



# OpenMRS Communicatiemodule

## Context

De platformen voor berichtenverkeer verschillen sterk per regio. In China worden bijvoorbeeld andere systemen gebruikt (zoals Baidu) dan in Europa (zoals WhatsApp). Door het ontbreken van standaardisatie is voor elk platform een aparte koppeling nodig. WhatsApp en Signal hebben bijvoorbeeld verschillende API's.

De opdrachtgever wil de communicatiemodule als zelfstandig product aanbieden aan OpenMRS-organisaties wereldwijd. Door het als SaaS te positioneren kunnen organisaties zich abonneren op de dienst zonder zelf infrastructuur te beheren, wat de adoptie vereenvoudigt.

## Doelstelling

Ontwerp en implementeer een Software-as-a-Service communicatiemodule die notificaties kan versturen voor OpenMRS organisaties via externe messaging providers. Gebruik hiervoor de fictieve messaging providers: SwiftSend, LegacyLink, AsyncFlow en SecurePost. De module dient te voldoen aan de HL7-standaarden volgens de FHIR-specificatie. De communicatiemodule dient configureerbaar en uitbreidbaar zijn, zodat OpenMRS-organisaties hun eigen abonnementen en services kunnen integreren. Daarnaast dient het ontwerp toekomstbestendig te zijn, zodat nieuwe communicatieproviders eenvoudig kunnen worden toegevoegd.

## Functionele eisen

1. Als een patiënt van een ziekenhuis wil ik een bericht op mijn telefoon ontvangen met de details van mijn afspraak (tijd, locatie, en eventuele voorbereidingen), zodat ik mijn ziekenhuisbezoek goed kan voorbereiden en op tijd kan verschijnen.
  - De notificatie wordt
    - 24 uur voor de afspraak verzonden
    - 1 uur voor de afspraak verzonden
  - De notificatie bevat:
    - Datum en tijd van de afspraak.
    - Locatie (bijvoorbeeld polikliniek en kamer).



- Eventuele specifieke instructies (bijvoorbeeld nuchter blijven of medicijnen meenemen).
  - Voor afspraken die reeds zijn aangevangen worden geen notificaties verstuurd.
  - Wanneer een afspraak binnen OpenMRS wordt geannuleerd of gewijzigd worden er geen notificaties meer verzonden of de tijdstippen waarop ze worden verzonden aangepast.
2. De communicatiemodule legt vast of de notificatie succesvol is verzonden zodat later overzichten kunnen worden gegenereerd welke notificaties er namens welke organisatie zijn verstuurd door welke messaging provider. Dit moet het controleren van facturen die de messaging providers sturen eenvoudiger maken.
  3. Als OpenMRS organisatie wil ik dat de communicatiemodule gebruik maakt van één van de ondersteunde messaging providers om berichten naar mijn patiënten te versturen.

### **Niet-functionele eisen**

1. De communicatiemodule dient zelfstandig te kunnen functioneren en integreren met meerdere OpenMRS-instanties. Hierdoor kunnen verschillende ziekenhuizen de module gebruiken op basis van de eigen abonnementen met messaging providers.
2. De integratie tussen de OpenMRS-instantie en de communicatiemodule is passend bij de doelstelling. De integratie is gedocumenteerd voor technische OpenMRS beheerders en beveiligd volgens de best practises die passen bij de gekozen integratie-oplossing.
3. Organisaties die gebruik willen maken van de communicatiemodule dienen gebruik te kunnen maken van één van de volgende messagingproviders die alle 4 door de communicatiemodule worden ondersteund:
  - SwiftSend
  - LegacyLink
  - AsyncFlow



