Netwerkprotocollen: remote healthcare

Projectgroep A4 periode 2.1

# Inhoudsopgave

[Inhoudsopgave 1](#_Toc1451968097)

[Introductie 2](#_Toc531416333)

[Server-Client communicatie 3](#_Toc1095588936)

[State-Machine Client 4](#_Toc583213735)

[State-Machine Client 5](#_Toc1978682693)

[Berichten 6](#_Toc1636124048)

[Server 7](#_Toc1034540217)

[Voorbeeld 9](#_Toc1510538098)

[Server-Doctor Communicatie 10](#_Toc233073801)

[State-Machine Server 11](#_Toc253106317)

[State-Machine dokter 12](#_Toc760426087)

[Berichten 14](#_Toc1448329675)

[Server 15](#_Toc1906154745)

[Dokter: 17](#_Toc1982761164)

[Voorbeeld: 19](#_Toc736704032)

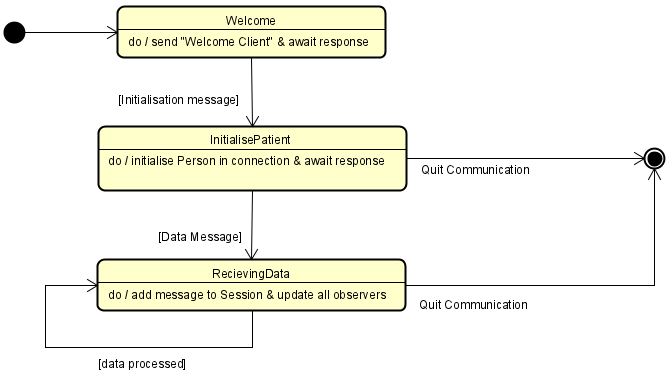
# Introductie

In dit document wordt er omschreven hoe het netwerkprotocol voor de remote Healthcare applicaties in elkaar steekt. Het protocol is tekst-gebaseerd en zal tijdens verzending worden versleuteld via RSA-encryptie en bij ontvangst worden.

# **Server-Client communicatie**

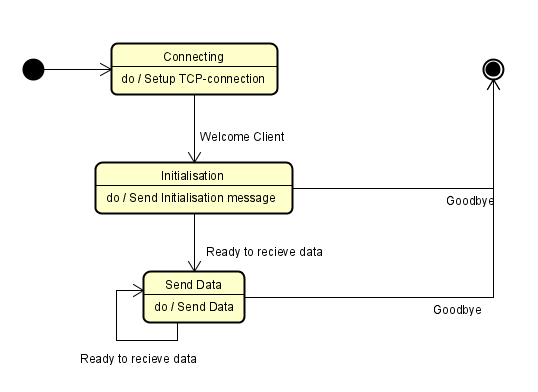
## State-Machine Server

|  |  |  |  |
| --- | --- | --- | --- |
| Server State | Invoer | Server Actie | Volgende Server State |
| Welcome Client | Geen invoer | Stuurt:  "Welcome Client" | Initialise Patient |
| Initialise Patient | [Initialisatie bericht] | Initialiseren Patiënt in Cliënt & stuurt “Ready to recieve data" | Recieving Data |
| Quit Communication | Stuurt "Goodbye" | Exit |
| Recieving Data | [Data bericht] | Data wordt toegevoegd aan sessie en alle observers geüpdatet & stuurt  "Ready to recieve data" | Blijft in State |
| Quit Communication | Stuurt "Goodbye" | Exit |
| Exit | Geen invoer | Sluit verbinding met Client | Geen volgende State |



## **State-Machine Client**

|  |  |  |  |
| --- | --- | --- | --- |
| Cliënt State | Invoer | Cliënt actie | Volgende Cliënt State |
| Connecting | Geen invoer | [Zet TCP-verbinding op met Server] | Initialisation |
| Initialisation | Welcome Client | [Initialisatie bericht] | SendData |
| Quit Connection | Exit |
| SendData | Ready to recieve data | [Data bericht] | Blijft in State |
| Quit Connection | Exit |
| Exit | Goodbye | Client stopt applicatie | Geen volgende State |



