ALIBABA CLOUD

Alibaba Cloud

Alibaba Cloud Message Queue for MQTT

Comparison Between MQTT and MQ

Document Version: 20210311

Alibaba Cloud

Legal disclaimer

Alibaba Cloud reminds you to carefully read and fully understand the terms and conditions of this legal disclaimer before you read or use this document. If you have read or used this document, it shall be deemed as your total acceptance of this legal disclaimer.

1. You shall download and obtain this document from the Alibaba Cloud website or other Alibaba Cloud-authorized channels, and use this document for your own legal business activities only. The content of this document is considered confidential information of Alibaba Cloud. You shall strictly abide by the confidentiality obligations. No part of this document shall be disclosed or provided to any third party for use without the prior written consent of Alibaba Cloud.

2. No part of this document shall be excerpted, translated, reproduced, transmitted, or disseminated by any organization, company or individual in any form or by any means without the prior written consent of Alibaba Cloud.

3. The content of this document may be changed because of product version upgrade, adjustment, or other reasons. Alibaba Cloud reserves the right to modify the content of this document without notice and an updated version of this document will be released through Alibaba Cloud-authorized channels from time to time. You should pay attention to the version changes of this document as they occur and download and obtain the most up-to-date version of this document from Alibaba Cloud-authorized channels.

4. This document serves only as a reference guide for your use of Alibaba Cloud products and services. Alibaba Cloud provides this document based on the "status quo", "being defective", and "existing functions" of its products and services. Alibaba Cloud makes every effort to provide relevant operational guidance based on existing technologies. However, Alibaba Cloud hereby makes a clear statement that it in no way guarantees the accuracy, integrity, applicability, and reliability of the content of this document, either explicitly or implicitly. Alibaba Cloud shall not take legal responsibility for any errors or lost profits incurred by any organization, company, or individual arising from download, use, or trust in this document. Alibaba Cloud shall not, under any circumstances, take responsibility for any indirect, consequential, punitive, contingent, special, or punitive damages, including lost profits arising from the use or trust in this document (even if Alibaba Cloud has been notified of the possibility of such a loss).

5. By law, all the contents in Alibaba Cloud documents, including but not limited to pictures, architecture design, page layout, and text description, are intellectual property of Alibaba Cloud and/or its affiliates. This intellectual property includes, but is not limited to, trademark rights, patent rights, copyrights, and trade secrets. No part of this document shall be used, modified, reproduced, publicly transmitted, changed, disseminated, distributed, or published without the prior written consent of Alibaba Cloud and/or its affiliates. The names owned by Alibaba Cloud shall not be used, published, or reproduced for marketing, advertising, promotion, or other purposes without the prior written consent of Alibaba Cloud. The names owned by Alibaba Cloud include, but are not limited to, "Alibaba Cloud", "Aliyun", "HiChina", and other brands of Alibaba Cloud and/or its affiliates, which appear separately or in combination, as well as the auxiliary signs and patterns of the preceding brands, or anything similar to the company names, trade names, trademarks, product or service names, domain names, patterns, logos, marks, signs, or special descriptions that third parties identify as Alibaba Cloud and/or its affiliates.

6. Please directly contact Alibaba Cloud for any errors of this document.

Document conventions

Style	Description	Example
Danger	A danger notice indicates a situation that will cause major system changes, faults, physical injuries, and other adverse results.	Danger: Resetting will result in the loss of user configuration data.
Warning	A warning notice indicates a situation that may cause major system changes, faults, physical injuries, and other adverse results.	Warning: Restarting will cause business interruption. About 10 minutes are required to restart an instance.
Notice	A caution notice indicates warning information, supplementary instructions, and other content that the user must understand.	Notice: If the weight is set to 0, the server no longer receives new requests.
Note	A note indicates supplemental instructions, best practices, tips, and other content.	Note: You can use Ctrl + A to select all files.
>	Closing angle brackets are used to indicate a multi-level menu cascade.	Click **Settings**> **Network**> **Set network type**.
Bold	Bold formatting is used for buttons , menus, page names, and other UI elements.	Click **OK**.
`Courier font`	Courier font is used for commands	Run the `cd /d C:/window` command to enter the Windows system folder.
Italic	Italic formatting is used for parameters and variables.	`bae log list --instanceid` *Instance_ID*
[] or [a|b]	This format is used for an optional value, where only one item can be selected.	`ipconfig [-all|-t]`
{} or {a|b}	This format is used for a required value, where only one item can be selected.	`switch {active|stand}`

