FHIR voor de keten. Dart voor de ervaring.

Verplaats de appgerichte mapping naar een centraal uitrolbare BFF. Geef Flutter precies het model dat het scherm nodig heeft.



We willen hetzelfde.
De vraag is waar het draait.

> “Het schone model bestaat al.
> De bouw ervan is het probleem.”

Appointment Details v3 · pagina 2

HET V3-VOORSTEL

FHIR → adapter in Flutter → model

Bronregels in de backend. Mappingcode in een gedeelde client-adapter.

DIT VOORSTEL

FHIR → Dart BFF → model → Flutter

Dezelfde heldere grens, met appgerichte mapping die centraal kan worden uitgerold.

Gedeeld doel: schone modellen, expliciete semantiek en helder eigenaarschap · PDF p. 2, 4, 25

Van FHIR-lijsten naar één schermwaarde.

extensions.dart — DART · HUIDIGE CODE

```dart
// Polikliniek · regel 86
return serviceCategory
    ?.firstOrNull
    ?.coding
    ?.firstOrNull
    ?.display
    ?.trim();

// Telefoonfallback · regel 65
fallbackPhoneNumber:
    subagendaHCSResource
        ?.telecom?.firstOrNull?.value,
```

01 **Vaak een lijst met één item**

FHIR-velden zoals `telecom` en `coding` zijn arrays. In onze praktijk bevatten ze vaak maar één waarde.

02 **Dezelfde mapping in de BFF**

Bij meerdere waarden pakken we hier de eerste. Diezelfde `firstOrNull` kan in de BFF blijven staan.

03 **Flutter ontvangt het resultaat**

Het DTO bevat de gekozen schermwaarde. Een compatibele mappingwijziging kan via een serverdeploy beschikbaar komen.

Code: extensions.dart:65, 80–96 · Vaak één item: praktijkervaring van het team. De BFF verplaatst de mapping en de uitrol ervan.

De app regisseert een kleine integratieketen.

1

PARALLEL STARTEN

- Appointment ophalen
- Educatie-Communication
- Zorginstellingen

→ IDs en context

2

DAARNA PARALLEL

- Videolink-Communication
- HealthcareServices
- CMS-items per educatie-ID

→ zo nodig

3

LEGACY-FALLBACK

- Subagenda zoeken
- Afspraakcode zoeken

Daarna pas samenvoegen tot `AppointmentDetails`.

↳ **Deze regie kan op de server.** De app vraagt om afspraakdetails; de BFF kent de benodigde bronnen en hun afhankelijkheden.

Code: getAppointmentDetails:439–595; extensions:216–242. Paden verschillen per bron/cache. Check-in, samenvatting en lazy locatie laden daarnaast.

De winst zit in de uitrol.

60.000*

toestellen, meerdere appversies

Een fix in de client-mapper gaat mee in een apprelease. Het bereik volgt de update-adoptie.



1

centraal beheerde service

Een compatibele mappingfix kan via een BFF-deploy beschikbaar komen bij de volgende fetch.

- Uitrollen · monitoren · terugrollen

Eén service kan meerdere replicas hebben. Een gedeelde SDK centraliseert de **code**; een BFF centraliseert ook de **uitrol van de mapper**.

* Omvang door het team genoemd, niet uit code gemeten. Geldt na BFF-adoptie, binnen ondersteunde contracten; cache kan verversing uitstellen.

FHIR blijft de ruggengraat.
De BFF wordt de appgrens.

DOMEIN / INTEGRATIE

{ }

Bestaande backend

HiX / Sanday

Genormaliseerde FHIR

Bronregels en autorisatie

FHIR



includes of meerdere calls

MOBILE-TEAM

↳

Dart Frog BFF

Orkestratie en mapping

Appgericht JSON-contract

Expliciete deelresultaten

Zelfstandig deploybaar

DTO



gedeelde Dart-modellen

PRESENTATIE

▯

Flutter-app

Schermen en interactie

Taal en tijdzone

Device- en offlinegedrag

App → BFF → .NET, met hetzelfde Bearer-token. .NET valideert het JWT en de patiënt-/node-toegang; de BFF mapt de toegestane responses.

