

云小微账号平台接入方案

版本	说明	日期
1.0.0	初稿	2019.05.02
1.0.1	新增无手机app参与的账号方案处理	2019.05.02
1.1.0	修改文档整体结构	2019.05.03

云小微账号平台接入方案

0 总述

1 名词解释

2 账号接入方案选型

2.1 账号类型分类

2.2 分类说明

2.2.1 微信

2.2.2 QQ

2.2.3 三方账号

2.2.4 访客账号

3 微信/QQ授权开放平台申请流程

4 账号接入场景选型

5 无屏方案 (TVSSDK)

5.1 ER流程图

5.2 手机端

5.2.1 第一步：账号登录

5.2.2 第二步：账号验证

5.2.3 第三步：账号设备绑定

5.2.4 第四步：授权信息生成

5.2.5 第五步：授权信息ClientID传递

5.3 设备端

5.3.1 第一步：授权

5.3.2 第二步：TVSToken的获取

5.4 音乐授权

5.4.1 第一步：厂商APP登录

5.4.2 第二步：厂商账号绑定

5.4.3 第三步：音乐授权管理页面

5.4.4 第四步：跳转云小微APP完成授权

6 无屏方案 (TVSAPI)

6.1 ER流程图

6.2 手机端

6.2.1 第一步：账号登录

6.2.2 第二步：账号验证

6.2.3 第三步：账号设备绑定

6.2.4 第四步：授权信息生成

6.2.5 第五步：授权信息ClientID传递

6.3 设备端

6.3.1 第一步：授权

6.3.1.1 接口描述

6.3.1.2 请求数据

6.3.1.3 返回参数

6.3.2 第二步：定时刷票

6.3.2.1 接口描述

6.3.2.2 请求参数

6.3.2.3 返回参数

6.3.3 第三步：启动刷票

6.3.4 第四步：网络切换刷票

6.4 音乐授权

6.4.1 第一步：厂商APP登录

6.4.2 第二步：厂商账号绑定

6.4.3 第三步：音乐授权管理页面

6.4.4 第四步：跳转云小微APP完成授权

7 无屏方案（基础API）

7.1 ER流程图

7.2 手机端

7.2.1 第一步：账号登录

7.2.2 第二步：账号验证

7.2.3 第三步：账号设备绑定

7.2.4 第四步：授权信息生成

7.2.5 第五步：授权信息ClientID传递

7.3 厂商云端

7.3.1 第一步：授权

7.3.1.1 接口描述

7.3.1.2 请求参数

7.3.1.3 返回参数

7.3.2 第二步：定时刷票

7.3.2.1 接口描述

7.3.2.2 请求参数

7.3.2.3 返回参数

7.4 音乐授权

7.4.1 第一步：厂商APP登录

7.4.2 第二步：厂商账号绑定

7.4.3 第三步：音乐授权管理页面

7.4.4 第四步：跳转云小微APP完成授权

8 有屏方案（TVSSDK）

8.1 ER流程图

8.2 手机端

8.2.1 第一步：账号登录

8.2.2 第二步：账号验证

8.2.3 第三步：账号设备绑定

8.2.4 第四步：授权信息生成

8.2.5 第五步：授权信息ClientID传递

8.3 设备端

8.3.1 第一步：授权

8.3.2 第二步：TVSToken的获取

8.4 音乐授权

8.4.1 第一步：TVSToken获取

8.4.2 第二步：接入音乐登录二维码页面

8.4.2.1 Android智能设备

8.4.2.2 Linux智能设备

8.4.3 第三步：使用微信或手Q扫码登录

9 有屏方案 (TVSAPI)

9.1 ER流程图

9.2 手机端

9.2.1 第一步：账号登录

9.2.2 第二步：账号验证

9.2.3 第三步：账号设备绑定

9.2.4 第四步：授权信息生成

9.2.5 第五步：授权信息ClientID传递

9.3 设备端

9.3.1 第一步：授权

9.3.1.1 接口描述

9.3.1.2 请求数据

9.3.1.3 返回参数

9.3.2 第二步：定时刷票

9.3.2.1 接口描述

9.3.2.2 请求参数

9.3.2.3 返回参数

9.3.3 第三步：启动刷票

9.3.4 第四步：网络切换刷票

9.4 音乐授权

9.4.1 第一步：TVSToken获取

9.4.2 第二步：接入音乐登录二维码页面

9.4.2.1 Android智能设备

9.4.2.2 Linux智能设备

9.4.3 第三步：使用微信或手Q扫码登录

10 有屏方案 (基础API)

10.1 ER流程图

10.2 手机端

10.2.1 第一步：账号登录

10.2.2 第二步：账号验证

10.2.3 第三步：账号设备绑定

10.2.4 第四步：授权信息生成

10.2.5 第五步：授权信息ClientID传递

10.3 厂商云端

10.3.1 第一步：授权

10.3.1.1 接口描述

10.3.1.2 请求参数

10.3.1.3 返回参数

10.3.2 第二步：定时刷票

10.3.2.1 接口描述

10.3.2.2 请求参数

10.3.2.3 返回参数

10.4 音乐授权

10.4.1 第一步：TVSToken获取

10.4.2 第二步：接入音乐登录二维码页面

10.4.2.1 Android智能设备

10.4.2.2 Linux智能设备

10.4.3 第三步：使用微信或手Q扫码登录

0 总述

为了对设备完成授权和绑定功能，我们在云小微设备开放平台引入了账号体系。

接入账号大体上分为以下几个步骤：

- 1) 设备开放平台上注册开发者，创建应用，选择设备类型与接入方式
- 2) 如需要接入微信/QQ授权请先在微信开放平台和QQ互联平台完成AppID的申请
- 3) 勾选接入云小微账号体系，填入必要账号开放平台信息
- 4) 在官方github下载DMSDK（手机SDK）、账号规范与流程说明和Demo
- 5) 手机端接入DMSDK主要完成授权、设备绑定、多端互动管理、CP授权等功能。
- 6) 设备端根据接入方式的不同，按照以下内容接入账号体系和流程。

1 名词解释

DMSDK

全称为DeviceManager SDK，用于云小微平台手机端接入的SDK，分为Android和iOS端，主要用来做账号授权、设备绑定、非语音功能管理接口的打通以及部分运营H5页面承载。该SDK是用来接入微信/QQ授权、厂商账号授权的必要条件。

ClientID

使用登录得到的账号信息、以及设备端的设备信息生成的客户端唯一标识，用作授权云小微服务。

云小微授权

指的是在设备端使用ClientID信息通过云小微规定的标准方式换取TVS票据的授权流程。

技能授权

指的是在手机端通过授权管理页面或设备端的扫码方式对不同的CP技能进行二次授权。

TVS

全称为Tencent Voice Service，是用在智能设备端的语音服务，集合提供了语音识别、语义理解、TTS播报功能。

openid

授权用户唯一标识

token

票据（Signature）也有时称为“签名”，形象一点，票据其实可以理解为现实世界中的“电影票”、“飞机票”等等，是一种凭证。以电影票为例，我们去看电影需要首先购买电影票，然后拿着电影票在规定的时刻去电影院，检票入场后观看电影。

access_token

用户的接口调用凭证，即真正的电影票凭据

refresh_token

用来刷新access_token的票据，假设观众错过了电影票的时间，可以通过一个额外的会员卡号来取电影院换取一个新的有效电影票凭据

expireTime

票据过期时间，微信access_token过期时间为两小时，refresh_token过期时间为30天，QQ互联access_token过期时间为90天

productId

同一品类的设备共享一个ProductID，ProductID由bot平台生成，具体格式为appkey:appaccesstoken

dsn

设备唯一序列码，每个设备不同，具体规则由厂商自行制定

TVSToken

TVS后台生成的用于设备访问的票据

2 账号接入方案选型

2.1 账号类型分类

云小微授权方式	微信	QQ	手机号	三方账号	访客账号
授权形式	微信手机客户端授权	QQ手机客户端授权	手机验证码形式授权	厂商开发	设备端自动授权
ClientID生成方式	DMSDK	DMSDK	DMSDK	DMSDK	设备SDK
账号类型字段	WX	QQOpen	PhoneNumber	ThirdParty	无

2.2 分类说明

2.2.1 微信

在微信开放平台申请AppID，在手机端App接入DMSDK，拉起微信/手Q方式进行登录与授权。如果是有屏设备，也可以通过在设备端拉起微信的授权二维码页面，通过扫码方式来授权。具体流程详见“3 微信/QQ授权开放平台申请流程”

2.2.2 QQ

在QQ互联平台申请AppID，在手机端App接入DMSDK，拉起微信/手Q方式进行登录与授权。如果是有屏设备，也可以通过在设备端拉起QQ的授权二维码页面，通过扫码方式来授权。具体流程详见“3 微信/QQ授权开放平台申请流程”

2.2.3 三方账号

即为厂商账号，授权和取消授权的操作由厂商自行完成，在手机端APP接入DMSDK，通过传入厂商账号ID和设备标识信息（productId,DSN），完成ClientID的生成后传入设备端完成设备端授权行为。如果需要访问到厂商特有需要鉴权的资源，也请传入厂商的票据信息。

2.2.4 访客账号

也称为无账号体系，将设备唯一标识作为“账号”信息，通过将ProductID和DSN按照一定的编码规则生成一个访客账号ClientID完成授权功能。

3 微信/QQ授权开放平台申请流程

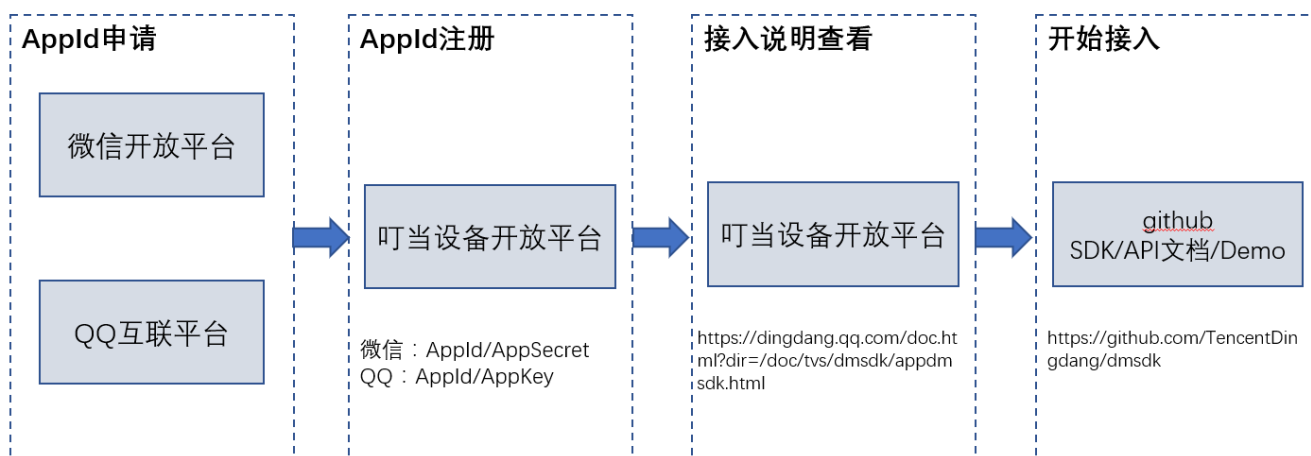
主要分为以下四个步骤

AppId申请：进度由微信开放平台和QQ互联平台决定（如有特殊需求，可邮件加急）

AppId注册：在云小微设备开放平台上填写相关平台级别AppId信息

接入说明查看：可以查看到整个账号体系在端到端之间的大体流程

开始接入：通过官方GitHub的链接可以获取到最新的DMSDK、API文档和APP Demo。



4 账号接入场景选型

从场景来看，智能设备主要分为无屏和有屏，在接入云小微智能服务时，目前主要分为TVSSDK、TVSAPI和基础API三种接入方式。

TVSSDK：基于TVSAPI定义的一套接入云小微语音能力的终端SDK，分为Android和Linux两个端版本。

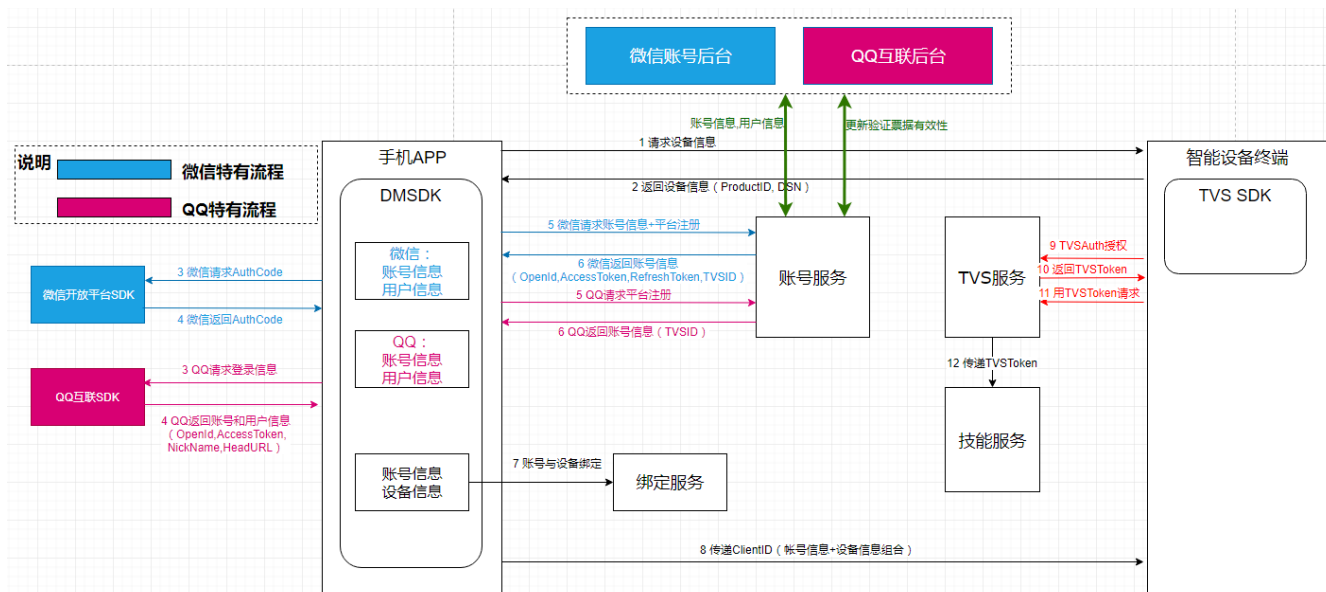
TVSAPI：一套基于HTTP的语音Web接口。它接收用户的语音或文本输入，传递给腾讯云小微后台进行处理，并将处理结果以语音、文本或卡片的形式，返回给用户。

