MQTT

Message Queue Telemetry Transport

Stefan Seifried

Institute of Automation
Automation Systems Group
Vienna University of Technology
Vienna, Austria

https://www.auto.tuwien.ac.at

12.06.2017

Outline

- Excursus: Internet of Things Overview
- MQTT
  - Overview
  - Publish / Subscribe
  - Topics
  - Connection
  - Quality of Service
  - Security

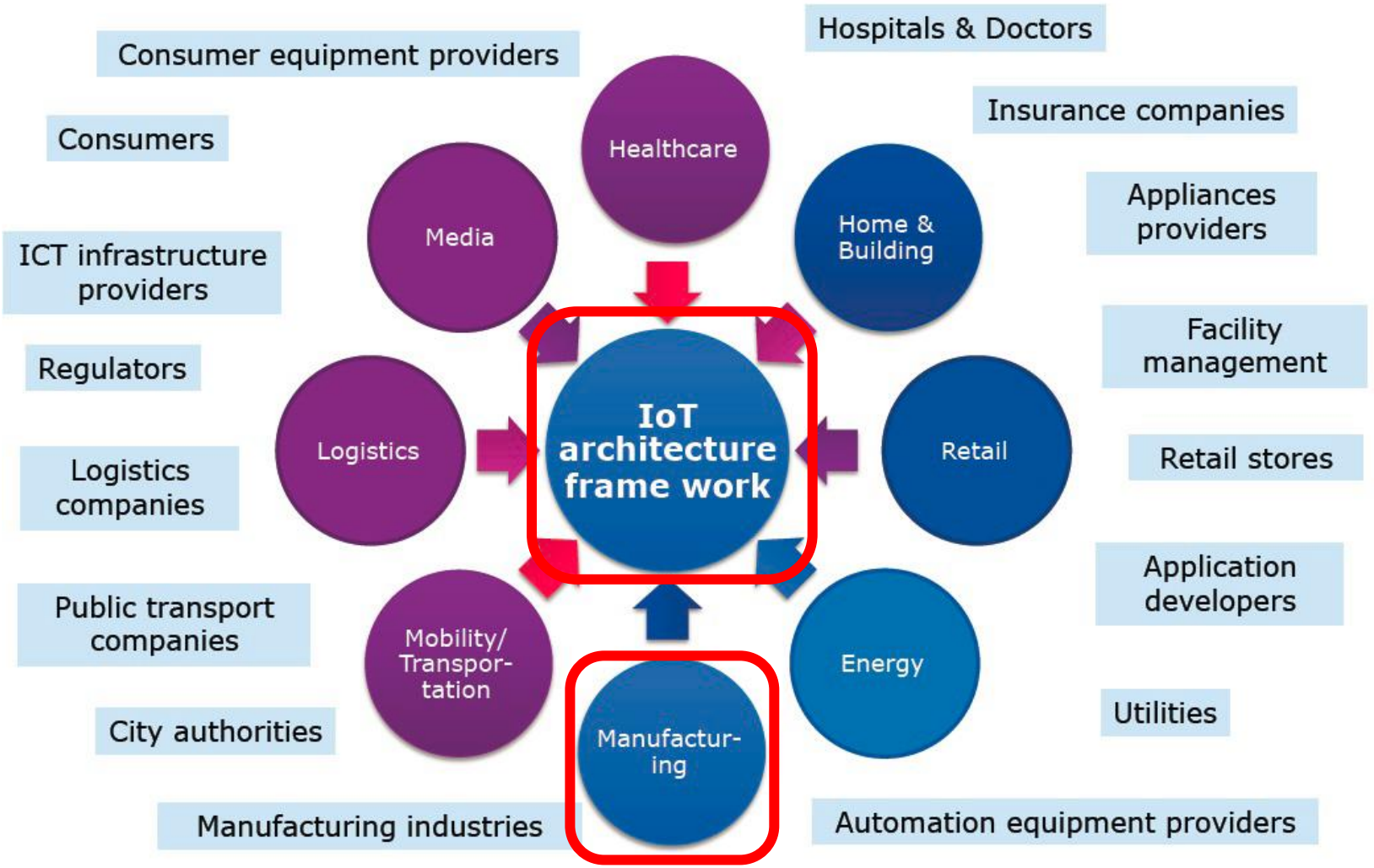
Excursus:
Internet of Things Overview

Internet of Things Overview | Definition

- **Internet of Things (IoT):** A global infrastructure for the information society, enabling advanced services by interconnecting (physical and virtual) things based on existing and evolving interoperable information and communication technologies.
- **thing:** With regard to the Internet of things, this is an object of the physical world (physical things) or the information world (virtual things), which is capable of being identified and integrated into communication networks.

(See reference [3])

Internet of Things Overview | Fields of Applications



IoT markets and stakeholders (See reference [2])

Internet of Things Overview | Challenges of the IoT

- **Identification** and **naming** of objects and services
- **Security** / **Privacy** / **Authority**
- **Presence** (of people and devices)
