流媒体应用平台概要设计

编制： 王 科 威

审核：

版本： V1.0.0

二〇二一年十一月八日

目录

[修改历史 I](#_Toc482802323)

[一 引言 1](#_Toc482802324)

[1.1 编写目的 1](#_Toc482802325)

[1.2 背景 1](#_Toc482802326)

[1.3 定义 1](#_Toc482802327)

[1.4 参考资料 1](#_Toc482802328)

[二 基本约定 1](#_Toc482802329)

[2.1 程序名称 1](#_Toc482802330)

[2.2 目录配置 2](#_Toc482802331)

[三 系统设计 3](#_Toc482802332)

[3.1 系统结构图 3](#_Toc482802333)

[3.2 网络通信 4](#_Toc482802334)

[3.3 URI 5](#_Toc482802335)

[3.4 验证和授权 7](#_Toc482802336)

[3.5 多级级联 7](#_Toc482802337)

[四 CMS设计 7](#_Toc482802338)

[4.1 CMS功能概述 7](#_Toc482802339)

[4.2 CMS配置 7](#_Toc482802340)

[4.3 CMS服务模块 7](#_Toc482802341)

[4.4 级联数据同步 8](#_Toc482802342)

[五 CMQ设计 8](#_Toc482802343)

[5.1 功能概述 8](#_Toc482802344)

[5.2 消息队列 8](#_Toc482802345)

[5.3 CMQ级联 8](#_Toc482802346)

[5.4 发送消息 8](#_Toc482802347)

[5.5 接收消息 8](#_Toc482802348)

[5.6 同步调用 8](#_Toc482802349)

[5.7 多回复包 8](#_Toc482802350)

[六 SMS设计 8](#_Toc482802351)

[6.1 功能概述 8](#_Toc482802352)

[6.2 设备组件直连 9](#_Toc482802353)

[6.3 SMS转发 9](#_Toc482802354)

[6.4 SMS级联 9](#_Toc482802355)

[6.5 实时流 9](#_Toc482802356)

[6.6 回放流 9](#_Toc482802357)

[6.7 连接数限制 9](#_Toc482802358)

[6.8 带宽限制 9](#_Toc482802359)

[七 报警消息设计 9](#_Toc482802360)

[7.1 接收报警消息 9](#_Toc482802361)

[7.2 报警脉冲计数 9](#_Toc482802362)

[八 Device Host设计 9](#_Toc482802363)

[九 Device Component设计 9](#_Toc482802364)

[9.1 功能概述 9](#_Toc482802365)

[9.2 设备搜索 9](#_Toc482802366)

[9.3 获取设备基本信息 10](#_Toc482802367)

[9.4 动态IP设备 10](#_Toc482802368)

[9.5 设备主动注册 10](#_Toc482802369)

[9.6 设备并发访问数限制 10](#_Toc482802370)

[9.7 语音对讲 10](#_Toc482802371)

[十 GB/T 28181标准平台对接 10](#_Toc482802372)

[10.1 功能概述 10](#_Toc482802373)

[10.2 系统结构 10](#_Toc482802374)

[10.3 GB/T 28181平台级联 10](#_Toc482802375)

[10.4 流编码格式转换映射表 10](#_Toc482802376)

[十一 CMS SDK设计 10](#_Toc482802377)

[11.1 功能概述 10](#_Toc482802378)

[11.2 设备管理SDK 10](#_Toc482802379)

[11.3 流媒体SDK 10](#_Toc482802380)

[11.4 播放库SDK 10](#_Toc482802381)

[11.5 兼容性 10](#_Toc482802382)

[十二 产品授权 10](#_Toc482802383)

[12.1 版本 10](#_Toc482802384)

[12.2 授权方案 11](#_Toc482802385)

[十三 附表 11](#_Toc482802386)

# 修改历史

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **版本** | **说明** | **作者** | **审核** | **日期** |
| V0.0.1 | 初稿 | 王科威 |  | 2021-11-08 |

# 引言

## 编写目的

对流媒体应用平台进行整体的规划和设计，以此作为后续代码开发工作的基础。

## 背景

## 定义

|  |  |
| --- | --- |
| **缩写** | **全称** |
| **CMS** | **应用平台管理服务** |
| **CMD** | **应用平台部署服务** |
| **CMQ** | **应用消息队列服务** |
| **SMS** | **流媒体分发服务** |
| **CDB** | **应用平台数据库管理服务** |
| **DVS** | **对DVR、NVR和IPC等物理设备的统称** |
| **DC** | **DVS接入服务** |
| **DH** | **DC控制服务** |
| **SDK** | **应用平台SDK接口** |

