

Verteilte Systeme

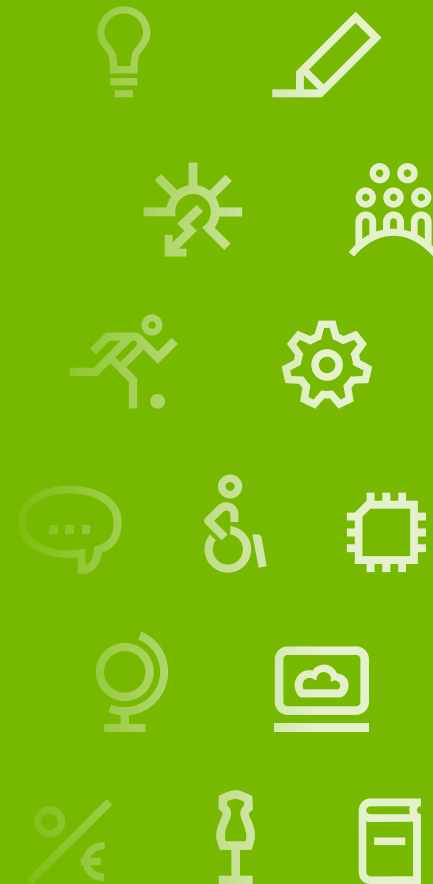
MQTT

Tarik Selcuk Balci (582157)

Sean Bien (589477)

Sebastian Gey (589413)

Ilkkan Berke Savasci (587653)



Hochschule für Technik
und Wirtschaft Berlin

University of Applied Sciences

Gliederung

1 Theoretischer Teil

.1 Was ist MQTT?

.2 Entstehung

.3 Control Packet Format

.4 Publish/Subscribe-Modelle

.5 Funktionen von MQTT

.6 Fazit

2 Praktischer Teil

Was ist MQTT?

»It is a publish/subscribe, extremely simple and lightweight messaging protocol, designed for constrained devices and low-bandwidth, high-latency or unreliable networks.« ^[1]

Entstehung [Vgl. 2]

1999 entwickelt von Andy Stanford-Clark (IBM) und Arlen Nipper (Arcom, nun Cirrus Link)

Zweck: Protokoll, welches Verbindungen zwischen Ölpipelines und Satelliten ermöglichen sollte und dabei einen geringen Batterieverlust und eine geringe Bandbreite ermöglicht.

Früher Akronym für MQ (**M**essage **Q**ueuing) **T**elemetry **T**ransport

Heutzutage nur noch Name des Protokolls

Control Packet Format [Vgl. 3]

Paket besteht aus bis zu drei Bestandteilen

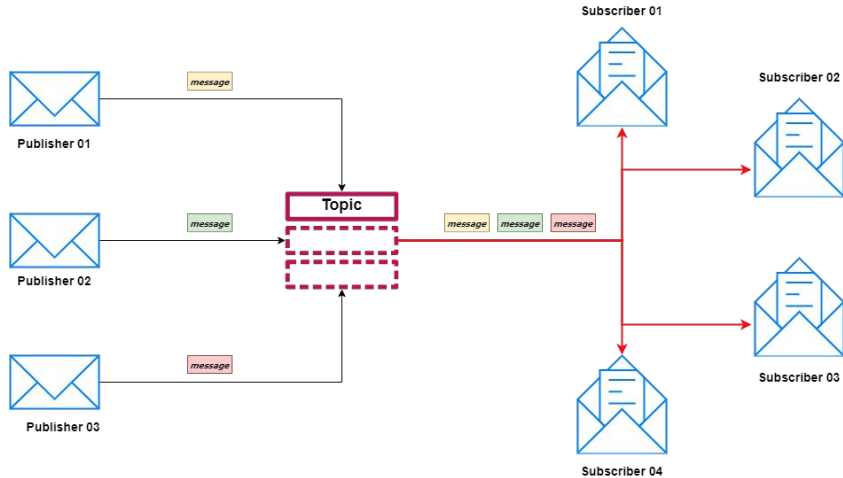
1. Fixen Header (Pflicht)
2. Variablen Header
3. Payload