- **Geographic location**, esp. **mobility support** and **tracking**
- **Discovery** and **search**
- **Data processing**, **computing** and various volumes of data traffic
- **Heterogeneous networking environment** (IP and non-IP)
- **Global connectivity** (IP based) and **Web Services**
- **Autonomics** (Self configuring, intelligence for control)
- **Constraint objects**

(See also Reference [1])

Internet of Things Overview | Protocols (1)

- The **Constrained Application Protocol (CoAP)** is a specialized web transfer protocol for use with constrained nodes and constrained networks in the **IoT.** The protocol is designed for M2M applications such as smart energy and building automation.”
  - Specified in RFC 7252
  - Based on REST model (like HTTP)
  - Various available Data Models like JSON, XML or CBOR
  - Builtin DTLS security
  - Small code size (10 KiB RAM, 100 KiB Program memory)

Internet of Things Overview | Protocols (2)

- **XMPP** - Extensible Messaging and Presence Protocol
  - Originated as a chat protocol (Jabber)
  - Standardized in RFC 6120, 6121, 7622
  - Not widely deployed, but used in Smart Grid communication
- **AMQP** - Advanced Message Queuing Protocol
  - Publish/Subscriber Model (similar to MQTT)
  - OASIS and ISO standard (ISO/IEC 19464:2014)
- **MQTT** – Message Queue Telemetry Transport