基础API：针对语音识别、语义理解、TTS接口服务定义的一套基础的http接口。

下面就针对每种场景组合进行详细说明。

5 无屏方案（TVSSDK）

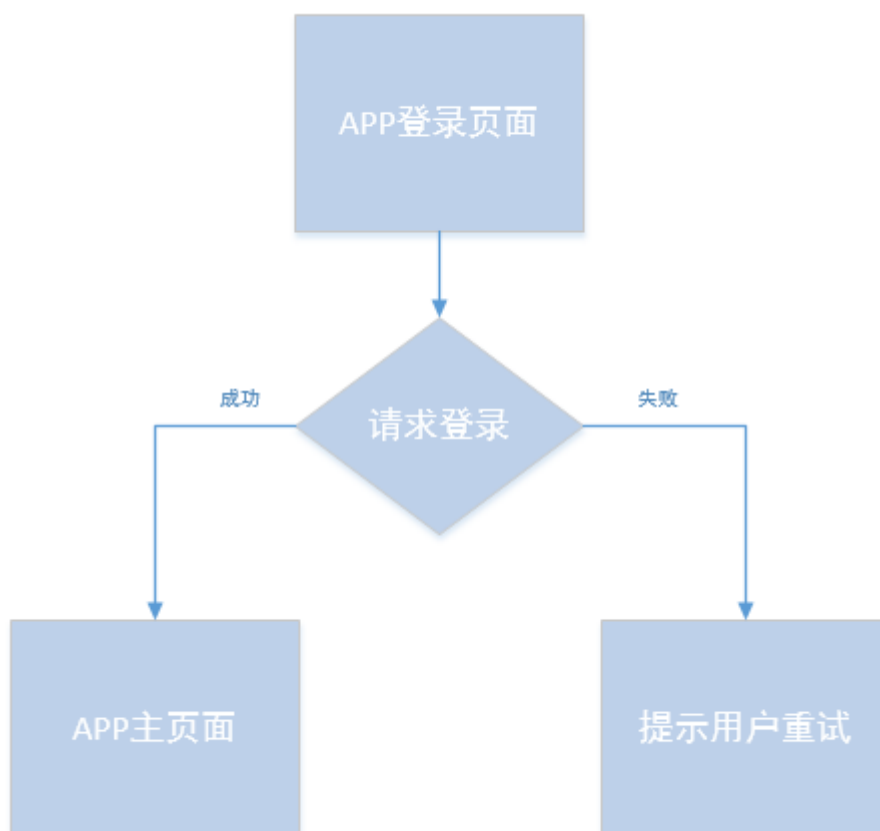
5.1 ER流程图



5.2 手机端

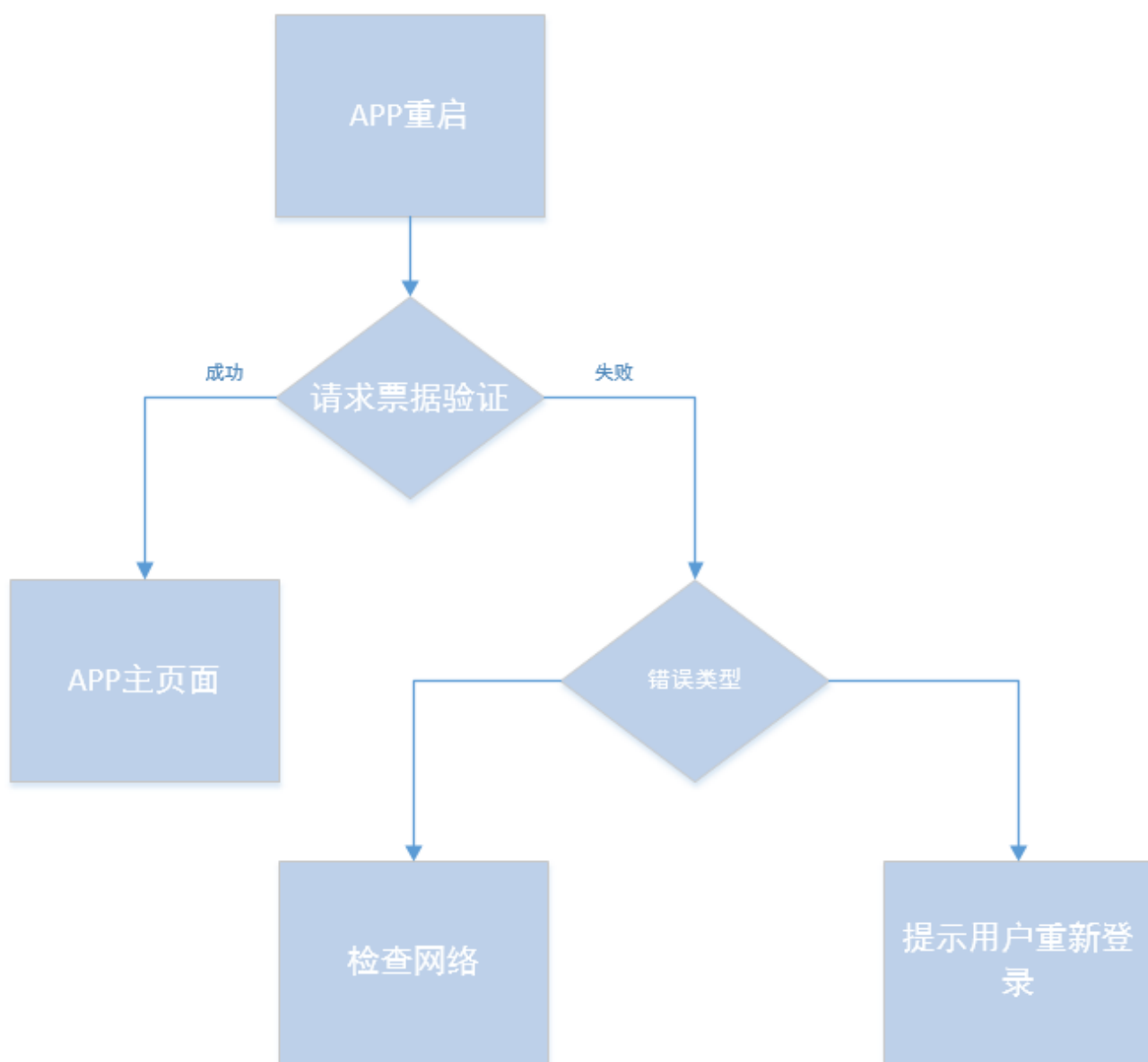
5.2.1 第一步：账号登录

手机APP在初始化状态下，应该是无登录态的，DMSDK中提供了针对微信和QQ的授权接口tvLogin可以通过拉起微信或手Q的授权页面来给厂商APP提供授权能力，根据回调成功或失败接口来进行下一步操作，例如成功后带上登录信息和用户信息进入下一个页面，失败后就通过强提示功能提示用户重新登录。



5.2.2 第二步：账号验证

手机APP在有登录态的场景下，进入后，必须调用requestTokenVerify进行票据验证。成功后带上最新的登录信息和用户信息进行正常操作，失败后通过强提示功能：清除本地的账号信息缓存并跳转到登录页面要求用户重新登录

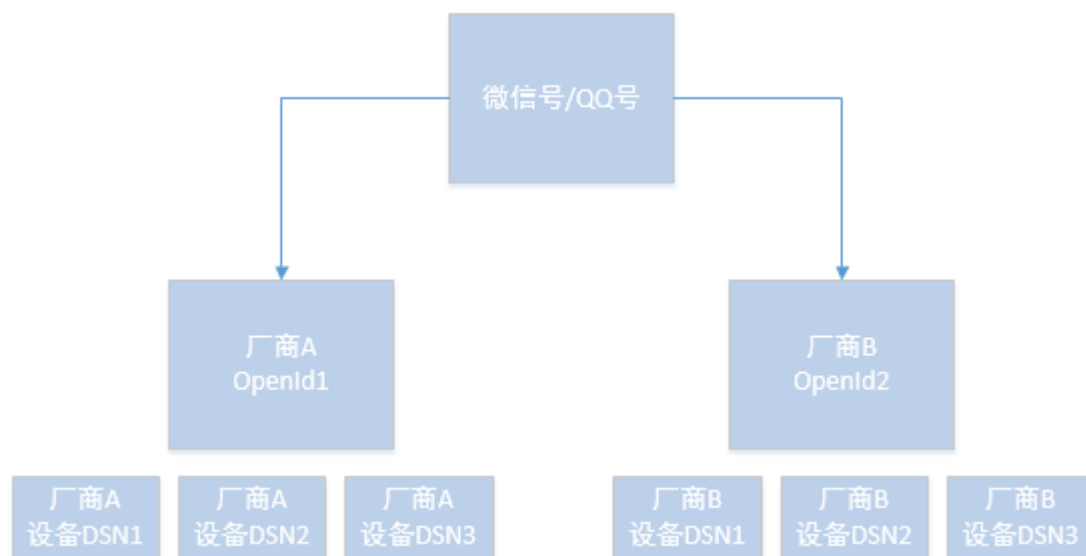


5.2.3 第三步：账号设备绑定

需要完成账号信息与设备信息的绑定，对于支付、IoT、闹钟提醒、推送等场景需要校验云端绑定态。

对于绑定，解绑时填写的字段有一定的要求，具体可详见DMSDK API文档。

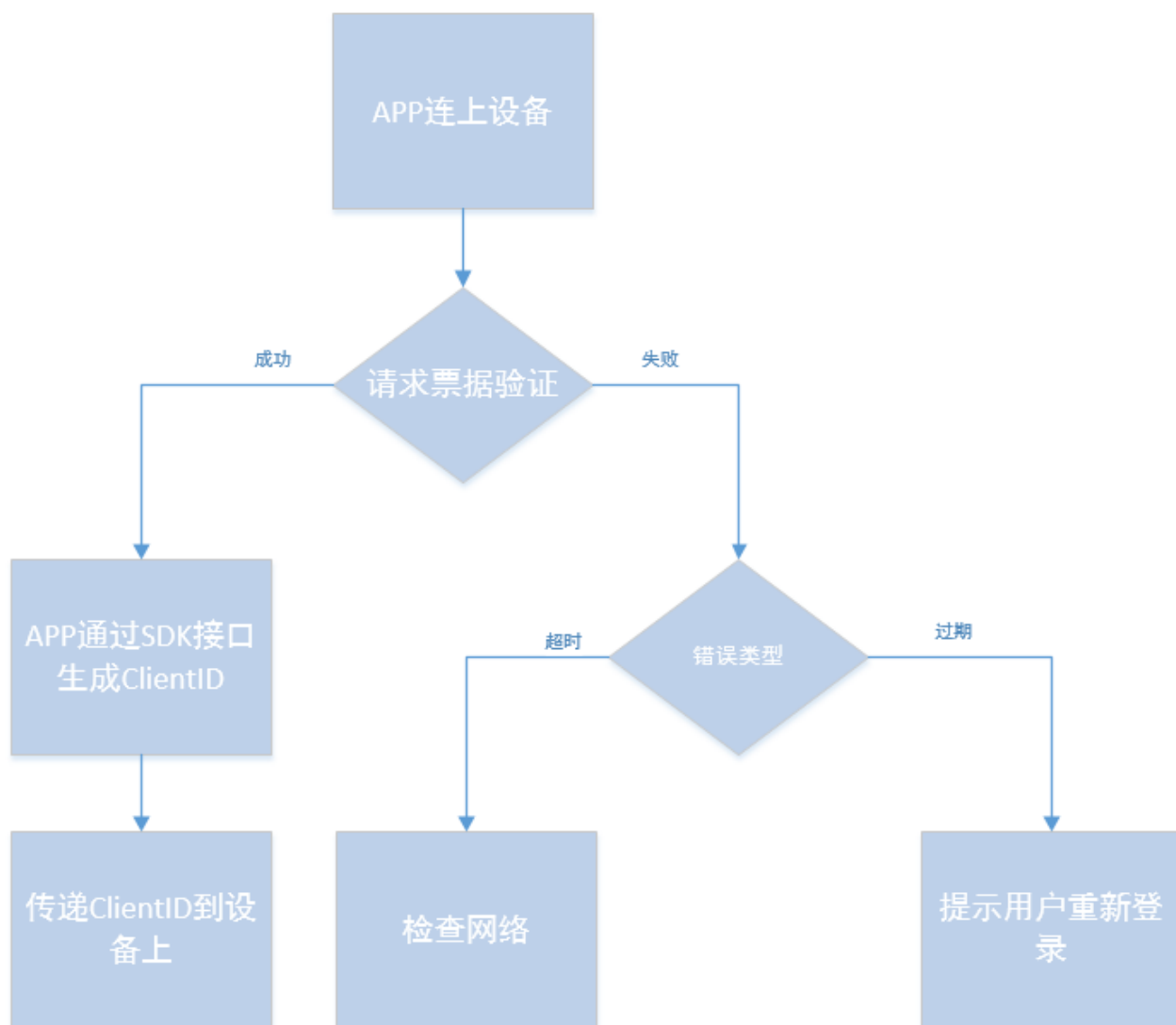
如果是非微信或QQ授权类型的，请按照DMSDK API文档，传入第三方账号字段进行绑定、解绑。



