Politecnico di Torino

# Smart Pregnancy

*a new way to live your pregnancy*


Team members:

289512 Aime Elisa
292257 Codagnone Giulia
284277 Cubeddu Laura
282133 De Luca Antonio

# The problem

Not keeping track of vital parameters could cause **difficulties**

Late intervention

It's more complicated to understand the causes of a possible disorder

Keep track of pregnancy complications recording and analyzing data from **sensors**

Remote patient monitoring

Data storage and visualization

- Temperature
- Heart rate
- Pressure
- Glycemia

# The problem

Not keeping track of vital parameters could cause **difficulties**

- Late intervention
- It's more complicated to understand the causes of a possible disorder

Keep track of pregnancy complications recording and analyzing data from **sensors**

- Remote patient monitoring
- Data storage and visualization

- Temperature
- Heart rate
- Pressure
- Glycemia

# The problem

Not keeping track of vital parameters could cause difficulties

Late intervention

It's more complicated to understand the causes of a possible disorder

Keep track of pregnancy complications recording and analyzing data from sensors

Remote patient monitoring

Data storage and visualization

- Temperature
- Heart rate
- Pressure
- Glycemia

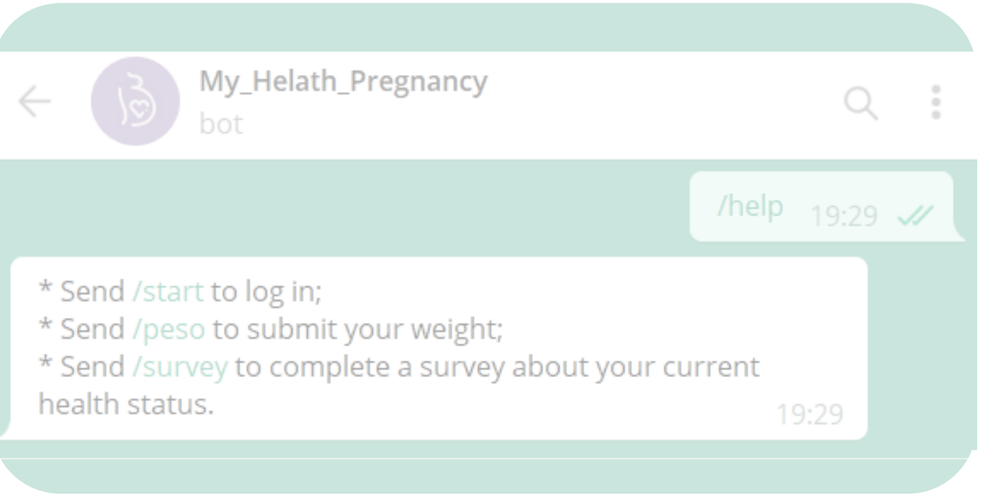
# The flow

## for Doctor

- Allow doctor registration
- Allow  patient registration
- Allow to manage anomalous data
- Allow to always be updated on patient's heart rate, pressure, temperature and glycemia


## for Patient

- Allow to sign in
- Allow to submit weight
- Allow to complete a health survey



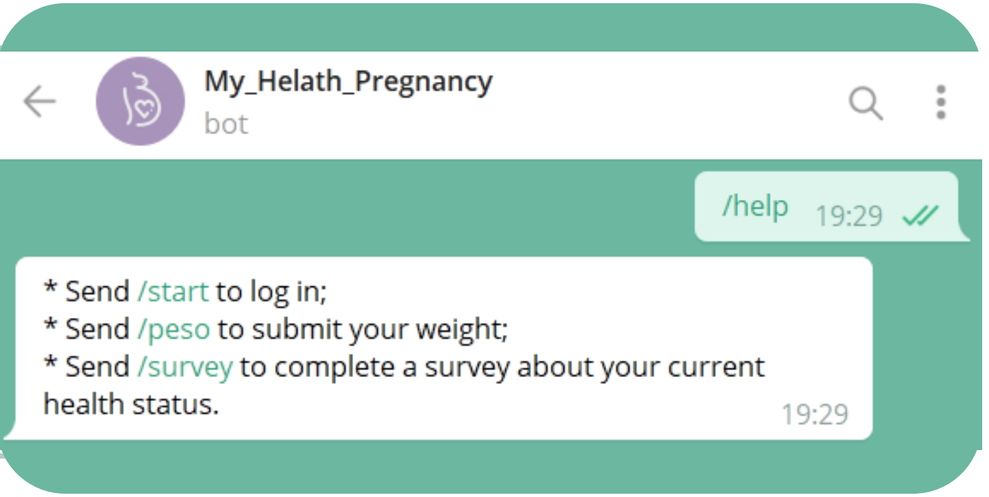
# The flow

## for Doctor

- Allow doctor registration
- Allow  patient registration
- Allow to manage anomalous data
- Allow to always be updated on patient's heart rate, pressure, temperature and glycemia


## for Patient

- Allow to sign in
- Allow to submit weight
- Allow to complete a health survey



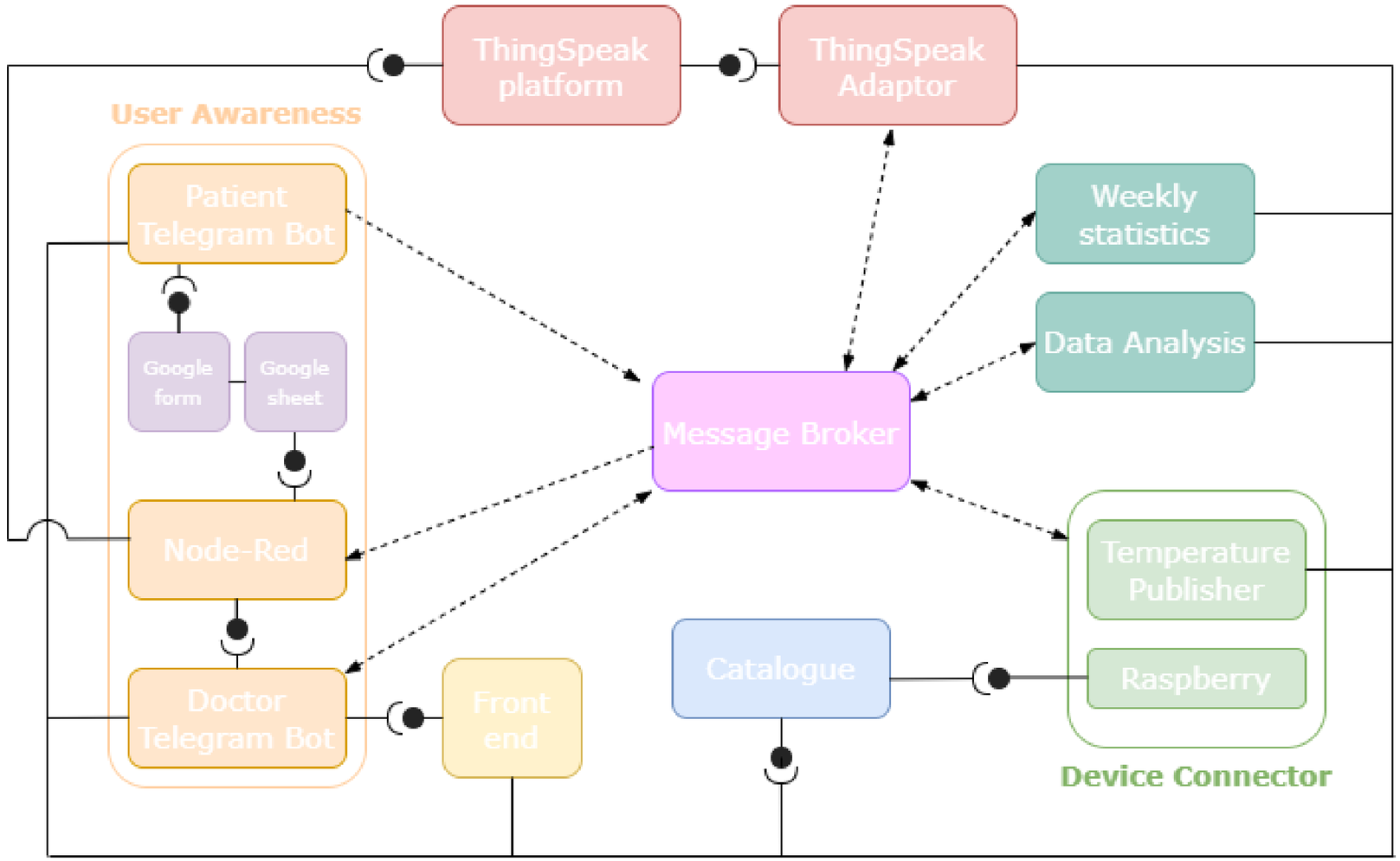
# Block Diagram - proposal

Legend

MQTT Comunication

REST Web Services (**consumer**)

REST Web Services (**provider**)



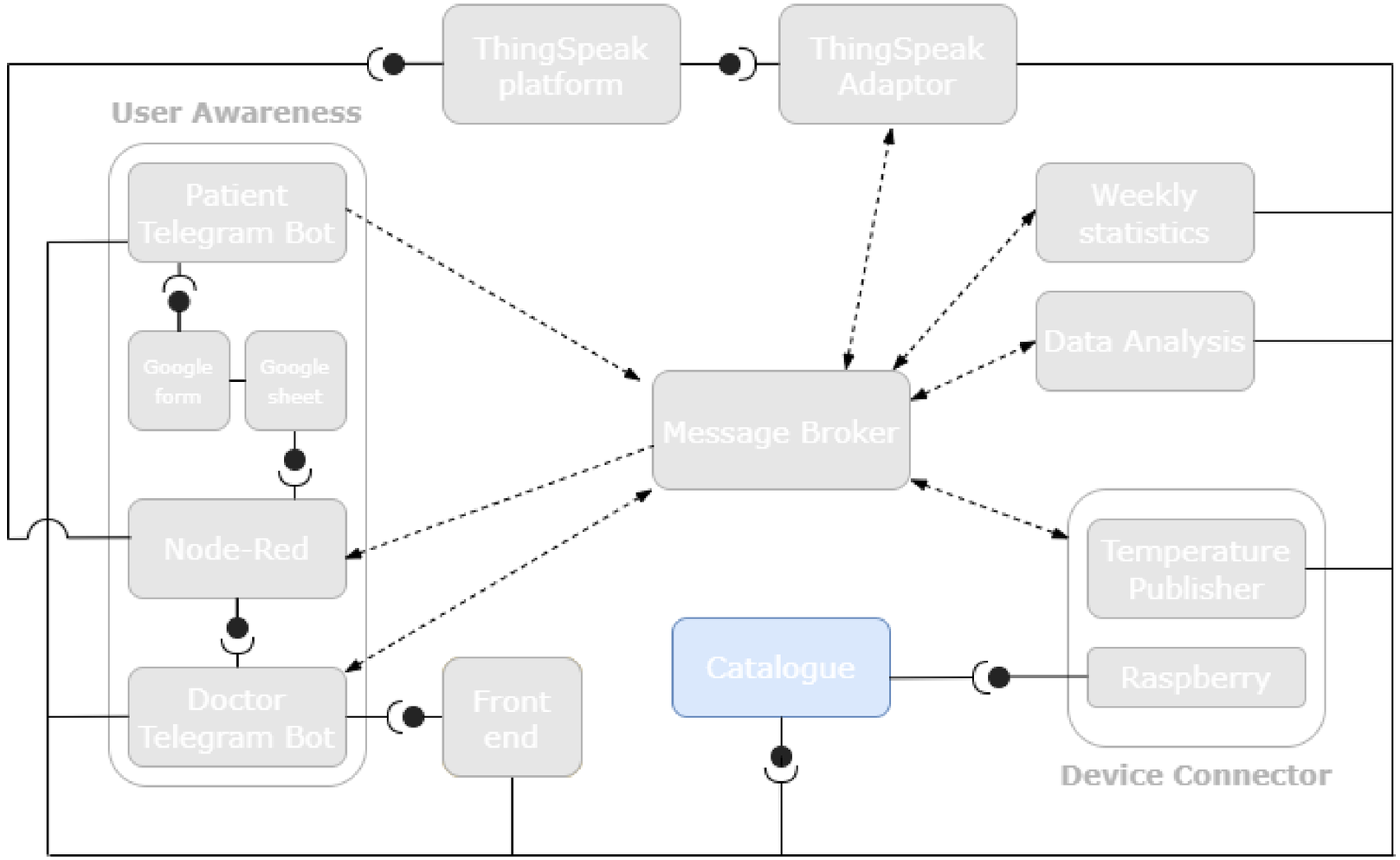
# Block Diagram - catalog

Legend

MQTT Comunication

REST Web Services (**consumer**)

REST Web Services (**provider**)



# Catalog

A json file that communicates with all the other actors in the platform exploiting **REST communication with the catalogue host and a port**

It has a section for all the services it uses
**"services"**

It has a section for keeping track of patient and doctor ID
**"resourceState"**

It has a section for a list that includes all the doctors signed
**"resources"**


```
{
  "server_host": "Catalog",
  "server_port": 8085,
  "services": [
  "resourceState": {
    "LastPatientID": 14,
    "LastDoctorID": 10
  },
  "resources": [
    {
      "doctorID": 1,
      "doctorName": "Perry",
      "doctorSurname": "Cox",
      "doctorMail": "perrycox@scrubs.com",
      "lastUpdate": "2022-07-04",
      "connectedDevice": {
        "telegramID": 298694124
      },
      "patientList": [
      "devicesList": [
```

# Catalog

A json file that communicates with all the other actors in the platform exploiting REST communication with the catalogue host and a port

It has a section for all the services it uses
"services"

It has a section for keeping track of patient and doctor ID
"resourceState"

It has a section for a list that includes all the doctors signed
"resources"


```
{
  "server_host": "Catalog",
  "server_port": 8085,
  "services": [
  "resourceState": {
    "LastPatientID": 14,
    "LastDoctorID": 10
  },
  "resources": [
    {
      "doctorID": 1,
      "doctorName": "Perry",
      "doctorSurname": "Cox",
      "doctorMail": "perrycox@scrubs.com",
      "lastUpdate": "2022-07-04",
      "connectedDevice": {
        "telegramID": 298694124
      },
      "patientList": [
      "devicesList": [
```

# Catalog

A json file that communicates with all the other actors in the platform exploiting REST communication with the catalogue host and a port

It has a section for all the services it uses
"services"

It has a section for keeping track of patient and doctor ID
"resourceState"

It has a section for a list that includes all the doctors signed
"resources"


```
{
  "server_host": "Catalog",
  "server_port": 8085,
  "services": [
  "resourceState": {
    "LastPatientID": 14,
    "LastDoctorID": 10
  },
  "resources": [
    {
      "doctorID": 1,
      "doctorName": "Perry",
      "doctorSurname": "Cox",
      "doctorMail": "perrycox@scrubs.com",
      "lastUpdate": "2022-07-04",
      "connectedDevice": {
        "telegramID": 298694124
      },
      "patientList": [
      "devicesList": [
```

# Catalog

A json file that communicates with all the other actors in the platform exploiting REST communication with the catalogue host and a port

It has a section for all the services it uses
"services"

It has a section for keeping track of patient and doctor ID
"resourceState"

It has a section for a list that includes all the doctors signed
"resources"


```
{
  "server_host": "Catalog",
  "server_port": 8085,
  "services": [
  "resourceState": {
    "LastPatientID": 14,
    "LastDoctorID": 10
  },
  "resources": [
    {
      "doctorID": 1,
      "doctorName": "Perry",
      "doctorSurname": "Cox",
      "doctorMail": "perrycox@scrubs.com",
      "lastUpdate": "2022-07-04",
      "connectedDevice": {
        "telegramID": 298694124
      },
      "patientList": [
      "devicesList": [
```

# Catalog - resources


# Catalog - resources

```
"resources": [
  {
    "doctorID": 1,
    "doctorName": "Perry",
    "doctorSurname": "Cox",
    "doctorMail": "perrycox@scrubs.com",
    "lastUpdate": "2022-07-04",
    "connectedDevice": {
      "telegramID": 298694124
    },
    "patientList": [
    "devicesList": [
```

```
    "patientList": [
      {
        "patientID": 3,
        "patientName": "Robin",
        "patientSurname": "Scherbatsky",
        "personalData": {
          "taxIDcode": "RBNSCH",
          "userEmail": "robin@himym.com",
          "pregnancyDayOne": "2022-05-29"
        },
        "idRegistratoSuRaspberry": "no",
        "state": "attivo",
        "monitoring": "off",
        "connectedDevice": {
          "deviceName": "rpi98",
          "onlineSince": "2022-07-14",
          "telegramID": "786029508",
          "thingspeakInfo": {
            "channel": 1788970,
            "apikeys": [
              "FRN2A7XGJHIUSN24",
              "ZDNDUY17T8SJSXRM"
            ]
          }
        }
      }
    ],
    "devicesList": [
      {
        "deviceName": "rpi98",
        "patientID": 3,
        "measureType": [
          "Temperature",
          "Battito cardiaco",
          "Pressione",
          "Glicemia"
        ],
        "availableServices": [
          "raspberry_mqtt",
          "raspberry_rest"
        ],
        "activeService": "application_to_raspberry",
        "activeProtocol": "mqtt",
        "lastUpdate": "2022-07-04"
      }
    ]
  },
```

# Catalog - services

APIs follow MQTT and **HTTP paradigms.**
HTTP APIs, in compliance to CRUD methods, are further distinguished into:

**GET**

PUT

POST

DELETE


```
"services": [
  {
    "service_name": "ResourceService",
    "description": "Servizio per la registrazione di dottori e pazienti",
    "host": "Catalog",
    "port": 8085,
    "APIs": [
      {
        "functionality_name": "set_patient_in_monitoring",
        "description": "aggiorna i pazienti a stato di monitoraggio (da -1 a monitorato)",
        "access_mode": "GET",
        "uri": "set_patient_in_monitoring?patient_id={{patient_ID}}"
      },
      {
        "functionality_name": "update_chat_id",
        "description": "Aggiorna il telegram ID dopo che il paziente ha risposto al messaggio /start",
        "access_mode": "PUT",
        "uri": "update_telegram_id?patient_id={{message}}&chat_id={{chat_ID}}"
      },
      {
        "functionality_name": "doctors",
        "description": "Registrazione nel catalogo del dottore (usata da doctors.html)",
        "access_mode": "POST",
        "uri": "doctors"
      },
      {
        "functionality_name": "contolla_scadenza_week",
        "description": "controllo sulla week per eliminare paziente dal catalogo",
        "access_mode": "DELETE",
        "uri": "contolla_scadenza_week"
      }.
```

# Catalog - services

APIs follow MQTT and HTTP paradigms. HTTP APIs, in compliance to CRUD methods, are further distinguished into:

GET

PUT

POST

DELETE


```
"services": [
  {
    "service_name": "ResourceService",
    "description": "Servizio per la registrazione di dottori e pazienti",
    "host": "Catalog",
    "port": 8085,
    "APIs": [
      {
        "functionality_name": "set_patient_in_monitoring",
        "description": "aggiorna i pazienti a stato di monitoraggio (da -1 a monitorato)",
        "access_mode": "GET",
        "uri": "set_patient_in_monitoring?patient_id={{patient_ID}}"
      },
      {
        "functionality_name": "update_chat_id",
        "description": "Aggiorna il telegram ID dopo che il paziente ha risposto al messaggio /start",
        "access_mode": "PUT",
        "uri": "update_telegram_id?patient_id={{message}}&chat_id={{chat_ID}}"
      },
      {
        "functionality_name": "doctors",
        "description": "Registrazione nel catalogo del dottore (usata da doctors.html)",
        "access_mode": "POST",
        "uri": "doctors"
      },
      {
        "functionality_name": "contolla_scadenza_week",
        "description": "controllo sulla week per eliminare paziente dal catalogo",
        "access_mode": "DELETE",
        "uri": "contolla_scadenza_week"
      },
```

# Catalog - services

APIs follow MQTT and HTTP paradigms.
HTTP APIs, in compliance to CRUD methods, are further distinguished into:

GET

PUT

POST

DELETE


```
"services": [
  {
    "service_name": "ResourceService",
    "description": "Servizio per la registrazione di dottori e pazienti",
    "host": "Catalog",
    "port": 8085,
    "APIs": [
      {
        "functionality_name": "set_patient_in_monitoring",
        "description": "aggiorna i pazienti a stato di monitoraggio (da -1 a monitorato)",
        "access_mode": "GET",
        "uri": "set_patient_in_monitoring?patient_id={{patient_ID}}"
      },
      {
        "functionality_name": "update_chat_id",
        "description": "Aggiorna il telegram ID dopo che il paziente ha risposto al messaggio /start",
        "access_mode": "PUT",
        "uri": "update_telegram_id?patient_id={{message}}&chat_id={{chat_ID}}"
      },
      {
        "functionality_name": "doctors",
        "description": "Registrazione nel catalogo del dottore (usata da doctors.html)",
        "access_mode": "POST",
        "uri": "doctors"
      },
      {
        "functionality_name": "contolla_scadenza_week",
        "description": "controllo sulla week per eliminare paziente dal catalogo",
        "access_mode": "DELETE",
        "uri": "contolla_scadenza_week"
      }.
```

# Catalog - services

APIs follow MQTT and HTTP paradigms.
HTTP APIs, in compliance to CRUD methods, are further distinguished into:

GET

PUT

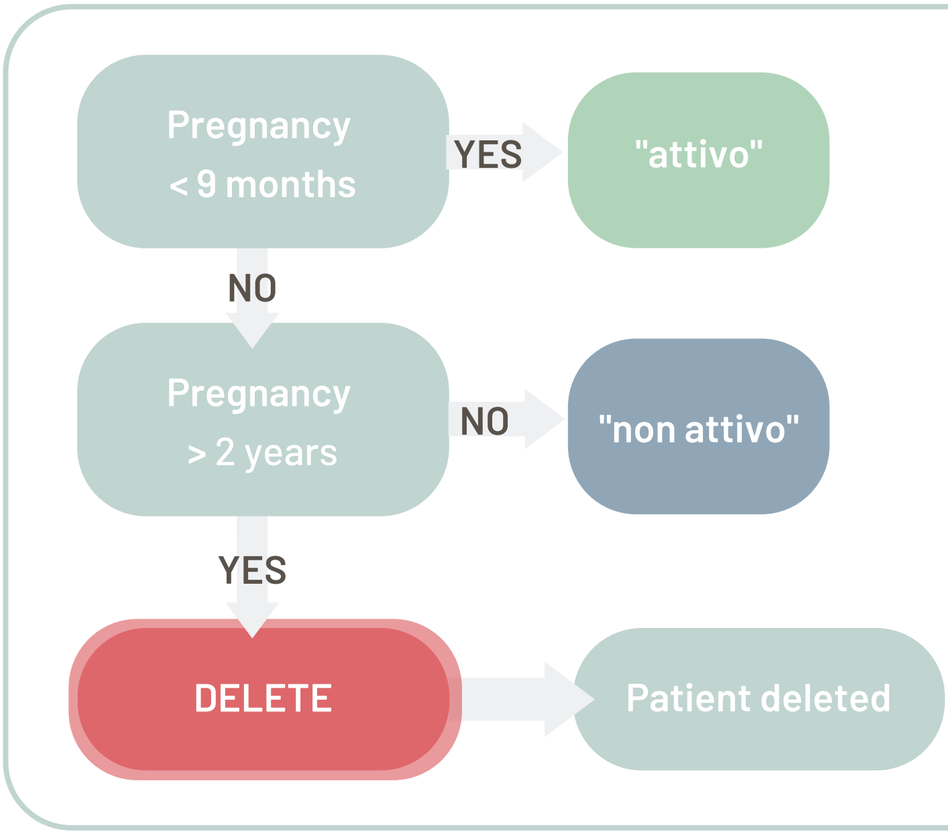
POST

**DELETE**


```
"services": [
  {
    "service_name": "ResourceService",
    "description": "Servizio per la registrazione di dottori e pazienti",
    "host": "Catalog",
    "port": 8085,
    "APIs": [
      {
        "functionality_name": "set_patient_in_monitoring",
        "description": "aggiorna i pazienti a stato di monitoraggio (da -1 a monitorato)",
        "access_mode": "GET",
        "uri": "set_patient_in_monitoring?patient_id={{patient_ID}}"
      },
      {
        "functionality_name": "update_chat_id",
        "description": "Aggiorna il telegram ID dopo che il paziente ha risposto al messaggio /start",
        "access_mode": "PUT",
        "uri": "update_telegram_id?patient_id={{message}}&chat_id={{chat_ID}}"
      },
      {
        "functionality_name": "doctors",
        "description": "Registrazione nel catalogo del dottore (usata da doctors.html)",
        "access_mode": "POST",
        "uri": "doctors"
      },
      {
        "functionality_name": "contolla_scadenza_week",
        "description": "controllo sulla week per eliminare paziente dal catalogo",
        "access_mode": "DELETE",
        "uri": "contolla_scadenza_week"
      }.
```

# Catalog - delete

## Delete mechanism





```
"patientList": [
  {
    "patientID": 3,
    "patientName": "Robin",
    "patientSurname": "Scherbatsky",
    "personalData": {
      "taxIDcode": "RBNSCH",
      "userEmail": "robin@himym.com",
      "pregnancyDayOne": "2022-05-29"
    },
    "idRegistrateSuRaspberry": "no",
    "state": "attivo",
    "monitoring": "off",
    "connectedDevice": {
      "deviceName": "rpi98",
      "onlineSince": "2022-07-14",
      "telegramID": "786029508",
      "thingspeakInfo": {
        "channel": 1788970,
        "apikeys": [
          "FRN2A7XGJHIUSN24",
          "ZDNDUY17T8SJSXRM"
        ]
      }
    }
  }
],
```

# Catalog - services

**MQTT services** with a specific Broker, a port and a base topic

MQTT APIs for weekly statistics of data

MQTT APIs to send patient's weight with telegram bot


```
{
  "service_name": "MQTT_analysis",
  "description": "Servizio utilizzato per l'invio dei dati in analisi",
  "broker": "broker.hivemq.com",
  "port": 1883,
  "base_topic": "P4IoT/SmartHealth",
  "APIs": [
    {
      "functionality_name": "weeklystats_sub",
      "description": "Invio dei file di statistiche da TS a WeeklyStats",
      "topic": "{{base_topic}}/statistics_info/{{patientID}}/{{measure}}"
    },
```

```
  "service_name": "TelegramClient",
  "patientTelegramToken": "5156513440:AAHF2KgmzHrZJah1990ukAroqTNOCAlVK-U",
  "APIs": [
    {
      "functionality_name": "send_peso",
      "description": "Invio del peso del paziente",
      "topic": "{{base_topic}}/{{patientID}}/peso"
    },
```

# Catalog - services

MQTT services with a specific Broker, a port and a base topic

MQTT APIs for weekly statistics of data

MQTT APIs to send patient's weight with telegram bot


```
{
  "service_name": "MQTT_analysis",
  "description": "Servizio utilizzato per l'invio dei dati in analisi",
  "broker": "broker.hivemq.com",
  "port": 1883,
  "base_topic": "P4IoT/SmartHealth",
  "APIs": [
    {
      "functionality_name": "weeklystats_sub",
      "description": "Invio dei file di statistiche da TS a WeeklyStats",
      "topic": "{{base_topic}}/statistics_info/{{patientID}}/{{measure}}"
    },
```

```
  "service_name": "TelegramClient",
  "patientTelegramToken": "5156513440:AAHF2KgmzHrZJah1990ukAroqTNOCAlVK-U",
  "APIs": [
    {
      "functionality_name": "send_peso",
      "description": "Invio del peso del paziente",
      "topic": "{{base_topic}}/{{patientID}}/peso"
    },
```

# Catalog - services

MQTT services with a specific Broker, a port and a base topic

MQTT APIs for weekly statistics of data

MQTT APIs to send patient's weight with telegram bot


```
{
  "service_name": "MQTT_analysis",
  "description": "Servizio utilizzato per l'invio dei dati in analisi",
  "broker": "broker.hivemq.com",
  "port": 1883,
  "base_topic": "P4IoT/SmartHealth",
  "APIs": [
    {
      "functionality_name": "weeklystats_sub",
      "description": "Invio dei file di statistiche da TS a WeeklyStats",
      "topic": "{{base_topic}}/statistics_info/{{patientID}}/{{measure}}"
    },
```

```
  "service_name": "TelegramClient",
  "patientTelegramToken": "5156513440:AAHF2KgmzHrZJah1990ukAroqTNOCAlVK-U",
  "APIs": [
    {
      "functionality_name": "send_peso",
      "description": "Invio del peso del paziente",
      "topic": "{{base_topic}}/{{patientID}}/peso"
    },
```

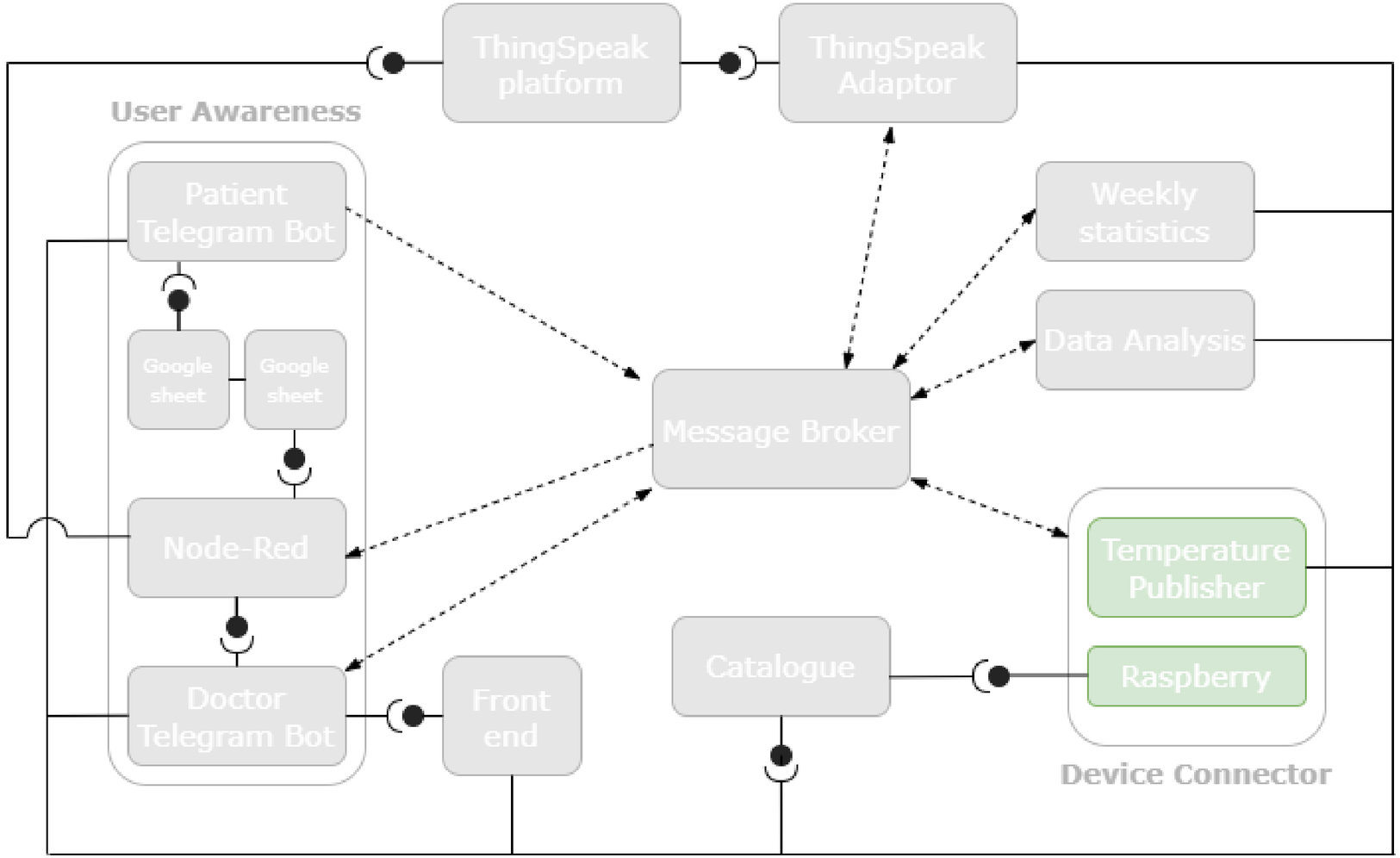
# Device Connector

## Legend

MQTT Comunication

REST Web Services (**consumer**)

REST Web Services (**provider**)



