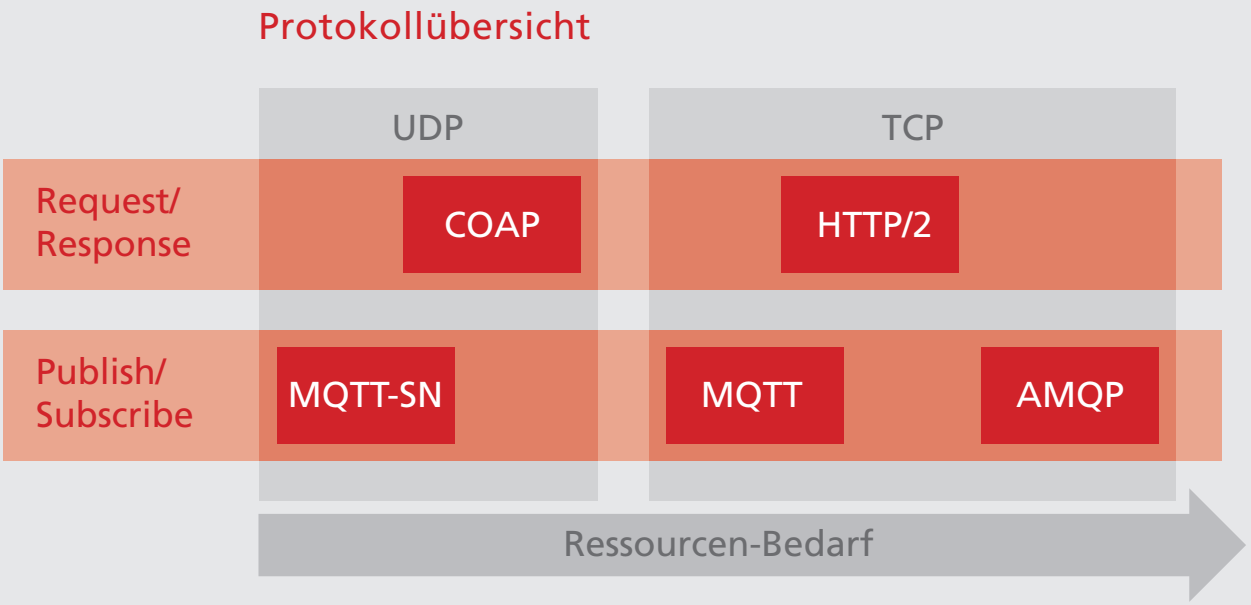
bbv
MAKING VISIONS WORK.



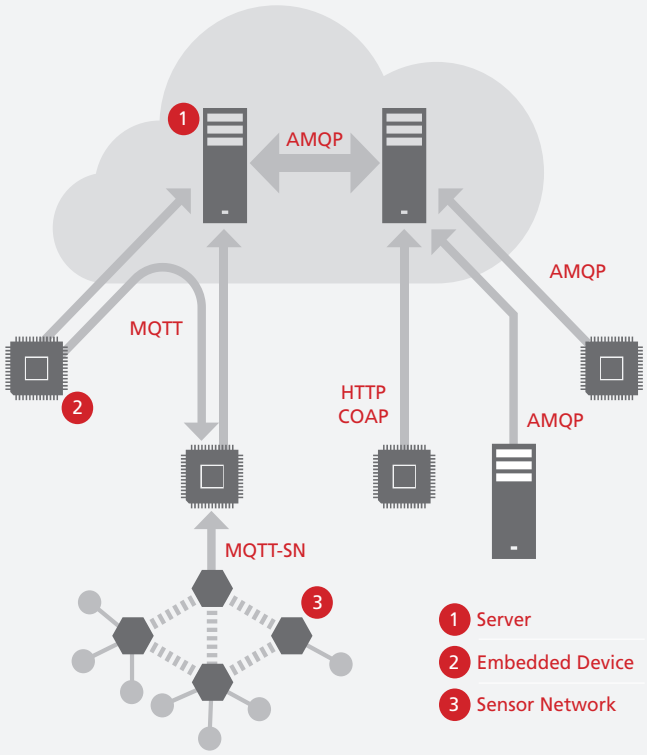
TECHNICAL ADVICE

DURCHBLICK IM DSCHUNGEL DER IOT-PROTOKOLLE

In naher Zukunft werden die meisten Geräte ans Internet angeschlossen sein. Um Aufgaben effektiv und effizient zu bewältigen, müssen die Geräte miteinander kommunizieren. Bei der Auswahl bildet die Vielzahl der vorhandenen Protokolle einen dichten Dschungel. Dieses Factsheet dient als Guide bei der Findung eines geeigneten Protokolls.