5.2.4 第四步：授权信息生成

授权信息ClientID是手机到设备的需要传递的唯一账号数据信息，并且只有刷票成功才可以去生成ClientID。

如果是非微信或QQ授权类型的，请按照DMSDK API文档，传入第三方账号字段进行ClientID的获取。



5.2.5 第五步：授权信息ClientID传递

使用蓝牙、局域网方式，或通用设备通信协议完成，该部分为厂商自行实现

5.3 设备端

设备上是用来接收账号和真正使用账号信息的。对于设备上来说使用账号的场景也应该遵守若干规范。

5.3.1 第一步：授权

调用SDK的AuthManager的setClientID接口，注册授权回调接口，完成TVS的授权功能。

当采用第三方账号授权方式且无手机APP参与流程的时候，ClientID生成需要按照以下格式完成："ThirdParty","",厂商id,厂商token,"", "",ProductID,dsn。

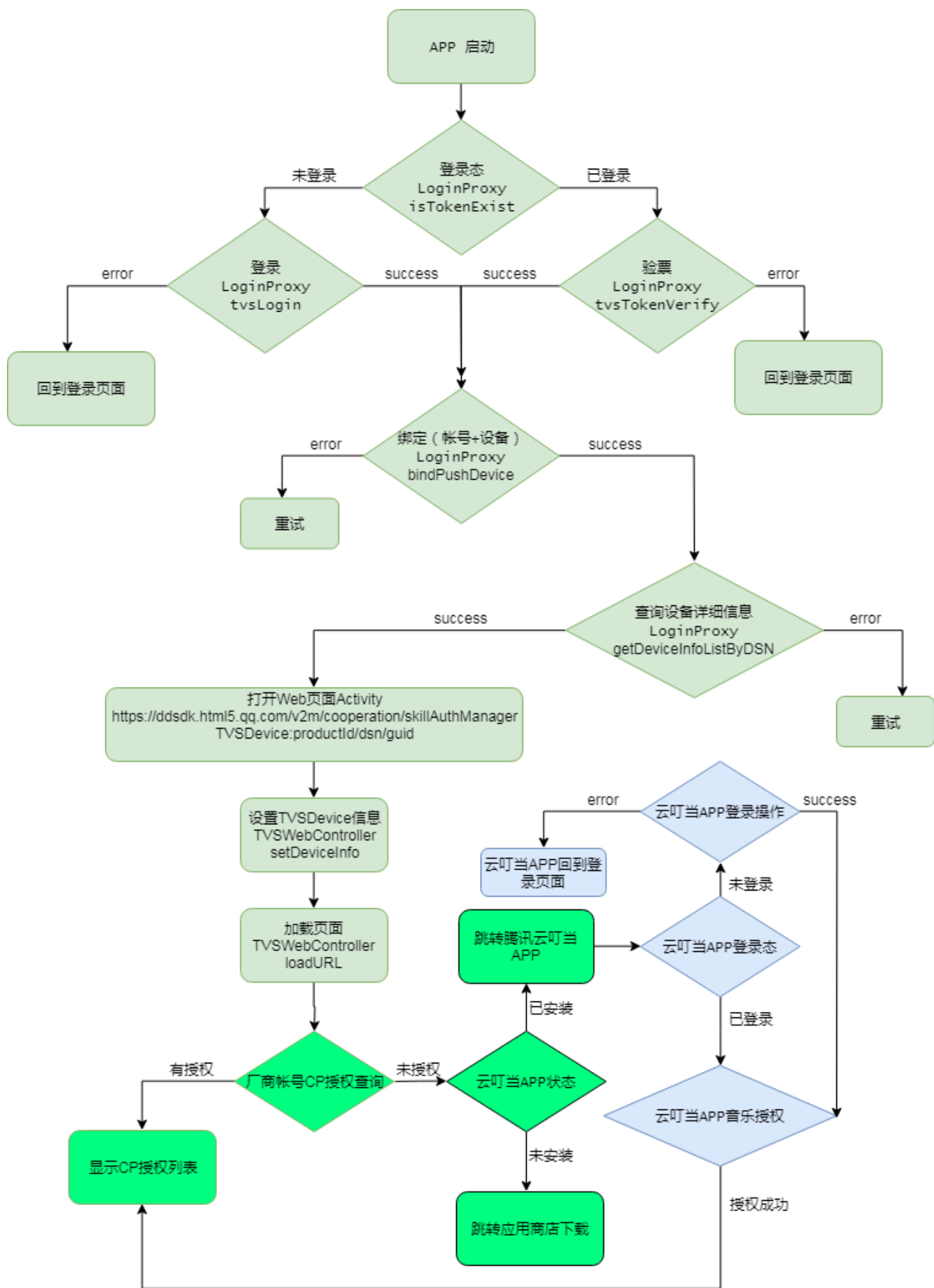
当采用访客账号，即无手机APP参与流程时，请使用带productID和dsn参数的ClientID生成方法完成全逻辑。

5.3.2 第二步：TVSToken的获取

通过授权或刷票获取的最新的TVSToken需要缓存起来，访问服务的时候这是重要凭证信息。

5.4 音乐授权

无屏授权方案目前QQ登录的话可以直接打通与QQ音乐的数据，微信的话需要拉起云小微APP来授权，具体授权流程如下



5.4.1 第一步：厂商APP登录

通过DMSDK相关登录、刷票接口获取厂商APP自身的登录态

5.4.2 第二步：厂商账号绑定

将厂商账号绑定设备相关信息（ProductID/DSN），并通过获取绑定设备详情信息获取到设备的GUID信息

5.4.3 第三步：音乐授权管理页面

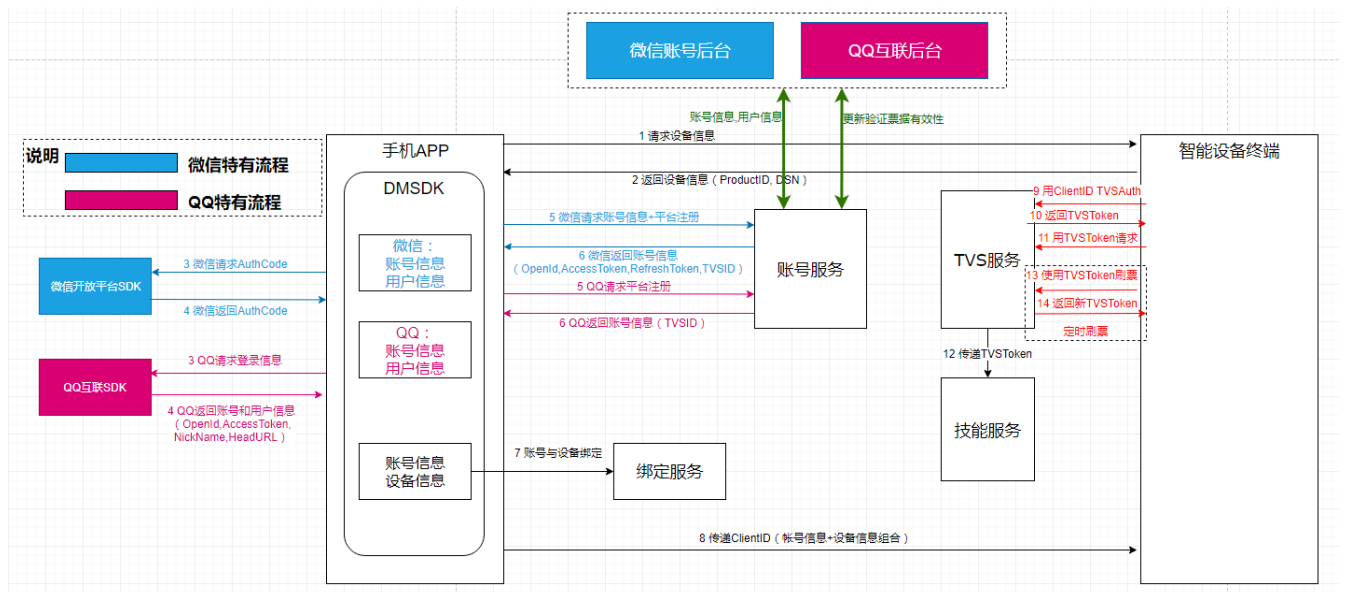
将以上信息传入云小微标准的音乐授权管理页面中。如果未进行过音乐授权，则弹出腾讯云小微APP的下载和跳转引导页面；如果已完成授权，则跳转到CP授权列表页面，在该页面可以查看授权用户，并可以解除某一授权用户。

