <Effectuations>
        <PageSize>25</PageSize>
        <LastReceived>
            <Field name="Identifier">8236765124124</Field>
        </LastReceived>
    </Effectuations>
</Paging>

```

## FMK16-63 Mulighed for at returnere, at der er flere (e)CPR-numre, der repræsenterer samme person

En person kan i løbet af sin levetid have flere CPR-numre (i tilfælde af cpr-skift), og/eller et eller flere e-CPR numre. I e-CPR servicen kan flere e-cpr numre og et evt. cpr-nummer linkes til at være samme person, og denne relation mellem flere person identifiers importeres til FMK vha. krsclient. For at give klientsystemerne en bedre mulighed for at præsentere de alternative person identifiers for brugeren, hvilket især er relevant hvis der findes et nyere person id end det der forespørges med, returneres en liste over alle tilgængelige (e)cpr numre.

I dag er det ikke muligt at forespørge på et tidl. e-cpr nummer for en person, da dette er markeret udløbet. Dette bør ændres af hensyn til systemer der kun har oplysninger om det lukkede e-cpr nummer, og dermed ellers ikke vil kunne få information om det nye cpr nummer.

```

<ResponseInfo>
...
    <PersonIdentifierList>

```

```

    <PersonIdentifier>
      <Name>Ronny Andersen</Name>
      <Identifier source="CPR">1111110001</Identifier>
      <Validity>
        <Status>Changed</Status>
        <StatusCode>60</StatusCode>
        <Since>2021-01-01</Since>
      </Validity>
    </PersonIdentifier>
    <PersonIdentifier>
      <Name>Rebecca Andersen</Name>
      <Identifier source="CPR">1111110002</Identifier>
      <Validity>
        <Status>Active</Status>
        <StatusCode>01</StatusCode>
        <Since>2021-01-01</Since>
      </Validity>
    </PersonIdentifier>
    <PersonIdentifier>
      <Name>Rebecca Andersen</Name>
      <Identifier source="X-eCPR">1111118RA2</Identifier>
      <Validity>
        <Status>Inactive</Status>
        <Since>2021-05-11</Since>
      </Validity>
    </PersonIdentifier>
  </PersonIdentifierList>
</ResponseInfo>

```

For PersonIdentifiers med Identifier source="CPR" kan CPR registerets statuskode endvidere findes i elementet <StatusCode> som supplement til Status-elementets Changed/Inactive/Active muligheder. Det må antages, at der kun er ét aktivt CPR nummer/e-CPR nummer.

## FMK16-64 Rækkefølge af elementer i bulkald

I den nuværende implementation, udføres handlingerne i et UpdateMedicineCard kald i en rækkefølge der er bestemt af FMK implementationen. Ideelt set skal klienterne selv kunne bestemme, hvilken rækkefølge de enkelte handlinger skal udføres i. Det kan gøres ved at ændre kaldet til at være en række generiske handlinger, i stedet for som nu: en række "Opret lægemiddelordination" kald, derefter en række "Opdater lægemiddelordination" kald osv. Desuden tilbydes også muligheden for at angive commit-grupper, d.v.s. en gruppering af operationer, hvor alle operationer i én gruppe commit'es samtidigt til databasen - eller rulles tilbage sammen i tilfælde af fejl. Dermed vil det være klienterne der, ud fra den givne use case, vil kunne sammensætte den optimale samling operationer.

Handlinger der vedrører recepter og handlinger i receptmodulet vil dog ikke kunne indgå i disse commit-grupper, idet recept-modulet udfører sine handlinger i sine egne transaktioner, og p.t. ikke kan dele transaktioner med FMK. Derfor bevares opfordringen til, at recepthandling udføres vha. separate kald, som allerede angivet i den eksisterende snitfladedokumentation.

Eksempel på kald, der foretager opdateringer i to forskellige commit-grupper. Første gruppe indeholder et seponér kald og et "opret LMO" kald. Næste gruppe indeholder kun et "Opret LMO" kald.

