WHITE PAPER

Wouter van der Beek
CASCODA
w.vanderbeek@cascoda.com

OCF and MQTT

September 30, 2022

Introduction

The OCF Secure IP Device Framework is cloud enabled via HTTP and CoAP. However, there is another IP based protocol that is very popular in many industries, Message Queuing Telemetry Transport (MQTT). The MQTT protocol is publish-subscriber based, and the OCF protocols are REST based. With the latest version of the MQTT spec, MQTT can also be used as a request-response based protocol, similar to REST. This paper outlines how OCF can be used in conjunction with MQTT.

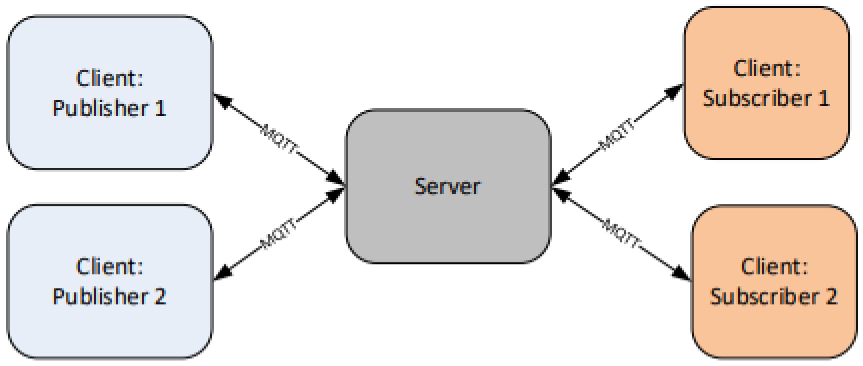


Figure 1: MQTT solution with Publisher and Subscribers talking to an MQTT server

Why?

The MQTT protocol is used in various domains such as automotive, logistics, manufacturing and transportation (a full list of MQTT use cases can be found here). It is therefore logical for OCF devices to feed in to the MQTT domain. MQTT is a transport protocol and does not specify device hierarchy and payload contents. This is reflected in the OCF specification.

The MQTT domain is highly suited to data conversion and has existing low code tooling (Node-Red) to convert and visualize data. MQTT has already been used to connect to data lakes from different vendors such as Amazon AWS, IBM Cloud, Microsoft Azure and Oracle.

Connecting to MQTT enables users to make use of tools such as Node-Red and Grafana to display data. Other MQTT uses include using databases such as MySQL.

MQTT is also used on the eco-system level, for example OPC-UA and sparkplug, for industrial internet of things (IIOT). In Building Automation (BIoT) various eco-systems have (vendor specific) connectors to MQTT, including BACNet, DALI, DMX and KNX.

Vendors are using MQTT for their communication between their products. For example, Cisco Meraki products are using MQTT.

These wide-ranging use cases of MQTT demonstrate the value OCF brings through its interoperability with MQTT as a transport.

Technical

Technically one can view MQTT as a mechanism where data can be easily exchanged without the burden of security. The security in MQTT is on socket level (TLS), but all data on all MQTT clients is in the clear. Since MQTT does not prescribe the topics and the payload, a system designer must design the topic structure and payload definitions. However, in existing systems these are already defined. OCF defines topic and payload structures. A positive for MQTT is the (open source) MQTT clients available in various languages like python, C, C#, Go etc. This means there is a productive and suitable language available for any task.

Node-Red

Node-RED is a programming tool for wiring together hardware devices, APIs and online services in new and exciting ways. It also provides a browser-based editor, making it easy to wire.

As MQTT does not provide topics & payload definitions, each deployment of MQTT must create its own. Node-Red is a visual programming language that can send and receive data on MQTT topics and can be used to simplify definition of these payloads. Node-Red also has a set of libraries to manipulate data and another set to visualize data for dashboard purposes.