5.4.4 第四步：跳转云小微APP完成授权

跳转到云小微APP，完成音乐账号的授权

6 无屏方案（TVSAPI）

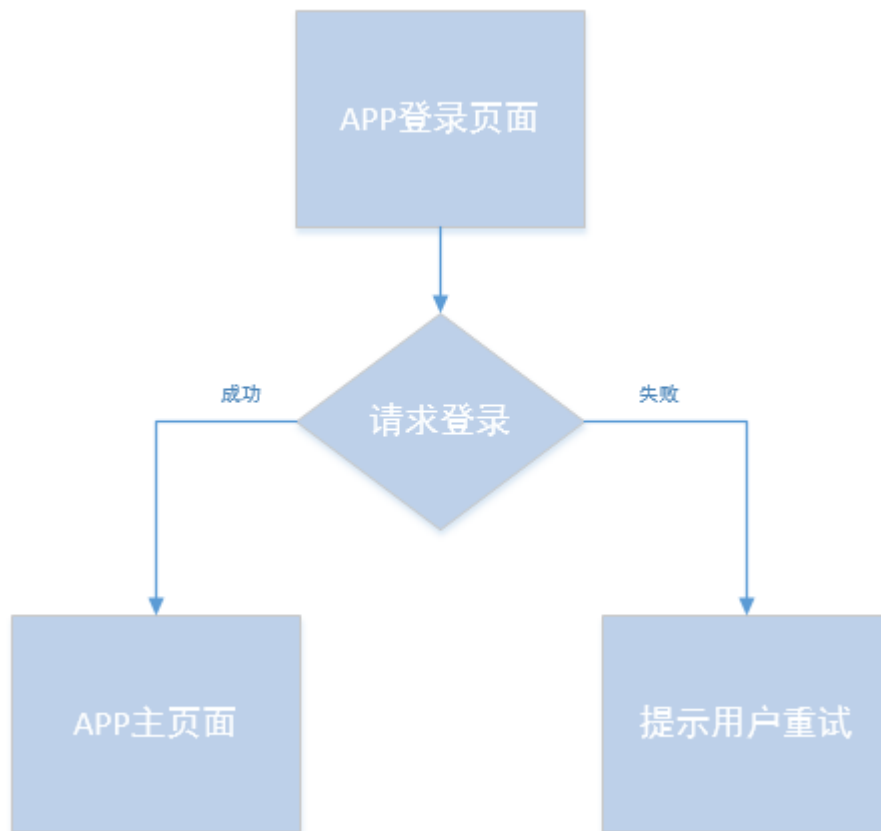
6.1 ER流程图



6.2 手机端

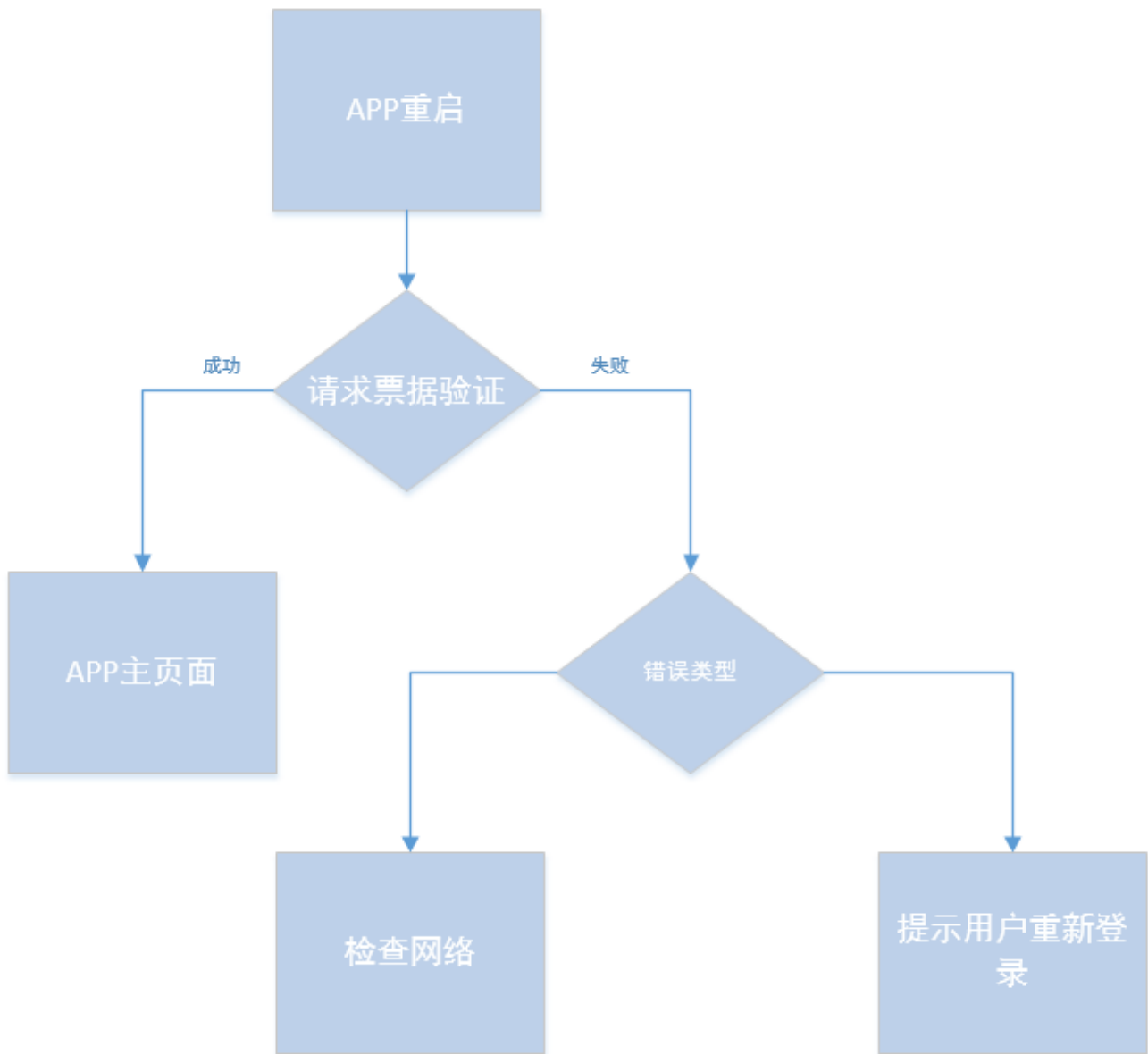
6.2.1 第一步：账号登录

手机APP在初始化状态下，应该是无登录态的，DMSDK中提供了针对微信和QQ的授权接口tvLogin可以通过拉起微信或手Q的授权页面来给厂商APP提供授权能力，根据回调成功或失败接口来进行下一步操作，例如成功后带上登录信息和用户信息进入下一个页面，失败后就通过强提示功能提示用户重新登录。



6.2.2 第二步：账号验证

手机APP在有登录态的场景下，进入后，必须调用requestTokenVerify进行票据验证。成功后带上最新的登录信息和用户信息进行正常操作，失败后通过强提示功能：清除本地的账号信息缓存并跳转到登录页面要求用户重新登录

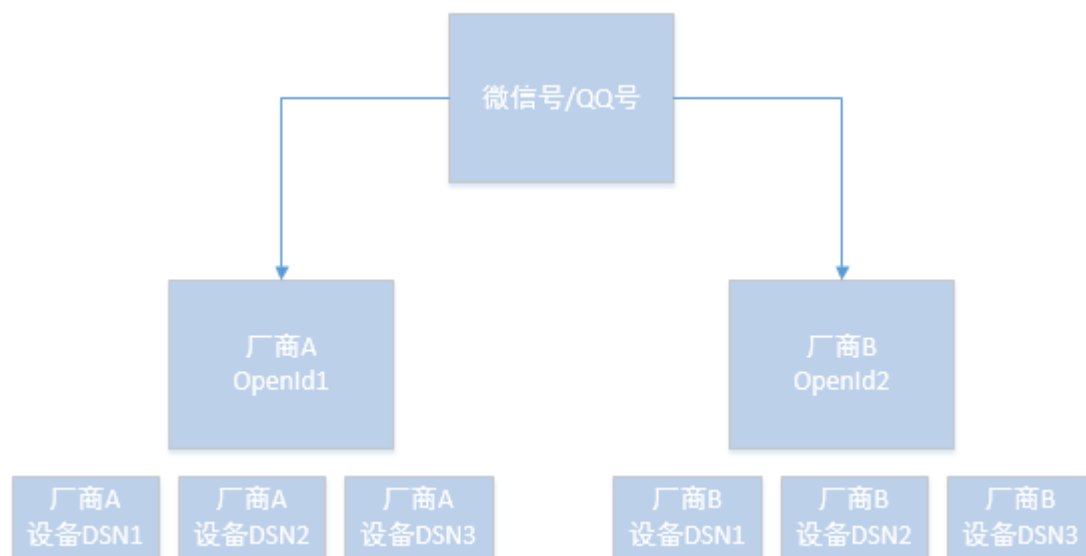


6.2.3 第三步：账号设备绑定

需要完成账号信息与设备信息的绑定，对于支付、IoT、闹钟提醒、推送等场景需要校验云端绑定态。

对于绑定，解绑时填写的字段有一定的要求，具体可详见DMSDK API文档。

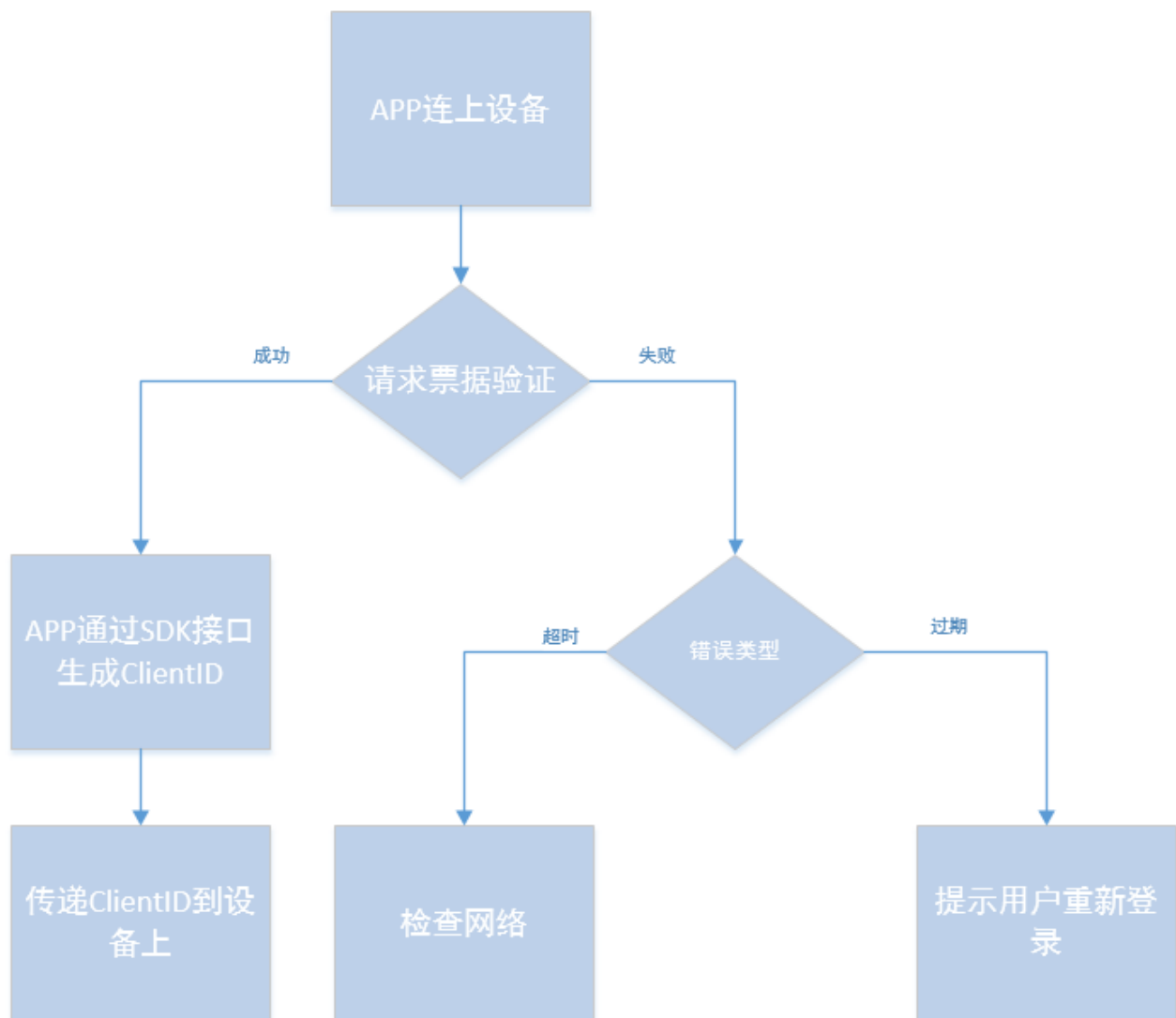
如果是非微信或QQ授权类型的，请按照DMSDK API文档，传入第三方账号字段进行绑定、解绑。



6.2.4 第四步：授权信息生成

授权信息ClientID是手机到设备的需要传递的唯一账号数据信息，并且只有刷票成功才可以去生成ClientID。

如果是非微信或QQ授权类型的，请按照DMSDK API文档，传入第三方账号字段进行ClientID的获取。



6.2.5 第五步：授权信息ClientID传递

使用蓝牙、局域网方式，或通用设备通信协议完成，该部分为厂商自行实现

6.3 设备端

```
// TVSAPI接入方式正式环境授权/验票API  
https://tvs.htm15.qq.com/auth/o2/token  
// TVSAPI接入方式体验环境授权/验票API  
https://tvsexp.htm15.qq.com/auth/o2/token  
// TVSAPI接入方式测试环境授权/验票API  
https://tvstest.htm15.qq.com/auth/o2/token
```

6.3.1 第一步：授权

当采用第三方账号授权方式且无手机APP参与流程的时候，ClientID生成需要按照以下格式完成："ThirdParty","",厂商id,厂商token,"","",ProductID,dsn。

当采用访客账号，即无手机APP参与流程时，请使用以下编码格式完成ClientID生成。

```
//终端的ProductId
string strProductId;
//终端的Dsn
string strDsn;
//拼接文本
string strPlain = strProductId + strDsn;
/*生成终端ClientId
"ENCRYPT:0001","0001","MD5"是魔数，不能改变（包括大小写）
md5表示采用MD5算法，生成32字节的字符串
upper表示将字符串转换成大写
*/
string strClientId = "ENCRYPT:0001," + upper(md5(upper(md5(strPlain + "0001"))) + "MD5")) +
", " + strProductId + ", " + strDsn;
```

6.3.1.1 接口描述

设备上拿到手机传过来的ClientId，去TVS后台进行鉴权操作，换取真正用于服务合法访问的TVSToken，对于无屏设备来说TVS鉴权失败的话，就要进行重试操作，或是以语音方式提示用户检查网络，对于有屏设备来说失败的话，也是需要重试操作，或是以界面UI的形式拉起网络设置页面，让用户检查当前设备连接的网络是否正常。

URL：POST https://tvs.html5.qq.com/auth/o2/token

6.3.1.2 请求数据

```
{
  "grant_type": "authorization_code",
  "code": "{{STRING}}",
  "redirect_uri": "{{STRING}}",
  "client_id": "{{STRING}}",
  "code_verifier": "{{STRING}}"
}
```

Parameter	Type	必选	描述
grant_type	string	Yes	请求类型
code	string	Yes	获取票据授权Code
redirect_uri	string	Yes	重定向URL
client_id	string	Yes	设备ID
code_verifier	string	Yes	获取票据校验码

6.3.1.3 返回参数

响应状态码说明

(1) HTTP 200且数据完整

终端授权成功，接下来终端发起建立下行通道请求

(2) 其他情况(包括无响应)

终端提示用户重新通过APP登录，并重新发起授权请求

响应数据

```
{
  "access_token": "{{STRING}}",
  "refresh_token": "{{STRING}}",
  "token_type": "bearer",
  "expires_in": LONG
}
```

Parameter	Type	必选	描述
access_token	string	Yes	访问票据
refresh_token	string	Yes	刷票票据
token_type	string	Yes	访问票据类型
expires_in	long	Yes	访问票据有效时间

