TinyCLR.it

La Community italiana su NET Micro framework

e101
embedded101

Microsoft
EMBEDDED
CONFERENCE

Napoli 15 febbraio 2014

Smart Home & Smart Factory systems

MQTT & IoT protocols comparison

Paolo Patierno

Software Embedded Engineer

Microsoft EMBEDDED CONFERENCE

Sponsor

Who am I ? Contacts

- «DotNetCampania» member
  - http://dotnetcampania.org/blogs/paolopat/default.aspx
- «TinyCLR.it» member
  - http://www.tinyclr.it
- «Embedded101» board of director member
  - http://www.embedded101.com/Blogs/PaoloPatierno.aspx
- Twitter
  - @ppatierno
- Linkedin
  - http://it.linkedin.com/in/paolopatierno
- Skype
  - paolopat80
- Email
  - ppatierno@live.com

Microsoft EMBEDDED CONFERENCE

Agenda

- IoT protocols war !
- MQTT
  - Introduction
  - Architecture
  - Features
- Comparison with …
  - HTTP
  - CoAP
  - AMQP

Microsoft EMBEDDED CONFERENCE

IoT protocols war !

AMQP

HTTP

CoAP

MQTT

XMPP

DDS

STOMP

Microsoft EMBEDDED CONFERENCE

MQTT : introduction

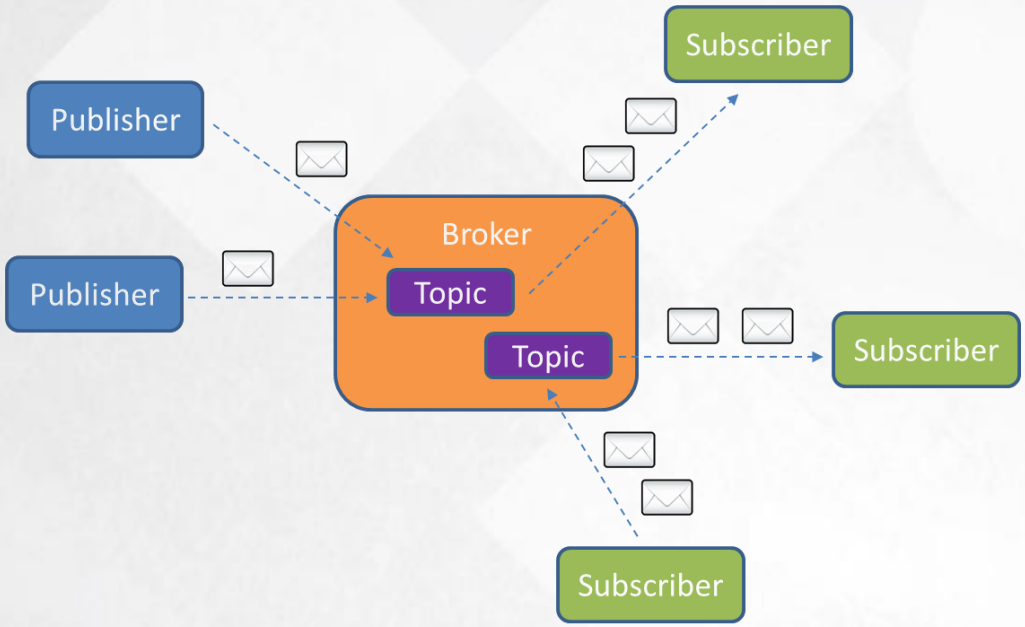
- MQTT (Message Queue Telemetry Transport)
  - Open : created by IBM & Eurotech and donated to Eclipse “Paho” M2M project (OASIS standard in 2014)
  - Lightweight : smallest packet size 2 bytes (header), reduced clients footprint (C# M2Mqtt library 30 KB)
  - Reliable : three QoS and patterns to avoid packet loss on client disconnection
  - Simple :
    - TCP based
    - Asynchronous
    - Publish/Subscribe
    - Few verbs
    - Payload agnostic