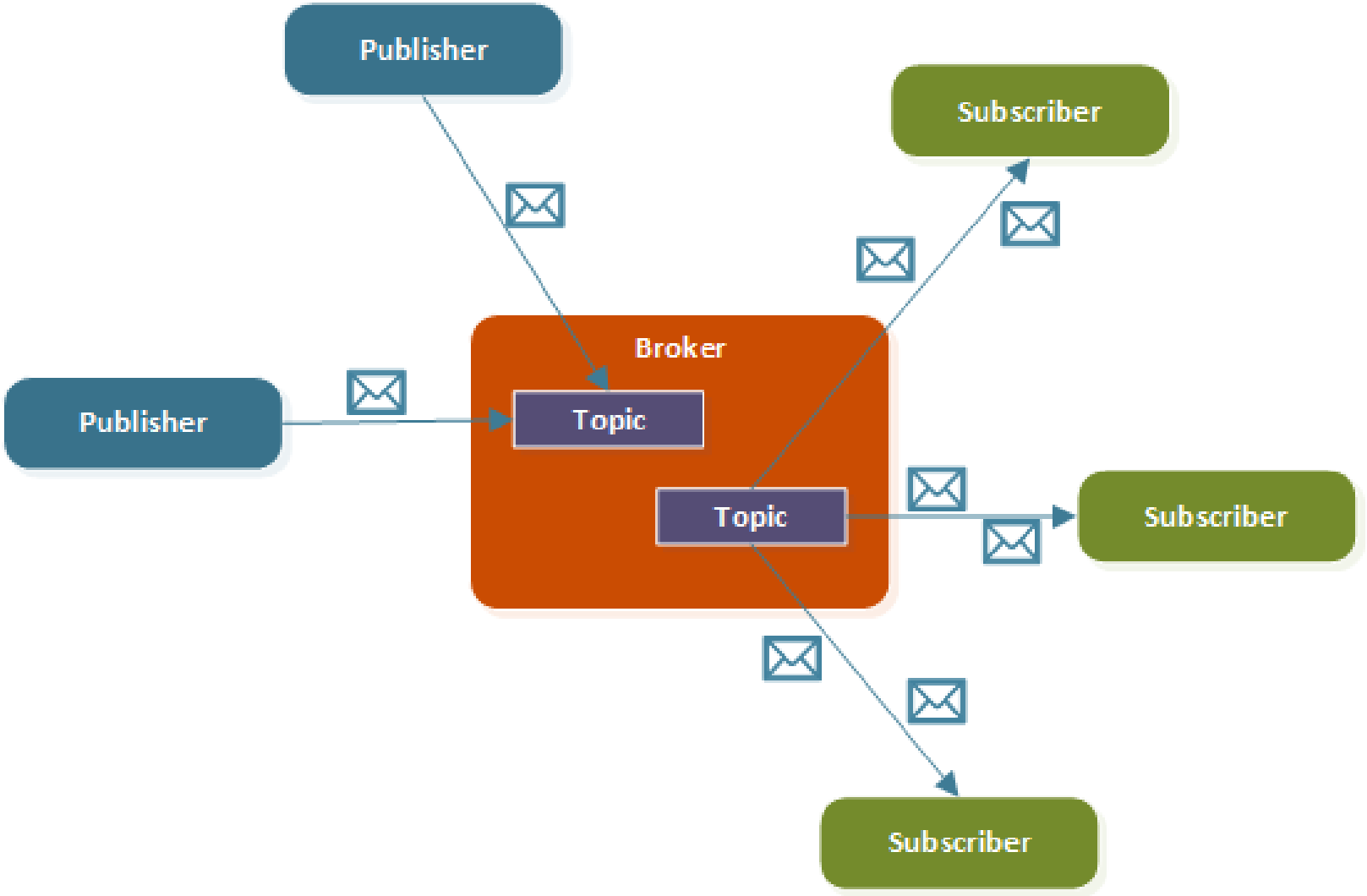
(See reference [4])

MQTT

Overview

- Originally created by IBM in the late 90‘s
- Lightweight message protocol for Machine-to-Machine communication (minimal message size 2 bytes)
- MQTT originally stood for Message Qeue Telemetry Transport, this is not true any more
- MQTT does NOT have a Message Queue (Historic Reasons)!
- Standardized by OASIS and ISO/IEC 20922:2016
- TCP/IP based
- Asynchronous Publish/Subscribe Architecture
- Different Quality of Service Levels

Publish / Subscribe | Architecture



Publish / Subscribe | Notions

- **Topics** are in general strings with an hierarchical structure, that allow filtering based on a limited number of expression
- **Client** is any device from a micro controller up to a full fledged server, that has a MQTT library running and is connecting to an MQTT broker over any kind of network. In general a MQTT client can be both a publisher & subscriber at the same time
- **Broker** is primarily responsible for receiving all messages, filtering them, decide who is interested in it and then sending the message to all subscribed clients.

Publish / Subscribe | Aims

- **Space decoupling:** Publisher and subscriber do not need to know each other (by ip address and port for example)
- **Time decoupling:** Publisher and subscriber do not need to run at the same time.
- **Synchronization decoupling:** Operations on both components are not halted during publish or receiving

Publish / Subscribe | Message Filtering

- **Subject-based filtering:** The filtering is based on a subject or topic. The receiving client subscribes on the topics it is interested in with the broker and from there on it gets all message based on the subscribed topics.
- **Content-based filtering:** Content-based filtering is as the name already implies, when the broker filters the message based on a specific content filter-language. Therefore clients subscribe to filter queries of messages they are interested in.
- **Type-based filtering:** When using object-oriented languages it is a common practice to filter based on the type/class of the message (event). In this case a subscriber could listen to all messages, which are from type Exception or any subtype of it.