## 参考资料

# 基本约定

## 程序名称

|  |  |  |
| --- | --- | --- |
| 序号 | 程序名 | 简述 |
| 应用平台基础(核心)部分 | | |
| 1 | cms | 查询和配置平台内所有的资源 |
| 2 | web | 基于WEB的平台部署 |
| 3 | sms | 音视频流数据分发 |
| 4 | cmq | 平台通信数据分发 |
| 5 | cdb | 数据库访问接入 |
| 应用平台扩展部分 | | |
| 1 | log | 平台日志记录和查询 |
| 2 | storage | 平台音视频数据集中存储和播放 |
| 3 | gb28181upward | GB28181上级平台接入 |
| 4 | gb28181downward | GB28181下级平台接入 |
| 5 | rtppublisher | RTSP/RTP流接入 |
| 6 | rtmppublisher | HTTP/RTMP流接入 |
| 7 | hlspublisher | HTTP/HLS流接入 |
| … | … | … |
| DVS设备接入部分 | | |
| 1 | devicehost | 分发和监视DC组件运行 |
| 2 | dc\_hik | 基于HIK SDK访问的组件 |
| 3 | dc\_dh | 基于DH SDK访问的组件 |
| 4 | dc\_onvif | 基于ONVIF协议访问的组件 |
| 5 | dc\_gb28181 | 基于GB28181协议访问的组件 |
| 6 | dcmonitor | 与WEB服务同步的DC监视器 |
| … | … | … |
| Demo工具 | | |
| 1 | demo | 平台接入测试与演示工具 |
| 2 | dct | 设备组件接入测试工具 |
| … | … | … |

## 目录配置

### 程序文件

* Windows

|  |  |
| --- | --- |
| 目录 | 描述 |
|  |  |

* Linux

|  |  |
| --- | --- |
| 目录 | 描述 |
|  |  |

### 配置文件

* Windows

|  |  |
| --- | --- |
| 目录 | 描述 |
|  |  |

* Linux

|  |  |
| --- | --- |
| 目录 | 描述 |
|  |  |

### 日志

* Windows

|  |  |
| --- | --- |
| 目录 | 描述 |
|  |  |

* Linux

|  |  |
| --- | --- |
| 目录 | 描述 |
|  |  |

# 系统设计

## 系统结构图

一套完整的流媒体应用系统由平台基础(核心)部分和DVS设备接入部分组成。其中：

* 平台基础(核心)部分的3种服务是不可选配的，它由一个CMS服务、一个CMQ服务和一个或多个SMS服务组成，这3种服务缺一不可；
* 每个平台可以接入一个或多个DeviceHost服务，该服务通过平台配置的DVS组件信息来控制其运行并监视其运行状态；
* 每个DeviceHost服务在同一时刻可以控制和监视一个或多个不同类型的DVS组件(如DC\_HIK, DC\_GB28181等)，但同一类型的DVS组件在同一时刻只能隶属于一个DeviceHost服务。

做为流媒体应用平台的服务，它必须尽可能适应各种各样不同格式的流媒体数据。所以流媒体应用平台提供了可扩展的应用部分，这些可扩展的服务作为流媒体应用平台内部和外部数据的桥梁，以流媒体应用平台内部数据为基础，向平台外部提供更多、更丰富的数据展示。另外，可扩展部分的服务可以是由我们自己按照一定的需求来完成，也可以由第三方通过遵照我们平台API接口的方式来完成。最后，将可扩展的服务部署到平台上即可使用。

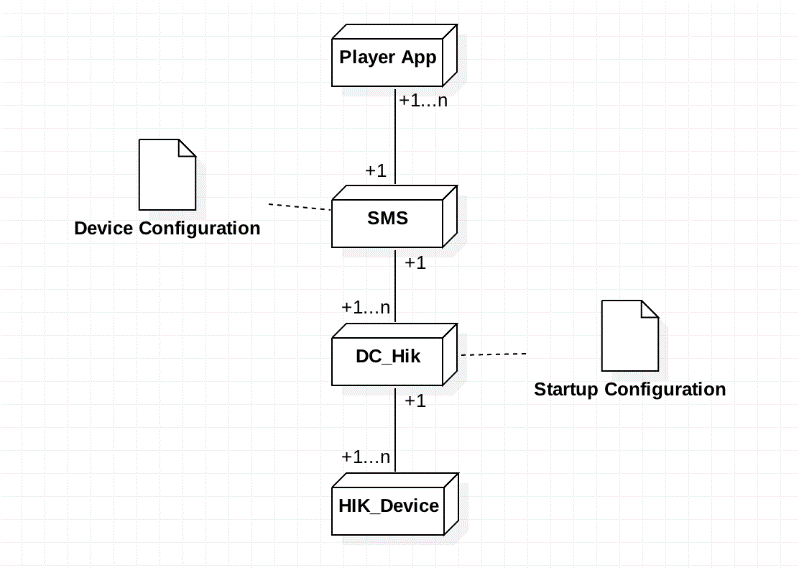
鉴于流媒体应用平台系统在不同行业中的应用场景需求，平台与平台间必须包含上下级级联结构，其基本规则如下：

