사물인터넷 실습 :

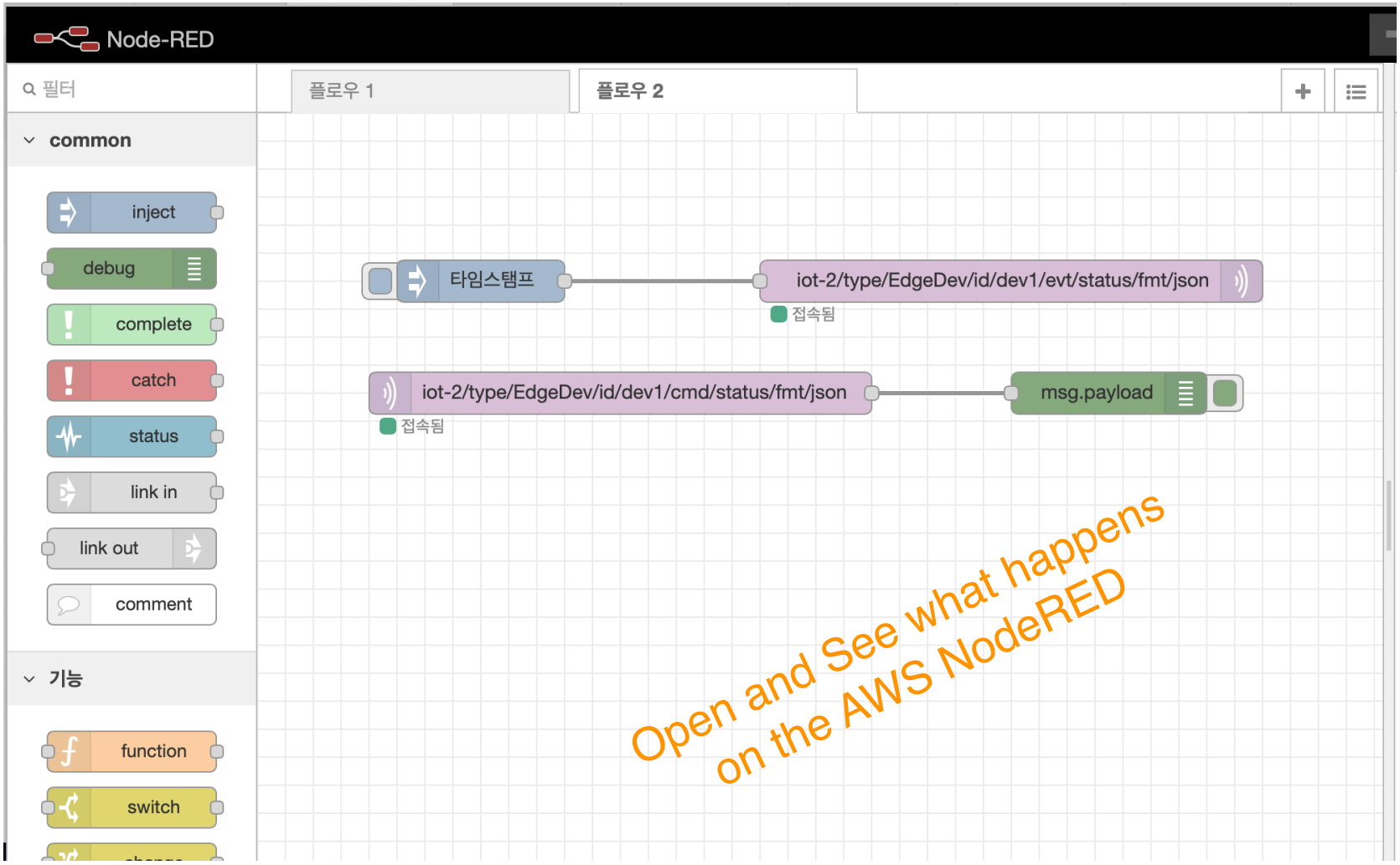
IBMIOT Device Code Porting for Edge Computing