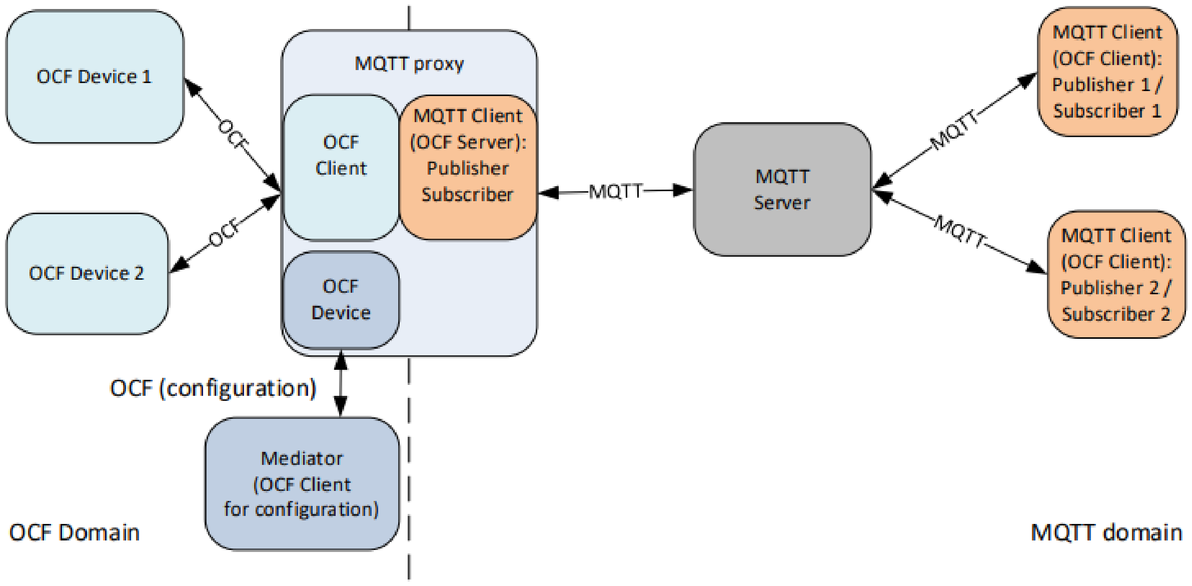
Enabling the OCF specification to use MQTT also ensures Node-Red can be used to display information from OCF devices.

Standardized solution

As OCF is creating ISO/IEC specifications the following 2 additions have been made to the OCF specifications:

1. Standardizing how OCF devices are communicating over MQTT.

2. How existing OCF installations can be linked with an MQTT Server (broker).



MQTT proxy
OCF Device 1
OCF Device 2
OCF
OCF
OCF Client
MQTT Client (OCF Server): Publisher Subscriber
OCF Device
MQTT
MQTT Server
MQTT
MQTT
MQTT Client (OCF Client): Publisher 1 / Subscriber 1
MQTT Client (OCF Client): Publisher 2 / Subscriber 2
OCF (configuration)
Mediator (OCF Client for configuration)
OCF Domain
MQTT domain

Figure 2: MQTT proxy: one side talking CoAP to OCF devices and the other side talking MQTT

The MQTT proxy has the function to make the connection between the OCF and MQTT worlds.

The MQTT proxy implements the following functions:

1. The translation of the REST Create Read Update Delete and Notify (CRUDN) messages with payload to MQTT topics and setting up the connection (with security) between OCF and MQTT. The CRUDN is translated to a topic and the New MQTT5.0 feature of return topic is used to send the response.

2. The topic is formatted to convey the device identifier, resource URL and the CRUDN message identification. The request and response payloads are identical to the CoAP world: e.g. payload in CBOR.

The OCF Client in the MQTT proxy talks to the OCF Devices on the network via CoAP. The MQTT client in the MQTT proxy publishes and subscribes data to the predefined topics and represents the proxied OCF devices in the MQTT world.

End2End System

An end-to-end system can now comprise of:

- OCF Sensors/Actuators running on a Thread network
- MQTT proxy (to OCF) running on a Thread border router
- MQTT Server running in the cloud
- MQTT clients running anywhere to use the data