Doelarchitectuur · FHIR blijft beschikbaar voor andere afnemers. Een eventuele pilot kan zich beperken tot de read-compositie voor afspraakdetails.

Een dunne app.
Een duidelijke verdeling.

01 **FHIR-backend** Backend-team	**Wat betekent de bron?** Normalisatie van HiX/Sanday, relaties, profielen en domeinregels. .NET verzorgt JWT-validatie, patiënt-/node-autorisatie en mutaties.	Bijvoorbeeld betekenis van afspraakstatus
02 **Dart BFF** Mobile + platform	**Wat heeft deze ervaring nodig?** Token doorgeven, toegestane brondata combineren en mappen. Appgerichte veldselectie, fallbacks en stabiele sectiestatussen.	Bijvoorbeeld één contactveld voor dit scherm
03 **Flutter** Mobile-team	**Hoe ervaart de patiënt dit?** Weergave, lokalisatie, loading, interactie, lokale cache en device-permissies. De app blijft verantwoordelijk voor goede UX.	Bijvoorbeeld een afspraak in de kalender

Bronsemantiek wordt samen vastgelegd. Tijdelijke compatibiliteitsregels in de BFF krijgen een eigenaar en een verwijdermoment.

Betere FHIR-queries helpen.
Ook achter een BFF.

A Includes + veldselectie

`_include` volgt bestaande referenties. `_elements` selecteert in R4 top-level velden, niet de included resources.

> `&-include-elements=…`
>
> De V3-uitbreiding kan ook includes minimaliseren, zodra de backend dit expliciete custom contract ondersteunt.

B Meerdere servercalls

De BFF gebruikt de beschikbare endpoints. Onafhankelijke calls kunnen parallel; afhankelijke calls blijven geordend.

> OOK ZONDER CUSTOM UITBREIDING
>
> FHIR- en CMS-data samenstellen via de bestaande backend. Token-doorgifte, capabilities en paginering zijn bij uitwerking te toetsen.

Beide routes leveren **hetzelfde DTO-contract aan Flutter.**

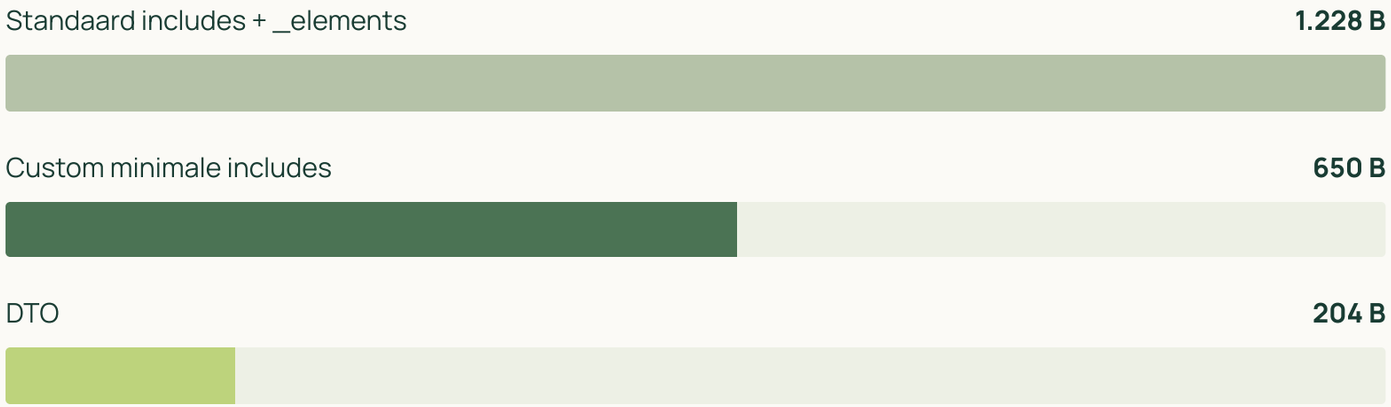
De querystrategie kan veranderen zonder appupdate.

