

应用云渲染 快速入门



腾讯云

【 版权声明 】

©2013–2025 腾讯云版权所有

本文档（含所有文字、数据、图片等内容）完整的著作权归腾讯云计算（北京）有限责任公司单独所有，未经腾讯云事先明确书面许可，任何主体不得以任何形式复制、修改、使用、抄袭、传播本文档全部或部分内容。前述行为构成对腾讯云著作权的侵犯，腾讯云将依法采取措施追究法律责任。

【 商标声明 】



及其它腾讯云服务相关的商标均为腾讯云计算（北京）有限责任公司及其关联公司所有。本文档涉及的第三方主体的商标，依法由权利人所有。未经腾讯云及有关权利人书面许可，任何主体不得以任何方式对前述商标进行使用、复制、修改、传播、抄录等行为，否则将构成对腾讯云及有关权利人商标权的侵犯，腾讯云将依法采取措施追究法律责任。

【 服务声明 】

本文档意在向您介绍腾讯云全部或部分产品、服务的当时的相关概况，部分产品、服务的内容可能不时有所调整。您所购买的腾讯云产品、服务的种类、服务标准等应由您与腾讯云之间的商业合同约定，除非双方另有约定，否则，腾讯云对本文档内容不做任何明示或默示的承诺或保证。

【 联系我们 】

我们致力于为您提供个性化的售前购买咨询服务，及相应的技术售后服务，任何问题请联系 4009100100或 95716。

文档目录

快速入门

基本技术概念

快速入门

接入 Demo

技术接入

启动应用

关闭应用

重连应用

无客户端模式

排队功能

数据通道

心跳连接

启用麦克风

启用摄像头

自适应分辨率

鼠标、键盘、触摸交互处理

快速入门

基本技术概念

最近更新时间：2025-01-09 10:39:32

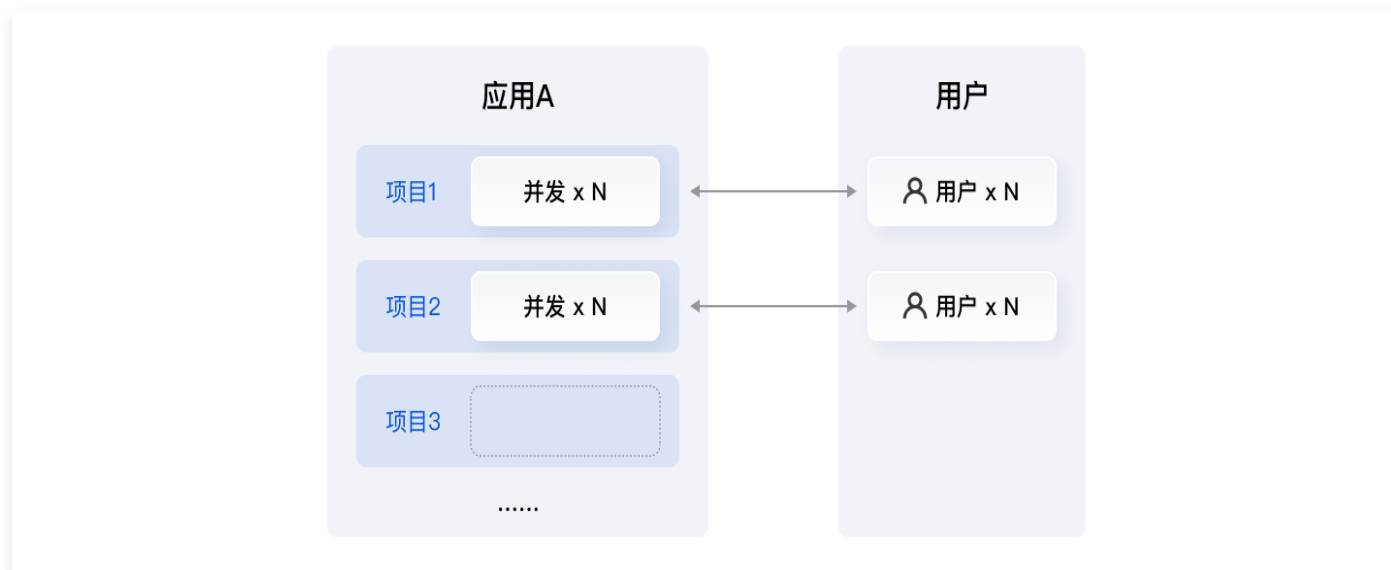
什么是应用云渲染

应用云渲染是一套整合了应用管理、并发资源管理与调度、音视频串流等能力的 PaaS 产品。它可以将您的应用程序（如 UE、Unity 等应用）运行在云端并发实例上，并通过音视频串流技术，让用户通过视频流的方式访问云上应用，无需下载安装。一般情况下，您上传的应用无需使用任何特殊插件，应用云渲染已实现将并发实例中渲染出来的音视频结果传输到终端，并将终端用户的输入实时发送给并发实例。



应用云渲染存在应用、项目和并发三个基础概念：

- **应用：** 您开发好的应用程序，需要确保可在 Windows Server 或 Android 环境下稳定运行。您需在控制台上传应用，让它在云端运行。
- **项目：** 您可以通过项目将并发分组，并通过关联应用来指定可以使用这些并发的应用。您可针对一个应用新建不同项目，满足不同的测试、上线场景而无需重复上传应用。
- **并发：** 一个并发即一台虚拟化的云端实例（包括 CPU、带宽、磁盘、GPU 等虚拟计算资源），用于在云端运行您的应用。如需了解并发配置、规格等详细信息，请查看 [计费说明](#)。



运行环境

云端软件环境

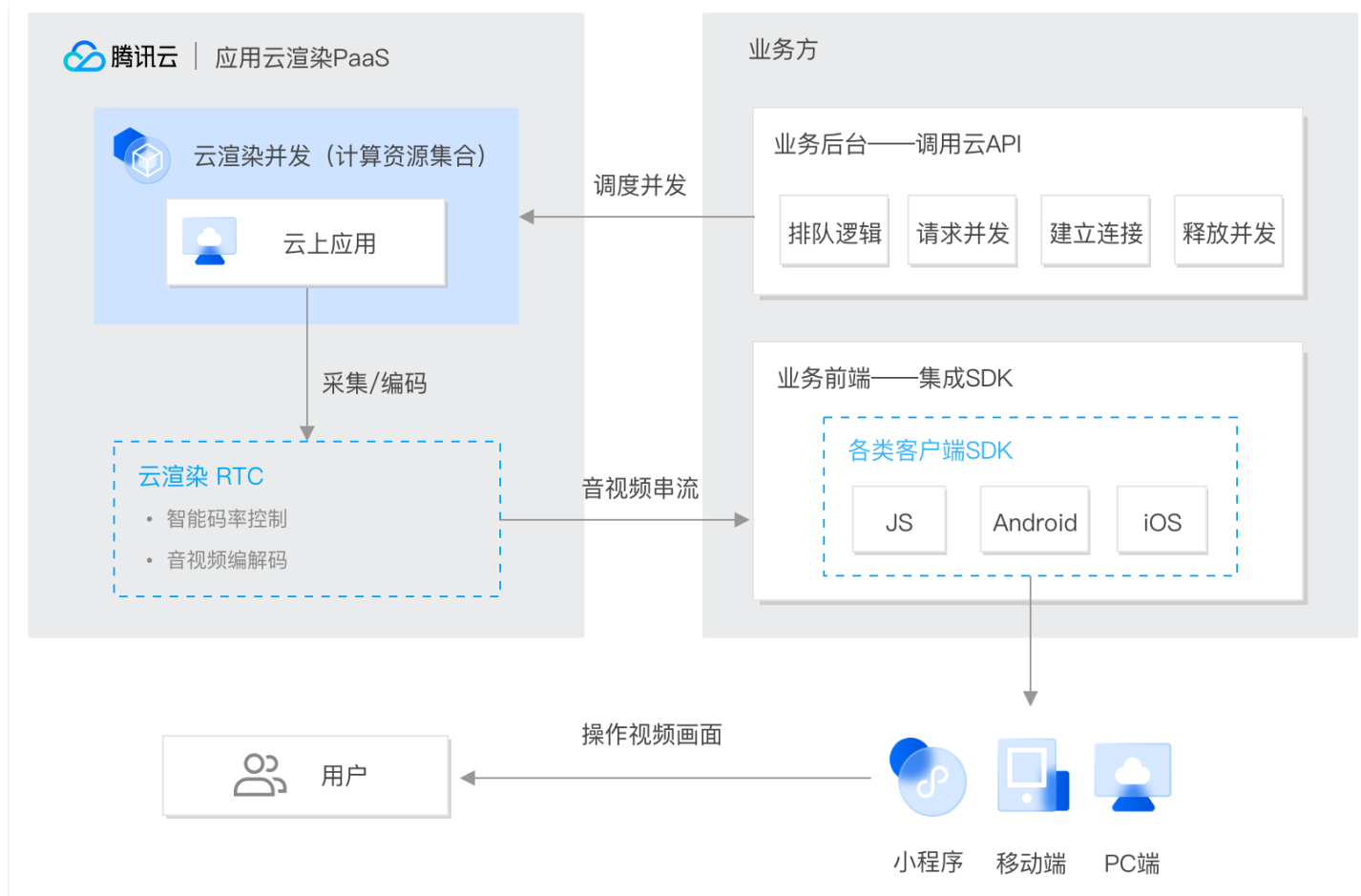
- **部署条件：**您所提供的云应用安装包需要支持“绿色”部署安装，即应用的运行不需要依赖修改注册表等其他系统配置。
- **系统限制：**不支持调用 Windows 系统的 PowerShell 和 CMD 服务。
- **端口限制：**不支持端口监听，因为服务提供的外网 IP 是不固定的；不支持子网广播，如有相关需求建议通过公网方式获取。
- **网络条件：**云端软件向外访问网络没有限制。

云渲染服务环境

网络条件：应用云渲染服务网络依赖 UDP 数据收发，用户端需要放开访问 UDP 8000 出口端口的限制，如无特殊安全问题，建议放开所有 UDP 端口限制。如果 Wi-Fi 或公司内网无法访问，可先用手机4G网络测试，若4G网络可以访问云渲染服务，说明 Wi-Fi 或公司内网可能有网络限制。

接入服务能力

应用云渲染提供 [0开发快速上线](#) 的能力，您可以通过控制台配置一个标准化的前端页面并正式上线使用。同时，作为前后端一体的 PaaS 产品，应用云渲染也提供后端 API 和各终端 SDK。您可以通过我们提供的 [接入 Demo](#)，搭建自己的后台服务和客户端程序，从而满足您独特的业务需求。



应用云渲染后台 API:

应用云渲染提供了业务所需调用的后台 [云 API](#)，通过云 API 您的业务后台可以完成请求您的并发资源以及创建会话等功能。

什么是云API:

- 云 API:** 云 API 3.0 版本 是腾讯云开放生态的基石，具备易于自动化、易于远程调用、兼容性强及对系统要求低等优势。通过云 API，您只需少量的代码即可快速操作云产品，针对频繁调用的功能可以提高效率。您还可将云 API 进行组合，实现更高级的功能。更多腾讯云 API 3.0 使用介绍请参见 [云API快速入门](#)。
- 访问密钥:** 访问密钥即云 API 密钥，是用户访问腾讯云 API 进行身份验证时需要用到的安全凭证，由 SecretId 和 SecretKey 一起组成。若用户还没有 API 密钥，则需要在 API 密钥管理中新建，否则就无法调用云 API 接口。访问密钥请在 [API 密钥管理](#) 页面中获取。

应用云渲染终端 SDK:

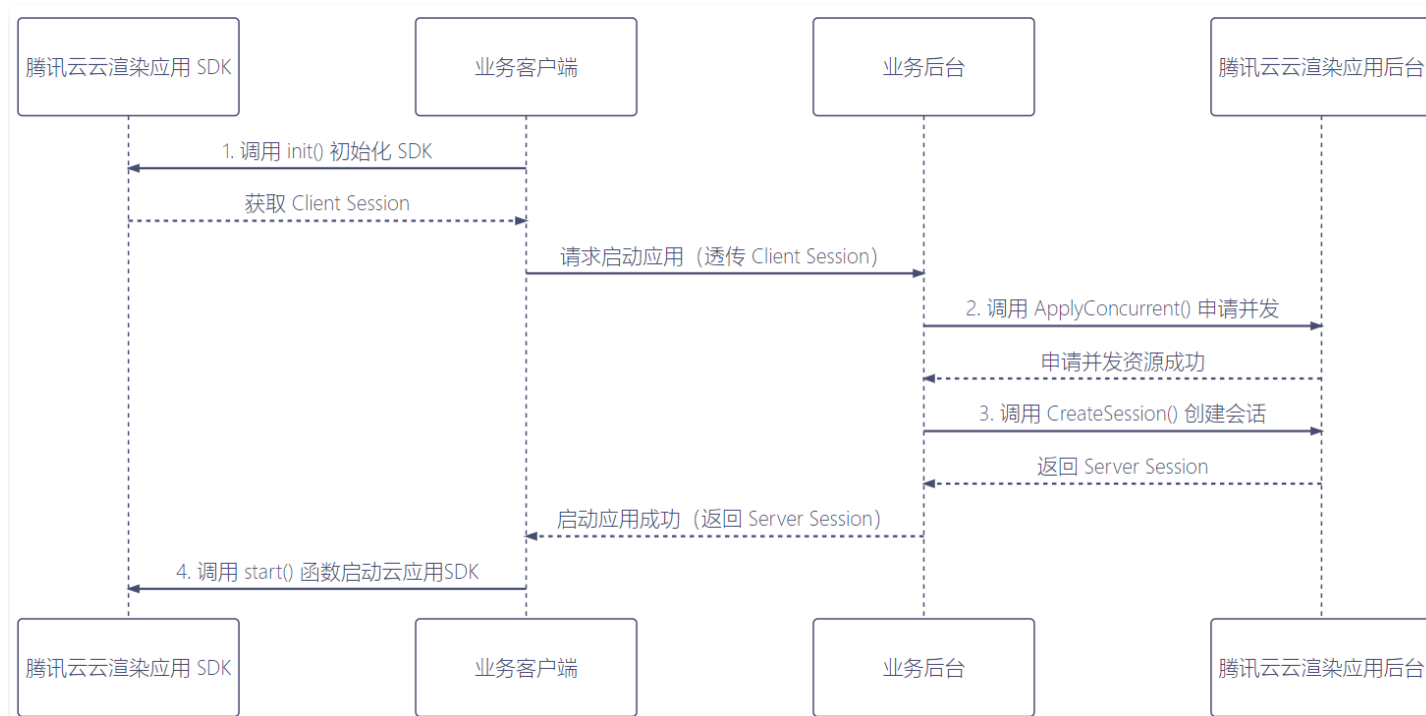
应用云渲染提供各终端 SDK 用于支持 Android 端、iOS 端、H5 以及小程序平台接入。您的客户端需要集成终端 SDK。SDK 提供了丰富的接口，满足大部分接入需求。更多详情可参见应用云渲染终端 [JS SDK](#)、[Android SDK](#)、[iOS SDK](#)。

● 业务后台：

业务需要搭建自己的后台服务，以此来分配和管理云渲染资源，我们提供了 [后台 Demo](#) 供参考，具体接入功能可以参考 [技术接入](#)。

● 业务客户端：

业务需要搭建自己的客户端服务，以此来获取和使用云渲染资源，我们提供了 [JS Demo](#)、[Android Demo](#) 以及 [iOS Demo](#) 供参考。



快速入门

最近更新时间：2024-09-23 16:39:21

什么是应用云渲染

应用云渲染是一套整合了应用管理、并发资源管理与调度、音视频串流等能力的 PaaS 产品。它可以在云端并发实例上运行您开发好的应用（EXE/APK 应用），终端用户无需下载应用包即可通过小程序、网页等体验应用。我们提供的终端 SDK 可将并发实例中渲染出来的音视频结果传输到终端，并将终端用户的操作实时发送给并发实例，您仅需确保应用可在 Windows Server 或 Android 环境正常运行，**无需任何特殊插件**即可轻松实现应用云化。



在阅读快速入门前希望您已经了解云渲染的 [基本技术概念](#)。

注册腾讯云账号并申请应用云渲染

1. 若您未注册腾讯云账号，请前往 [注册腾讯云账号](#)。
2. [申请应用云渲染服务](#) 后即可进入控制台。

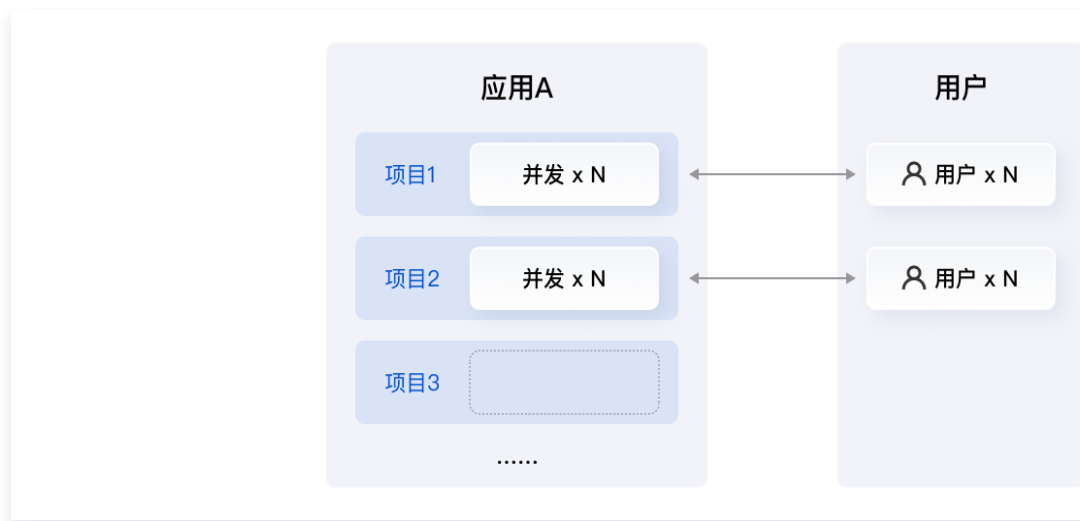
说明：

建议您将腾讯云账号进行 [企业实名认证](#)，以便我们更好的为您提供服务。

应用云渲染接入

接入前，希望您先理解应用、项目和并发三个基础概念的关系。

- **应用**：您开发好的应用程序，需要确保可在 Windows Server 或 Android 环境下稳定运行。您需在控制台上上传应用，让它在云端运行。
- **项目**：您可以通过项目将并发分组，并通过关联应用来指定可以使用这些并发的应用。您可针对一个应用新建不同项目，满足不同的测试、上线场景而无需重复上传应用。
- **并发**：即应用云渲染唯一计费项，一个并发即一台虚拟化的云端实例（包括 CPU、带宽、磁盘、GPU 等虚拟计算资源），用于在云端运行您的应用。如需了解并发配置、规格等详细信息，请查看 [计费说明](#)。