* 平台内的资源只能由该平台管理员进行配置和修改，其他任何上级和下级平台都无权进行操作；
* 下级平台主动向它隶属的上一级平台推送该平台内的所有资源信息以及相应的变更通知或报警通知等；
* 上级平台能够请求播放隶属于它自己的所有下级平台内的流媒体数据，但下级平台不能请求播放上级平台的流媒体数据；
* 隶属于同一个上级的兄弟平台间不能请求播放流媒体数据。



### V1.0.0.6120设计结构图

V1.0.0.6120作为当前第一个版本的设计结构图，遵照优先构建私有格式流媒体数据传输的目的，其系统结构图如下：



该结构结构设计实现以下功能：

1. DC\_Hik模块通过HIK SDK和一个或多个物理HIK设备建立连接，并从物理HIK设备上获取音视频流数据，最后将音视频流数据主动推送到SMS中；
2. SMS服务完成音视频流数据的管理和转发，以及通知DC\_Hik模块登录哪些物理HIK设备；
3. Player App是使用SMS\_Client SDK的客户端应用程序，实现音视频流的播放。

## 网络通信

应用平台网络通信采用异步消息队列分发的模式，消息的传递类似于邮局寄信的方式。每个应用平台都仅有一个CMQ服务，它就是邮局，所有的其他服务都向CMQ服务投递消息；投递结束后不必等待消息的回复(如果外部逻辑要求等待，也是可以的)，CMQ会自动对回复消息进行处理。对于级联模式来说，消息就是在上下级的CMQ之间进行分发。

应用平台网络通信包含的方式有多种，比如：

* 请求-应答
* 发布-订阅
* 管道
* 代理

等。

### 信令格式

应用平台内包含了私有和标准信令数据2种封装格式，其中：

* 私有信令采用protocol buffers规范进行封装，封装后的数据必须进行串形化后才能通过CMQ进行传输和分发；
* 标准信令遵守标准规范的要求对信令进行封装，封装后的数据由第三方传输方法传输；
* 对于可扩展服务来说，需要在私有信令和标准信令间进行相互转换，可扩展服务通过调用应用平台API接口和平台进行交互。

### 流媒体格式

与应用平台内信令相同，流媒体也包含了私有和标准数据2种封装格式，其中：

* 私有数据格式参见《流媒体私有格式定义》，其中包括了数据帧格式和文件格式的定义；
* 标准流格式遵守标准规范的要求进行封装，封装后的数据由第三方传输方法传输；
* 对于可扩展服务来说，需要对私有数据和标准数据进行相互转换，可扩展服务通过调用应用平台API接口和平台进行交互。

## URI格式定义

任何通过应用平台SDK对平台进行访问的外部调用者，都必须遵守统一的URI标识定义，该URI标识准确且唯一的表示了应用平台内所有的数据资源。

URI定义规则如下：

* 最大长度为2048字节；
* 字符串区分大小写；
* 其中，参数字符串由键-值对组成，且键的字符串中只能使用下划线、字母和数字（[\_A-Za-z0-9]）；
* 时间为全球标准时间（UTC），在需要显示的地方应转为当地时间。格式使用yyyy-mm-dd hh-mm-ss，例如：2014-8-14 18:50:55。若要带时区格式为yyyy-mm-dd hh-mm-ss zzz，例如：2014-8-14 18:51:06 +08:00
* 如果在URI中包含了路径字符串，必须使用转意字符，转意后的字符串长度不能超过256字节。例如：d%3a%2f 20122322.avi转意后为d:/20122322.avi；
* 协议名固定使用URI表示。

### 设备

### 摄像机

### 存储设备

### 硬盘

### 实时视频流

#### V1.0.0.6120实时音视频流URI定义

根据当前系统结构设计，URI的完整描述为uri://local/dvsid/2/camera/3?stream=0。各部分定义如下：

1. uri表示协议，它是固定不变的；
2. local表示级联关系，由于当前系统中不包括级联关系，所以也是固定不变的；
3. dvsid表示设备ID标识，它在该设备隶属的平台中是唯一的，如果Device Component服务重启，该值会发生变化，设备ID标识从1开始计数；
4. camera表示摄像机ID标识，它在该摄像机隶属的设备中是唯一的，摄像机ID标识从0开始计数；
5. stream表示视频流的码流类型，0表示主码流，1表示子码流，2表示第二子码流，该参数是可选的，如果用户不设置则默认使用主码流；

