

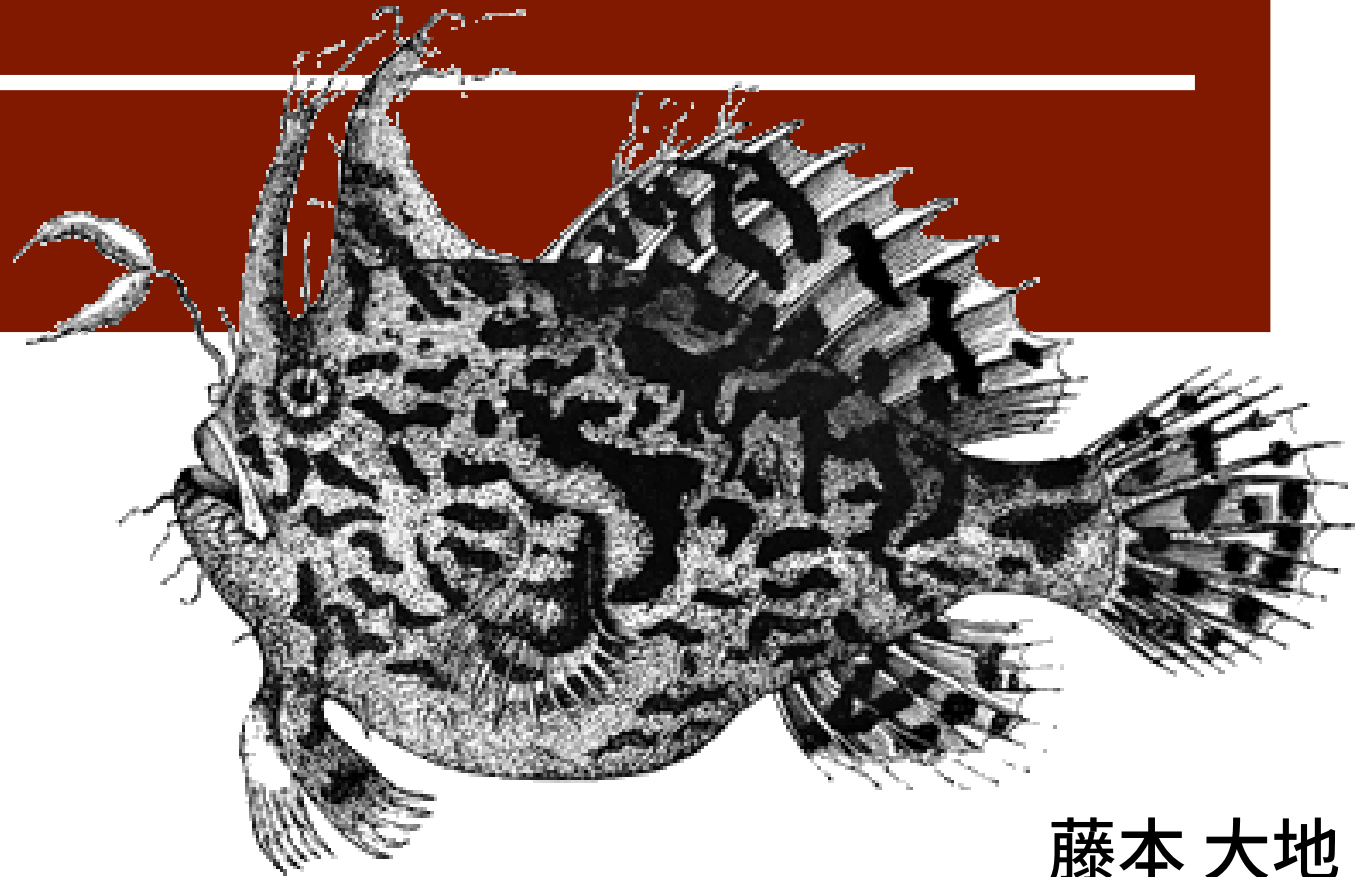
E2F

組み込みエンジニアリングフォーラム

MQTTで誰でも簡単IoT

IoTって結局なにすればええの？

実装編



藤本 大地

お題目

- 自己紹介
- ターゲット
- MQTTとは
- MQTTの実装

自己紹介

- 職業：ファームウェアエンジニア
- 趣味：
 - 電子工作
 - アウトドア

ターゲット

- IoTの下回りを支える技術に興味がある
- そもそもIoTってなんなんだよ！
- 何で今IoTがアツイの？

準備

- mqtt-spyのダウンロード
 - <https://kamilfb.github.io/mqtt-spy/>
 - ※ 要Java

このあたりの話

7 Layers of the Internet of Things (IoT)

Business Value

People & Process

Layer 7 – Transformational decision making based on “Thing” Apps & Data

Applications

Layer 6 – Custom Apps built using “Thing” data

Big Data

Data Analysis

Layer 5 – Reporting, Mining, Machine Learning

Data Ingestion

Layer 4 – Big Data, Harvest & storage of “Thing” data

Cloud

Global Infrastructure

Layer 3 – Cloud infrastructure (public, private, hybrid, managed)

Fog

Connectivity/Edge Computing

Layer 2 – Communications, Protocols, Networks, M2M, Wifi, Telecom, HW Kits

Things

Layer 1 - Devices, sensors, controllers, etc.

MQTTとは

- 公式 <<http://mqtt.org/>>
- MQ Telemetry Transportの略
 - MQはMessage Queueに非ず
 - MQにキューイング機能はない