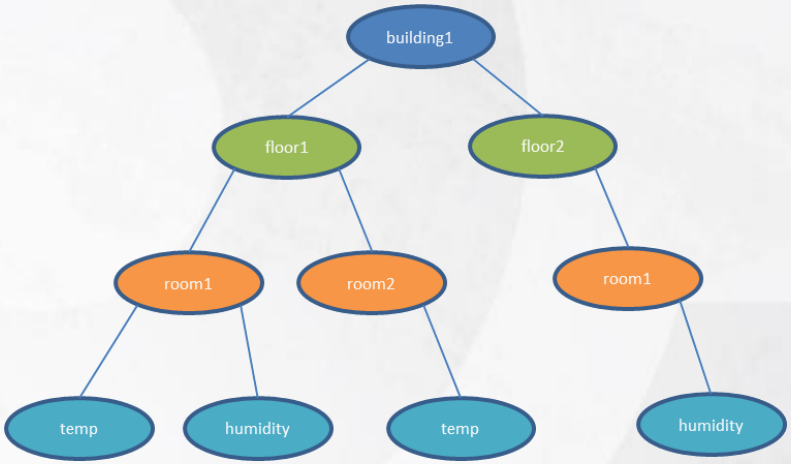
MQTT.ORG

Microsoft EMBEDDED CONFERENCE

MQTT : publish/subscribe

- Broker and connected Clients
  - Broker receives subscription from clients on topics
  - Broker receives messages and forward them
  - Clients subscribe/publishes on topics
- Topics for publish and subscribe (like queue)
- Brokers bridge configuration

