

Webとの連携

Internet of Things

- モノのインターネット
- 人主体のインターネットから, モノ主体のインターネットへ. モノ同士の相互連携, ビッグデータ



Internet of Things

640 Companies

Contact info@venturescanner.com to see all



Venture Scanner

IoTの通信方式

- HTTP:
- TCP:
- MQTT (Message Queue Telemetry Transport)
- OSGi
- Echonet-lite

MQTT

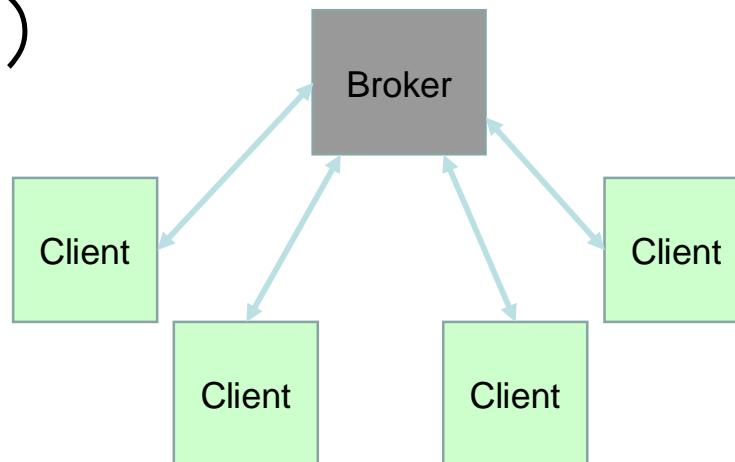
- IBMの提唱する, 軽量プロトコル
- Publish/Subscribeモデル
- TCP/IPの上で動作 (HTTPと同層?)
- HTTPよりかなり軽量. ビッグなデータを山ほどやりとりしても大丈夫

MQTT特徴

- Topic
 - 購読する項目を指定する
 - スラッシュ区切りの階層構造
 - 例えば, room1/sensor/temperatureなど,
 - ワイルドカード
 - #:前方一致:a/# なら, a/b, a/c, a/r/kすべて購読できる
 - +:部分一致:a/+bなら, a/x/b, a/y/bに一致. a/z/cは一致しない.
- QoS/到達保証
 - QoS 0: At most once -- 最高1回。届くかは保証しない。
 - QoS 1: At least once -- 少なくとも一回。重複する可能性がある。
 - QoS 2: Exactly once -- 正確に一回
- Will/遺言
 - 最後にPublishされた内容を保持しておき, Subscribeされた際に
- Retain/保持
- Security/認証と暗号化

MQTT構成

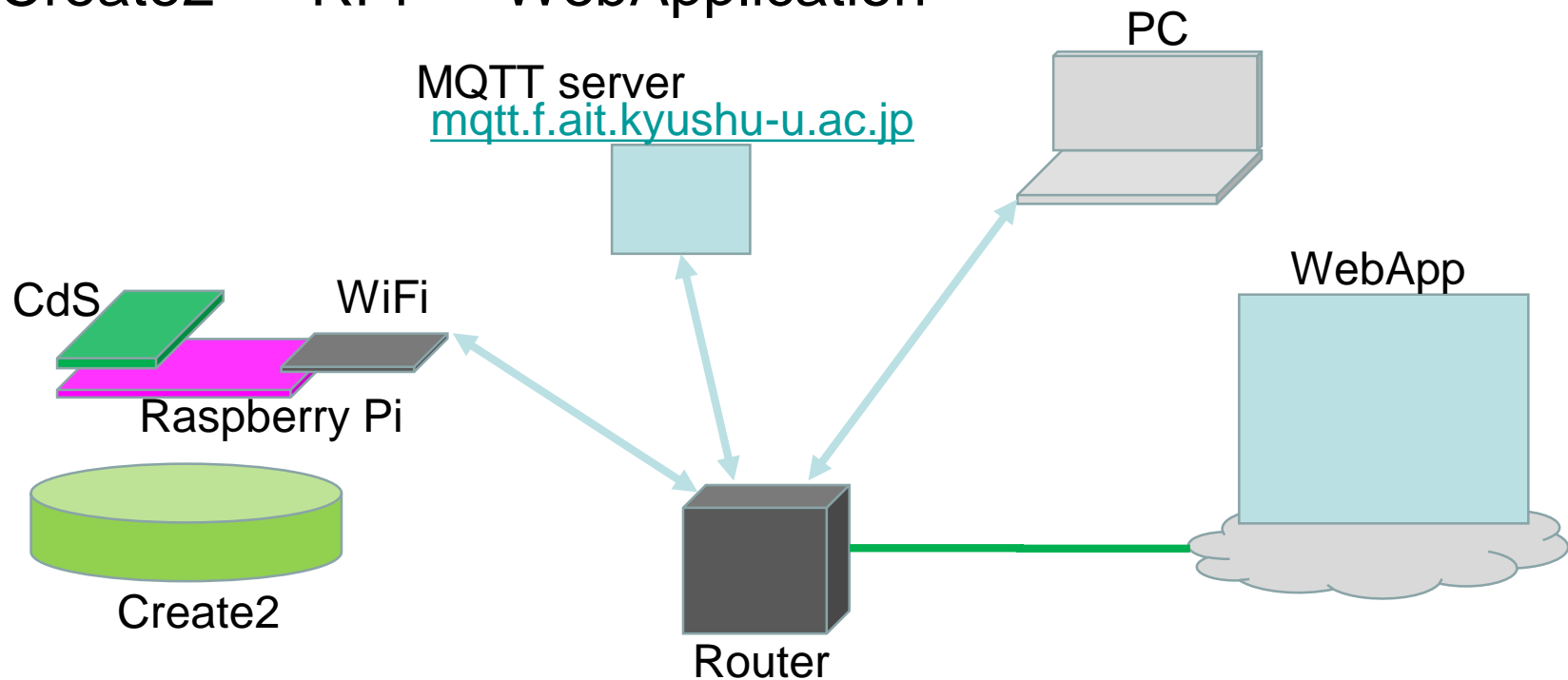
- 構成要素はBrokerとClient
- 通信は一旦すべてBrokerに送られる.
(ClientはBrokerのIPアドレスだけ知っていればよい)
- ClientはPublisherまたはSubscriber(両方同時も可能)