Bit	7	6	5	4	3	2	1	0
byte 1	Type				Flags			
byte 2...	Remaining Length							

Fixer Header

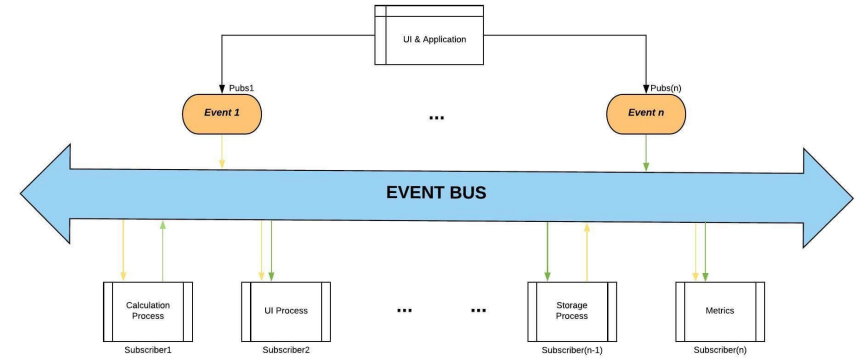
Pub/Sub Modelle [Vgl. 4, 5]

Shared Database basierte Modell:



[Abb. 1]

Event-Bus basierte Modell:

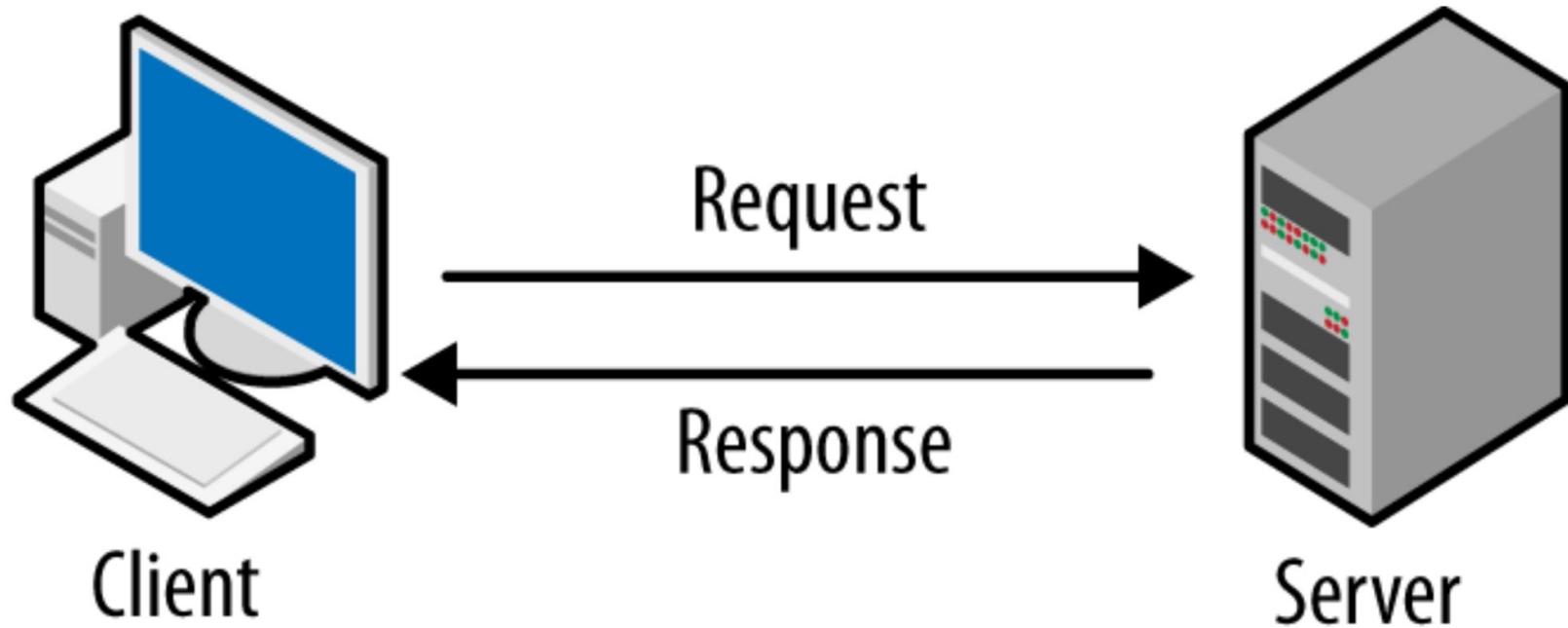


Event Bus Overview

source: n8n.io

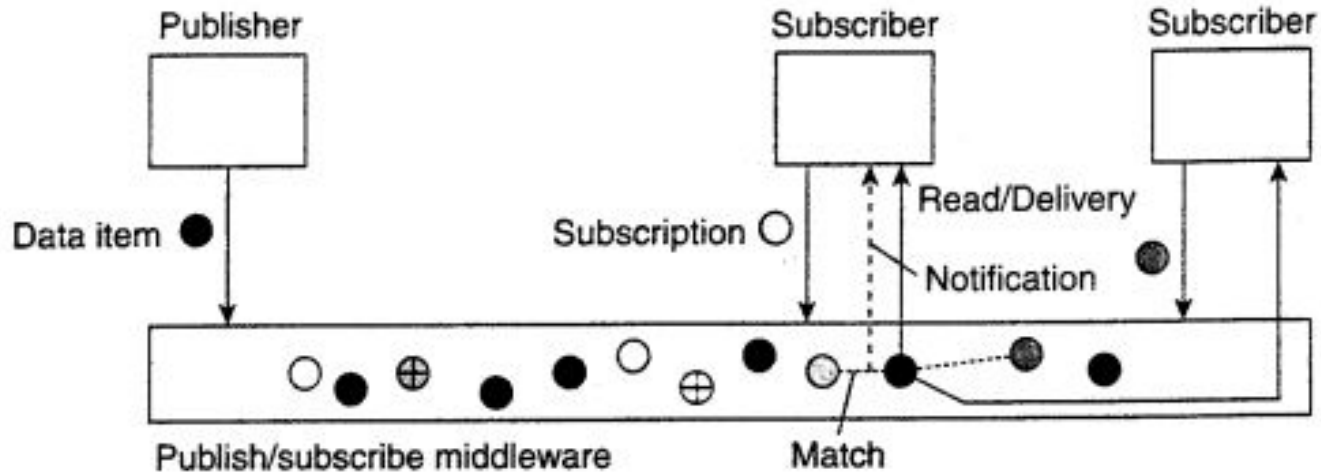
[Abb. 2]

Client-Server



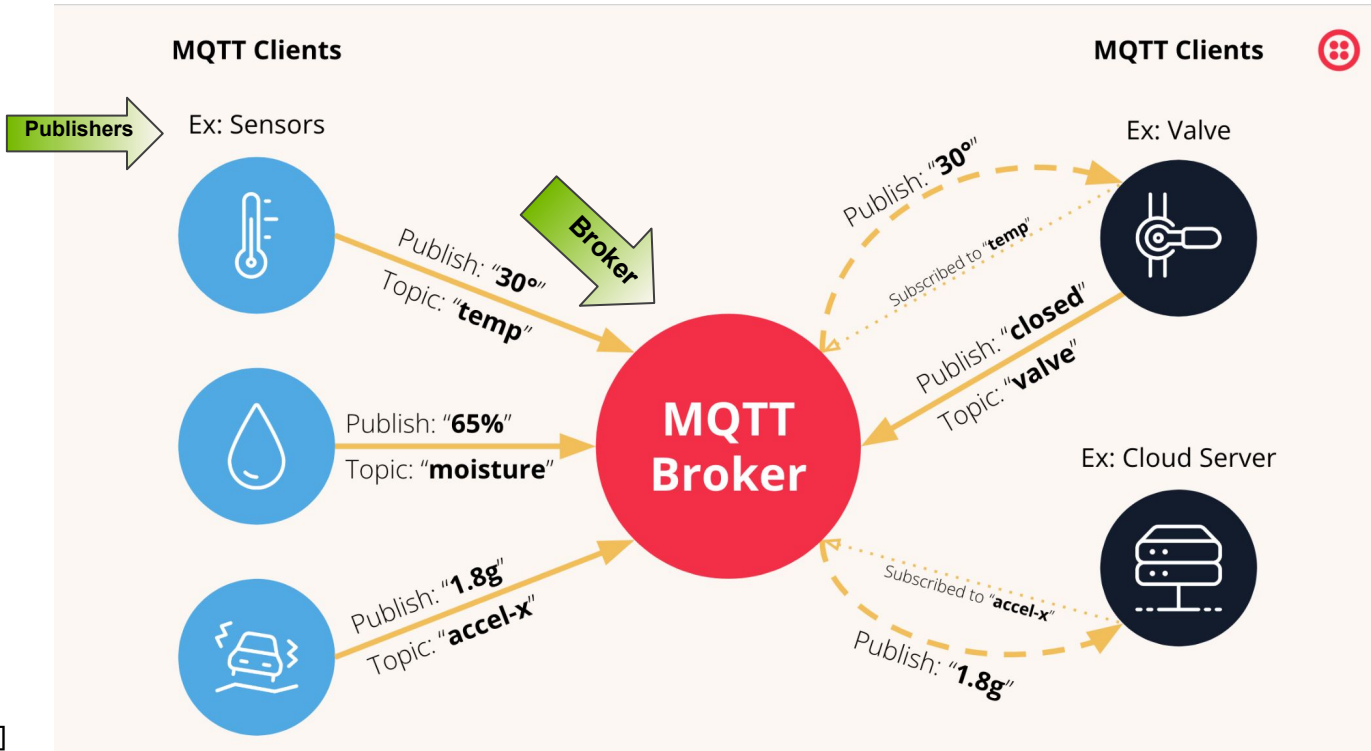
[Abb. 3]