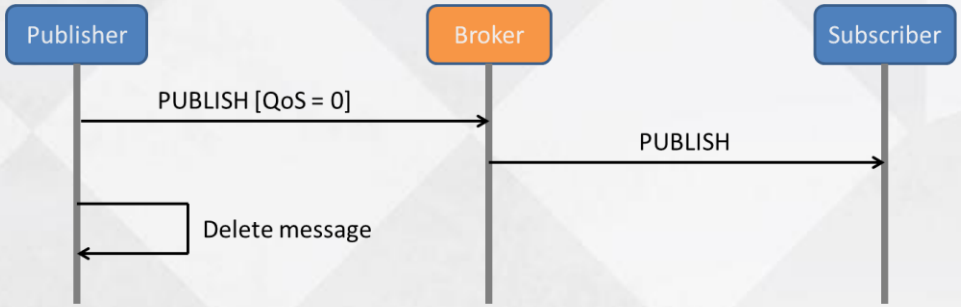




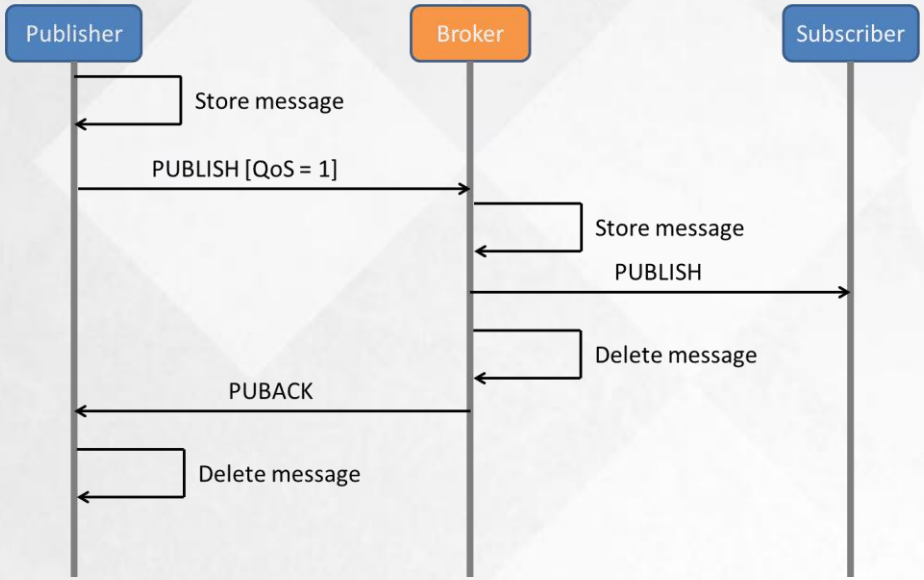
Microsoft EMBEDDED CONFERENCE

MQTT : Quality of Service

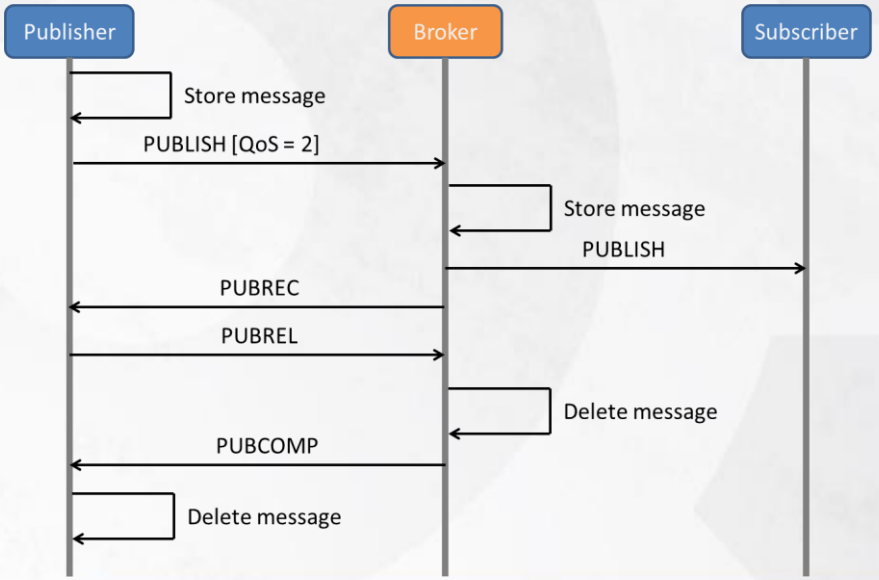
QoS 0 : At most once (fire and forget)



QoS 1 : At least once



QoS 2 : Exactly once



Microsoft EMBEDDED CONFERENCE

MQTT : security ?

- Common big problem for all IoT/M2M protocols
- MQTT is over TCP … use SSL/TLS for secutiry
- Username/Password in the CONNECT message
- Encrypt payload (MQTT is payload agnostic)

Microsoft EMBEDDED CONFERENCE

MQTT : main features

- Keep-Alive message (PINGREQ, PINGRESP)
  - Broker can detect client disconnection (if it doesn't send explicit DISCONNECT)
- Will message : specified in CONNECT message with topic, QoS and retain. On unexpected client disconnection, it is sent to subscribed clients
- Retain message : a PUBLISH message on a topic is kept on the broker. A new connected subscriber on the same topic receives this message (last known good message)
- Durable subscription : on client disconnection, all subscriptions are kept on the broker and recovered on client reconnection

Microsoft EMBEDDED CONFERENCE

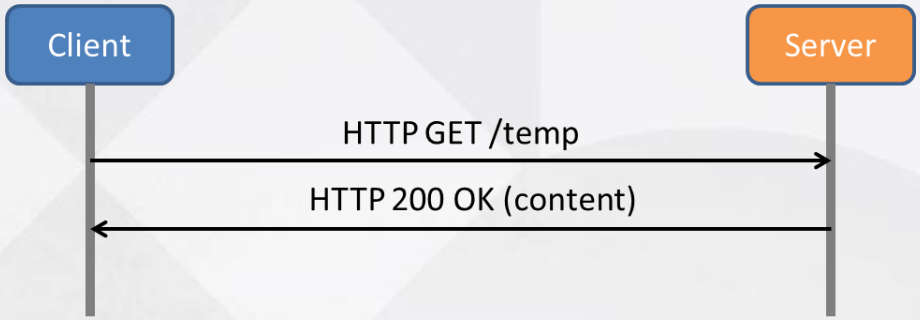
HTTP vs MQTT

- Request/Response (1-1, 1-n more POST)
- Push on client with (long) polling (or WebSocket)
- More bandwidth (ASCII, headers, …)
  - More battery consumption
- No “messaging middleware” integration
- Client more complex (ASCII parser)
- No Quality of Service
- Security based on SSL/TLS
- RESTful