## Berichten

### Server

Welcome Client:

In de welcome State stuurt de Server een bericht om aan de Client te laten weten dat het opzetten van de verbinding is gelukt, de server verwacht hier dus ook geen invoer. Het bericht wat gestuurd wordt is als volgt:

Welcome Client

Initialise Patient:

In de Initialise patient state ontvangt het initialisatie bericht van de Cliënt die aangeeft welke patient er verbonden is met de server, met welke ergo- en hartslagmeter er verbinding is gemaakt en de datum met tijd van wanneer de cliënt zijn eerste bericht heeft gestuurd. Zodra deze data verwerkt is wordt het volgende bericht gestuurd om aan te geven dat actuele data ontvangen kan worden:

Ready to recieve Data

Recieving Data:

In de Recieving Data state ontvangt de server huidige fiets en hartslag data in combinatie met de datum en tijd waarop de metingen zijn gedaan. Zodra de ontvangen data zijn verwerkt en de server klaar is om data te ontvangen zal het volgende bericht worden gestuurd:

Ready To Recieve Data

Exit:

De exit state is eigenlijk geen echte “state”, zodra de server het bericht "Quit Communication" ontvangt zal het een eindigend bericht sturen dat aangeeft dat het verbreken van de verbinding is binnengekomen. Nadat hij dit bericht heeft verstuurd zal de server de verbinding verbreken. Het bericht wat verstuurd wordt na het ontvangen van "Quit Communication" is:

Goodbye

Client  
Connecting:

De Connecting State is de eerste state van de client. In deze state wordt de verbinding met de server opgezet en wordt er gewacht op het bericht "Welcome client". In deze staat worden er dus ook geen berichten verstuurd.

Initialisation:

De initialisation state is verantwoordelijk voor het versturen van de initialisatie data van de server. Dit bericht bevat de naam van de patient, verbonden hartslag- en ergometer en de datum en tijd van het eerste bericht. Dit bericht is een JSON bericht dat op de volgende manier wordt geformatteerd (het volgende bericht is in de realiteit één regel):

{

“ClientName”: (naam patient, String),

“ConnectedErgometer”: (verbonden ergometer, String),

"ConnectedHeartRateMonitor": (verbonden hartslagmeter, String),

"dateTime": (datum en tijd van verbinding bericht, )

}

Sending Data:

In de sending data state wordt de huidige fiets en ergometer data door gestuurd naar de server samen met de tijd dat deze metingen verzonden worden. De sending state zal nooit de data door sturen zonder het bericht "Ready to recieve Data" te ontvangen. Het bericht is een JSON bericht die op de volgende manier wordt geformatteerd (het volgende bericht is in de realiteit één bericht):

{

"Bicycle speed": (huidige fiets snelheid),

"Heart rate": (huidige hartslag),

"Date Time": (huidige datum en tijd van verzending)

}

Sluiten van verbinding:

De verbinding kan tijdens elke staat waar een bericht verstuurd kan worden geëindigd worden door het volgende bericht te sturen:

Quit Connection

Exit:

De Exit state is in tegenstelling tot de Exit state van de server wél een echte state. In deze state wordt er gewacht op het sluit bericht van de server. Zodra deze het sluit bericht heeft ontvangen sluit de client zichzelf af. Er wordt in deze state dus ook geen bericht meer gestuurd.

## Voorbeeld

S: Welcome Client

C: {

C: “ClientName”: "David",

C: “ConnectedErgometer”: "Tacx Flux 0123",

C: "ConnectedHeartRateMonitor": "Decathlon Dual HR",

C: "dateTime": "2024-01-01T00:00:00.0000000+02:00"

C: }

S: Ready to recieve data

C: {

C: "Bicycle speed": 20,

C: "Heart rate": 80,

C: "Date Time": "2024-01-01T00:00:00.0000000+02:00"

C: }

S: Ready to recieve data

C: {

C: "Bicycle speed": 20,

C: "Heart rate": 80,

C: "Date Time": "2024-01-01T00:00:00.0000000+02:00"