```
<UpdateMedicineCardRequest>
  <PersonIdentifier>1111111118</PersonIdentifier>
  <MedicineCardVersion>1341404077657004001</MedicineCardVersion>
  <CommitGroup>
    <WithdrawDrugMedication>
      <Identifier>13143641110921</Identifier>
      <WithdrawnBy>
        ...
      </WithdrawnBy>
    </WithdrawDrugMedication>

    <CreateDrugMedication>
      <CreatedBy>
        ...
      </CreatedBy>
      <BeginEndDate>
        <TreatmentStartDate>2012-08-09</TreatmentStartDate>
        <TreatmentEndDate>2012-08-19</TreatmentEndDate>
      </BeginEndDate>
      <Indication>
        <Code source="Medicinpriser" date="2012-08-06">121</Code>
      </Indication>
      <RouteOfAdministration>
        <Code source="Medicinpriser" date="2012-08-06">OR</Code>
      </RouteOfAdministration>
      <Drug>
        <Identifier source="Medicinpriser"
date="2012-08-06">28100902676</Identifier>
        <Name>Primcillin</Name>
      </Drug>
      <Dosage>
        <SubstitutionAllowed>true</SubstitutionAllowed>
        <ReimbursementClause>klausulbetingelse opfyldt</ReimbursementClause>
      </CreateDrugMedication>
    </CommitGroup>
  </CommitGroup>
  <CommitGroup>
    <CreateDrugMedication>
      <CreatedBy>
        ...
      </CreatedBy>
      <BeginEndDate>
        <TreatmentStartDate>2012-08-09</TreatmentStartDate>
        <TreatmentEndDate>2012-08-19</TreatmentEndDate>
      </BeginEndDate>
      <Indication>
        <Code source="Medicinpriser" date="2012-08-06">121</Code>
      </Indication>
      <RouteOfAdministration>
        <Code source="Medicinpriser" date="2012-08-06">OR</Code>
      </RouteOfAdministration>
      <Drug>
        <Identifier source="Medicinpriser"
date="2012-08-06">28100902676</Identifier>
        <Name>Primcillin</Name>
```

```

        </Drug>
        <Dosage>
            <SubstitutionAllowed>true</SubstitutionAllowed>
            <ReimbursementClause>klausulbetingelse opfyldt</ReimbursementClause>
        </CreateDrugMedication>
    </CommitGroup>
</UpdateMedicineCardRequest>

```

Hvis et af kaldene i første gruppe fejler, rulles både seponer og opret LMO kaldet tilbage. Derefter fortsættes med næste commit-gruppe. Dette er nyt i forhold til de tidl. snitfladeversioner, hvor en enkelt fejl resulterede i, at alt bortset fra receptoprettelser blev rullet tilbage. Derfor er der også behov for et mere detaljeret svar, der beskriver, hvilke handlinger der blev gennemført, og hvilke der fejlede. For hver commit-gruppe i et request, returneres et svar der angiver, om handlingerne i gruppen blev gennemført, eller rullet tilbage. Dermed vil et svar til ovenstående request, hvor alle kald er gået godt, se således ud:

```

<UpdateMedicineCardResponse>
    <PersonIdentifier>1111111118</PersonIdentifier>
    <MedicineCardVersion>1341404077657004001</MedicineCardVersion>
    <CommitGroup>
        <WithdrawnDrugMedication>
            <Identifier>13143641110921</Identifier>
            <Version>7623672632</Versions>
        </WithdrawnDrugMedication>
        <CreatedDrugMedication>
            <Identifier>3443243641110921</Identifier>
            <Version>9862332</Versions>
        </CreatedDrugMedication>
    </CommitGroup>
    <CommitGroup>
        <CreatedDrugMedication>
            <Identifier>1112222333444</Identifier>
            <Version>12134333</Versions>
        </CreatedDrugMedication>
    </CommitGroup>
</UpdateMedicineCardResponse>

```

Hvis den første commit block er fejlet og dermed rullet tilbage, vil den commit-blok, hvori en request er fejlet, indeholde et fejlsvar der følger samme skema som de øvrige fejlkode og -meddelelser der bliver returneret (SOAPFault) og dermed vil svaret kunne se således ud. Bemærk specielt, at Opret LMO handlingen i den efterfølgende commit-group er gået godt.:

```

<UpdateMedicineCardResponse>
    <PersonIdentifier>1111111118</PersonIdentifier>
    <MedicineCardVersion>1341404077657004001</MedicineCardVersion>
    <CommitGroup>
        <Fault>
            <faultcode>Server</faultcode>
            <detail>

```