Microsoft EMBEDDED CONFERENCE

HTTP vs MQTT

Request/Response



Sending (1024 msg – 1 byte)

	3G		Wifi	
	HTTPS	MQTT	HTTPS	MQTT
% Battery / Hour	18.43%	**16.13%**	**3.45%**	4.23%
Messages / Hour	1708	**160278**	3628	**263314**
% Battery / Message *	0.01709	**0.00010**	0.00095	**0.00002**
Messages Received	240 / 1024	**1024 / 1024**	524 / 1024	**1024 / 1024**

Receiving (1024 msg – 1 byte)

	3G		Wifi	
	HTTPS	MQTT	HTTPS	MQTT
% Battery / Hour	18.79%	**17.80%**	5.44%	**3.66%**
Messages / Hour	1926	**21685**	5229	**23184**
% Battery / Message *	0.00975	**0.00082**	0.00104	**0.00016**

http://stephendnicholas.com/archives/1217

Microsoft EMBEDDED CONFERENCE

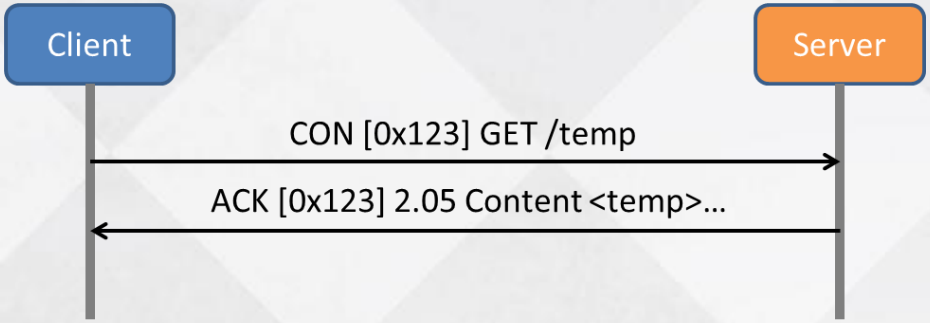
CoAP vs MQTT

- HTTP-like but based on UDP (no TCP)
  - Packets order and retransmission into the sw stack
- Client/Server (HTTP verbs, status codes HTTP-like)
- "Options" (like HTTP headers) are binary
  - Client more simple (than HTTP)
- "Observer" pattern available and "Response back after a while", avoid HTTP (long) polling
- Quality of Service with "confirmable" messages
- Security with DTLS (Datagram TLS)
- Resource discovery
- CoAP node has also server role → NAT problem

