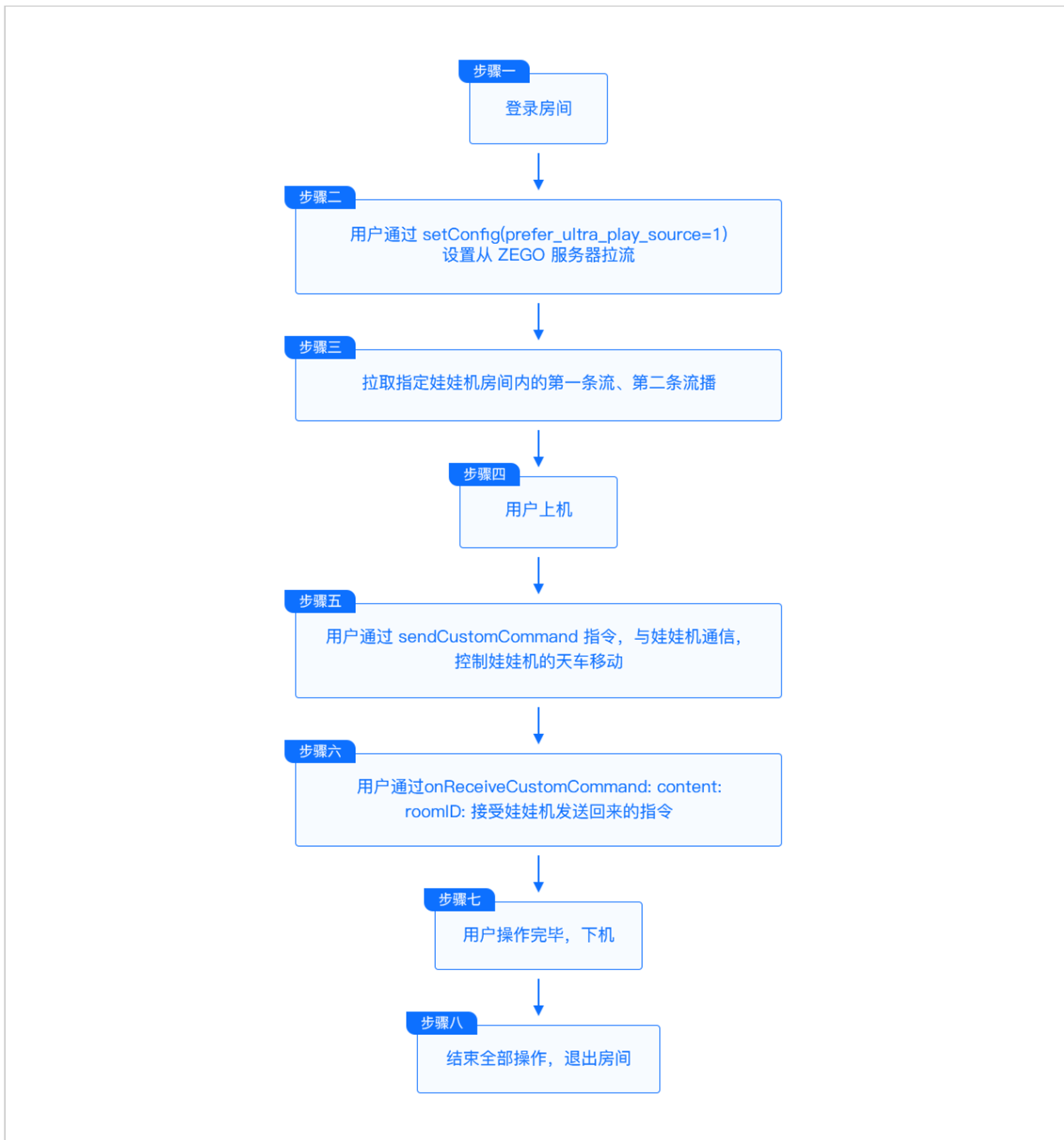


功能实现流程

娃娃机 App 端场景的主要结构及流程如下图所示：



请注意：

1. 上图中的 **用户选择上机** 步骤，不需要进行流的切换，开发者仅处理业务信令、UI 交互等。
2. **开发者注意区分 娃娃机客户端 和 娃娃机控制端**。前者指的是，提供给玩家抓娃娃的、安装在 iOS 或 Android 平台的 App。后者指的是，直接与娃娃机通过串口通信的、安装在 Android 平台上的 App，类似于娃娃机 Server。
3. 为了便于开发者更快理解 WaWaJi Client 中的逻辑，下述每节会将功能核心源码片段挑出来并加以讲解。开发者亦可直接阅读 WaWaJi Client 源码，两者是一致的。