Electronics Everywhere
Network Everything
Cloud Intelligence

허윤석, 공학박사

email: yoonseok@gmail.com

Lab 2 - NodeRED Edge Object



Lab 3 - Python Edge Object

- sudo apt install python-pip
- pip install paho-mqtt
- Create a Python program to talk to mosquitto
- Explore the IBM IOT Console
- Create a NodeRED Flow on AWS Cloud
- Create another NodeRED flow on Edge Server

IBM IOT Python Edge Device

```
import paho.mqtt.client as mqtt
import RPi.GPIO as g
import time
import json
from time import sleep

g.setmode(g.BCM)
g.setup(14, g.OUT)

def millis():
    return int(round(time.time() * 1000))

def on_connect(client, userdata, flags, rc):
    print("Connected " + str(rc))
    client.subscribe("iot-2/type/Python/id/dev2/cmd/status/fmt/json")

def on_message(client, userdata, msg):
    m = json.loads(msg.payload)
    if m['d']['led'] == 'on':
        g.output(14, g.HIGH)
    else:
        g.output(14, g.LOW)
    print(msg.topic+" " +str(msg.payload))

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

client.connect("127.0.0.1", 1883, 60)

pubInt = 3000
lastPub = 0

cnt = 0
evt = json.loads('{"d":{}}')

while True:
    client.loop()
    if millis() - pubInt > lastPub:
        lastPub = millis()
        cnt = cnt + 1
        evt['d']['count'] = cnt
        client.publish("iot-2/type/Python/id/dev2/evt/status/fmt/json", json.dumps(evt))
```

IBM IOT Platform Device Topics

- MQTT Topics and the Devices

```
iot-2/type/+/id/+/evt/status/fmt/json
      vs.
iot-2/evt/status/fmt/json
```

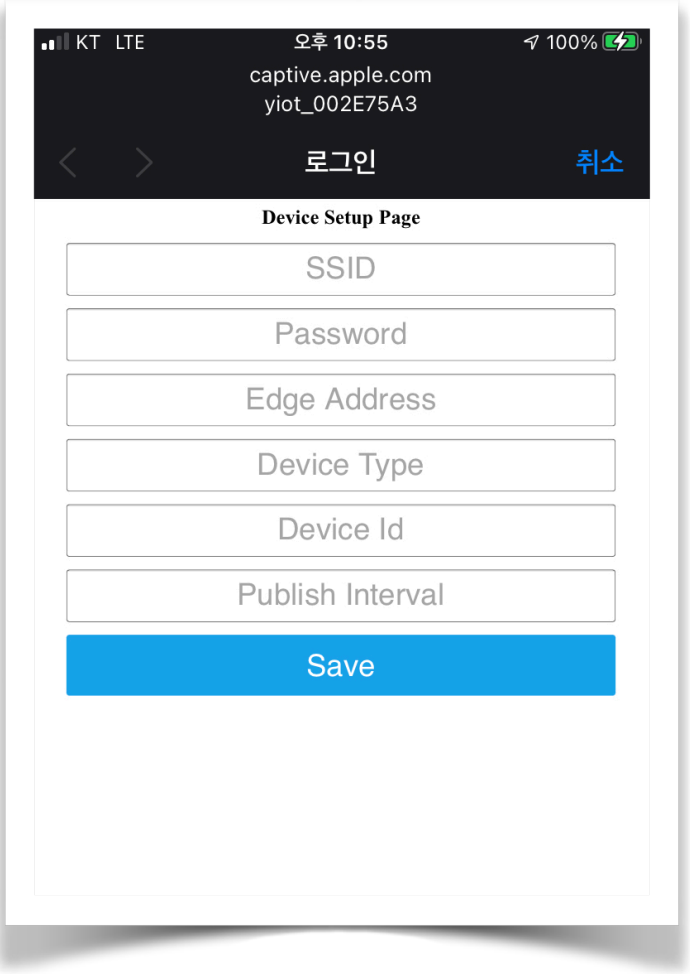
- Topics walkthrough

https://www.ibm.com/support/knowledgecenter/SSQP8H/iot/platform/devices/mqtt.html