6.3.2 第二步：定时刷票

6.3.2.1 接口描述

使用返回的过期时间，在终端启动定时器，定时执行刷票操作，防止使用过期的票据请求失效。

URL：POST https://tvs.html5.qq.com/auth/o2/token

6.3.2.2 请求参数

```
{
  "grant_type": "refresh_token",
  "client_id": "{{STRING}}",
  "refresh_token": "{{STRING}}"
}
```

Parameter	Type	必选	描述
grant_type	string	Yes	请求类型
client_id	string	Yes	设备ID
refresh_token	string	Yes	刷票票据

6.3.2.3 返回参数

响应状态码说明

(1) HTTP 200且数据完整

终端刷票成功，接下来终端发起建立下行通道请求

(2) 其他情况(包括无响应)

终端重试(次数由终端控制)，若重试后仍然不成功，则提示用户重新通过APP登录，并重新发起授权请求

响应数据

```
{
  "access_token": "{{STRING}}",
  "refresh_token": "{{STRING}}",
  "token_type": "bearer",
  "expires_in": LONG
}
```

Parameter	Type	必选	描述
access_token	string	Yes	访问票据
refresh_token	string	Yes	刷票票据
token_type	string	Yes	访问票据类型
expires_in	long	Yes	访问票据有效时间

6.3.3 第三步：启动刷票

在设备端有票据登录态的场景下，执行刷票操作，防止使用过期的票据请求失效。

URL：POST https://tvs.htm15.qq.com/auth/o2/token

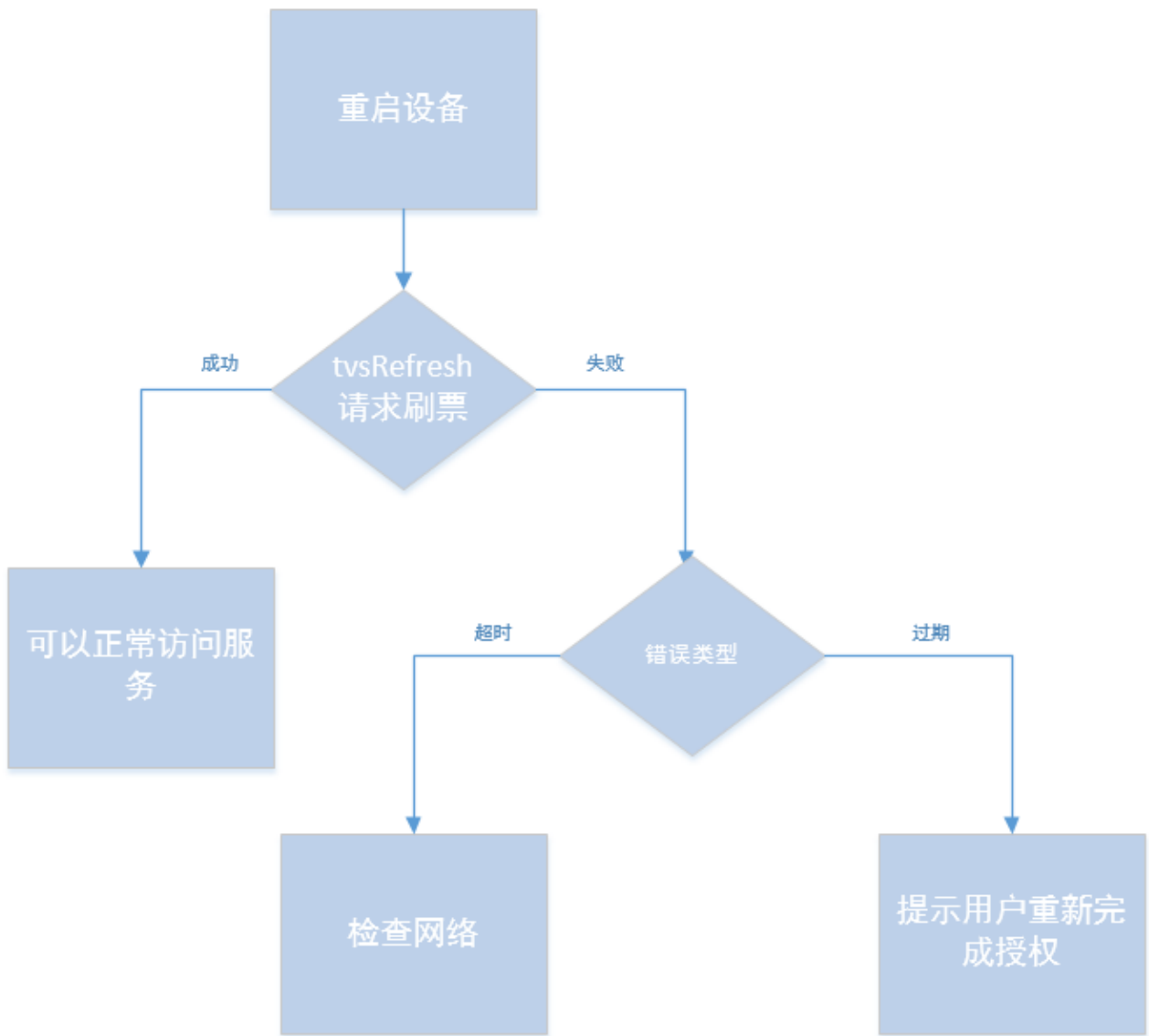
接口描述与参数同定时刷票

6.3.4 第四步：网络切换刷票

在设备端开机状态下网络状态从无到有的场景下，执行刷票操作，防止使用过期的票据请求失效。

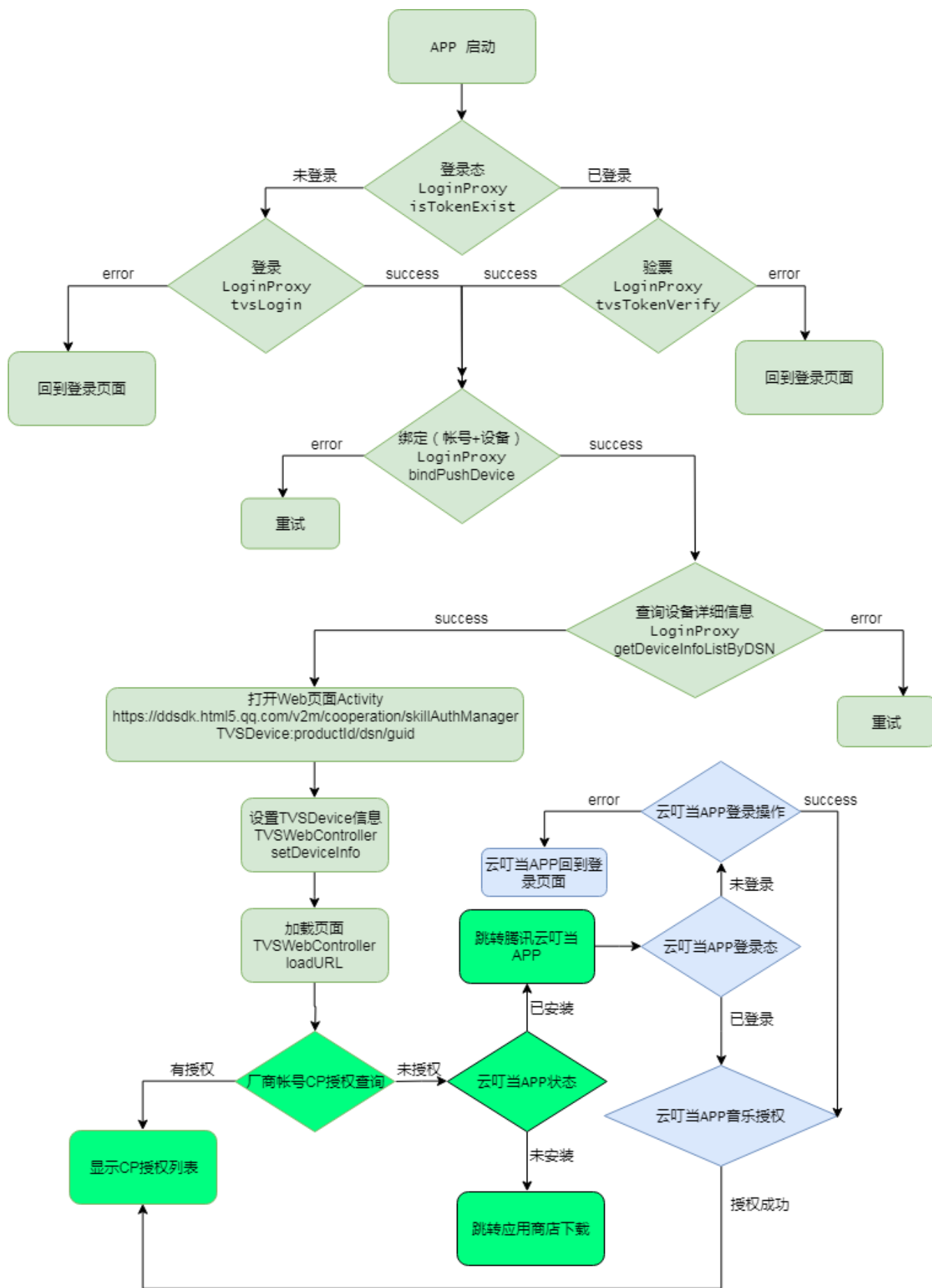
URL：POST https://tvs.htm15.qq.com/auth/o2/token

接口描述与参数同定时刷票



6.4 音乐授权

无屏授权方案目前QQ登录的话可以直接打通与QQ音乐的数据，微信的话需要拉起云小微APP来授权，具体授权流程如下



6.4.1 第一步：厂商APP登录

通过DMSDK相关登录、刷票接口获取厂商APP自身的登录态

6.4.2 第二步：厂商账号绑定

将厂商账号绑定设备相关信息（ProductID/DSN），并通过获取绑定设备详情信息获取到设备的GUID信息

6.4.3 第三步：音乐授权管理页面

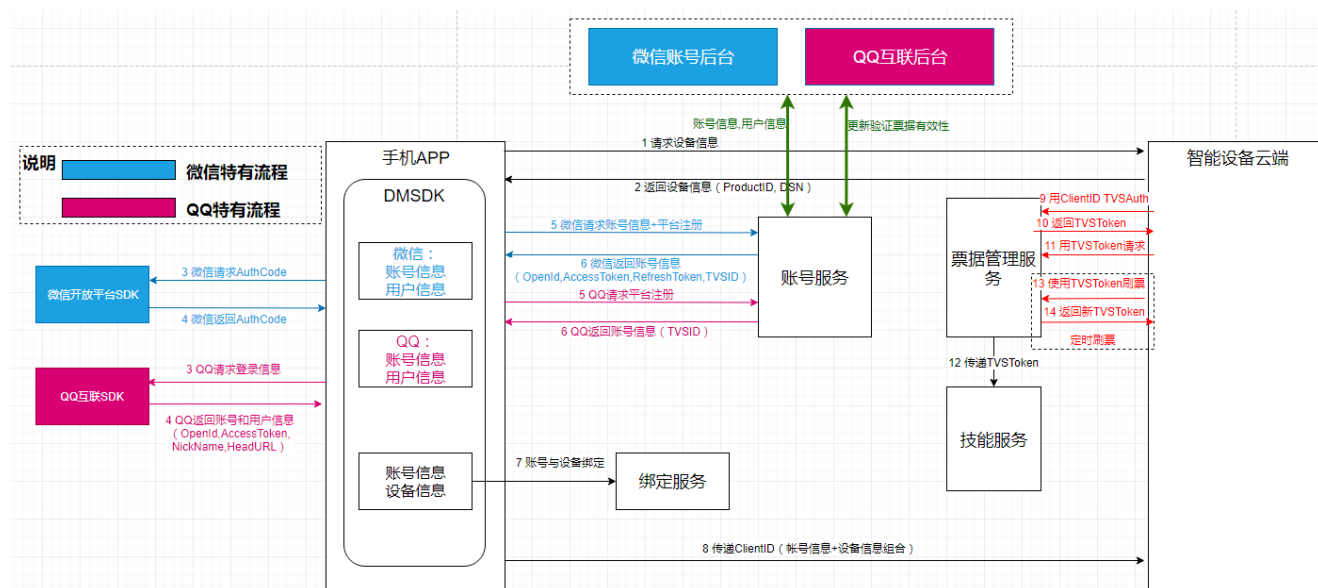
将以上信息传入云小微标准的音乐授权管理页面中。如果未进行过音乐授权，则弹出腾讯云小微APP的下载和跳转引导页面；如果已完成授权，则跳转到CP授权列表页面，在该页面可以查看授权用户，并可以解除某一授权用户。