在控制台中，建议您按照**上传应用**、**新建项目**、**购买并发**的顺序操作，得到一个关联某款应用，并且分配有可用并发的项目。然后您可以针对该项目和应用进行**效果测试**，体验应用云化的基础效果，并选出最合适的并发规格。完成效果测试后，您可以对项目进行**上线发布**，我们支持：

- 0开发，在控制台上通过简单配置快速生成一个标准化的前端页面。
- 通过我们提供的接入 Demo，搭建自己的后台服务和客户端程序，从而满足您独特的业务需求。

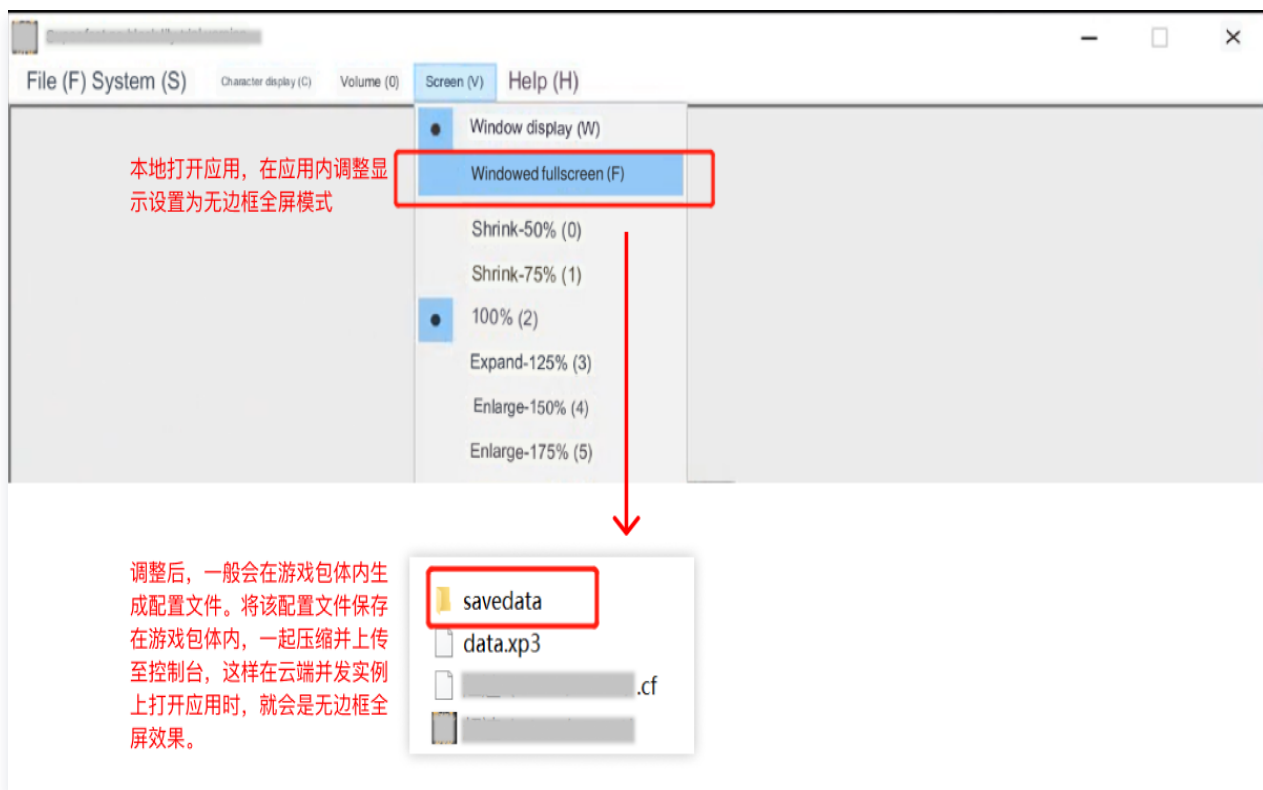


步骤1：新建应用

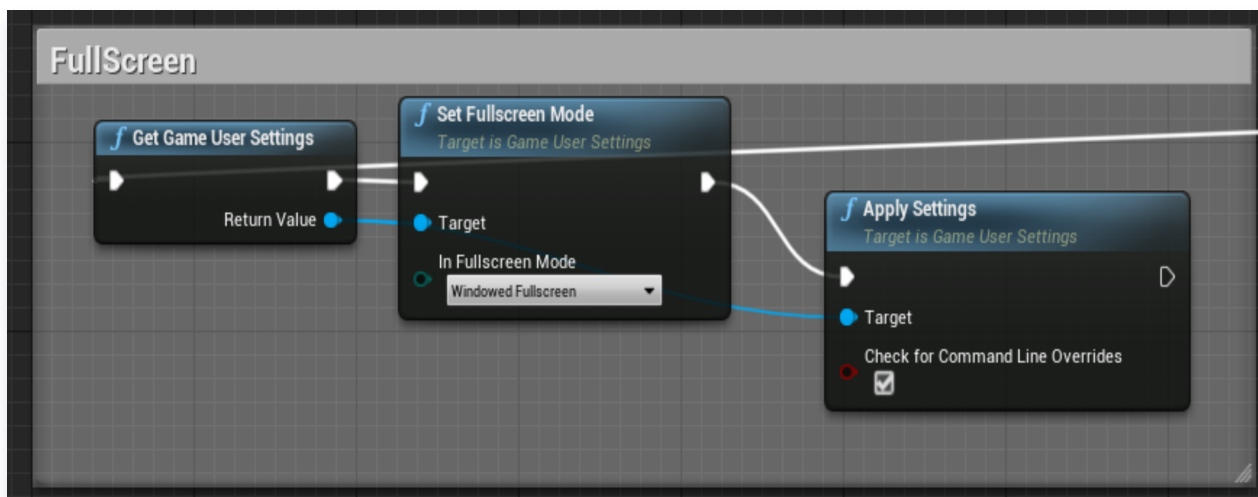
1. 上传应用：

1.1 在上传至云渲染平台前，**建议您先将应用设置为无边框全屏模式并开启自适应桌面分辨率**，便于后续使用 [自适应分辨率](#) 功能，实现画面全屏、自适应终端设备分辨率且无黑边的效果。

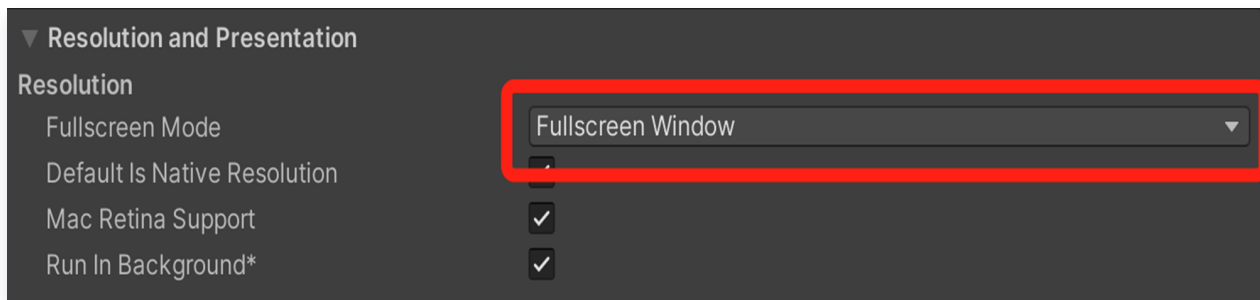
- 这步操作需要在应用层面完成。
- 有些应用可以在设置中调整显示模式，将调整后生成的配置文件一同打包上传至控制台即可。下图为某应用的全屏设置示例：



- 您也可以在应用开发时调整相关显示配置。UE 应用可以参考下面一段蓝图加到 Map 的 BeginPlay 里面开启自适应桌面分辨率。其中 In Fullscreen Mode 应该配置为 Windowed Fullscreen（窗口化全屏）。



Unity 应用可以参考下图配置为全屏模式。



或者增加应用启动参数 `-screen-fullscreen 1`，Unity 应用参数参考 [文档](#)。

`-screen-fullscreen`

Override the default full-screen state. This must be 0 or 1.

⚠ 注意：

- **应用独占全屏不属于无边框全屏模式**

云渲染的 [自适应分辨率](#) 功能是通过修改“云端桌面分辨率”来实现的。然而，应用独占全屏时，显示器分辨率是由应用控制的，此时强行修改桌面分辨率可能导致应用崩溃。

- **如何区分无边框全屏和独占全屏**

无边框全屏应用按 Alt+Tab 切换窗口不会导致显示器闪烁，独占全屏应用切换会有闪烁的现象。

1.2 将应用压缩打包为 ZIP / RAR / 7z 文件，在**我的应用**处单击**新建应用**上传压缩包，等待应用创建完成。

- 请使用 UTF-8 编码进行压缩（建议使用 7-Zip），否则可能导致解压后的文件或文件夹名称出现乱码等问题；
- 应用云渲染支持云 3D、云 XR 和云 APK 三种应用类型：
 - **云 3D**：非 AR/VR 类桌面应用，键鼠或触摸操控，例如 UE/Unity 虚拟应用、PC 游戏等。在云端运行的应用需为 EXE 格式，可适配网页、小程序、Android/IOS App 等终端。
 - **云 XR**：VR/AR/MR 应用，通常用双手柄操控，在云端运行的 XR 应用需为 EXE 格式，可适配 Pico/Oculus、手机/Pad、PC、全息等终端。
 - **云 APK（内测中）**：APK 格式应用。

❗ 说明：

- 目前云 APK 内测中，咨询请联系您的商务/架构师。
- 为保证云 XR 应用的体验，请在上传完应用（步骤1）并新建好项目（步骤2）后，**提供项目 ID（cap-xxxx）给我们进行必要后台配置**，您可以通过商务/架构师联系到云渲染工作人员。



2. 进行应用配置

- 主执行程序路径：为避免填写错误导致应用无法正常启动，建议您等待应用创建完成后直接单击选择获取启动路径。
- 云端画面捕捉区域：
 - 如果您已根据 前文1.1步骤，已将应用调整为无边框全屏模式并开启自适应桌面分辨率，请将画面捕捉模式设置为捕捉整个桌面。
 - 如您的应用只有非全屏窗口模式，无法调整为无边框全屏模式，可以使用仅捕捉应用窗口模式。您需要 根据 文档指引，填写正确的应用窗口名称和类名。当开发过程中未自定义窗口标题时，Demo.exe 启动后的窗口名称一般为 Demo；如您的应用是 UE 应用，类名一般为 UnrealWindow。

应用在本地图形卡上的显示模式：无边框全屏

应用云渲染画面捕捉模式（下图红框内部分）：捕捉整个桌面



应用在本地图形卡上的显示模式：非全屏窗口

应用云渲染画面捕捉模式（下图红框内部分）：仅捕捉应用窗口



应用在本地图形卡上的显示模式：非全屏窗口

应用云渲染画面捕捉模式（下图红框内部分）：捕捉整个桌面



- 详细配置说明请查看 [应用配置](#) 和 [应用增值功能（日志上传与用户存档功能）](#)。

应用云渲染

概览

我的应用

管理中心

项目管理

并发管理

场景服务

云端推流

应用配置

应用信息

应用ID

app-gx1qlvkq

应用名称

test

只支持中文、字母、数字或连接符“-”（16字符以内）

基础配置 查看填写示例

应用主执行程序路径

请填写应用包中主执行程序文件的相对路径

选择

进程列表-选填

应用拉起的所有进程exe，例“demo.exe”，多个进程之间使用英文“|”隔开

应用高级配置

应用启动参数配置-选填

若需要，可指定exe启动时传入的参数

云端画面捕捉区域

云端画面捕捉区域 整个桌面

终端用户看到的

① 请确保您的应用打包前已完成无边框全屏设置，若未设置，则终端用户无法看到应用全屏效果。如何将应用设置为无边框全屏模式 如果您的应用为非全屏窗口，则需设置仅捕捉应用窗口才可实现全屏效果

增值功能配置

日志上传

可自动上传指定文件，适用于获取用户操作日志等场景

该功能依赖腾讯云COS产品，正确配置后，可在用户关闭应用时自动上传指定文件夹（如Log日志文件）内容

用户存档读取与上传

可存档用户操作进度，适用于单机游戏存档等场景

该功能依赖腾讯云COS产品，正确配置后，会根据用户ID保存存档，并在用户使用或关闭应用时自动下载或上传存档

选择/填写正确的主执行程序路径，即应用 exe 文件的完整相对路径

[为避免填写错误导致应用无法正常启动，建议您等待应用创建完成后直接点击[选择] 获取启动路径]

推荐将应用调整为无边框全屏模式（需在应用打包上传前完成），并选择捕捉整个桌面的画面捕捉模式。

如有需要可以配置增值功能，可能产生一定腾讯云对象存储 COS 费用。

敬请期待

步骤2：新建项目并在项目下购买并发

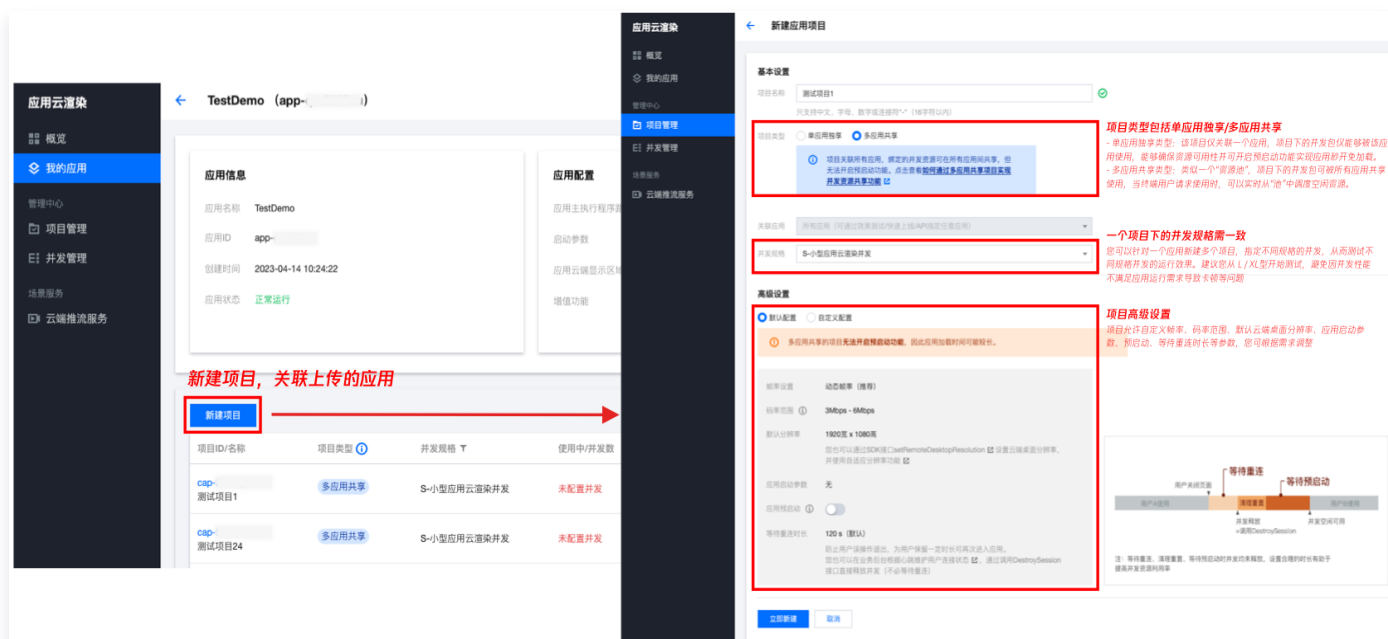
❗ 默认测试项目（Test Project）说明：

应用云渲染新用户可以在正式购买并发之前，可以通过默认测试项目（Test Project）**免费测试您上传的应用**。免费资源有限，因此**每次测试限时2分钟**。更多说明请查看 [默认测试项目与新手免费并发包](#)。

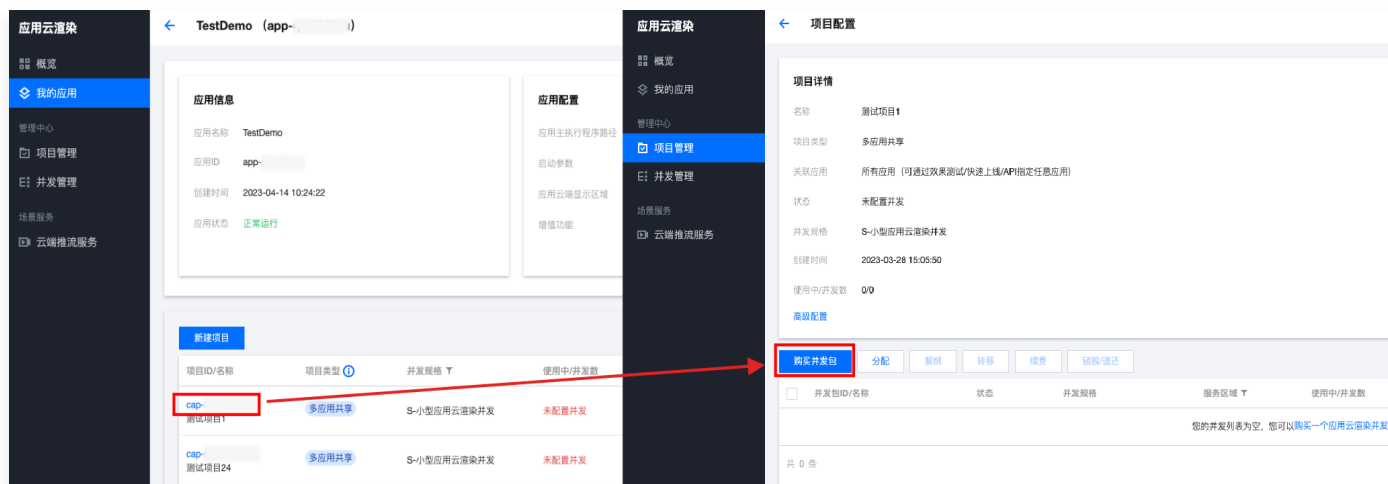
如您期望购买并发并正式上线使用，请新建一个项目。

1. 新建项目，关联上传的应用并指定并发规格。

- 项目类型：支持单应用独享和多应用共享两种类型的项目。单应用独享类型可以确保项目下的并发包仅被单个应用使用，而多应用共享类型下的并发包则可以作为“资源池”被多个应用共享使用。单击查看 [如何通过多应用共享项目实现并发资源共享](#)。
- 并发规格：一个项目下的并发规格需一致。您可以针对一个应用新建多个项目，指定不同规格的并发，从而测试不同规格并发的运行效果。建议您从 L / XL 型开始测试，避免因并发性能不满足应用运行需求导致卡顿等问题。不同规格并发的具体配置请参考 [计费说明](#)。
- 项目高级设置：项目允许自定义帧率、码率范围、默认云端桌面分辨率、应用启动参数、预启动、等待重连时长等参数，您可根据需求调整。

