

# Wichtige Funktionen & Eigenschaften

Die Kern-Funktionalitäten lassen sich in mehrere Kategorien unterteilen:

## Skalierbarkeit & Verfügbarkeit

- Horizontal skalierbar: Clusterbetrieb mit identischen Knoten ohne Master-Single-Point. [ThingsBoard](#)
- Fehlertoleranz: kein zentraler Koordinator, jeder Knoten kann Clients bedienen und Nachrichten routen. [Medium](#)
- Dauerhaftigkeit (Durability): Nachrichten und Sessions können auch bei Node-Ausfällen gesichert werden. [ThingsBoard](#)

## Kommunikationsmuster

TBMQ unterstützt typische IoT-Szenarien:

- „Fan-in“: Viele Geräte senden Daten an wenige Anwendungen.
- „Fan-out“: Wenige Quellen senden Updates, viele Geräte abonnieren.
- P2P (Point-to-Point): Direkte Kommunikation Gerät ↔ Gerät oder Gerät ↔ Anwendung über dedizierte Topics.

## Sicherheits- & Managementfunktionen

- Authentifizierung: z. B. via JWT, X.509 Zertifikate. [ThingsBoard](#)
- Authorisierung/ACL: Zugriffskontrollen basierend auf ClientID, Username oder Zertifikat. [ThingsBoard](#)
- Monitoring & Management: Sessions- und Subscription-Überwachung, Clients-Metriken. [ThingsBoard+1](#)

## Integration mit externen Systemen

- Weiterleitung von MQTT-Nachrichten an externe Systeme wie HTTP, Kafka oder andere MQTT-Broker. [GitHub](#)
- Unterstützung von MQTT über WebSocket. [ThingsBoard](#)

## Technische Architektur

- Intern nutzt TBMQ Plattformen wie Apache Kafka und Redis je nach Einsatzszenario für persistente Sessions, hohe Last, geringe Latenz. [DEV Community](#)

- Datenbank für Metadaten (z. B. PostgreSQL) wird eingesetzt, aber nicht als Bottleneck im Messaging-Pfad.
- 

Revision #1

Created 2026-01-13 06:51:03 UTC by Marcel Sauer

Updated 2026-01-13 06:51:37 UTC by Marcel Sauer