C: }

S: Ready to recieve data

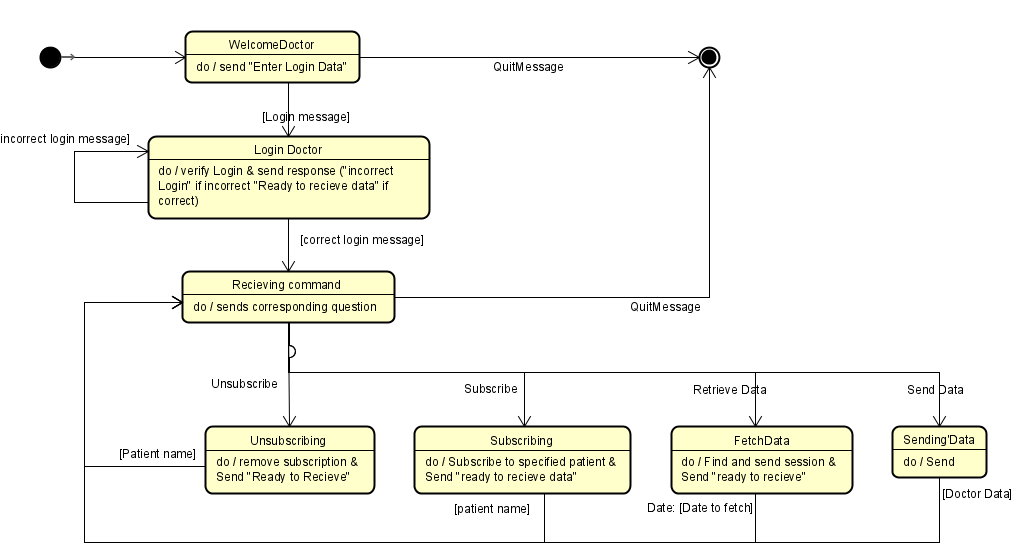
C: Quit communication

S: Goodbye

# **Server-Doctor Communicatie**

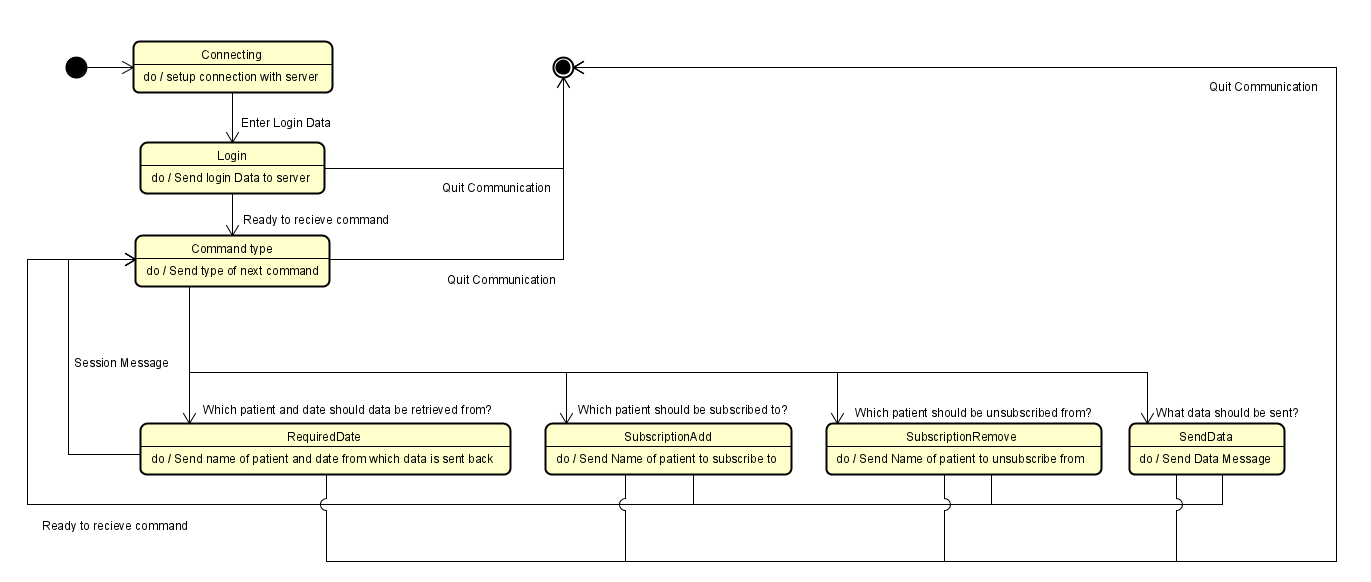
## State-Machine Server

|  |  |  |  |
| --- | --- | --- | --- |
| Server State | Invoer | Server actie | Volgende State |
| Welcome Doctor | Geen invoer | Enter Login Data | Login Doctor |
| Login Doctor | (Login bericht) | [Bericht geaccepteerd]  Ready to recieve data | Recieving Command |
| [Bericht geweigerd]  Incorrect Login | Geen volgende State |
| Quit Communication | Goodbye | Exit |
| Recieving Command | Retrieve Data | Which Date should data be retrieved from? | Fetching Data |
| Subscribe | Which patient should be subscibed to?  (+ Lijst met huidige clients) | Subscribing |
| Unsubscribe | From Who do you want to stop recieving data  (+ Lijst met huidige subscriptions) | Unsubscribing |
| Send Data | What data do you want to send | Sending Data |
| Start Session | Whicht Patient Should session start (+lijst van patients) | StartingSession |
| End Session | Whicht Patient Should session end (+lijst van patients) | EndingSession |
| Quit Communication | Goodbye | Exit |
| Fetching Data | Date: (datum van oude sessie) | [opzoeken en terugsturen sessie met die datum]  Ready to recieve data | Recieving Command |
| Quit Communication | Goodbye | Exit |
| Subscribing | (patient naam) | [Voegt zichzelf toe aan lijst met observers van client]  Ready to recieve data | Recieving Command |
| Quit Communication | Goodbye | Exit |
| Unsubscribing | (patient naam) | [Verwijderd zichzelf uit observers lijst van client]  Ready to recieve data | Recieving Command |
| Quit Communication | Goodbye | Exit |
| Sending Data | (Weerstand bericht) | [Voegt bericht toe aan lijst en update observers]  Ready to recieve | Recieving Command |
| Quit Communication | Goodbye | Exit |
| StartingSession | (patient naam) | [Maakt een nieuwe currentSession aan]  Ready to recieve | Recieving Command |
| Quit Communication | Goodbye | Exit |
| EndingSession | (patient naam) | [eindigd currentsession]  Ready to recieve | Recieving Command |