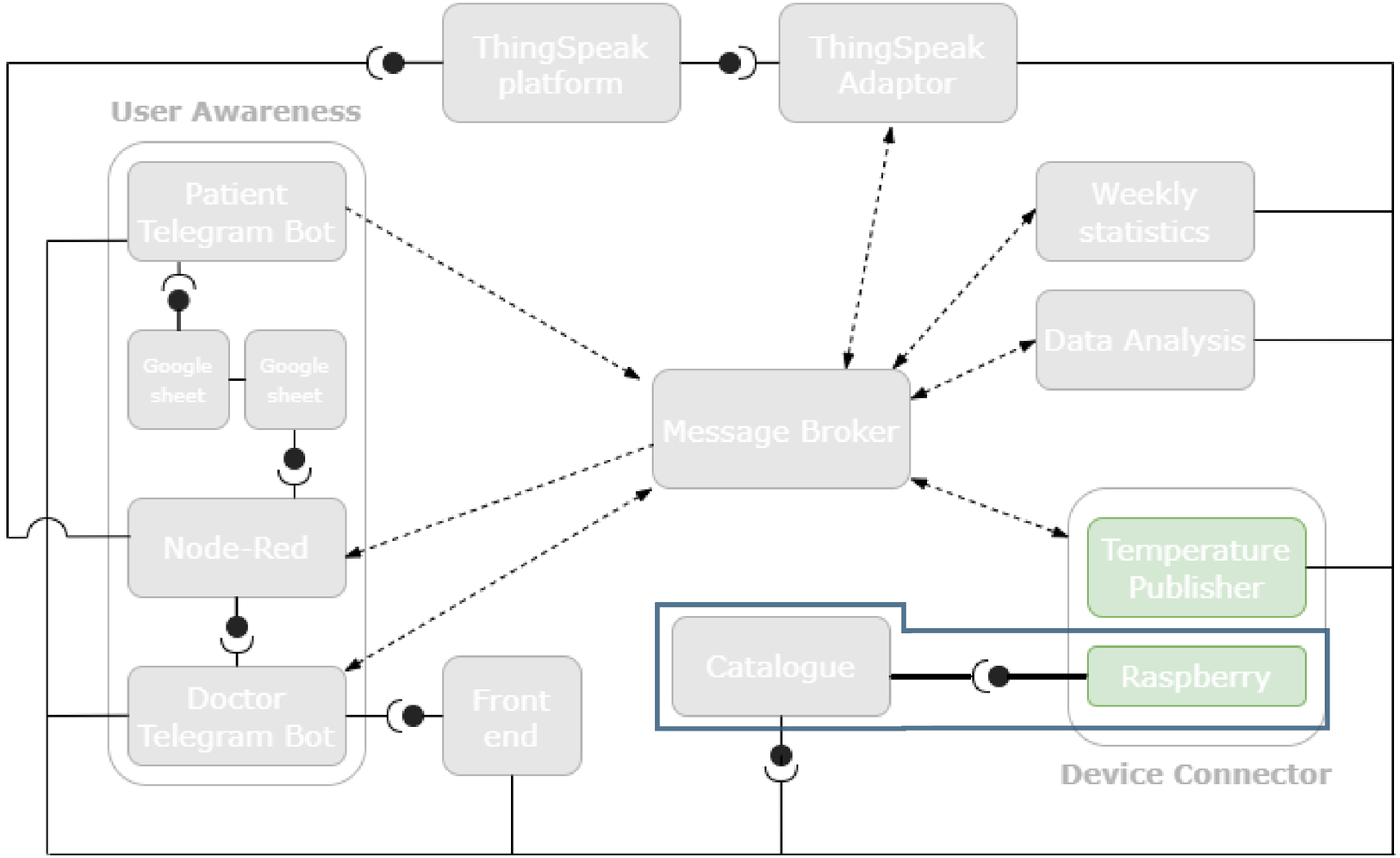
# Device Connector

Legend

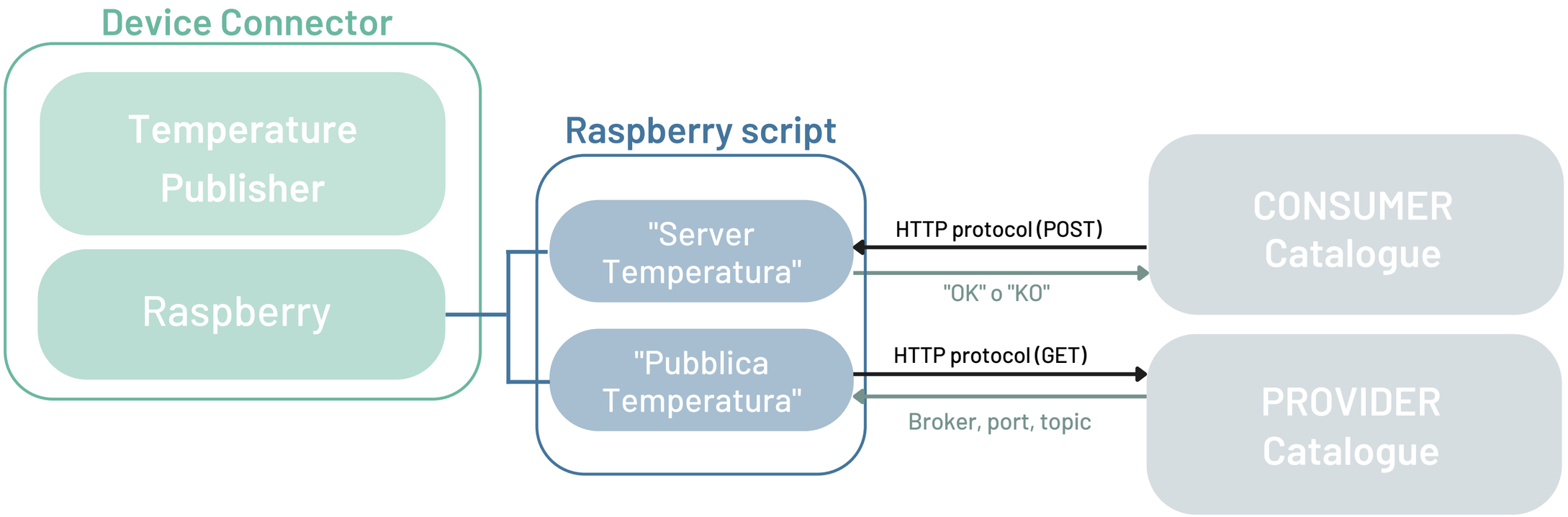
MQTT Comunication

REST Web Services (**consumer**)

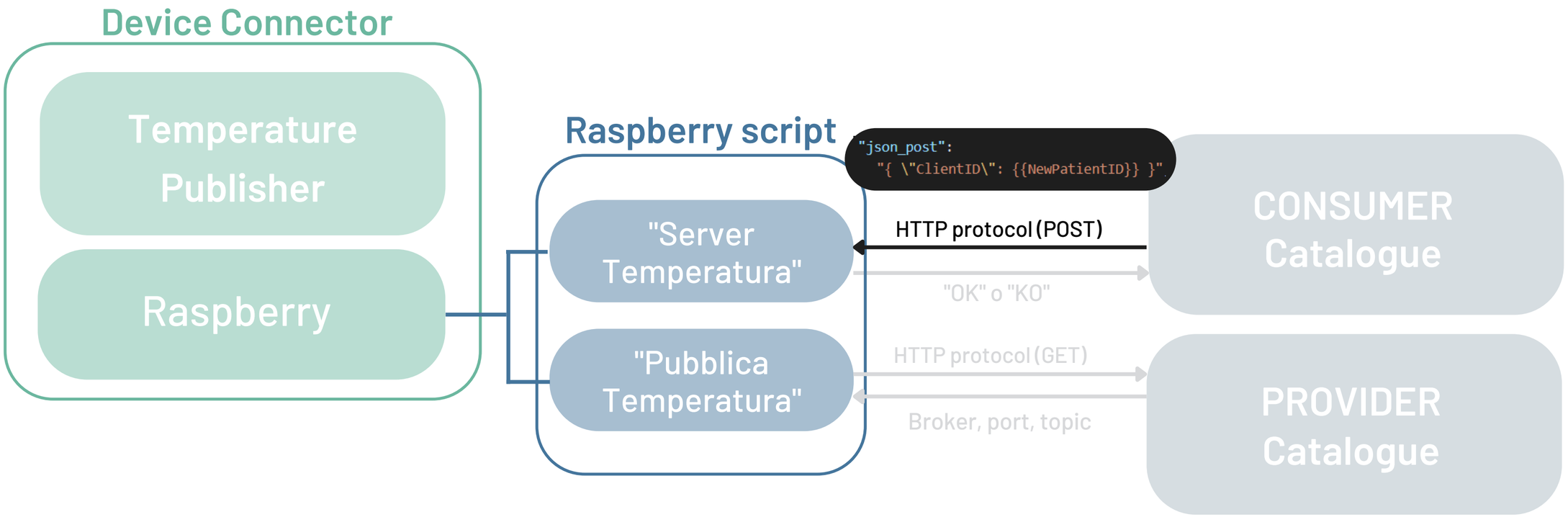
REST Web Services (**provider**)



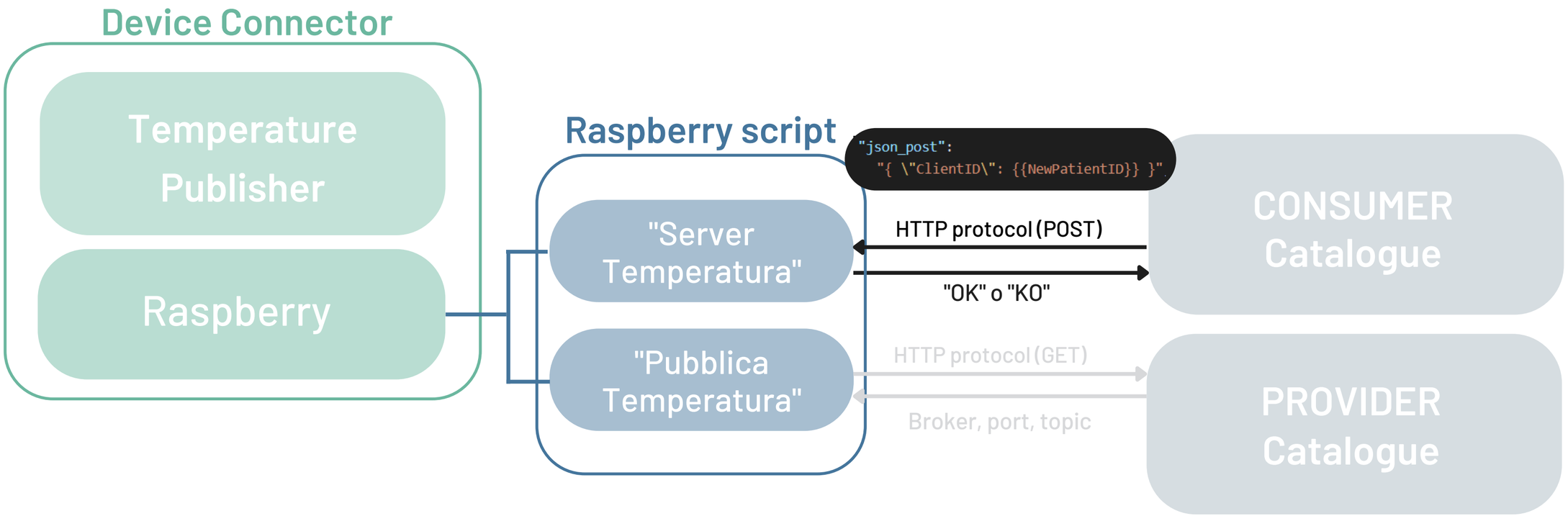
# Raspberry


Device Connector
Temperature Publisher
Raspberry
Raspberry script
"Server Temperatura"
"Pubblica Temperatura"
HTTP protocol (POST)
"OK" o "KO"
HTTP protocol (GET)
Broker, port, topic
CONSUMER Catalogue
PROVIDER Catalogue

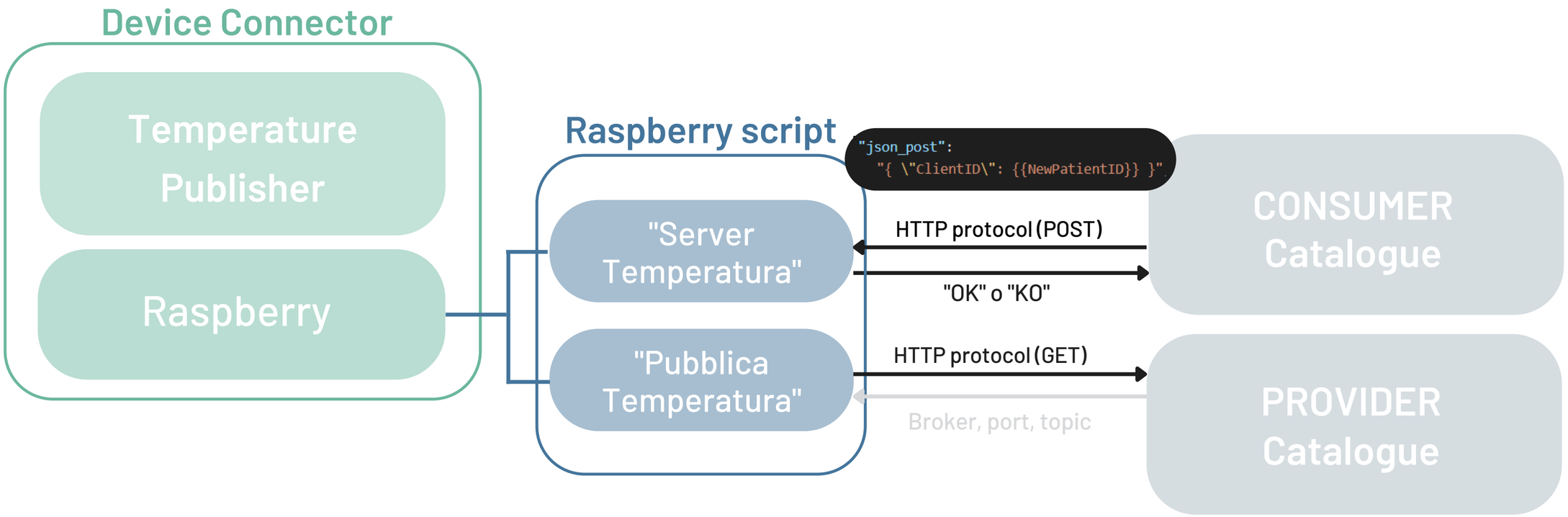
# Raspberry


Device Connector
Temperature Publisher
Raspberry
Raspberry script
"Server Temperatura"
"Pubblica Temperatura"
"json_post":
"{ \"ClientID\": {{NewPatientID}} }"
HTTP protocol (POST)
"OK" o "KO"
HTTP protocol (GET)
Broker, port, topic
CONSUMER Catalogue
PROVIDER Catalogue

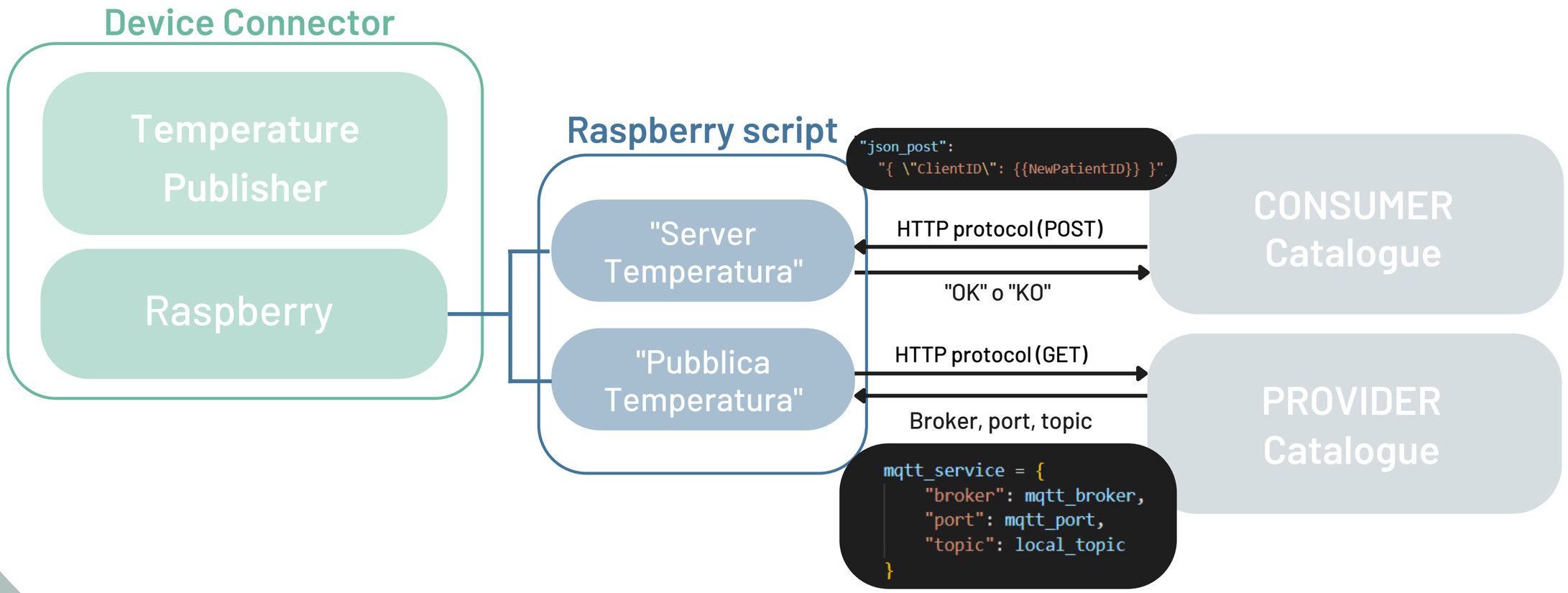
# Raspberry


Device Connector
Temperature Publisher
Raspberry
Raspberry script
"Server Temperatura"
"Pubblica Temperatura"
"json_post":
"{ \"ClientID\": {{NewPatientID}} }"
HTTP protocol (POST)
"OK" o "KO"
HTTP protocol (GET)
Broker, port, topic
CONSUMER Catalogue
PROVIDER Catalogue

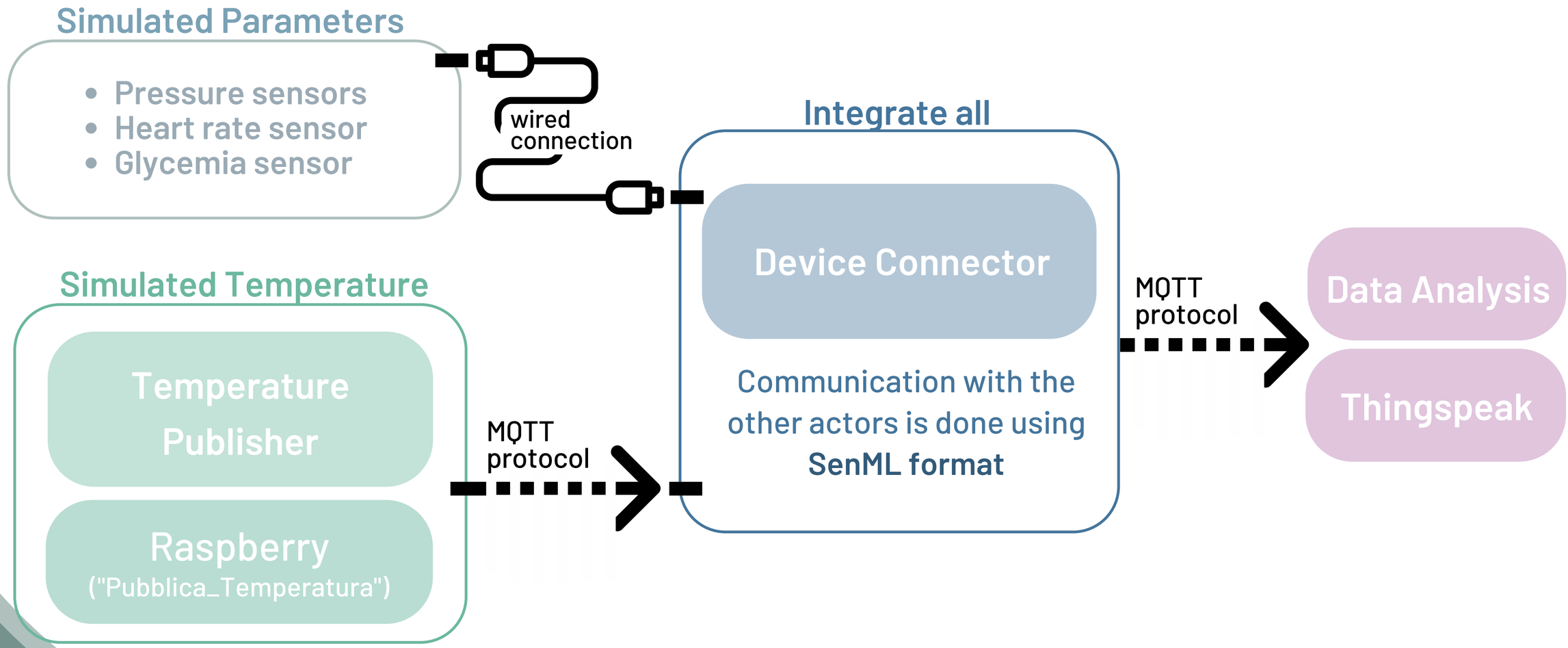
# Raspberry


Device Connector
Temperature Publisher
Raspberry
Raspberry script
"Server Temperatura"
"Pubblica Temperatura"
"json_post":
"{ \"ClientID\": {{NewPatientID}} }"
HTTP protocol (POST)
"OK" o "KO"
HTTP protocol (GET)
Broker, port, topic
CONSUMER Catalogue
PROVIDER Catalogue