Publish / Subscribe | Distinction from Message Queues

- **A message queue stores message until they are consumed**
  - each incoming message will be stored on a queue until it is picked up by any consumer. Otherwise the message will remain in the queue.
  - In MQTT, messages may not be processed by any client.
- **A message will only be consumed by one client**
  - In MQTT, every subscriber to a topic gets the message
- **Queues are named and must be created explicitly**
  - In MQTT, topics are created on the fly

Topics

(See reference [8])

- Lightweight alternative to message queues
  - No special operation for creation needed
  - Special **SYS-topics** that reveal broker information
  - Must have **at least 1-character**
  - Can **contain spaces**
  - Is **case-sensitive**

Topics | Wildcards: Single Level +

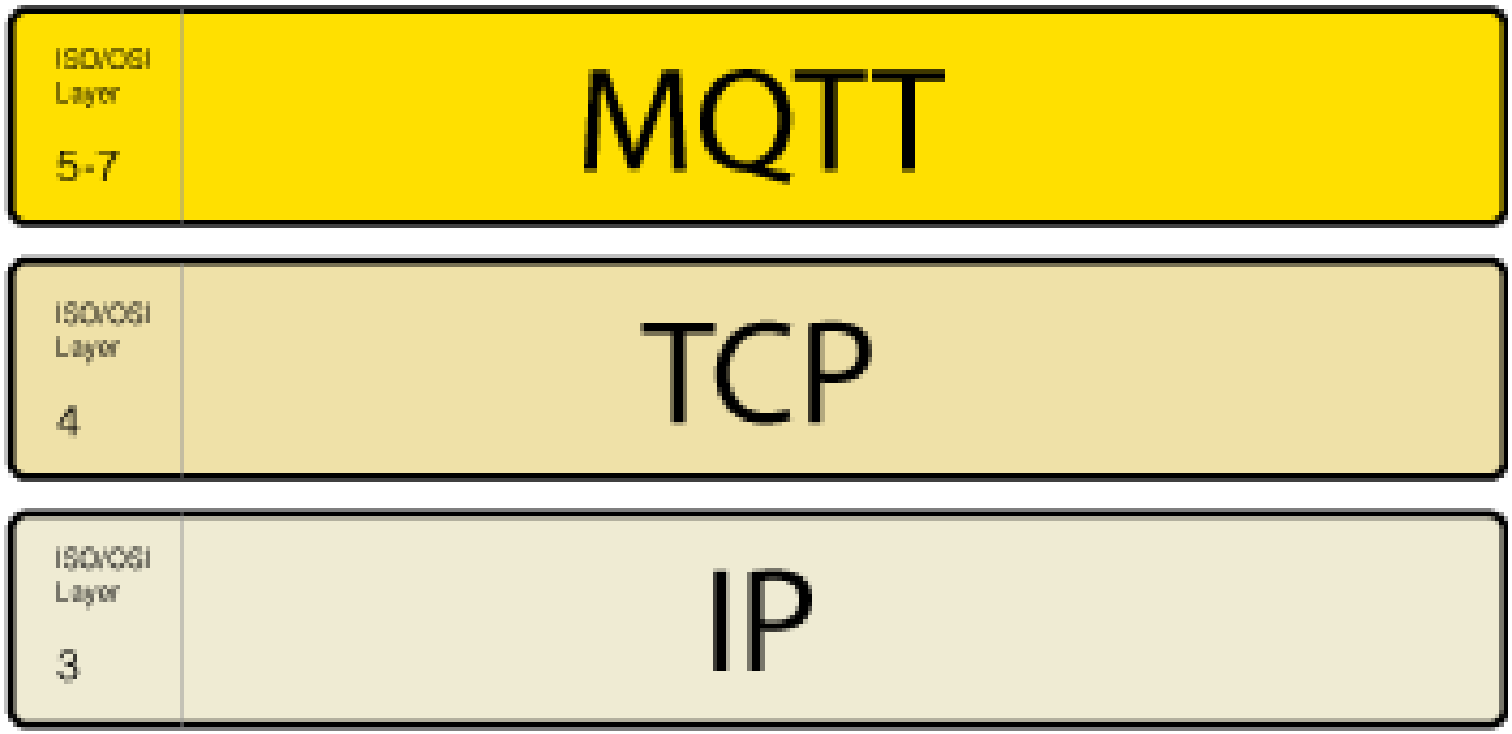
- myhome / groundfloor / livingroom / temperature
- myhome / groundfloor / kitchen / temperature
- myhome / groundfloor / kitchen / brightness
- myhome / firstfloor / kitchen / temperature
- myhome / groundfloor / kitchen / fridge / temperature