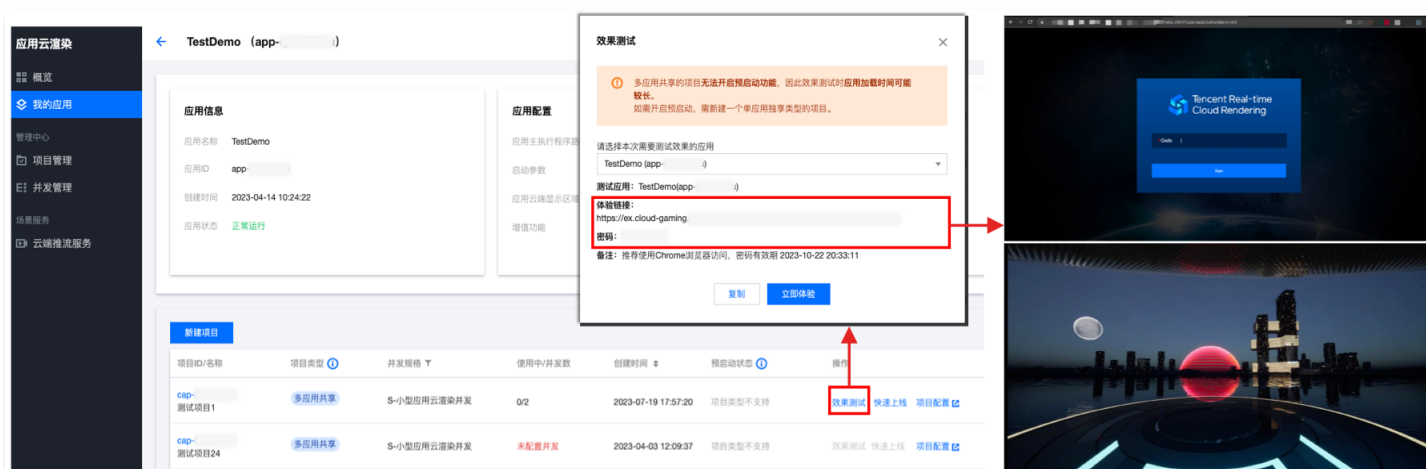


2. 在项目下购买并发包，等待并发包创建完成后，可以进入下一步效果测试。

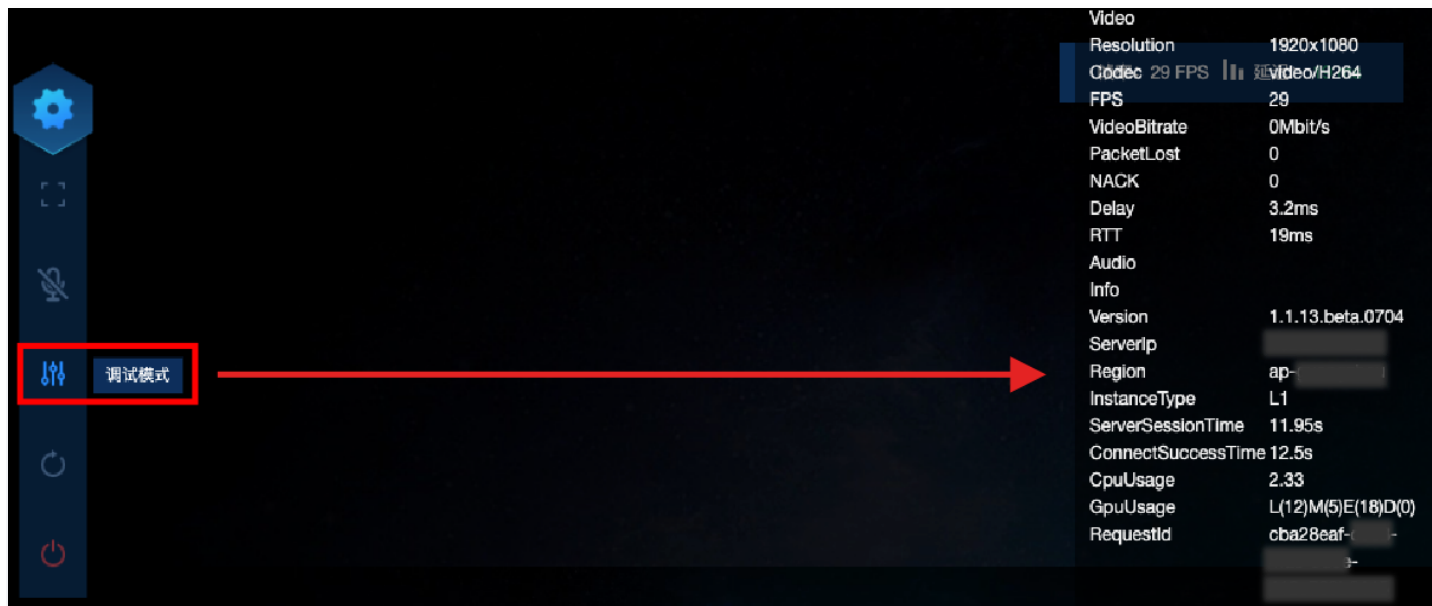


步骤3：效果测试

在项目的操作处，单击**效果测试**，生成体验链接和密码。进入体验页面，测试应用云化的基础效果。



您可以打开效果测试页面的工具栏中的“调试”（快捷键 `Ctrl+~` ），并关注以下数据，以选择最合适的并发规格：



- **FPS (帧率)**：正常情况下帧率应保持在30以上。如您发现进入某些特定场景，或进行特定操作时，帧率突然下降，可能是因为此时对 GPU 算力的消耗突然增加，而目前规格的并发带不动，从而导致掉帧，建议您尝试更高规格的并发。
- **RTT (往返时延)**：当 RTT 高于100ms，可能具有明显的延迟感。建议您先检查本地网络是否有抖动问题，可以尝试使用4G/5G网络访问。如果并发区域距离您的物理位置过远，也会造成RTT过高。
- **Region (并发区域)**：即并发所在的区域。“云渲染会根据终端用户的IP地址，自动调度距离最近的空闲并发”。如您发现并发区域距离过远，可能是距离更近的空闲并发数量不足导致的，请联系我们帮助您解决。
- **InstanceType (并发规格)**：云渲染并发规格分为 S、M、L、XL，分别适用于小型、中型、大型和超大型应用，详细配置请参考官方计费说明文档。如果您的应用对算力要求很高，却使用了较低规格的并发，可能导致 CPU/GPU 使用率过高，从而导致卡顿、闪退等问题。
- **CpuUsage (CPU 使用率)**：如果您发现 FPS 数据下降，检查是否出现了90–100%的 CPU 使用率，若是，说明该并发规格不足以处理负载，请尝试更高规格的并发。
- **GpuUsage (GPU 使用率)**：关注L (load) 的数值即可。如果您发现FPS数据下降，可以看下 L 是否出现了90–100%的数值，若是，说明该并发规格不足以处理负载，请尝试更高规格的并发。
- **RequestId**：如遇到问题/疑问，请尽量保持连接，同时联系我们并提供 RequestId。您也可以通过以下方式获取 RequestId：
 - 在 Chrome 浏览器开发者工具中 NetWork 获取 CreateSession 的 RequestId。
 - 如在下文步骤4-方式二中接入了云渲染 SDK，云 API 返回值中会包含 RequestId，建议业务后台记录下来。

更多效果测试指引，请参见 [效果测试](#)。

❗ 效果测试常见问题说明：

- 提示“没有空闲并发”：

控制台查看请求项目下是否有空闲并发，“上一个用户退出后并发需进行自动清理（约1分钟），完成清理后才会恢复空闲可用状态并被下一个用户使用”。

- **应用无法启动或黑屏：**

- 若一直在启动中/加载中，无法进入：如果正在使用 Wi-Fi 或公司内网，可先用手机4G网络测试，若4G网络可以访问云渲染服务，说明 Wi-Fi 或公司内网可能限制了 UDP。应用云渲染服务网络依赖 UDP 数据收发，用户端需要放开访问 UDP 8000 出口端口的限制，如无特殊安全问题，建议放开所有 UDP 端口限制。
- 若提示建议“启动失败，请检查应用启动路径配置”：检查是否正确填写了 [应用主执行程序路径](#)。
- 若进入后，没有加载出应用画面/黑屏：如该应用 [使用了捕捉应用窗口模式](#)，需确保窗口名称与类名已正确填写，否则将会是黑屏画面。

- **应用启动慢：**

- 多应用共享类型的项目不支持预启动（即在并发实例中提前运行应用从而让用户使用时可以直接进入加载后的画面），因此应用启动时间相对会增加。
- 单应用独享类型的项目可以开启预启动，从而实现应用秒开加载。
- 上一个用户退出后并发会被回收和清理重置，待恢复空闲可用状态后并发会重新进行应用预启动。如果下一个用户在并发刚恢复空闲时进入，可能预启动尚未完成。

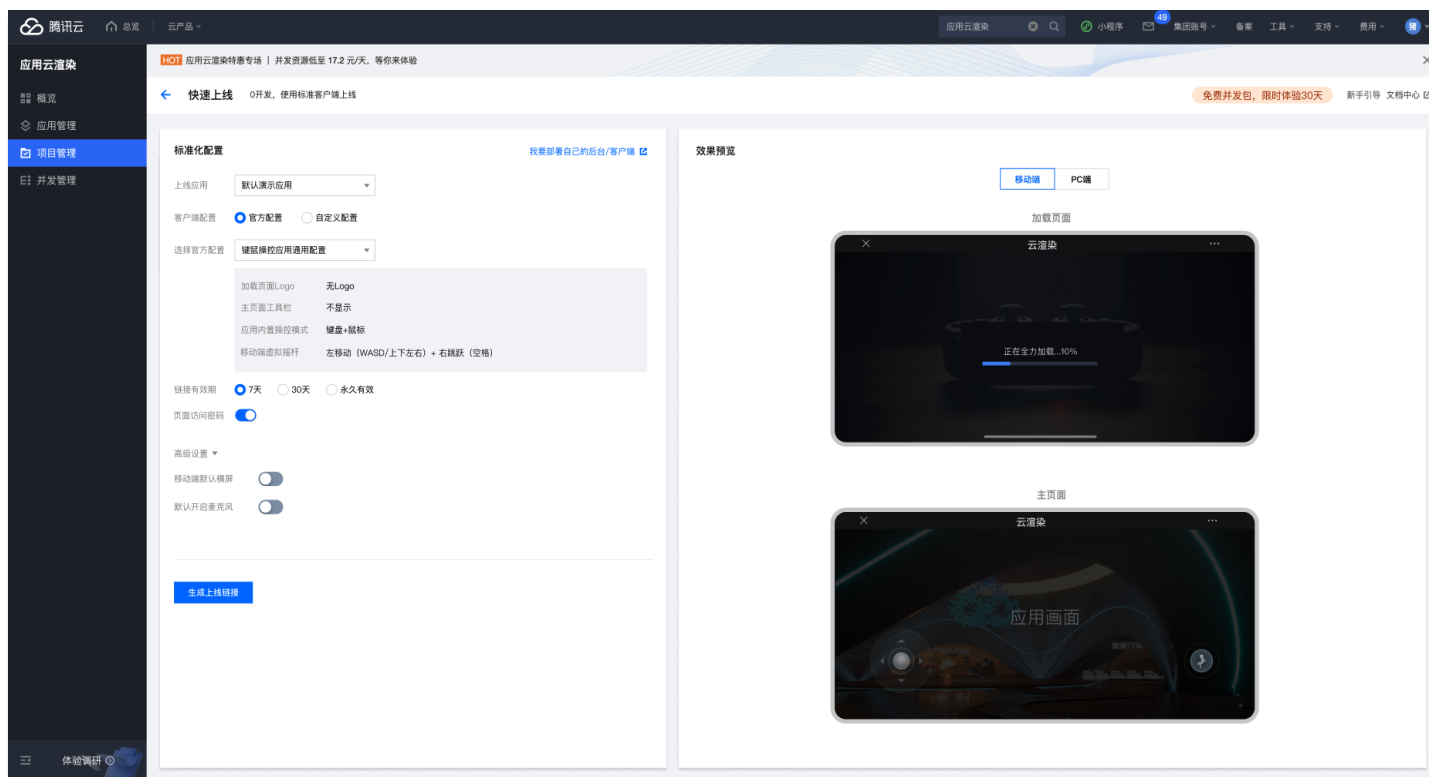
- **延时高/RTT数据不正常：**建议您先检查本地网络是否有抖动问题，可以尝试使用4G/5G网络访问。
- **卡顿：**有可能是因为应用对算力要求较高而使用了过低规格的并发（如 S 型），建议您先尝试更高规格并发的效果。
- **提示“云渲染连接已断开”：**如您使用了公司 Wi-Fi 网络，建议您检查网络是否满足 [应用云渲染服务环境](#)，即已开放 UDP 端口限制。同时您也可以检查本地网络是否有抖动问题，可以尝试使用 4G/5G 移动网络访问。

如以上建议仍无法解决您的问题，请联系我们。更多常见问题请参见 [云上应用相关](#)。

步骤4：上线发布

方式一：0开发快速上线

应用云渲染支持对页面进行标准化配置，0开发生成上线链接，详细操作步骤请参考 [快速上线](#)。



方式二：接入云渲染 PaaS

为实现您的独特业务需求、保证最佳用户体验，您可以进行后台与客户端开发来搭建自己的后台服务和客户端程序，接入应用云渲染 PaaS 提供 [云 API](#) 和各终端 SDK（[JavaScript SDK](#)、[Android SDK](#)、[iOS SDK](#)）。您无需从头开始接入，我们提供接入 Demo 帮助您快速实现业务上线。



1. 部署业务后台 Demo

根据 [指引](#)，在本地或者任意服务器部署业务后台 Demo，Demo 较为轻量对服务器配置没有特殊要求。

❗ 业务后台必须部署无法省略：

因为访问应用云渲染 云 API 服务需要提供腾讯云账号的 SecretId 和 SecretKey（可在 [控制台 API 密钥管理](#) 中获取），为了您的财产和服务安全，该信息需要在后台处理。另外您也需要业务后台来管理用户会话，完成用户排队等功能。更多信息可查看 [客户端与后台基础逻辑说明](#)。

2. 部署客户端 Demo（以 JavaScript Demo 为例）

2.1 下载 [TCGSDK](#)，保存在本地 PC 的 C 或 D 盘内，或 Mac 的 Downloads 文件夹。保留该文件夹的路径（例如：`c:/cloudgame-js-sdk`）。

2.2 下载 [Demo HTML 文件](#)。

2.3 替换 Demo 中的以下路径/参数，具体请参见 [更多 JavaScript Demo 相关指引](#)：

需调整路径/参数	说明
<code>src="path/to/tcg-sdk"</code>	TCGSDK 的路径，例如： <code>src="./cloudgame-js-sdk/dist/tcg-sdk/index.js"</code>
<code>url = 'http://xxxx/StartProject'</code>	将 <code>xxxx</code> 替换为业务后台 Demo 的地址，例如： <code>url = 'http://192.168.0.1:3000/StartProject'</code>
<code>ProjectId: 'project-id'</code> <code>UserId: 'user-id'</code>	<code>ProjectId</code>：控制台内创建的项目 ID <code>UserId</code>：指当前连接至云渲染服务的终端用户的唯一标识，由业务方自定义，云渲染平台不予理解。（可根据业务需要决定使用用户的唯一身份标识或是使用时间戳随机生成；在用户重连时应保持 <code>UserId</code> 不变） ⚠ 注意： 同一 <code>UserId</code> 同一时间只能使用一个并发。若客户端 Demo 中的 <code>UserId</code> 被设定为固定值而不是随机数，那么当不同用户请求连接云渲染时，会被识别成同一个用户，导致互相抢占同一个并发实例，云渲染会自动把目前已连接使用的并发给到最近一次请求的用户设备，原来的用户设备会出现黑屏断连。

2.4 对于移动端操作，应用云渲染官方有提供摇杆插件（[Joystick](#)），可用于映射键盘 WASD/上下左右按键。

2.5 当用户在线数大于资源并发数量时，建议业务引入用户排队系统，提升用户体验。排队逻辑涉及前后台，前端代码可参考 [排队示例](#)，签名逻辑可以根据业务需求决定是否添加，后台部分请查看 [技术接入-排队功](#)

能。

3. 测试调试，业务上线

如正确完成控制台操作，并成功部署业务后台和客户端，您可以通过 Chrome 浏览器（建议）打开 HTML 演示文件，画面加载成功后会显示云渲染画面，您可以通过鼠标拖动 / 键盘输入等方式与应用互动。您可以基于 Demo 进行更多定制化开发。测试 CheckList 参考：