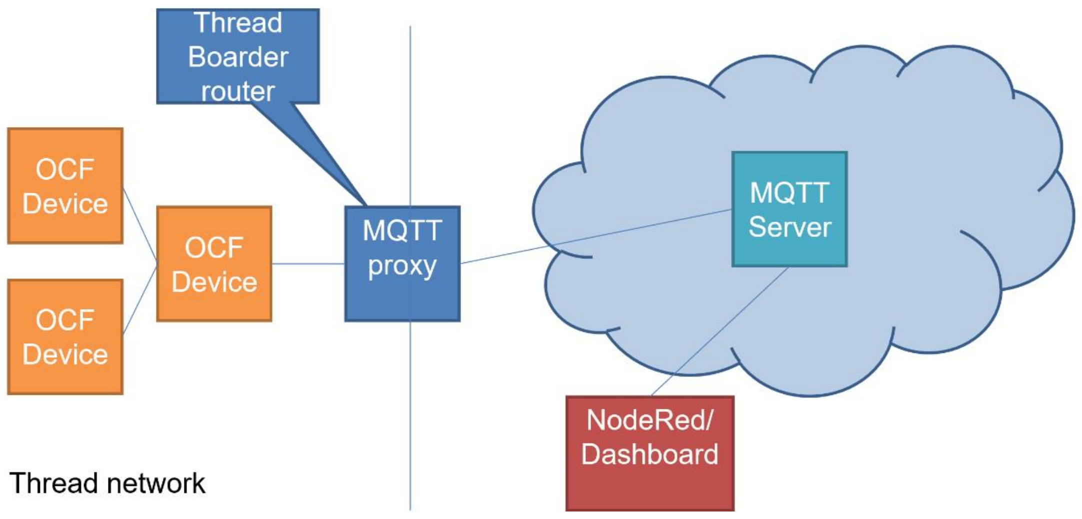


Figure 3: The different protocols working as a system

Technical Information

The Thread border router is running the MQTT proxy as an extra service.

The OCF onboarding tool (OBT) can configure the MQTT proxy to demonstrate:

- Which OCF devices needs to be configured

- MQTT configuration data

- OCF credentials, e.g. who will be able to talk to the MQTT proxy from the OCF domain.

The OBT determines (by configuring) the MQTT proxy which OCF devices should be proxied to the MQTT domain.

The OBT can configure: the MQTT client in the MQTT proxy; the configuration to talk to the MQTT server: e.g. the URL of the server; which security credentials are used to set up a secure TLS connection between the MQTT proxy and the MQTT server. This means that the MQTT proxy can be headless (no specific UI needed to enter this information).

Demonstrator

To demonstrate this new suite of technologies, Cascoda, an OCF and Thread member company, has built the below system (fig 3).

A fully functional MQTT proxy and a dashboard in Node-Red has been created, where the Node-Red flow interacts with the OCF devices as per OCF specifications for topics and payloads.

