阿里云
微消息队列MQTT

场景化解决方案

文档版本：20190913

法律声明

阿里云提醒您在阅读或使用本文档之前仔细阅读、充分理解本法律声明各条款的内容。如果您阅读或使用本文档，您的阅读或使用行为将被视为对本声明全部内容的认可。

1. 您应当通过阿里云网站或阿里云提供的其他授权通道下载、获取本文档，且仅能用于自身的合法合规的业务活动。本文档的内容视为阿里云的保密信息，您应当严格遵守保密义务；未经阿里云事先书面同意，您不得向任何第三方披露本手册内容或提供给任何第三方使用。
2. 未经阿里云事先书面许可，任何单位、公司或个人不得擅自摘抄、翻译、复制本文档内容的部分或全部，不得以任何方式或途径进行传播和宣传。
3. 由于产品版本升级、调整或其他原因，本文档内容有可能变更。阿里云保留在没有任何通知或者提示下对本文档的内容进行修改的权利，并在阿里云授权通道中不时发布更新后的用户文档。您应当实时关注用户文档的版本变更并通过阿里云授权渠道下载、获取最新版的用户文档。
4. 本文档仅作为用户使用阿里云产品及服务的参考性指引，阿里云以产品及服务的”现状“、“有缺陷”和“当前功能”的状态提供本文档。阿里云在现有技术的基础上尽最大努力提供相应的介绍及操作指引，但阿里云在此明确声明对本文档内容的准确性、完整性、适用性、可靠性等不作任何明示或暗示的保证。任何单位、公司或个人因为下载、使用或信赖本文档而发生任何差错或经济损失的，阿里云不承担任何法律责任。在任何情况下，阿里云均不对任何间接性、后果性、惩戒性、偶然性、特殊性或刑罚性的损害，包括用户使用或信赖本文档而遭受的利润损失，承担责任（即使阿里云已被告知该等损失的可能性）。
5. 阿里云网站上所有内容，包括但不限于著作、产品、图片、档案、资讯、资料、网站架构、网站画面的安排、网页设计，均由阿里云和/或其关联公司依法拥有其知识产权，包括但不限于商标权、专利权、著作权、商业秘密等。非经阿里云和/或其关联公司书面同意，任何人不得擅自使用、修改、复制、公开传播、改变、散布、发行或公开发表阿里云网站、产品程序或内容。此外，未经阿里云事先书面同意，任何人不得为了任何营销、广告、促销或其他目的使用、公布或复制阿里云的名称（包括但不限于单独为或以组合形式包含”阿里云”、Aliyun”、“万网”等阿里云和/或其关联公司品牌，上述品牌的附属标志及图案或任何类似公司名称、商号、商标、产品或服务名称、域名、图案标示、标志、标识或通过特定描述使第三方能够识别阿里云和/或其关联公司）。
6. 如若发现本文档存在任何错误，请与阿里云取得直接联系。

通用约定

格式	说明	样例
	该类警示信息将导致系统重大变更甚至故障，或者导致人身伤害等结果。	**禁止:** 重置操作将丢失用户配置数据。
	该类警示信息可能导致系统重大变更甚至故障，或者导致人身伤害等结果。	**警告:** 重启操作将导致业务中断，恢复业务所需时间约10分钟。
	用于补充说明、最佳实践、窍门等，不是用户必须了解的内容。	**说明:** 您也可以通过按Ctrl + A选中全部文件。
>	多级菜单递进。	设置 > 网络 > 设置网络类型
粗体	表示按键、菜单、页面名称等UI元素。	单击 确定。
`courier` 字体	命令。	执行 `cd /d C:/windows` 命令，进入Windows系统文件夹。
*## *	表示参数、变量。	`bae log list --instanceid` *`Instance_ID`*
[]或者[a|b]	表示可选项，至多选择一个。	`ipconfig` *`[-all|-t]`*
{}或者{a|b}	表示必选项，至多选择一个。	`swich` *`{stand | slave}`*

目录

法律声明.....I
通用约定.....I
1 音视频通信解决方案（MQTT）..... 1
2 新零售电子价签解决方案（MQTT）..... 6
3 云存储网关利用 MQTT 服务实现统一管控.....11

1 音视频通信解决方案（MQTT）

音视频通信解决方案是由阿里云微消息队列 MQTT 和音视频通信 RTC 联合推出的有助于快速搭建各种实时通信场景产品，譬如在线音视频会议、1 对 1 语音通话应用的解决方案。本文将详细描述该解决方案的系统架构、数据流设计以及相关注意事项。