- 移动端& PC 端的鼠标、键盘、触摸交互操作体验是否正常。

尤其需注意移动端，由于应用运行在云端 Windows 系统上，当云渲染 Web 页面运行在用户的 Android/iOS 的浏览器中时，用户触摸消息被转换为鼠标消息传至云端 Windows 系统。因此，为保证体验，您可能需要对应用做些适配，详细请查看 [鼠标、键盘、触摸交互处理](#)。

- 若应用内有麦克风语音和摄像头功能，需测试是否正常。

- 麦克风/摄像头上行功能默认关闭，如需要使用，需 [在 InitConfig 中启用](#)，或通过 [TCGSDK.switchMic / TCGSDK.switchCamera](#) 接口动态开关。
- 终端用户需授权使用麦克风/摄像头数据，如获取权限失败，会导致麦克风/摄像头上行功能异常。
- Web 端需要域名开启 HTTPS 协议才能正常获取麦克风/摄像头权限。

- 提示最终用户检查本地网络是否满足 [应用云渲染服务环境](#)。

应用云渲染服务基于 WebRTC 协议，数据传输依赖 UDP。如果出现了必现的连通性问题请检查网络配置是否限制了 UDP，如无特殊安全问题，请放开 UDP 的限制，或者使用其他无网络限制的环境测试。

- 注意不要仅使用一个并发来测试，应注意测试多用户同时连接的情况。

同一 `UserId` 同一时间只能使用一个并发。若客户端 Demo 中的 `UserId` 被设定为固定值而不是随机数，那么当不同用户请求连接云渲染时，会被识别成同一个用户，导致互相抢占同一个并发实例，云渲染会自动把目前已连接使用的并发给到最近一次请求的用户设备，原来的用户设备会出现黑屏断连。因此，请注意正确设置 `UserId`，并使用多个并发测试多人同时在线的效果。

如测试过程中遇到问题，请联系我们并提供 [RequestId](#)。

❗ 相关指引：

- [技术接入文档](#)：包括排队功能、数据通道、心跳连接、自适应分辨率、启动麦克风等详细指引。
- [最佳实践文档](#)：基于全真互联未来会场小程序案例，剖析如何接入应用云渲染产品，如何使用排队、数据通道、多点触控、自适应分辨率等功能。
- [JavaScript SDK 接口文档](#)和 [Demo](#)：包括 JavaScript Demo 和 TCGSDK 丰富接口的详细说明文档，方便业务基于 JavaScript Demo 进行定制化开发。
- [Android SDK 接口文档](#)和 [Demo](#)：包括 Android SDK 的详细说明文档，以及三个示例工程和一个用于体验的 APK。
- [iOS SDK 接口文档](#)和 [Demo](#)：包括 iOS SDK 的详细说明文档，以及集成 SDK 的简单 Demo。

附录

业务后台与客户端基础逻辑说明

1. 客户端初始化云渲染 SDK:

无论您使用的哪种客户端，在初始化 SDK 成功后都可以获取到 ClientSession。ClientSession 将用于后续业务 Server 获取 ServerSession，各端具体初始化及获取方式如下：

- **JavaScript SDK:** 调用 `TCGSDK.init(params)` 接口完成初始化构建。初始化完成后，客户端调用 `TCGSDK.getClientSession()` 函数，获取 Client 端的 ClientSession。
- **Android SDK:** 通过 `TcrSdk.getInstance().init(context, null, callback)` 接口完成SDK初始化。初始化完成后，客户端在 `TcrSession.Observer#onEvent` 中收到 `STATE_INITED` 通知后，获取 Client 端的 ClientSession。

2. 后台服务锁定应用云渲染并发:

您的后台服务调用云渲染 API `ApplyConcurrent()` 锁定云渲染并发，收到成功回调后执行下一步。

3. 后台服务获取 ServerSession:

您的后台服务调用云渲染 API `CreateSession(ClientSession)`，获取成功回调中的 ServerSession 返回给客户端。

4. 启动云渲染服务:

各端 SDK 在收到 ServerSession 后启动云渲染服务的调用方式略有不同，请根据您的需求查看对应指引。

- **JavaScript SDK:** 客户端调用 `TCGSDK.start(ServerSession)` 函数启动云渲染。
- **Android SDK:** 客户端调用 `TcrSession.start(serverSession,callback)` 函数启动云渲染。

接入 Demo

最近更新时间：2025-01-09 10:39:32

为实现您的独特业务需求、保证最佳用户体验，您可以进行后台与客户端开发来搭建自己的后台服务和客户端程序，接入应用云渲染 PaaS 提供 [云 API](#) 和各终端 SDK（[JavaScript SDK](#)、[Android SDK](#)、[iOS SDK](#)）。您无需从头开始接入，我们提供接入 Demo 帮助您快速实现业务上线。

部署业务后台 Demo

根据 [指引](#)，在本地或者任意服务器部署业务后台 Demo，Demo 较为轻量对服务器配置没有特殊要求。后台 Demo 支持实现包括启动应用、退出应用、用户排队等功能，更多具体接入功能请参见 [技术接入](#)。

❗ 业务后台必须部署无法省略：

因为访问应用云渲染 云 API 服务需要提供腾讯云账号的 SecretId 和 SecretKey（可在 [控制台 API 密钥管理](#) 中获取），为了您的财产和服务安全，该信息需要在后台处理。另外您也需要业务后台来管理用户会话，完成用户排队等功能。更多信息可查看 [客户端与后台基础逻辑说明](#)。

部署客户端 Demo

JavaScript Demo

应用云渲染 JavaScript Demo 提供了基础的云渲染能力示例代码，支持包括 PC 浏览器、移动端 H5 页面以及小程序内嵌 WebView 等多种落地场景。

1. [下载TCGSDK](#)，保存在本地 PC 的 C 或 D 盘内，或 Mac 的 Downloads 文件夹。保留该文件夹的路径（例如：`c:/cloudgame-js-sdk`）。
2. [下载 Demo HTML 文件](#)
3. 替换 Demo 中的以下路径/参数，单击可查看 [更多 JavaScript Demo 相关指引](#)：

需调整路径/参数	说明
<code>src="path/to/tcg-sdk"</code>	TCGSDK 的路径，例如： <code>src="./cloudgame-js-sdk/dist/tcg-sdk/index.js"</code>
<code>url = 'http://xxxx/StartProject'</code>	将 <code>xxxx</code> 替换为业务后台 Demo 的地址，例如： <code>url = 'http://192.168.0.1:3000/StartProject'</code>
<code>ProjectId: 'project-id'</code> <code>UserId: 'user-id'</code>	• <code>ProjectId</code>：控制台内创建的项目 ID • <code>UserId</code>：指当前连接至云渲染服务的终端用户的唯一标识，由业务方自定义，云渲染平台不予理解。（可根据业务需要决定使用

用户的唯一身份标识或是使用时间戳随机生成；在用户重连时应保持 UserId 不变）

4. 对于移动端操作，应用云渲染官方有提供摇杆插件（[Joystick](#)），可用于映射键盘 WASD/上下左右按键。
5. 当用户在线数大于资源并发数量时，建议业务引入用户排队系统，提升用户体验。排队逻辑涉及前后台，前端代码可参考 [排队示例](#)，签名逻辑可以根据业务需求决定是否添加，后台部分请查看 [技术接入-排队功能](#)。
6. 测试调试

如正确完成控制台操作，并成功部署业务后台和客户端，您可以通过 Chrome 浏览器（建议）打开 HTML 演示文件，画面加载成功后会显示云渲染画面，您可以通过鼠标拖动 / WASD 等方式与应用互动。如测试过程中遇到问题，请联系我们并提供 [RequestId](#)。

TCGSDK 提供了丰富接口，您可以基于 Demo 开发更多功能，单击可查看 [接口文档](#)。

Android Demo

应用云渲染 Android 入门 Demo 提供了基础的云试玩能力，包含简单示例、虚拟键盘和一些常见 API 的使用，您可以在入门 Demo 的基础上运行您自己的云应用。下述内容将演示如何快速配置 Demo 运行云应用，步骤如下：

1. 单击 [下载](#) Android 端入门 Demo 工程，将工程导入 AndroidStudio 工具。
2. 参考 [云应用接入指南](#) 集成应用。

iOS Demo

应用云渲染 iOS 入门 Demo 提供了基础的云试玩能力，包含简单入门 Demo、集成虚拟按键 Demo 以及在 Demo 中集成了一些常见的 API 使用，您可以在入门 Demo 的基础上运行您自己的云应用。下述内容将演示如何快速配置 Demo 运行云应用，步骤如下：

1. 单击 [下载](#) iOS 端入门 Demo 工程。
2. 参考 [iOS 快速入门](#) 集成应用。

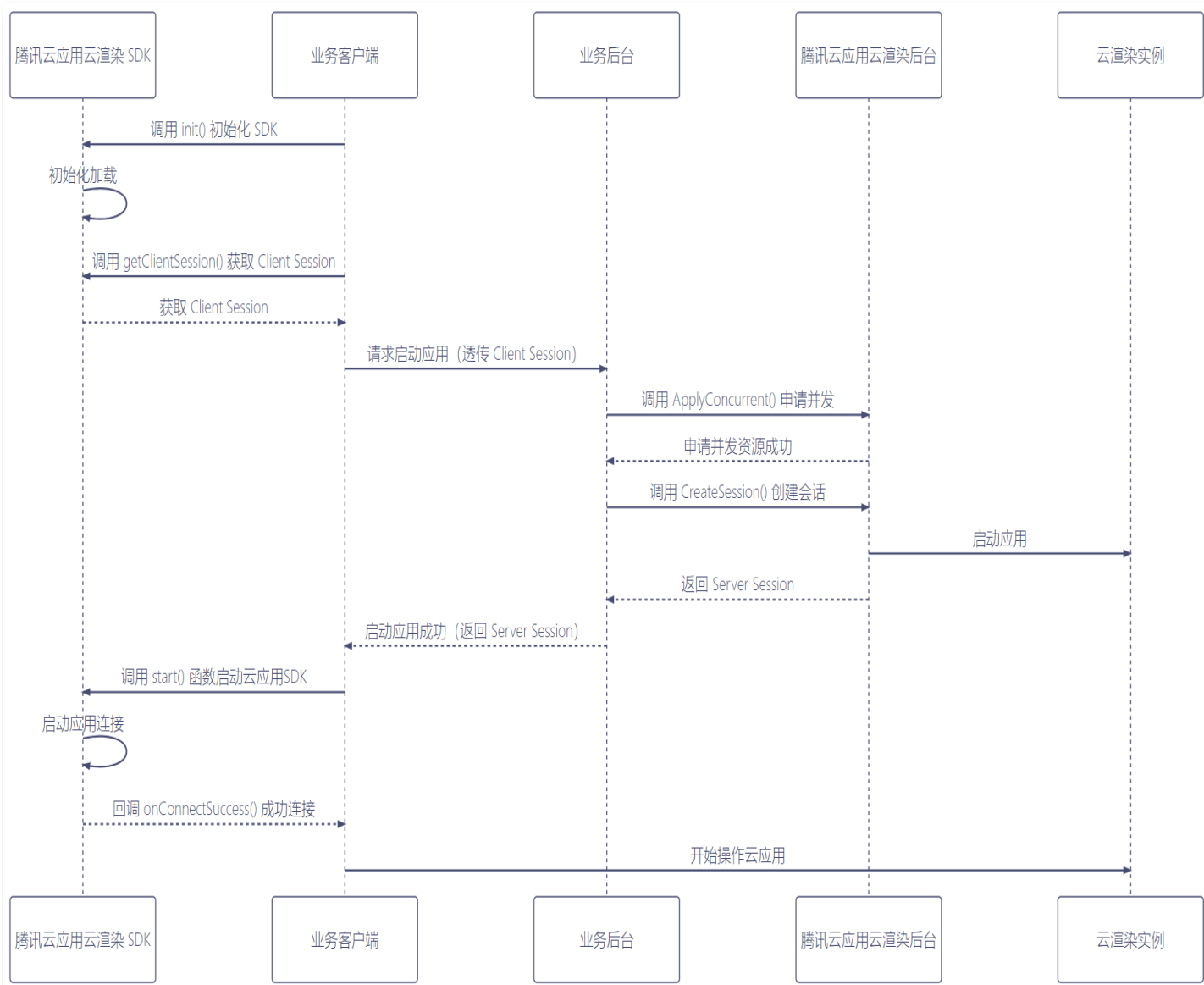
技术接入

启动应用

最近更新时间：2025-06-05 14:40:11

本节主要介绍如何启动云应用。

时序图



其中：

时序角色	说明
------	----

腾讯云应用云渲染 SDK	应用云渲染终端 SDK 包括 JavaScript SDK 、 Android SDK 、 iOS SDK （本章内容以 JavaScript SDK 为例）
业务客户端	业务提供给用户使用的平台，我们提供了 JavaScript Demo 和 Android Demo 供参考
业务后台	业务后台服务，管理用户会话和调用 云 API ，需要部署在本地或者任意服务器。我们提供了 后台 Demo 供参考
腾讯云应用云渲染后台	应用云渲染提供 云 API
云渲染实例	通过控制台购买的云渲染并发实例

步骤说明

1. 业务客户端调用 [TCGSDK.init\(\)](#) 接口完成初始化构建。初始化完成后，再调用 [TCGSDK.getClientSession\(\)](#) 获取 Client 端的 ClientSession。
2. 业务客户端通过传入 UserId 和 ClientSession 等参数向业务后台请求启动应用，其中 UserId 是由业务方自定义的唯一用户标识，在用户排队和重连应用时应保持不变。

⚠ 注意：

同一 UserId 同一时间只能使用一个应用并发，如果用相同 UserId 请求会出现并发抢占。云渲染后台会把目前已连接的并发给到最新一次请求的设备，原来的设备会出现黑屏断连。

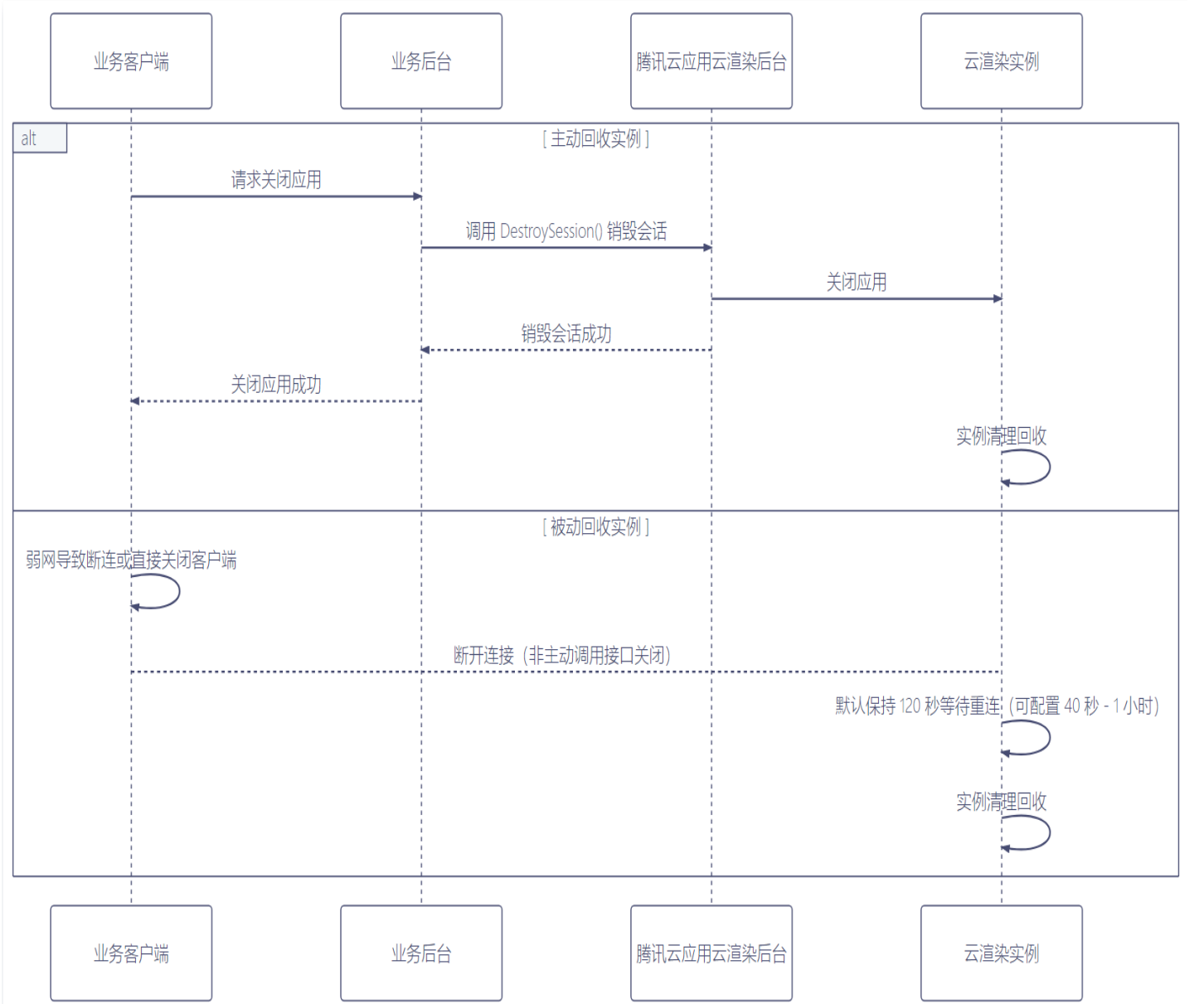
3. 业务后台通过云渲染 API 调用 [ApplyConcurrent\(\)](#) 申请锁定云渲染并发，如果返回没有空闲并发或者其他错误则跳转回 [步骤2](#) 重新请求。
4. 业务后台通过云渲染 API 调用 [CreateSession\(\)](#) 创建会话，将返回的服务端 ServerSession 回传给业务客户端。
5. 业务客户端调用 [TCGSDK.start\(\)](#) 接口启动云渲染应用，SDK 完成应用连接后会触发回调 [onConnectSuccess\(\)](#)，建议包括数据通道等其他功能都在这之后调用创建。

