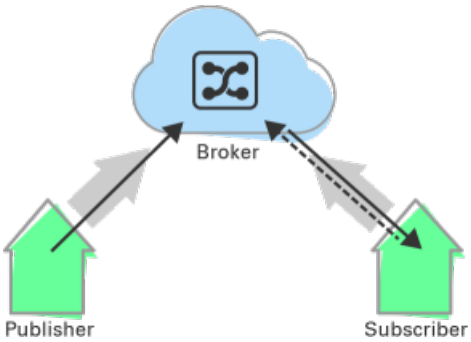
Hintergrundwissen/Motivation zu MQTT:

# MQTT: Kommunikation im Internet der Dinge



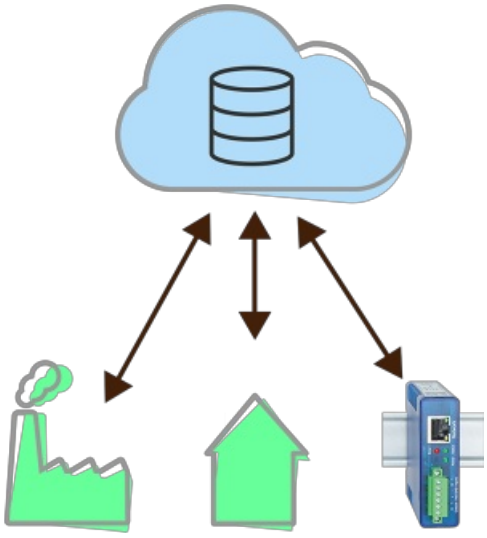

Die Idee eines Internets der Dinge bringt ein zentrales Problem mit sich: Wie sollen unzählige Geräte - mit unterschiedlicher Leistungsfähigkeit, teilweise mobil, teilweise stationär, oftmals über unzuverlässige Leitungen und mit hohen Latenzzeiten angebunden - verlässlich und effizient miteinander kommunizieren?

## Direktverbindungen über TCP/IP

Die meisten Endgeräte im Internet der Dinge haben keine öffentliche IP-Adresse. Wie ein Telefon ohne eigene Rufnummer können Sie zwar andere Geräte anrufen, sind aber selbst nicht erreichbar. Falls doch, ist der eingehende Datenverkehr oftmals über eine Firewall gefiltert. Ein direkter, netzübergreifender Verbindungsaufbau zwischen zwei Endgeräten ist in den meisten Fällen nicht möglich.

Lässt sich ein IoT-Gerät doch direkt ansprechen, ergeben sich Probleme in Hinblick auf die Datensicherheit. Nicht grundlos hört man zunehmend von Botnetzen, die sich etwa über günstige Router oder IP-Kameras mit veralteter Firmware ausbreiten. Nicht jeder Entwickler eines neuen, spannenden Gadgets arbeitet mit der gebotenen Sorgfalt - und nicht jeder Besitzer ist sich seiner Verantwortung als Administrator bewusst.

## IoT-Clouds: Ein datenbasierter Ansatz

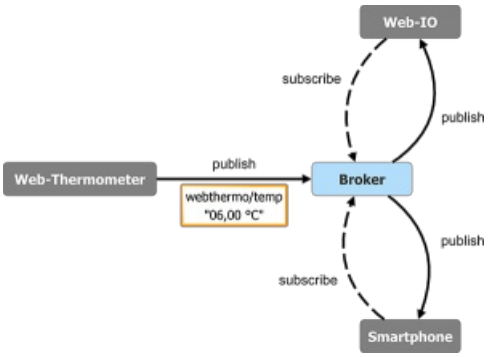


Eine mögliche Lösung bieten IoT-Clouds. Diese sind zentrale Internet-Datenspeicher, die von Anbietern wie Google, SalesForce oder IBM bereitgestellt werden. Kunden von Wiesemann & Theis steht auch die kostenlose W&T-Cloud zur Verfügung.

