# Alibaba Cloud MQTT

## Use Case

Issue: 20190914

# 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 due to product version upgrades , adjustments, or other reasons. Alibaba Cloud reserves the right to modify the content of this document without notice and the updated versions of this document will be occasionally released through Alibaba Cloud-authorized channels. You shall 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 the document in the context that Alibaba Cloud products and services are provided on an "as is", "with all faults " and "as available" basis. 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 bear any liability for any errors or financial losses incurred by any organizations, companies, or individuals arising from their download, use, or trust in this document. Alibaba Cloud shall not, under any circumstances, bear responsibility for any indirect, consequential, exemplary, incidental, 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 content of the Alibaba Cloud website, including but not limited to works, products, images, archives, information, materials, website architecture, website graphic layout, and webpage design, 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 the Alibaba Cloud website, product programs, or content 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 contact Alibaba Cloud directly if you discover any errors in this document.

# Generic conventions

Table -1: Style conventions

| Style | Description | Example |
| --- | --- | --- |
|  | This warning information indicates a situation that will cause major system changes, faults, physical injuries, and other adverse results. |  Danger:<br>Resetting will result in the loss of user configuration data. |
|  | This warning information indicates a situation that may cause major system changes, faults, physical injuries, and other adverse results. |  Warning:<br>Restarting will cause business interruption. About 10 minutes are required to restore business. |
|  | This indicates warning information, supplementary instructions, and other content that the user must understand. |  Notice:<br>Take the necessary precautions to save exported data containing sensitive information. |
| | This indicates supplemental instructions, best practices, tips, and other content that is good to know for the user. |  Note:<br>You can use Ctrl + A to select all files. |
| > | Multi-level menu cascade. | Settings > Network > Set network type |
| Bold | It is used for buttons, menus , page names, and other UI elements. | Click **OK**. |
| `Courier font` | It is used for commands. | Run the `cd / d C :/ windows` command to enter the Windows system folder. |
| *Italics* | It is used for parameters and variables. | `bae log list --instanceid` *Instance_ID* |
| [] or [a\|b] | It indicates that it is a optional value, and only one item can be selected. | `ipconfig` *[-all\|-t]* |

| Style | Description | Example |
| --- | --- | --- |
| {} or {a\|b} | **It indicates that it is a required value, and only one item can be selected.** | `swich {stand \| slave}` |

# Contents

# 1 Audio/video communication solution

The audio/video communication solution is co-developed by Alibaba Cloud Message Queue for MQTT and Real-Time Communication (Alibaba Cloud RTC). It supports the rapid construction of products for a variety of real-time communication scenarios, such as online audio and video conferencing and one-to-one voice call applications. This topic describes the system architecture, data flow design, and related matters for attention in detail.