关闭应用

最近更新时间：2024-05-31 16:29:31

本节主要介绍如何对并发资源进行回收，包括主动关闭应用和被动关闭应用的回收流程。建议业务及时释放并发资源，更合理利用并发资源。

时序图



步骤说明

主动回收实例

1. 业务客户端向业务后台主动请求关闭应用。

2. 业务后台通过云渲染 API 调用 `DestroySession()` 销毁会话，业务云端应用将被主动关闭，业务客户端断开与云应用连接。
3. 并发实例进行清理重置回收，等待并发恢复空闲可用状态，详细状态恢复说明参见 [项目高级配置](#)。

被动回收实例

1. 业务客户端在弱网导致断连或直接关闭客户端时，若非主动调用接口 `DestroySession()` 销毁云渲染实例，则云渲染实例会默认保持 120 秒（可配置）的重连时间等待重连。
2. 并发等待时间内客户端未进行重连，业务云端应用将被关闭，若用户需要再次连接，则需要重新 [启动应用](#)。
3. 并发实例进行清理重置回收，等待并发恢复空闲可用状态，详细状态恢复说明参见 [项目高级配置](#)。

❗ 说明：

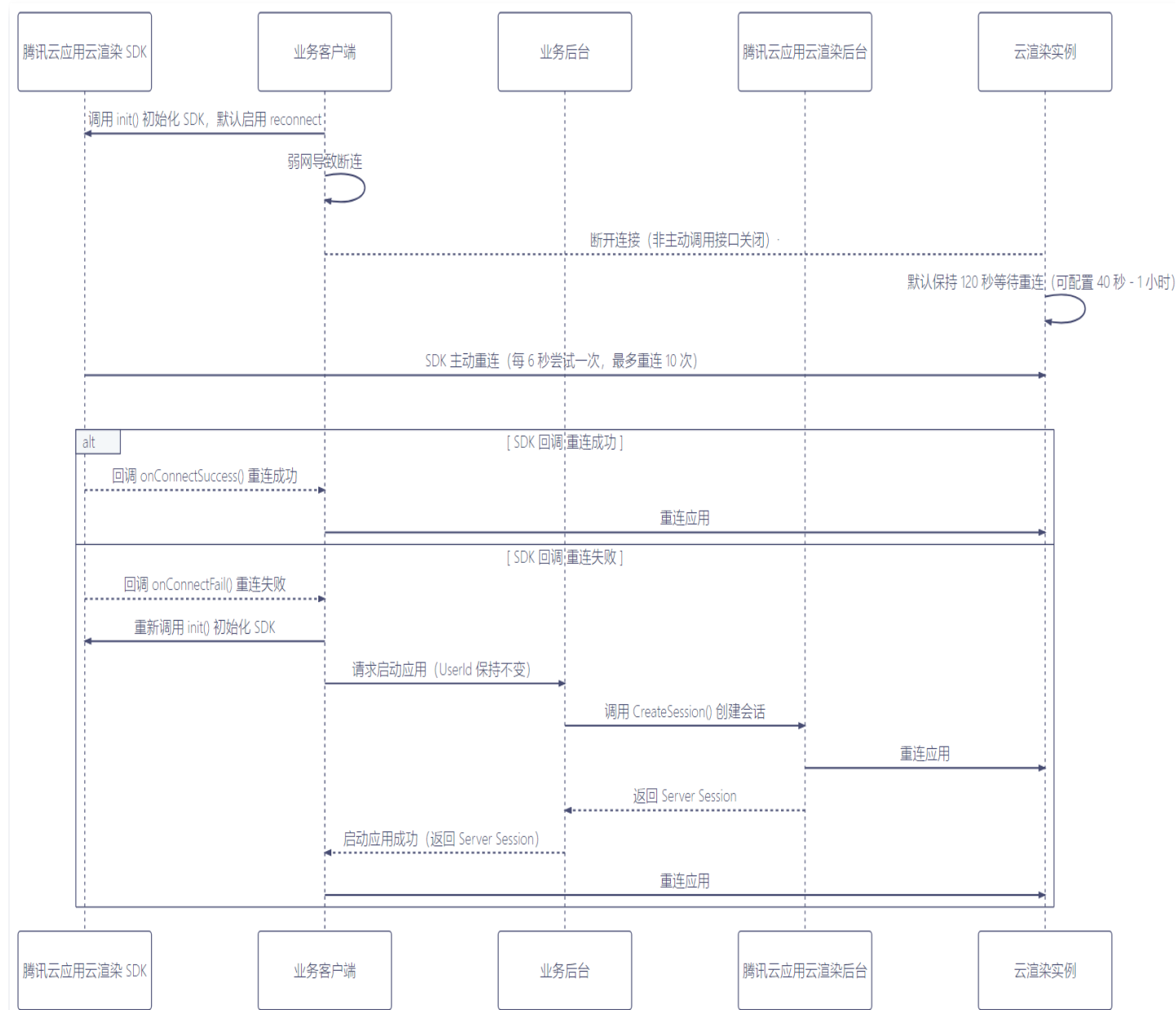
业务可以根据用户连接场景需求，对重连时长进行配置 40 秒 – 1 小时（详细配置说明参见 [项目高级配置](#)），业务也可以自行控制销毁时间，在重连时间内主动调用接口 `DestroySession()` 销毁云渲染实例。

重连应用

最近更新时间：2024-05-31 16:29:31

本节主要介绍如何重连应用。当用户网络断连或者直接关闭客户端而未主动调用关闭应用功能时，业务可以通过 SDK 自动重连或者重新创建会话来实现重连应用功能。

时序图



步骤说明

1. 业务客户端调用 `TCGSDK.init()` 接口完成初始化构建，其中 `reconnect` 参数默认为 `true`，启用自动重连。

2. 业务客户端在弱网导致断连时，若非主动调用接口 `DestroySession()` 销毁云渲染实例，则云渲染实例会默认保持120秒（可配置）的重连时间等待重连。
3. 应用云渲染 SDK 主动发起重连，每6秒尝试一次，最多重连10次，根据连接回调可进行如下处理：
 - 重连成功触发回调 `onConnectSuccess()`，完成重连应用操作。
 - 重连失败触发回调 `onConnectFail()`，当返回 code 大于 0 或者等于 -3 时，需要重新发起启动应用操作，进入 [步骤4](#)。
4. 业务客户端重新调用 `TCGSDK.init()` 接口完成初始化构建，通过传入相同的 UserId 向业务后台请求启动应用，若 UserId 不同则不能实现重连功能。
5. 业务后台通过云渲染 API 调用 `CreateSession()` 重新创建会话，根据会话返回状态进行如下处理：
 - 用户连接的并发超过等待时间，并发已回收，则需要重新 [启动应用](#)。
 - 用户连接的并发还未回收，返回上一次连接的并发资源，并将返回的服务端 `ServerSession` 回传给业务客户端，完成重连应用操作。

❗ **说明：**

当用户网络不稳定导致断连或者关闭客户端时，云渲染实例以及云端直播转推（如有）会保持一段时间，保证用户可以重连应用。业务可以根据用户连接场景需求，对重连时长进行配置40秒 - 1小时（详细配置说明参见 [项目高级配置](#)），业务也可以自行控制销毁时间，在重连时间内主动调用接口 `DestroySession()` 销毁云渲染实例。

无客户端模式

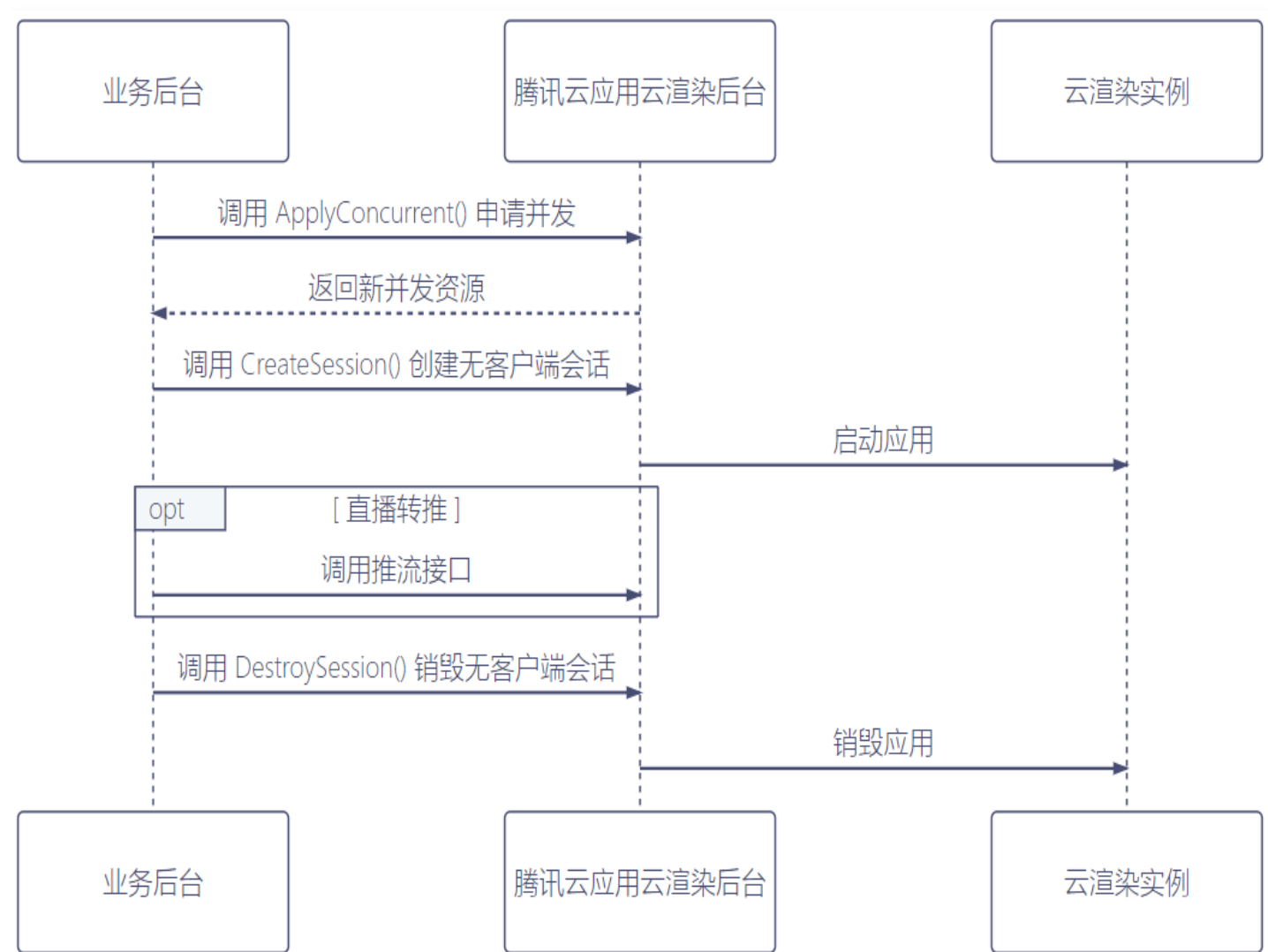
最近更新时间：2024-05-31 16:29:31

无客户端模式是云渲染中一种特殊的连接模式，在该模式下，云渲染实例可以不依赖客户端进行会话创建、保活等操作。本章指南主要介绍了无客户端模式的使用方式。

无客户端实例的创建与销毁

在使用无客户端模式时，可以在无客户端交互的情况下，创建云渲染实例。整个流程以及实例生命周期的维护都由客户业务后台进行控制。

时序图



步骤说明

1. 业务后台通过云渲染 API 调用 `ApplyConcurrent()` 申请云渲染实例并发，云渲染应用后台收到请求后，如有空闲并发会锁定实例。

⚠ 注意:

调用 `ApplyConcurrent()` 时指定的 `UserId` 将和申请到的实例相绑定，后续调用其他接口时，通过设置该 `UserId` 来指定对应无客户端实例。

2. 业务后台申请云渲染实例成功后，通过云渲染 API 调用 `CreateSession()` 接口，并将 `RunMode` 参数设置为 `RunWithoutClient`，以无客户端模式启动该会话。云渲染应用后台收到请求后将在对应实例上启动应用。
3. （可选）在以无客户端模式成功启动会话后，业务后台可以调用直播转推接口，将云上画面进行推流。
4. 当连接生命周期结束时，业务后台通过云渲染 API 调用 `DestroySession()` 接口，指定 `UserId` 销毁对应无客户端模式并发，云渲染应用后台收到请求后，会强制销毁并回收对应实例。

⚠ 注意:

在使用无客户端模式时，云渲染实例的生命周期将完全由业务后台决定，云渲染应用后台**不会自动回收并发**。如不调用销毁会话的接口，云渲染实例将**无期限**的运行下去。在无客户端模式运行过程中，云渲染实例不会进行应用重启，版本更新，环境清理等维护性操作，故在接入无客户端模式时，业务后台一定要考虑实际生命周期，在合适的时机**调用销毁接口**使云端实例强制回收。

适用场景

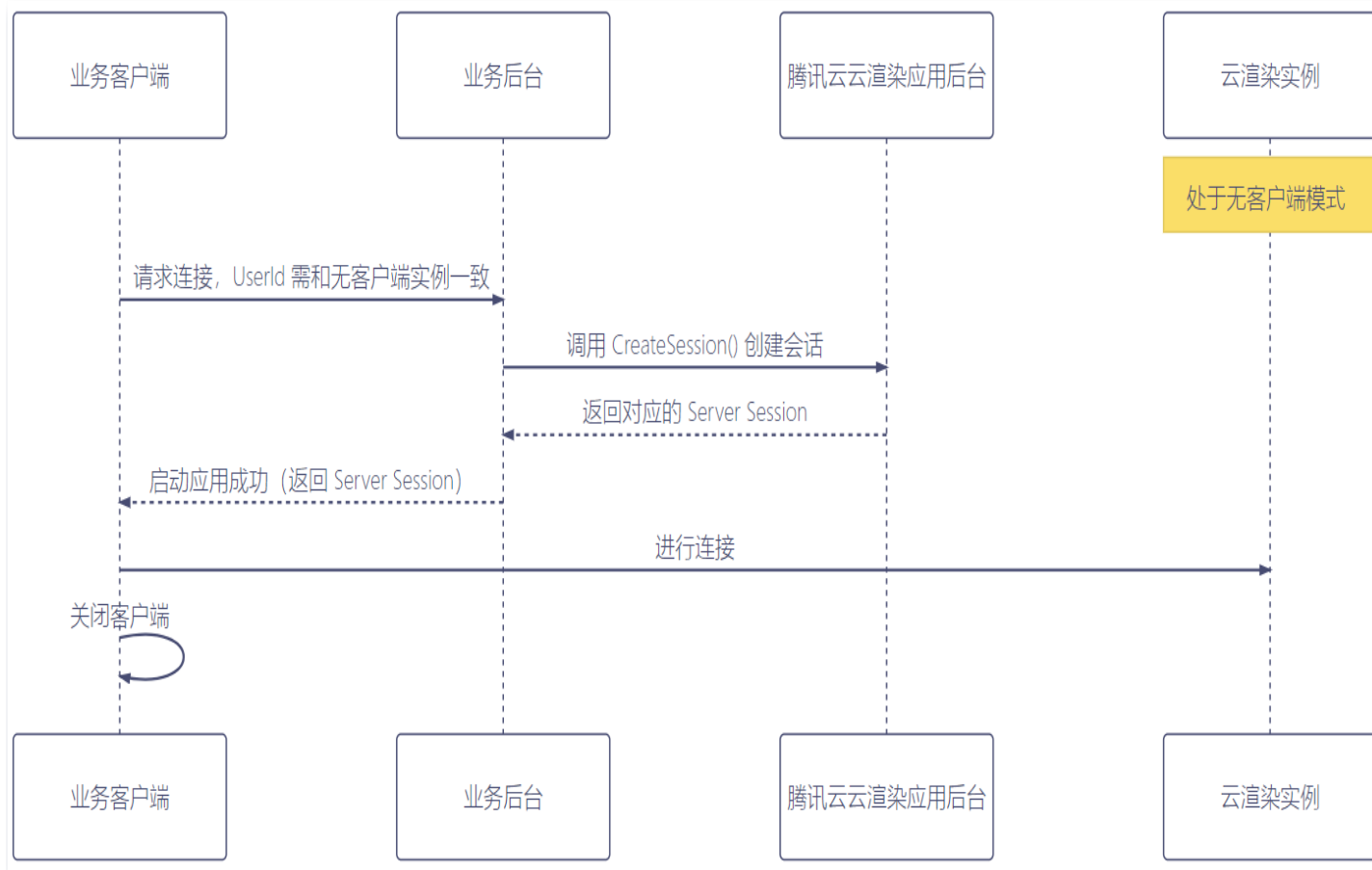
通过创建无客户端实例，再结合 [推流至云直播](#) 或 [推流至指定地址](#) 功能，可以适用于以下场景：

当云上应用**不需要客户端**介入，观众通过直播观看应用画面时，可以使用这种无客户端模式，例如弹幕游戏直播。

客户端连接无客户端实例

在已创建无客户端实例的情况下，如有连接客户端交互的需求，可使用申请无客户端实例并发时使用的 `UserId`，让客户端连接至无客户端模式的云渲染实例进行交互。客户端连接行为不会影响无客户端实例的生命周期，实例的销毁仍依赖客户业务后台调用销毁接口，指定对应 `UserId` 来销毁指定无客户端实例。

时序图



步骤说明

1. 业务客户端首先要**获取 ClientSession**，然后通过指定创建无客户端实例时使用的 UserId 向业务后台请求连接，业务后台收到请求后通过云渲染 API 调用 **CreateSession()** 接口，将客户端信息填入对应参数，云渲染应用后台收到请求后返回 Server Session，业务后台再将对应 Server Session 返回给业务客户端，完成客户端连接。

⚠ 注意：

使用客户端连接到无客户端模式实例时，**CreateSession()** 调用中的 RunMode 将会被忽略，ClientSession 为必填项。

2. 关闭客户端的行为不会销毁云渲染实例，后续可以继续用对应 UserId 再次进行连接，且多次连接上的是同一个实例。要销毁对应实例仍依赖业务后台调用 **DestroySession()** 接口，传入对应 UserId 进行实例销毁。