# Raspberry

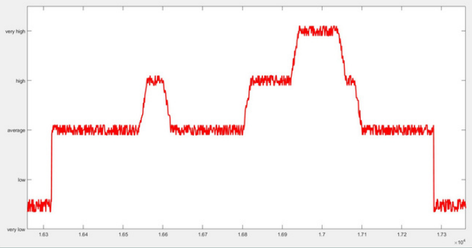

Device Connector
Temperature Publisher
Raspberry
Raspberry script
"Server Temperatura"
"Pubblica Temperatura"
"json_post":
"{ \"ClientID\": {{NewPatientID}} }"
HTTP protocol (POST)
"OK" o "KO"
HTTP protocol (GET)
Broker, port, topic
mqtt_service = {
"broker": mqtt_broker,
"port": mqtt_port,
"topic": local_topic
}
CONSUMER Catalogue
PROVIDER Catalogue

# Device Connector



# Device Connector

Simulated Parameters




wired connection

Integrate all

Device Connector

Communication with the other actors is done using **SenML format**

MQTT protocol

Data Analysis

Thingspeak

Simulated Temperature

Temperature Publisher

Raspberry ("Pubblica_Temperatura")

MQTT protocol

# Device Connector

## Comunication Legend

MQTT Comunication

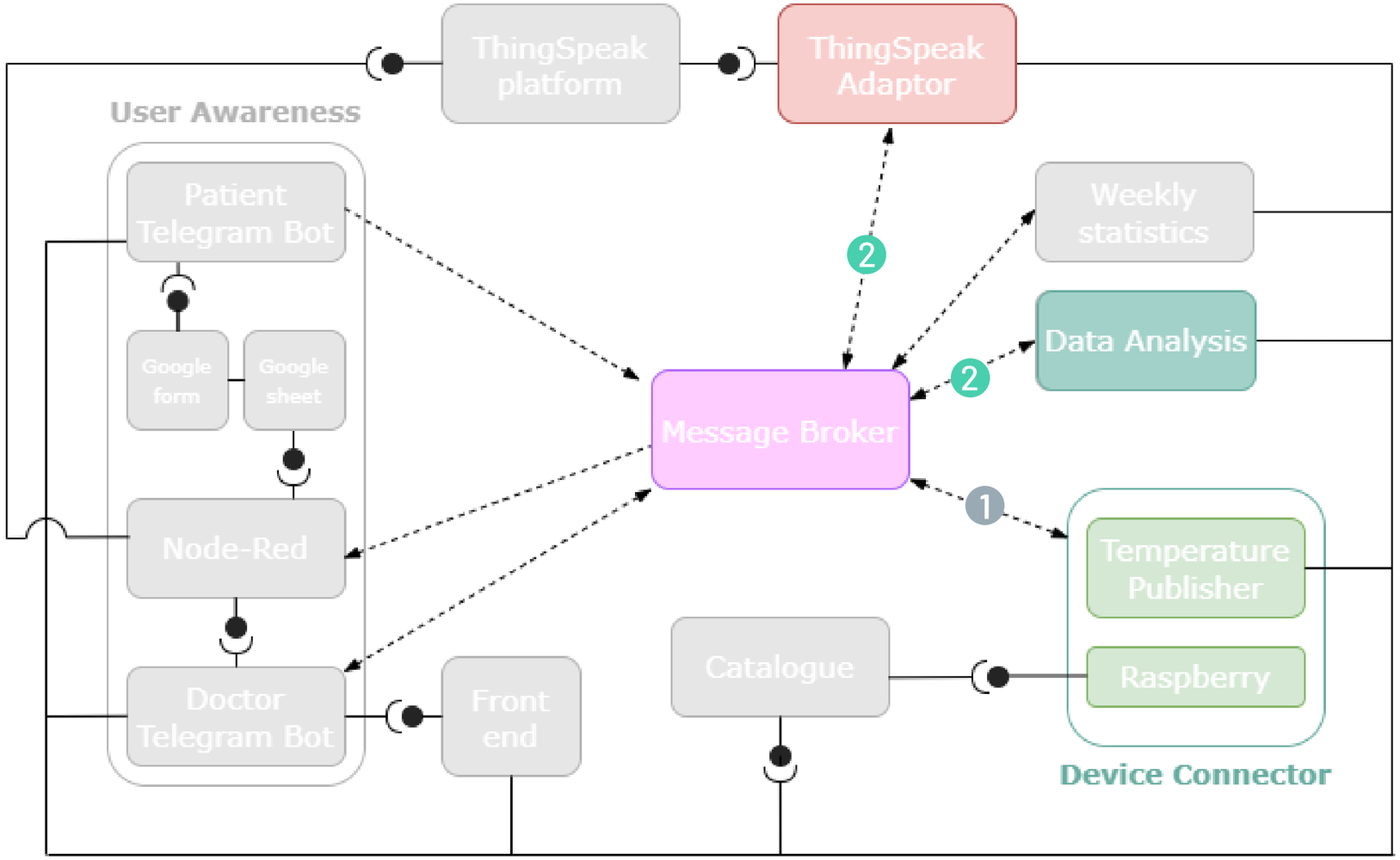
REST Web Services (**consumer**)

REST Web Services (**provider**)

## Topic legend

1) {basetopic}/{patientID}/sensor/glycemia

2) {basetopic}/+/sensor/#



# Data Analysis

## Legend

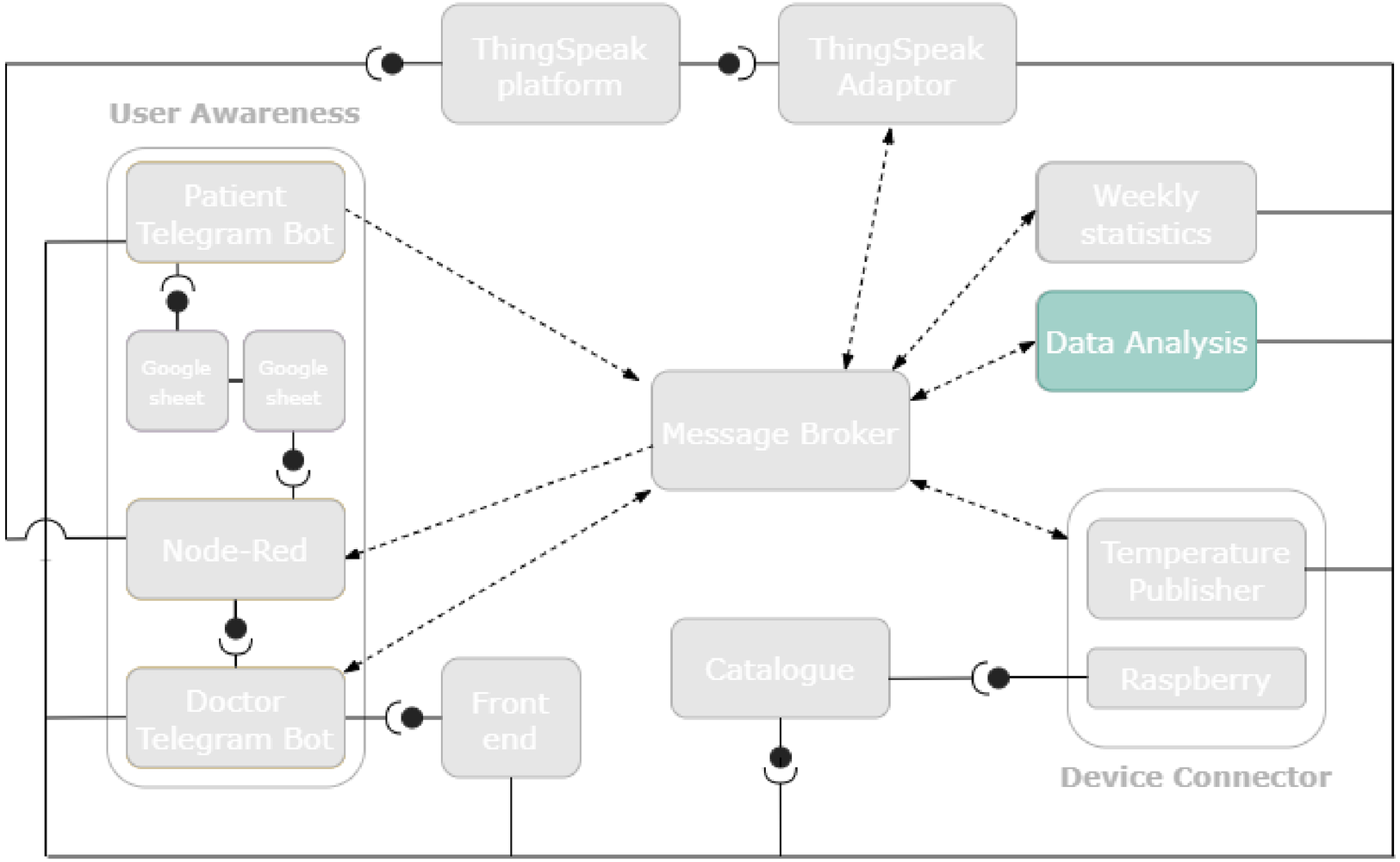
MQTT Comunication

REST Web Services (**consumer**)

REST Web Services (**provider**)

How it works:

- **Trashold** to identify measures out of range
- **Send** message alert when measures are out of range



# Data Analysis

## Comunication Legend

MQTT Comunication

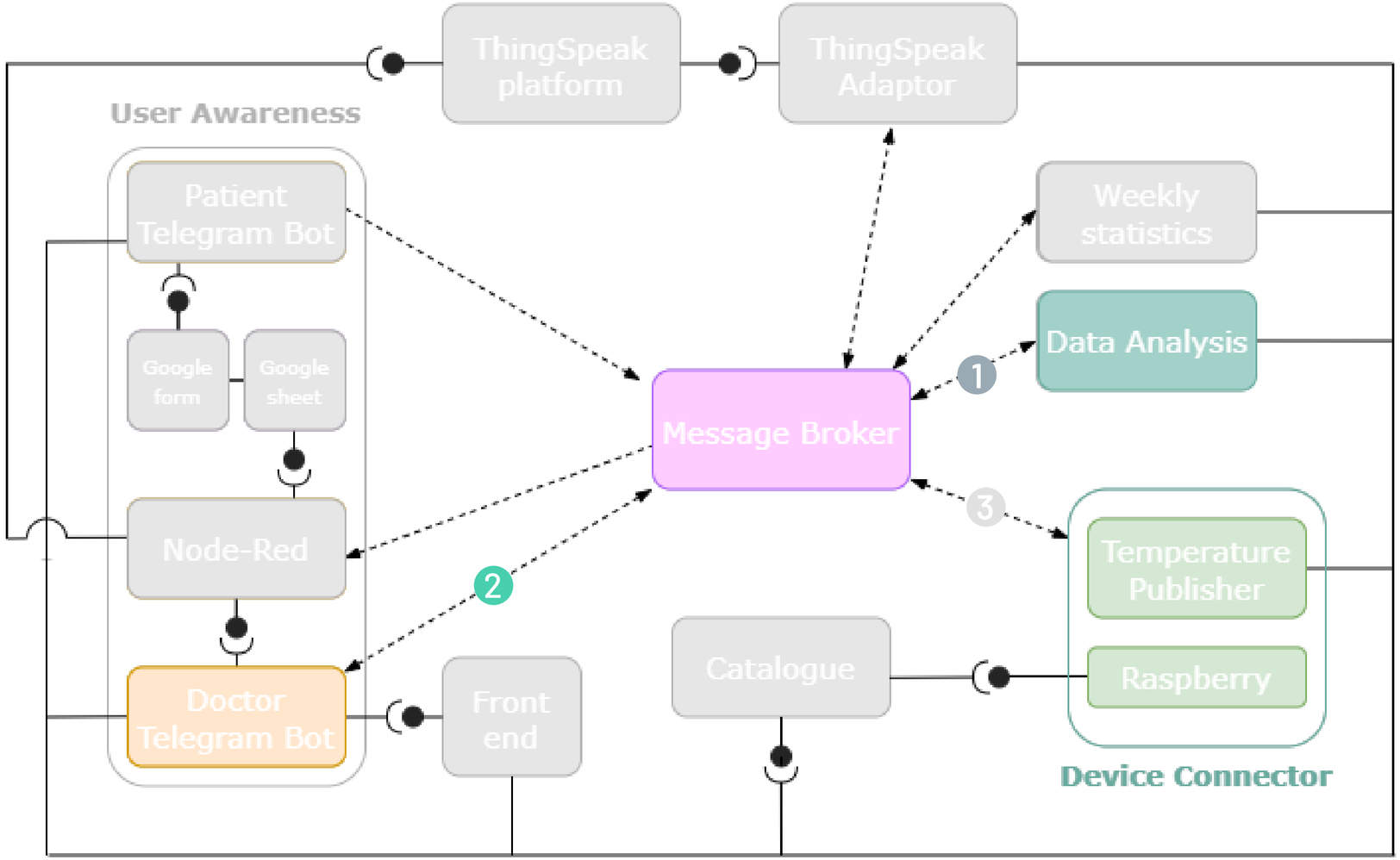
REST Web Services (**consumer**)

REST Web Services (**provider**)

## Topic legend

1) {basetopic}/{patientID}/send_alert
2) {basetopic}/+/send_alert
   {basetopic}/{patientID}/monitoring
3) {basetopic}/+/monitoring

Data Analysis: publishes a **message** to the telegram bot if it detects vital parameters out of range



# Data Analysis

## Comunication Legend

MQTT Comunication

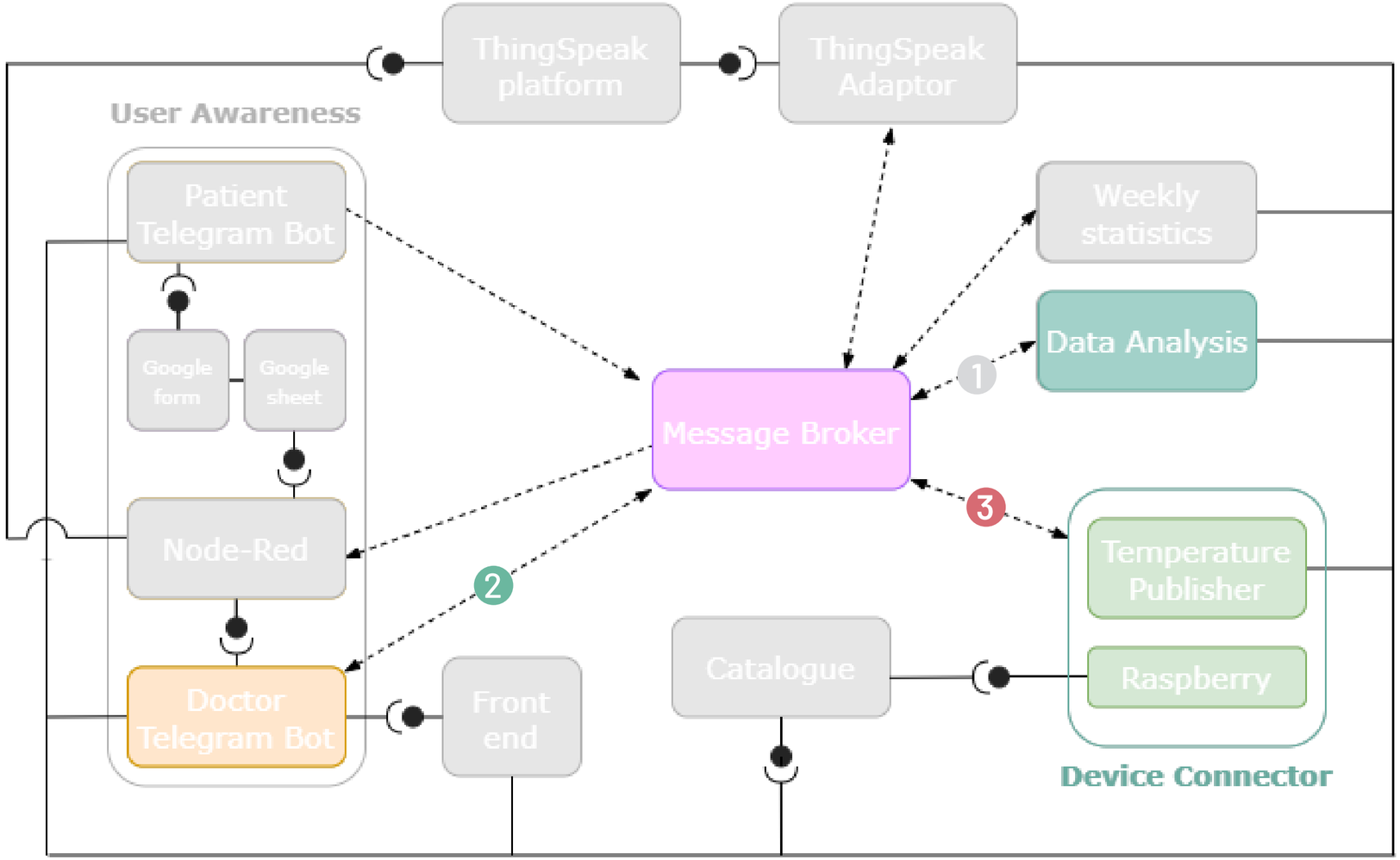
REST Web Services (**consumer**)