PDF p. 5–10 · HL7 FHIR R4 Search §3.1.1.5.10 · Eén apprequest betekent niet één upstream-call of gegarandeerd minder latency.

446 bytes is niet de businesscase.

De conclusie van V3 is terecht: kies hier op onderhoudbaarheid.

DEZELFDE 7 VELDEN + NODE-ID · GZIP



Overgenomen uit de aangeleverde PDF; niet opnieuw gemeten.

3,57 ms

extra body-overdracht voor custom FHIR versus DTO bij 1 Mbit/s.

Niet gemeten in deze PoC:
de volledige appflow, productie-I/O, een extra BFF-hop en Dart op een toestel.

↳ De CPU-metingen zijn Node-hostmetingen. Ze bewijzen geen mobiele snelheidswinst. Een praktijktest zou de echte keten moeten meten, inclusief de extra hop.

PDF p. 18–23 · 650 − 204 = 446 B · 446 × 8 / 1.000.000 s = 3,568 ms. Geen end-to-end laadtijd.

Waarom een Dart-server naast onze .NET-backend?

.NET beheert FHIR en domeinregels. De Dart BFF geeft het mobile-team een eigen appcontract.

01 De kennis is al in het team

Mobile-developers kennen Dart. Dezelfde taal voor app en BFF verlaagt de instap en vermindert overdracht voor appgerichte wijzigingen.

02 Exact hetzelfde DTO-package

Server en app delen velden, nullability en JSON-codecs. Eén Dart-definitie, direct aan beide kanten te importeren.

03 Mapping en tests hergebruiken

Bestaande pure-Dart-logica en testfixtures kunnen na ontkoppeling meeverhuizen. De bronkennis blijft bij dezelfde developers.

packages/contracts — ILLUSTRATIEF · VERKORT

```dart
@JsonSerializable()
class AppointmentDetailsDto {
  // Constructor en codegen weggelaten
  final String appointmentId;
  final String? polyclinic;
  final String? phoneNumber;
}
```

DART BFF `toJson()` → JSON → FLUTTER `fromJson()`

↳ **Dart Frog maakt de HTTP-laag concreet:** routes, middleware en dependency injection. De reden voor Dart is teamkennis en codehergebruik.

Onafhankelijke uitrol is een BFF-voordeel, ook met .NET. Dart voegt een runtime toe; serverbeheer en compatibiliteit blijven nodig.

Samen ontwikkelen. Onafhankelijk uitrollen.



01 **Eén wijziging in context**

DTO, mapper en consument in dezelfde pull request beoordelen.

02 **Een smalle gedeelde kern**

Contracten zonder Flutter-, FHIR- of serverdependencies. UI-state en bronadapters blijven in hun eigen laag.

03 **Twee releasepaden**

App naar de stores; server naar het platform. CI controleert ook eerder uitgeleverde contracten.

Voorgestelde structuur · De huidige AppointmentDetails is een domeinmodel met EducationResource-afhankelijkheid, geen direct herbruikbaar transportpackage.

Eén repository.
Meerdere versies in het wild.

APP N−2

APP N−1

APP N



ONDERSTEUND CONTRACT

`/v1/…/details`

Een nieuwe mapper, dezelfde afgesproken veldbetekenis.

Server eerst.

Daarna de app die nieuwe optionele velden gebruikt. Oudere apps blijven werken.

Compatibel toevoegen

Optionele velden en defaults. Onbekende enumwaarden expliciet afhandelen.

Breuken versioneren

Een andere betekenis of vereist veld vraagt migratie, eventueel een nieuw endpoint.