Wie funktioniert ein Pub/Sub Systeme? [Vgl. 6]



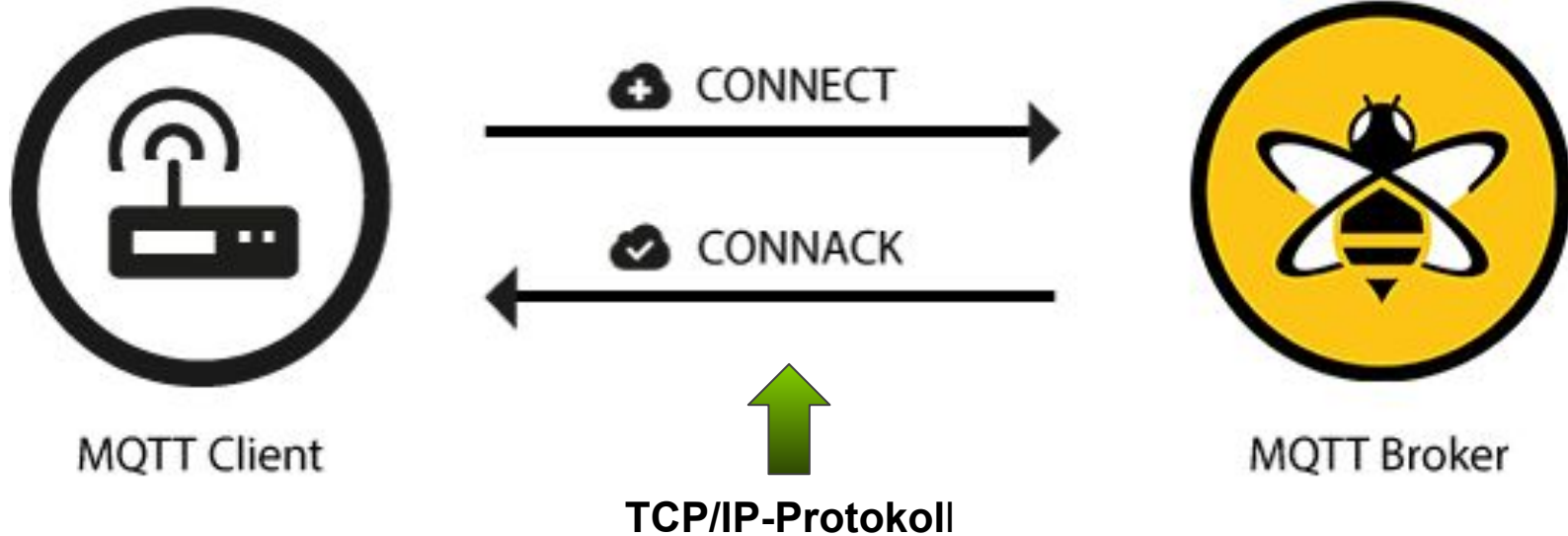
[Abb. 4]

Wie funktioniert MQTT? [Vgl.7, 8]



[Abb. 5]

MQTT Broker - Client Connection^[Vgl. 3, 8]



[Abb. 6]

Nachricht Beispiele

Ein Publish Message:

```
Topic: "home/room_1/temperature"  
QoS: 1  
Payload: "23.5C"
```

Eine Last Will and Testament Message:

```
Topic: "device/status"  
Payload: "Connection Lost!"
```

Funktionen von MQTT^[Vgl. 3, 9]

Quality of Service (QoS)^[Vgl. 10]

- QoS 0: “At most once”
- QoS 1: “At least once”
- QoS 2: “Exactly once”

Keep Alive^[Vgl. 11]