REST Web Services (**provider**)

## Topic legend

1) {basetopic}/{patientID}/send_alert
2) {basetopic}/+/send_alert
   {basetopic}/{patientID}/monitoring
3) {basetopic}/+/monitoring

**Doctor Bot:** in case of "Monitoing ON" published to the Device Connector to **increse** the sampled **frequency** of data



# ThingSpeak

## Legend

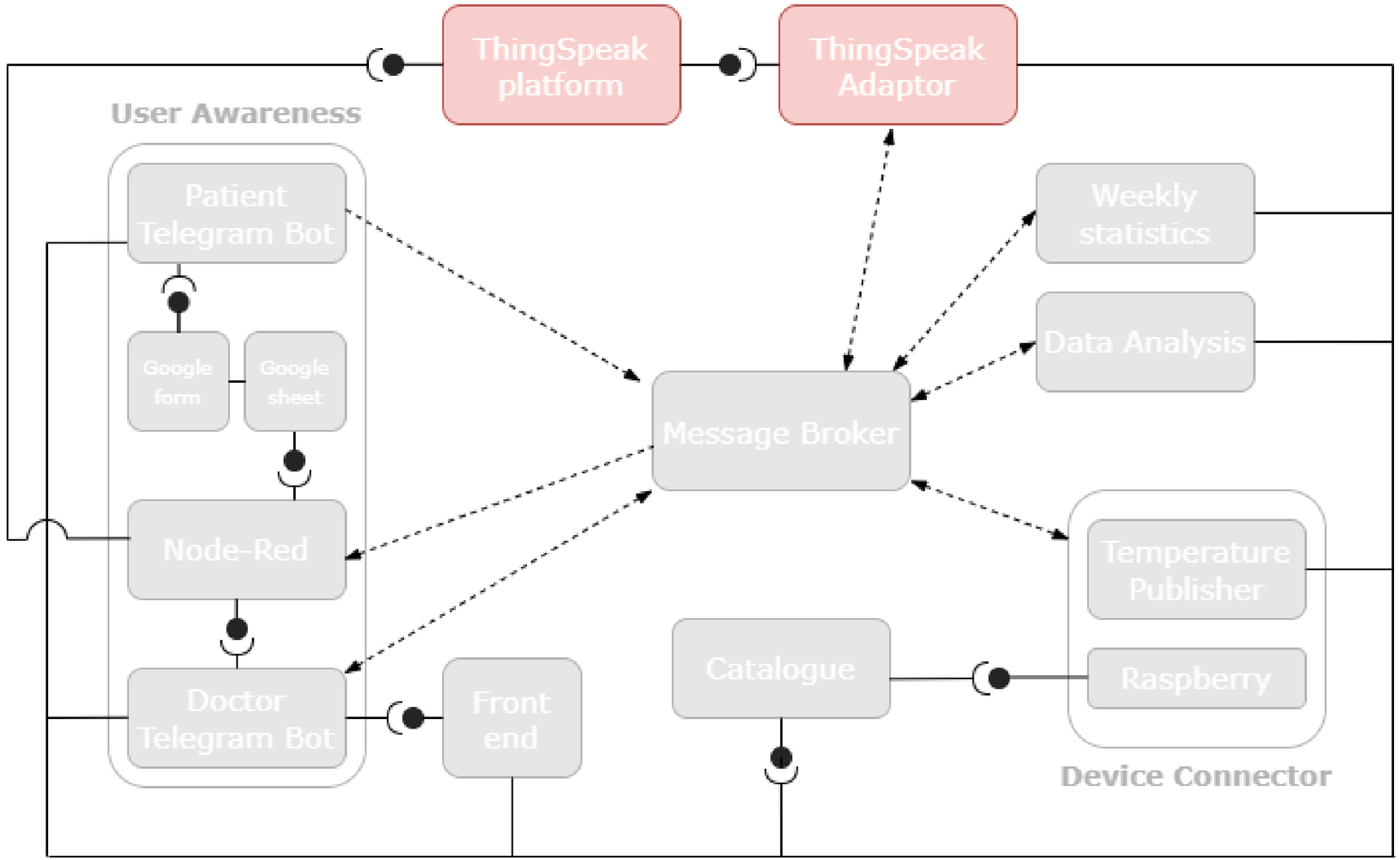
MQTT Comunication

REST Web Services (**consumer**)

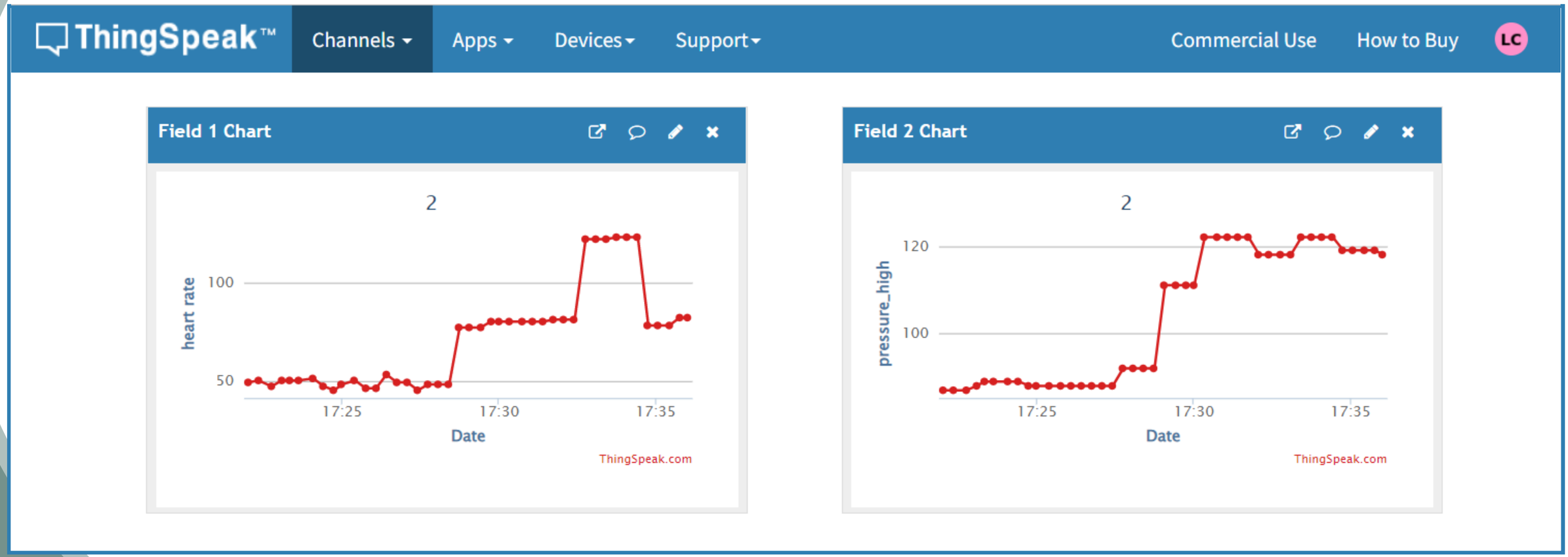
REST Web Services (**provider**)

How it works:

- **Graphing data:** it's a platform used like a database in which there are sensors' measurement
- **Send** data weekly
  - **ThingSpeak Adaptor:** allows you to implement communication via MQTT



# ThingSpeak


ThingSpeak™
Channels
Apps
Devices
Support
Commercial Use
How to Buy
LC
Field 1 Chart
2
heart rate
100
50
17:25
17:30
17:35
Date
ThingSpeak.com
Field 2 Chart
2
pressure_high
120
100
17:25
17:30
17:35
Date
ThingSpeak.com

# ThingSpeak

## Comunication Legend

MQTT Comunication

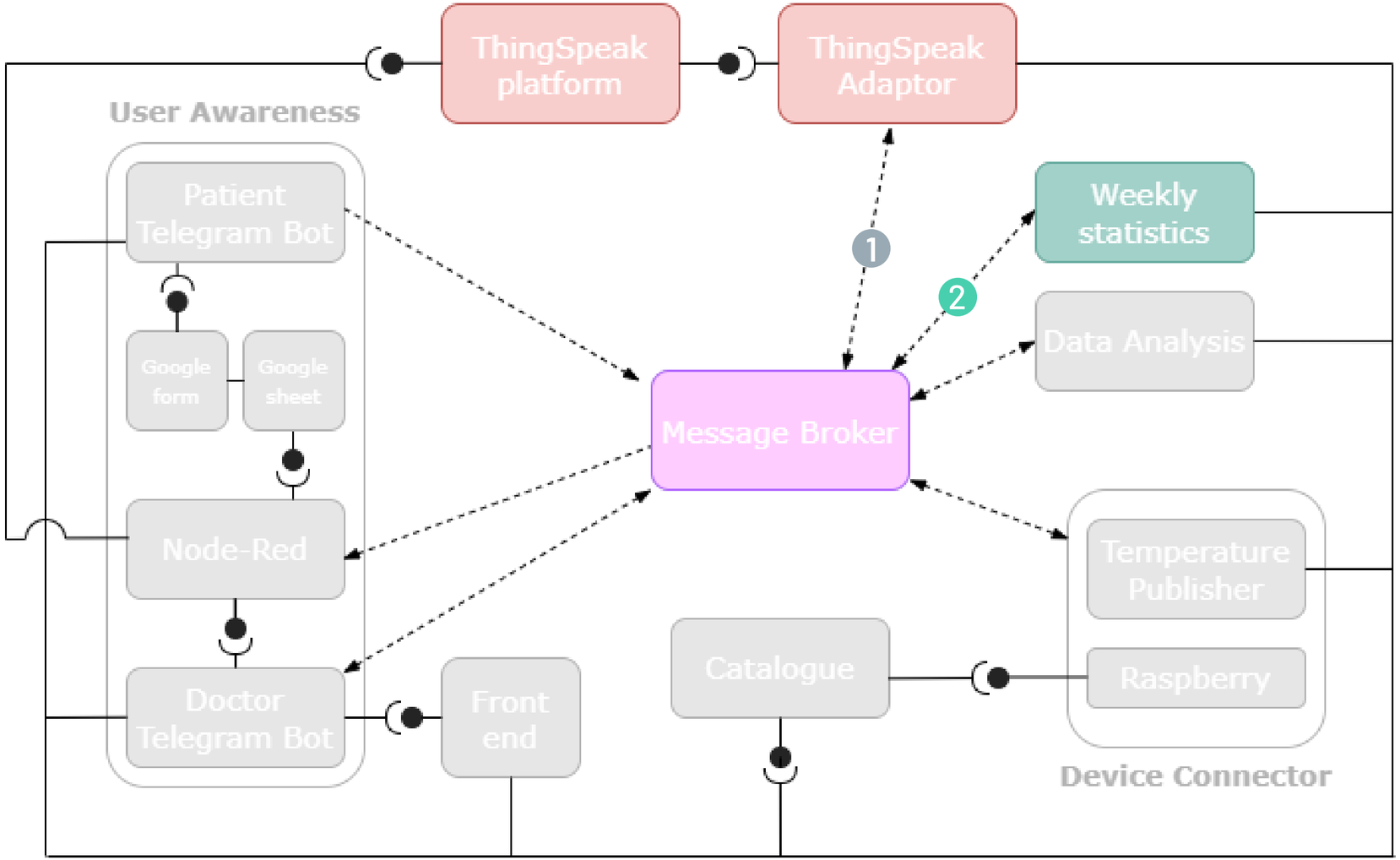
REST Web Services (**consumer**)

REST Web Services (**provider**)

## Topic legend

1) {basetopic}/statistics_info/{patientID}/#
2) {basetopic}/statistics_info/#

Thingspeak sends **measurements** every **week** to Weekly Statistics



# Telegram Bot - Patient

## Comunication Legend

MQTT Comunication

REST Web Services (**consumer**)

REST Web Services (**provider**)

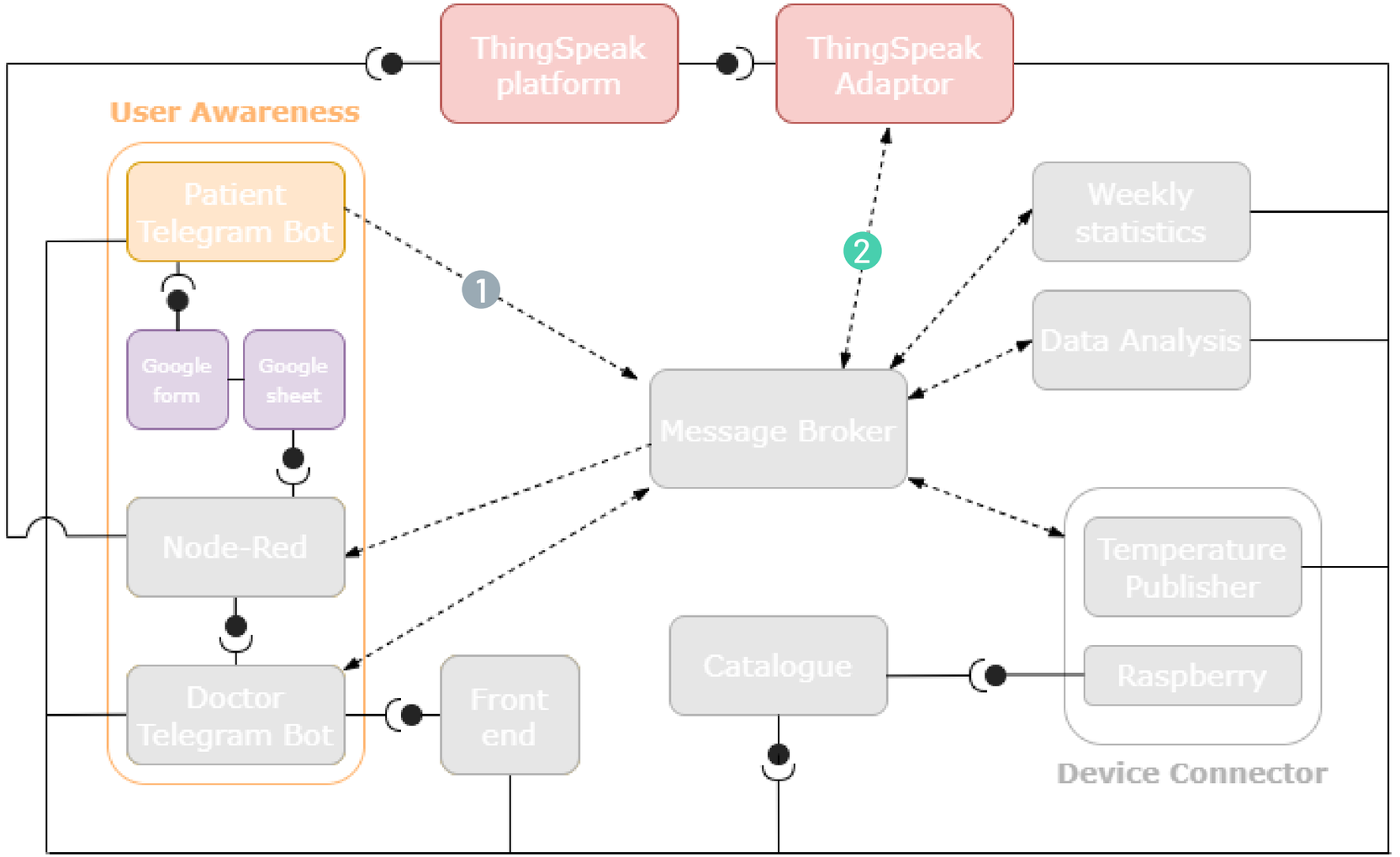
## Topic legend

1) {basetopic}/{patientID}/peso

2) {basetopic}/+/peso

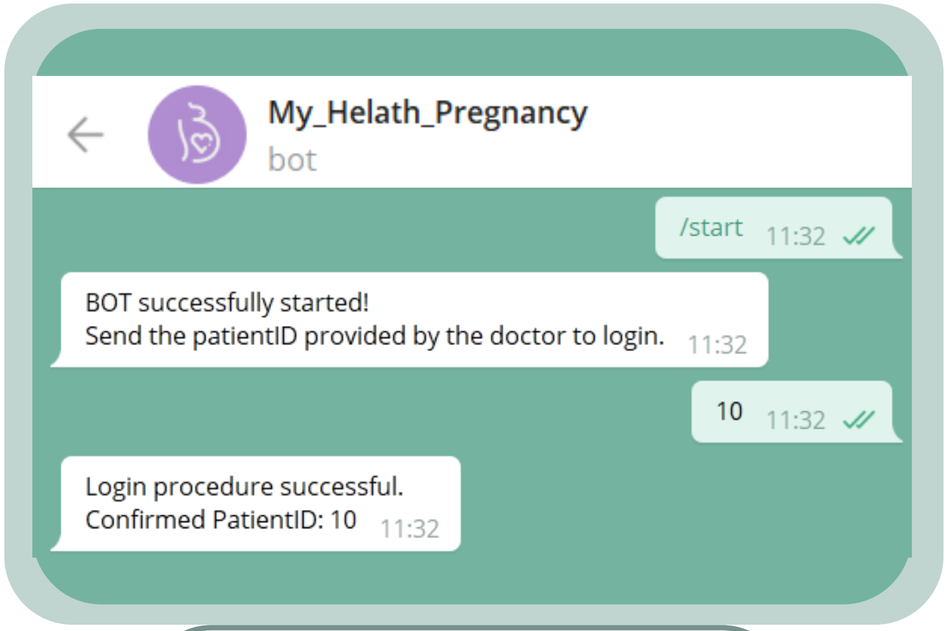
Patient Telegram Bot sends:

- **weight** to Thingspeak
- **survey** to Node Red

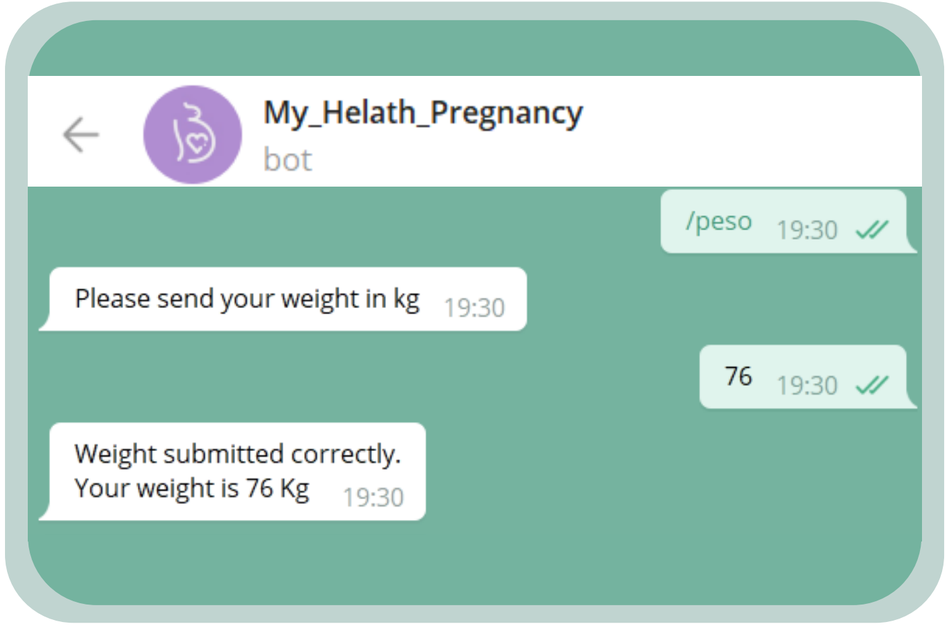


# Telegram Bot - Patient

## Patient registration




## Send weight




PUT request to update
Telegram ID in the catalog

# Telegram Bot - Patient

## Smart Heath survey

Insert your patient code *

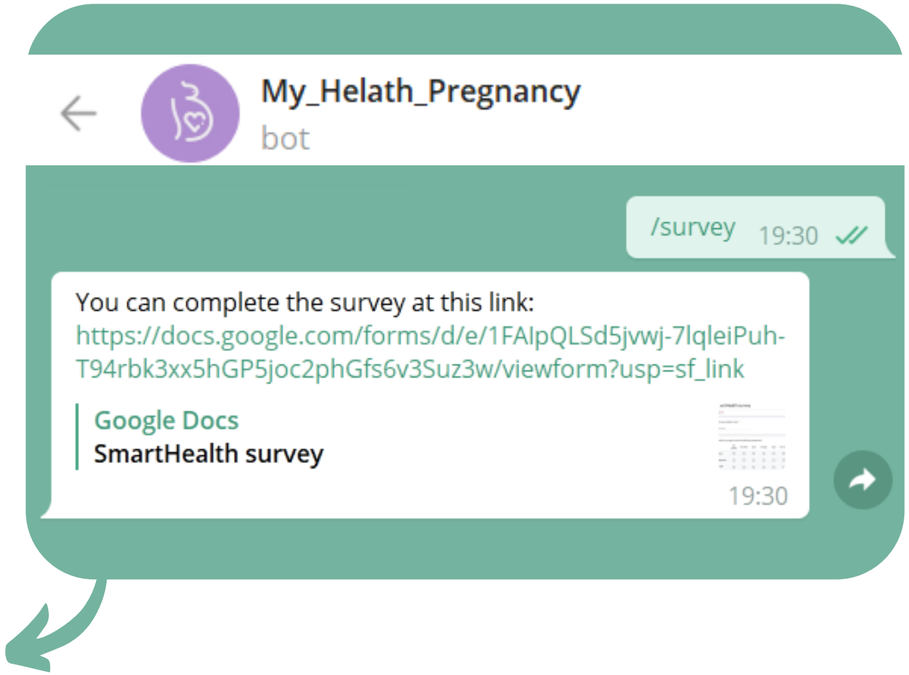
1

Indicate if you got any of the following symptoms *

| | Not present | Very little | Little | Avarage | High | Very high |
|---|---|---|---|---|---|---|
| Nausea | ○ | ○ | ○ | ○ | ◉ | ○ |
| Constipation | ○ | ○ | ◉ | ○ | ○ | ○ |
| Stomach acidity | ○ | ○ | ○ | ◉ | ○ | ○ |
| Urine leakage | ◉ | ○ | ○ | ○ | ○ | ○ |

Any further symptom to report?

Frequent need of drinking water




Google Sheets

Google Forms

# Telegram Bot - Doctor

## Legend

MQTT Comunication

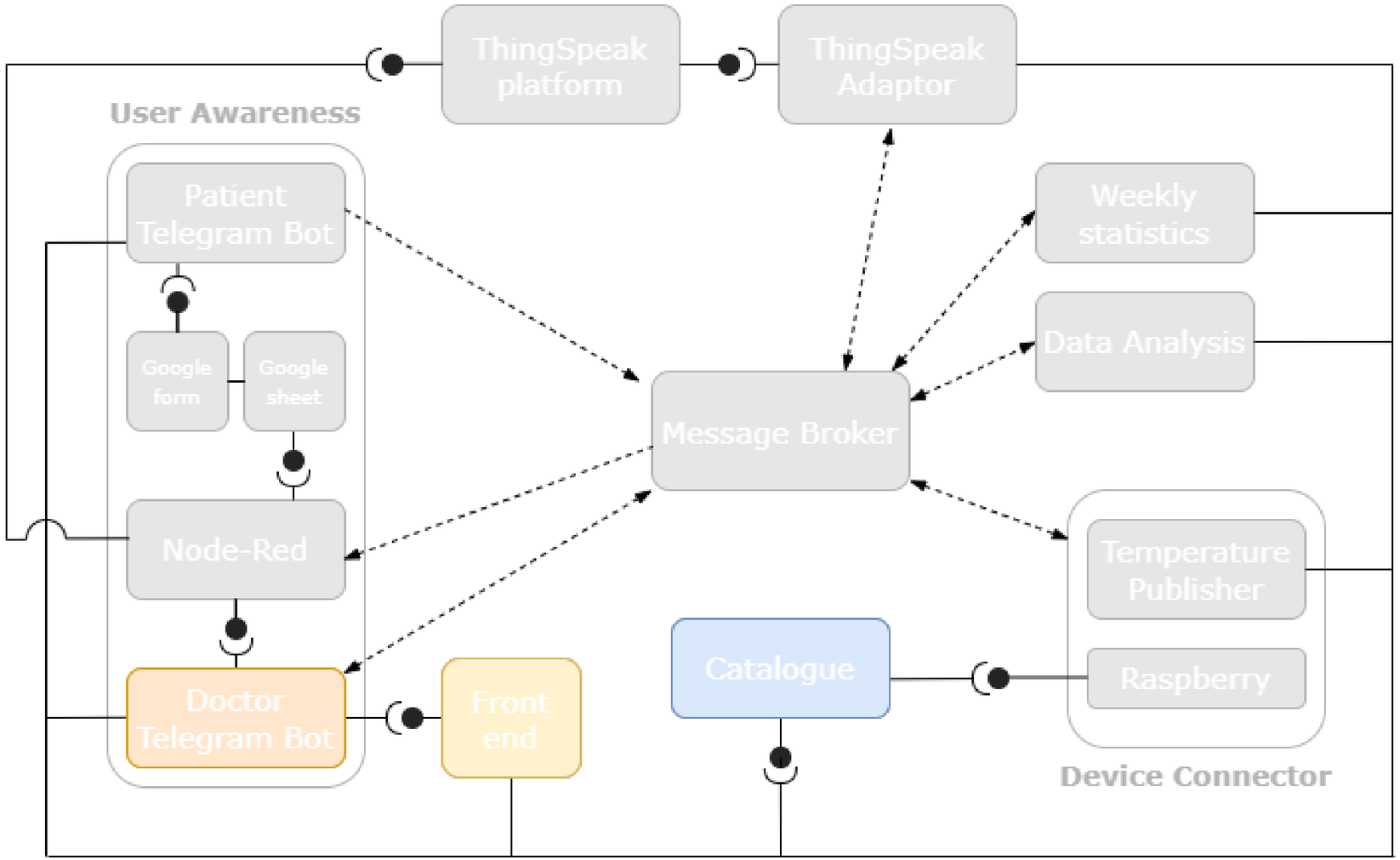
REST Web Services (**consumer**)

REST Web Services (**provider**)

## DoctorTelegram Bot to:

- sign in **doctor**
- sign in **patient**

- **monitoring patient**
- **acess data's patient**



# Telegram Bot - Doctor

## Doctor registration page

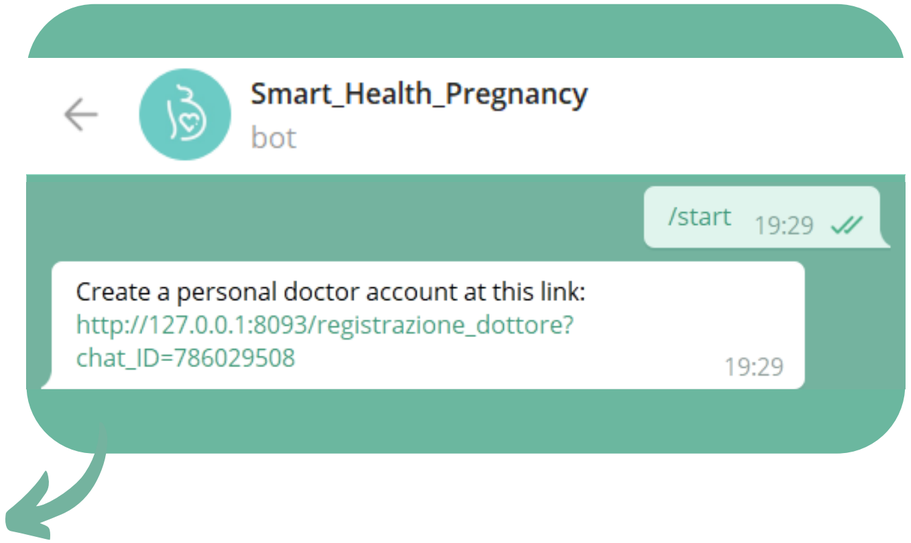
Name

Surname

Mail

Submit

POST request to the catalog



# Telegram Bot - Doctor

## Patient registration page

Name

Laura

Surname

Cubeddu

Tax ID code

CBDLRA98L35B367K

Patient Mail

laura.c@gmail.com

First Pregnancy Day (aaaa-mm-gg)

2022-05-12

Device Name

3

Submit

POST request to the catalog


| Patient ID | Name | Surname | Tax ID code | Patient Mail | First Pregnancy Day | Device Name | Rasp. Config. |
|---|---|---|---|---|---|---|---|
| 16 | Laura | Cubeddu | CBDLRA98L35B367K | laura.c@gmail.com | 2022-05-12 | 3 | no |

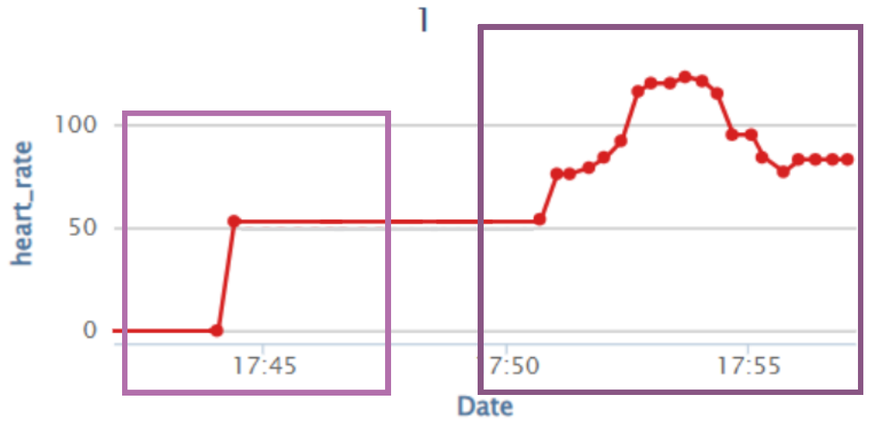
# Telegram Bot - Doctor

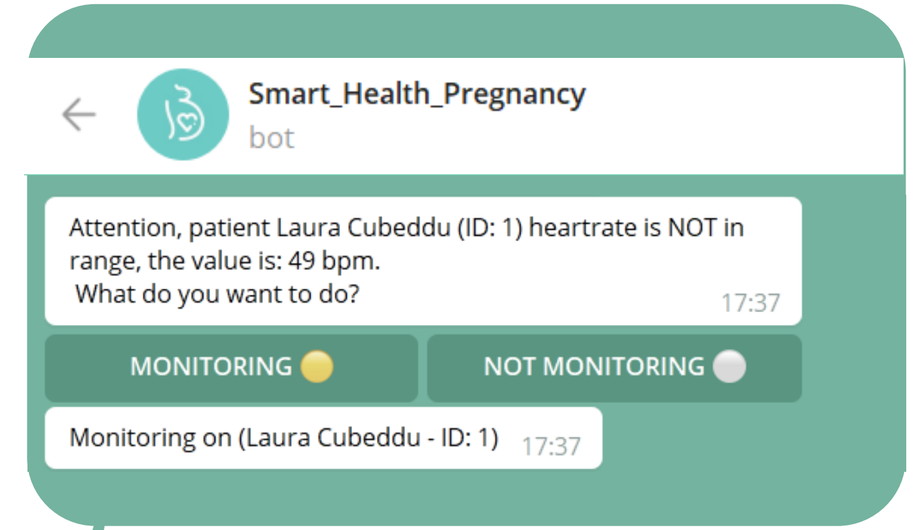
## ThingSpeak graphs modified

Getting alert message from the **Patient Control microservice** in case some vital parameters not in phisiological range

**Monitoring OFF:** doctor receives again alert message and the high monitoring frequency is stopped;

**Monitoring ON:** doctor doesn't receive any other alert message but the monitoring frequency is improved from the **Monitoring microservice;**