元々IBMが開発、IBM MQの一連のシリーズ

- ということでMQがついている（プリフィックス）

でも実はIBM MQのMQはMessage Queueの略
言いたいのはプリフィックスのMQであって
Message QueueのMQではないということ

MQTTとは

- TCP/IPをベースとするM2M通信仕様
 - HTTPと比べてヘッダ情報が小さいため、比較的小さなデータを複数回送る場合に適している

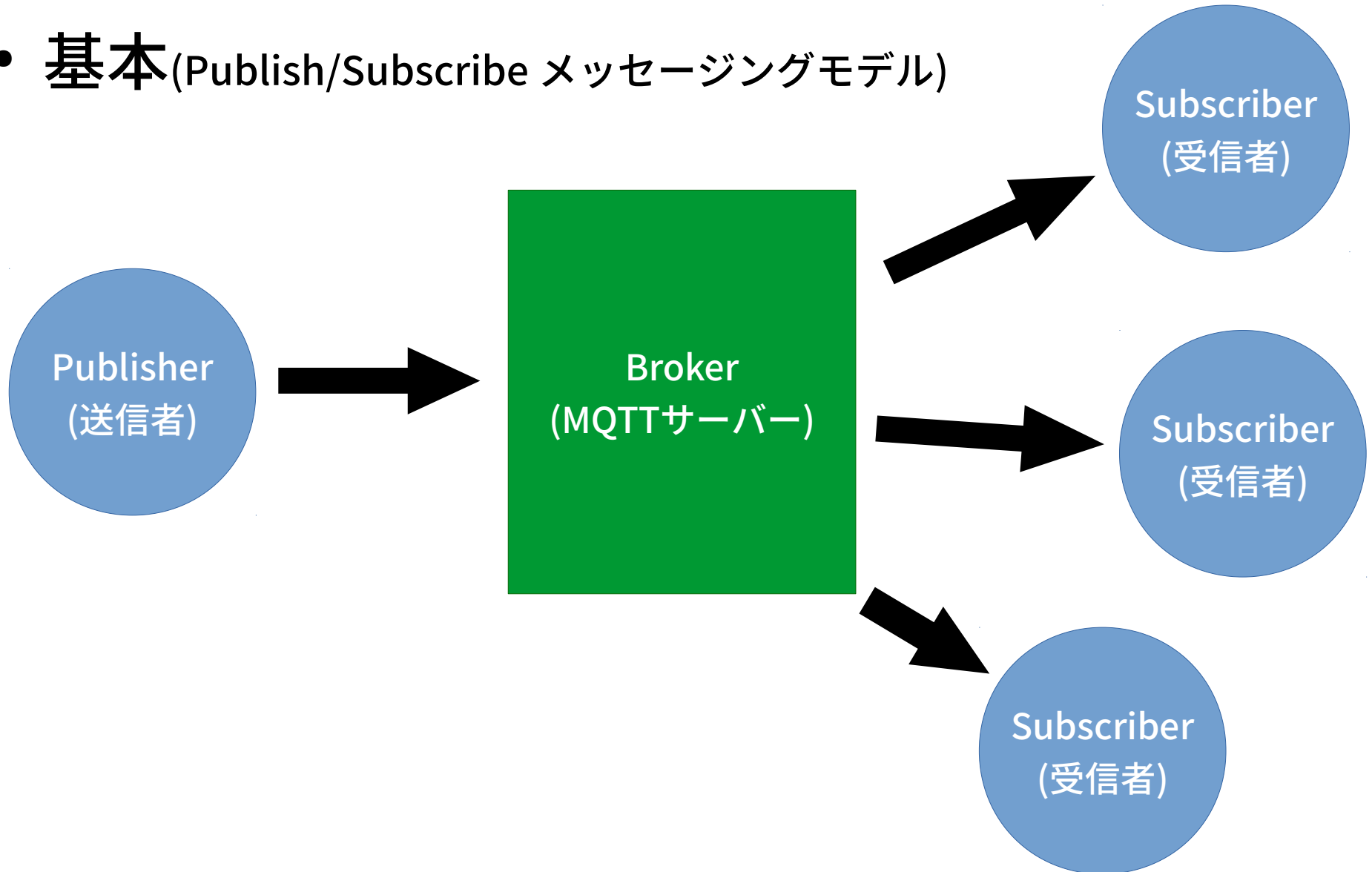
MQTTをひとことで

IoTでいうなら

デバイスどうし、デバイスとサーバーで
少量のデータを送りあう仕組み

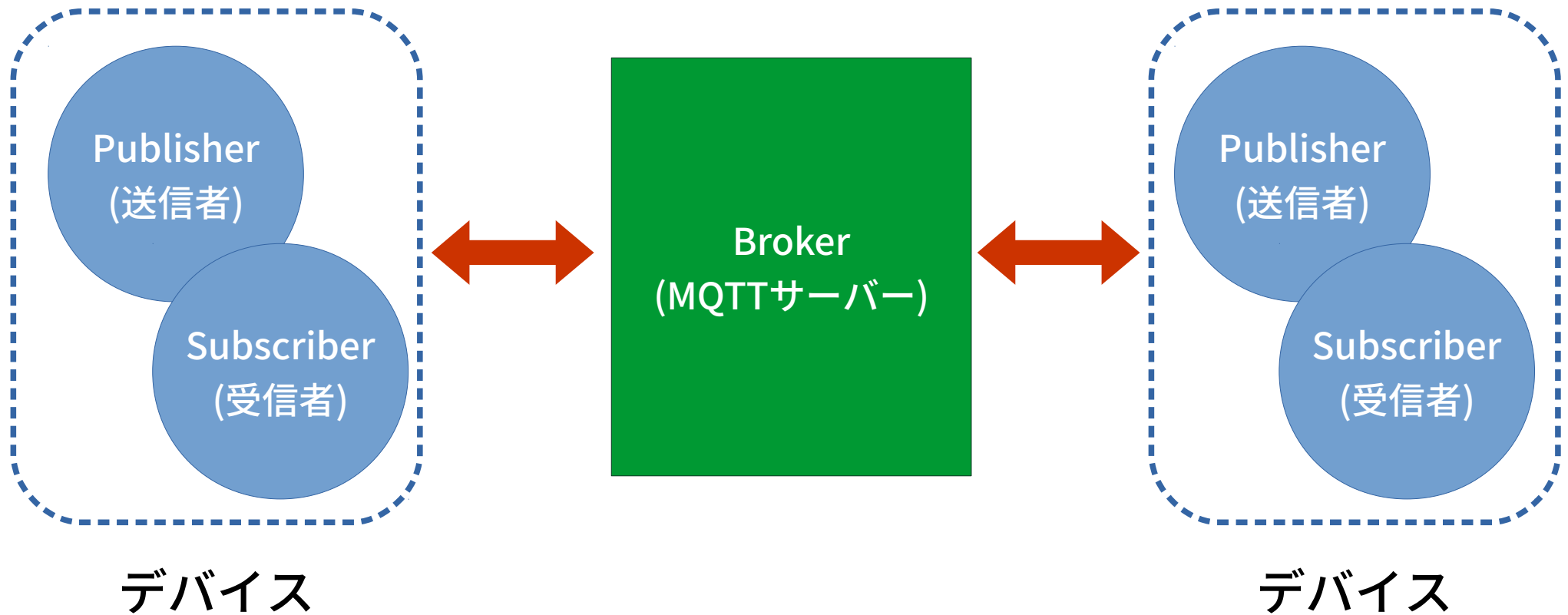
しくみと役割

- **基本**(Publish/Subscribe メッセージングモデル)