## Terms

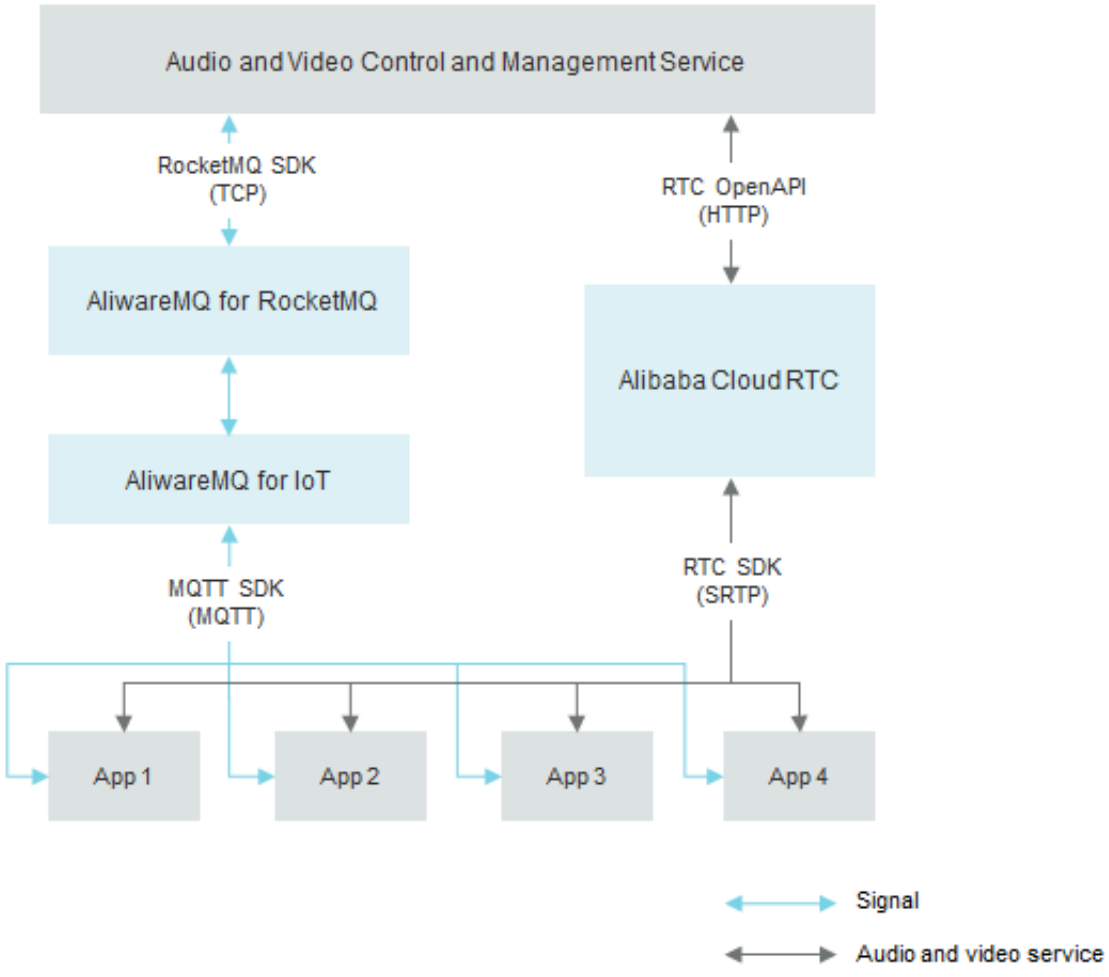
| | |
|---|---|
| MQTT | MQTT is a standard industry protocol for IoT and mobile Internet that is suitable for data transmission between mobile terminals.Message Queue for MQTT supports this protocol, by default. |
| MQTT broker | Message Queue for MQTT provides the MQTT broker that is used to interact with the MQTT protocol, and is used with the MQTT client and MQ to send and receive messages. |
| MQTT client | The MQTT client is the node that is used to interact with the MQTT broker. In this solution, it specifically indicates the audio/video mobile application that sends or receives audio/video calling requests. |
| P2P message | Message Queue for MQTT provides P2P messages based on the standard MQTT protocol. These messages are special messages because they can be directly sent to a specific target MQTT client without going through regular subscription relationship matching. For details, see #unique_4. |
| Real-time communication | Real-time communication is a network communication method that is mainly used for the audio and video fields. Currently, the mainstream application scenarios include audio calling, video calling, and video conferencing. |
| RTC server | This server hosts the audio/video and related media channel services provided by Alibaba Cloud RTC. |
| Control and management | These servers are the control nodes in the audio/video communication system, which are referred to as audio/video management services in |

| | |
|---|---|
| server of the audio/video service | this topic. The audio/video management service is self-constructed, and is used to control the life cycle of all audio/video communication sessions. The management node is normally deployed on the cloud, and is constructed using Alibaba Cloud products. |
| Audio/video mobile application | This is the application on the terminal that is used by the end user in the audio/video communication system, which is referred to as the terminal application in this topic. The user uses this application to initiate or participate in audio/video calls. |

## Solution architecture

Figure 1-1: Solution architecture shows the architecture of the audio/video communication solution.

Figure 1-1: Solution architecture



As shown in Figure 1-1: Solution architecture, the audio/video management service and the terminal application signal by using the AliwareMQ for MQ, and implement service data interactions through Alibaba Cloud RTC. For more information, see Data interaction.

### Advantages

The advantages of the audio/video communication solution are as follows:

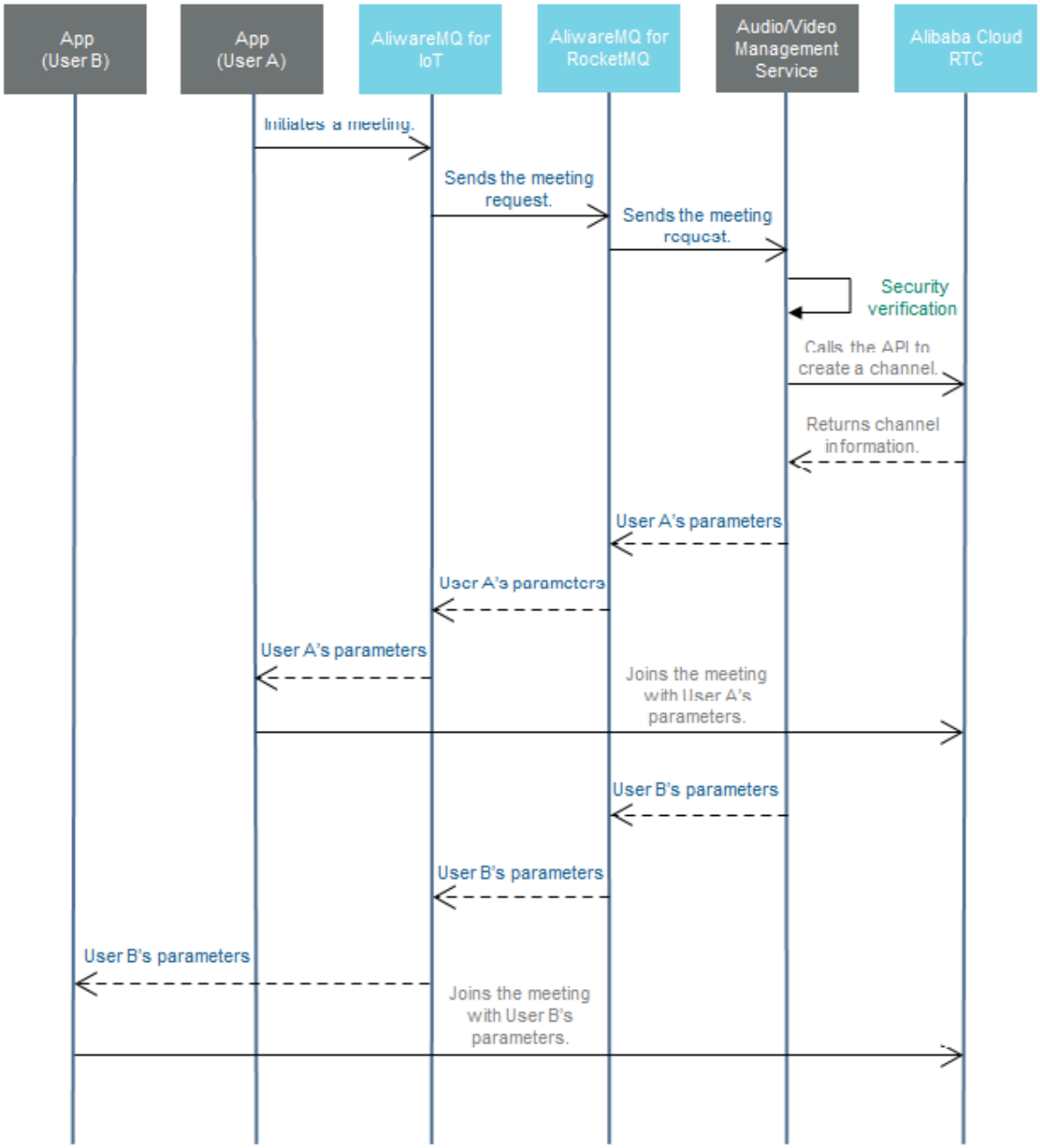
- Service capabilities are scalable.
  - Message Queue for MQTT and Alibaba Cloud RTC can be used on demand and dynamically scaled up to handle burst traffic peaks.
- Network coverage is widespread.
  - Message Queue for MQTT and Alibaba Cloud RTC provide global deployment capabilities to achieve local service access and save cross-zone and cross-nation costs.
- The construction period is short, supporting easy access.
  - The construction process is O&M-free, reducing labor and hardware costs.
  - The API is easy to use, supporting rapid implementation.
- Security and reliability are high.
  - All service nodes are highly available and stable.
  - Message Queue for MQTT supports SSL/TLS encryption and media streams support SRTP protection.

### Data interaction

Figure 1-2: Data flow shows the process of a real-time conference call based on Message Queue for MQTT and Alibaba Cloud RTC. In this figure, the gray parts represent the self-built development programs or services, and the blue parts

represent the services provided by Message Queue for MQTT, MQ, and Alibaba Cloud RTC.

Figure 1-2: Data flow




As shown in Figure 1-2: Data flow, User A invites User B to join an audio/video conference. The specific process is as follows:

1. User A of the terminal application initiates a meeting request and sends the request to the MQTT broker by sending an MQTT message. The message is routed through Message Queue for MQTT to the MQ queue. The audio/video management

service developed by the business side processes the meeting request by receiving the message. After verification, it calls the Alibaba Cloud RTC API to register the relevant resources and parameters of this communication.