Endgeräte verbinden sich mit den Servern der Cloudprovider, um ihre Daten dort zu hinterlegen oder abzurufen. Freie Softwarelösungen wie ThinkSpeak bieten die Möglichkeit, einen solchen Server(-verbund) auch im eigenen Netzwerk einzurichten und die Daten vor Ort zu speichern.

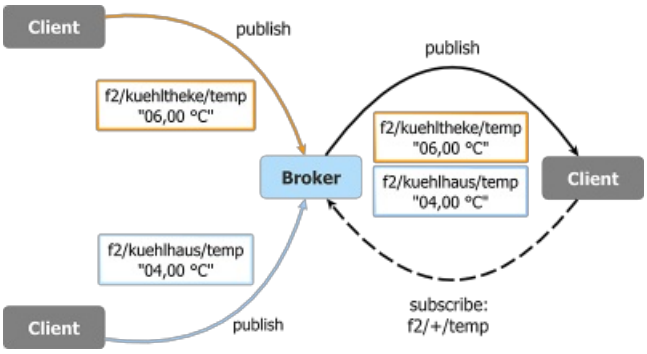
Clouds arbeiten datenbasiert. Vielfach sind sie mit Web-Frontends ausgestattet, um Messwerte und Zählerstände zu visualisieren. Der M2M-Zugriff auf Einzelwerte und Zeitreihen erfolgt üblicherweise über plattformspezifische REST-Schnittstellen. Zunehmend bieten die Cloudprovider auch kommunikationsbasierte Zugriffsverfahren an. So unterstützen etwa IBM BlueMix, Microsoft Azure und Xively das IoT-Protokoll MQTT.

## MQTT: Der kommunikationsbasierte Ansatz




Wie auch beim Cloud-Ansatz, bauen die Kommunikationsteilnehmer (Clients) bei MQTT aktiv eine Verbindung zu einem zentralen Dienst, dem Broker, auf.

Der Broker ist aber kein Datenspeicher, sondern agiert als Vermittlungsstelle, bei dem verbundene Clients Nachrichten hinterlegen (publish) oder abonnieren (subscribe) können. Anhand des Themas der Nachricht, des Topics, entscheidet der Broker, an welchen Client er empfangene Nachrichten zustellt (publish).




Ein Topic ist ein hierarchisch aufgebauter String, dessen Glieder mit einem Schrägstrich voneinander getrennt werden. Ähnlich wie bei einem Dateisystempfad lassen sich Topics so baumartig strukturieren und unter der Verwendung von Wildcards gebündelt abonnieren.

Verschiedene Eigenschaften des Protokolls ermöglichen eine stabile Kommunikation auch unter ungünstigen Voraussetzungen. Geringe Datenübertragungsraten, hohe Latenzzeiten oder gelegentliche Verbindungsabbrüche stellen kein Problem dar. Daher ist MQTT auch gut für die mobile Datenkommunikation geeignet.

## Verbindungsabbrüche beim Subscriber:

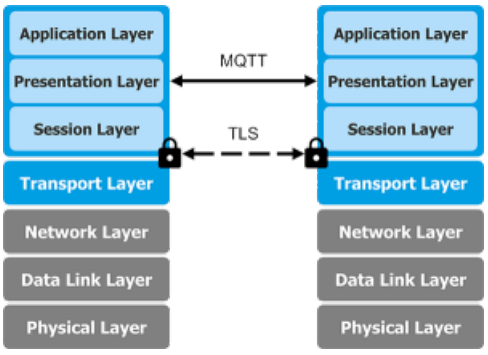
**Persistent Session und Quality of Service**
Ein Subscriber kann beim Verbindungsaufbau angeben, dass er eine dauerhafte Sitzung (Persistent Session) aufbauen möchte. In diesem Fall speichert der Broker die vom Subscriber abonnierten Topics zwischen. Im Fall eines Verbindungsabbruchs speichert er alle Nachrichten, deren Empfang vom Subscriber bestätigt werden müssen. Ob eine Bestätigung erforderlich ist, wird über die Servicequalität (Quality of Service, kurz: QoS) einer Nachricht bestimmt. Eine Nachricht mit Servicequalität QoS0 wird “höchstens einmal” zugestellt und dementsprechend verworfen, wenn der Subscriber nicht erreichbar ist. Eine Nachricht mit der Servicequalität QoS1 wird so oft zugestellt, bis der Subscriber den Empfang bestätigt hat. Servicequalität QoS2 stellt sicher, dass der Subscriber die Nachricht genau einmal erhält.