```

        <FaultCode>10001</FaultCode>
        <FaultText>Lægemiddelordinationen 13143641110921 har åbne recepter der
bør lukkes før seponering</FaultText>
        <FaultDetails>
        <KeyValueSet>
            <Key>DrugMedicationIdentifier</Key>
            <Value>13143641110921</Value>
        </KeyValueSet>
        <KeyValueSet>
            <Key>WarningQuestion</Key>
            <Value>Lægemiddelordinationen på Morfin har åbne recepter. Disse
skal annulleres før du seponerer ordinationen.
Hvis du fortsætter vil seponeringen finde sted og
efterlade åbne recepter på den behandling, du er ved at seponere.</Value>
        </KeyValueSet>
        <KeyValueSet>
            <Key>ElementPath</Key>
            <Value>
                WithdrawDrugMedicationRequest.DrugMedication[1]
            </Value>
        </KeyValueSet>
        </FaultDetails>
    </detail>
</Fault>
</CommitGroup>
</CommitGroup>
    <CreatedDrugMedication>
        <Identifier>1112222333444</Identifier>
        <Version>12134333</Versions>
    </CreatedDrugMedication>
</CommitGroup>
</UpdateMedicineCardResponse>

```

For klienterne betyder dette, at de får meget bedre muligheder for at styre den rækkefølge, som handlingerne i et bulk-kald udføres i, samt bedre muligheder for styring af deltransaktioner, således at en fejl i et kald nødvendigvis medføre, at alle øvrige handlinger rulles tilbage. Dette forudsætter en mere avanceret håndtering af svar fra servicen, idet der samtidig kan komme fejl- og successvar i det samlede svar. Tidligere var der kun ét svar evt. med en enkelt fejl i, der skulle håndteres. Nu skal klienten ind og se på delelementerne i svaret. Ønsker klienterne ikke disse mere avancerede muligheder, kan de blot fortsætte med den nuværende funktionalitet, ved kun at anvende én commit-group for alle de handlinger, der udføres udført samtidigt. Da returneres selvfølgelig også kun ét svar, og dermed er der ingen ændring i forhold til den eksisterende snitflade.

## FMK16-66 Adresser

Adresser for personer og organisationer er i dag kun på struktureret form, d.v.s. opdelt i vejnavn, nummer, etage, postnr. osv. Dette giver en udfordring med de personer der er registreret i eCPR registeret, idet deres adresser kun er opdelt i hhv. postnr., by og en adressetekst, hvori både vejnavn, husnr., evt. etageangivelse findes på en eller flere tekstlinjer.

Det foreslås at overgå til ustrukturerede adresser i alle FMK responses, ud fra en betragtning om, at de adresser leveres ud af FMK ikke kan betragtes som autoritative kilder til adresseoplysninger, men udelukkende en videregivelse af en adresse. Autoritative kilder bør lægge sig op ad grunddataprogrammets standard for adresser, men i de tilfælde, hvor man blot viderekommunikerer en adresse, der kommer fra en autoritativ kilde, er det ofte bedre blot at have et antal adresselinjer.

I de nuværende snitflader, er det kun person-adresser der er problematiske, idet skemaet definerer, at adresserne er strukturerede. For organisationer der anvendes i fx modifikatorer, receptanmodninger, bestillinger m.v. udgøres adressen af 5 fritekst-linjer. For at have en ensartet form på såvel person- som organisationsadresser, foreslås derfor anvendt samme struktur med 5 fritekstlinjer som person-adresser.

Da de eksisterende person-adresser er gemt i databasen som strukturerede skal der foretages en databasemigrering fra strukturerede til ustrukturerede adresser.

## FMK16-67 Tillad tom FMKConfigurationList

I det nuværende skema for FMK findes en 'FMKConfigurationList', som er en del af FMKs response-header. Denne liste er tiltænkt informationer til klient systemerne omkring nogle tekniske konfigurationer i FMK, og er relevante for hvordan systemet skal handle i nogle situationer.

Igennem tiden har der været enkelte værdier i denne liste, bl.a:

- Værdien 'UseNewSuspension' omkring skiftet fra suspendering af medicinkortet til angivelsen af at medicinkortet ikke er ajourført.
- Værdien 'PausePeriodEnabled' som blev brugt under indførelsen af en ny pauserings muligheder. Denne element findes stadig i den nuværende FMKConfigurationList.

FMKConfigurationList er i skemaet defineret som en ikke tom liste. P.t. er der dog ikke nogle relevante konfigurationsværdier at sende ud, og det ville derfor give god mening, at tillade, at listen er tom. Listen vil aldrig være helt udeladt for at undgå fejl for de klienter der forventer den er til stede.

FMKConfigurationList kommer til indgå i den nye generelle response-header sektion. Se [FMK16-57](#)

*BEMÆRK: Hvis FMK16-86 implementeres, udgår dette punkt.*

## FMK16-68 Både navn/adresse og flag retur ved navne- / adressebeskyttelse

Når apotekerne slår op på FMK returneres navn og adresse, uanset om borgeren har adressebeskyttelse eller ej. Det gælder hvis kalderen har apoteks-rettigheden, alle øvrige får AddressProtected-elementet retur i stedet.

I skemaet er der en choice-konstruktion, så man kan ikke få både Address og AddressProtected retur. Til næste version af FMK-snitfladen ønsker apotekerne sig, at modtage en markering af navnebeskyttelsen sammen med navnet, såfremt borgeren har adressebeskyttelse.

Der er flere situationer, hvor det er hensigtsmæssigt at vide at navn og/eller adressen er "beskyttet". Løsningen er at tilføje et Protected element til Name strukturen. Elementet er tomt, men har en source, der indikerer at beskyttelsen er kommer fra CPR registeret. Derved kan apotekerne f.eks. anvende adresse ved forsendelse direkte til kunden, men undlade at printe den på poserne som sendes til medicinudleveringsstederne.

Eksempel på returnering af patient med navne- og adressebeskyttelse vha. apoteks-rettigheden:

```
<Patient>
  <Person>
    <Name>
      <Protected src="CPR"/>
      <GivenName>Anita</GivenName>
      <Surname>Andersen</Surname>
    </Name>
    <PersonIdentifier source="CPR">
      1401006789
    </PersonIdentifier>
    <BirthDate>2000-01-14</BirthDate>
    <Gender>male</Gender>
  </Person>
  <Address>
    <Protected source="CPR"/>
    <MailDeliverySublocationIdentifier>
      Bakkegården
    </MailDeliverySublocationIdentifier>
    <StreetName>Vestergade</StreetName>
    <StreetBuildingIdentifier>2</StreetBuildingIdentifier>
    <FloorIdentifier>4</FloorIdentifier>
    <SuiteIdentifier>TH</SuiteIdentifier>
    <DistrictSubdivisionIdentifier>
      Svejstrup
    </DistrictSubdivisionIdentifier>
    <PostCodeIdentifier>8660</PostCodeIdentifier>
    <DistrictName>Skanderborg</DistrictName>
  </Address>
</Patient>
```

Hvis den kaldende rolle ikke har apoteks-rettigheden returneres name og address elementerne med udelukkende Protected elementet:

```
<Patient>
  <Person>
    <Name>
      <Protected src="CPR"/>
    <Name/>
    <PersonIdentifier source="CPR">
      1401006789
    </PersonIdentifier>
    <BirthDate>2000-01-14</BirthDate>
    <Gender>male</Gender>
  </Person>
  <Address>
    <Protected scr="CPR"/>
  </Address>
</Patient>
```

## FMK16-70 Tilføje oplysninger i adviser, der gør det lettere at afgøre om det er en avis man tidligere har behandlet.

I tilfælde af genudsendelse af avis'er kan klienterne have brug for en måde entydigt at identificere en avis på, og via det afgøre, om de allerede har behandlet avis'en og dermed kan ignorere den. Det kan fx gøres vha. entydig nøgle, eller ved kombination af receipt-id og dato el.lign. Generelt kan det være en god ide at sætte tidsstempel på alle avis'er helt generelt af hensyn til genudsendelse af avis'er hvis kafka pegepinden på pullpoints i nas2 sættes tilbage, og i tilfælde af mistanke om tabte avis'er, hvor systemerne skal gennemføre en bootstrap af avis'er.

I dag er en avis vedr. en bestilling på flg. form:

```
<Notify xmlns="http://docs.oasis-open.org/wsn/b-2">
  <NotificationMessage>
    <Topic Dialect="http://docs.oasis-open.org/wsn/t-1/TopicExpression/Simple">
http://www.dkma.dk/medicinecard/xml.schema/2012/06/01:OrderedEffectuation</Topic>
    <Message>
      <NotifyContent
        xmlns="http://nsi.dk/advis/v10"
        id="1111111118"
        idType="http://nsi.dk/advis/v10/CPR">
        <OrderedEffectuationModification>
          <Action>OrderedEffectuationCreated</Action>
        </OrderedEffectuationModification>
      </NotifyContent>
    </Message>
  </NotificationMessage>
</Notify>
```



```

        <OrderedEffectuation>
            <PersonIdentifier>111111118</PersonIdentifier>
            <Identifier>111333</Identifier>
            <DrugMedicationIdentifier>123</DrugMedicationIdent
            ifier>
        </OrderedEffectuation>
    </OrderedEffectuationModification>
</NotifyContent>
</Message>
</NotificationMessage>
</Notify>

```

Afklaring: hvad er behovet mere præcist? Hvis der er tale om, at man skal ignorere alle advis'er indtil et givet tidspunkt, fordi en kafka pegepind på et pullpoint er blevet nulstillet, burde et tidsstempel på advis beskederne være tilstrækkeligt. Hvis hver klient derimod skal kunne identificere hver enkelt advisbesked på en entydig og generel måde, kan det gøres ved at stemple en GUID på hver enkelt advis-besked.

I henhold til NAS XML-skemaerne er det kun tilladt at tilføje ét any-element til NotifyContent elementet. Dermed er vi tvunget til at placere tidsstempellet/GUID'en i de enkelte adviseringsbeskeder, hvilket vil kræve ændringer i skemaerne for alle beskederne.

Eksempel med timestamp:

```

<NotifyContent
  xmlns="http://nsi.dk/advis/v10"
  id="111111118"
  idType="http://nsi.dk/advis/v10/CPR">
  <OrderedEffectuationModification>
    <Action>OrderedEffectuationCreated</Action>
    <OrderedEffectuation>
      <PersonIdentifier>111111118</PersonIdentifier>
      <Identifier>111333</Identifier>
      <DrugMedicationIdentifier>123</DrugMedicationIdent
      ifier>
    </OrderedEffectuation>
    <Timestamp>2022-08-02T13:07:01</Timestamp>
  </OrderedEffectuationModification>
</NotifyContent>

```

For at kunne gennemløbe et antal modtagne advis'er, og filtrere dem efter tidsstempel, vil det derfor være nødvendigt at se på indholdet af adviseringsbeskeden, og ud fra typen finde <Timestamp> værdien. Hvis feltet havde kunnet placeres under NotifyContent, havde man kunnet lave et helt generisk gennemløb uafhængigt af indholdstypen. Man kan evt. definere, at elementet altid er på 1. niveau af indholdselementet i adviseringsbeskeden.

Bemærk også, at hvis der er behov for entydig identifikation af en advis, er det ikke tilstrækkeligt at tilføje et timestamp, idet der eksempelvis kan oprettes flere orders på samme tid, og dermed vil de to udsendte advis'er være identiske, med mindre man også ser på Identifier-elementet.

Eksempel med GUID:

```

<NotifyContent
  xmlns="http://nsi.dk/advis/v10"
  id="1111111118"
  idType="http://nsi.dk/advis/v10/CPR">
  <OrderedEffectuationModification>
    <Action>OrderedEffectuationCreated</Action>
    <OrderedEffectuation>
      <PersonIdentifier>1111111118</PersonIdentifier>
      <Identifier>111333</Identifier>
      <DrugMedicationIdentifier>123</DrugMedicationIdent
        ifier>
    </OrderedEffectuation>
    <GUID>a2d5a74a-6960-40de-8e3f-056f9fddc974</GUID>
  </OrderedEffectuationModification>
</NotifyContent>

```

Som for Timestamp er det her nødvendigt at lave et gennemløb der afhænger af hver enkelt adviseringsbesked-type, eller om muligt igen definere, at GUID elementet altid er på første niveau af indholdselementet.

## Kompatibilitet/udrulning

Der foreligger allerede nu ønsker om at ændre indholdet af de oprindelige advis-beskeder der udsendes af FMK. Dette er dog ikke ukompliceret, da det skal sikres, at de nuværende klienter ikke fejler ved modtagelse af en advis med yderligere felter. Ønskerne vedrører bl.a. tilføjes af et source-element på PersonIdentifier elementet for derved at åbne op for anvendelsen af ecpr i advis-beskeder. Det kræver dog at de nuværende abonnenter på FMK beskeder kan håndtere en sådan ny attribut. Når der alligevel skal røres ved indholdet af advis-beskederne kunne man samtidig få klientudviklernes feedback til muligheden for at tilføje enten et GUID- eller dato-felt.

## FMK16-70 Tilføj versionId til alle entiteter, der kan opdateres

I de aktuelle snitflader er det kun DrugMedication og PlannedDispensing, der har et versionsid, der checkes i forbindelse med opdatering. For konsistensens skyld bør det også tilføjes til følgende entiteter: Dosisdispenseringsperiode, Dosisdispenseringskort og Pakkegruppe.

Rettelsen består dels af en tilføjelse af en kolonne til de respektive database tabeller samt en migreringskørsel, der opretter versions-id'er på de eksisterende data. Desuden en udvidelse af det interne VO objekt, xml mapning og skemaopdatering samt validering med tilhørende version-mismatch-warning.

## FMK16-76 Fælles elementer i request block

Det kan overvejes, at samle felter, der går på tværs af alle requests, i en fælles header. I de nuværende snitflader er der allerede en række separate headers, som kunne samles i en fælles request-header.

Fig. headers indgår i dag som en del af et request:

- SecurityHeader (jvf. DGWS standarden)
- MedcomHeader (også jvf. DGWS standarden)
- OnBehalfOfHeader: FMK specifik header med oplysninger om person, som der foretages en handling på vegne af
- WhitelistingHeader: FMK specifik header med oplysninger der anvendes til validering af FMK adgangen i henhold til aktive whitelisting, såsom systemnavn, -ejer og -version, OrgUsingId og -Name
- ConsentHeader: FMK specifik header der angiver, om en handling er tilladt som følge af samtykke eller værdispring.

Da det rent skemamæssigt kun er de FMK specifikke headers vi kan flytte rundt på, kan en fælles request header udformes således:

```
<RequestHeader>
  <OnBehalfOfHeader>
  ...
</OnBehalfOfHeader>
<WhitelistingHeader>
  ...
</WhitelistingHeader>
<ConsentHeader>
  ...
</ConsentHeader>
</RequestHeader>
```

Den eksisterende opdeling i 3 separate headers bibeholdes, da det vil lette ændringen for klientsystemerne, og da det giver en mere overskuelig opdeling af request headerens felter. Request-headeren bør kunne udvides med yderligere felter efter behov.

Kandidater til nye felter i request-headeren:

- MinLog2 session id
- Angivelse af, at et kald er en del af en batch-operation, hvorfor udvidede valideringer skal deaktiveres (se også [Flag for angivelse af batch-operationer](#) )

## FMK16-77 Bløde valideringer som tilbud

I dag er der mulighed for at fravælge en blød validering, hvis den håndteres i klientsystemet. Det betyder at som udgangspunkt vil en ny "ukendt" validering slå igennem i alle systemer. Man kunne have typer af bløde valideringer, som kunne opfattes som tilvalg, hvor klientsystemerne skulle bede om dem. F.eks. kunne man bede om at få kaldt beslutningstøtte i forbindelse med oprettelse eller opdatering af en ordination, eller man kunne bede om at få advarsler om leveringsproblemer af specifikke præparater i forbindelse med receptudskrivning.

Bløde valideringer, der laves som tilbud, behøver ikke samme godkendelsesproces som de nuværende, da klienterne fri kan vælge om de ønsker at bruge dem.

Man kan eventuelt anbefale leverandørerne at lave listen af ønskede bløde valideringer til en configurations parameter, så nye valideringer kan slås til i enkelte systemer (på installations niveau), uden at skulle rulle ny version ud.

Hvis alle systemer understøtter konfigurerbare bløde valideringer som tilbud, kan man lade eksisterende bløde valideringer, som mange slår fra, overgå til at være tilbud.

### Løsning

Valideringskoden tilrettes således, at valideringer der udvælges til at være en tilvalgsvalidering, aktiveres vha. værdien "Extended validation enabled" i elementet <ModificationMetadata>. Er denne værdi ikke anført i elementet, udføres valideringen ikke.

Hvis målet er at få alle bløde valideringer gjort til tilbud, kan man overveje at styre hvilke valideringer der er krævede og hvilke der er tilbud, ved at lave en property der indeholder en liste over alle påtvungne bløde valideringer. Med tiden vil denne property blive tom og kan fjernes helt sammen med koden til de påtvungne bløde valideringer.

XML eksempel:

```
<WithdrawDrugMedicationRequest>
  <PersonIdentifier>1111111118</PersonIdentifier>
  <MedicineCardVersion>1341404077657004001</MedicineCardVersion>
  <WithdrawnBy>
    <AuthorisedHealthcareProfessional>
      <AuthorisationIdentifier>2Q5TK</AuthorisationIdentifier>
      <Name>Tess Christoffersen</Name>
    </AuthorisedHealthcareProfessional>
    <Organisation>
      <Name>Lægerne Vestergade</Name>
      <TelephoneNumber>86521348</TelephoneNumber>
      <Type>Yder</Type>
      <Identifier source="Yder">66974</Identifier>
    </Organisation>
  </WithdrawnBy>
</WithdrawDrugMedicationRequest>
```

```

        </Organisation>
    </WithdrawnBy>
    <DrugMedication>
        <Identifier>1971221231</Identifier>
        <ModificationMetadata>Extended validation
enabled</ModificationMetadata>
    </DrugMedication>
</WithdrawDrugMedicationRequest>

```

## FMK16-86 Adgang til FMK konfiguration

I forbindelse med visse typer af snitflade ændringer, er det behov for at klienter ændres adfærd, når en given ændring træder i kraft. Da det ikke er muligt synkront at opdatere alle FMK-systemer, kan det derfor være nødvendigt at klient-systemer udvikler den nye funktionalitet parallelt med den gamle, så det at system selv kan skifte over, når den nye feature slå til i FMK.

For at dette kan lade sig gøre, skal klient-systemerne således kunne se, om FMK kernen kører i den gamle eller i den nye konfiguration.

Der skal tages udgangspunkt i, at konfigurationsændringer som hovedregel ikke ændrer sig dynamisk, men på et planlagt og udmeldt tidspunkt, således at f.eks. en indlæsning ved opstart af et klientsystem vil være tilstrækkeligt. En fuldt dynamisk parametrisering af klienter ud fra FMK konfigurationen skønnes ikke at være nødvendig.

Hidtil har der i samtlige svar været medsendt en "FMKConfigurationList" der beskriver den aktuelle FMK konfiguration. Den har dog vist sig at være vanskelig at håndtere for bl.a. FMK-Online, og desuden er det overflødigt at udsende den i samtlige svar.

I stedet foreslås indført en ny usikret service "GetFMKConfiguration", der på samme vis som den eksisterende konfigurationsliste returnerer en Key/Value liste med oplysninger om de dele af FMK konfigurationen, der er relevante for klienterne. Desuden tilføjes en dato der angiver hvornår konfigurationen senest er ændret. Denne dato kan, hvis klienterne ønsker det, anvendes i headeren til alle FMK kald, og i tilfælde af konfigurationsændringer foretaget efter klienten senest har hentet FMK konfigurationen, returneres en advarsel i responseheaderen om outdated konfigurationsdata.

Eksempel på svar fra GetFMKConfiguration:

```

<GetFMKConfigurationResponse>
  <ConfigurationKey>2022-09-06T11:38:00Z</ConfigurationKey>
  <KeyValuePair>
    <Key>PausePeriodEnabled</Key>
    <Value>true</Value>
  </KeyValuePair>
</GetFMKConfigurationResponse>

```

Eksempel på kald vha. optionel konfigurationsheader:

```
...
<ConfigurationHeader>
  <ConfigurationKey>2022-09-03T03:18:02Z</ConfigurationKey>
</GetFMKConfigurationResponse>
...

<GetFMKConfigurationRequest>
...
</GetFMKConfigurationRequest>
```

Eksempel på advarsel i tilfælde af outdated konfigurationsnøgle:

```
<ResponseInfo>
  <Warnings>
    <FMKDetail>
      <FaultCode>701111</FaultCode>
      <FaultText>ConfigurationKey is out of date</FaultText>
    </FMKDetail>
  </Warnings>
</ResponseInfo>
```