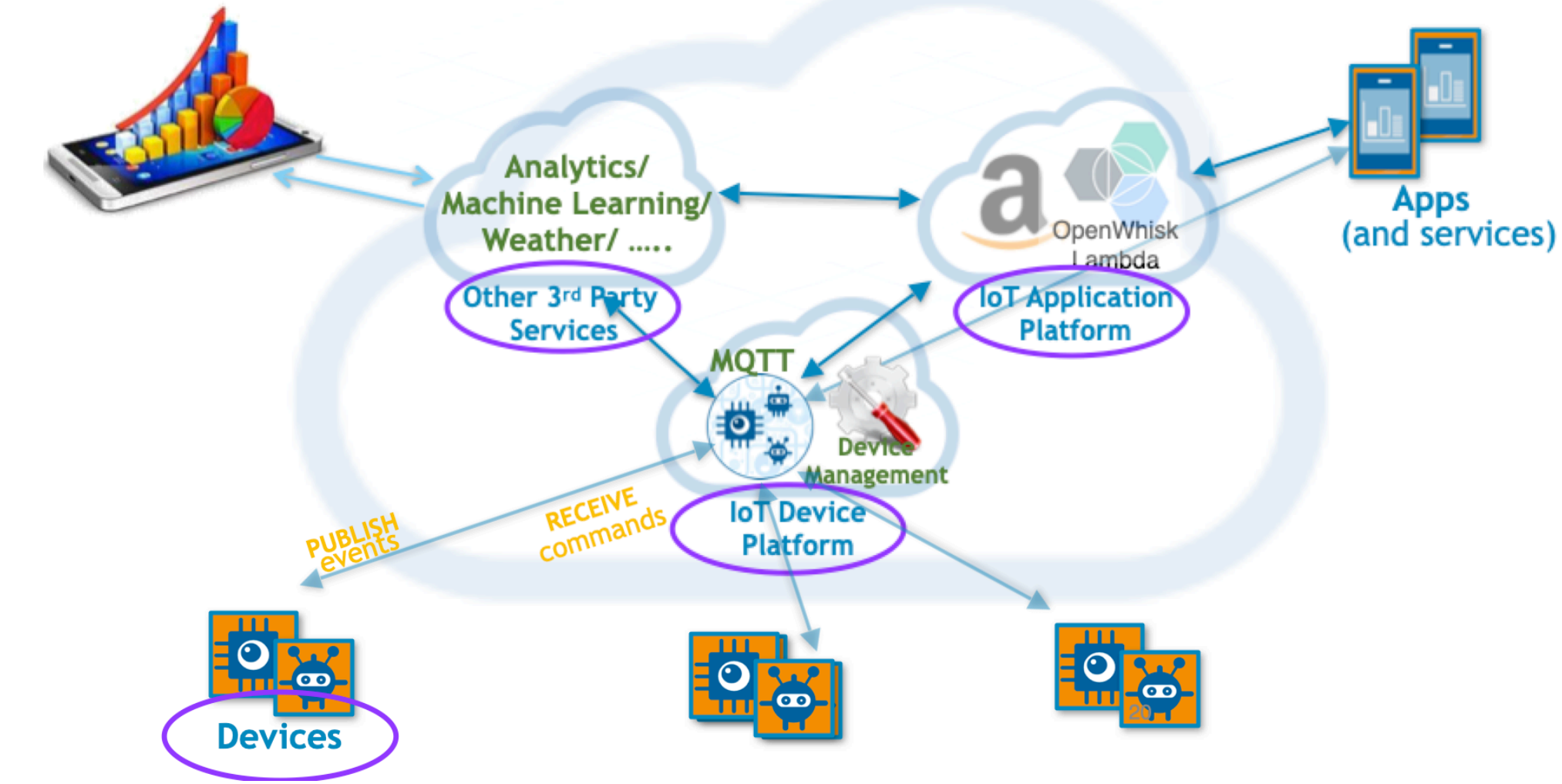
https://www.ibm.com/support/knowledgecenter/SSQP8H/iot/platform/gateways/mqtt.html

