**MQTT**

## Geschichte des MQTT-Protokolls

MQTT oder wie früher MQ Telemetry Tranpsort geheißen ist ein etabliertes Kommunikationsprotokoll zwischen Maschinen und insbesondere in dem IoT Sektor. Im Jahr 1999 benötigte die damalige Gas- und Ölindustrie einen Kommunikationsweg (Protokoll) mit minimaller Bandbreite und Batterieverlust, um Ölpipelines über Satelliten zu überwachen. So entstand ein mächtiges Protokoll, welches heute fast ausschließlich für die Kommunikation zwischen Maschinen und IoT Geräten verwendet wird.

## Prinzip des MQTT-Protokolls

Im Gegensatz zu der herkömmlichen Kommunikation, welcher eine direkte Verbindung zwischen Client und Server anbietet, entkoppelt das MQTT den Publisher (Herausgeber der Nachricht) und Subscriber (Empfänger der Nachricht) mithilfe eines Brokers (Server).   
Die Entkoppelung lässt sich in drei Kategorien unterteilen:

* **Entkopplung von Platzbedarf:** Der Publisher und der Subscriber kennen den Standort des jeweils anderen Netzwerkes nicht und tauschen somit keine Informationen wie IP-Adressen oder Portnummern aus.
* **Entkopplung von Zeit:** Publisher und Subscriber sind nicht im gleichen Netz verbunden und laufen nicht parallel.
* **Entkopplung der Synchronisation**: Publisher und Subscriber haben die Möglichkeit Nachrichten zu senden oder zu empfangen, ohne sich gegenseitig zu unterbrechen.

Obwohl MQTT nicht RESTful ist, bietet er mithilfe von WSS (WebSockets) den direkten Empfang von Daten in einem Webbrowser. Auf der Message werden zusätzliche Header hinzugefügt und mithilfe eines definierten JavaScript-Clients kann eine solche Kommunikation erfolgen. Die Prinzipien der MQTT-Kommunikation ändern sich nicht. Die Daten werden vom Broker (Server) dem Webbrowser und nicht direkt von einem Client zur Verfügung gestellt.

## MQTT-Client

Client heißt jedes Gerät (Server, Microcontroller, Rechner), auf dem eine MQTT-Bibliothek läuft. Der Client, welcher Nachrichten sendet, heißt Publisher (Herausgeber) und der Client, welcher Nachrichten empfängt, heißt Subscriber (Empfänger).

## MQTT-Broker

Der MQTT-Broker (Server) hat folgende Aufgabe: die Kommunikation zwischen den verschiedenen Clients zu steuern und zu koordinieren. Sen Aufgaben ist der Empfang, Filterung und Weiterleitung der Messages. Darüber hinaus ist er auch zuständig für die Autorisierung und Authentifizierung von MQTT-Clients. Im Weiteren hat er folgende Verantwortung: die Nachrichten an anderen Systemen zur weiteren Analyse weiterzugeben und zu guter Letzt beschäftigt sich mit verpassten Nachrichten und Client-Sitzungen.

## MQTT-Verbindung

Clients und Broker stellen eine MQTT-Verbindung her. Eine CONNECT-Nachricht wird vom Client an den MQTT-Broker verschickt und der antwortet mit einer CONNACK-Nachricht. Beide Seiten benötigen einen TCP/IP Stack, um miteinander zu kommunizieren. Es ist wichtig zu beachten, dass Client nie miteinander kommunizieren, sondern nur mit dem Broker.

## MQTT-Thema

Ist ein Schlüsselwort, das der Broker verwendet, um Nachrichten zu filtern. Es gibt auch eine hierarchische Struktur wie beim Datei- Ordnerverzeichnis.  
Ein Beispiel wäre:  
Fabrik x/Anlage x/Maschine

## MQTT-Datenformat

Die Veröffentlichung von Messages (Thema und Daten) erfolgt in Byte-Format. Der Client bestimmt das Datenformat, und zwar zum Beispiel Textdaten, Binärdaten, XML- oder JSON-Dateien.

## MQTT-Sicherheit

Drei Konzepte sind grundlegend für die MQTT-Sicherheit.

* Identität: Die Benennung des Clients
* Authentifizierung: Die Überprüfung der Client-Identität
* Berechtigung/Autorisierung: Verwaltung der Berechtigung, die dem Client erteilt werden.

Der MQTT-Broker (Server) verwendet sein digitales Zertifikat, um vom Client identifiziert zu werden und der MQTT-Client verwendet das SSL-Protokoll, um das von Server gesendete Zertifikat zu authentifizieren.

## Fazit

Insgesamt ist MQTT ein sicheres Protokoll, welches in den meisten Fällen für die Kommunikation zwischen Maschinen verwendet wird. Seine minimalle Bandbreite ermöglicht seine Einsetzung in Geräten und Microcontrollern, die nicht eine „leistungsstarke“ Prozessgeschwindigkeit besitzen.