6.4.4 第四步：跳转云小微APP完成授权

跳转到云小微APP，完成音乐账号的授权

7 无屏方案（基础API）

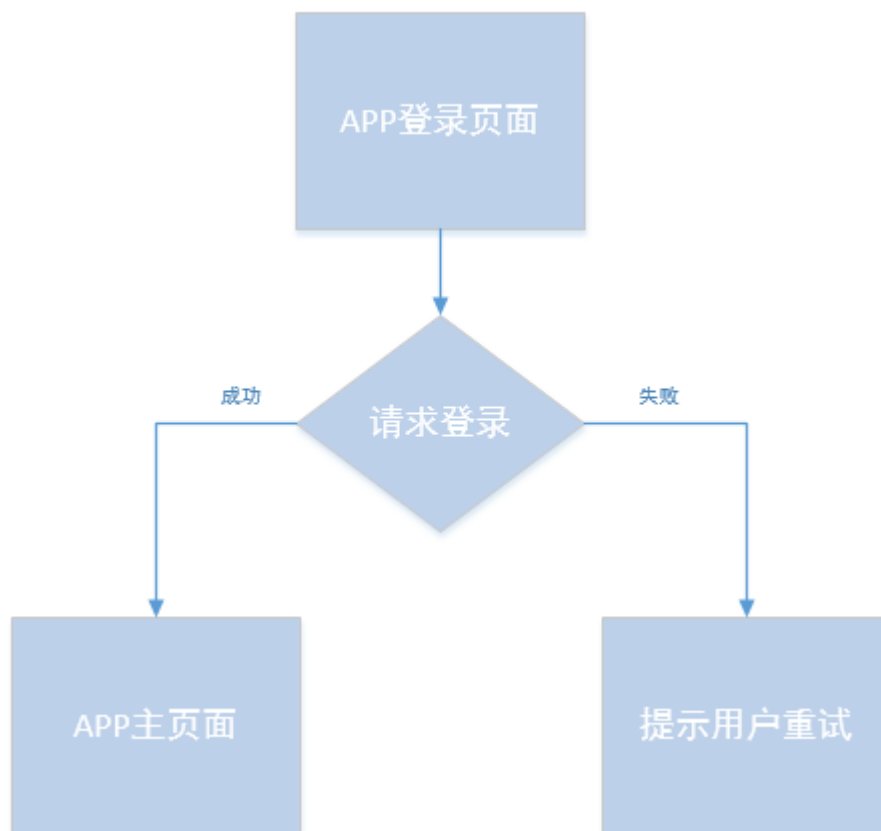
7.1 ER流程图



7.2 手机端

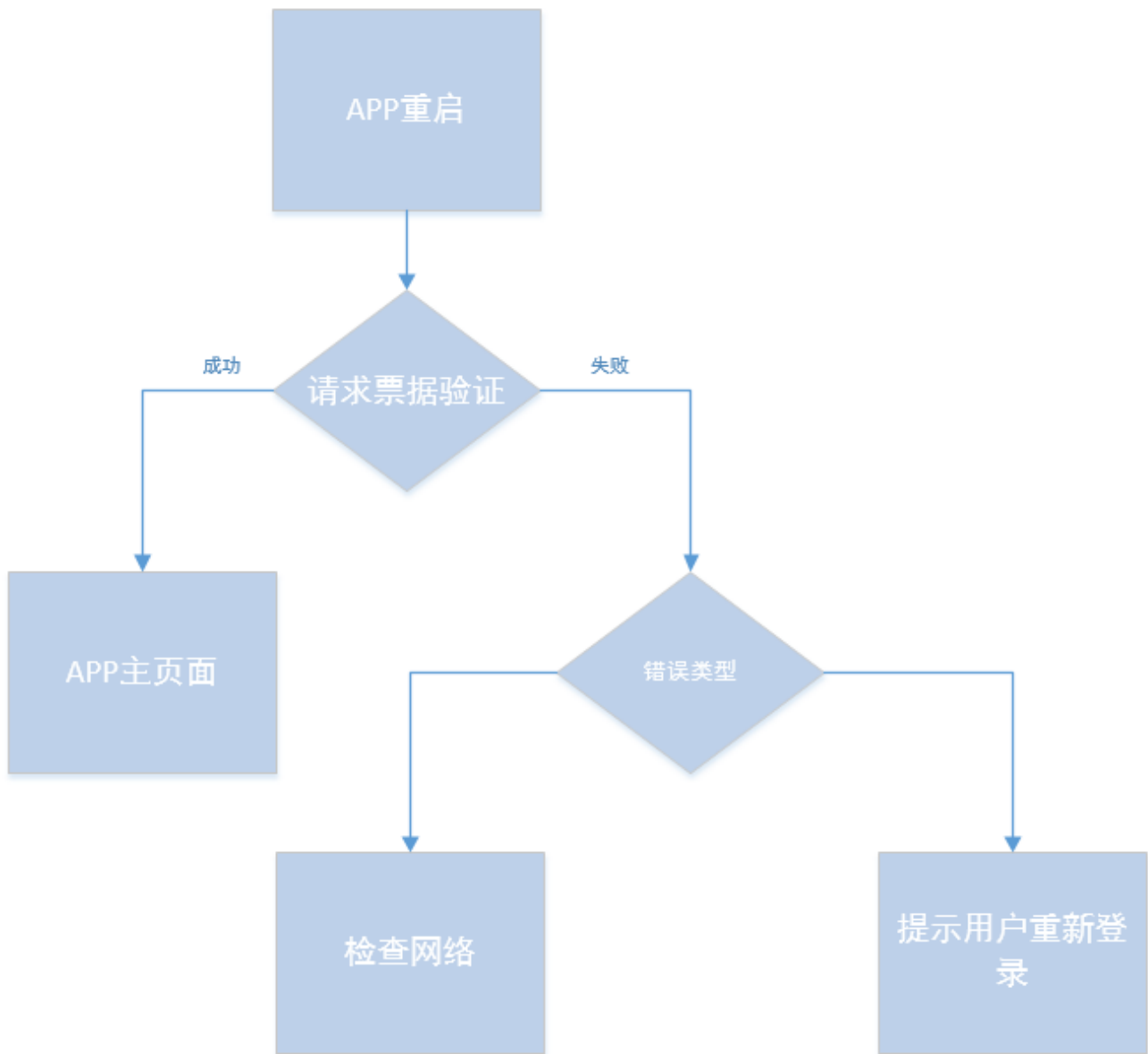
7.2.1 第一步：账号登录

手机APP在初始化状态下，应该是无登录态的，DMSDK中提供了针对微信和QQ的授权接口tvslLogin可以通过拉起微信或手Q的授权页面来给厂商APP提供授权能力，根据回调成功或失败接口来进行下一步操作，例如成功后带上登录信息和用户信息进入下一个页面，失败后就通过强提示功能提示用户重新登录。



7.2.2 第二步：账号验证

手机APP在有登录态的场景下，进入后，必须调用requestTokenVerify进行票据验证。成功后带上最新的登录信息和用户信息进行正常操作，失败后通过强提示功能：清除本地的账号信息缓存并跳转到登录页面要求用户重新登录

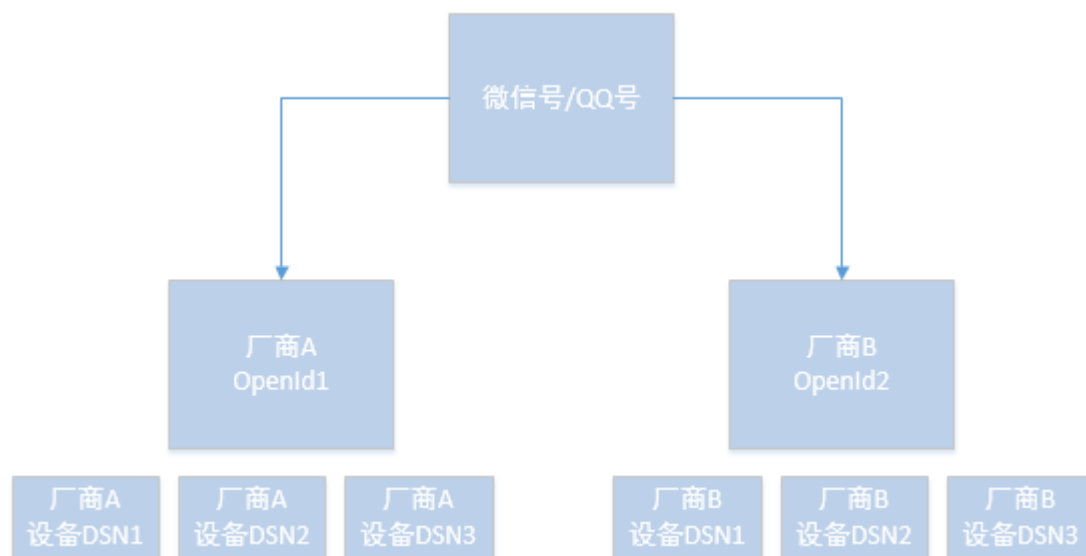


7.2.3 第三步：账号设备绑定

需要完成账号信息与设备信息的绑定，对于支付、IoT、闹钟提醒、推送等场景需要校验云端绑定态。

对于绑定，解绑时填写的字段有一定的要求，具体可详见DMSDK API文档。

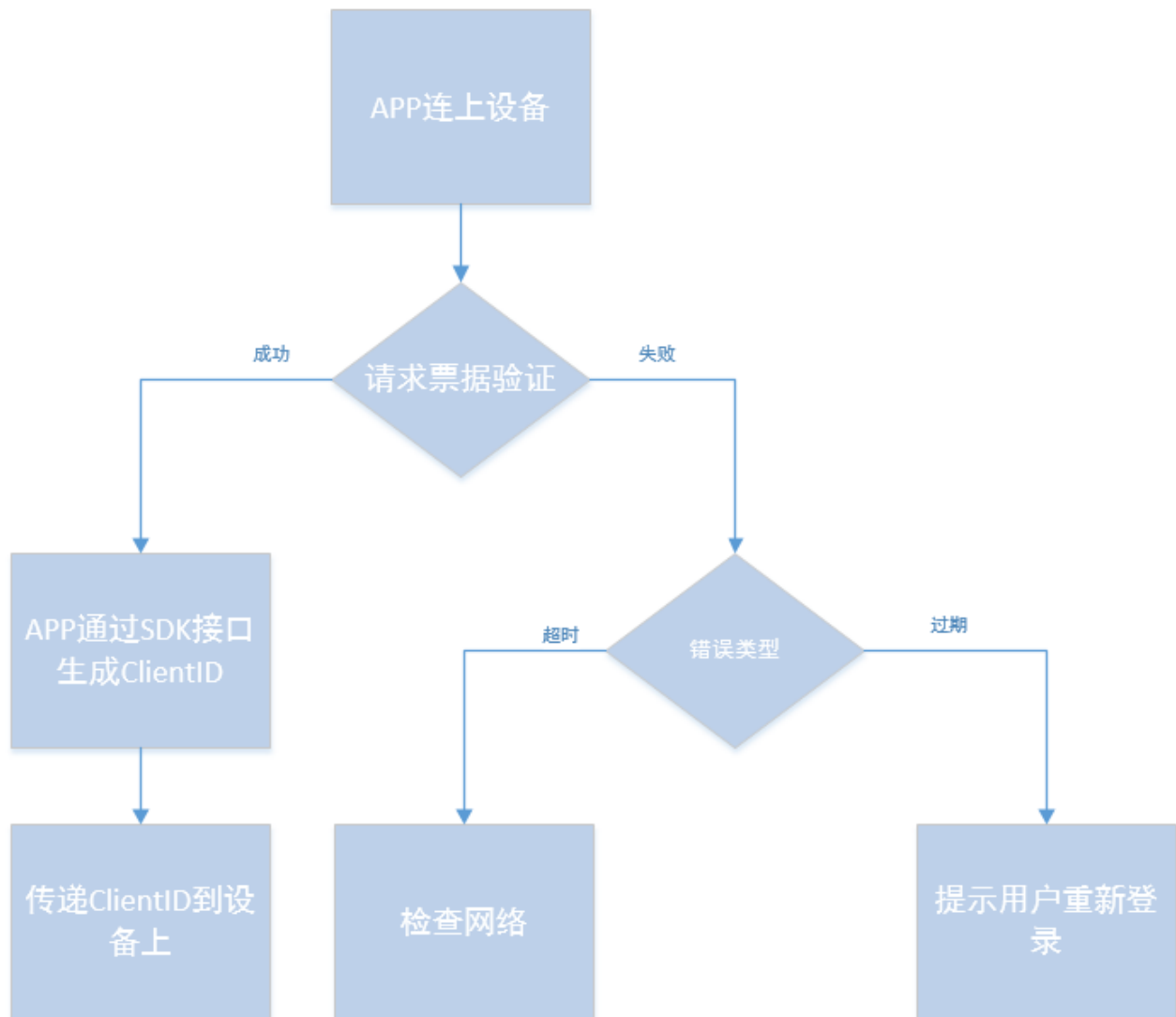
如果是非微信或QQ授权类型的，请按照DMSDK API文档，传入第三方账号字段进行绑定、解绑。



7.2.4 第四步：授权信息生成

授权信息ClientID是手机到设备的需要传递的唯一账号数据信息，并且只有刷票成功才可以去生成ClientID。

如果是非微信或QQ授权类型的，请按照DMSDK API文档，传入第三方账号字段进行ClientID的获取。



7.2.5 第五步：授权信息ClientID传递

使用蓝牙、局域网方式，或通用设备通信协议完成，该部分为厂商自行实现

7.3 厂商云端

```
// 基础API方式正式环境授权API
https://aiwx.html5.qq.com/api/v1/account/authorize
// 基础API方式正式环境验票API
https://aiwx.html5.qq.com/api/v1/account/refresh

// 基础API方式体验环境授权API
https://aiwx.html5.qq.com/exapi/v1/account/authorize
// 基础API方式体验环境验票API
https://aiwx.html5.qq.com/exapi/v1/account/refresh

// 基础API方式测试环境授权API
https://aiwx.html5.qq.com/testapi/v1/account/authorize
// 基础API方式测试环境验票API
https://aiwx.html5.qq.com/testapi/v1/account/refresh
```