Lab 4 - Modify IBMIOTDevce.h to talk to Edge Server

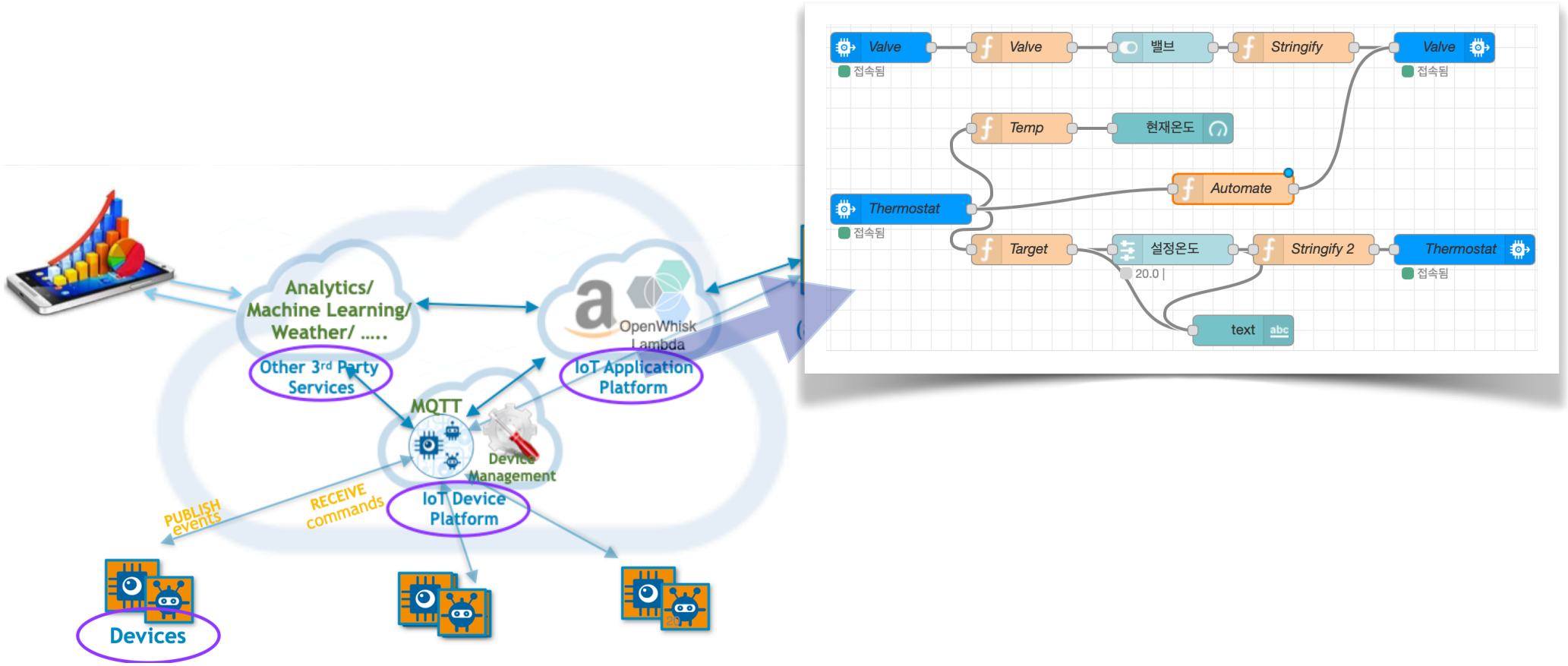
- Modify the IBMIOTDevice.h to talk to Edge Server
  - Subscribe/Publish Topics
  - WiFiClientSecure -> WiFiClient
  - remove fingerprint related logic
  - remove ‘org’ related code
- Recompile the ESP8266 Thermostat and Valve
- Verify the exiting IOT Thermostat Dashboard



