BEUTH HOCHSCHULE FÜR TECHNIK BERLIN
University of Applied Sciences

Python und IoT: MQTT

vorgelegt von

Thee Vanichangkul

814517

Studiengang: Technische Informatik

Modul: Advanced Python

betreut durch: **Prof. Dr. rer. nat. Rüdiger Weis**

Berlin, 19.Januar 2017

Contents

Abkürzungsverzeichnis **3**

1 Motivation **1**

2 Was ist IoT? **1**

3 Was ist MQTT? **1**

4 Warum MQTT? **2**

5 Wie Funktioniert MQTT? **3**

5.1 Publish/Subscribe 3

5.2 Topic 4

5.3 Quality of service (QoS) 5

5.4 Last Will Testament (LWT) 5

6 Python und MQTT **6**

6.1 So installiert man MQTT für Python 6

6.2 Test Programme 6

7 Quellen **8**

Abkürzungsverzeichnis

IoT	Internet of Things	M2M	Machine to Machine
TCP	Transmission Control Protocol	WWW	World Wide Web
URL	Uniform Resource Locator		

1 Motivation

Als ich noch bei Wincor Nixdorf als Werkstudent gearbeitet habe, habe ich viele Testskripte für Kassensysteme in Ruby geschrieben. Meine Interesse an Skriptsprachen hat sich seitdem stark intensiviert. Danach habe ich auch als Werkstudent bei Siemens angefangen Skripte in Python zu Schreiben und habe gleichzeitig auch mehr über IoT gelesen.Obwohl Ich die Syntax von Ruby schöner als Python finde, ist Python mehr populär und hat mehr Bibliotheken bezüglich Iot. Deswegen habe ich in meiner Freizeit Python und Iot ausprobiert. Im Rahmen des Moduls Advanced Python wurde mir die Möglichkeit geboten mich in ein für mich interessantes Python Thema zuvertiefen.

2 Was ist IoT?

Internet of Things (IoT) bzw. Das Internet der Dinge ist die Digitalisierung von physischen Objekten, die normallerwise in der "Offline-welt" leben wie Toaster und Kühlschränke und deren Vernetzung über das Internet. Um die Dinge (Geräte) im Netz zu vernetzen braucht man aber ein Kommunikationsprotokoll, die geeignet ist um den Anforderungen des Internet of Things zu erfüllen. Mein ausgewältes Protokoll ist MQTT.

3 Was ist MQTT?

Das Protokoll MQTT wurde 1999 von Andy Stanford-Clark (IBM) und Arlen Nipper (Cirrus Link) als ein M2M-Kommunikationsprotokoll für SCADA-Systeme zum Monitoring von Ölpipelines entwickelt. Es ist ein auf TCP basierendes, leichtgewichtiges und schlankes Protokoll. Es bietet bandbreit- und powerlimitierenden Geräten eine Möglichkeit, miteinander zu kommunizieren. Das Protokoll MQTT ist auch clientseitig sehr einfach zu implementieren. Das Protokoll ermöglicht eine effiziente Datenübertragung und sichere Zustellung der Daten in instabiles Netz. Da die Zahlen der "Smart-Geräte" stark zugestiegen ist und diese Anforderungen heute immer noch aktuell sind wurde der Source Code 2010 unter einer freien Lizenz veröffentlicht. Heute ist MQTT eines der besten und meist benutzen IoT Protokoll.

4 Warum MQTT?

Es stellt sich die Frage, welches Kommunikationsprotokoll geeignet ist um den Anforderungen des Internet of Things gerecht zu werden. Neben dem Klassiker unter den Internetprotokolle in, HTTP, etabliert sich das Protokoll MQTT immer mehr zu einem IoT-Standard. Das sehr schlanke und leichtgewichtige Kommunikationsprotokoll punktet durch folgende Eigenschaften:

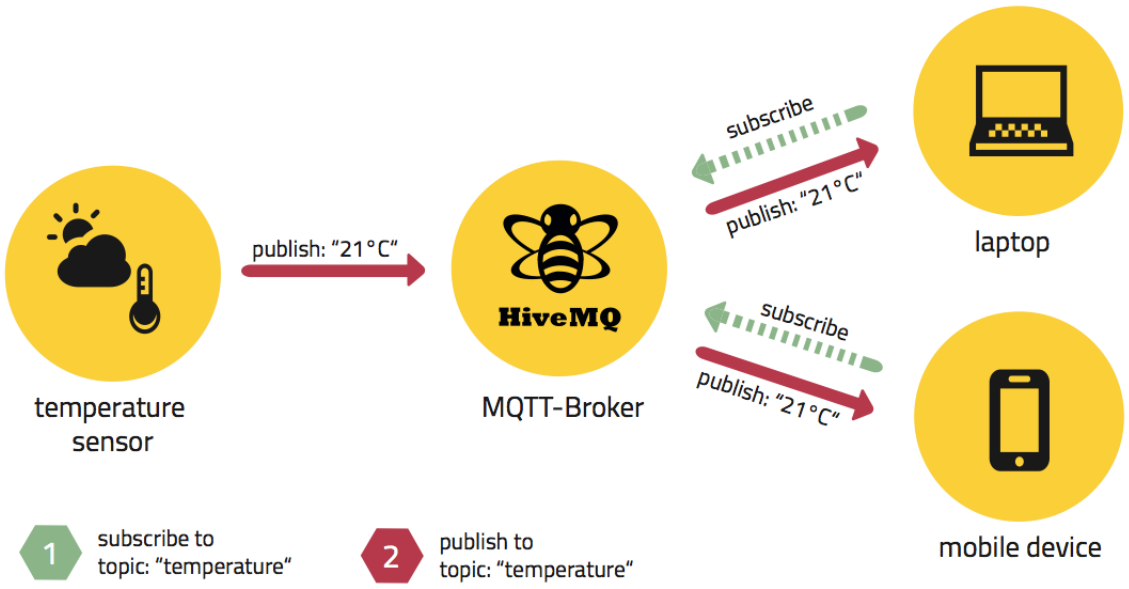
- MQTT ist schlank und leichtgewichtig und sehr einfach zu implementieren.
- MQTT implementiert das Publish / Subscribe Pattern und erlaubt damit echtes Push-Messaging zwischen allen Kommunikationsteilnehmern.
- MQTT gewährleistet durch verschiedene Quality of Service Levels die Datenübertragung in instabilen Netzen, wie etwa Mobilfunknetzen.
- MQTT besitzt einen minimalen Protokolloverhead.
- MQTT ist Session-Aware. Das bedeutet, Metainformationen müssen im Idealfall nur einmal übertragen werden, da der Server die Daten nach einem Verbindungsabbruch weiterhin vorhält.
- MQTT ist datenagnostisch. Es ist daher möglich, Daten in jedem möglichen Format zu übertragen.

All diese Eigenschaften machen MQTT zu einer sehr guten Wahl, da mittels dem Publish / Subscribe Pattern eine hohe Skalierbarkeit erreicht werden kann und mit dem minimalen Overhead und verschiedenen Quality of Service Levels können selbst instabile und teure Kommunikationskanäle wie Mobilfunknetze und Satellitenübertragung kosteneffizient genutzt werden. [1]

[1] https://www.informatik-aktuell.de/betrieb/netzwerke/internet-der-dinge-protokolle-verfahren-und-integration-von-mqtt.html

