

## WARUM MQTT FÜR INDUSTRIE 4.0

## WHY MQTT FOR INDUSTRY 4.0



**Joerg Wende**  
Client Technical Specialist  
IBM Germany GmbH, SW Group  
jwende@de.ibm.com  
www.ibm.com

Stärke des Standorts Deutschland sind die Technologieschnittstellen. Kommen Maschinenbau, Elektrotechnik und Automobilbau mit der IT-Industrie zusammen, entstehen riesige Chancen. Es sind aber große Herausforderungen zu bewältigen, zum Beispiel eine Architektur-Plattform für Industrie 4.0.

Die Industrie 4.0 Platform Initiative arbeitet derzeit an einer gemeinsamen Architektur für eine solche Plattform. IBM ist als Marktführer im Bereich Service Oriented Architecture (SOA) eine der treibenden Kräfte.

Für die gemeinsame Architektur, die die horizontale und vertikale Integration von Geräten und Anwendungen vereinfachen und möglichst flexibel gestalten soll, gibt es viele Kriterien zu berücksichtigen. Zentral ist die Wahl der Protokolle, um die Integration zu vereinfachen und damit die Kosten zu optimieren. Wichtige Einflussgrößen sind auch nicht-funktionale Anforderungen, wie Echtzeit- oder Sicherheitsanforderungen.

Aus den gesetzten Auswahlkriterien von Industrie 4.0 empfiehlt sich MQTT als das Kommunikations- und Kooperationsprotokoll für die von der Industrie 4.0 vorgegebene vertikale und horizontale Integration.

### MQTT – Übersicht

MQTT wurde für den bidirektionalen Nachrichtenaustausch von kompakten Endgeräten mit geringer Bandbreite, kostenintensiven Verbindungen oder herstellerspezifischen Protokollen und für eine zuverlässige und sichere Nachrichtenübertragung entwickelt. Die Spezifikation ist beim OASIS-Konsortium zur Standardisierung eingereicht.

MQTT adressiert Limitierungen im Netzwerkbereich, die durch folgende Faktoren hervorgerufen werden:

- Begrenzte Bandbreite
- Volumenbegrenzungen
- Unsichere Verbindungen

Hauptmerkmale des Protokolls sind

- Einfachheit
- Sicherheit
- Zuverlässigkeit
- Skalierbarkeit

Einige dieser Hauptmerkmale werden im Folgenden erläutert.

### MQTT – das Protokoll Publish/Subscribe

MQTT nutzt ein asynchrones, nachrichtenorientiertes Übertragungsprotokoll unter Verwendung von Publish/Subscribe, bei dem eine sendende Anwendung (Publisher) die Information nicht direkt an einen Empfänger (Subscriber) versendet, sondern diese mit einer Klassifizierung (Topic) versieht und an einen Vermittler (Broker) sendet. Ein Empfänger registriert sich unter der Angabe von einem oder mehreren Topics beim Broker. Wenn zu einem Topic Nachrichten veröffentlicht (publiziert) wurden, übernimmt der Broker die Verteilung an alle registrierten Empfänger. Der Broker fügt damit auch automatisch eine Entkopplung zwischen sendenden und empfangenden Anwendungen ein. Die Verbindungen zwischen den Anwendungen und dem Broker können zusätzlich durch Autorisierung und Verschlüsselung gesichert werden.

### Effizienz und Wirtschaftlichkeit

MQTT kann auf kompakten Endgeräten folgende Limitierungen adressieren:

- Begrenzungen in der Verarbeitungsgeschwindigkeit oder beim Speicherplatz
  - Java Library: 53 kBytes
  - C Library: 88 kByte
- Restriktionen beim Einsatz von speziellen Kommunikationsprotokollen
- Begrenzter Energieverbrauch bei mobilen Endgeräten

Eine Untersuchung zum Energieverbrauch ergab signifikante Unterschiede in der Batterielaufzeit von mobilen Endgeräten:

### MQTT – Programmierung

Die Programmierschnittstelle für MQTT ist einfach gehalten und beinhaltet im Prinzip nur vier API Aufrufe:

- Herstellung der Verbindung
- Senden (Publizieren) von Nachrichten
- Empfangen (Subskription)
- Beenden der Verbindung

Die Implementierungen für die Programmiersprachen Java und C wurden von IBM an die Eclipse Foundation übergeben und stehen über das Eclipse Paho Projekt zur Verfügung (EPL). Clients für andere Programmiersprachen können über die Webseite MQTT.org gefunden werden.