Oude clients testen

Golden JSON-fixtures én oude codecs. Ondersteuningsvenster baseren op appadoptie.

N−2 / N−1 / N is illustratief, geen afgesproken supportbeleid. Nieuwe schermmogelijkheden kunnen nog steeds een apprelease vragen.

Eén foutcontract voor de app. Bronkennis op de server.

De app kan vandaag al deelresultaten tonen. De BFF centraliseert de vertaling van bronfouten.

01 / BRONNEN · VOORBEELD

Verschillende oorzaken

`Timeout` bij ophalen videolink

`HTTP 503` van de achterliggende API

De BFF bepaalt: tijdelijk niet beschikbaar. Een geslaagde lookup zonder link is juist `absent`.

→

02 / BFF · STABIEL CONTRACT

Eén appgerichte foutcode

```
{
  "video": {
    "status": "unavailable",
    "errorCode": "VIDEO_UNAVAILABLE",
    "retryable": true
  }
}
```

De overige afspraakdetails worden gewoon meegeleverd.

→

03 / FLUTTER · PRESENTATIE

Reageren op dat contract

- Behoud de afspraakdetails.
- Toon een eigen vertaalde melding voor de videolink.
- Bied opnieuw laden aan als `retryable` waar is.

De app blijft ook netwerkfouten naar de BFF zelf afhandelen.

↳ **Nieuwe bronfout? Pas de vertaling centraal aan.** Zolang het foutcontract gelijk blijft, hebben ondersteunde apps daarvoor geen nieuwe release nodig.

Voorstel · Een eigen foutcode kan ook uit .NET komen. De BFF bundelt de appgerichte vertaling bij mobile. Een 401/403 van .NET wordt doorgegeven.

De argumenten staan. De grens kan een stap opschuiven.

TERECHT PUNT UIT V3	WAT DAT BETEKENT VOOR DE BFF
“Het vlakke model bestaat al.” p. 2 en 11	Precies. Behoud het repositorycontract en vervang de implementatie erachter. De winst zit in het opbouwen en uitrollen.
“De prestatiewinst is marginaal.” p. 23	Eens, voor deze fixture. Maak wijzigbaarheid, foutdiagnose en release-afhankelijkheid de hoofdcriteria.
“Een SDK bundelt de FHIR-kennis.” p. 12 en 17	Dat helpt. Als de mapper in de app draait, blijft een mapperfix een apprelease. Dezelfde helpers kunnen ook in de BFF draaien.
“Een extra service kost beheer.” p. 17	Dat is de echte tegenprestatie. Vergelijk die met clientcompatibiliteit, custom querycontracten en de tijd om fixes uit te rollen.

Beknopte parafrases van de aangeleverde PDF. De V3-route blijft een valide optie; een eventuele pilot kan toetsen of centrale uitrol de meerkosten waard is.

Een service erbij moet zijn plek verdienen.

Een extra hop

Meet end-to-end p50/p95. Draai dicht bij de backend, begrens fan-out en stel per bron timeouts in.

Een extra storingspunt

Healthchecks, traces, capaciteitsgrenzen en rollback. Mobile beheert de service samen met platform.

Toegang blijft in .NET

BFF geeft Bearer-token door. .NET valideert JWT en patiënt-/node-toegang. Een 401/403 gaat terug naar de app.

Een extra contract

Onderhoud oudere clients. Begin zonder persistente patiëntcache; log metadata en foutcodes, geen payloads of tokens.

Dart-kennis verlaagt de instap, niet de operationele verantwoordelijkheid. Benoem een service-eigenaar en maak beheerkosten onderdeel van het besluit.

Een pilot kan de afweging concreet maken.

Als we deze richting kansrijk vinden, biedt **Appointment Details** een afgebakende casus om de meerwaarde te toetsen.

01

Afbakenen

Kies samen representatieve afspraken en bronregels. Bepaal welke vragen een proef moet beantwoorden.