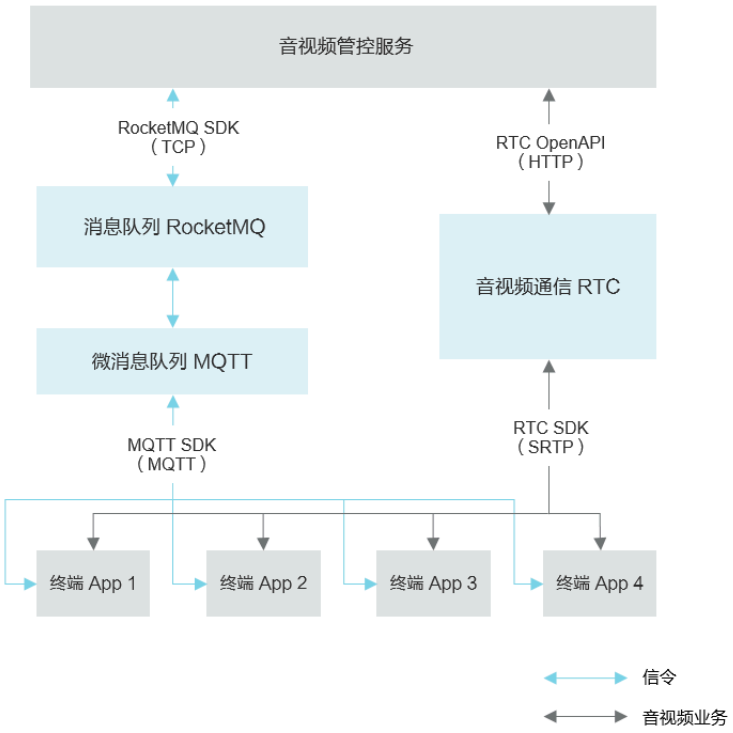
名词解释

MQTT	一种物联网和移动互联网领域的行业标准协议，适合移动终端之间的数据传输。微消息队列 MQTT 默认支持该协议。
MQTT 服务器	微消息队列 MQTT 提供的 MQTT 协议交互的服务端节点，用于完成与 MQTT 客户端和消息队列 MQ 各自的消息收发。
MQTT 客户端	用于和 MQTT 服务器交互的移动端节点，本方案中特指发送或接收音视频通话请求的音视频移动端应用。
P2P 消息	微消息队列 MQTT 在标准的 MQTT 协议基础上提供的一种特殊消息，该类型消息无需普通的订阅关系匹配，便可直接发送给指定的单个目标 MQTT 客户端。详情请参见 #unique_4。
RTC	实时通信，一种主要面向语音、视频领域的网络通信方式。目前比较主流的应用场景是语音通话、视频通话、视频会议等。
RTC 服务器	阿里云音视频通信 RTC 提供的音视频相关媒体通道服务。
音视频业务管控服务器	音视频通信系统中的管控节点（下文简称音视频管控服务）。音视频管控服务需要由业务方自行建设，用于控制所有音视频通信会话的生命周期。该管控节点一般部署在云端，使用阿里云的基础产品搭建。
音视频移动端应用	音视频通信系统中最终用户持有的终端 App（下文简称终端 App）。用户使用该 App 发起或者参与音视频通话。

方案架构

图 1-1: 方案架构 展示了音视频通信解决方案的架构。

图 1-1: 方案架构



如图 1-1: 方案架构 所示，音视频管控服务和终端 App 之间通过阿里云消息队列产品完成信令传输，通过阿里云音视频通信 RTC 产品完成业务数据交互。详情可参见下文中的数据交互部分。