1 娃娃机系统实现流程

- 1、安装娃娃机控制端APK到安卓板子上
- 2、娃娃机启动，推流成功后，Zego后台会给业务后台POST 流创建的相关信息(业务后台提供回调地址)，用于业务侧维护娃娃机列表
- 3、客户端的开发

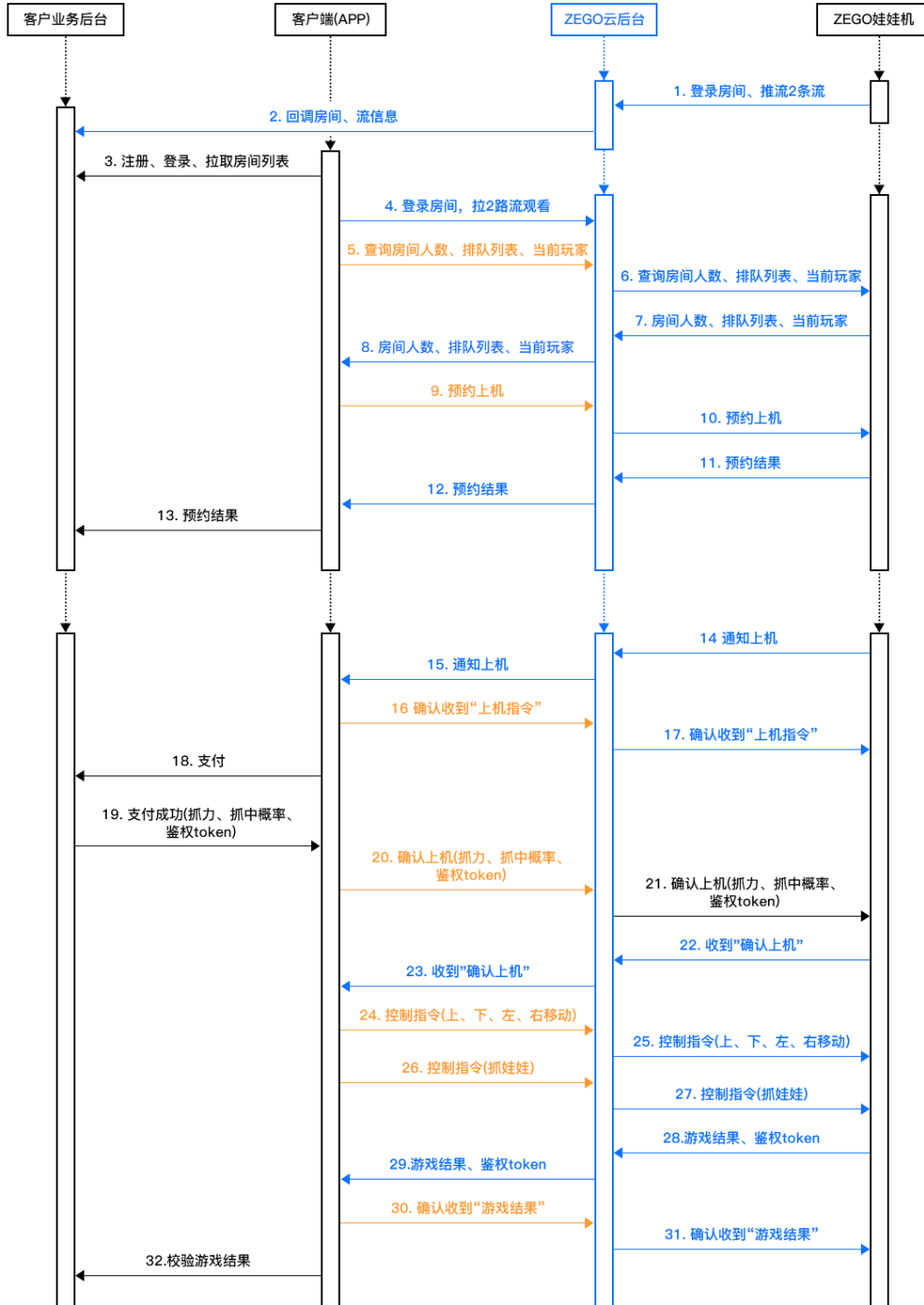
娃娃机系统实现流程如下图所示。该方案中，娃娃机控制端无需与业务后台直接通信。

娃娃机系统实现流程图

蓝色部分：调用ZEGO SDK实现

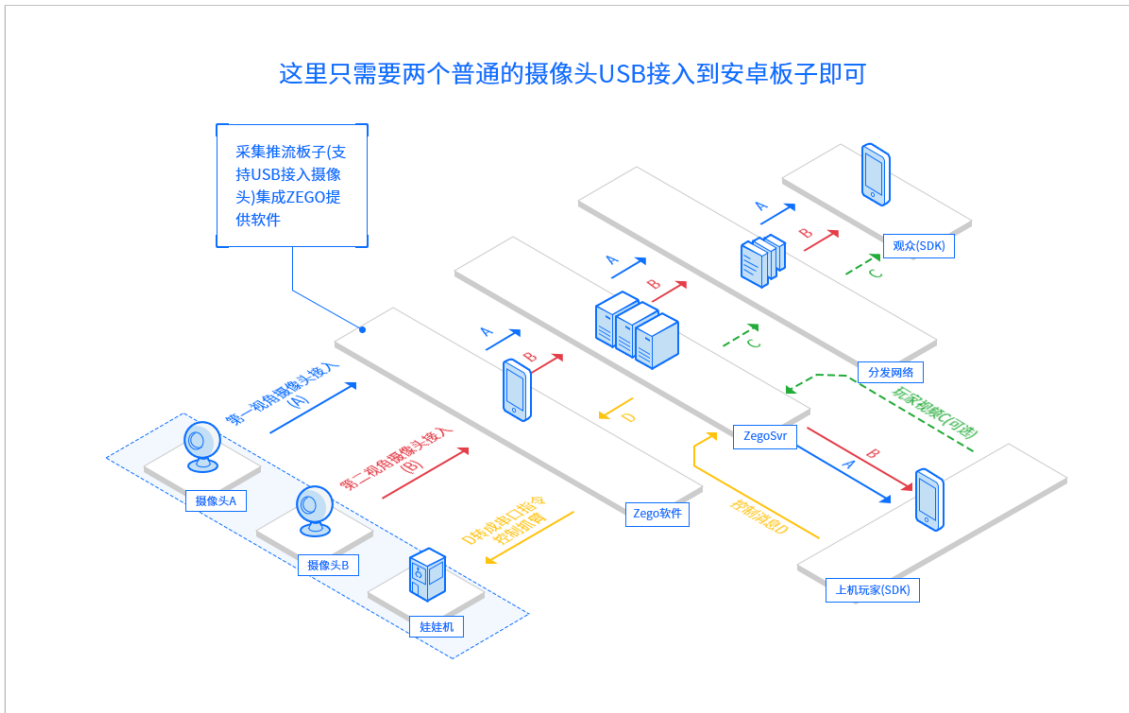
