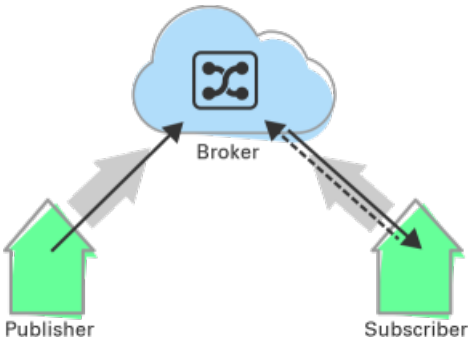
Background knowledge/motivation for MQTT:

MQTT: Communication in the Internet of Things



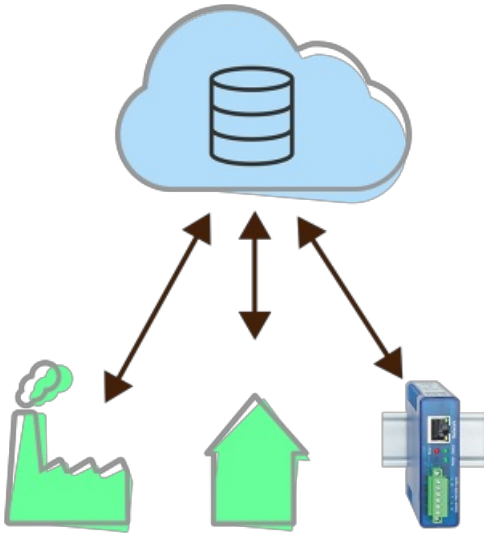
The idea of an Internet of Things carries with it a central problem: How do untold number of devices - with different performance capability, some mobile and some stationary, often connected over unreliable lines and with high latency times - communicate with each other reliably and efficiently?

Direct connections via TCP/IP

Most terminal devices in the Internet of Things have no public IP address. Like a telephone without its own call number they can call other devices, but are themselves incapable of being called. And even when they are, the incoming data traffic is often filtered through a firewall. A direct, network-wide connection between two terminal devices is in most cases not possible.

If an IoT device can be accessed directly, there are still problems with respect to data security. It's not without reason that we are hearing more and more about botnets, which spread out through low-cost routers or IP camera using out-of-date firmware. Not every developer of a new, exciting gadget is working with due care - and not every owner is aware of his responsibility as an administrator.

IoT clouds: A data-based approach

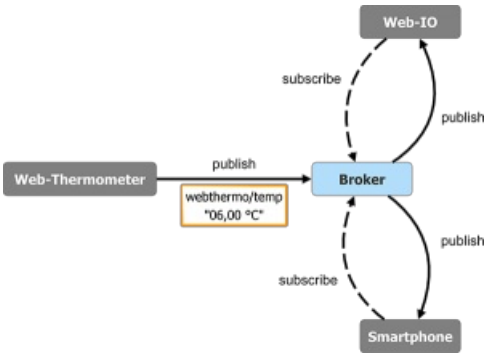


One possible solution is offered by IoT clouds. These are central internet data storage locations provided by entities such as Google, SalesForce or IBM. Customers of Wiesemann & Theis can also make use of the free W&T cloud.

Terminal devices are connected to the servers of the cloud providers to store or retrieve their data from there. Free software solutions such as ThinkSpeak make it possible to set up such a server (or server cluster) in one's own network and to store the data there locally.

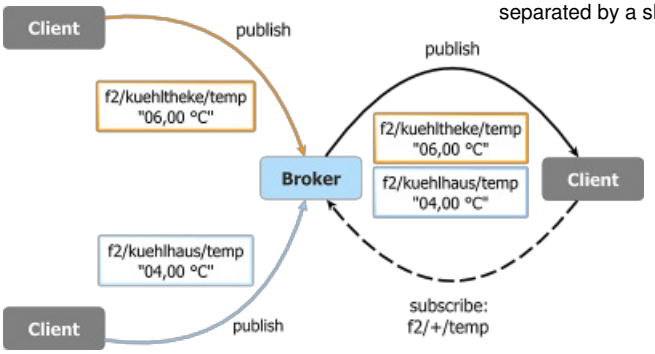
Clouds are data-based. They are often equipped with web front-ends for visualizing measurement values and counter states. M2M access to individual values and time series is done generally through platform-specific REST interfaces. Cloud providers are increasingly offering communication-based access procedures as well. For example IBM BlueMix, Microsoft Azure and Xively offer the MQTT IoT protocol.