方案优势

音视频通信解决方案的优势如下所述：

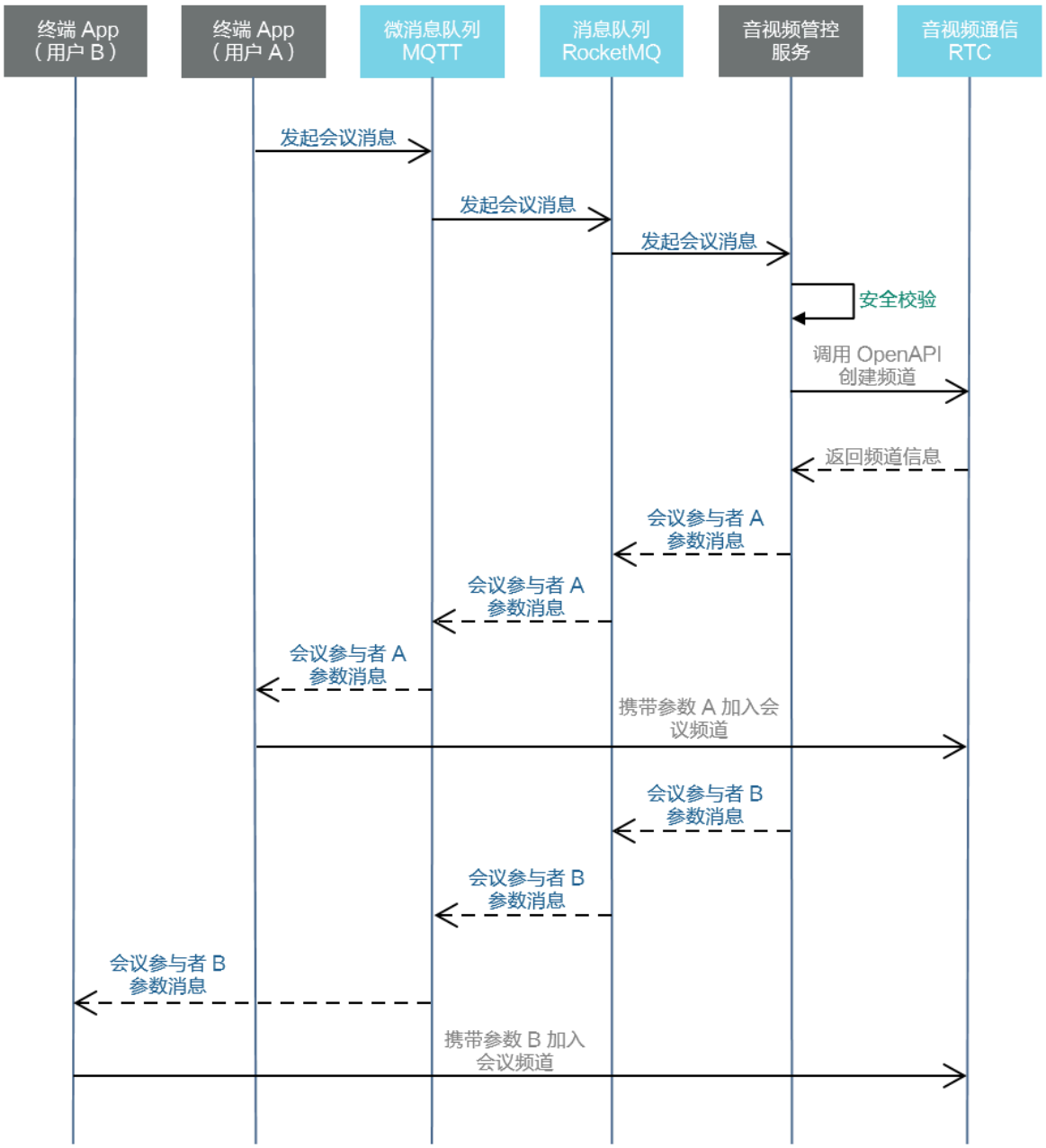
- 服务能力弹性扩缩
  - 微消息队列 MQTT 和音视频通信 RTC 服务都能实现按需使用，动态扩缩，轻松应对突发流量高峰。
- 网络覆盖范围广
  - 微消息队列 MQTT 和音视频通信 RTC 都提供全球部署，实现服务就近接入，避免自建跨区跨国的成本。
- 建设周期短，接入简单
  - 无运维建设过程，人力和硬件成本降低。
  - API 简单易用，快速上手。

- 安全可靠
  - 所有服务节点高可用，稳定性高。
  - 微消息队列 MQTT 支持 SSL/TLS 加密，媒体流支持 SRTP 保护。

数据交互

图 1-2: 数据流 所展示的是基于微消息队列 MQTT 和音视频通信 RTC 实现一次音视频通话会议的调用流程，其中灰色部分为您的自建开发程序或服务，蓝色部分是微消息队列 MQTT、消息队列 MQ 以及音视频通信 RTC 所提供的服务。