(See reference [8])

Topics | Wildcards: Multi Level

(See reference [8])

Connection | Stack



MQTT network stack
(See reference [6])

Connection | CONNECT & CONNACK messages

MQTT-Packet:

CONNECT

contains:	Example
clientId	"client-1"
cleanSession	true
username (optional)	"hans"
password (optional)	"letmein"
lastWillTopic (optional)	"/hans/will"
lastWillQos (optional)	2
lastWillMessage (optional)	"unexpected exit"
lastWillRetain (optional)	false
keepAlive	60

Client initates connection by **CONNECT** message

Broker responds with **CONNACK** message

MQTT-Packet:

CONNACK

contains:	Example
sessionPresent	true
returnCode	0

CONNECT & CONNACK messages (See reference [6])

Connection | PUBLISH message

MQTT-Packet:

PUBLISH

contains:	Example
packetId (always 0 for qos 0)	4314
topicName	"topic/1"
qos	1
retainFlag	false
payload	"temperature:32.5"
dupFlag	false

PUBLISH message (See reference [7])

Connection | SUBSCRIBE & SUBACK message

MQTT-Packet:

SUBSCRIBE

contains:		Example
packetId		4312
qos1	} (list of topic + qos)	1
topic1		"topic/1"
qos2	}	0
topic2		"topic/2"
...		...

MQTT-Packet:

SUBACK

contains:		Example
packetId		4313
returnCode 1	(one returnCode for each	2
returnCode 2	topic from SUBSCRIBE,	0
...	in the same order)	...

SUBSCRIBE & SUBACK message (See reference [7])

Connection | UNSUBSCRIBE & UNSUBACK message

MQTT-Packet:

UNSUBSCRIBE

contains:	Example
packetId	4315
topic1 } (list of topics)	"topic/1"
topic2	"topic/2"
...	...

MQTT-Packet:

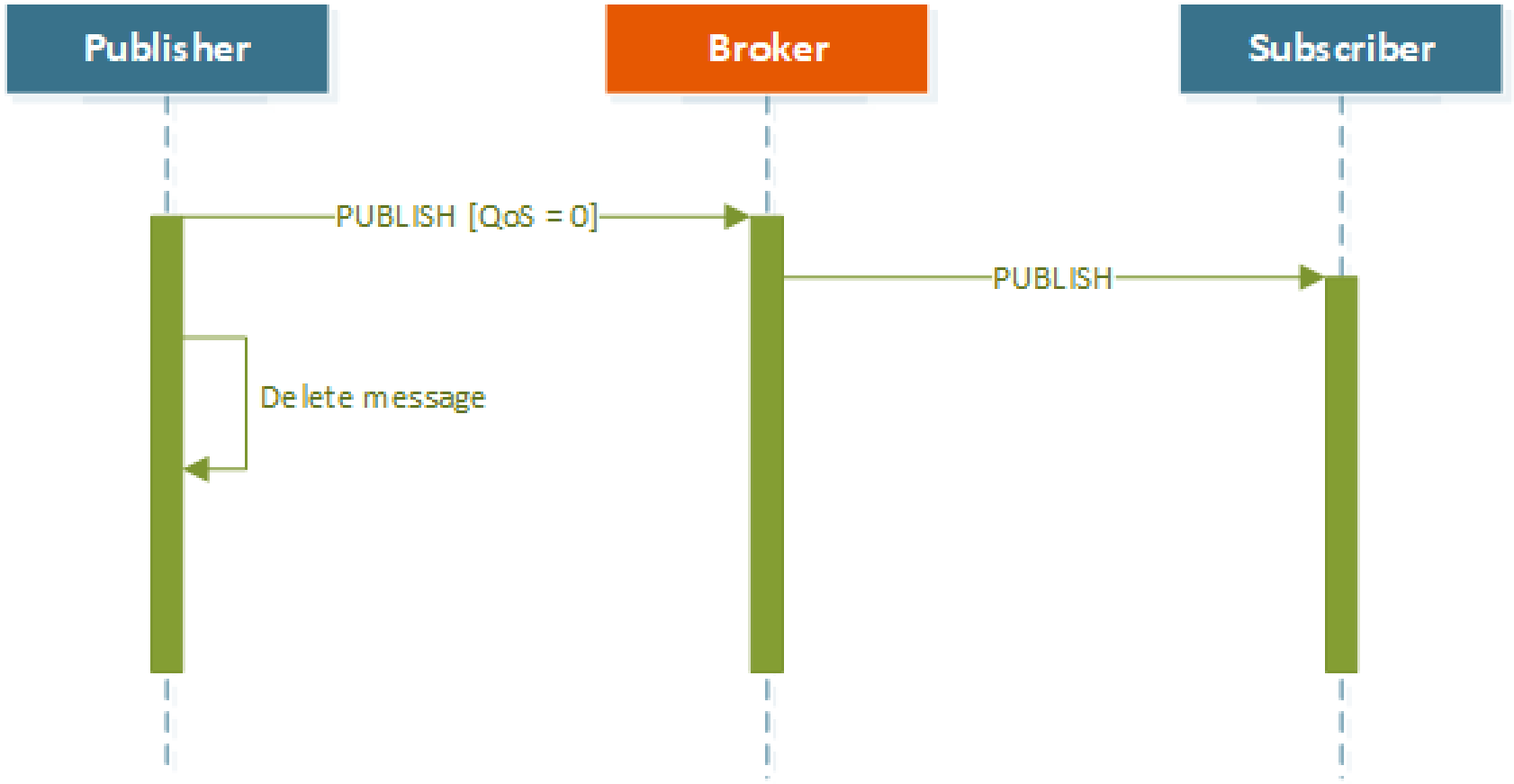
UNSUBACK

contains:	Example
packetId	4316

SUBSCRIBE & SUBACK message (See reference [7])