Table of Contents

1.Scenario comparison between Message Queue for MQTT and Message Queue for Apache RocketMQ

This topic describes the association and differences between Message Queue for MQTT and traditional message-oriented middleware (MOM) such as

Message Queue for Apache RocketMQ

, and provides model selection recommendations in actual scenarios.

Background

Traditional MOM, such as

Message Queue for Apache RocketMQ

and Message Queue for Apache Kafka, is intended for microservices and big data and implements message storage and forwarding. The message producer and consumer are applications on servers.

Traditional MOM is applicable to scenarios where applications are deployed on servers that have fixed technology stacks and language platforms. However, traditional MOM cannot deal with scenarios where massive multi-platform devices developed in multiple languages need access and service properties assume an important role during message production and consumption. Examples are the mobile Internet and Internet of things (IoT) scenarios.

Designed based on the single-responsibility principle, Message Queue for MQTT is a stateless gateway designed for the mobile Internet and IoT, focusing only on the access, management, and messaging of massive mobile devices. Message Queue for MQTT can interact with other Alibaba Cloud services such as

Message Queue for Apache RocketMQ

based on data interaction rules.

Based on such division of responsibilities, Message Queue for MQTT routes the messages sent by devices to the specified service based on the rules defined in Message Queue for MQTT. Cloud-side applications can still use traditional microservice development solutions and interact with devices by connecting to a cloud-side storage service. Message Queue for MQTT enables data interaction between cloud-side applications and devices.

Scenario comparison

A service scenario may include different types of application components, each of which assumes a different role. Therefore, when you select a message service, you must understand the association and differences between Message Queue for MQTT and traditional MOM and use them together properly. For example, component A uses Message Queue for MQTT to send and receive messages, whereas component B uses

Message Queue for Apache RocketMQ

for the same purpose.

The following table describes the differences between Message Queue for MQTT and traditional MOM based on scenarios.

Message Queue for Apache RocketMQ

is used as an example for comparison. Other message services, such as Message Queue for Apache Kafka and Alibaba Cloud Message Queue for AMQP, observe the same rules.

Scenario comparison

Service	Scenario
Message Queue for MQTT	Message Queue for MQTT is applicable to mobility scenarios with access by massive devices, each of which maintains a relatively small data volume. Therefore, Message Queue for MQTT can be used to process messages transmitted by a large number of online Message Queue for MQTT clients, each of which maintains a relatively small message volume. For example, many enterprises have tens of thousands of and even millions of devices.
Message Queue for Apache RocketMQ	Message Queue for Apache RocketMQ is a messaging engine designed for user services deployed on servers, which is used for decoupling, asynchronous notification, and load shifting between service components. It is applicable to scenarios where a relatively small number of servers must process massive messages and require high throughput. In general, only a few enterprises have more than 10,000 servers. Therefore, Message Queue for Apache RocketMQ can be used to assist servers with massive data processing and analysis.