### 录像回放流

### 录像下载流

### 报警输入

### 报警输出

### 语音对讲

## 验证和授权

### 应用验证

### 用户验证

### 第三方用户认证

### 权限控制

## 多级级联

# CMS设计

## CMS功能概述

## CMS配置

## CMS服务模块

### **用户管理**

### **设备管理**

### **SMS管理**

### **CMS级联管理**

### **报警订阅管理**

### **数据库访问**

## 级联数据同步

### **信息同步**

### **状态同步**

# CMQ设计

## 功能概述

## 消息队列

## CMQ级联

## 发送消息

## 接收消息

## 同步调用

## 多回复包

# SMS设计

## 功能概述

SMS是流媒体服务器的简称，它根据流通道的播放类型（实时/回放）对流通道进行管理、复用和播放控制。SMS是流媒体应用平台的基础服务，与其进行数据交互所有其他外部服务都必须遵守流媒体应用平台内部的私有协议；外部服务也可以通过使用SMS提供的SDK接口完成向SMS推送流和从SMS获取流。

在SMS服务中，只有实时流通道才能进行复用，并且复用流采用unicast的方式进行传输。

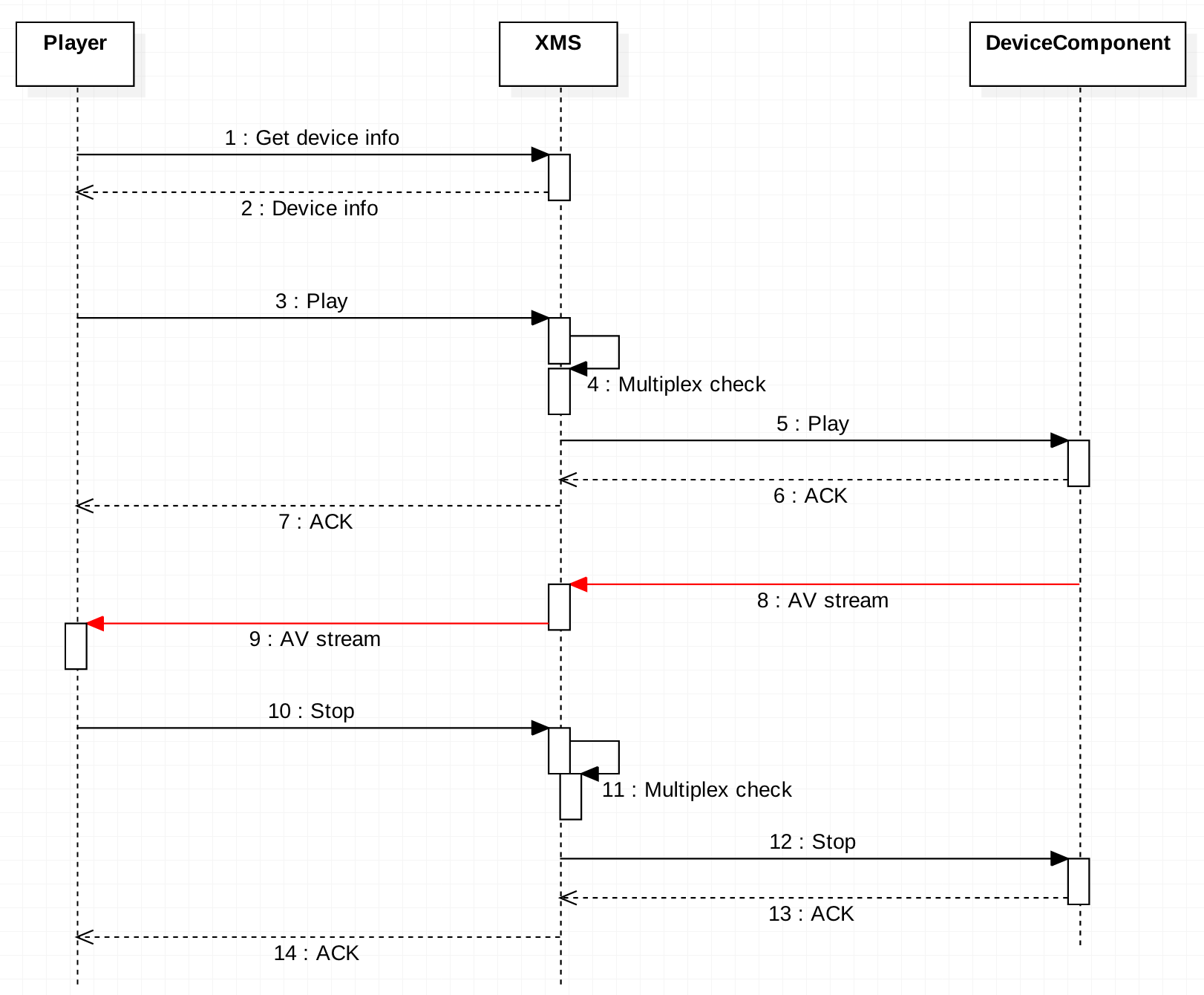
## 设备直连

## SMS转发

## SMS级联

## 实时流

### V1.0.0.6120实时音视频流



## 回放流

## 连接数限制

### 实时预览

### 回放及下载

## 带宽限制

### 设计思路

### 实现原理

# 报警消息设计

## 接收报警消息

## 报警脉冲计数

# Device Host设计

# Device Component设计

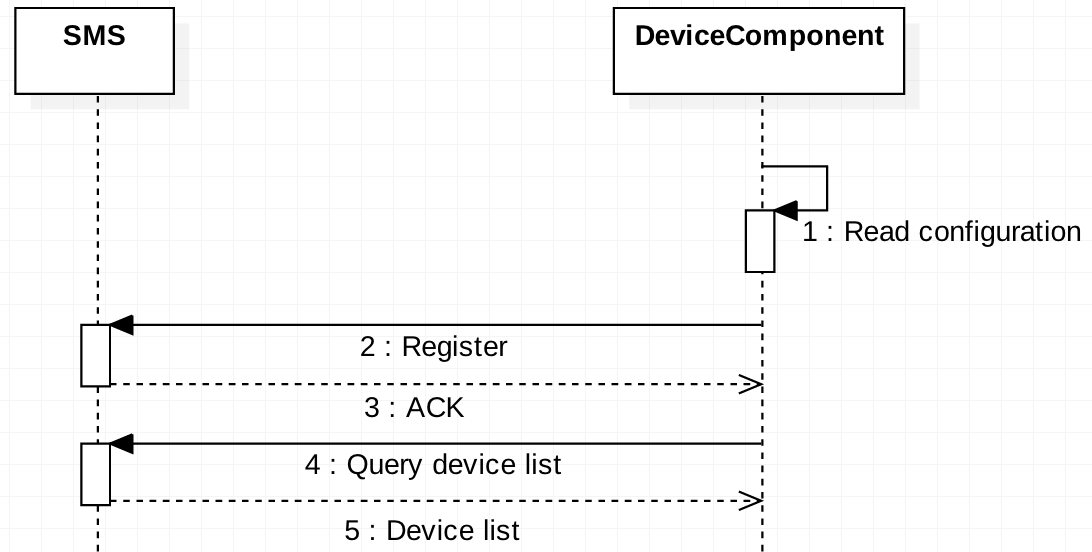