图 1-2: 数据流



如图 1-2: 数据流 所示，该场景中用户 A 将邀请用户 B 加入音视频会议，具体流程如下所述：

1. 终端 App 的某个用户 A 发起会议请求，通过发送 MQTT 消息将请求传递到 MQTT 服务端，消息经过微消息队列 MQTT 路由到消息队列 MQ，业务方自行开发的音视频管控服务通过接收消息处理该会议请求，验证通过后调用音视频通信 RTC 的 API 注册本次通信的相关资源和参数。
2. 音视频管控服务收到参数后将参数封装成邀请入会的消息，发送到消息队列 MQ，经过微消息队列 MQTT 路由后消息会投递给发起者用户 A 使用的终端 App，用户 A 的终端 App 根据参数加入会议频道完成入会操作。
3. 音视频管控服务还需要根据用户 A 邀请的信息找到用户 B 信息，同样封装一条邀请入会的消息，传递流程同步骤 2。
4. 会议参与者用户 B 收到参数后加入会议，本次通信初始化完成。

基于上述设计思路，可以使用微消息队列 MQTT 消息实现其他自定义流程，例如销毁会议、中途拉人入会、禁言等操作。微消息队列 MQTT 在音视频会议场景中充当了信令传输的角色。

注意事项

上述流程简要描述了如何使用微消息队列 MQTT 和音视频通信 RTC 快速构建自己的音视频通话 App。具体的 SDK 说明请参见微消息队列 MQTT、消息队列 MQ 以及音视频通信（RTC）文档。

其中使用微消息队列 MQTT 构建音视频通话场景的信令传输时，相关的消息类型设计以及参数设计请尽可能遵循如下原则：

- 客户端 ID 映射

  MQTT 协议要求每个客户端都有一个全局唯一的 Client ID，Client ID 由以下两部分组成，这两部分通过 “@@@” 分隔符连接，只需要保证最终的 Client ID 唯一且总长度不超过 64 个字符即可：

  - 前缀 Group ID：Group ID 需在微消息队列 MQTT 控制台申请。建议 Group ID 按照 App 的平台或者渠道进行粗分类，例如 Android 和 iOS 的客户端分成不同的 Group ID，或者不同版本的客户端使用不同的 Group ID，方便问题定位。
  - 后缀 Device ID：Device ID 由应用生成。Device ID 可以和用户 App 账号 ID 进行一一映射，确保全局唯一。

  Client ID 的更多信息请参见#unique_7。

- Topic 名称映射

  使用微消息队列 MQTT 收发消息需要了解 MQTT 协议订阅关系的模型，详情请参见协议文档和官网文档。

  MQTT 是遵循发布/订阅模型的消息协议，订阅关系和 Topic 符合目录树格式，Topic 可分为父级 Topic 和子级 Topic，Topic（包含父级 Topic 和子级 Topic）的总长度不能超过 64 个字符：

  - 父级 Topic：通常称目录树第一级的 Topic 为父级 Topic。父级 Topic 需要在微消息队列 MQTT 控制台申请后才可使用，申请后相当于一个 Namespace。
  - 子级 Topic：目录树第一级的 Topic 的后续部分称为子级 Topic。子级 Topic 无需申请，业务方可以随意指定。

  Topic 的更多信息请参见#unique_7。

  业务方设计用于消息收发的 Topic 时，需要遵循以下原则：

  - 上行消息（终端 App 发给管控服务的消息）和下行消息（管控服务发给终端 App 的消息）使用不同的父级 Topic。
  - 不同优先级或者消息量级差别比较大的消息使用不同的父级 Topic。

  对于上文描述的交互流程，建议使用微消息队列 MQTT 提供的 P2P 消息，P2P 消息不需要订阅，发送方直接指定对端接收即可，详情请参见 #unique_4。

- 收发消息参数设计

  由于移动 App 可能存在应用切入后台被杀死（Kill）导致移动 App 不在线的情况，需对终端 App 做以下配置，以确保终端 App 离线重连后可以收到之前的消息：

  - `cleanSession` 参数设置为 “false”
  - QoS 设置成 “1”

  终端 App 应该对收到的消息做去重以及时效性校验（比如终端 APP 离线超过 1 天，再次上线收到了 1 天前的消息）。

  `cleanSession` 和 QoS 的更多信息，请参见#unique_7。

2 新零售电子价签解决方案（MQTT）

新零售电子价签解决方案由阿里云微消息队列 MQTT 推出，通过 MQTT 以实现商场超市、公共场所电子标签、多媒体屏幕的数据更新管理。本文将以电子价签为例详细描述该解决方案的系统架构、数据流设计以及注意事项，其他类似行业可参考该方案修改适配。