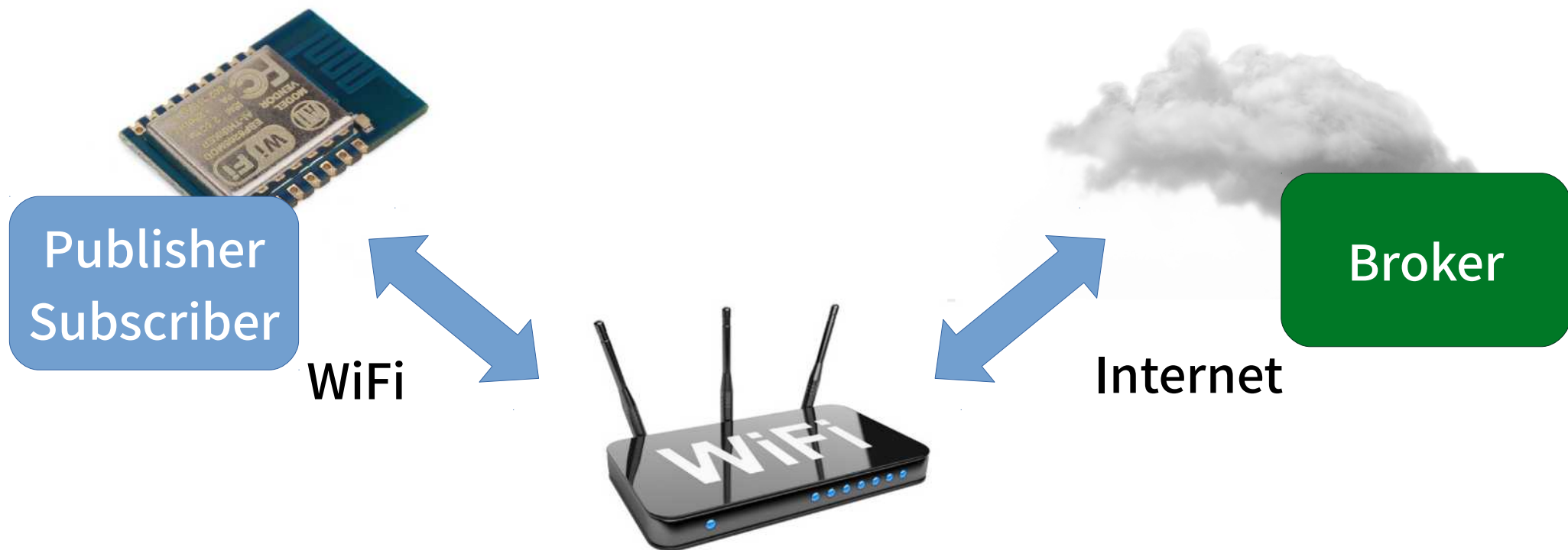
しくみと役割

- こういうのもあり



どうやってやるの？

- ESP8266を使ったちょー簡単MQTT入門



ESP8266開発環境

- ESP8266
 - USB-Serial付きでFW書き込み可能なものが便利
 - <https://www.switch-science.com/catalog/2500/>
 - <http://www.runele.com/ca1/40/p-r-s/>
- Arduino
 - ボードマネージャ: esp8266
 - <https://github.com/esp8266/Arduino>
 - ライブラリ: PubSubClient

実装ステップ

- Step1: WiFiにつなぐ
- Step2: ブローカーに接続する
- Step3: Pubする
- Step4: Subする

topicを階層化

- e2f_mqtt_test
 - 温度 : "e2f_mqtt_test/sensor/temperature"
 - 湿度 : "e2f_mqtt_test/sensor/humidity"
 - 全部 : "e2f_mqtt_test/sensor/#"

DEMO

- 会場の温度・湿度をPub/Subする

まとめ

- ワンコインのデバイスで簡単にPub/Subできた
 - 無料のお試しブローカーがたくさん
 - 誰でも簡単IoT
- データを上げる！はできた
 - 貯める、解析するのはこれから