- SecurePost
4. De communicatiemodule dient gekoppeld te kunnen worden aan het OpenMRS platform vanaf versie 2.7.x
  5. Gevoelige informatie over abonnementen en platformen van organisaties dienen veilig te worden opgeslagen. Bij ongevoegde toegang mag deze informatie niet bruikbaar zijn. Authenticatiegegevens voor externe messaging providers worden niet in code of configuratiebestanden opgeslagen. Alle gevoelige gegevens (zoals credentials, tokens en berichteninhoud) dienen te worden versleuteld met minimaal AES-256 voor opslag en TLS 1.3 voor transport. De communicatiemodule mag gevoelige data verwerken, maar mag deze nooit onbeveiligd opslaan, inclusief logbestanden.
  6. De module dient zich te houden aan de HL7-standaarden. HL7-systemen ondersteunen onder andere:
    - Berichtontvangst en validatie (controle op structuur, verplichte velden en syntaxis)
    - Acknowledgements (ACK) voor ontvangstbevestiging of foutmeldingen
    - Logging en tracking van berichten voor audit en troubleshooting
    - Berichttransformatie (mapping tussen HL7-versies of lokale formaten)
    - Queueing en retry-mechanismen bij netwerkproblemen
  7. De communicatiemodule dient te worden gerealiseerd als zelfstandig proces en dient in hoge mate onafhankelijk te zijn van andere systemen. Downtime bij communicatieproviders of OpenMRS instanties dient te worden opgevangen door een zelfontworpen en gedocumenteerde fallback- of retrymechanisme.
  8. De communicatiemodule dient in staat te zijn om berichten te kunnen verwerken in diverse karaktersets.
  9. De werking van de communicatiemodule dient volledig inzichtelijk te zijn via geschikte monitoringtooling, bijvoorbeeld op basis van OpenTelemetry. Er dient een real-time dashboard beschikbaar zijn waarop OpenMRS beheerders de status van berichten, prestaties (throughput) en eventuele foutmeldingen kunnen volgen om het systeem te bewaken.



10. De communicatiemodule verwijdert patiënt- en gerelateerde gegevens automatisch binnen 14 dagen na afhandeling van de communicatie.
11. De communicatiemodule bewaart maximaal een jaar meta-informatie van verstuurd berichten voor traceerbaarheidsdoeleinden. Deze meta-informatie bevat geen direct identificeerbare patiëntgegevens of afspraakgegevens, maar bevat voldoende informatie zodat facturen van messaging providers kunnen worden gecontroleerd.
12. De communicatiemodule dient zodanig ontworpen te zijn dat andere functionele OpenMRS modules - zoals modules voor medische testresultaten – kunnen worden geïntegreerd.
13. De communicatiemodule dient diverse tijdzones te ondersteunen, zodat alle notificaties en tijdstippen waarop ze worden verstuurd rekening houden met de lokale tijdzone van de betreffende OpenMRS-organisatie.

### **Deliverables**

Als groep lever je op:

- Documentatie voor technisch beheerders van een OpenMRS-organisatie, waarin wordt beschreven welke stappen en maatregelen nodig zijn om de koppeling tussen de systemen te realiseren, inclusief de bijbehorende aandachtspunten.
- Een codebase die kan worden gestart in een Docker omgeving inclusief de instructie en voorbeeld commando om de oplossing op te starten en een voorbeeld request die de oplossing in werking zet.
- **Architectural Decision Record**-logboek (in de code repo) waarin je de belangrijkste architectuurbeslissingen vastlegt: wat het probleem is, welke opties je hebt overwogen, en waarom je voor een bepaalde aanpak hebt gekozen. Aanvullend lever je een visualisatie van de communicatiemodule tot op applicatiearchitectuur niveau (C4 levels 1, 2 en 3) en een proces visualisatie over het proces door het systeem loopt.
- Realisatielogboek met daarin:
  - Overzicht gebruikte ontwikkeltools (IDE's)



- Overzicht gebruikte AI-tools (indien van toepassing) inclusief enkele representatieve voorbeelden van hoe ze zijn ingezet (prompts, screenshots, logs).
- Overzicht van de commits per teamlid (gelieve ook een link mee te leveren).
- Testrapportage die de betrouwbaarheid en de uitbreidbaarheid aantonen.