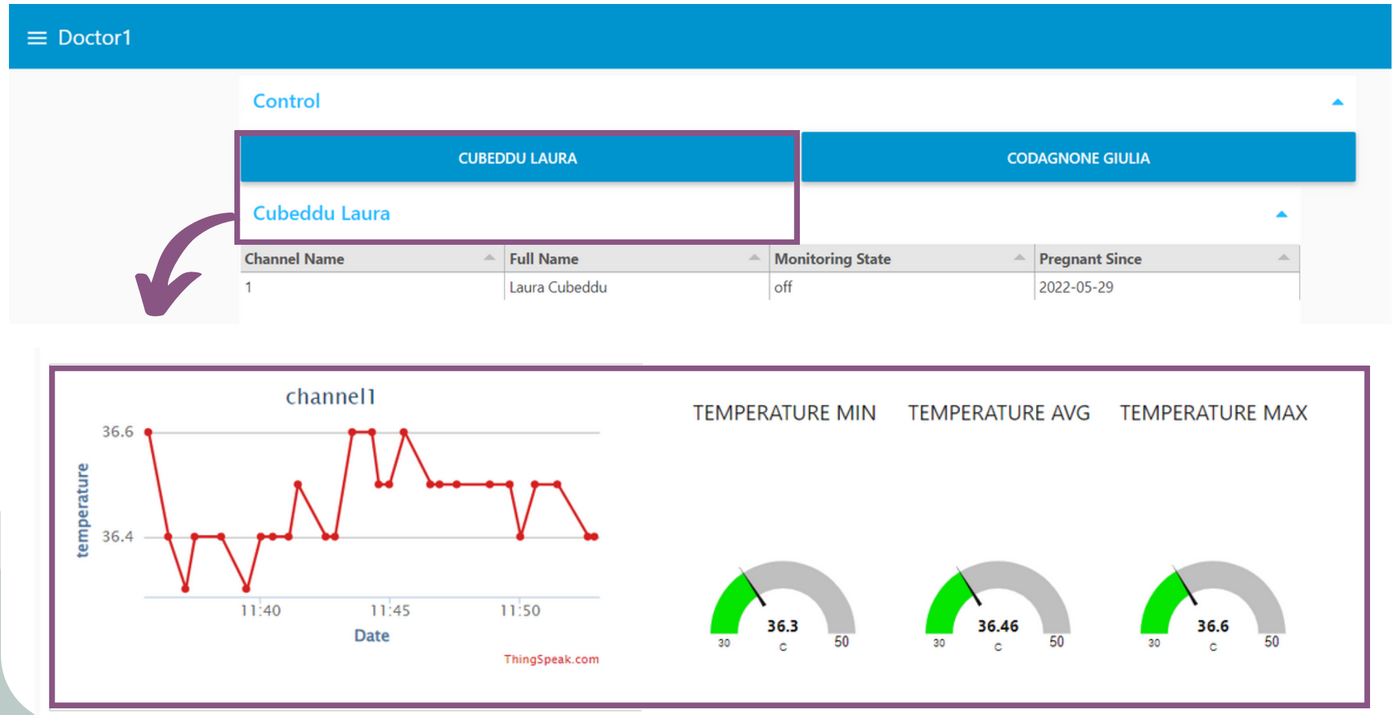


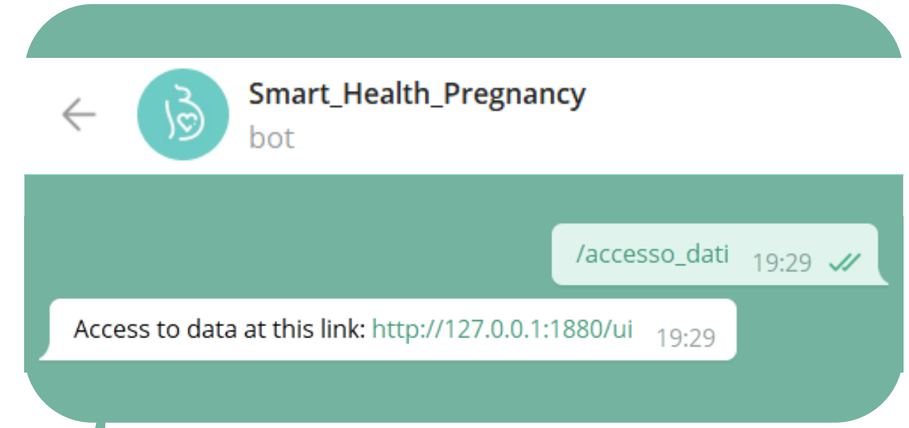



# Telegram Bot - Doctor

## Node - RED graphs

Visualization of ThingSpeak graphs about vital signs



# Node - RED

## Comunication Legend

MQTT Comunication

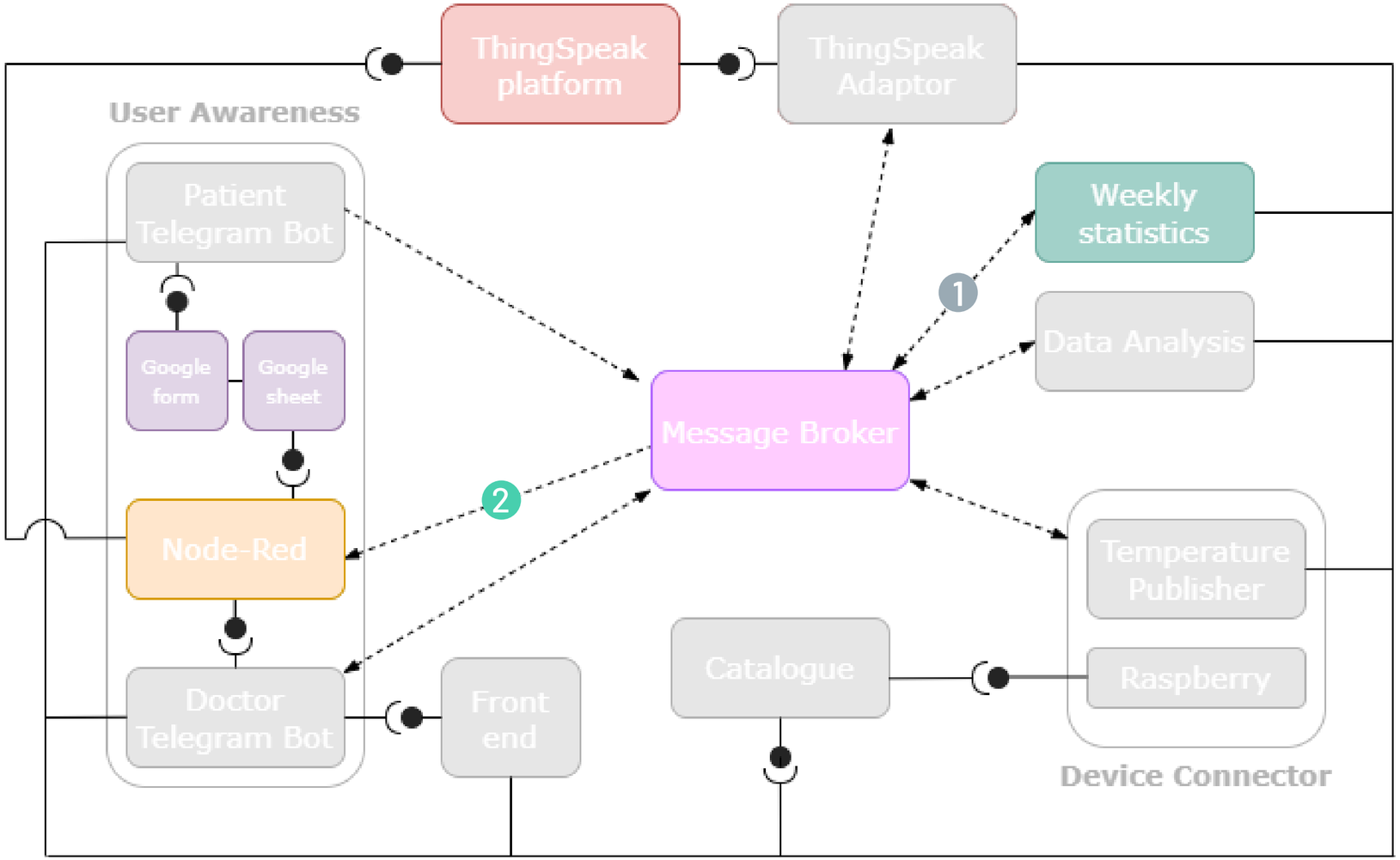
REST Web Services (**consumer**)

REST Web Services (**provider**)

## Topic legend

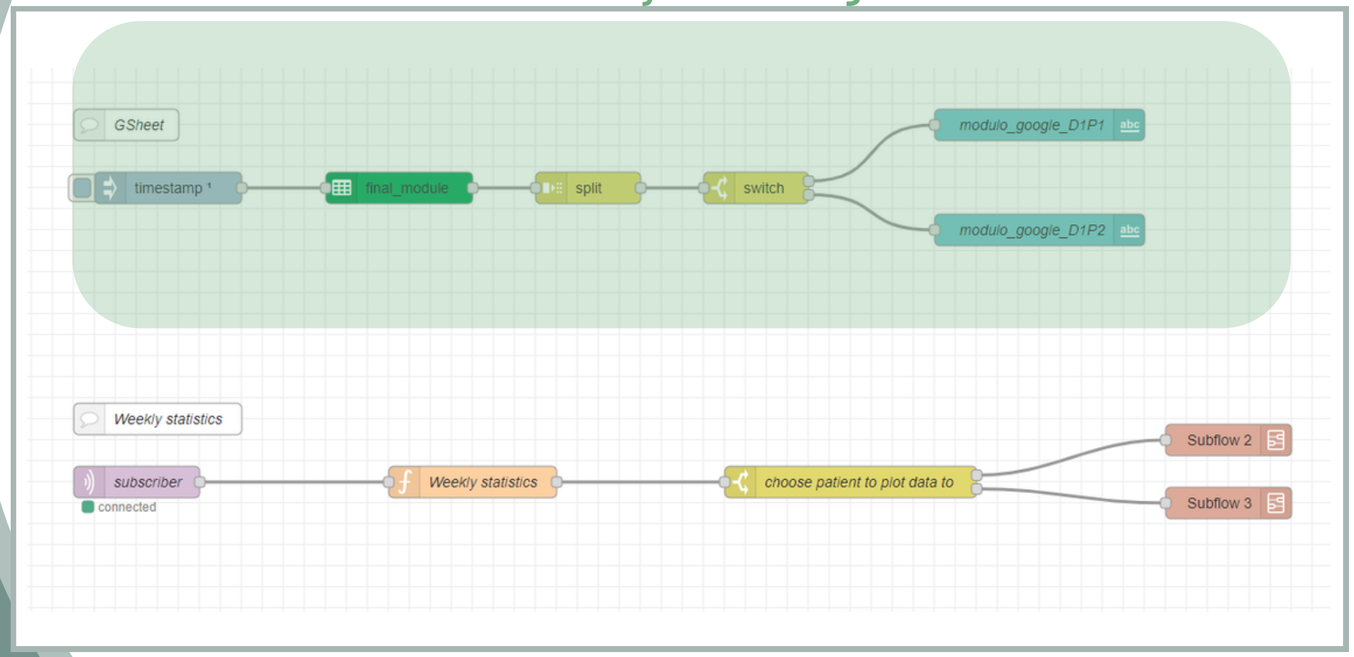
1) {basetopic}/statistics/{patientID}/#
{basetopic}/info/{patientID}/#
2) {basetopic}/statistics/#
{basetopic}/info/#

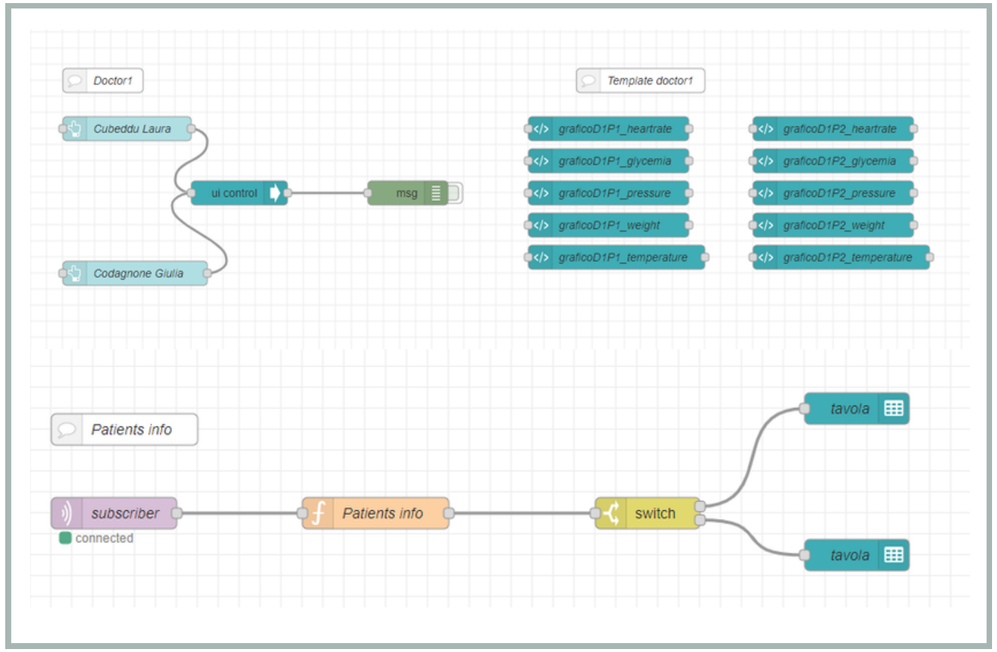
Node Red takes **statistics** and **personal** data of patients to visualize them