橙色部分：ZEGO开放源代码给开发者

黑色部分：客户自己开发

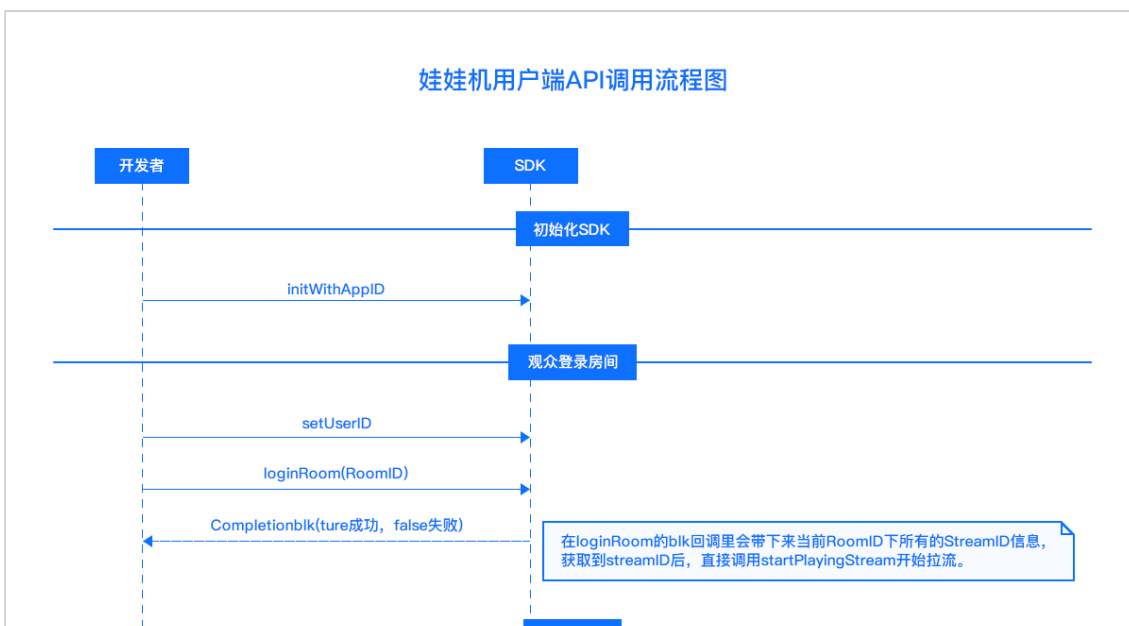


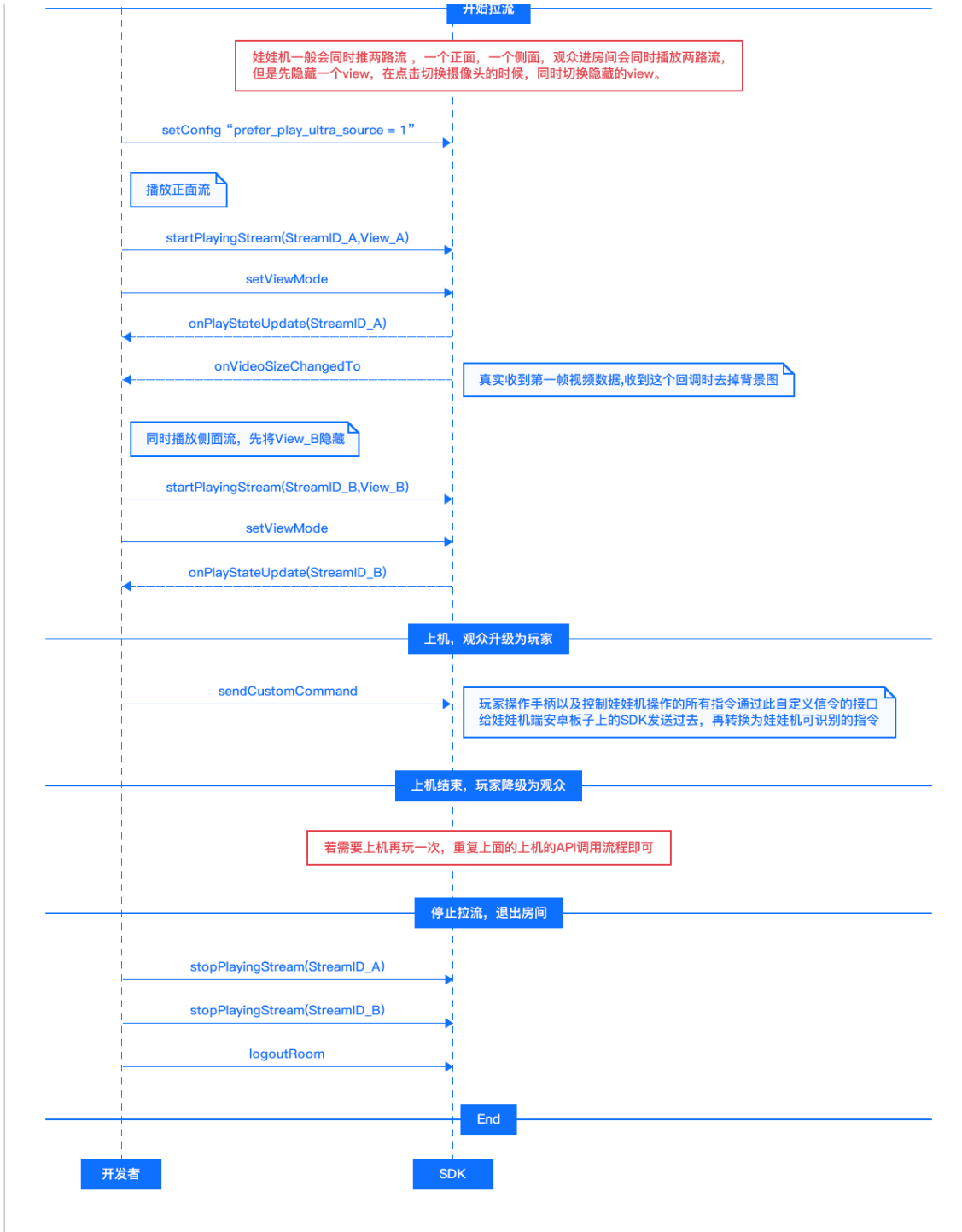
2 系统架构图

娃娃机整个系统架构图如下：



3 API 调用时序图





4 实现流程

4.1 登录房间

以下所有步骤均基于登录房间成功的前提。WaWaJi Client 中相关源码片段如下，仅供参考：

```
[[ZegoManager api] loginRoom:self.roomID role:ZEGO_AUDIENCE withCompletionBlock:^(int
errorCode, NSArray<ZegoStream *> *streamList) {
    NSLog(@"%s, error: %d", __func__, errorCode);

    if (errorCode == 0) {
        // 登录房间成功的后续操作
    }

}];
```