⚠ 注意：

无客户端模式下，云渲染实例不会进行应用重启、环境清理的操作，客户端连接到对应实例上时，将看到上一次连接退出时的应用状态。

适用场景

使用无客户端模式，并结合客户端连接实例，可以适用于以下场景：

当希望自定义控制云上应用时，可以使用这种无客户端模式，后续在进行一些维护工作时集中进行连接重建。

容灾措施

使用无客户端模式，通常应用在实例上需要长时间运行，可能还需要长时间推流，所以服务的可用性至关重要。对此，我们采取了以下容灾措施来提高服务可用性：

- 1、当应用异常退出时，实例服务会将应用重新启动。
- 2、当实例异常宕机时，后台服务将选择一台新实例并使用相同的请求参数启动应用；如果宕机实例有推流的话，新实例将使用相同的推流地址启动推流。

注意事项

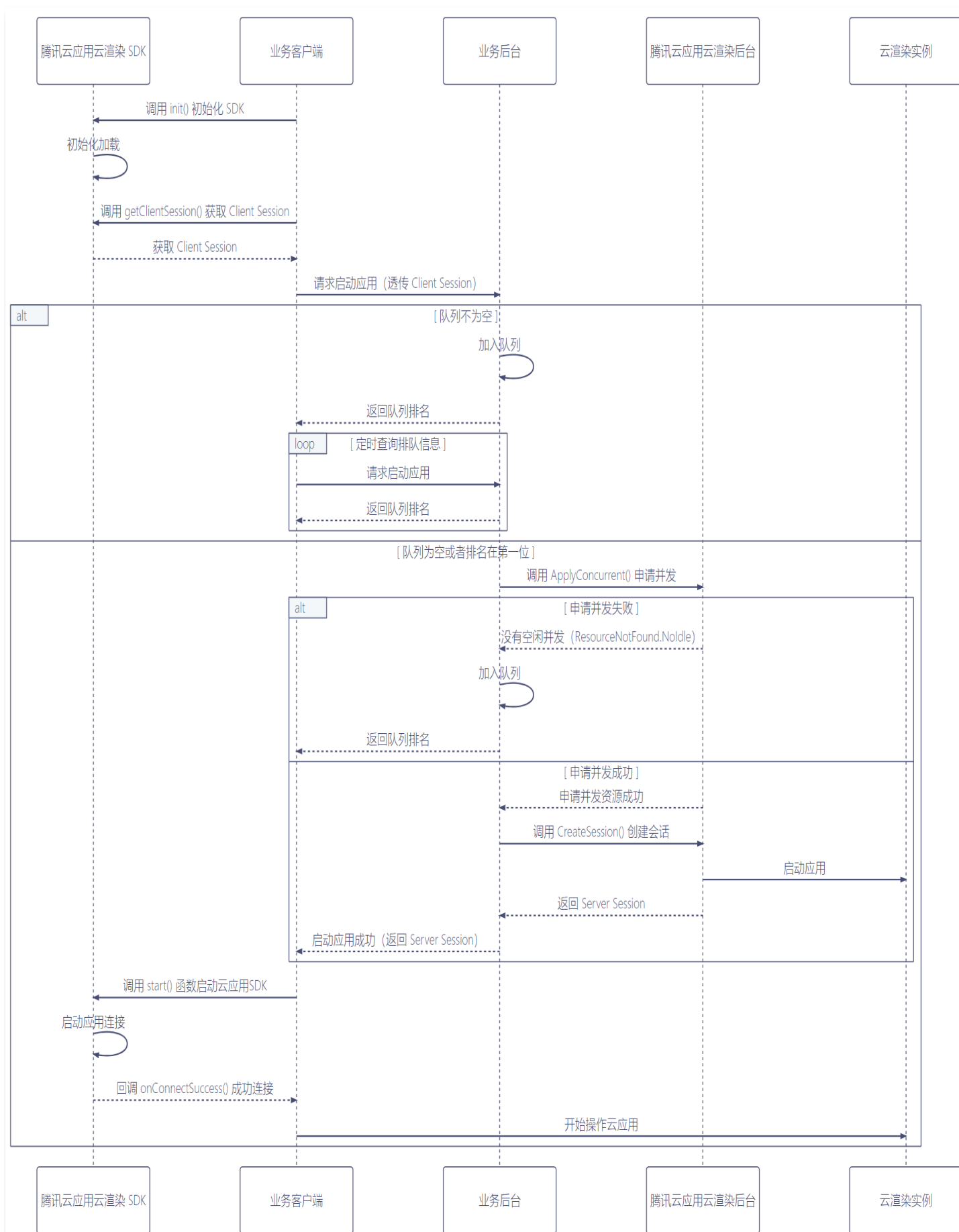
使用无客户端模式时，并发的生命周期完全由客户侧业务后台进行掌控，因此需要保存 **UserId** 列表。当调用 **CreateSession()** 成功后，将对应的 **UserId** 加入到列表中，表示该 **UserId** 创建了一个实例；当调用 **DestroySession()** 成功后，从列表中删除对应的 **UserId**，表示对应的实例已被销毁。**UserId** 应该是列表的唯一 **Key**，因为一个 **UserId** 只能创建一个实例。

排队功能

最近更新时间：2023-12-29 12:05:21

本节主要介绍如何引入用户排队系统。当用户在线数大于资源并发数量时，建议业务引入用户排队系统，提升用户体验。

时序图



步骤说明

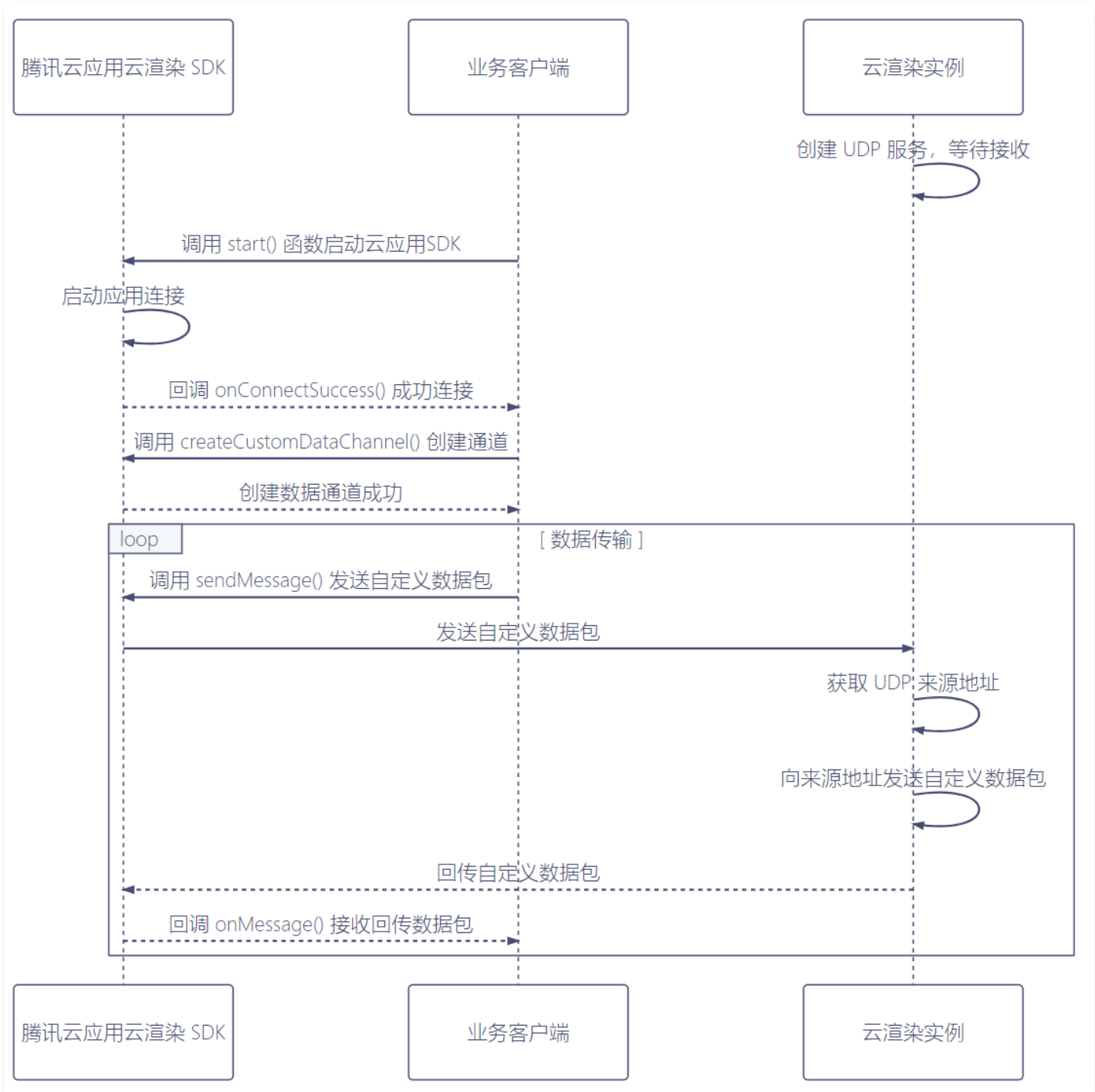
1. 业务客户端调用 `TCGSDK.init()` 接口完成初始化构建。初始化完成后，再调用 `TCGSDK.getClientSession()` 获取 Client 端的 `ClientSession`。
2. 业务客户端通过传入 `UserId` 和 `ClientSession` 等参数向业务后台请求启动应用，其中 `UserId` 是由业务方自定义的唯一用户标识，在用户排队和重连应用时应保持不变。业务后台需要根据当前排队状况进行如下处理：
 - 队列不为空时，则将用户加入队列并返回当前队列排名信息。业务客户端收到排队信息后定时发起请求，重复 [步骤2](#) 直到启动应用成功。
 - 如果队列为空或者用户排在第一位跳转到 [步骤3](#)。
3. 业务后台通过云渲染 API 调用 `ApplyConcurrent()` 申请锁定云渲染并发，根据返回结果进行如下处理：
 - 没有空闲并发或者其他错误时，则将用户加入队列并返回当前队列排名信息，跳转回 [步骤2](#) 重新请求。
 - 申请并发成功则跳转到 [步骤4](#)。
4. 业务后台通过云渲染 API 调用 `CreateSession()` 创建会话，将返回的服务端 `ServerSession` 回传给业务客户端。
5. 业务客户端调用 `TCGSDK.start()` 接口启动云渲染应用，SDK 完成应用连接后会触发回调 `onConnectSuccess()`，建议包括数据通道等其他功能都在这之后调用创建。

数据通道

最近更新时间：2024-06-27 12:05:21

本节主要介绍如何使用数据通道与云端运行的应用直接进行通信。当业务需要建立客户端与云端应用通信，传输相关启动参数、指令、消息或其他数据时，例如透传微信名和头像信息至云端应用，可以使用数据通道能力。

时序图



步骤说明

1. 业务云端应用创建 UDP 服务，监听一个本地 UDP 端口 (`localhost 127.0.0.1` 端口范围建议为 10000 – 20000)，并开始等待接收 UDP 包。
2. 业务客户端调用 `TCGSDK.start()` 接口启动云渲染应用，SDK 完成应用连接后会触发回调 `onConnectSuccess()`，建议业务客户端在这之后创建数据通道。
3. 业务客户端调用 `TCGSDK.createCustomDataChannel()` 接口创建透传通道，接口里的目标端口参数应为 [步骤1](#) 中云端应用监听的端口，如果创建失败则重复本步骤直到创建成功。
4. 数据通道创建成功后，业务客户端可以调用 `sendMessage()` 发送业务自定义的数据包，云端应用 UDP 服务收到请求，解析出 UDP 来源地址。
5. 云端应用向 [步骤4](#) 拿到的 UDP 来源地址发送自定义数据包，数据包将通过创建好的数据通道回传，业务客户端可以在回调 `onMessage()` 接口中处理回传的数据包。

示例代码

业务前端（以 JS SDK 为例）

```
let timer = null;

const { sendMessage, code } = await TCGSDK.createCustomDataChannel({
  destPort: xxxx, // destPort 端口范围建议为 10000-20000
  onMessage: (res) => {
    console.log('CustomDataChannel onMessage', res);

    // 收到该回调后，表示云端应用已成功拉起并成功向web 端发送数据，可以
    clearInterval，之后正常发送数据
    // clearInterval(timer);
  },
});

// code 为 0 表示 web 端与云端数据通道已建立成功，但可能此时云端应用并未完全拉起（应用不能收发数据），web 端可以通过 setTimeout/setInterval/轮询等形式发送初始数据，直到云端应用正常回包。
if (code === 0) {
  // 发送自定义数据
  sendMessage('test');

  // timer = setInterval(() => {
  //   sendMessage('test');
  // }, 5000);
} else {
```

```
// 考虑重新创建数据通道  
}
```

业务云端应用（以 C/C++ 为例）

```
int main() {  
    int udp_socket_fd = socket(AF_INET, SOCK_DGRAM, 0);  
    if (udp_socket_fd == -1) {  
        printf("socket failed!\n");  
        return -1;  
    }  
  
    // 设置目的 IP 地址  
    struct sockaddr_in bind_addr = { 0 };  
    bind_addr.sin_family = AF_INET;  
    bind_addr.sin_port = htons(XXXX); // htons(XXXX) 中的端口范围建议为  
10000-20000  
    bind_addr.sin_addr.s_addr = inet_addr("0.0.0.0"); // 绑定 IP  
  
    // 绑定端口  
    int ret = bind(udp_socket_fd, (struct sockaddr *)&bind_addr,  
sizeof(bind_addr));  
    if (ret < 0) {  
        perror("bind fail:");  
        close(udp_socket_fd);  
        return -1;  
    }  
  
    // 开始等待客户端消息  
    struct sockaddr_in upstream_addr = { 0 }; // 用来存放云渲染代理的地址  
    int len = sizeof(upstream_addr);  
    char buf[1024] = { 0 }; // 接收消息缓冲区  
    while (true) {  
        ret = recvfrom(udp_socket_fd, buf, sizeof(buf), 0, (struct  
sockaddr *)&upstream_addr, &len);  
        if (ret == -1) {  
            break;  
        }  
        // buf 为前端发来的消息 "test"  
        // 后续可以用 upstream_addr 回传消息给前端  
        const char* response = "response";  
    }  
}
```

```
sendto(udp_socket_fd, response, strlen(response), 0, (struct
sockaddr *)&upstream_addr, sizeof(upstream_addr));
}
return 0;
}
```

常见问题

数据通道创建成功，业务前端发送数据成功，但没有收到云端应用回复的数据？

- **TCGSDK.createCustomDataChannel()** 接口调用成功表示业务前端与云端数据通道已建立成功，但可能此时云端应用并未完全拉起，业务前端可以通过 timeout/interval/轮询等形式发送自定义数据，确保云端应用成功拉起后正常接收到业务前端发送的自定义数据。只要数据通道创建成功，默认数据能发送成功。
- 在onMessage 回调中确定收到云端应用数据，可以取消轮询，之后正常发送数据。

可以收发的数据类型有哪些？

sendMessage() 接口支持 string | ArrayBuffer 数据类型。

数据包传输大小是否有限制？

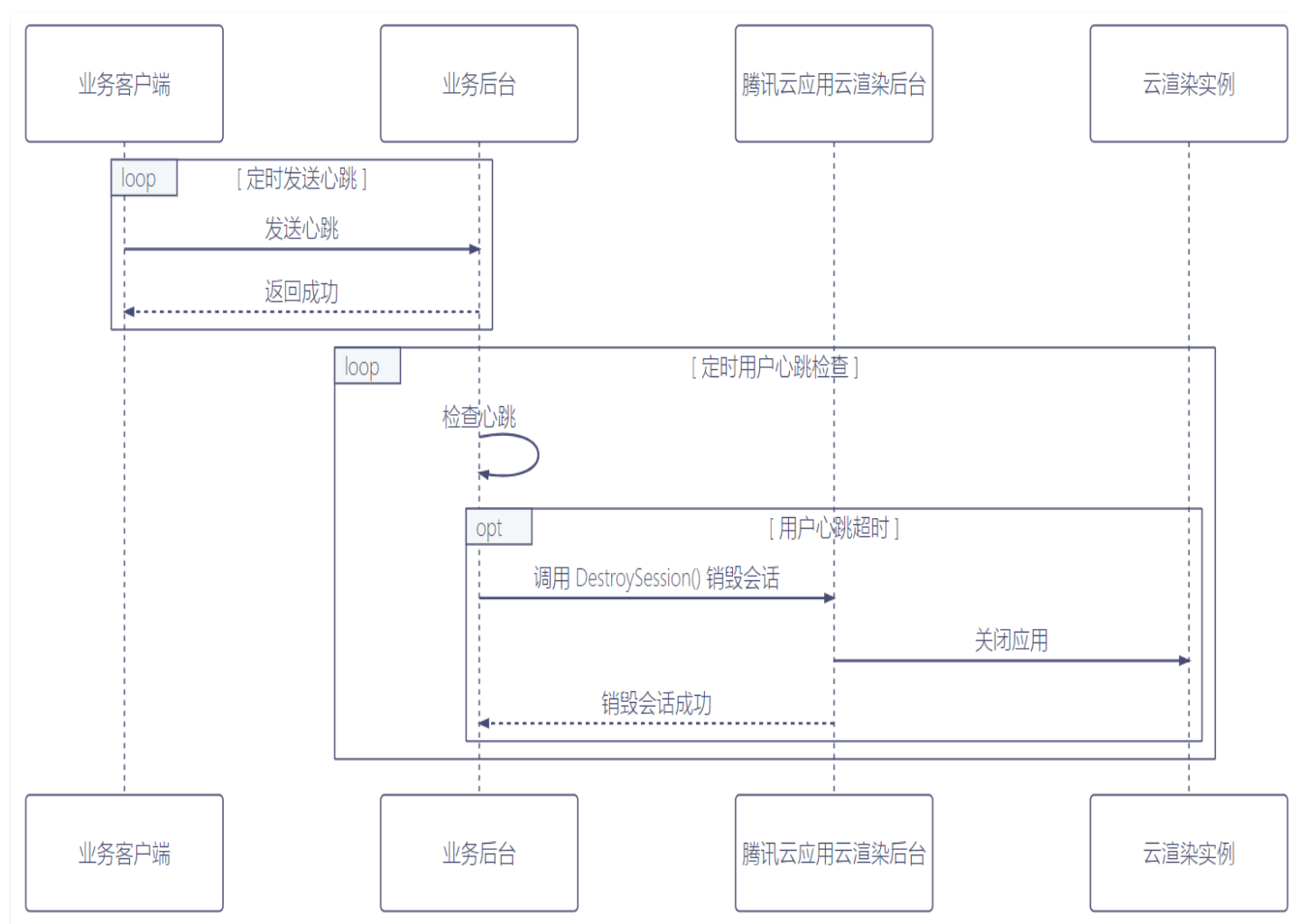
服务对传输数据包大小没有限制，但是需要注意 UDP 最大包长是 64KB，建议包大小应该小于 MTU 1500。如果包体过大，建议通过分包形式传输，如果太多数据上行容易导致拥塞影响体验。