| Quit Communication | Goodbye | Exit |
| Exit | Geen invoer | Sluit verbinding | Geen volgende State |



### State-Machine dokter

|  |  |  |  |
| --- | --- | --- | --- |
| Server State | Invoer | Dokter actie | Volgende State |
| Connecting | (Geen invoer) | [TCP-verbinding opzetten met server] | Login |
| Login | Enter Login Data | (Dokter inlog bericht) | Geen volgende state |
| Login Successful | Geen actie | Command Type |
| Incorrect Login | (Dokter inlog bericht) | Geen volgende state |
| (Elke invoer) | Quit communication | Exit |
| Command Type | Ready to recieve Command | Retrieve Data | Required Data |
| Subscribe | Subscription Add |
| Unsubscribe | Subscription Remove |
| Send Data | Send Data |
| Start Session | StartingSesion |
| End Session | EndingSession |
| Quit communication | Exit |
| Required Date | Which patient and date should data be retrieved from? | (Data opvragen bericht) | Geen volgende State |
| (Opgevraagde Sessie) | Geen actie | Command Type |
| (Elke invoer) | Quit communication | Exit |
| Subscription Add | Which patient should be subscribed to? (+ lijst met huidige patiënten) | (Naam patient) | Command Type |
| Quit communication | Exit |
| Subscription Remove | Which patient should be unsubscribed from? (+ lijst met huidige abonnees) | (Naam patient) | Command Type |
| Quit communication | Exit |
| StartSession | Whicht Patient Should session start (+lijst van patients) | (Naam patient) | Command Type |
| Quit communication | Exit |
| EndSession | Whicht Patient Should session end (+lijst van patients) | (Naam patient) | Command Type |
| Quit communication | Exit |
| Send Data | What data should be sent? | (Data bericht) | Command Type |
| Quit Communication | Exit |
| Starting Session | Whicht Patient Should session start (+lijst van patients) | (Naam patient) | Command Type |
| Quit Communication | Exit |
| Ending Session | Whicht Patient Should session end (+lijst van patients) | (Naam patient) | Command Type |
| Quit Communication | Exit |
| Exit | Goodbye | Client Stopt communicatie | Geen volgende state |