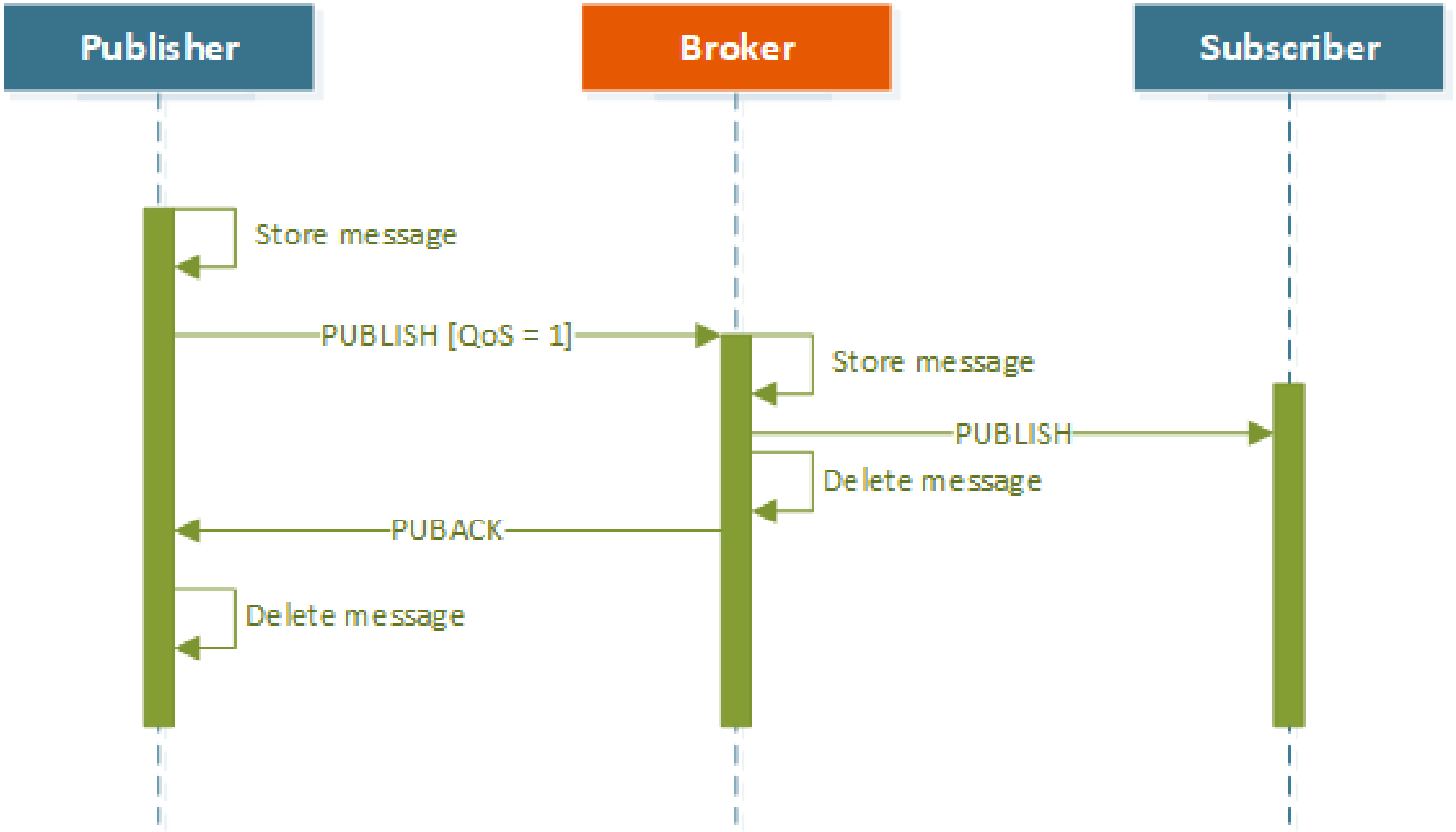
Quality of Service | QoS Level 0

- At most once (fire and forget)