7.3.1 第一步：授权

当采用第三方账号授权方式且无手机APP参与流程的时候，ClientId生成需要按照以下格式完成："ThirdParty","",厂商id,厂商token,"", "",ProductID,dsn。

当采用访客账号，即无手机APP参与流程时，请使用以下编码格式完成ClientId生成。

```
//终端的ProductId
string strProductId;
//终端的Dsn
string strDsn;
//拼接文本
string strPlain = strProductId + strDsn;
/*生成终端ClientId
"ENCRYPT:0001","","0001","MD5"是魔数，不能改变（包括大小写）
md5表示采用MD5算法，生成32字节的字符串
upper表示将字符串转换成大写
*/
string strClientId = "ENCRYPT:0001," + upper(md5(upper(md5(strPlain + "0001"))) + "MD5")) +
"," + strProductId + "," + strDsn;
```

7.3.1.1 接口描述

提供终端ClientId票据授权接口。

调用时机：设备侧第一次拿到手机端传过来的ClientId，需要调用本接口换取对应的 tvsRefreshToken 和 authorization，tvsRefreshToken 用于刷票接口，authorization 内部有账号与设备信息，调用其他API请求接口时，把 authorization 填入 header.user.authorization。如果填入 header.user.authorization，那么 header.guid,header.user 其他字段都不需要填写。

7.3.1.2 请求参数

URL：POST https://aiwx.html5.qq.com/api/v1/account/authorize

```
{
  "header": {
    "qua": "{{STRING}}",
    "ip": "8.8.8.8",
    "device": {
      "network": "4G"
    }
  },
  "payload": {
    "clientId": "{{STRING}}"
  }
}
```

参数名	类型	是否必选	描述
header	-	是	请求头
header.qua	string	是	设备及应用信息，详细说明见 附录-QUA字段说明
header.ip	string	否	终端IP，仅限云服务使用。终端直接调用本节课不要填这个字段。
header.device	-	否	
header.device.network	string	否	网络类型：4G / 3G / 2G / Wi-Fi
payload	-	是	请求内容
payload.clientId	-	是	DMSDK ClientId

7.3.1.3 返回参数

```
{
  "header": {
    "retCode": 0,
    "errMsg": "{{STRING}}"
  },
  "payload": {
    "tvsRefreshToken": "{{STRING}}",
    "authorization": "{{STRING}}",
    "expiredTimeInSeconds": {{INT}}
  }
}
```

Header Parameters

参数名	类型	是否必选	描述
header	object	Yes	头部
header.retCode	long	Yes	返回码。如果错误码不等于0且大于-1000000，可以认为票据无效。
header.errMsg	string	Yes	错误消息

Payload Parameters

参数名	类型	是否必选	描述
<code>payload</code>	<code>object</code>	Yes	负载
<code>payload.tvRefreshToken</code>	<code>string</code>	Yes	tvRefreshToken，刷票用
<code>payload.authorization</code>	<code>string</code>	Yes	账户信息。用在其他请求的 <code>header.user.authorization</code> 中。
<code>payload.expiredTimeInSeconds</code>	<code>int</code>	Yes	票据过期时间(秒)。到达过期时间，需要调用7.8刷票接口重新刷票。

7.3.2 第二步：定时刷票

7.3.2.1 接口描述

提供终端刷票接口。根据授权返回的payload.expiredTimeInSeconds字段，启动云端定时刷票

7.3.2.2 请求参数

URL：POST `https://aiwx.html5.qq.com/api/v1/account/refresh`

```
{
  "header": {
    "qua": "{{STRING}}",
    "ip": "8.8.8.8",
    "device": {
      "network": "4G"
    }
  },
  "payload": {
    "tvRefreshToken": "{{STRING}}"
  }
}
```

参数名	类型	是否必选	描述
header	-	是	请求头
header.qua	string	是	设备及应用信息，详细说明见 附录-QUA字段说明
header.ip	string	否	终端IP，仅限云服务使用。终端直接调用本接口不要填这个字段。
header.device	-	否	
header.device.network	string	否	网络类型：4G / 3G / 2G / wi-Fi
payload	-	是	请求内容
payload.tvsRefreshToken	-	是	TVS刷票Token

7.3.2.3 返回参数

```
{
  "header": {
    "retCode": 0,
    "errMsg": "{{STRING}}"
  },
  "payload": {
    "tvsRefreshToken": "{{STRING}}",
    "authorization": "{{STRING}}",
    "expiredTimeInSeconds": {{INT}}
  }
}
```

Header Parameters

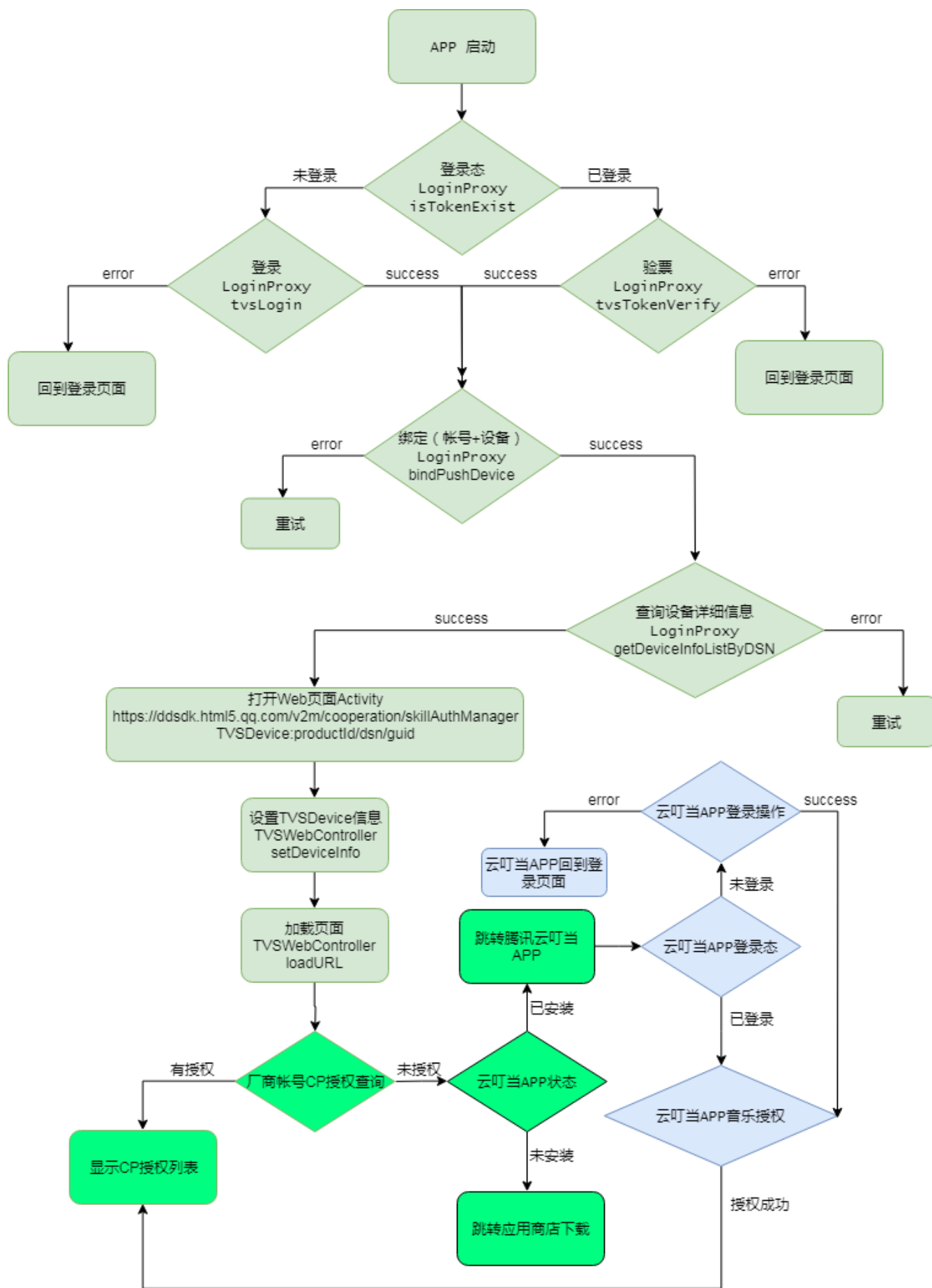
参数名	类型	是否必选	描述
header	object	Yes	头部
header.retCode	long	Yes	返回码。如果错误码不等于0且大于-1000000，可以认为票据无效。
header.errMsg	string	Yes	错误消息

Payload Parameters

参数名	类型	是否必选	描述
<code>payload</code>	<code>object</code>	Yes	负载
<code>payload.tvRefreshToken</code>	<code>string</code>	Yes	最新tvRefreshToken，下次刷票使用。
<code>payload.authorization</code>	<code>string</code>	Yes	最新账户信息。每次请求7.1-7.6的接口，都需要用最新账户信息。
<code>payload.expiredTimeInSeconds</code>	<code>int</code>	Yes	过期时间 (秒)。到达过期时间，需要重新刷票。

7.4 音乐授权

无屏授权方案目前QQ登录的话可以直接打通与QQ音乐的数据，微信的话需要拉起云小微APP来授权，具体授权流程如下



7.4.1 第一步：厂商APP登录

通过DMSDK相关登录、刷票接口获取厂商APP自身的登录态

7.4.2 第二步：厂商账号绑定

将厂商账号绑定设备相关信息（ProductID/DSN），并通过获取绑定设备详情信息获取到设备的GUID信息

7.4.3 第三步：音乐授权管理页面

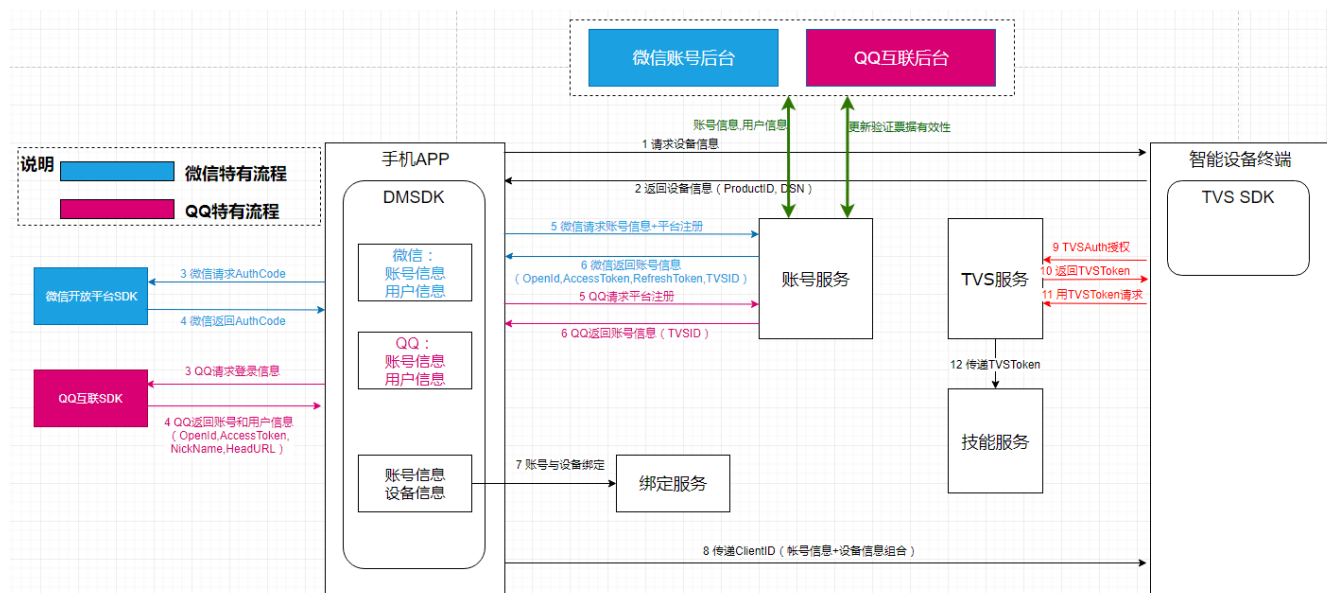
将以上信息传入云小微标准的音乐授权管理页面中。如果未进行过音乐授权，则弹出腾讯云小微APP的下载和跳转引导页面；如果已完成授权，则跳转到CP授权列表页面，在该页面可以查看授权用户，并可以解除某一授权用户。