## Berichten

### Server

Welcome Doctor:

In de Welcome doctor state wordt er een bericht gestuurd naar de Doctor-applicatie om te laten weten dat de connectie correct is opgezet, nadat het bericht is verstuurd gaat het zonder reactie meteen door naar de Login State. Dit doet het door het volgende bericht te sturen:

Enter Login Data

Login Doctor:

In login Doctor worden er 2 verschillende type berichten gestuurd gebaseerd op of de Login Data van die de server ontvangt. In het geval dat de login data van de doctor correct is wordt het volgende bericht gestuurd:

Login successful gevolgt door het bericht: Ready to Recieve Command

In het geval dat de inlog gegevens van de doctor niet correct zijn wordt het volgende bericht gestuurd en wordt er niet door gegaan naar de volgende state:

Incorrect Login

Recieving Command:

In de Recieving command state wordt gewacht op een van de volgende berichten:

* Retrieve Data
* Subscribe
* Unsubscribe
* Send Data

Als de Server een van deze commando's heeft ontvangen wordt er gebaseerd op het antwoord naar de juiste staat gegaan en een van de volgende bericht gestuurd naar de Doctor:

* Which patient and date should data be retrieved from? (Fetching Data)
* {

“Message”: “Which patient should be subscribed to?”,

"Patients": [(array met alle patiënten)]

} (Subscribing)

* {

“Message”: “Which patient should be unsubscribed from?”,

"Patients": [(array met alle observers)]

} (Unsubscribing)

* What data should be sent? (Sending Data)

Fetching data:

In de fetching data State wordt data van een gegeven Datum opgezocht en doorgestuurd de state wacht op een Naam en Date Time, zodra deze ontvangen is wordt de sessie met die begindatum opgehaald in een JSON bericht met de volgende formattering (in de realiteit is dit één regel):

{

"Client Messages": [(array met berichten van client, String)],

"Doctor Messages": [(array met berichten van dokter, String)]

}

gevolgd door het bericht: Ready to recieve command

Subscribing:

In de subscribing state wordt er gewacht op de naam van een patient. Zodra deze is ontvangen wordt de cliënt als observer toegevoegd aan de lijst met observers van de doctor zodat deze berichten kan versturen naar de Cliënt en andersom zodat de dokter berichten van de cliënt ontvangt. Zodra dit is gebeurd zal de server aangeven dat het weer klaar is om commando's te ontvangen met het volgende bericht:

Ready to recieve command

Unsubscribing:

In de unsunscribing state wordt er net als de subscribing state gewacht op de naam een patient. Vervolgens wordt deze patient opgezocht uit de lijst met abonnees en verwijderd uit de lijst van observers. Vervolgens zal de server een bericht sturen om aan te geven dat het klaar is om een bericht te ontvangen. Dit bericht gaat als volgt:

Ready to recieve command

Sending Data:

In de sending data state wordt er gewacht op het data bericht van de dokter (zie dokter berichten). Zodra dit bericht is ontvangen voegt de server dit bericht toe aan de lijst van dokter berichten en worden alle observers genotificeerd. Vervolgens zal het volgende bericht worden verstuurd naar de dokter:

Ready to recieve command

Exit:

De exit state is net als de exit state van de Server communicatie met client niet een echte state. Er wordt niet meer gewacht op vervolg berichten. Zodra er in elk gegeven state het bericht "Quit Communication" wordt ontvangen sluit de server de verbinding nadat het volgende bericht wordt verstuurd:

Goodbye

### Dokter:

Connecting:

In de connecting state wordt de Tcp-verbinding opgezet en gewacht op het eerste bericht van de server. Deze staat stuurd zelf dus ook geen berichten.