2. After receiving the parameters, the audio/video management service encapsulates the parameters into an invitation message and sends it to MQ. After the message is routed through Message Queue for MQTT, it is delivered to the terminal application of User A. Then, the terminal application of User A is added to the conference channel based on the parameters, and the access operation is completed.
3. The audio/video management service also needs to find User B's information based on the information in User A's invitation. Similarly, the service also encapsulates the parameters into an invitation message, and the transfer process for User B is the same as that for User A described in Step 2.
4. The conference member User B joins the conference after receiving the parameters, and the communication initialization is completed.

Based on the preceding outline of the process, you can use Message Queue for MQTT messages to perform other custom processes and operations, such as ending conferences, inviting others to join an ongoing conference, and muting certain users.Message Queue for MQTT plays the signaling role in audio/video conferencing scenarios.

### Considerations

The preceding process describes how to use Message Queue for MQTT and Alibaba Cloud RTC to quickly build your own real-time communication application. For more information about the SDK, see the AliwareMQ for IoT and AliwareMQ for MQ documents.

When using Message Queue for MQTT to construct a signaling channel for real-time communication, follow these principles for message type and parameter design:

- Client ID mapping

  The MQTT protocol requires that each client has a globally unique client ID. The client ID consists of two parts concatenated with the "@@@" separator. The final

client ID must be unique and its length cannot exceed 64 characters. The two parts of the client ID are described as follows:

- Prefix group ID: Apply for the group ID on the Message Queue for MQTT console. You are advised to classify the group IDs by application platform or channel to facilitate troubleshooting. For example, Android and iOS clients can be divided into different group IDs, or clients of different versions use different group IDs.
- Suffix device ID: The device ID is generated by the application. Device IDs can be mapped to application account IDs to ensure their global uniqueness.

For more information about client IDs, see #unique_5.

· Topic name mapping

To use Message Queue for MQTT, you need to understand the MQTT subscription model. For more information, see the protocol documentation and official documentation.

MQTT is a message protocol that follows the publish/subscribe model. The subscription relationship and topic follow the directory tree format. Topics can be divided into parent topics and subtopics. The total length of a topic (including parent topics and subtopics) cannot exceed 64 characters. The types of topics are described as follows:

- Parent topic: The topic at the first level of the directory tree is a parent topic. The parent topic must be applied for on the Message Queue for MQTT console

before it can be used. After successful application, the parent topic is equivalent to a namespace.

- Subtopic: The parts of the topic subsequent to the first-level topic of the directory tree are referred to as the subtopics. You do not need to apply for a subtopic, and you can specify a subtopic as needed.

For more information about topics, see #unique_5.

When designing a topic for sending and receiving messages, the business side must follow these principles:

- Different parent topics are used for upstream messages (messages sent from the terminal application to the management service) and downstream messages ( messages sent from the management service to the terminal application).
- Messages with different priorities or large size differences use different parent topics.

For the interaction process described above, you are advised to use P2P messaging provided by Message Queue for MQTT. P2P messages do not need to be subscribed, allowing the producer to directly specify the peer consumer to receive them. For more information, see #unique_4.

- Parameter design for message sending and receiving

  Mobile applications may be killed in the background, making the mobile application offline. To handle this situation, you are advised to configure the terminal application as follows to ensure that the terminal application receives the previous message after it goes back online:

  - Set the `CleanSessi on` parameter to "false".
  - Set QoS to "1".

  The terminal application should perform deduplication and timeliness verification on the received messages (this is applicable if the terminal application remains offline for more than one day and then receives the messages from the previous day when it goes online again).

  For more information about `CleanSessi on` and QoS, see #unique_5.

# 2 New retail digital price tag solution

The new retail digital price tag solution is launched by Alibaba Cloud Message Queue for MQTT. It uses MQTT to manage data updates for digital price tags and multimedia screens in malls, supermarkets, and other public places. This topic takes digital price tags as an example to describe the system architecture, data flow design, and other key components of the solution in detail. Other similar industries can refer to this solution, making modifications as necessary for implementation.