## Verbindungsabbrüche beim Publisher:

**Last Will und Retained Messages**
In seinem “letzten Willen” teilt ein Publisher dem Broker beim Verbindungsaufbau mit, welche Nachricht im Falle eines plötzlichen Verbindungsabbruchs an die Subscriber gesendet werden soll. Für jeden Publisher kann genau ein letzter Wille angelegt werden.

Sendet ein Publisher eine Nachricht, in der das Retained-Flag gesetzt ist, speichert der Broker diese Nachricht für ein Topic zwischen. Diese Nachricht wird einerseits gesendet an Subscriber, die das Topic neu abonnieren, andererseits an Subscriber mit der Servicequalität 1, die noch keine Empfangsbestätigung gesendet haben. Eine Beispielanwendung ist die Übermittlung des letzten bekannten Messwerts eines Temperatursensors. Für jedes Topic kann genau eine Nachricht vorgehalten werden.

## Gesicherte Kommunikation:




**Authentifizierung und Verschlüsselung**
MQTT unterstützt die Authentifizierung der Clients über Benutzernamen und Passwort. Dadurch lassen sich auf der Brokerseite Berechtigungen vergeben und so beispielsweise sicherstellen, dass nur der Temperatursensor vor Ort Nachrichten mit dem Topic f2/kuehlthekte/temp veröffentlichen darf. Benutzername und Passwort werden beim Verbindungsaufbau übermittelt und unverschlüsselt übertragen. Daher ist es eine gute Idee, jegliche MQTT-Kommunikation von Anfang an zu verschlüsseln. Da das Protokoll auf den oberen Schichten des OSI-Modells zu finden ist, lässt sich die Verschlüsselung problemlos mit TLS realisieren - sofern das entsprechende Endgerät über die dazu notwendigen Ressourcen verfügt.

## Universell und portabel

MQTT hat nicht nur einen sehr einfachen Grundaufbau, es bietet auch große Freiheiten bei der Gliederung der Topics und macht keine Vorgaben zum Inhalt der Payload. Somit ist MQTT ein sehr generisches und vielseitig einsetzbares Protokoll. Auch bei der Wahl eines Brokers gibt es kaum Einschränkungen: Diese sind als freie Software und von etablierten kommerziellen Anbietern erhältlich, als Service im Web oder als Hardware-Appliance. Eine Anwendung, die für die

Kommunikation MQTT verwendet, funktioniert mit jedem beliebigen MQTT-Broker. Somit ist sichergestellt, dass ein Plattformwechsel jederzeit und ohne besonderen Aufwand möglich ist.

## Selber ausprobieren

Sie möchten gerne MQTT ausprobieren? Unser Web-IO Digital ist der ideale Einstieg in den IoT-Kosmos. Ganz einfach das Web-IO Digital als Muster anfordern.

Wird das Muster innerhalb von 30 Tagen zurückgegeben, tragen sie nur die Rücksendekosten (BRD). Möchten Sie das Testgerät behalten zahlen Sie einfach beiligende Rechnung.

## Fragen zu Web-IO Digital mit MQTT?

Herr Thiel hilft Ihnen gerne weiter.
Telefon: 0202/2680-110 (Mo-Fr. 8-17 Uhr)
E-Mail: f.thiel@wut.de

W&T
www.WuT.de

Wir sind gerne persönlich für Sie da:

Wiesemann & Theis GmbH
Porschestr. 12
42279 Wuppertal
Tel.: 0202/2680-110 (Mo-Fr. 8-17 Uhr)
Fax: 0202/2680-265
info@wut.de