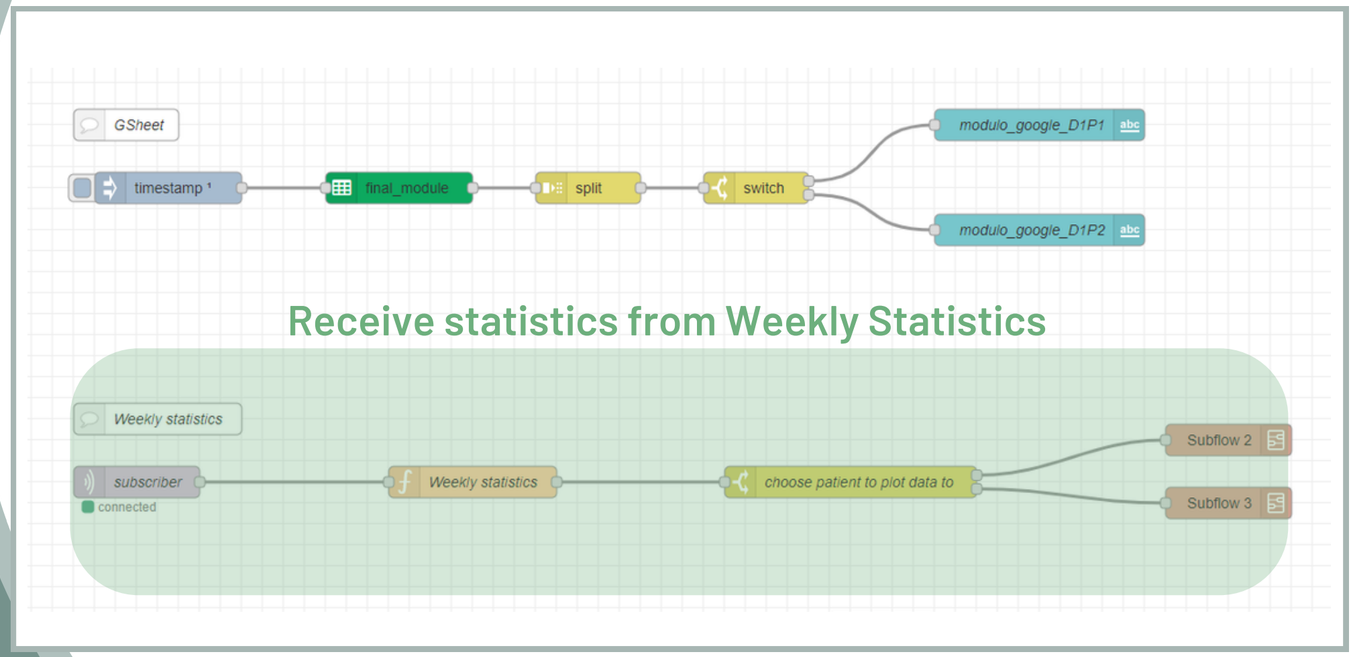
# Node - RED - Flow and nodes

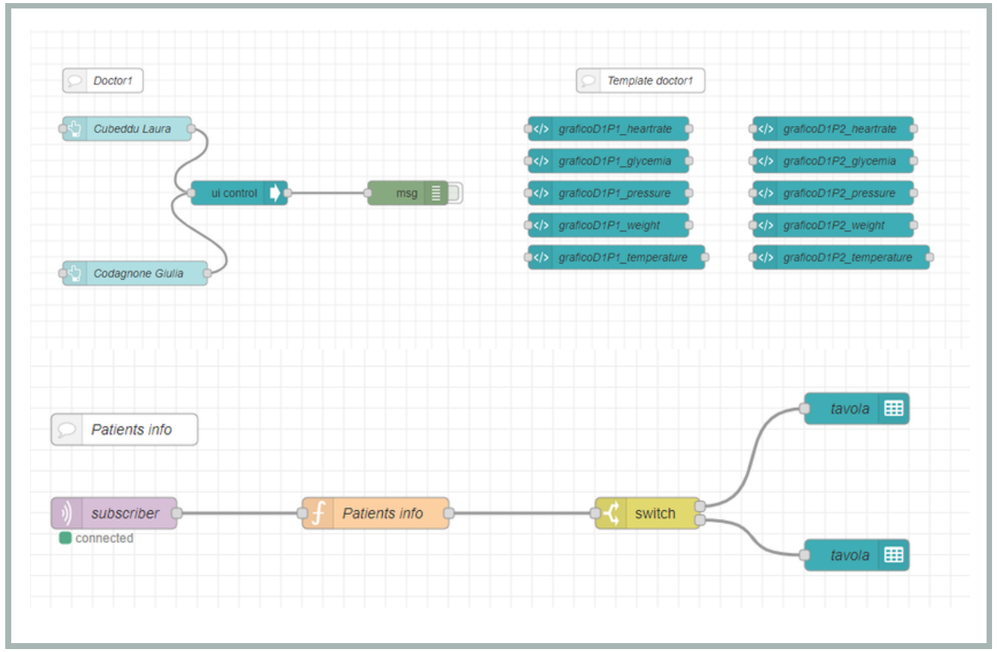
## Receive survey from Googl Sheet



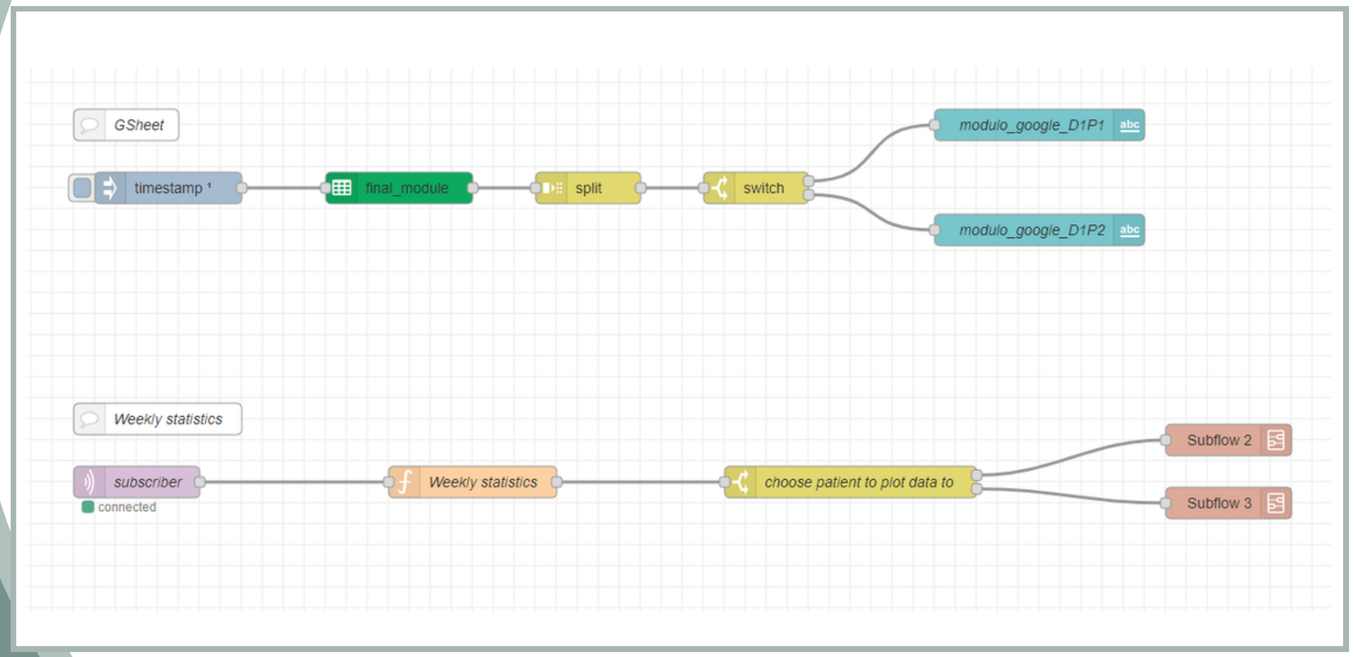



# Node - RED - Flow and nodes

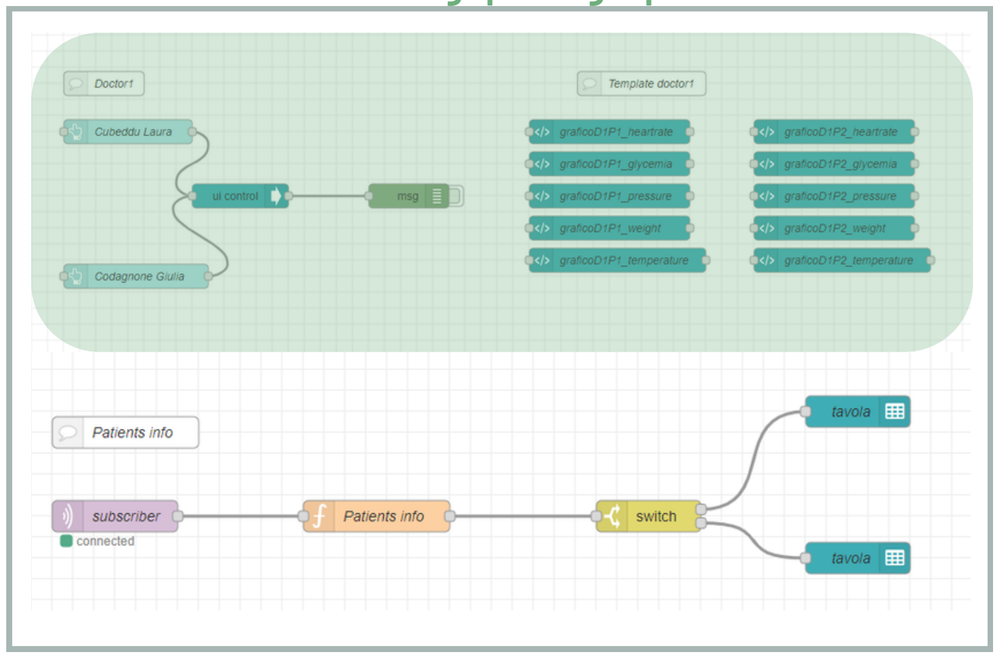


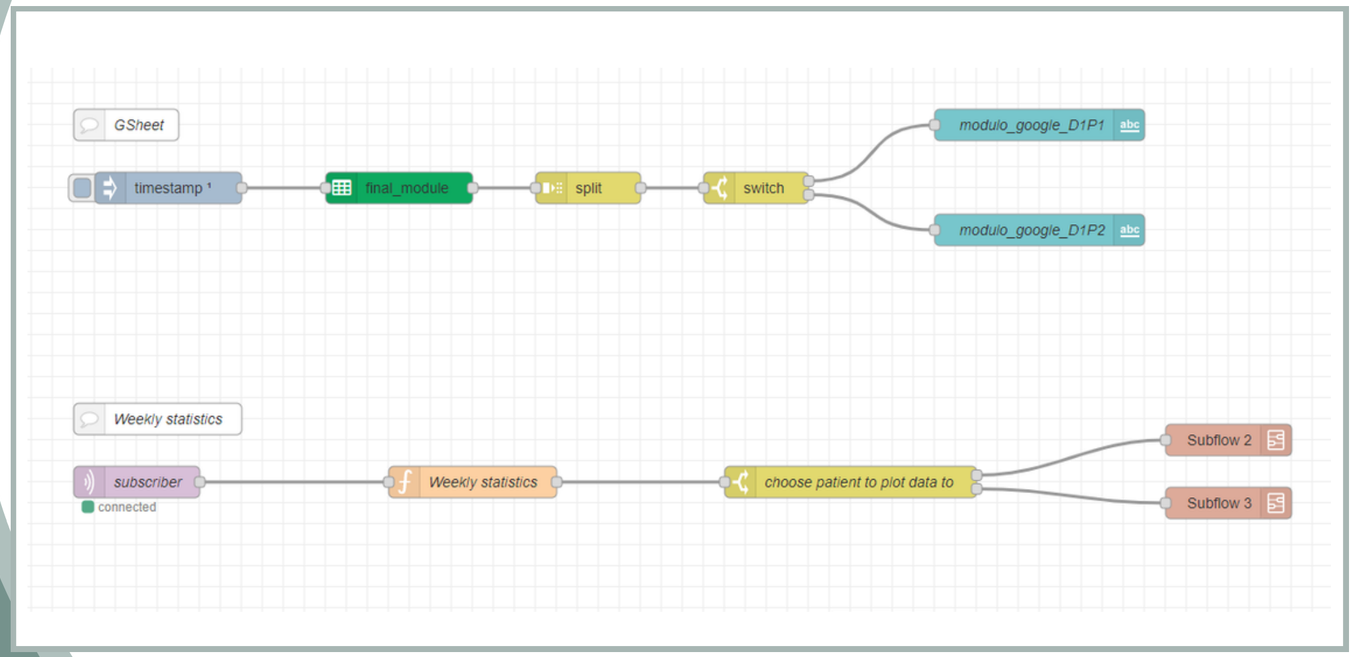
# Node - RED - Flow and nodes

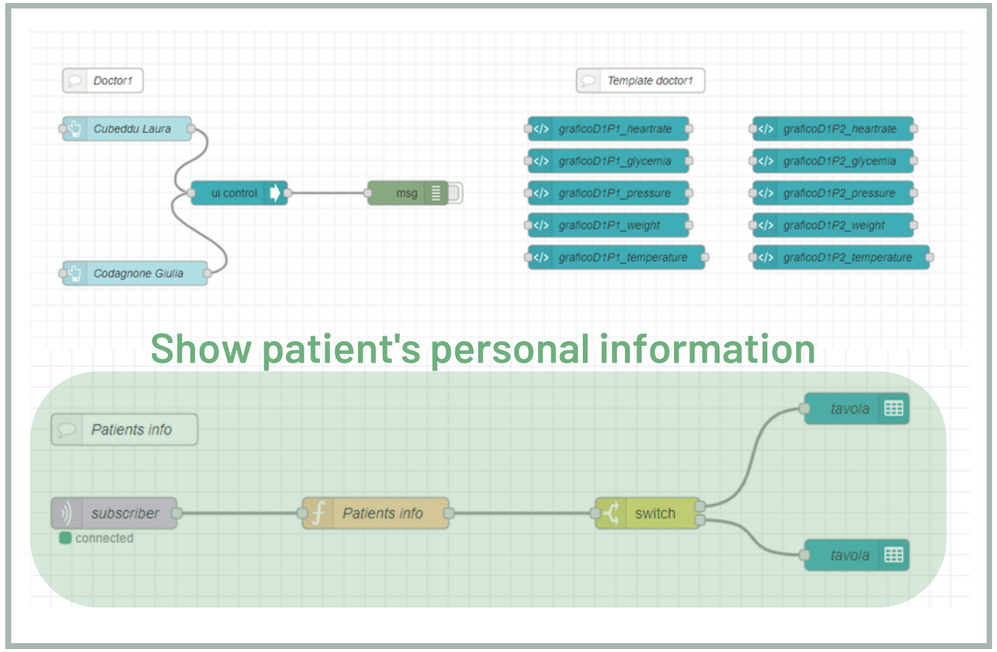



Show thingspeak graph



# Node - RED - Flow and nodes

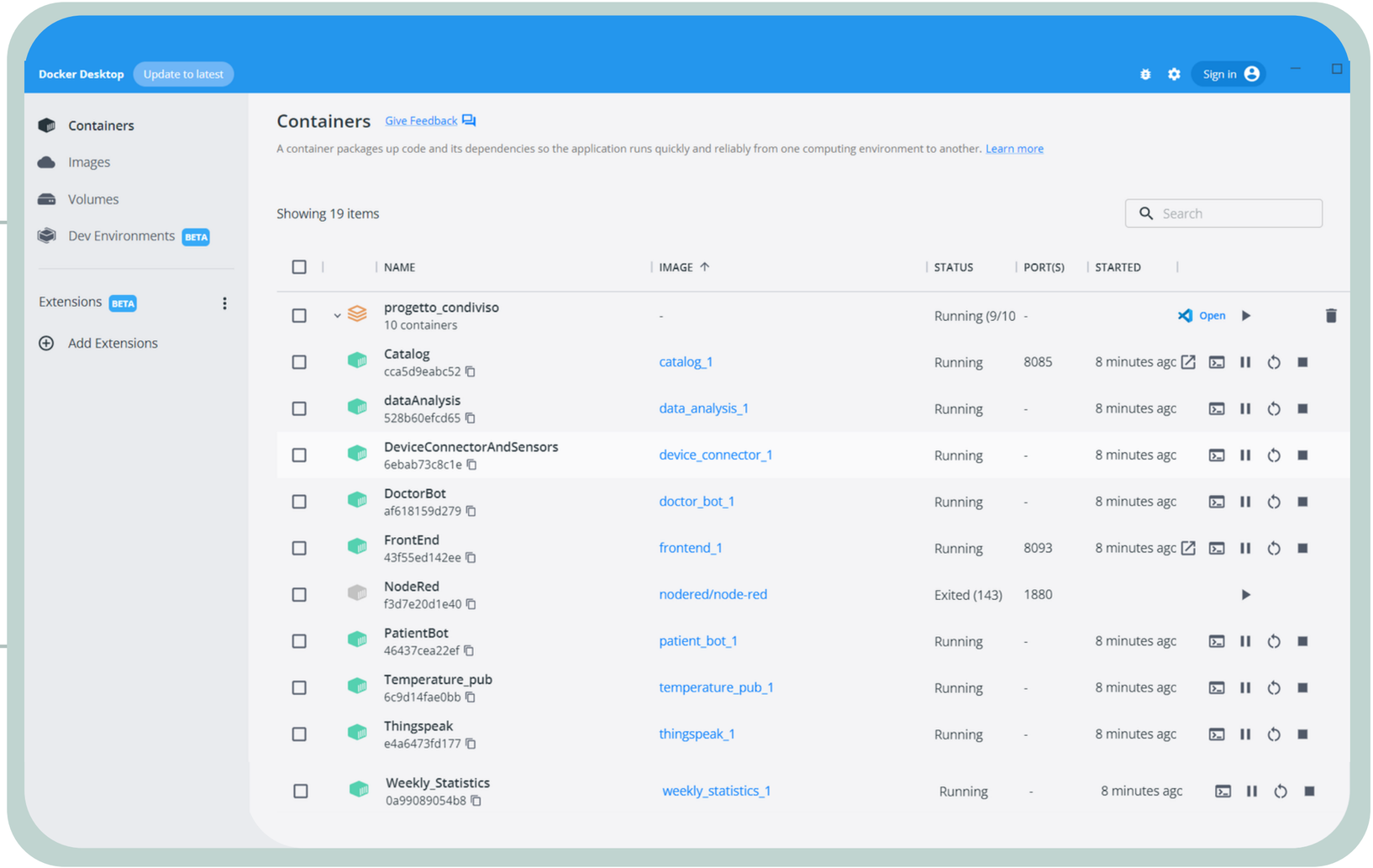


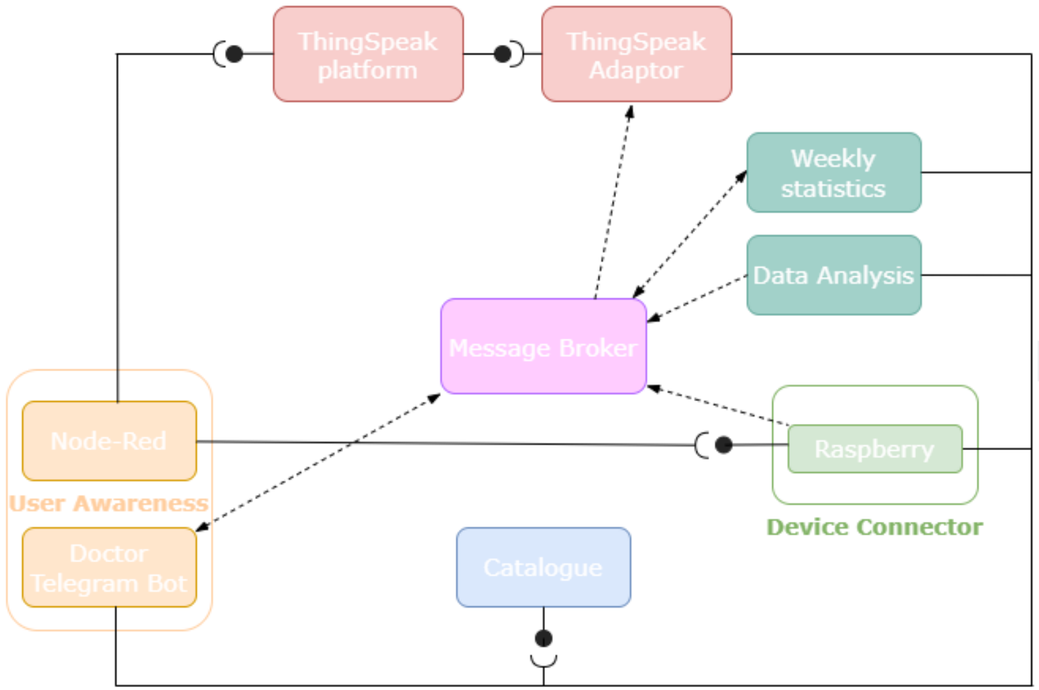
# Docker Compose

## How it works:

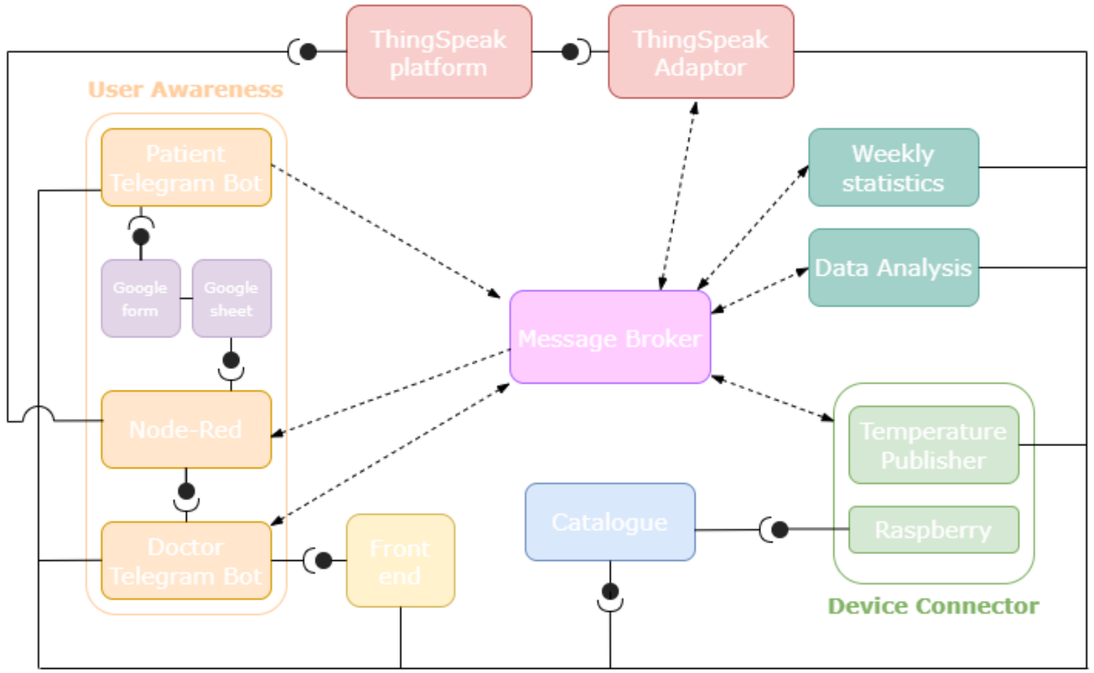
Docker Compose can run multiple containers at the same time, while Docker can only start one container at a time



# Block Diagram




**initial** proposal




**modified** proposal

# Thanks for your attention


Team members:

289512 Aime Elisa
292257 Codagnone Giulia
284277 Cubeddu Laura
282133 De Luca Antonio