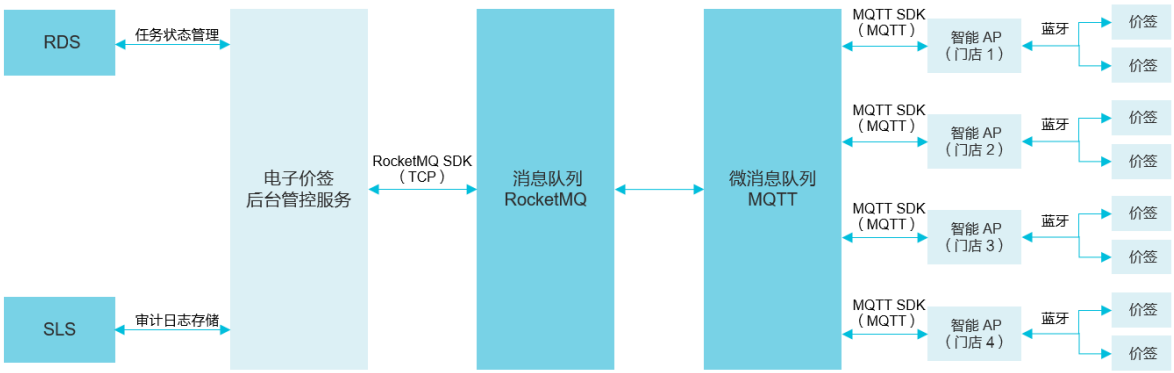
名词解释

MQTT	一种物联网和移动互联网领域的行业标准协议，适合移动终端之间的数据传输。微消息队列 MQTT 默认支持该协议。
MQTT 服务器	微消息队列 MQTT 提供的 MQTT 协议交互的服务端节点，用于接收消息并转发消息。
MQTT 客户端	用于和 MQTT 服务器交互的节点，本方案中特指发送或接收价格变更消息的智能 AP。
P2P 消息	微消息队列 MQTT 在标准的 MQTT 协议基础上提供的一种特殊消息，该类型消息无需普通的订阅关系匹配，便可直接发送给指定的单个目标 MQTT 客户端。详情请参见 #unique_4。
智能 AP	市面常见的智能路由器等网络设备，支持应用编程，可以同时承担互联网接入以及局域网设备控制等工作。
电子价签	实际分布在商场超市等场所中的电子显示屏幕，一般使用蓝牙、ZigBee 等无线传感网络协议和智能 AP 节点组网。
电子价签管控服务	电子价签系统中用于管理电子屏幕显示内容的后台服务，主要承担改价等人工操作的任务管理和查询工作。
RDS	阿里云推出的一种稳定可靠、可弹性伸缩的在线数据库服务。在电子价签系统中用来持久化改价等任务的状态变更。
SLS 日志存储	阿里云推出的日志存储服务，在电子价签系统中用来持久化保存所有操作日志，用于审计和溯源。

方案架构

在电子价签解决方案中，微消息队列 MQTT 与阿里云多个产品结合使用，实现价签的数据更新管理。图 2-1: 方案架构 展示了针对电子价签系统的解决方案架构。

图 2-1: 方案架构



如图 2-1: 方案架构 所示，在电子价签系统中，主要包含价签节点、智能 AP 节点、微消息队列 MQTT、消息队列 MQ、电子价签后台管控服务、RDS 以及 SLS。各个组件介绍如下：

- 智能 AP 负责转发上报价签的状态数据，并接收改价指令。智能 AP 按照门店或者场所分布，内部使用 MQTT SDK 从公网接入阿里云微消息队列 MQTT，该链路采用 SSL/TLS 加密传输，防止数据泄露。
- 一个智能 AP 下行链路和若干价签节点通过蓝牙、ZigBee 等无线传感网络协议组网，完成局域网内部数据交互。
- 电子价签后台管理服务部署在云端（云服务 ECS），使用消息队列 MQ 的 SDK 和消息队列 MQ 交互，MQTT 服务端和 MQ 服务端天然互通。
- 电子价签后台管理服务可以将改价等任务的状态变更持久化到 RDS 数据库以确保任务变更成功，并将价签上报数据和操作日志存储到 SLS，方便溯源和审计。

方案优势

新零售电子价签解决方案的优势如下所述：

- 服务能力强，可弹性伸缩
  - 微消息队列 MQTT 消息传输能力无限扩展，智能终端数量增加无需担心系统能力不足。
  - 微消息队列 MQTT 支持百万级设备毫秒级推送完成，电子价签屏显更新延迟更小。
- 适用范围广，通用性好，可快速复制
  - 基于 MQTT 标准协议实现，通用性好，方案只需简单适配数据内容即可快速复制到其他相似场景。