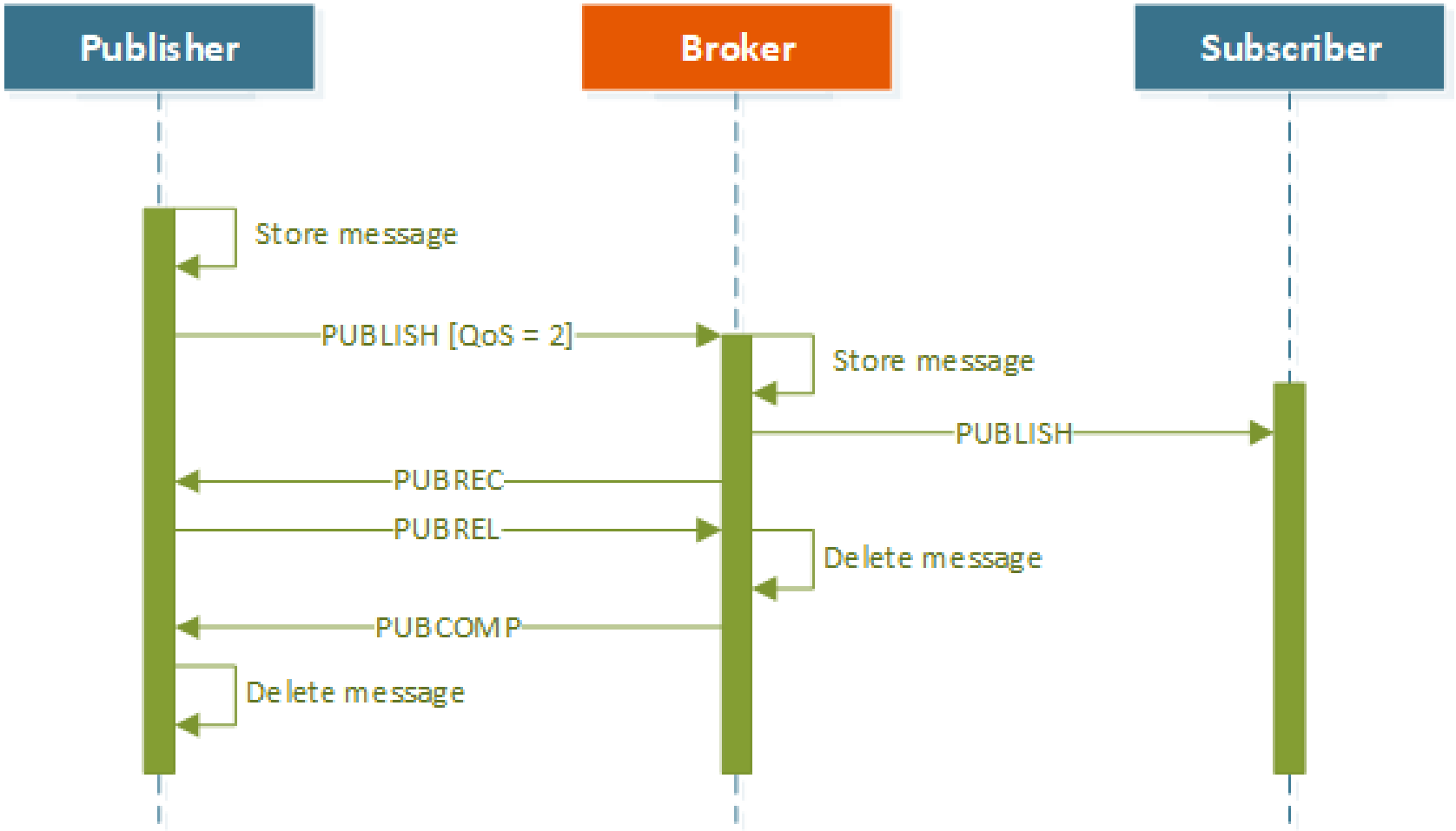
Quality of Service | QoS Level 1

- At least once



Quality of Service | QoS Level 1

- Exactly once



References

[1] The Internet of Things - Concept and Problem Statement, https://tools.ietf.org/html/draft-lee-iot-problem-statement-05, (Expired)

[2] Towards a definition of the Internet of Things (IoT), http://iot.ieee.org/images/files/pdf/IEEE_IoT_Towards_Definition_Internet_of_Things_Revision1_27MAY15.pdf

[3] ITU-T Y.4000/Y.2060, http://www.itu.int/ITU-T/recommendations/rec.aspx?rec=y.4000, 06.2012

[4] Application layer protocols for the Internet of Things: A survey, http://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=7745303

References

[5] MQTT Version 3.1.1, http://docs.oasis-open.org/mqtt/mqtt/v3.1.1/os/mqtt-v3.1.1-os.pdf

[6] MQTT Essentials Part 3, http://www.hivemq.com/blog/mqtt-essentials-part-3-client-broker-connection-establishment

[7] MQTT Essentials Part 4, http://www.hivemq.com/blog/mqtt-essentials-part-4-mqtt-publish-subscribe-unsubscribe

[8] MQTT Essentials Part 5, http://www.hivemq.com/blog/mqtt-essentials-part-5-mqtt-topics-best-practices