心跳连接

最近更新时间：2023-12-29 12:05:21

本节主要介绍如何维持服务心跳连接。建议业务通过心跳来感知用户连接情况，可以收集管理用户连接时长等数据，当客户端异常退出之后业务后台可以快速回收资源，更合理利用并发资源。

时序图



步骤说明

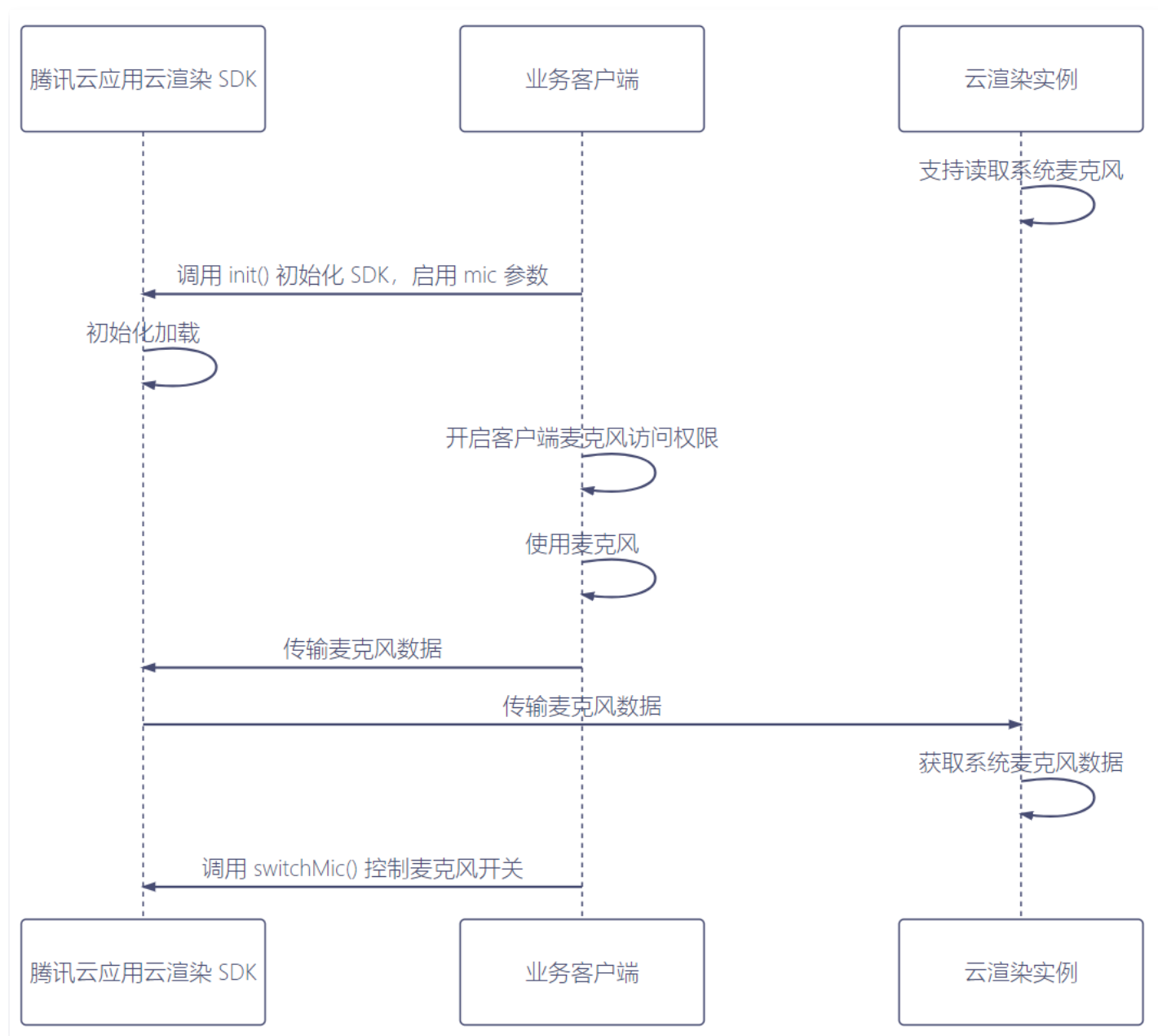
1. 业务客户端定时向业务后台发送心跳上报，业务后台根据心跳上报维护用户连接状态。
2. 业务后台定时检查用户连接状态，如果有用户心跳超时，业务后台通过云渲染 API 调用 [DestroySession\(\)](#) 销毁会话，业务云端应用将被主动关闭，业务客户端断开与云应用连接。

启用麦克风

最近更新时间：2023-12-29 12:05:21

本节主要介绍业务如何启用麦克风。当业务需要通过麦克风传输语音时，可以通过 SDK 启用麦克风功能。

时序图



步骤说明

1. 业务云端应用需要支持获取系统麦克风。
2. 业务客户端调用 `TCGSDK.init()` 接口，并设置 `mic` 参数为 `true`，完成初始化构建。
3. 业务客户端需要开启麦克风访问权限。

⚠ 注意:

业务客户端以浏览器为例，首先需要系统开启浏览器的麦克风权限，然后需要浏览器开启访问页面的麦克风权限，这样业务客户端才能完成开启麦克风访问权限。

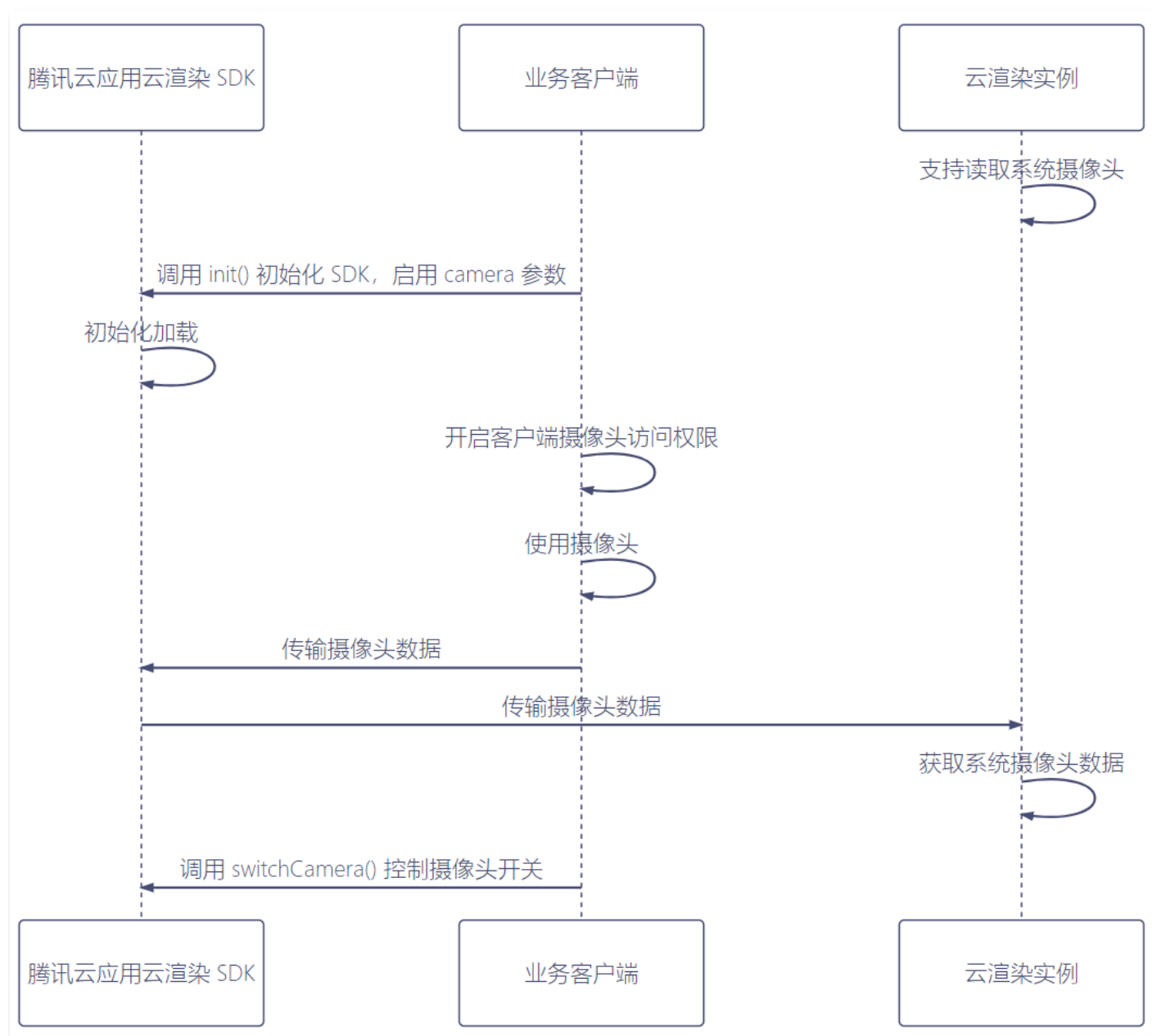
4. 业务客户端使用麦克风，数据通过 SDK 传输给业务云端应用。
5. 连接过程中如果业务需要控制麦克风打开或者关闭，业务客户端可以调用 `TCGSDK.switchMic()` 控制麦克风开关。

启用摄像头

最近更新時間：2023-12-29 12:05:21

本节主要介绍业务如何启用摄像头。当业务需要摄像头传输视频画面时，可以通过 SDK 启用摄像头功能。

时序图



步骤说明

1. 业务云端应用需要支持获取系统摄像头。
2. 业务客户端调用 `TCGSDK.init()` 接口，并设置 `camera` 参数为 `true`，完成初始化构建。
3. 业务客户端需要开启摄像头访问权限。

⚠ 注意:

业务客户端以浏览器为例，首先需要系统开启浏览器的摄像头权限，然后需要浏览器开启访问页面的摄像头权限，这样业务客户端才能完成开启摄像头访问权限。

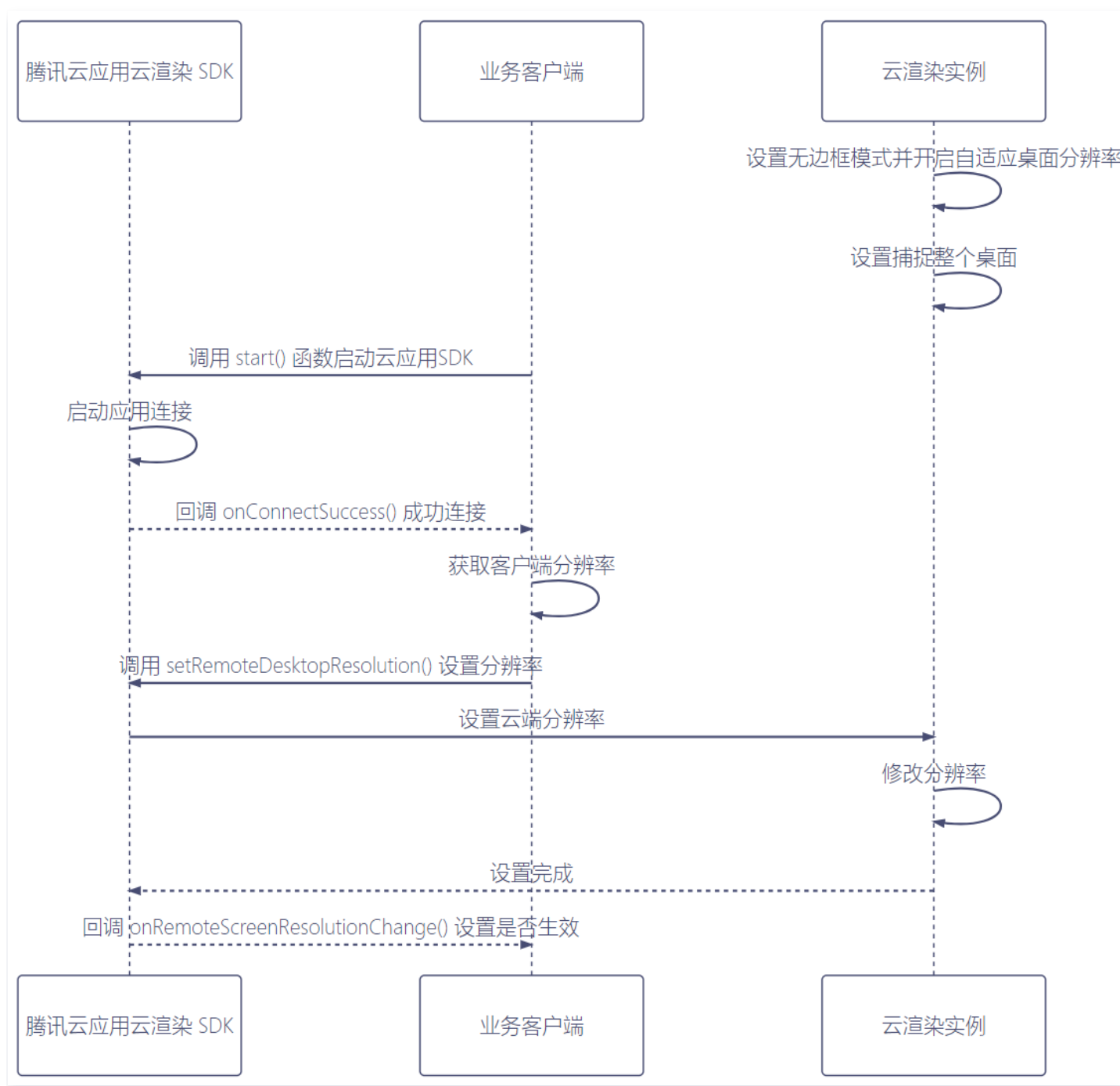
4. 业务客户端使用摄像头，数据通过 SDK 传输给业务云端应用。
5. 连接过程中如果业务需要控制摄像头打开或者关闭，业务客户端可以调用 `TCGSDK.switchCamera()` 控制摄像头开关。

自适应分辨率

最近更新时间：2024-02-01 21:47:31

本节主要介绍云端应用如何支持自适应分辨率。当业务需要适配不同用户客户端的分辨率，实现画面全屏无黑边时，可以使用自适应分辨率的能力。

时序图



步骤说明

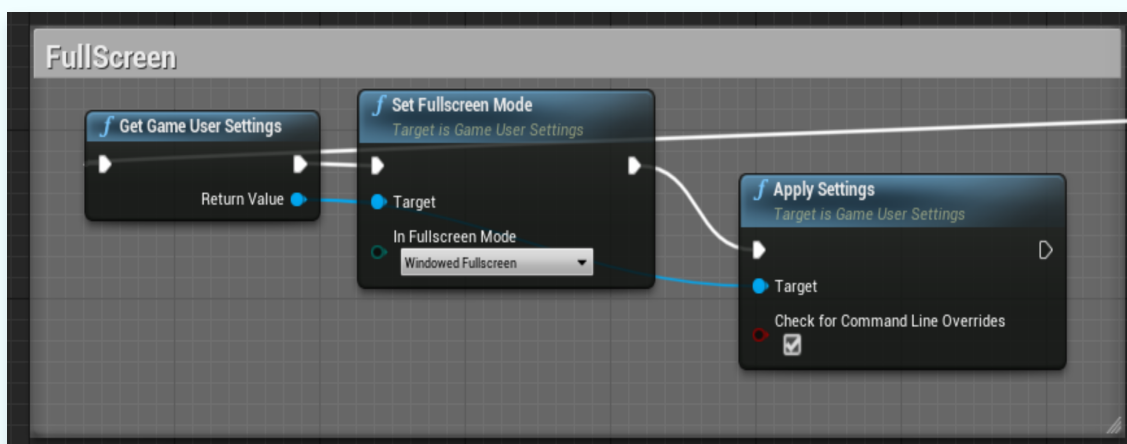
推荐方式

1. 前置条件:

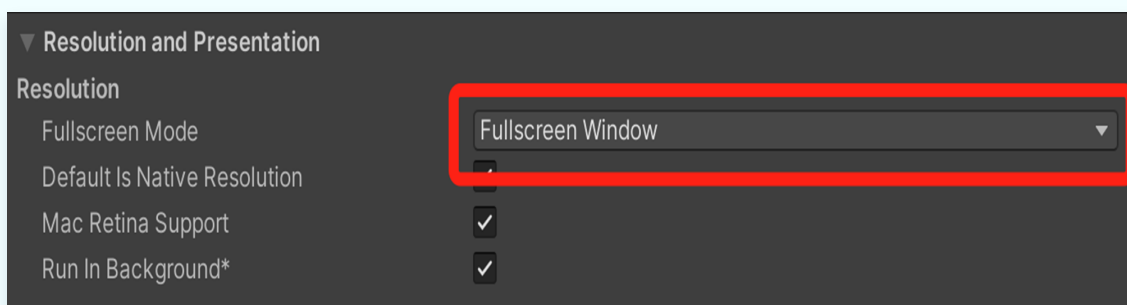
1.1 在上传应用至云渲染平台前，将应用设置为无边框全屏模式并开启自适应桌面分辨率。

⚠ 注意:

- 这步操作需要在应用层面完成。
 - UE 应用可以参考下面一段蓝图加到 Map 的 BeginPlay 里面开启自适应桌面分辨率。其中 In Fullscreen Mode 应该配置为 Windowed Fullscreen（窗口化全屏）。