Scenarios of using Message Queue for MQTT and Message Queue for Apache RocketMQ together

- Scenario 1

  In IoT scenarios, tens of thousands and even millions of sensors can use Message Queue for MQTT to upload data, and applications deployed on servers can use

  Message Queue for Apache RocketMQ

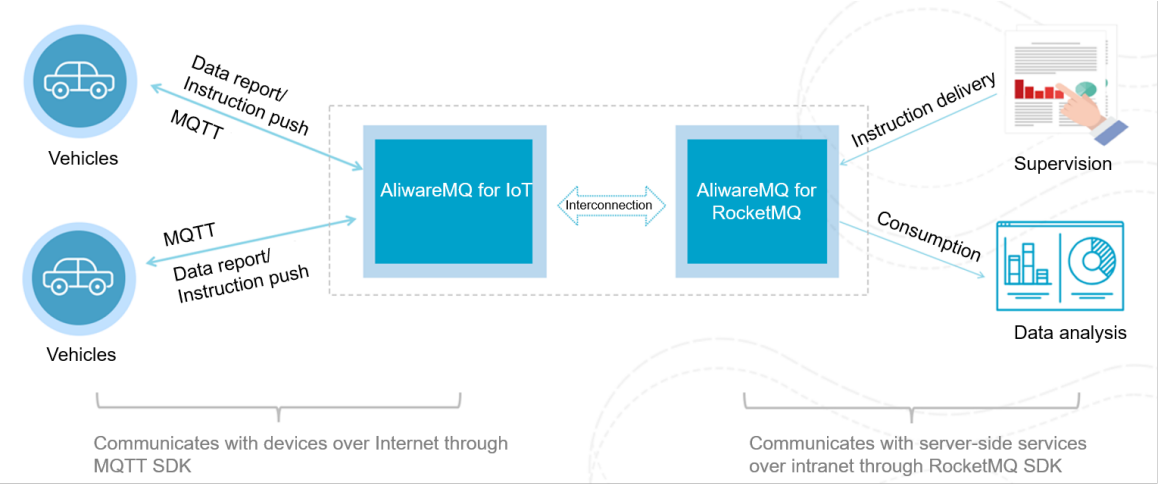
  to analyze and process the data.

- Scenario 2

  In Internet of Vehicles (IoV) scenarios, you may need to upload information about millions of vehicles to the cloud (servers), and the cloud delivers commands to a specific vehicle or broadcasts commands to all vehicles. Vehicles can connect to Message Queue for MQTT by using the Message Queue for MQTT SDK to upload data and receive commands. Monitoring and management systems (data analysis systems) can use the

  Message Queue for Apache RocketMQ

SDK to subscribe to messages and deliver commands, as shown in the following figure.



Based on the preceding differences, we recommend that you use Message Queue for MQTT for mobile devices and use

Message Queue for Apache RocketMQ

or other message services for applications deployed on servers.

Feature comparison

The following table compares the features of Message Queue for MQTT and

Message Queue for Apache RocketMQ

.

Feature comparison

Feature	Message Queue for MQTT	Message Queue for Apache RocketMQ
Client connections	Message Queue for MQTT supports message processing for massive clients, which reach millions or tens of millions in quantity.	Message Queue for Apache RocketMQ supports message processing for a relatively small number of servers, which is generally less than 10,000.
Message volume per client	Each Message Queue for MQTT client processes a small number of messages, and sends and receives messages at regular intervals.	Each Message Queue for Apache RocketMQ client processes a large number of messages and requires high throughput.
Deployment	Message Queue for MQTT can be used for mobile devices, apps, and HTML5 pages.	Message Queue for Apache RocketMQ can be used for server-side applications.

Feature	Message Queue for MQTT	Message Queue for Apache RocketMQ
Consumption mode	Message Queue for MQTT supports broadcasting consumption.	Message Queue for Apache RocketMQ supports clustering consumption and broadcasting consumption. For more information, see Clustering consumption and broadcasting consumption.
Sequence	Messages can be sent in order but cannot be received in order (which will be available in the future).	Messages can be sent and received in order.
Multi-language/system support (TCP)	Message Queue for MQTT supports many languages such as Java, C, C++, .NET, Python, JS, and Go, and systems such as Android and iOS.	Message Queue for Apache RocketMQ supports Java, C++, and .NET.
Access credential	Message Queue for MQTT supports granting permissions to Resource Access Management (RAM) users for access and using a Message Queue for MQTT token for temporary access. For more information, see Grant permissions to RAM users and Authentication overview.	Message Queue for Apache RocketMQ supports granting permissions to RAM users for access and granting permissions to another Alibaba Cloud account based on STS token for temporary access. For more information, see Grant permissions to RAM users and Grant permissions to another Alibaba Cloud account.