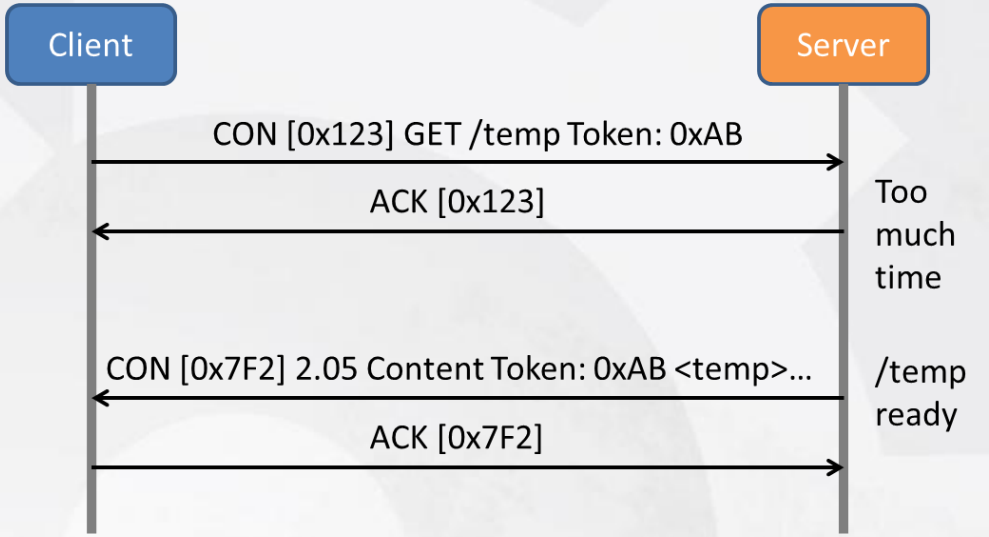
Microsoft EMBEDDED CONFERENCE

CoAP vs MQTT

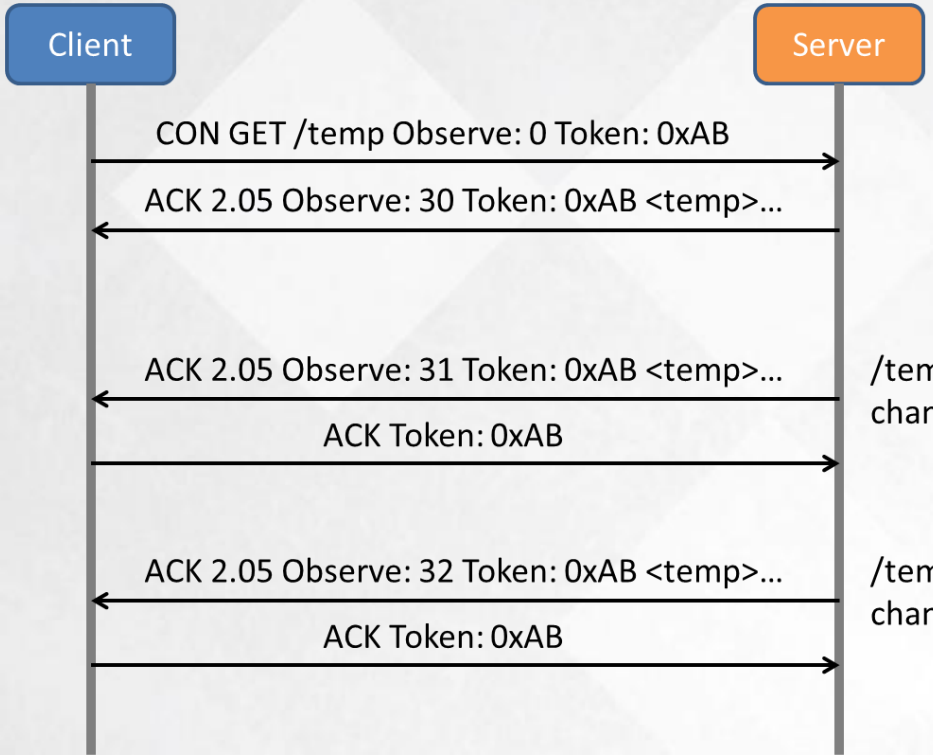
Confirmable request



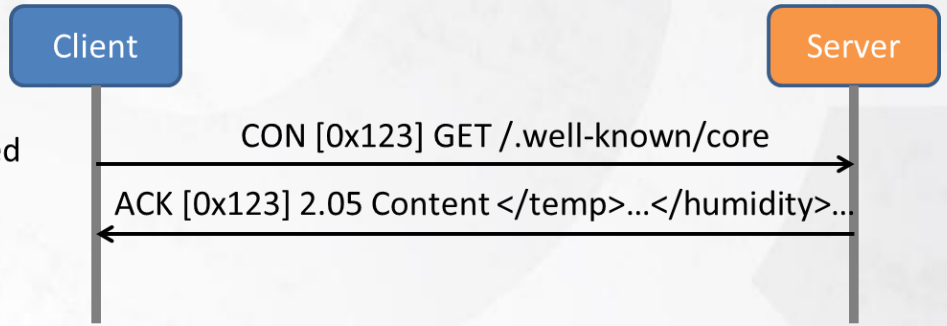
Response back after a while



Observer



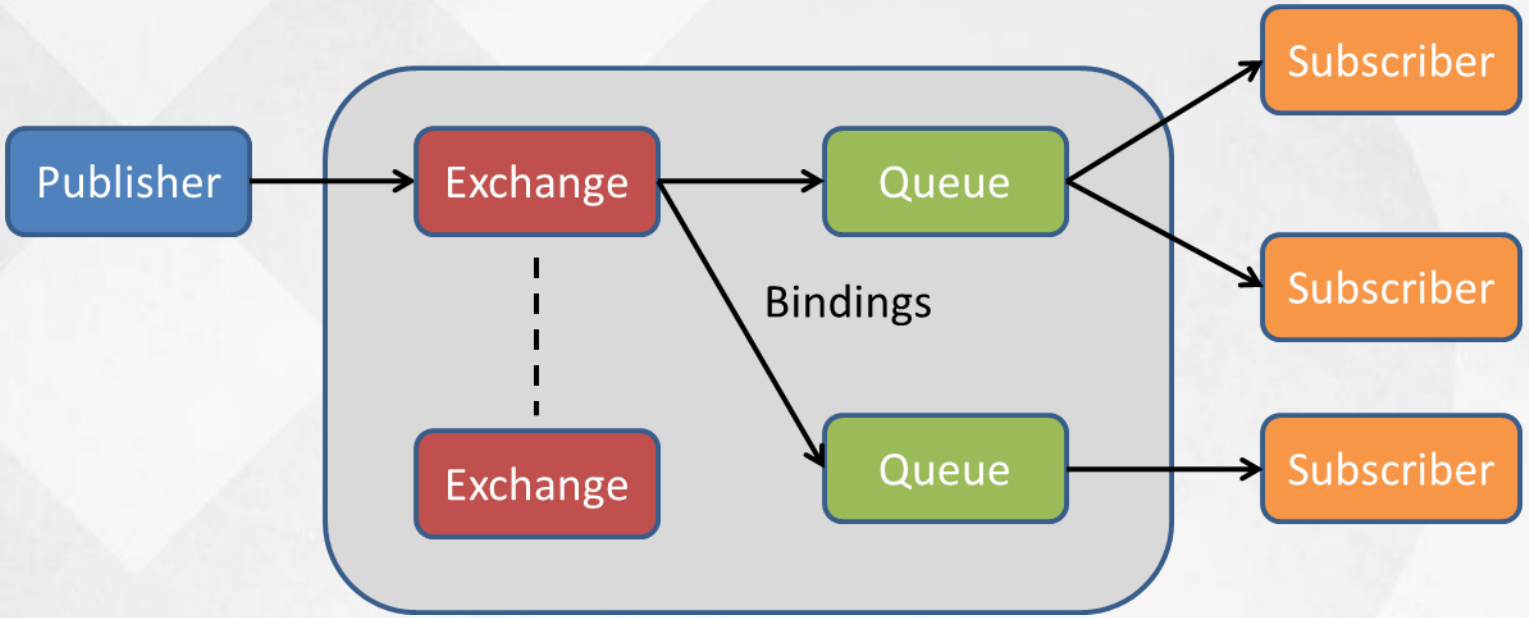
Resource discovery



Microsoft EMBEDDED CONFERENCE

AMQP vs MQTT

- Exchange : receive messages and apply routing
- Binding : define rules to bind exchange to queue
- Queue : simple … it is a queue !



AMQP
Advanced Message Queuing Protocol

Microsoft EMBEDDED CONFERENCE

AMQP vs MQTT

- Default exchange (without a name)
  - Routing message to a queue (routing key = name queue)
- Direct exchange
  - Routing message to a queue based on routing key (not necessary queue name, routing key = bind key)
- Fanout exchange
  - Routing message to more queue (publish/subscribe) and not use a routing key
- Topic exchange
  - Routing message to a queue based on routing key like a topic (routing key match a pattern)
- Header exchange
  - Routing message to queue based on header filters