Terms

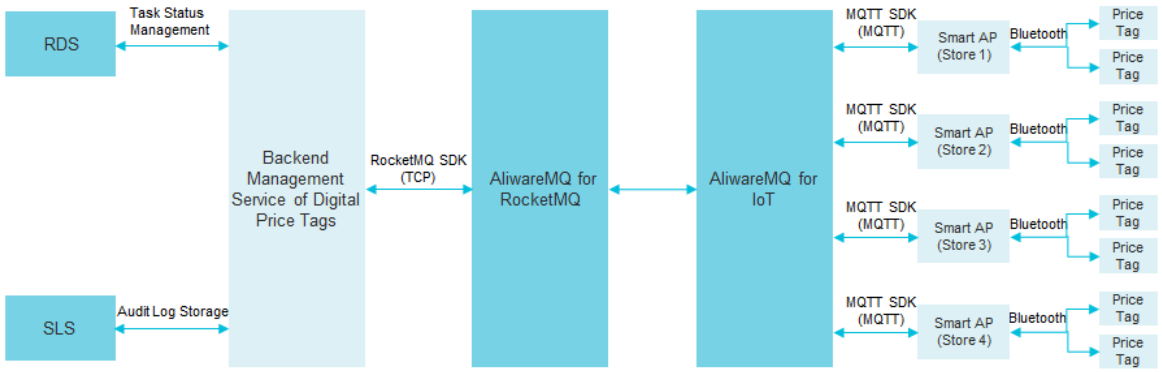
| | |
|---|---|
| MQTT | MQTT is a standard industry protocol for IoT and mobile Internet that is suitable for data transmission between mobile terminals.Message Queue for MQTT supports this protocol, by default. |
| MQTT broker | Message Queue for MQTT provides the MQTT broker that is used to interact with the MQTT protocol, and is used to receive and send messages. |
| MQTT client | The MQTT client is the node used to interact with the MQTT broker. In this solution, it specifically indicates a smart access point for sending or receiving price change messages. |
| P2P message | Message Queue for MQTT provides P2P messages based on the standard MQTT protocol. These messages are special messages because they can be directly sent to a specific target MQTT client without going through regular subscription relationship matching. For details, see #unique_4. |
| Smart AP | Smart access points (APs) are the smart routers and other network devices commonly seen on the market, which support application programming and can simultaneously handle Internet access and LAN device control. |
| Digital price tag | Digital price tags are distributed to digital displays in malls, supermarkets, and other places. Generally, they use wireless sensor network protocols such as Bluetooth and ZigBee as well as smart AP nodes for networking. |

| | |
|---|---|
| Digital price tag management service | In the digital price tag system, this backend service is used to manage the content displayed on the digital displays. This service primarily handles tasks that would otherwise be performed manually, like price changes. |
| ApsaraDB for RDS | This is a highly available and scalable online database service provided by Alibaba Cloud. It is used in the digital price tag system to preserve status changes for tasks such as price changes. |
| Log Service (SLS) | Alibaba Cloud's log storage service is used in the digital price tag system to persistently store all operation logs for auditing and tracing purposes. |

## Solution architecture

In the digital price tag solution, Message Queue for MQTT is used in conjunction with multiple Alibaba Cloud products to implement update management for price tag data.Figure 2-1: Solution architecture shows details on how the solution architecture of the digital price tag system works.

Figure 2-1: Solution architecture




As shown in Figure 2-1: Solution architecture, the digital price tag system primarily includes price tag nodes, smart AP nodes, Message Queue for MQTT, MQ, the backend control service for digital price tags, ApsaraDB for RDS, and Log Service. Each of these components is described as follows:

- The smart AP forwards the status data of the price tag and receives price change commands. The smart AP uses the MQTT SDK to access Alibaba Cloud Message Queue for MQTT over the public network based on the distribution of the stores or locations. This link uses SSL/TLS to encrypt transmissions, preventing data leaks.

- One smart AP downlink and several price tag nodes can communicate with each other through a wireless sensor network such as Bluetooth or ZigBee.
- The backend management service for digital price tags is deployed on the cloud through ECS. The MQ SDK is used to interact with MQ. The MQTT broker and the MQ broker are natively interconnected.
- The backend management service for digital price tags can persistently change the status in the RDS databases when price changes and other tasks are performed . It can store price tag report data and operations logs to Log Service to facilitate tracing and auditing.

### Advantages

The advantages of the new retail digital price tag solution are as follows:

- Powerful service capabilities that can be scaled automatically.
  - Message Queue for MQTT The message transmission capability is infinitely scalable, allowing you to increase the number of smart terminals without suffering from compromised system capabilities.
  - Message Queue for MQTT Information pushing in milliseconds is supported for millions of devices, with an even smaller latency for display updates of digital price tags.
- Extensive application range, versatile generability, and rapid duplication.
  - Based on the MQTT standard protocol, this solution is universally applicable. It can be replicated in similar scenarios by adapting the solution for different data content.
- Security and reliability are high.
  - Message Queue for MQTT and smart AP nodes supports SSL/TLS encryption in data transmission, preventing media business data leakage.
  - All service nodes are highly available and stable.