5 Wie Funktioniert MQTT?

5.1 Publish/Subscribe

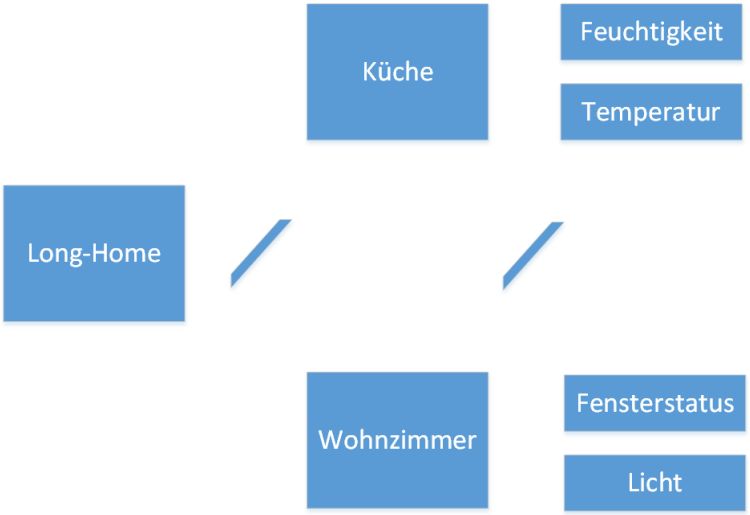


src: Dominik Obermaier http://www.hivemq.com/

MQTT benutzt das Publish/subscribe das heißt Jeder MQTT-Client, der Nachrichten für dieses Topic empfangen möchte, abonniert es beim Message Broker. Da die interessierten Clients beim Eintreffen neuer Nachrichten durch den Broker benachrichtigt werden, anstatt selbst beim Server nach Änderungen zu fragen, wird eine hocheffiziente Kommunikation zwischen den Teilnehmern gewährleistet. Somit ist eine echte Push-Kommunikation möglich. Entscheidend ist, dass die Teilnehmer der Kommunikation nichts voneinander wissen, da jeder Client nur den Message Broker kennt, nicht jedoch die anderen Teilnehmer. [2]

[2] https://jaxenter.de/iot-allrounder-27208

5.2 Topic



Topics sind einfache Zeichenketten welche eine Art Nachrichten-Hierarchie abbilden und mit einer einfachen URL vergleichbar sind. Hier sind paar wichtige Beispiele wie man mit Topics in MQTT benützen kann.

Nur die Temperatur aus Longs Küche

Subscriber: Long-Home/Küche/Temperatur

Alles aus Longs Wohnzimmer

Subscriber: Long-Home/Wohnzimmer/#

Alles aus Long-Home

Subscriber: Long-Home/#

Temperatursensor wert publishen

Publisher: Long-Home/Küche/Temperatur + ”wert”

5.3 Quality of service (QoS)

Da MQTT auf dem TCP Protokoll basiert, kommt es von Haus aus mit einer großen Zuverlässigkeit in Sachen Datenaustausch. Allerdings kann es gerade im mobilen Bereich, zu Verbindungsengpässen oder gar Ausfällen kommen. Zu diesem Zweck wurde das Protokoll um den Quality of Service Mechanismus erweitert, welcher das Übertragen von Nachrichten garantiert. Dabei unterscheidet MQTT in drei Servicequalitäten:

- QoS = 0: fire and forget
- QoS = 1: Nachricht kommt mindestens einmal an
- QoS = 2: Nachricht kommt genau einmal an

Je nach Einsatzgebiet oder Gewichtung der zu versendeten Nachricht, kann der MQTT-Client selbst entscheiden welcher QoS für ihn am geeignetsten ist. Um bei unserem Beispiel mit den Flotten-Fahrzeugen zu bleiben: Nehmen wir an, das Fahrzeug sendet im Sekunden-Takt seine Position an den Broker. Hier wäre es angebracht die Nachricht mit einem Wert von 0 zu versenden, da eine einzelne verloren gegangene Geo-Position verkraftbar ist (da in einer Sekunde bereits die nächste folgt) und dadurch nicht unnötiger Rückkopplungs-Overhead entsteht. Die Servicequalität von 2 wird beim subscriben eines Clients am Broker eingesetzt. Hier ist es wichtig, dass die Abonierung nur ein einziges mal ankommt und der Client sich nicht versehentlich doppelt registiert, wenn z. B. die Rückantwort des Servers verloren ging. [3]