MQTT: The communication-based approach



As with the cloud approach, the communication participants (clients) in MQTT active open a connection to a central service, the broker.

But the broker is itself not a data storage entity, but rather acts as a mediator by which connected clients can publish or subscribe to messages. Based on the subjects of the message - the topics - the broker decides which client to publish received messages to.



A topic is a hierarchically constructed string whose members are separated by a slash. Similar to a file system path, topics can be structured in tree form and subscribed as bundles using wildcards.

Various protocol properties enable stable communication even under unfavorable conditions. Slow data transfer rates, high latency times or occasional connection breaks are not a problem. This makes MQTT also well suited for mobile data communication.

Subscriber connection breaks:

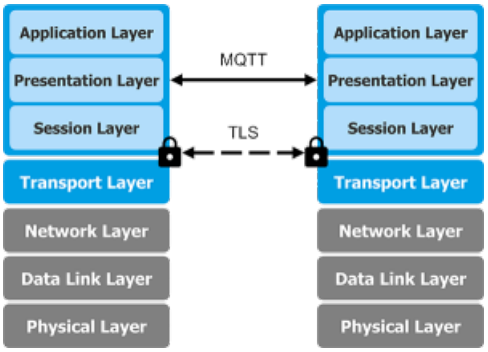
Persistent Session and Quality of Service
When a connection is opened a subscriber can specify that he wants to open a continuous (persistent) session. In this case the broker buffer stores the topics subscribed by the subscriber. When a connection is interrupted, it buffer stores the messages whose receipt needs to be confirmed by the subscriber. Whether a confirmation is required is determined by the Quality of Service (QoS) of a message. A message with a Quality of Service of QoS0 is delivered "no more than once" and accordingly rejected if the subscriber cannot be reached. A message with QoS1 is delivered as often as necessary until the subscriber has confirmed receipt. QoS2 ensures that the subscriber receives the message exactly once.

Publisher connection breaks:

Last Will and Retained Messages
In its "last will" a publisher tells the broker when a connection is opened which message should be sent to the subscribers in case of a sudden connection break. Exactly one last will can be created for each publisher.

When a publisher sends a message in which the retained flag is set, the broker buffer stores this message for a topic. This message is sent to subscribers which are newly subscribing to the topic, and also to subscribers with Quality of Service 1 which have not yet send a confirmation of receipt. An application example would be sending the last known measurement value of a temperature sensor. Exactly one message can be made available for each topic.

Assured communication:



Authentication and Encryption
MQTT supports authentication of the clients through user names and a password. This allows permissions to be assigned on the broker side, to ensure for example that only the temperature sensor on site is permitted to publish messages with the topic f2/kuehlthekte/temp. The user name and password are conveyed when the connection is opened and sent unencrypted. This means it is a good idea to encrypt any MQTT communication from the very beginning. Since the protocol is found on the upper layers of the OSI model, TLS is easy to use for the encryption - as long as the corresponding terminal device has the necessary resources.

Universal and portable

MQTT not only has a very simple basic structure, it also offers great freedom in organizing the topics and has no requirements for the content of the payload. This makes MQTT a very generic and versatile protocol. Nor are there hardly any limitations when selecting a broker: These are available as free software and from established commercial vendors, as a service on the Web or as a hardware appliance. An application which uses MQTT for communication works with any MQTT broker. This ensures that the platform can be changed at any time and without great effort.

Try it yourself

Would you like to try out MQTT? Our Web-IO Digital is the ideal entry into the IoT universe. Simply request the Web-IO Digital as a sample.
If the sample is returned within 30 days, you pay only the return shipping costs. To keep the test unit, simply pay the accompanying invoice.

Questions about the Web-IO Digital with MQTT?

Mr. Thiel will be glad to help.

Phone: +49 202/2680-110 (Mon.-Fri. 8 a.m. to 5 p.m.)
E-mail: f.thiel@wut.de

W&T
www.WuT.de

We are available to you in person:

Wiesemann & Theis GmbH
Porschestr. 12
42279 Wuppertal
Phone: +49 202/2680-110 (Mon.-Fri. 8 a.m. to 5 p.m.)
Fax: +49 202/2680-265
info@wut.de

Data Privacy