## Anwendungsbeispiele für MQ Telemetrie Transport

Durch die vielfältigen Eigenschaften des Protokolls wird MQTT bereits in einer großen Anzahl von Industrien eingesetzt. Beispiele sind Track & Trace-Anwendungen, die Überwachung von Messwerten (z.B. Energie, Wasser, Durchfluss, Temperatur) und die Fahrzeugkommunikation. ▲

▲ *Selection criteria for a communication protocol in the context of Industry 4.0*

*The strength of Germany is the technology interface. Mechanical, electrical and automotive engineering come together with the IT industry, creating huge opportunities. However, there are major challenges to deal with, for example, an architecture platform for Industrial 4.0.*

*The Industry 4.0 Platform initiative is currently working on a common architecture for such a platform. IBM is the market leader and one of the driving forces in the area of Service Oriented Architecture (SOA).*

*For the common architecture that is designed to simplify the horizontal and vertical integration of devices and applications and make them as flexible as possible, there are many criteria to consider. Central is the choice of protocols to facilitate integration and thus to optimise the costs. Important factors are also non-functional demands, such as real-time or safety requirements.*

*From the set selection criteria of Industry 4.0, MQTT is recommended as the communication and co-operation protocol as predetermined by the vertical and horizontal integration of Industry 4.0.*

## MQTT Overview

*MQTT is designed for the bi-directional exchange of messages between compact devices with low bandwidth connections or costly vendor-specific protocols as well as for reliable and secure message transmission. The specification has been submitted to the OASIS consortium for standardisation.*

*MQTT addressed limitations in the network area, which are caused by the following factors:*

- Limited bandwidth
- Volume restrictions
- Unsafe connections

*Main features of the protocol are*

- Simplicity
- Security
- Reliability
- Scalability

*Some of these main features will be explained below.*

## MQTT – the protocol Publish / Subscribe

*MQTT uses an asynchronous, message-oriented transmission protocol using Publish/Subscribe, in which a sending application (Publisher) does not send the information directly to a receiver (Subscriber), but sends with a classification (Topic) band to an intermediary (Broker). A receiver is registered under the specification of one or more Topics from the Broker. If a Topic publishes news (published), the Broker takes over the distribution to all registered recipients. The Broker automatically adds a decoupling between sending and receiving applications. The connections between the applications and the Broker may be additionally secured by authorisation and encryption.*

## Efficiency and economy

*On compact devices MQTT can address the following limitations:*

- limitations in processing speed or storage space
  - Java Library: 53 kBytes
  - C Library: 88 kByte
- restrictions on the use of special



**Renate Franken**  
Client Technical Specialist  
IBM Deutschland GmbH, SW Group  
Renate.Franken@de.ibm.com  
www.ibm.com