5.4 Last Will Testament (LWT)

MQTT Clients können beim Verbinden mit dem MQTT Broker eine Nachricht als sog. "Last Will and Testament" angeben. Falls nun der Client unerwartet die Verbindung zum MQTT Broker verliert, sendet der Broker diese Nachricht an alle Subscriber. Damit ist es möglich, interessierte Clients mit einer Nachricht zu informieren, dass ein anderer Client nicht mehr online ist. Sehr häufig wird dieser Mechanismus verwendet um schnell auf Probleme einzelner Clients reagieren zu können, da man sofort ein Feedback bekommt, falls es Störungen gibt.

```
1 client = mqtt.Client(client_id="1234")
2
3 # so konfigurioert man LWT
4 client.will_set(topic, payload="Unexpected disconnect", qos=0,
5 retain=False)
```

[3] http://www.sic-software.com/das-mqtt-protokoll-1/

```
6  client.connect("broker.mqttdashboard.com", 1883, 60)
7
8  # do work
9
10 client.disconnect()
```

6 Python und MQTT

Eclipse Paho ist ein Projekt unter dem Schirm der Eclipse Foundation, das MQTT-Clientimplementierungen in verschiedenen Programmiersprachen zur Verfügung stellt. Das sind neben C die folgenden: C++, C#, Python, JavaScript, Go und natürlich auch Java. Das Paho-Projekt existiert seit 2012, die Java- und C-Implementierungen werden jedoch seit vielen Jahren bei IBM, die auch die initiale Implementierungen bereitgestellt haben, eingesetzt. Sie sind für den produktiven Einsatz freigegeben. Neben einem synchronen API bietet Paho auch ein asynchrones API, das mehr Performance verspricht, dafür etwas komplizierter in der Benutzung ist. In den folgenden Beispielen wird das synchrone Paho API verwendet.

6.1 So installiert man MQTT für Python

paho-mqtt ist kompatibel mit Python 2.7 und 3.X und kann man ganz einfach installieren von Terminal oder Konsole mit dem foldenden Befehl.

Installation

```
$ pip install paho-mqtt
```

6.2 Test Programme

Zum testen habe ich 2 MQTT Broken benütz. Bei Mqttdashboard.com kann man MQTT Applikationen kosten frei testen. Und der 2. Broker habe ich Lokal auf mein Raspberry Pi 3 installiert.

Subscribe:

In diesem Beispiel Abonniere ich das Topic "testtopic/thee" und gebe ich alle kommende Nachrichte auf dem Bildschrim

```
import paho.mqtt.client as mqtt

def on_connect(client, userdata, rc):
    print("Connected with result code "+str(rc))
    client.subscribe("testtopic/thee")

#Beim erhalten einer Nachricht, gibt das Programm das gesammte-
#Payload auf der Konsole

def on_message(client, userdata, msg):
    print(str(msg.payload))

client = mqtt.Client(client_id="1234")
client.on_connect = on_connect
client.on_message = on_message

client.connect("broker.mqttdashboard.com", 1883, 60)

client.loop_forever()
```

Publish:

Hier publiziere ich zufällige temperatur Werte mit einem intervall von 5 Sekunden an "test/topic"

```
import time
import random

import paho.mqtt.client as mqtt

def on_connect(client, userdata, flags, rc):
    print("Connected with result code " + str(rc))

client = mqtt.Client()
client.on_connect = on_connect

client.connect("localhost", 1883, 60)

client.loop_start()

while True:
    time.sleep(5)
    temp = random.randrange(-20, 40, 3)
    client.publish("test/topic", "Temperature = %d" % temp)
```

7 Quellen

- http://mqtt.org/
- https://pypi.python.org/pypi/paho-mqtt/1.2
- https://www.dinotools.de/2015/04/11/mosquitto-als-mqtt-broker/
- https://www.dinotools.de/2015/04/12/mqtt-mit-python-nutzen/
- https://www.informatik-aktuell.de/betrieb/netzwerke/internet-der-dinge-protokolle-verfahren-und-integration-von-mqtt.html
- http://www.sic-software.com/das-mqtt-protokoll-1/