Login:

In de Login state wordt het inlog bericht van de dokter verstuurd naar de Server. Vervolgens wordt er in deze state gewacht op een "reactie" van de server, bij "Incorrect Login" wordt een nieuw bericht gestuurd met de juiste login, bij "Login Successful" wordt er geen bericht verder gestuurd en gewacht op het bericht dat de volgende State aangeeft. Het inlog bericht is als volgt (in de realiteit is dit één regel):

{

"DoctorID": (docter IP, integer)

"DoctorName": (docter naam, Sting)

"DoctorPassword": (wachtwoord, String)

}

CommandType:

In de command Type state geeft de dokter aan wat voor type data of handeling de dokter wil uitvoeren. Nadat dit bericht is gestuurd en het bijpassende bericht is ontvangen gaat de dokter door naar de bijpassende state. De berichten die de docter kan sturen samen met hun bijpassende state zijn:

* Retrieve Data (Required Date)
* Subscribe (Subscription Add)
* Unsubscribe (Subscription Remove)
* Send Data (Send Data)

Required Date:

In de required date state wordt de naam van de patient en datum verstuurd van de Sessie die de dokter op wil vragen. De inhoud van het bericht is als volgt (in de realiteit is dit één regel):

{

"PatientName": (naam patient, String)

"SessionDate": (begintijd sessie, DateTime)

}

Subscription Add:

In de subscription add state wordt er doorgegeven aan de server naar welke patient geluisterd moet worden. Het bericht is erg eenvoudig en gaat als volgt:

(Naam patient)

Subscription Remove:

In de subscription remove state wordt doorgegeven van welke patient het abonnement aan data moet stoppen. Het bericht is hetzelfde als die van subscription add, namelijk:

(Naam patient)

Send Data:

In de Send data State wordt er een bericht gestuurd waar een bericht in staat die ontvangen moet worden door de patient en/of een aangepaste weerstand waarde van de de fiets. Het bericht is een JSON-String dat op de volgende manier is geformatteerd:

{

"PatientName": (naam van patient, String)

"Message": (bericht naar patient, String)

"NewResistance": (nieuwe weerstand waarde, Integer)

}

Sluiten van verbinding:

De verbinding kan tijdens elke staat waar een bericht verstuurd kan worden geëindigd worden door het volgende bericht te sturen:

Quit Connection

Exit:

De Exit state is net als de Client Exit State. In deze state wordt er gewacht op het sluit bericht van de server. Zodra deze het sluit bericht heeft ontvangen sluit de client zichzelf af. Er wordt in deze state dus ook geen bericht meer gestuurd.

## Voorbeeld:

S: Enter Login Data

D: {

D: "DoctorID": 54321

D: "DoctorName": "Friddo"

D: "DoctorPassword": "AvansDoctor"

D: }

S: Incorrect Login

D: {

D: "DoctorID": 12345

D: "DoctorName": "Frido"

D: "DoctorPassword": "AvansDokter"

D: }

S: Login Successful

S: Ready to recieve command

D: Retrieve data

S: which patient and date should data be retrieved from?

D: {

D: "PatientName": (naam patient, String)

D: "SessionDate": (begintijd sessie, DateTime)

D: }

S: {

S: "Client Messages": [“message 1”, “message 2”, “message 3”],

S: "Doctor Messages": [“message 1”, “message 2”, “message 3”]

S: }

S: Ready to recieve command

D: Subscribe

S: {

S: “Message”: “Which patient should be subscribed to?”,

S: "Patients": [David, Siem, Fedde, Tiemen, Mohammed]

S: }

D: David

S: Ready to recieve command

D: Unsubscribe

S: {

S: “Message”: “Which patient should be unsubscribed from?”,

S: "Patients": [David, Siem]

S: }

D: Siem

S: Ready to recieve command

D: Send data

S: What data should be sent?

D: {

D: "PatientName": "David",

D: "Message": "Try to pick up the pace, I have increased the resistance",

D: "NewResistance": 60

D: }

S: Ready to recieve Command

D: Quit Communication

S: Goodbye