- 安全可靠
  - 微消息队列 MQTT 服务和智能 AP 节点数据传输支持 SSL/TLS 加密，无需担心媒体商业数据泄露。
  - 所有服务节点高可用，稳定性高。

数据交互

状态上报

1. 电子价签节点会采用定时轮询机制和智能 AP 节点交换数据，上报自己当前的显示状态、节点电量等信息。
2. 智能 AP 节点组织数据，并发送 MQTT 消息到 MQTT 服务器。
3. MQTT 服务器会将上报消息写入业务方指定的消息队列 MQ Topic。
4. 电子价签管控服务通过接收消息队列 MQ 消息，处理分析当前系统中在线的价签节点的状态，并将数据记录到 SLS。

更新屏显

1. 电子价签管控服务发送改价的消息队列 MQ 消息，触发改价操作。
2. MQTT 服务端会路由该消息队列 MQ 消息，将消息通过 MQTT 协议推送给目标智能 AP 节点。
3. 智能 AP 节点收到改价通知，将任务暂存。
4. 电子价签节点会采用轮询机制和智能 AP 节点交换数据，感知新的屏显内容。
5. 目标电子价签节点改价成功后，智能 AP 节点回发一条应答 MQTT 消息，通知电子价签管控服务当前任务已完成。
6. 电子价签管控服务将当前任务的执行记录写入 SLS 日志，方便后续溯源查询。

注意事项

上述流程简要描述了如何使用微消息队列 MQTT 和消息队列 MQ 来搭建电子价签系统，具体的 SDK 说明请参见微消息队列 MQTT 以及消息队列 MQ 文档。

其中使用微消息队列 MQTT 和消息队列 MQ 进行指令传输时，相关的消息类型设计以及参数设计请尽可能遵循如下原则：

- SDK 和协议选择

  电子价签场景中，一个应用可能存在成百上千的线下门店，一般每个门店配备若干个智能 AP 节点，智能 AP 节点会随着业务规模上升而增加，所以智能 AP 节点适合使用 MQTT 协议接入，而电子价签管控服务由于部署在云端，适合使用云上的消息队列 MQ 接入。

- 客户端 ID 映射

  MQTT 协议要求每个客户端都有一个全局唯一的 Client ID，Client ID 由以下两部分组成，这两部分通过 “@@@” 分隔符连接，只需要保证最终的 Client ID 唯一且总长度不超过 64 个字符即可：

  - 前缀 Group ID：Group ID 需在微消息队列 MQTT 控制台申请。Group ID 按照平台供应商或者渠道进行粗分类，例如不同的行业、批次分成不同的 Group ID，或者不同版本的客户端使用不同的 Group ID，方便问题定位。
  - 后缀 Device ID：Device ID 由应用生成。Device ID 可以使用智能 AP 节点的 MAC 地址等唯一性信息编码。

  Client ID 的更多信息请参见#unique_7。

- Topic 名称映射

  使用微消息队列 MQTT 收发消息需要了解 MQTT 协议订阅关系的模型，详情请参见协议文档和官网文档。

  MQTT 是遵循发布/订阅模型的消息协议，订阅关系和 Topic 符合目录树格式，Topic 可分为父级 Topic 和子级 Topic，Topic（包含父级 Topic 和子级 Topic）的总长度不能超过 64 个字符：

  - 父级 Topic：通常称目录树第一级的 Topic 为父级 Topic。父级 Topic 需要在微消息队列 MQTT 控制台申请后才可使用，申请后相当于一个 Namespace。
  - 子级 Topic：目录树第一级的 Topic 的后续部分称为子级 Topic。子级 Topic 无需申请，业务方可以随意指定。

  Topic 的更多信息请参见#unique_7。

  业务方设计用于消息收发的 Topic 时，需要遵循以下原则：

  - 不同类型的任务使用不同的父级 Topic，例如本场景中，改价任务和终端状态上报使用不同的父级 Topic。
  - 对于电子价签系统中，改价任务的交互消息建议使用微消息队列 MQTT 提供的 P2P 消息，P2P 消息不需要订阅，发送方直接指定对端接收即可，详情请参见#unique_4。

- **收发消息参数设计**

  由于电子价签场景改价任务一般要求实时推送，建议在智能 AP 和 MQTT 服务器交互过程中，智能 AP 做以下配置，以确保智能 AP 无需处理掉线期间的任务：

  - `cleanSession` **参数设置为 “true”**
  - **QoS 设置成 “1”**

  智能 AP 应该对收到的消息做去重以及时效性校验。

  `cleanSession` **和 QoS 的更多信息，请参见#unique_7。**

3 云存储网关利用 MQTT 服务实现统一管控

通过微消息队列 MQTT 所提供的服务，可采用统一的框架，即云控制台，来管理不同部署模式的云存储网关 CGS（Cloud Storage Gateway），并实现安全合规的公网和内网穿透，以此提供一致的用户体验。