Model selection instructions

The following information describes the general principle of model selection:

- We recommend that you use

  Message Queue for Apache RocketMQ

  for applications that are deployed on servers.
- We recommend that you use Message Queue for MQTT for applications that are deployed on mobile devices, apps, or web pages.

For common scenarios, we provide the recommendations of using Message Queue for MQTT and

Message Queue for Apache RocketMQ

, as described in the following table.

Model selection recommendation

Scenario	Deployment	Message Queue for MQTT	Message Queue for Apache RocketMQ
Status data reporting by devices	Mobile device	√	×
Receiving, processing, and analysis of device-reported data	Mobile device	×	√
Delivery of control commands to multiple devices	Server	×	√
Live streaming, bullet screen, and chat apps that require messaging	App	√	×
Receiving and analysis of chat messages on servers	Server	×	√

Note

√ indicates that we recommend that you use the corresponding service, whereas × indicates that we recommend that you do not use the corresponding service.

2.Message structure mappings between Message Queue for MQTT and Message Queue for Apache RocketMQ

This topic describes mappings between the message structures and properties involved in interaction with Message Queue for MQTT by using the

Message Queue for Apache RocketMQ

SDK, helping you better understand and use Message Queue for MQTT and Message Queue for Apache RocketMQ.

Message Queue for MQTT is a gateway service that is intended for mobile devices. Message Queue for MQTT can interact with other Alibaba Cloud services such as

Message Queue for Apache RocketMQ

based on data interaction rules. For more information, see Manage rules.

If you use Message Queue for MQTT independently, ignore the mappings described in this topic and observe the Message Queuing Telemetry Transport (MQTT) protocol.

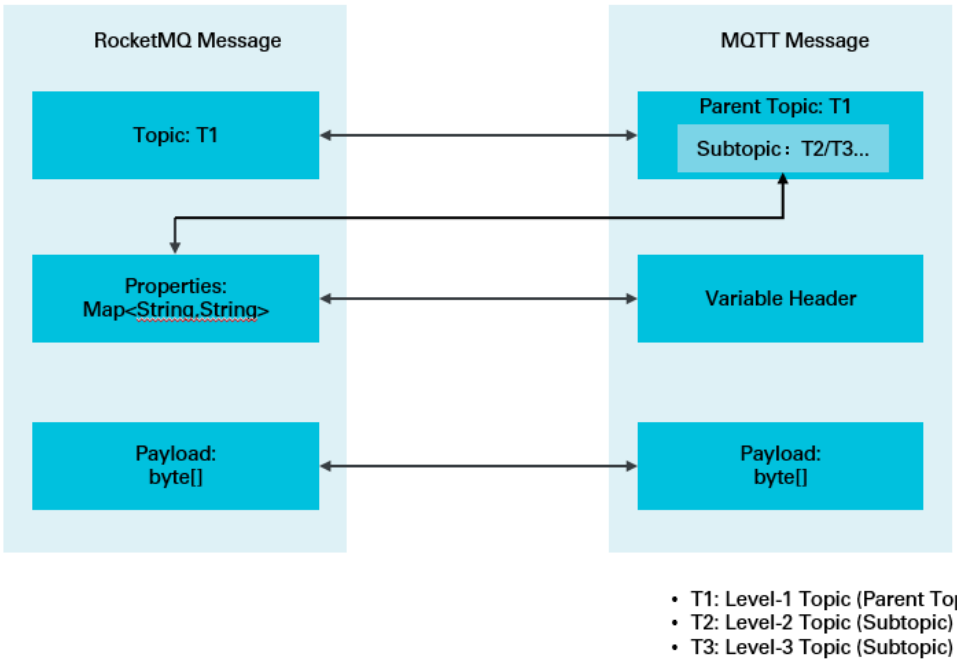
For more information about Message Queue for MQTT, see What is Message Queue for MQTT? and Terms.