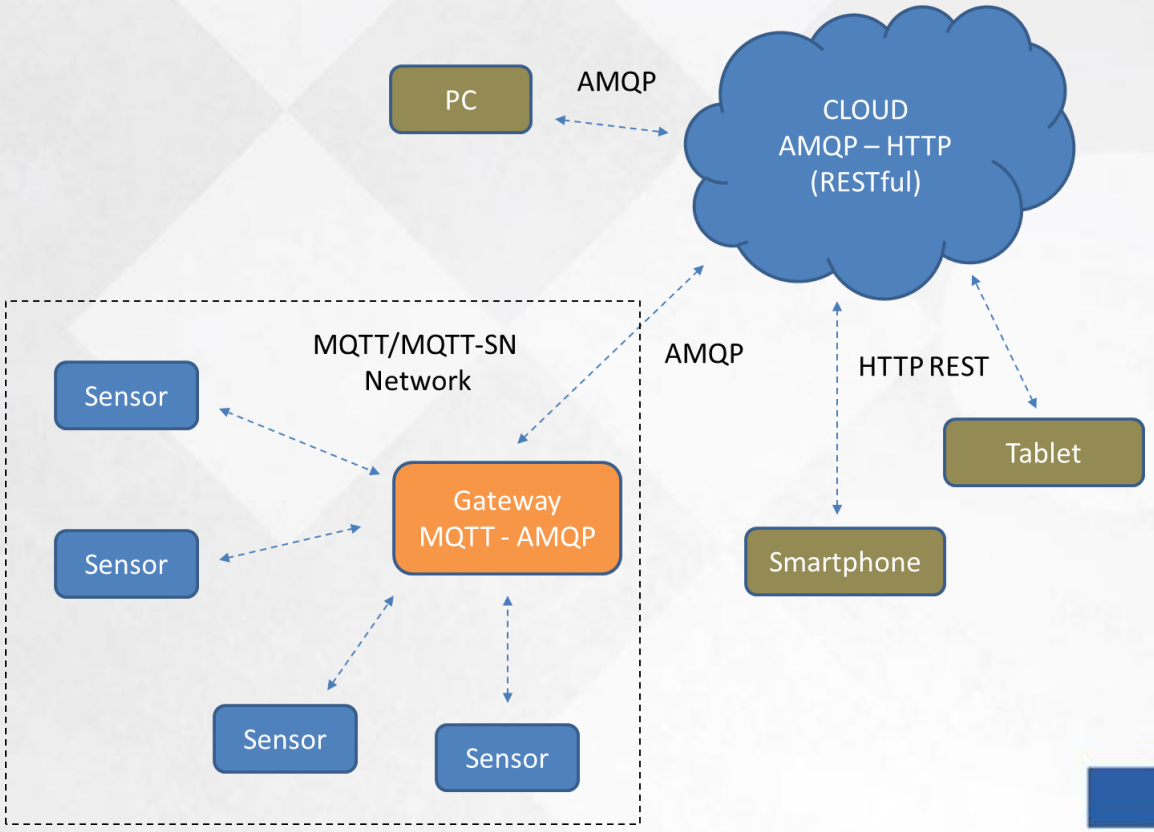
Microsoft EMBEDDED CONFERENCE

AMQP vs MQTT

- Enterprise application oriented
  - Messaging server to server
- More messaging patterns :
  - Load balancing on a queue (more consumers)
  - Publish/Subscribe queue (MQTT topic)
  - Routing with messages redirection to queues based on filters (on header fields or body message)
  - RPC with “correlation id”
- Smallest packet size 60 bytes
- Quality of Service (MQTT-like)
  - Also use “dead letter queue”
- Security based on SSL/TLS

Conclusions

- Protocol choice depends on scenario
- Some protocols have more features than other
- A complex system can use more protocols :



Microsoft EMBEDDED CONFERENCE

References

- IoT/M2M
  - Embedded101 free ebook : http://bit.ly/m2miotbook
  - IBM redbook : http://www.redbooks.ibm.com/abstracts/sg248054.html
- MQTT web site
  - Official web site : http://mqtt.org
  - Mosquitto : http://mosquitto.org
  - HiveMQ : http://www.hivemq.com
  - M2Mqtt C# Client : http://m2mqtt.codeplex.com
  - Eclipse Paho project : http://www.eclipse.org/paho/
  - WebSphere MQ : http://www-03.ibm.com/software/products/en/wmq/
- CoAP
  - IETF : https://datatracker.ietf.org/doc/draft-ietf-core-coap/
  - CoAP for .Net : http://www.coapsharp.com
- AMQP
  - Official web site : http://www.amqp.org
  - RabbitMQ : http://www.rabbitmq.com
  - Windows Azure Service Bus support : http://www.windowsazure.com/en-us/documentation/articles/service-bus-amqp-overview/