Einsatzszenario

Bei der Auswahl eines IoT-Protokolls spielt das Einsatzszenario eine entscheidende Rolle. Der Entwickler muss diverse Aspekte beachten. Kommunizieren die Geräte untereinander oder mit einem Server? Kann die Infrastruktur am Einsatzort mitgestaltet werden oder muss mit einer unbekannten Umgebung gerechnet werden? In welcher Grössenordnung muss die Lösung skalierbar sein? Ist ein Austausch zwischen Geräten unterschiedlicher Hersteller geplant?



Message Pattern

Die Art der Kommunikation lässt sich grob in zwei Pattern einteilen. In Client-Server-Architekturen wird oft das Request/Response Pattern verwendet. Der Client schickt einen Request an den Server. Dieser prozessiert die Anfrage, erzeugt eine Antwort und sendet diese zurück an den Client. Beim Publish/Subscribe Pattern ordnen Sender (Publisher) ihre Message einer Klasse zu und senden diese an einen Message Broker, der sie an Abonnenten (Subscriber) weiterleitet.

Transport Layer

IoT-Protokolle verwenden häufig TCP oder UDP. Während UDP verbindungslos und unzuverlässig ist, stellt TCP einen zuverlässigen Kommunikationskanal bereit. Dafür bezahlt man bei TCP mit grösserer Code Size, längeren Latenzzeiten und mehr Bandbreite. Beide Protokolle erlauben eine verschlüsselte Kommunikation.

Presentation Layer

Durch die Auswahl des IoT-Protokolls ist der Presentation Layer nicht definiert. Hier kommen Datenformate wie JSON, XML oder CBOR zum Einsatz.

Quality of Service

Trotz Einsatz von TCP im Transport Layer können im Session Layer Messages verloren gehen. Aus diesem Grund unterstützen manche Protokolle die Definition unterschiedlicher Quality of Service (QoS). MQTT, MQTT-SN und AMQP bieten die Optionen, eine Message höchstens, mindestens oder exakt einmal zuzustellen.

Protokolle

Für alle hier vorgestellten Protokolle existiert eine Vielzahl an Implementierungen für alle gängigen Plattformen und Betriebssysteme. Es handelt sich dabei um offene Standards.

MQTT

MQTT ist ein auf TCP basierendes Publish/Subscribe-Protokoll, das sich für Geräte mit beschränkten Ressourcen eignet. Geräte können eine Testament-Message hinterlegen. Diese wird vom Broker verschickt, wenn sich das Gerät nicht mehr meldet. Messages können als «Retained» markiert werden. So erhält ein Subscriber beim Abonnieren unmittelbar die letzte Retained Message, um nicht auf ein Update warten zu müssen.

MQTT-SN

MQTT-SN ist eine Variation von MQTT unter Verwendung von UDP. Geräte können in einen Schlaf-Modus gehen. Während dieser Zeit speichert der Broker die Messages, um sie später zuzustellen.

AMQP

AMQP ist ein Publish/Subscribe-Protokoll, das aus dem Finanzsektor stammt. Es eignet sich für Geräte mit mehr Ressourcen in Message orientierten Szenarien. Dies zeigt sich in Features wie Transaktionen, flexibles Routing und Metadaten für Messages. Auf dem Broker lassen sich mehrere Queues und Zugriffsrechte konfigurieren. Damit lässt sich z. B. steuern, ob eine Message an einen oder alle Subscriber geht.

HTTP/2

HTTP ist ein etabliertes Request/Response-Protokoll. Dazu existieren z. B. viele Webserver, Analysetools und Literatur. HTTP/2 ist einfacher zu verarbeiten als HTTP/1.1, erlaubt standardmässig eine Kompression und führt Server Push ein. Mit Letzterem kann der Server eine Message an das Gerät schicken, ohne dass dieses einen Request getätigt hat.

CoAP

CoAP ist ein Request/Response-Protokoll nach der Idee von HTTP mit dem Fokus auf Geräte mit beschränkten Ressourcen. Im Gegensatz zu HTTP ist CoAP binär und verwendet UDP. CoAP unterstützt Service Discovery. Zusätzlich können Clients auch Ressourcen eines Servers abonnieren.

bbv
MAKING VISIONS WORK.

MAKING VISIONS WORK.

bbv Software Services, +41 41 429 01 11
www.bbv.ch · info@bbv.ch