演習

今回のシステム構成

- PC \Leftrightarrow PC
- RPi \Leftrightarrow PC
- RPi \Leftrightarrow WebApplication
- Create2 \Leftrightarrow RPi \Leftrightarrow WebApplication



PC ⇔ PCのMQTT接続

- Mosquitto

PC用のMQTTクライアント, サーバ

- Publish

赤字を各自の情報に変更してください.

```
mosquitto_pub -t /pbl1/unitN/name -h  
mqtt.f.ait.kyushu-u.ac.jp -m message
```

-t トピック名を指定 -h ホストを指定 -m メッセージ

- Subscribe

```
mosquitto_sub -t /pbl1/unitN/# -h  
mqtt.f.ait.kyushu-u.ac.jp
```

+, # はワイルドカード, 例えば下記
ならユニットNに届く
すべてのメッセージを
購読できる

ワイルドカード

- # : この後ろはなんでも

/pbl1/hoge/#

– /pbl1/hoge/aaa

– /pbl1/hoge/bbb/cc

– /pbl1/foo/aaa

- + : この部分はなんでも

/pbl1/+/aaa

– /pbl1/unit1/aaa

– /pbl1/unit2/aaa

– /pbl1/unit1/bbb

Will/Retain/QoS

QoSはテストが難しいので今回は割愛

- Will
 - subscriberに以下を付けて起動.
--will-topic *topic* --will-payload *dyingmessage*
 - 別のsubscriberで上記のtopicを購読しておき,
上記のsubscriberを強制終了
- Retain
 - publisherに-rを付けてメッセージ送信.
 - 送信後にsubscriberで購読する

PC \Leftrightarrow Raspberry Pi

Raspberry Pi MQTT Libraryのインストール

- チームで一人だけ実行してください
- `sudo pip install paho-mqtt`

サンプルプログラム

各ユーザのディレクトリに適当な名前で作成して実行してください。

```
#!/usr/bin/python
import paho.mqtt.client as mqtt
import time

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

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

client = mqtt.Client()
client.on_connect = on_connect
client.on_message = on_message
client.connect("mqtt.f.ait.kyushu-u.ac.jp", 1883, 60)
client.loop_start()
while True:
    client.publish("/pbl1/unitN/name", "hello")
    time.sleep(1)
```


サンプルプログラム2

各ユーザのディレクトリに適当な
名前で保存して実行してください。

```
#!/usr/bin/python
import paho.mqtt.client as mqtt
import time
import 前回作った光センサを読み取るプログラム
def on_connect(client, userdata, flags, rc):
    print("Connected with result code "+str(rc))
    client.subscribe("/hello")
def on_message(client, userdata, msg):
    print(msg.topic+" "+str(msg.payload))

client = mqtt.Client()
client.on_connect = on_connect
client.on_message = on_message
client.connect("mqtt.f.ait.kyushu-u.ac.jp", 1883, 60)
client.loop_start()
while True:
    light_level = ReadChannel(0)
    client.publish("/pbl1/unit5/light",light_level)
    time.sleep(1)
```

PC \Leftrightarrow Raspberry Pi

サンプルWebアプリケーションのClone(1)

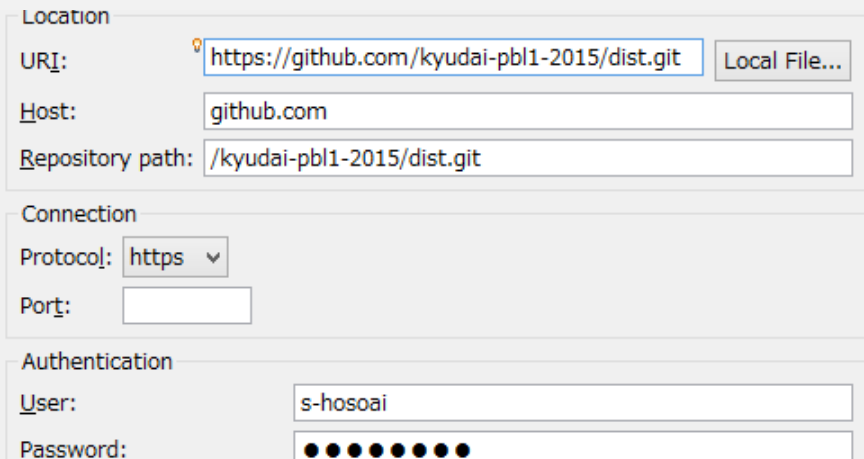
1.この辺で右クリック
⇒Import

2.Project from Git
⇒Next⇒
Clone URI

3.下記の通り入力

<https://github.com/kyudai-pbl1-2015/dist>

のURLをコピーした状態で開くと勝手に埋まる



Location

URI: Local File...

Host:

Repository path:

Connection

Protocol:

Port:

Authentication


User:

Password:

3.Authentication

は鍵の扱いが未確認なのでとりあえずHTTPS
で認証してください

各ユーザ情報を入力

 < Back Next > Finish Cancel

サンプルWebアプリケーションのClone(2)