MOGELIJKE TOETS · SCOPE

02

Uitproberen

Eén gedeeld DTO en één BFF-route achter de bestaande repositorygrens, met beschikbare endpoints.

MOGELIJKE TOETS · HAALBAARHEID

03

Vergelijken

Vergelijk schermuitkomsten, oudere clients en de hele keten. Probeer een mappingfix zonder apprelease.

MOGELIJKE TOETS · MEERWAARDE

04

Afwegen

Leg onderhoudswinst naast beheer en kosten. Bepaal samen welk vervolg de uitkomsten rechtvaardigen.

MOGELIJKE TOETS · VERVOLG

Denkbare scope | Eén read-contract | Mapping en foutvertaling | Bestaande repositorygrens

Een pilot is een optie ter bespreking. Scope, capaciteit en eventuele planning bepalen we samen als daar draagvlak voor is.

Hier moet de BFF het verschil maken.

01 Gelijke betekenis

Dezelfde bedoelde schermvelden, met expliciete bronprioriteiten. Toets dit op representatieve afspraken en ontbrekende data.

BETROUWBAARHEID BEHOUDEN

02 Een fix zonder apprelease

Een compatibele mappingfix via een serverdeploy. Maak zichtbaar wat dit oplevert in doorlooptijd, bereik en rollback.

SNELLER KUNNEN CORRIGEREN

03 Een acceptabele keten

Vergelijk calls, gzip-bytes, foutpercentage en p50/p95 op dezelfde toestellen en scenario's, inclusief BFF-hop.

DE PATIËNTERVARING BEWAKEN

04 Beheer dat past

Weeg de verwachte onderhoudswinst af tegen hosting, observability, incidentrespons en het eigenaarschap van de service.

BATEN EN KOSTEN SAMEN AFWEGEN

Criteria voor de afweging en een eventuele praktijktest. Vergelijk ook met de verbeterde FHIR + SDK-route uit V3.

Geef de mapping een eigen releaseknop.

IS DIT EEN KANSRIJKE RICHTING?

Willen we appgerichte mapping centraal beheren, met het mobile-team als eigenaar van het contract?

1	2	3
richting bespreken	vragen verhelderen	vervolg kiezen

Als we die meerwaarde herkennen, kunnen we samen bepalen of een kleine pilot de juiste vervolgstap is.



De concrete aanknopingspunten.

Het vlakke model bestaat	`lib/repositories/appointments/appointments_repository/models/appointment_details.dart:7–70` Polikliniek, instructie, tijdvenster, resources en een telefoongetter met locatieprioriteit.
Orkestratie en legacy-fallback	`lib/repositories/appointments/api_appointments_repository/api_appointments_repository.dart:439–595` Twee parallelle ophaalfasen, conditionele HealthcareService-fallbacks en de mapping naar AppointmentDetails.
Selectie, bronregels en CMS	`lib/repositories/appointments/api_appointments_repository/extensions.dart:33–242` Participant-selectie op display, eerste coding/telecom, sentinel-tekst, tijdvensterprioriteit en CMS-requests.
Het scherm is breder dan de PoC	`lib/features/appointments_details/bloc/appointment_details_bloc.dart:47–125, 280–347` Details, check-in en samenvatting worden gestart; locatie volgt met een afzonderlijke laad- en foutstatus.
Compatibiliteit is nu al clientwerk	`lib/services/appointments/models/responses/appointment_location_route_response.dart:14–36` De parser ondersteunt tijdelijk zowel een object als een array-wrapped object uit verschillende backendversies.

Lokale checkout gelezen op 09-11-2026 · HEAD b7db10c73 · De PDF gebruikt develop a557ea3; dit zijn verschillende snapshots.

Voortbouwen op V3.