- Unity 应用可以参考下图配置为全屏模式。

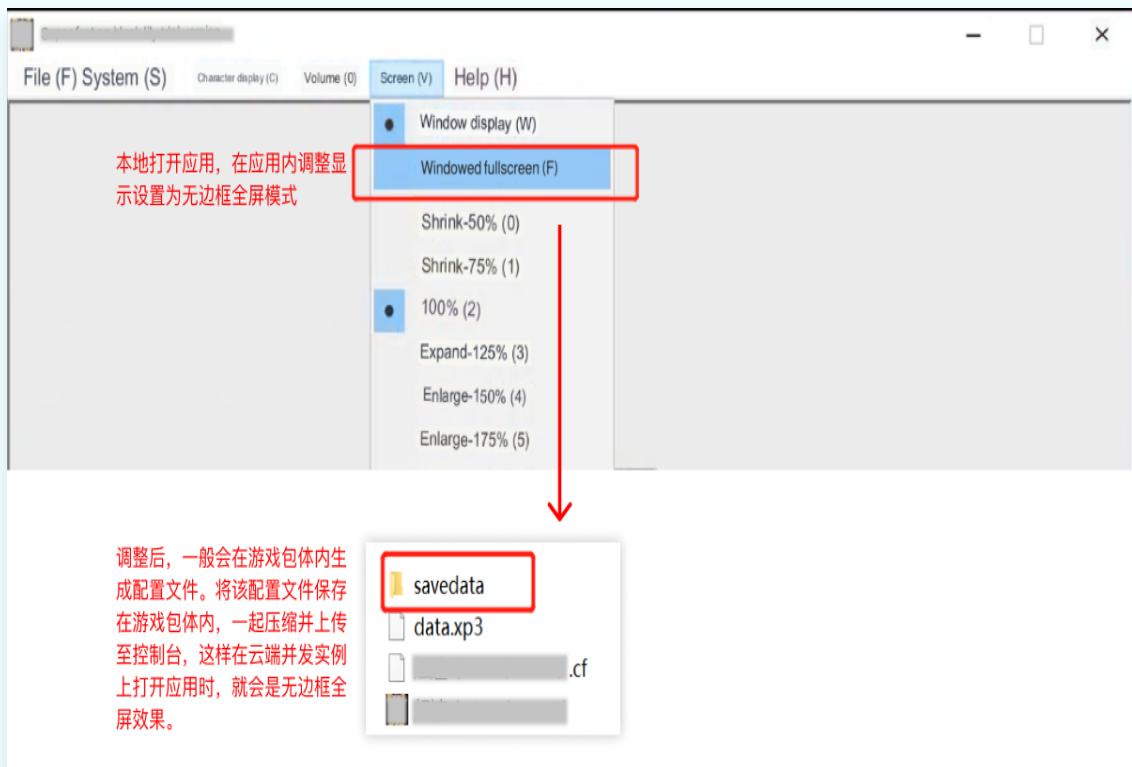


或者增加应用启动参数 `-screen-fullscreen 1`，Unity 应用参数参考 [文档](#)。

`-screen-fullscreen`

Override the default full-screen state. This must be 0 or 1.

- 有些应用可以在设置中调整显示模式，将调整后生成的配置文件一同打包上传至控制台即可。



- **应用独占全屏不属于无边框模式。**云渲染的自适应分辨率功能是通过修改云端桌面分辨率来实现的。然而，应用独占全屏时，显示器分辨率是由应用控制的，此时强行修改桌面分辨率可能导致应用崩溃。
- **如何区分无边框全屏和独占全屏？**无边框全屏应用按 Alt+Tab 切换窗口不会导致显示器闪烁，独占全屏应用切换会有闪烁的现象。

1.2 在控制台进行 [应用配置](#) 时，将画面捕捉模式设置为**捕捉整个桌面**。

2. 业务客户端调用 `TCGSDK.start()` 接口启动云渲染应用，SDK 完成应用连接后会触发回调 `onConnectSuccess()`，建议业务客户端在这之后设置分辨率。
3. 业务客户端调用 `TCGSDK.getPageSize()` 或者其他方式获取客户端分辨率。
4. 业务客户端调用 `TCGSDK.setRemoteDesktopResolution()` 接口设置云端分辨率。
5. 云端并发实例收到设置分辨率请求后，会自动修改云端桌面分辨率，云端应用根据桌面分辨率变化自适应修改。
6. 业务客户端可以在回调 `onRemoteScreenResolutionChange()` 接口中获取分辨率设置是否生效。

若应用仅为非全屏窗口，如何适配不同用户客户端的分辨率？（不推荐）

如您的应用只有非全屏窗口模式，无法调整为无边框全屏模式，可以在控制台 [应用配置](#) 中将画面捕捉模式设置为**仅捕捉应用窗口**。您需要 [根据文档指引](#)，填写正确的应用窗口名称和类名。当开发过程中未自定义窗口标题时，

Demo.exe 启动后的窗口名称一般为 Demo；如您的应用是 UE 应用，类名一般为 UnrealWindow。

后续步骤：

1. 业务客户端调用 `TCGSDK.start()` 接口启动云渲染应用，SDK 完成应用连接后会触发回调 `onConnectSuccess()`，建议业务客户端在这之后设置分辨率。
2. 业务客户端调用 `TCGSDK.getPageSize()` 或者其他方式获取客户端分辨率。
3. 业务客户端通过 [数据通道](#) 或其他自定义方式通知云端应用修改分辨率。
4. 云端应用收到设置分辨率请求后，**修改自身分辨率**。（该逻辑需要应用本身支持）
5. 由于应用分辨率在变化，为避免应用分辨率超过云端桌面分辨率所导致的画面不全、部分画面无法响应等问题，需要确保云端桌面分辨率始终大于应用分辨率。为此，您可以：
 - 方式一：在控制台 **项目配置 > 高级配置** 中，将云端桌面默认分辨率调整到足够的大小，如 2500x2500。
 - 方式二：同步调用 `TCGSDK.setRemoteDesktopResolution()` 接口修改云端桌面分辨率。业务客户端可以在回调 `onRemoteScreenResolutionChange()` 接口中获取桌面分辨率设置是否生效。

⚠ 注意：

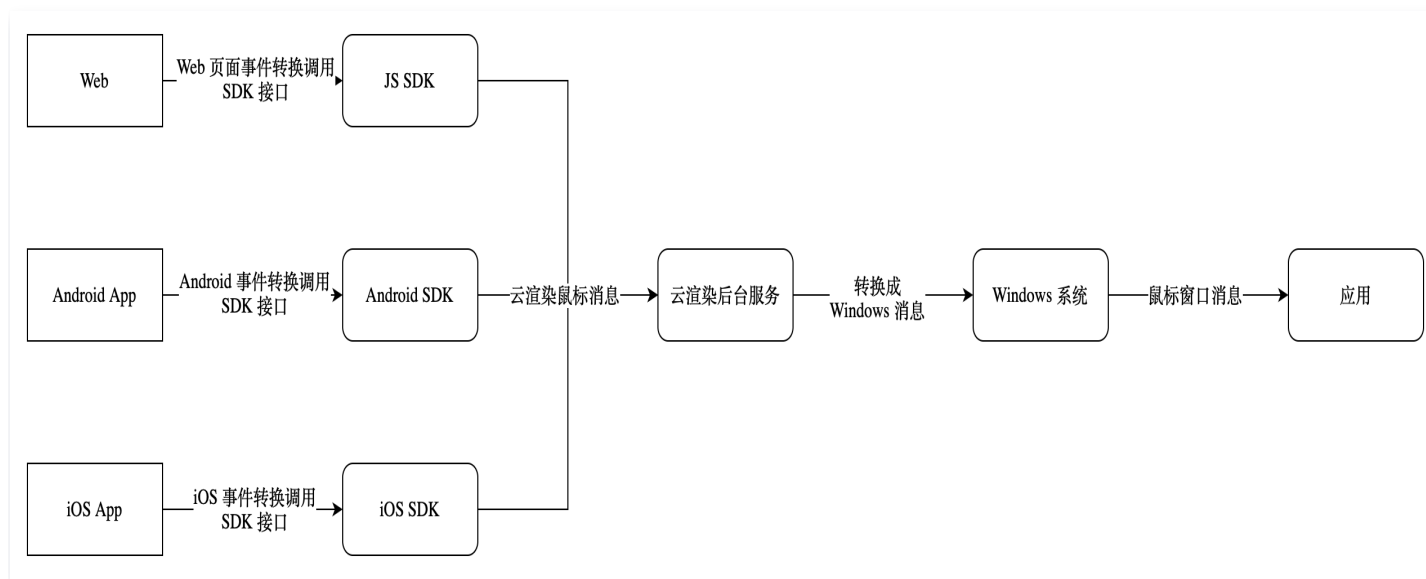
以上方式较为复杂，若配置不当可能导致画面不全、部分画面无法响应、客户端画面分辨率失常等问题，因此建议您选择 **推荐方式**，将应用设置为无边框全屏模式并开启自适应桌面分辨率。

鼠标、键盘、触摸交互处理

最近更新时间：2024-06-14 14:38:13

本节主要介绍鼠标、键盘、触摸的交互操作处理方法。

鼠标



从上图可以看出，云渲染交互中鼠标消息的转换流程。其中，应用软件是业务侧在控制台自助上传的应用，它运行在 Windows 系统上，响应 Windows 的鼠标窗口消息。业务侧接入 SDK 发送鼠标消息时，需要将 SDK 运行所在平台的消息转换为鼠标消息来操作应用。

一般来说，各个平台处理情况如下：

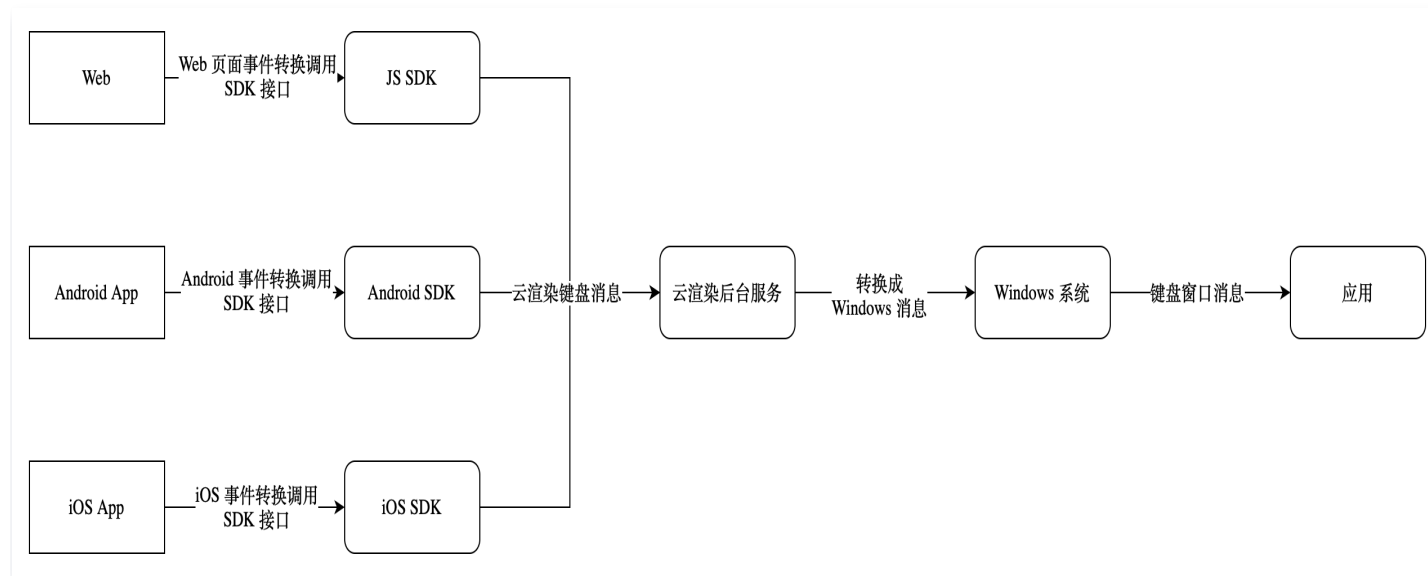
- 在 Android 和 iOS 平台是将系统的触摸消息转换成云渲染鼠标消息。
- 在 Web 页面中则根据页面运行的系统来决定，当 Web 页面运行在 Windows/macOS 的浏览器中时将鼠标消息转换为云渲染鼠标消息，当 Web 页面运行在 Android/iOS 的浏览器中时将触摸消息转换为云渲染鼠标消息。

其中，触摸消息转换为云渲染鼠标消息中，通常是将单指点击转换为云渲染鼠标点击，单指滑动转换为云渲染鼠标移动等。

因此，业务侧需要根据使用的 SDK 及运行的平台做相应的适配，具体请参见：

- Web 页面（小程序）参见文档：[onTouchEvent](#)，[mousemove](#)，[sendMouseEvent](#)。
- Android App 参见文档：[Android 外设交互处理](#)。
- iOS App 参见文档：[iOS 外设交互处理](#)。

键盘



从上图可以看出，云渲染交互中键盘消息的转换流程。其中，应用软件是业务侧在控制台自助上传的应用，它运行在 Windows 系统上，响应 Windows 的键盘窗口消息。业务侧接入 SDK 发送键盘消息时，需要将 SDK 运行所在平台的消息转换为键盘消息来操作应用。

一般来说，各个平台处理情况如下：

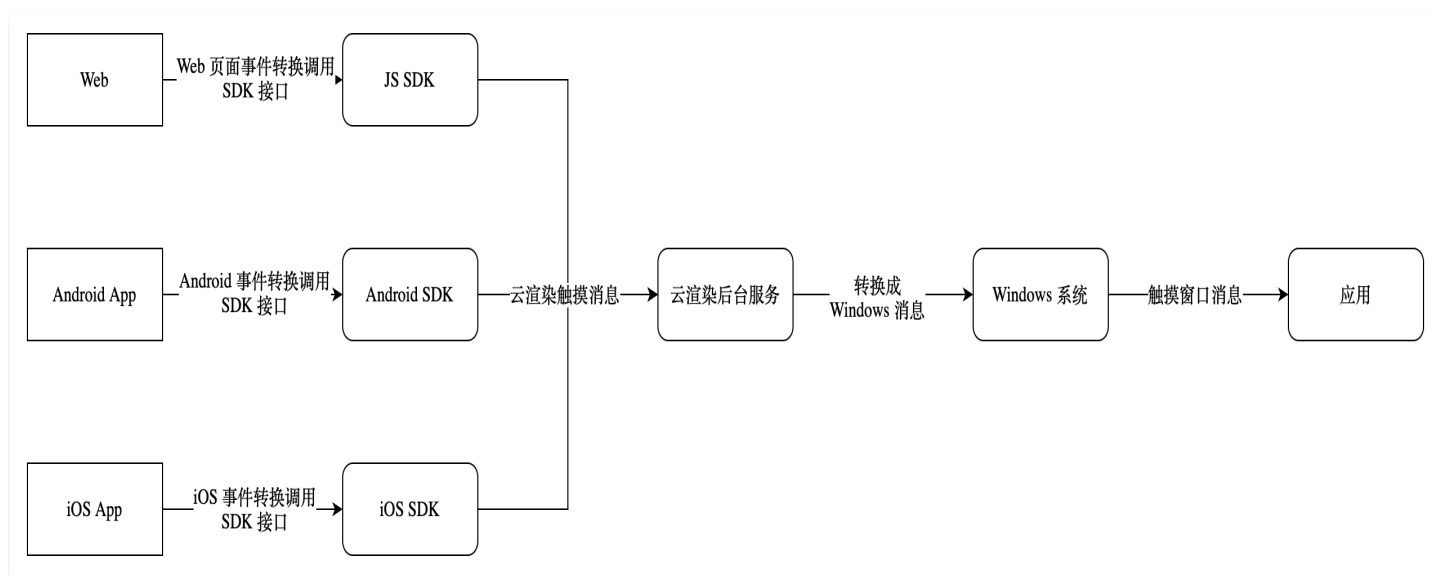
- 在 Android 和 iOS 平台是将系统的触摸消息或者软键盘消息转换为云渲染键盘消息。
- 在 Web 页面中则根据页面运行的系统来决定，当 Web 页面运行在 Windows/macOS 的浏览器中时将键盘消息转换为云渲染键盘消息，当 Web 页面运行在 Android/iOS 的浏览器中时将触摸消息或者软键盘消息转换为云渲染键盘消息。

其中，触摸消息或软键盘消息转换为云渲染键盘消息中，通常是将触摸方向盘操作转换为云渲染键盘的 WASD 消息，将软键盘消息直接转换为云渲染键盘按键消息等。

因此，业务侧需要根据使用的 SDK 及运行的平台做相应的适配，具体请参见：

- Web 页面（小程序）参见文档：[sendKeyboardEvent](#)，[Joystick](#)。
- Android App 参见文档：[Android 外设交互处理](#)。
- iOS App 参见文档：[iOS 外设交互处理](#)。

触摸



从上图可以看出，云渲染交互中触摸消息的转换流程。其中，应用软件是业务侧在控制台自助上传的应用，它运行在 Windows 系统上，响应 Windows 的触摸窗口消息。业务侧接入 SDK 发送触摸消息时，需要将 SDK 运行所在平台的消息转换为触摸消息来操作应用。

一般来说，各个平台处理情况如下：

- 在 Android 和 iOS 平台将系统的触摸消息转换为云渲染触摸消息。
- 在 Web 页面仅运行在 Android/iOS 的浏览器中时将触摸消息转换为云渲染触摸消息。

只有当应用软件支持 Windows 的触摸窗口消息处理，并且应用软件在物理的 Windows 触摸屏设备上测试正常时，才建议业务侧可以使用 SDK 将所在平台的触摸消息转换为云渲染触摸消息。

⚠ 注意：

不建议使用 UE、Unity 引擎的鼠标转触摸来测试应用软件对触摸消息的处理是否正常，因为触摸行为会在触摸屏设备上表现异常。

因此，业务侧需要根据使用的 SDK 及运行的平台做相应的适配，具体请参见：

- Web 页面（小程序）参见文档：[init.clientInteractMode](#)，[setClientInteractMode](#)。
- Android App 参见文档：[Android 外设交互处理](#)。
- iOS App 参见文档：[iOS 外设交互处理](#)。