### Data interaction

Status reporting

1. The digital price tag node uses a periodic polling mechanism to exchange data with the smart AP node, reporting its current display status, power capacity, and other information.
2. The smart AP node organizes data and sends MQTT messages to the MQTT broker.

3. The MQTT broker writes the reported message to the MQ topic specified by the business side.
4. The digital price tag management service uses the received MQ messages from the queue to process and analyze the status of price tag nodes that are online in the current system. Then, it writes the data to Log Service.

Update displays

1. The digital price tag management service triggers price change operations by sending MQ messages.
2. The MQTT broker routes the MQ message, pushing the message to the target smart AP node by using the MQTT protocol.
3. The smart AP node receives the price change notification and temporarily saves the task.
4. The digital price tag node uses the polling mechanism to exchange data with the smart AP node and receives the new content to be displayed.
5. After the target digital price tag node changes the price, the smart AP node returns an MQTT message to notify the digital price tag management service that the current task has been completed.
6. The digital price tag management service writes the execution log of the current task to Service Log, facilitating subsequent tracing queries.

## Considerations

The preceding procedure describes how to use Message Queue for MQTT and MQ to build a digital price tag system. For more information about the SDK, see AliwareMQ for IoT and AliwareMQ for MQ documents.

When using Message Queue for MQTT and MQ to send commands, follow these principles for message type design and parameter design:

- SDK and protocol selection

  In the digital price tag scenario, one application may be used by hundreds of offline stores. Generally, each store is equipped with several smart AP nodes. The number of smart AP nodes can be increased as the business scales out. This makes smart AP nodes suitable for access by using the MQTT protocol. The digital price tag management service is deployed on the cloud, making the use of cloud-based MQ suitable for access.

- Client ID mapping

  The MQTT protocol requires that each client has a globally unique client ID. The client ID consists of two parts concatenated with the "@@@" separator. The final client ID must be unique and its length cannot exceed 64 characters. The two parts of the client ID are described as follows:

  - Prefix group ID: Apply for the group ID on the Message Queue for MQTT console. Group IDs can be roughly classified by platform vendor or channel to facilitate troubleshooting. For example, different industries or batches can be divided into different group IDs, or clients of different versions can use different group IDs.
  - Suffix device ID: The device ID is generated by the application. Device IDs can be encoded by using the unique information, such as the MAC address of the smart AP node.

  For more information about client IDs, see #unique_5.

- Topic name mapping

  To use Message Queue for MQTT, you need to understand the MQTT subscription model. For more information, see the protocol documentation and official documentation.

  MQTT is a message protocol that follows the publishing/subscription model. The subscription relationship and topic follow the directory tree format. Topics can be divided into parent topics and subtopics. The total length of a topic (including parent topics and subtopics) cannot exceed 64 characters. The types of topics are described as follows:

  - Parent topic: The topic at the first level of the directory tree is a parent topic. The parent topic must be applied for on the Message Queue for MQTT console

before it can be used. After successful application, the parent topic is equivalent to a namespace.

- Sub topic: The parts of the topic subsequent to the first-level topic of the directory tree are referred to as the subtopics. You do not need to apply for a subtopic, and you can specify a subtopic as needed.

For more information about topics, see #unique_5.

When designing a topic for sending and receiving messages, the business side must follow these principles:

- Different types of tasks use different parent topics. For example, in this scenario , the price change tasks and the terminal status reporting tasks use different parent topics.
- In the digital price tag system, we recommend that you use P2P messaging provided by Message Queue for MQTT for the interactive messages of price change tasks. P2P messages do not need to be subscribed, allowing the sender to directly specify the peer to receive them. For more information, see #unique_4.

· Parameter design for message sending and receiving

Generally, price change tasks in the digital price tag scenario require real-time pushing. Therefore, we recommend that you configure the smart AP as follows during the interactions between the smart AP and the MQTT broker to ensure that the smart AP does not need to process the tasks that were pushed when it was disconnected:

- Set the `CleanSessi on` parameter to "true".
- Set QoS to "1".

The smart AP should perform deduplication and timeliness verification on received messages.

For more information about `CleanSessi on` and QoS, see #unique_5.