名词解释

MQTT	一种物联网和移动互联网领域的行业标准协议，适合移动终端之间的数据传输。微消息队列 MQTT 默认支持该协议。
云存储网关 CSG（Cloud Storage Gateway）	下文简称 CSG，是一款可在用户的互联网数据中心 IDC（Internet Data Center）和阿里云上部署的网关实例，以阿里云 OSS 为后端存储，通过低成本的虚拟机服务器，给云上和云下应用提供业界标准的 NFS 和 CIFS 文件存储服务，以及 iSCSI 的块存储服务。详情请参见云存储网关官网文档。
MQTT 服务器	微消息队列 MQTT 提供的 MQTT 协议交互的服务端节点，用于接收消息并转发消息。
MQTT 客户端	用于和 MQTT 服务器交互的节点，本方案中特指 CSG 实例侧的客户端。
RDS	阿里云推出的一种稳定可靠、可弹性伸缩的在线数据库服务。
SLS	阿里云推出的日志存储服务，用于审计和溯源。

背景信息

CSG 是一款帮助用户在现有本地应用程序、基础设施和数据存储与阿里云之间实现无缝集成的存储服务。通过可在本地和云上部署的兼容行业标准存储协议的虚拟设备，将现有的存储应用程序和工作负载连接阿里云存储服务，无缝对接阿里云的存储和计算平台。

CSG 有以下两种部署形态：

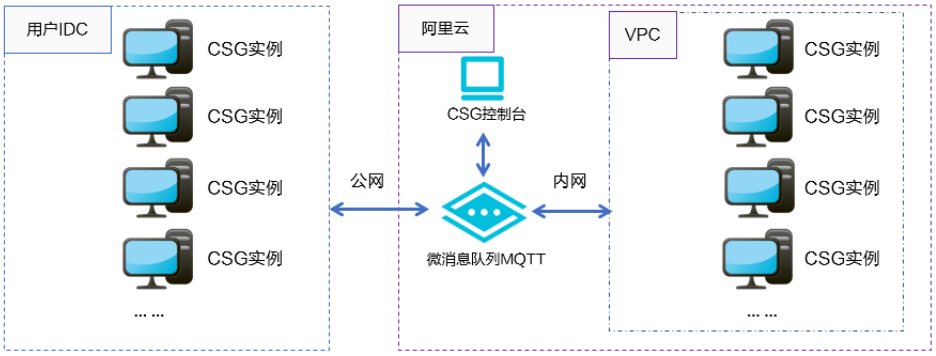
- 通过虚拟机方式部署和运行在用户自己的 IDC
- 直接部署和运行在阿里云的虚拟网络环境中的 ECS 上向用户提供服务

在这两种部署模式下，CSG 实例都是采用的内网部署，特别是在用户 IDC 部署的 CSG 虚拟机，出于安全的考虑，没有公网可以访问的 IP 地址，无法被 CSG 云控制台直接访问。部署在阿里云上的 CSG 也不太适合采用公网 IP 地址的形式和云控制台通讯，会有安全隐患和被攻击的可能。