Message structure mappings

Message Queue for MQTT and

Message Queue for Apache RocketMQ

are messaging systems that are based on the publish-subscribe model and have similar concepts. The following figure shows the differences in the major concepts and mappings between them.



Message Queue for MQTT supports multi-level topics, whereas

Message Queue for Apache RocketMQ

supports one-level topics, as shown in the preceding figure. Therefore, a level-1 topic in Message Queue for MQTT is mapped to a topic in

Message Queue for Apache RocketMQ

, and level-2 and level-3 topics in Message Queue for MQTT are mapped to the message properties in

Message Queue for Apache RocketMQ

.

The

Message Queue for Apache RocketMQ

protocol supports messages with custom properties, whereas the MQTT protocol does not support properties. However, part of the information in Message Queue for MQTT is mapped to message properties in

Message Queue for Apache RocketMQ

. This facilitates tracing of the headers and device information in the MQTT protocol and allows users of the

Message Queue for Apache RocketMQ

SDK to retrieve such information.

> **Note** For information about how to configure mappings from properties in Message Queue for Apache RocketMQ to parameters in Message Queue for MQTT, see the table in Property mappings.

Message Queue for Apache RocketMQ and Message Queue for MQTT use the data serialization results of your service messages as the payload.

Message Queue for Apache RocketMQ and Message Queue for MQTT do not further encode and decode the service messages.

Property mappings

The following table lists the property mappings supported between Message Queue for MQTT and Message Queue for Apache RocketMQ. You can set or retrieve information by reading and writing these properties during interaction between applications that use the Message Queue for Apache RocketMQ and Message Queue for MQTT SDKs.

For more information about QoS, cleanSession, topics, and client IDs, see Terms.

Message Queue for MQTT parameter	Message Queue for Apache RocketMQ property key	Valid property value	Description
QoS	qoslevel	0, 1, and 2	This property can be set when Message Queue for Apache RocketMQ sends messages to Message Queue for MQTT. If it is not set, the default value 1 is used. Message Queue for Apache RocketMQ can directly read the QoS parameter from the messages that are sent from Message Queue for MQTT.

Message Queue for MQTT parameter	Message Queue for Apache RocketMQ property key	Valid property value	Description
cleanSession	cleansessionflag	true and false	This property can be set when Message Queue for Apache RocketMQ sends P2P messages to Message Queue for MQTT clients. If it is not set, the default value true is used. This property cannot be set for other message types. Message Queue for Apache RocketMQ can directly read the cleanSession parameter from the messages that are sent from Message Queue for MQTT.
Subtopic	mqttSecondTopic	A string that indicates a specific subtopic	This property can be set when a subtopic is required to filter the messages that Message Queue for Apache RocketMQ sends to Message Queue for MQTT clients. If it is not set, the default value null is used. Message Queue for Apache RocketMQ can directly read the subtopic from the messages that are sent from Message Queue for MQTT.

Message Queue for MQTT parameter	Message Queue for Apache RocketMQ property key	Valid property value	Description
Topic in messages received on a client	mqttRealTopic	The sub-level string that services expect message-receiving clients to display	This property can be set when a Message Queue for MQTT client is expected to display the specified subtopic name after it receive messages from Message Queue for Apache RocketMQ . This property is typically applied to P2P messages. If it is not set, P2P messages use a fixed topic name by default. The messages that Message Queue for MQTT sends to Message Queue for Apache RocketMQ do not contain the corresponding parameter.
clientId	clientId	A string that indicates a specific client ID	This property cannot be set. When a Message Queue for MQTT client sends a message to Message Queue for Apache RocketMQ , the clientId parameter is used to trace the ID of the client.