- communication protocols*
- *restricted energy consumption in mobile devices*

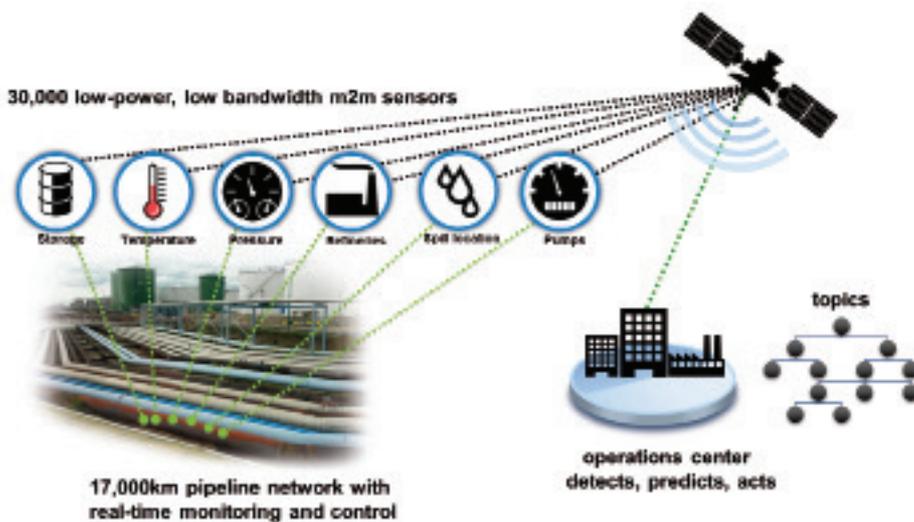
*A study on energy consumption revealed significant differences in the battery life of mobile devices:*

## MQTT – Programming

*The programming interface for MQTT is simple and, in principle, includes only four API calls:*

- *Preparation of connection*
- *Sending (Publishing) of messages*
- *Receiving (Subscription)*
- *Terminating the connection*

*The implementations for the programming languages Java and C were transferred from IBM to the Eclipse Foundation and are available for use on the Eclipse Paho project (EPL). Clients for other programming languages can be found on the website MQTT.org.*



## Application examples for MQ Telemetry Transport

*Due to the many features of the protocol, MQTT is already used in a large number of industries. Examples are track & trace applications, monitoring of measured values (e.g. energy, water, flow, temperature) and vehicle communication.▲*

Typischerweise erstellen OEMs und Entwickler ein M2M-Gesamtsystem aus Einzelkomponenten selbst. Der Trend geht jedoch in Richtung Vorintegration, wobei verschiedene Hardware-Komponenten mit integrierten Softwareentwicklungs- und -verwaltungstools in einem Modul kombiniert werden.

Dadurch bietet sich eine attraktive Alternative zu herkömmlichen Ansätzen. Mit einem M2M-Ökosystem auf einem einzigen Modul können OEMs die Größe und den Energieaufwand für ihre Lösungen reduzieren, die Sicherheit erhöhen und die Material- und Installationskosten senken.

Dazu sind drei Bestandteile entscheidend:

1. Eine integrierte Hardware-Lösung, die die drahtlose Verbindung mit dem Application Prozessor und den Schnittstellen zur Vereinfachung des Designs und der Entwicklung kombiniert
2. Instrumente zur Entwicklung und Application Frameworks für betriebsfertige und bewährte Softwarebausteine
3. Eine Plattform für die Verwaltung der Geräte und die Anwendungsentwicklung

Bisher wird für Standardanwendungen von M2M-Lösungen wie intelligente Stromzähler, industrielle Überwachungssysteme, mobile Geräte zur Erfassung des Gesundheitszustands, Connected-Home-Anwendungen oder vernetzte Automaten jeweils eine Systemarchitektur mit einem drahtlosen Modem oder einem Modul für Mobilfunk entwickelt. Diese Vorgehensweise gilt jedoch als überholt, da sich die M2M-Entwickler hierbei zu lange mit der Basisarbeit beschäftigen müssen. Zudem ist dieser Ansatz mit erheblichen Kosten verbunden und verzögert die Markteinführung.

Moderne Lösungen bieten daher eine vollständig entwickelte Systemarchitektur mit Mobilfunkverbindung sowie einem dedizierten Mikroprozessor, um eine M2M-Anwendung auf einem einzigen integrierten Multikern-Modul betreiben zu können.

Frühere Smartphone-Versionen wurden mit einem Standalone-Prozessor und Modems erstellt, die vom Hersteller integriert worden sind. Mittlerweile nutzen die meisten Anbieter vollständig integ-