## 功能概述

Device Component作为流媒体应用平台的边缘服务，负责将物理设备的信息和流媒体数据接入到流媒体应用平台中，其接入内容包括了：

1. 物理设备信息的接入；
2. 物理设备控制信令语义的转换；
3. 物理设备流数据格式的转换；

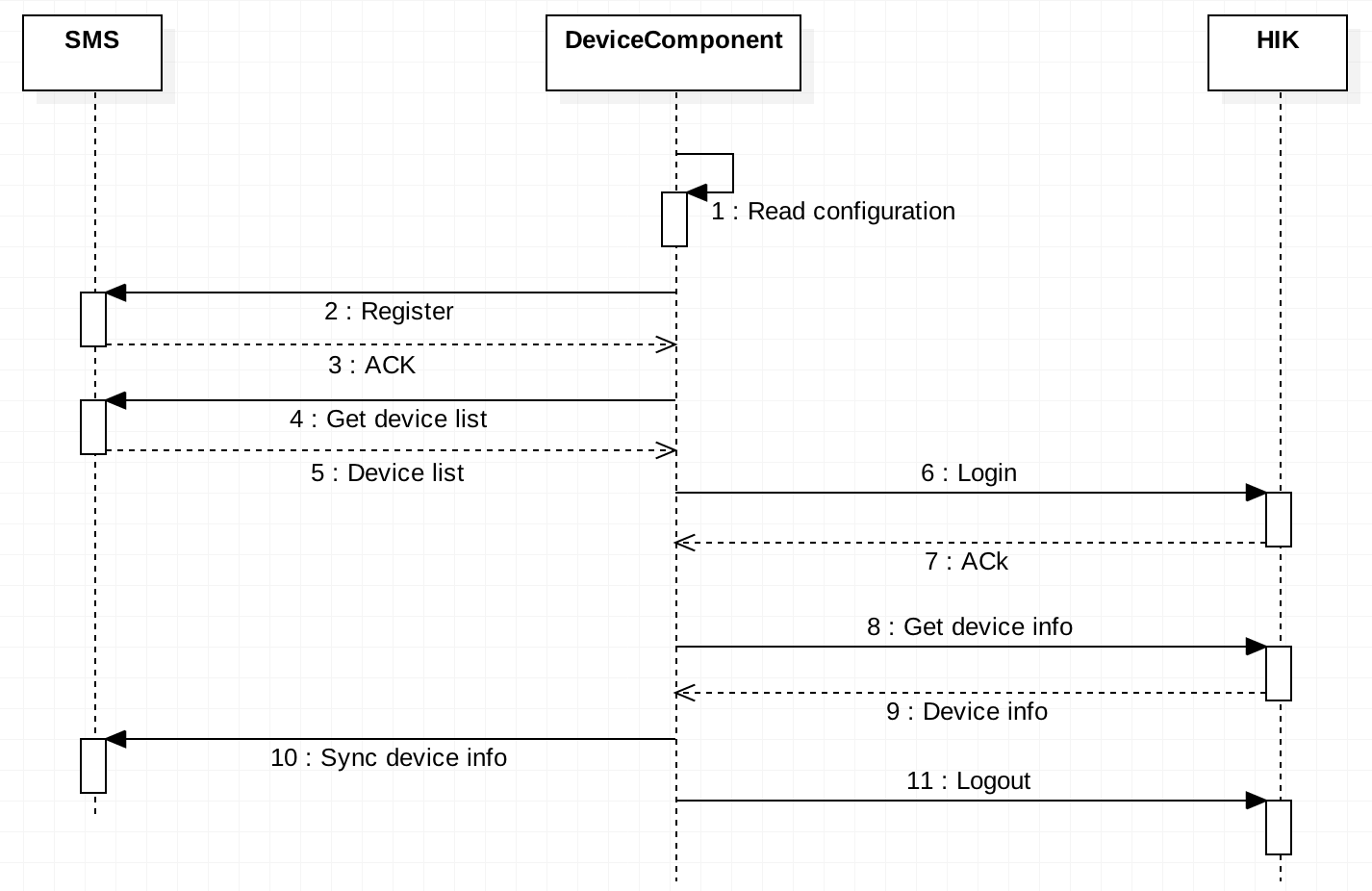
## Device Component服务注册

### V1.0.0.6120服务注册



## 设备登录/注销

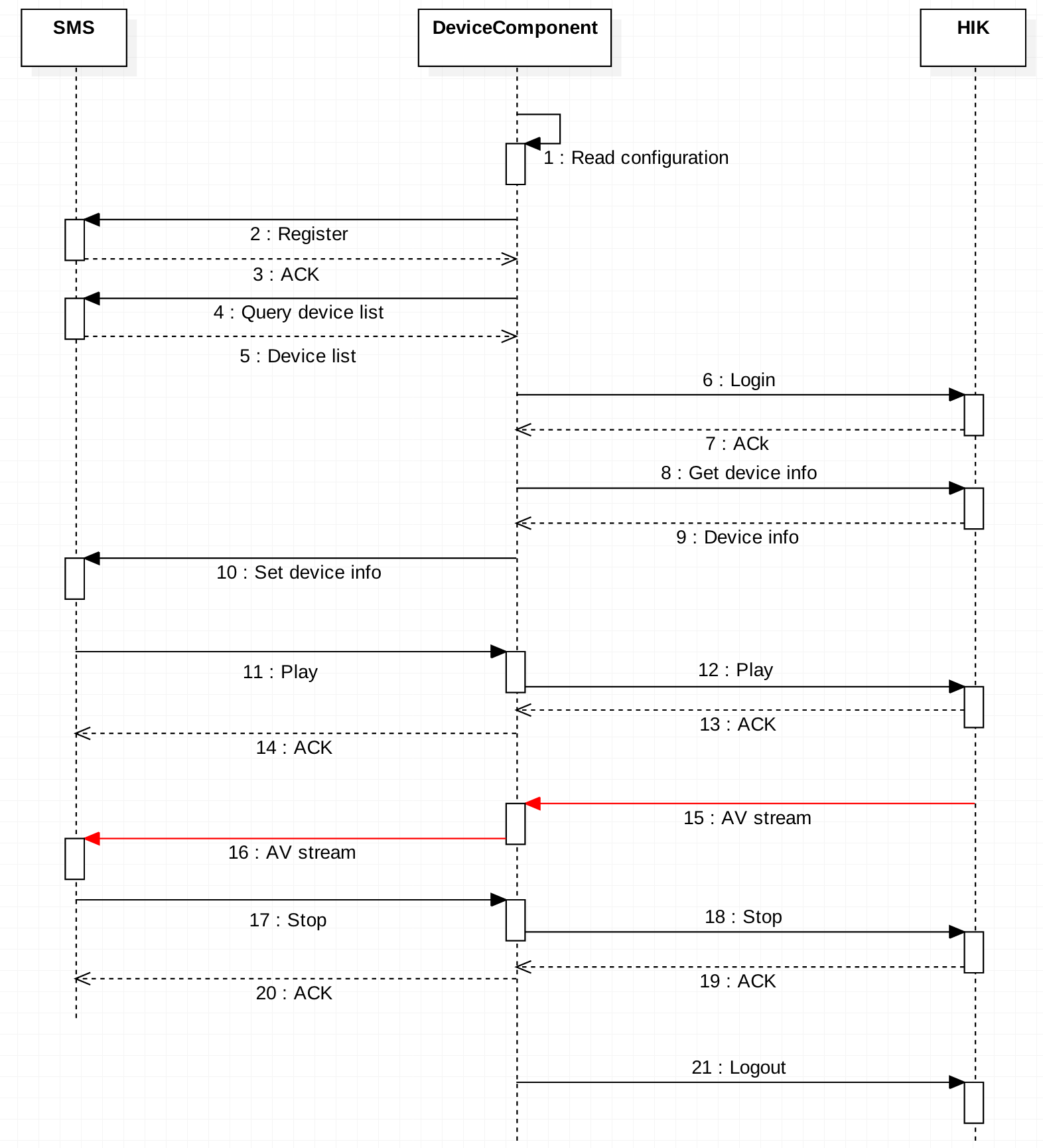
### V1.0.0.6120设备登录/注销



## 获取设备基本信息

## 实时音视频流

### V1.0.0.6120实时音视频流



## 回放音视频流

## 动态IP设备

## 设备主动注册

## 设备并发访问数限制

## 语音对讲

# GB/T 28181标准平台对接

## 功能概述

## 系统结构

## GB/T 28181平台级联

## 流编码格式转换映射表

# 流媒体应用平台SDK设计

流媒体应用平台自身是一个独立且封闭的应用环境，其自身内部使用的通信协议和流数据封装格式都是私有的。由此可见，如果需要将其他外部平台或设备的通信或流媒体数据引入到流媒体应用平台中，就必须使用平台提供的通信SDK和流媒体SDK来完成接入。