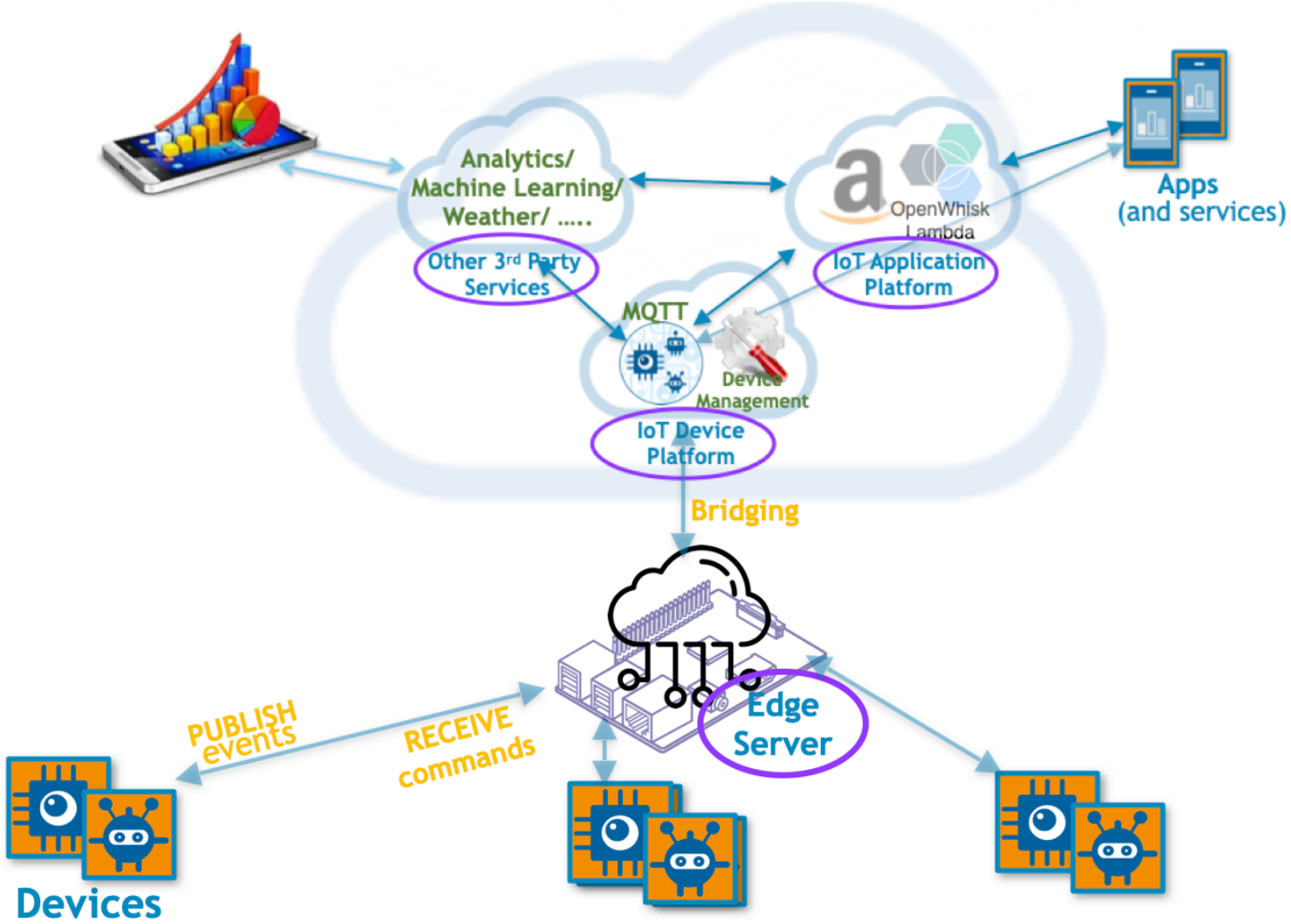
No Edge Internet of Things Architecture



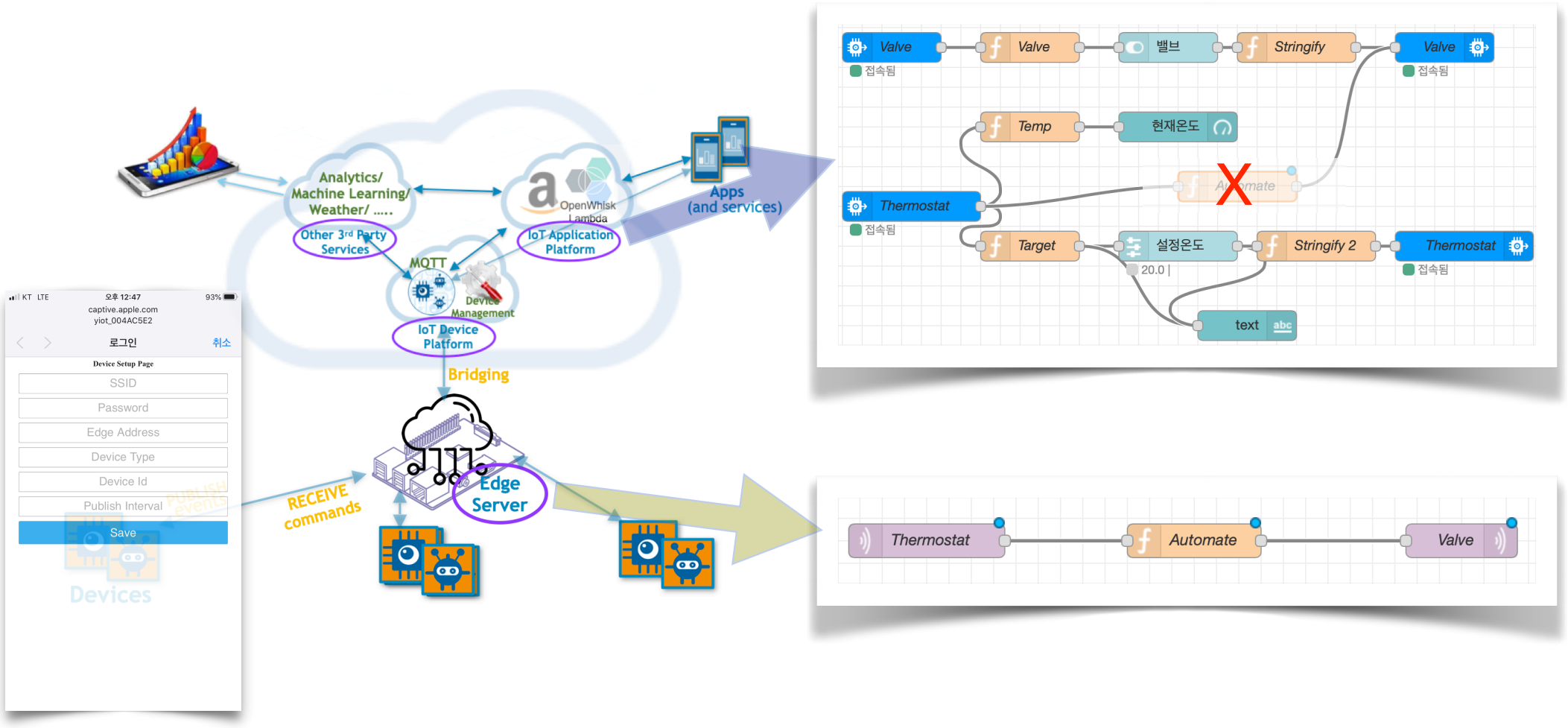
No Edge Internet of Things Architecture



Edge Internet of Things Architecture



Lab 5 - Cloud Intelligence and Edge Intelligence



Backup

참고 : IBM IOTF Python Module

```
import wiotp.sdk
import RPi.GPIO as g
from signal import pause

deviceOptions = {
    "identity": {"orgId": "ooo", "typeId": "RPi", "deviceId": "iotDev1"},
    "auth": {"token": “rpi11111"},
}
data = {
    "d" : {
    }
}

def commandProcessor(cmd):
    print(cmd.data["d"])
    if cmd.data["d"]["lamp"]:
        if cmd.data["d"]["lamp"] == "on":
            g.output(14, g.HIGH)
            data["d"]["lamp"] = "on"
        else:
            g.output(14, g.LOW)
            data["d"]["lamp"] = "off"
        deviceCli.publishEvent("status", "json", data, qos=0)

g.setmode(g.BCM)
g.setup(14, g.OUT)
deviceCli = wiotp.sdk.device.DeviceClient(deviceOptions)
deviceCli.commandCallback = commandProcessor
deviceCli.connect()
pause()
```