V3-pagina	Wat het voorstel laat zien	Vertaling naar dit voorstel
2–4, 11–15	Een vlak model, één mappinggrens en expliciete selectie zijn wenselijk.	Behoud de modelgrens; draai de appgerichte mapper op de server.
5–10	Includes vragen echte relaties. R4-veldselectie is beperkt. Custom include-selectie is nog te bouwen.	Gebruik beschikbare calls; verbeter upstream-queries zonder het appcontract te wijzigen.
16–17	De doorsnede is begrensd. Een BFF heeft extra service- en contractkosten.	Toets bij uitwerking de afgesproken read-scope en neem operationeel beheer mee.
18–23	Synthetische payloads; CPU op Node-hosts. Geen Dart-toestel- of productie-I/O-metingen.	Gebruik de getallen als context. Meet bij een praktijktest de echte keten, ook tegenover de SDK-route.
24–25	Klein beginnen, repositorycontract behouden en gefaseerd invoeren.	Een mogelijke aanpak als we besluiten deze richting te beproeven.

↳ De afweging gaat over **de locatie en het eigenaarschap van de mapping**. FHIR-normalisatie, heldere contracten en goede tests helpen beide routes.

Bron: aangeleverde tekst en pagina-afbeeldingen van “Appointment Details v3 — architectuur, dataminimalisatie en CPU”, 25 pagina’s.

Een DTO dat de bedoeling vertelt.

GET `/v1/nodes/{nodeId}/appointments/{id}/details` VOORGESTELD


```
{
  "appointmentId": "demo-42",
  "nodeId": "demo-node",
  "polyclinic": "Cardiologie",
  "phoneNumber": null,
  "importantInformation": "Neem uw overzicht mee.",
  "timeFrame": "exact",
  "video": {
    "status": "unavailable",
    "url": null,
    "errorCode": "VIDEO_UNAVAILABLE",
    "retryable": true
  },
  "education": {
    "status": "ready",
    "items": []
  }
}
```


Een schets, geen afgerond API.

Bij verdere uitwerking leggen we velden, typen, foutcodes en secties vast. Ook tijdzone-, null- en enumgedrag hoort bij het contract.

JWT-validatie blijft in .NET.

De BFF geeft de Authorization-header door bij elke patiëntgebonden backendcall. .NET controleert token en patiënt-/node-toegang; 401/403 wordt doorgegeven.

Model en weergave blijven apart.

Het gedeelde DTO is het transportcontract. De app kan dit dun omzetten naar het bestaande domeinmodel en eigen UI-state.

Volledig synthetische voorbeelddata · FHIR-mapping zit server-side; de runtime JSON-grens vraagt nog steeds validatie en contracttests.

Onderbouwd waar het kan. Meetbaar waar het moet.

PRIMAIRE BRONNEN

Patient App

Lokale broncode, HEAD `b7db10c73`, gelezen op 9 november 2026. Paden en regels staan in bijlage A.

Appointment Details v3

Aangeleverde PDF-inhoud, 25 pagina's. Referentiebranch volgens het document: `develop a557ea3`.

Technische documentatie

- HL7 · FHIR R4 Search en _elements ↗
- Dart Frog · routes, middleware en testing ↗
- Microsoft · Backends for Frontends-patroon ↗

ZORGVULDIG INTERPRETEREN

60.000 toestellen is input van het team, geen vastgesteld aantal actieve gebruikers of gelijktijdige requests.

212 keer firstOrNull in 79 bestanden is een teksttelling in `lib/**/*.dart`, exclusief `*.g.dart`, `*.freezed.dart` en `*.mocks.dart`. Dit is geen telling van fouten.

Benchmarkgetallen zijn overgenomen uit V3. Er is geen live backendbenchmark of Dart Frog-prototype uitgevoerd voor dit deck.

De architectuur en vervolgstappen zijn voorstellen. Een eventuele pilot kan onderbouwen of de winst opweegt tegen de extra service.

Verder lezen: SOURCES.md · Presentatie en printversie bevatten dezelfde inhoud.