## 功能概述

## 通信SDK接口设计

### V1.0.0.6120通信SDK

#### SDK初始化（HHMQ\_SDK\_Init）

|  |  |  |
| --- | --- | --- |
| 函数定义 | int HHMQ\_SDK\_Init(); | |
| 功能说明 | 初始化通信SDK。 | |
| 参数 |  |  |
| 返回值 | 错误码 | |
| 说明 | 调用通信SDK其他方法的前提。 | |

#### SDK资源释放（HHMQ\_SDK\_Release）

|  |  |  |
| --- | --- | --- |
| 函数定义 | int HHMQ\_SDK\_Release(); | |
| 功能说明 | 释放初始化SDK时在SDK内部创建的资源。 | |
| 参数 |  |  |
| 返回值 | 错误码 | |
| 说明 | 在结束之前最后调用。 | |

#### 外部服务注册（HHMQ\_SDK\_RegisterService）

|  |  |  |
| --- | --- | --- |
| 函数定义 | int HHMQ\_SDK\_RegisterService(); | |
| 功能说明 | 所有外部服务若要将服务信息和流数据接入到流媒体应用平台中，都必须首先向流媒体应用平台进行注册，以此表明外部服务的身份。外部应用程序（如Player等）不需要调用该接口。 | |
| 参数 |  |  |
| 返回值 | 错误码 | |
| 说明 | 不同的外部服务可按照各自的执行逻辑选择向流媒体应用平台注册的时间。 | |

#### 获取设备列表（HHMQ\_SDK\_GetDeviceList）

|  |  |  |
| --- | --- | --- |