4.2 拉流

建议开发者登录房间成功后，直接从 ZEGO 服务器拉流。这种方案画面延迟更小，体验效果更佳。

从服务器拉流，需要在**拉流前**调用 `setConfig` 进行设置（**请注意，拉流后设置无效**）。设置语句如下：

```
[ZegoLiveRoomApi setConfig:@"prefer_play_ultra_source=1"];
```

设置成从 ZEGO 服务器拉流后，具体的拉流流程请参考：[快速集成-拉流](#)

WaWaJi Client 中拉流相关源码片段如下，仅供参考：

```
- (void)playVisibleStream:(NSString *)streamID inView:(UIView *)view {
    // 开始拉流
    [[ZegoManager api] startPlayingStream:streamID inView:view];
    // 设置流展示模式
    [[ZegoManager api] setViewMode:ZegoVideoViewModeScaleAspectFit ofStream:streamID];
}

- (void)playInvisibleStream:(NSString *)streamID inView:(UIView *)view {
    // 开始拉流
    [[ZegoManager api] startPlayingStream:streamID inView:view];
    // 设置流展示模式
    [[ZegoManager api] setViewMode:ZegoVideoViewModeScaleAspectFit ofStream:streamID];
}
```

请注意：

1. 目前 WaWaJi Client 使用的方案是，进入房间后，创建两个 view 分别播放两条流数据，用户可通过手动切换 view，继而切换当前可见流。

4.3 上机

如果开发者设置了从 ZEGO 服务器拉流（参考 4.2 节），当用户选择上机（玩游戏）时，开发者不需要进行流的切换，仅处理业务信令、UI 交互等即可。

4.4 发送指令

开发者需要调用 ZegoLiveRoom SDK 提供的 `sendCustomCommand:content:completion:` 发送指令给**娃娃机**（此处的娃娃机，指的是控制娃娃机硬件的控制端，后面简称为娃娃机）。娃娃机收到指令后，做出对应的响应。

请注意，此处指令是发送给娃娃机控制端，而不是房间里的其他玩家或自己。

WaWaJi Client 中发出指令相关源码片段如下，仅供参考：

```
// 娃娃机 App 端向娃娃机发送预约命令
BOOL invokeSuccess = [[ZegoManager api] sendCustomCommand:@[self.serverUser] content:applyCommand completion:^(int errorCode, NSString *roomId) {
    NSLog(@"%@", [NSString stringWithFormat:@"[COMMAND] CMD_APPLY 发送结果: %d(0成功, 1失败), 第 %ld 次发送", errorCode, (current - self.applyCountdown) / 2 + 1]);
}];

NSLog(@"%@", [NSString stringWithFormat:@"[COMMAND] CMD_APPLY 调用结果: %d(1成功, 0失败), 第 %ld 次发送", invokeSuccess, (current - self.applyCountdown) / 2 + 1]);
```

娃娃机客户端与控制端信令交互流程请参考：[娃娃机-信令交互](#)

4.5 接收指令

如果有必要，娃娃机 App 端可以通过 `onReceiveCustomCommand:userName:content:roomId:` 接受娃娃机返回的命令。

接收命令之前，必须先设置代理：

```
- (void)setupLiveKit {
    // 设置房间代理
    [[ZegoManager api] setRoomDelegate:self];
}
```

然后接收命令:

```
- (void)onReceiveCustomCommand:(NSString *)fromUserID userName:(NSString *)fromUserName content:(NSString *)content roomID:(NSString *)roomID {
    // 处理收到的自定义指令
}
```

娃娃机客户端与控制端信令交互流程请参考: [娃娃机-信令交互](#)

4.6 下机

由于用户下机后,可能继续停留在房间内围观,或者进行下一轮游戏。因此,建议开发者在用户下机后,继续保持从 ZEGO 服务器拉流,维持画面和操作的低延迟效果。

4.7 退出房间

如果用户不再进行游戏,退出当前的娃娃机房间,注意调用退出房间,确保停止拉流,并清空状态。

WaWaJi Client 中退出房间相关源码片段如下,仅供参考:

```
- (IBAction)onClose:(id) sender {
    // 停止拉流
    [[ZegoManager api] stopPlayingStream:self.firstStreamID];
    [[ZegoManager api] stopPlayingStream:self.secondStreamID];

    // 退出房间
    if (self.loginRoomSucceed) {
        [[ZegoManager api] logoutRoom];
    }

    // 其他状态清理
}
```