微消息队列 MQTT 即可用于实现 CSG 云控制台和两种部署模式下的 CSG 实例的互通。

CSG 的两种部署模式下的公网和内网互通的架构图如图 3-1: 混合云存储部署 所示。

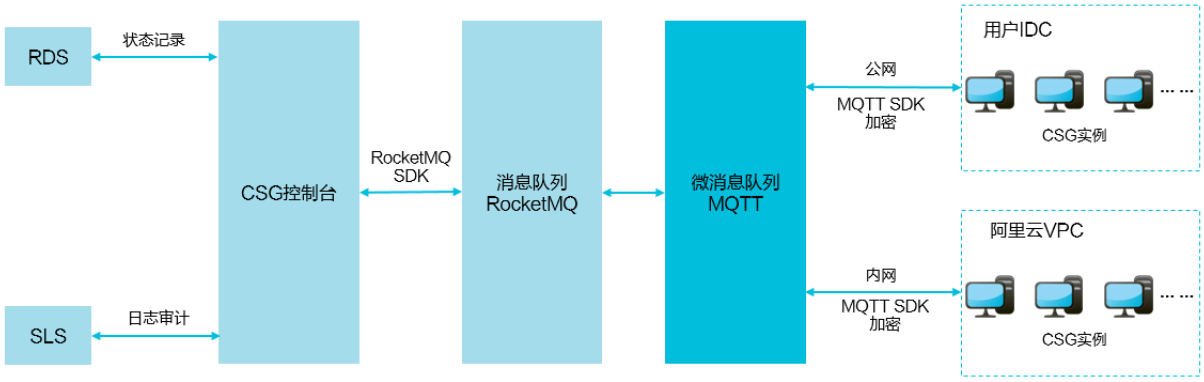
图 3-1: 混合云存储部署



方案架构

微消息队列 MQTT 用于消息的推送和收集，把部署在不同地理位置的 CSG 通过消息推送和主动拉取的方法，下发控制指令和收集 CSG 实例状态，用于在控制台的展示。同时基于微消息队列 MQTT 的推送功能也实现了输出 API 用于对每个 CSG 实例的管理。整体架构如图 3-2: 方案架构 所示。

图 3-2: 方案架构



方案优势

本方案的优势如下所述：

- 服务能力强，可弹性伸缩。
- 微消息队列 MQTT 消息传输能力无限扩展，CSG 实例数量增加无需担心系统能力不足。
- 微消息队列 MQTT 支持百万级设备毫秒级推送完成，CSG 的控制指令无拥塞。
- 微消息队列 MQTT 传输支持 SSL/TLS 加密，无需担心数据泄露。
- 所有服务节点高可用，稳定性高。
- 多种语言 SDK 支持方便服务端和应用端的开发。

- 内网和公网同时支持，适配多种网络需求。

注意事项

上述流程简要描述了如何使用微消息队列 MQTT 和消息队列 MQ 实现统一的跨网络的管理架构，具体的 SDK 说明请参见微消息队列 MQTT 和消息队列 MQ 文档。

其中使用微消息队列 MQTT 构建跨网络的信令传输时，相关的消息类型设计以及参数设计请尽可能遵循如下原则：

- 客户端 ID 映射

  MQTT 协议要求每个客户端都有一个全局唯一的 Client ID，Client ID 由以下两部分组成，这两部分通过 “@@@” 分隔符连接，只需要保证最终的 Client ID 唯一且总长度不超过 64 个字符即可：

  - 前缀 Group ID：Group ID 需在微消息队列 MQTT 控制台申请。建议 Group ID 按照 CSG 控制台所在的阿里云地域进行分类，例如华东1（杭州）和华东2（上海）采用不同的 Group ID，方便定位出有问题的地域和消息队列。
  - 后缀 Device ID：Device ID 由应用生成。Device ID 可以和每个 CSG 实例的 ID 进行一一映射，确保全局唯一。

  Client ID 的更多信息请参见#unique_7。

- Topic 名称映射

  使用微消息队列 MQTT 收发消息需要了解 MQTT 协议订阅关系的模型，详情请参见协议文档和官网文档。

  MQTT 是遵循发布/订阅模型的消息协议，订阅关系和 Topic 符合目录树格式，Topic 可分为父级 Topic 和子级 Topic，Topic（包含父级 Topic 和子级 Topic）的总长度不能超过 64 个字符：

  - 父级 Topic：通常称目录树第一级的 Topic 为父级 Topic。父级 Topic 需要在微消息队列 MQTT 控制台申请后才可使用，申请后相当于一个 Namespace。
  - 子级 Topic：目录树第一级的 Topic 的后续部分称为子级 Topic。子级 Topic 无需申请，业务方可以随意指定。

  Topic 的更多信息请参见#unique_7。

  业务方设计用于消息收发的 Topic 时，需要遵循以下原则：

  - 上行消息（CSG 实例发给管控服务的消息）和下行消息（管控服务发给 CSG 实例的消息）使用不同的父级 Topic。
  - 不同优先级或者消息量级差别比较大的消息使用不同的父级 Topic。

  对于上文描述的交互流程，建议使用微消息队列 MQTT 提供的 P2P 消息。P2P 消息不需要订阅，发送方直接指定对端接收即可，详情请参见 #unique_4。

- 收发消息参数设计

  由于 CSG 控制台是利用 MQTT 模拟的 RPC 调用，控制台下发的消息，没有返回值就直接用 `noreply` 表示，无需重复接受之前的消息。控制台会在接到错误码后，提示用户再次下发指令。需在 MQTT 客户端做以下配置：

  - `cleanSession` 参数设置为 “true”

  `cleanSession` 的更多信息，请参见#unique_7。