| 函数定义 | int HHMQ\_SDK\_GetDeviceList(DeviceList& devices); | |
| 功能说明 | Device Component服务使用该接口从流媒体应用平台中获取预先配置好的设备列表，设备的相关信息用于Device Component服务登录物理设备。 | |
| 参数 | devices | 设备参数集合 |
| 返回值 | 错误码 | |
| 说明 | Device Component只能登录设备列表中已包含的设备。 | |

#### 设置设备信息（HHMQ\_SDK\_SetDeviceInfos）

|  |  |  |
| --- | --- | --- |
| 函数定义 | int HHMQ\_SDK\_SetDeviceList(const DeviceInfos& devices); | |
| 功能说明 | Device Component服务登录物理设备成功并获取其相关信息后，可以使用该接口将设备信息设置到流媒体应用平台中。 | |
| 参数 | devices | 设备信息集合 |
| 返回值 | 错误码 | |
| 说明 |  | |

#### 获取设备信息（HHMQ\_SDK\_GetDeviceInfos）

|  |  |  |
| --- | --- | --- |
| 函数定义 | int HHMQ\_SDK\_GetDeviceList(DeviceInfos& devices); | |
| 功能说明 | 外部应用程序（如Player等）调用该接口从流媒体应用平台中获取预先配置好的设备的当前设备信息。 | |
| 参数 | devices | 设备信息集合 |
| 返回值 | 错误码 | |
| 说明 |  | |