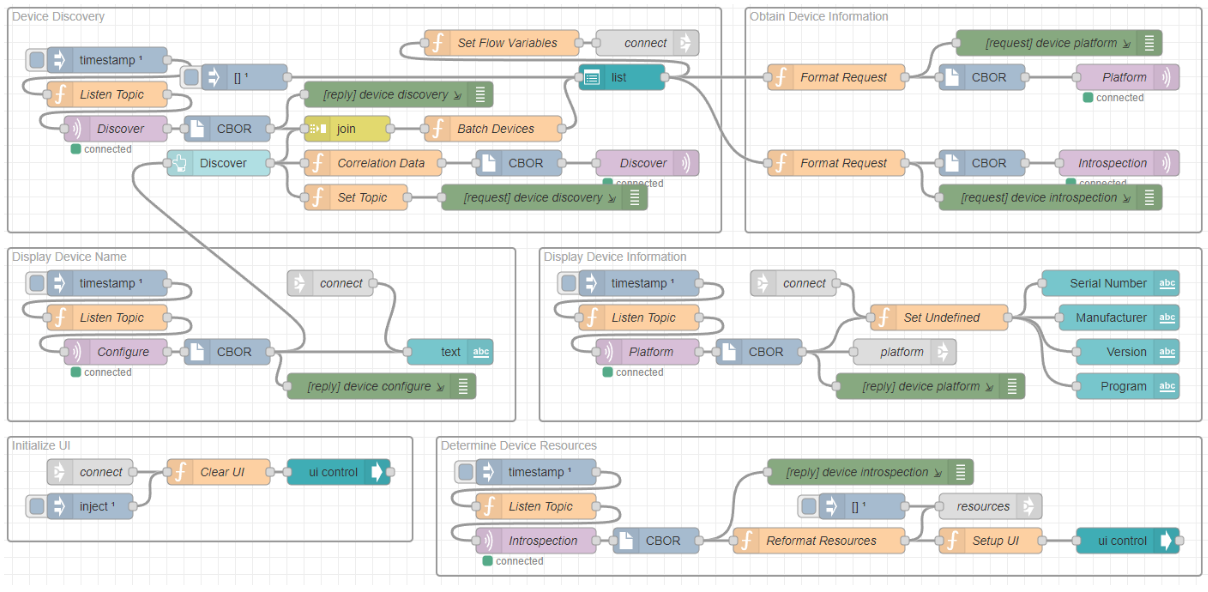
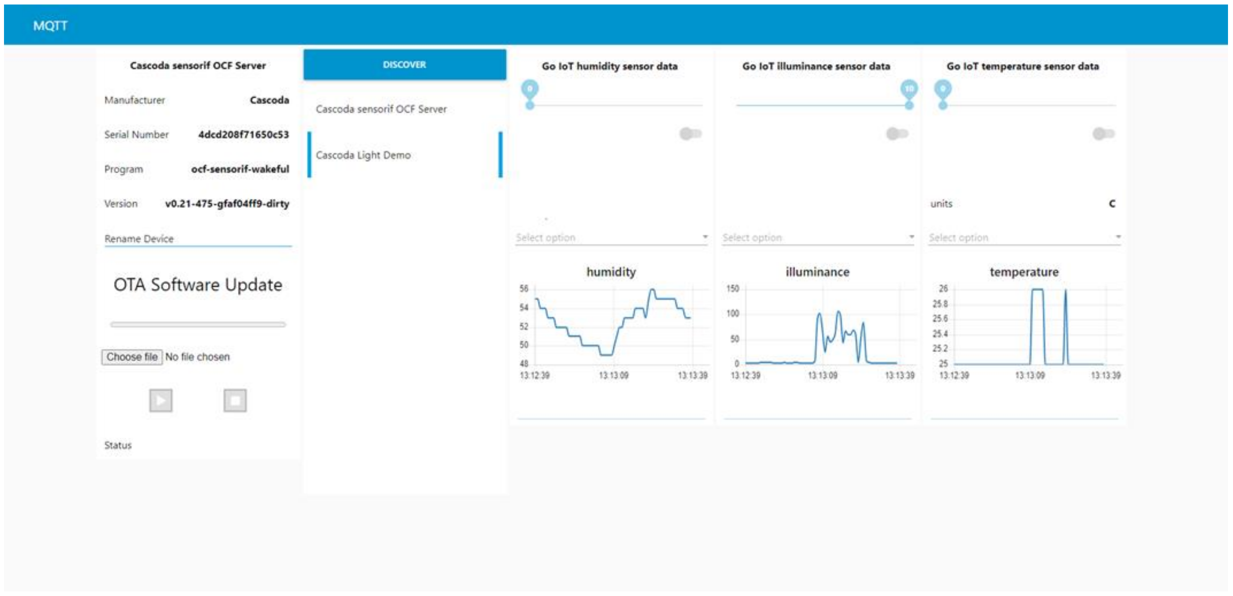


Figure 4: S4creenshot of the partial Node-Red flow to talk to OCF devices through an MQTT proxy

The flow is also used to create a generic dashboard.

The dashboard shows the data per resource. The resources that actuate (can receive data, denoted with interface “oic.if.a”) are interactive with the dashboard. The resources that can sense (only send data, denoted with interface “oic.if.s”) will display graphs.


MQTT

Cascoda sensorif OCF Server

Manufacturer	Cascoda
Serial Number	4dcd208f71650c53
Program	ocf-sensorif-wakeful
Version	v0.21-475-gfaf04ff9-dirty

Rename Device

OTA Software Update

Choose file No file chosen

Status

DISCOVER

Cascoda sensorif OCF Server

Cascoda Light Demo

Figure 5: 5Screenshot of the Node-Red Dashboard showing humidity, illuminance and temperature sensors

Background links:

Why the MQTT Protocol is So Popular | Automation World

Use Cases (mqtt.org)

MQTT Version 5.0 (oasis-open.org)

MQTT to Azure Data Lake Storage in Real-Time (striim.com)

Create a secure data lake by masking, encrypting data, and enabling fine-grained access with AWS Lake Formation | AWS Big Data Blog (amazon.com)

IBM Docs

Streaming real-time sensor data to Grafana using MQTT and Grafana Live | Grafana Labs

How to save MQTT messages to MySQL database | Farnell

Create and Manage an MQTT Connector in Oracle IoT Production Monitoring Cloud Service

OPC UA + MQTT = A Popular Combination for IoT Expansion – OPC Connect (opcfoundation.org)

MQTT Sparkplug Solution (hivemq.com)

Gateway / KNX / MQTT (adfweb.com) (list of system protocols)

Node-RED (nodered.org)

MR MQTT Data Streaming - Cisco Meraki