Last Will and Testament (LWT)^[Vgl. 12]

Retained Messages^[Vgl. 13]

Fazit

Charakteristika Verteilter Systeme:

- Kapselung, Schnittstellenbildung und Geheimnisprinzip:
⇒ Subscriber kennt Implementierung von Broker und Publisher nicht
- Verteilung:
⇒ Pub sendet Nachricht an Broker, Broker sendet an Sub, → Broker für Austausch von Nachrichten und Koordination
- Parallelität/Nebenläufigkeit:
⇒ Siehe Pub/Sub-Model
- Kommunikation:
⇒ Daten/Nachrichten werden ständig ausgetauscht
- Quantitative Zeit:
⇒ System läuft in Echtzeit und reagiert auf Änderungen der Umwelt (z.B. Keep Alive)
- Fehlertoleranz/-resilienz:
⇒ Umgang mit Fehler, bei der Übermittlung z.B. durch QoS, aber auch beim Abbruch der Verbindung des Publishers durch LWT

Sean Quelle (Inhalte)

[1] <https://mqtt.org/faq/> [Letzer Zugriff: 09.01.2025]

[2] <https://www.hivemq.com/blog/mqtt-essentials-part-1-introducing-mqtt/> [Letzer Zugriff: 09.01.2025]

[3] <https://docs.oasis-open.org/mqtt/mqtt/v5.0/os/mqtt-v5.0-os.html> [Letzer Zugriff: 09.01.2025]

[9] <https://www.hivemq.com/mqtt/> [Letzer Zugriff: 09.01.2025]

[10] <https://www.hivemq.com/blog/mqtt-essentials-part-6-mqtt-quality-of-service-levels/> [Letzer Zugriff: 09.01.2025]

[11] <https://www.hivemq.com/blog/mqtt-essentials-part-10-alive-client-take-over/> [Letzer Zugriff: 09.01.2025]

[12] <https://www.hivemq.com/blog/mqtt-essentials-part-9-last-will-and-testament/> [Letzer Zugriff: 09.01.2025]

[13] <https://www.hivemq.com/blog/mqtt-essentials-part-8-retained-messages/> [Letzer Zugriff: 09.01.2025]

Tarik Quelle (Inhalte)

[3] <https://docs.oasis-open.org/mqtt/mqtt/v5.0/os/mqtt-v5.0-os.html> (Zugriff am: 09.01.2025)

[4] Google Cloud. (2024). *Pub/Sub Übersicht*. Verfügbar unter: <https://cloud.google.com/pubsub/docs/overview?hl=de> (Zugriff am: 09.01.2025)

[5] Shared Database & Event Bus Modell
<https://www.enterpriseintegrationpatterns.com/patterns/messaging/SharedDataBaseIntegration.html>
(Zugriff am: 11.01.2025)

[6] Tanenbaum, A. & Van Steen, M. (2007). *Distributed Systems: Principles and Paradigms* (2nd Ed.).

[7] **F5 DevCentral (2021)**. *MQTT Protocol Deep Dive*. Verfügbar unter: <https://www.youtube.com/watch?v=eS4nx6tLSLs> (Zugriff am: 09.01.2025)

[8] Official MQTT Hive Website, <https://www.hivemq.com/blog/mqtt-essentials-part-3-client-broker-connection-establishment/>,
(Zugriff am: 10.01.2025)

Abbildungsverzeichnis - Tarik

1. <https://s7280.pcdn.co/wp-content/uploads/2021/07/Basics-Of-Pub-Or-Sub-Messaging.png>
(Zugriff am: 08.01.2025)
2. <https://www.stefanofrasca.com/static/77cad28a9cd05fd85381a0b5de9941ce/EventBusOverview.jpeg>
(Zugriff am: 08.01.2025)
3. https://darvishdarab.github.io/cs421_f20/assets/images/client-server-1-d85a93ea16590c10bed340dd78294d0d.png
(Zugriff am: 08.01.2025)
4. Andrew S. Tanenbaum, Maarten Van Steen – *Distributed Systems: Principles and Paradigms*, S. 592
5. https://www.twilio.com/content/dam/twilio-com/global/en/blog/legacy/2023/what-is-mqtt/MQTT_Diagram_gOmDdU4.png (Zugriff am: 10.01.2025)
6. <https://www.hivemq.com/sb-assets/f/243938/550x171/81094f3d9d/connect-flow.gif> (Zugriff am: 09.01.2025)

2. Praktischer Teil