## 流媒体SDK接口设计

### V1.0.0.6120流媒体SDK

#### SDK初始化（HHMS\_SDK\_Init）

|  |  |  |
| --- | --- | --- |
| 函数定义 | int HHMS\_SDK\_Init(); | |
| 功能说明 | 初始化流媒体SDK。 | |
| 参数 |  |  |
| 返回值 | 错误码 | |
| 说明 | 调用流媒体SDK其他方法的前提。 | |

#### SDK资源释放（HHMS\_SDK\_Release）

|  |  |  |
| --- | --- | --- |
| 函数定义 | int HHMS\_SDK\_Release(); | |
| 功能说明 | 释放初始化SDK时在SDK内部创建的资源。 | |
| 参数 |  |  |
| 返回值 | 错误码 | |
| 说明 | 在结束之前最后调用。 | |

#### 打开实时音视频流（HHMS\_SDK\_OpenStream）

|  |  |  |
| --- | --- | --- |
| 函数定义 | int HHMS\_SDK\_OpenStream (const int dvsid = 0, const int cameraid = 0, const char mediatype = 0, RealDataCallback handle = NULL, void\* userData = NULL); | |
| 功能说明 | 外部应用程序（如Player等）调用该接口打开流媒体应用平台中的某一个实时音视频流。 | |
| 参数 | dvsid | 设备ID标识 |
| cameraid | 摄像头ID标识 |
| mediatype | 码流类型：0表示主码流，1表示子码流，2表示第二子码流 |
| handle | 获取流数据的回调函数 |
| userData | 回调函数中的用户定义数据 |
| 返回值 | >0表示外部应用程序使用的流ID标识，==0表示打开失败。 | |
| 说明 | typedef void(CALLBACK \* RealDataCallback)(int streamid, DWORD dwDataType, BYTE \*pBuffer, DWORD dwBufSize, void \*pUser);  其中：   * streamid表示流ID表示； * dwDataType表示数据类型； * pBuffer数据缓存； * dwBufSize表示数据大小； * userData表示用户自定义数据 | |

#### 关闭实时音视频流（HHMS\_SDK\_CloseStream）

|  |  |  |
| --- | --- | --- |
| 函数定义 | int HHMS\_SDK\_CloseStream (const int streamid = 0); | |
| 功能说明 | 外部应用程序（如Player等）调用该接口关闭流媒体应用平台中的某一个实时音视频流。 | |
| 参数 | streamid | 外部应用程序使用的流ID标识，该值在使用HHMS\_SDK\_OpenStream方法时返回。 |
| 返回值 | 错误码 | |
| 说明 |  | |

## 播放库SDK接口设计

### V1.0.0.6120播放库SDK

#### SDK初始化（HHMP\_SDK\_Init）

|  |  |  |
| --- | --- | --- |
| 函数定义 | int HHMP\_SDK\_Init(); | |
| 功能说明 | 初始化播放库SDK。 | |
| 参数 |  |  |
| 返回值 | 错误码 | |
| 说明 | 调用播放库SDK其他方法的前提。 | |

#### SDK资源释放（HHMP\_SDK\_Release）

|  |  |  |
| --- | --- | --- |
| 函数定义 | int HHMP\_SDK\_Release(); | |
| 功能说明 | 释放初始化SDK时在SDK内部创建的资源。 | |
| 参数 |  |  |
| 返回值 | 错误码 | |
| 说明 | 在结束之前最后调用。 | |

#### 打开音视频流播放（HHMP\_SDK\_StartPlay）

|  |  |  |
| --- | --- | --- |
| 函数定义 | int HHMP\_SDK\_StartPlay (HWND hwnd= NULL); | |
| 功能说明 | 外部应用程序（如Player等）调用该接口在指定窗口上播放实时音视频流。 | |
| 参数 | hwnd | 播放实时音视频流的窗口句柄。 |
| 返回值 | >0表示外部应用程序使用的播放ID标识，==0表示打开失败。 | |
| 说明 |  | |

#### 关闭音视频流播放（HHMP\_SDK\_StopPlay）

|  |  |  |
| --- | --- | --- |
| 函数定义 | int HHMP\_SDK\_StopPlay (const int playid = 0); | |
| 功能说明 | 外部应用程序（如Player等）调用该接口在指定窗口上关闭实时音视频流播放。 | |
| 参数 | playid | 外部应用程序使用的播放ID标识，该值在使用HHMP\_SDK\_StartPlay方法时返回。 |
| 返回值 | 错误码 | |
| 说明 |  | |

## 兼容性

# 产品授权

## 版本

## 授权方案

# 附表