7.4.4 第四步：跳转云小微APP完成授权

跳转到云小微APP，完成音乐账号的授权

8 有屏方案（TVSSDK）

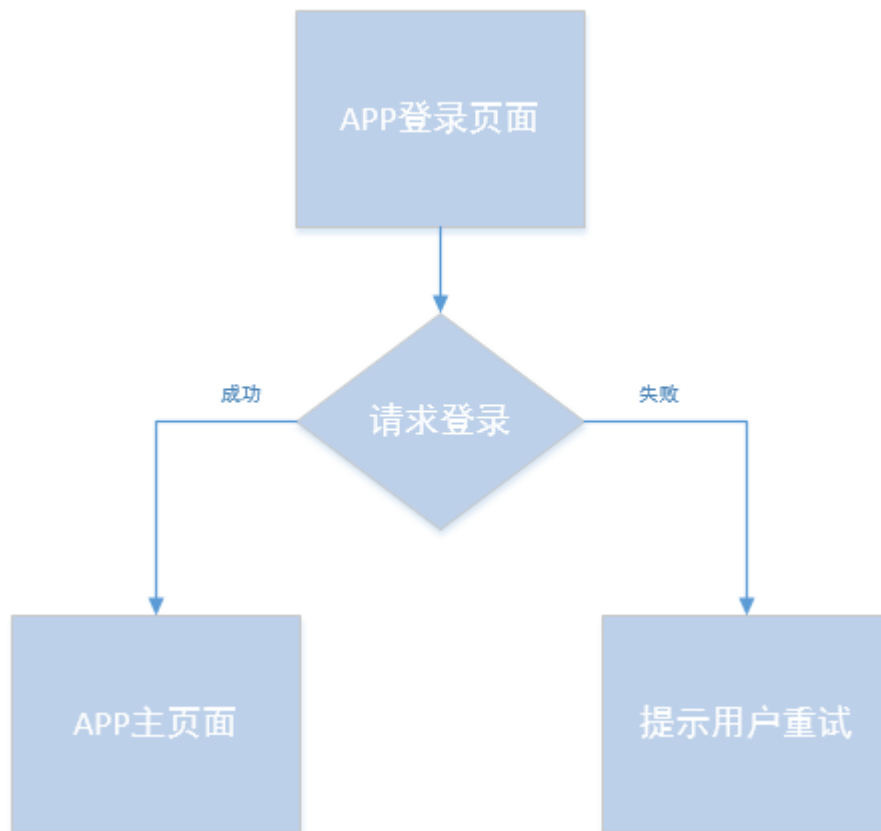
8.1 ER流程图



8.2 手机端

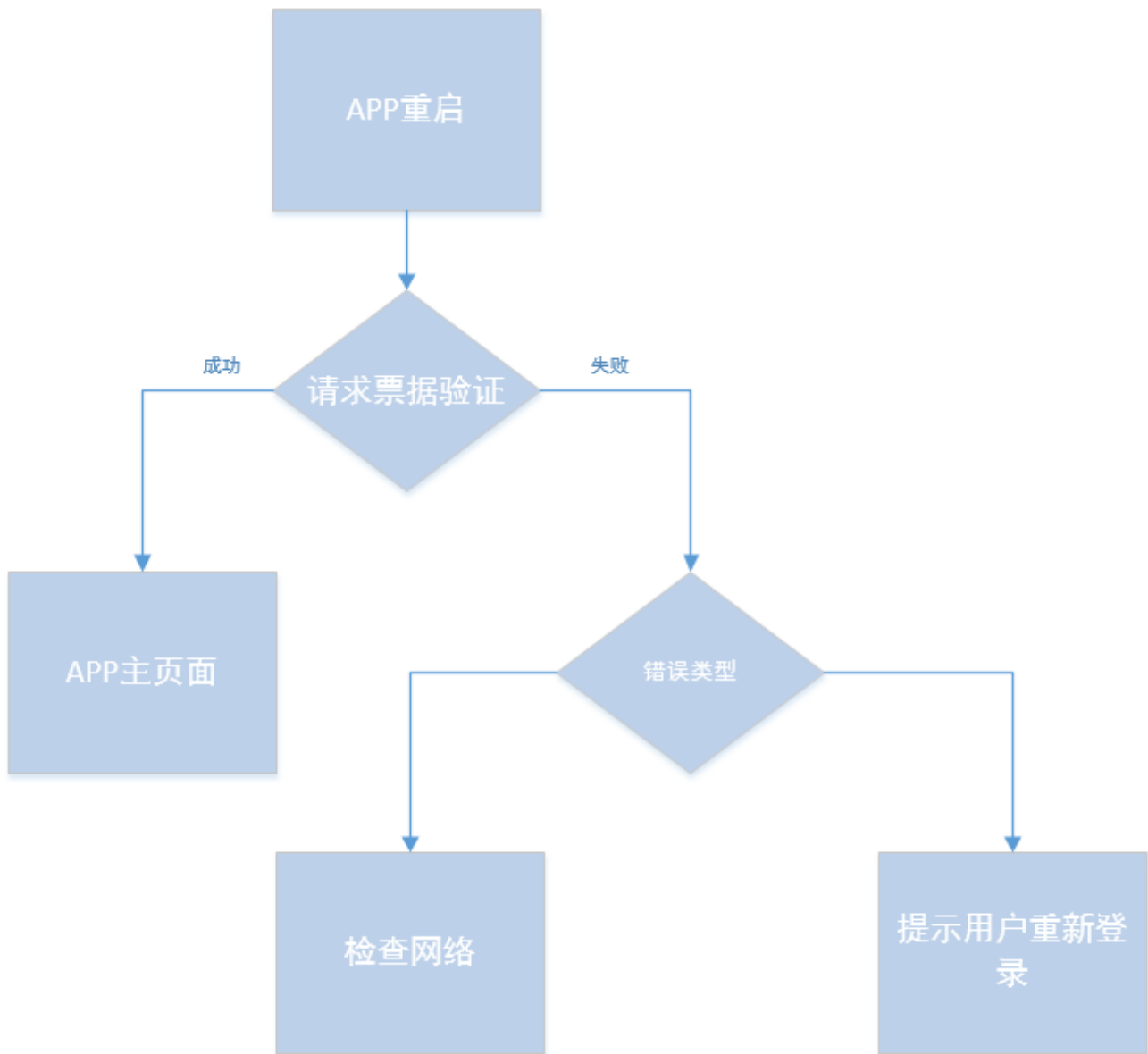
8.2.1 第一步：账号登录

手机APP在初始化状态下，应该是无登录态的，DMSDK中提供了针对微信和QQ的授权接口tvsLogin可以通过拉起微信或手Q的授权页面来给厂商APP提供授权能力，根据回调成功或失败接口来进行下一步操作，例如成功后带上登录信息和用户信息进入下一个页面，失败后就通过强提示功能提示用户重新登录。



8.2.2 第二步：账号验证

手机APP在有登录态的场景下，进入后，必须调用requestTokenVerify进行票据验证。成功后带上最新的登录信息和用户信息进行正常操作，失败后通过强提示功能：清除本地的账号信息缓存并跳转到登录页面要求用户重新登录

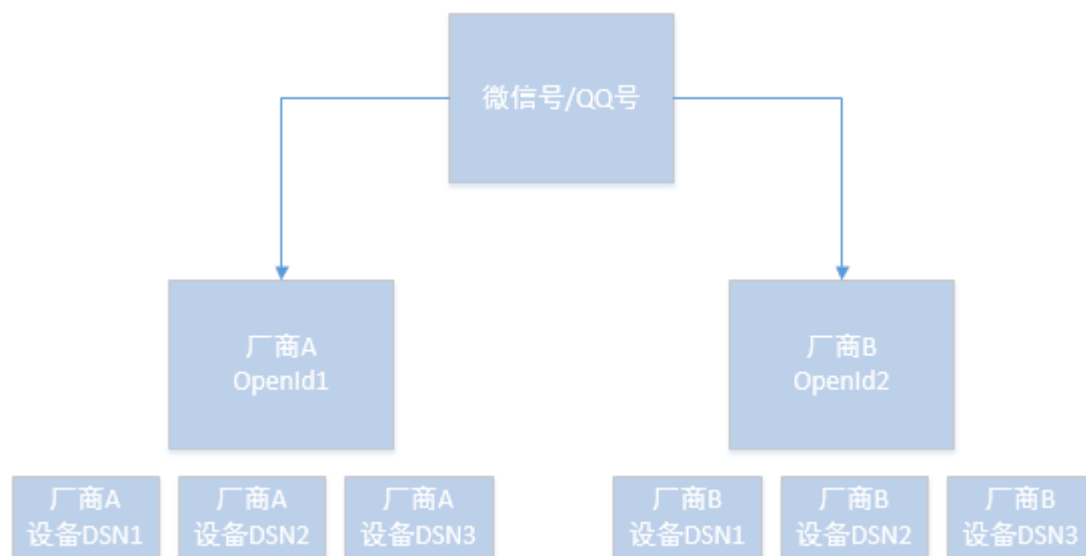


8.2.3 第三步：账号设备绑定

需要完成账号信息与设备信息的绑定，对于支付、IoT、闹钟提醒、推送等场景需要校验云端绑定态。

对于绑定，解绑时填写的字段有一定的要求，具体可详见DMSDK API文档。

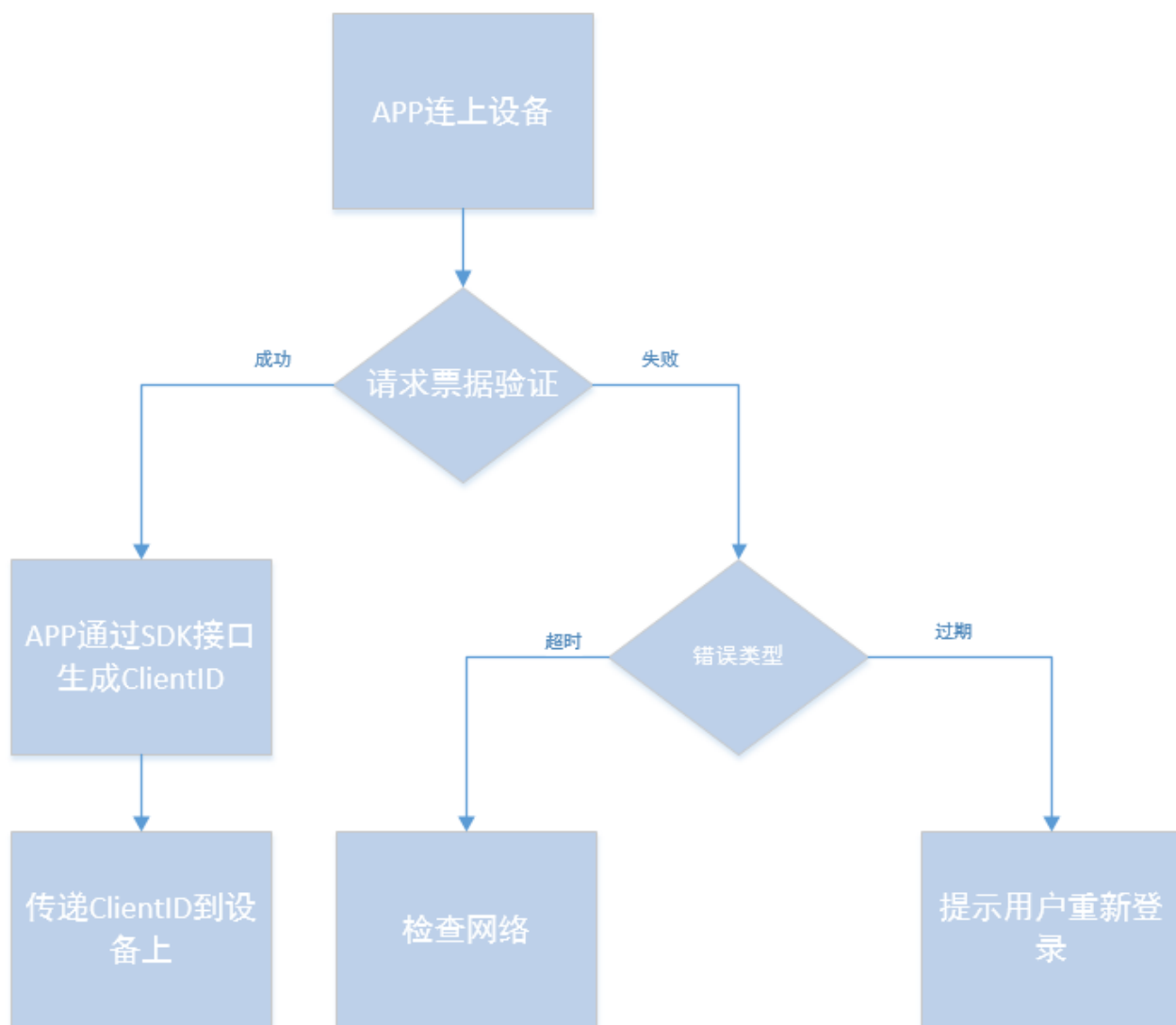
如果是非微信或QQ授权类型的，请按照DMSDK API文档，传入第三方账号字段进行绑定、解绑。



8.2.4 第四步：授权信息生成

授权信息ClientID是手机到设备的需要传递的唯一账号数据信息，并且只有刷票成功才可以去生成ClientID。

如果是非微信或QQ授权类型的，请按照DMSDK API文档，传入第三方账号字段进行ClientID的获取。



8.2.5 第五步：授权信息ClientID传递

使用蓝牙、局域网方式，或通用设备通信协议完成，该部分为厂商自行实现

8.3 设备端

设备上是用来接收账号和真正使用账号信息的。对于设备上来说使用账号的场景也应该遵守若干规范。

8.3.1 第一步：授权

调用SDK的AuthManager的setClientID接口，注册授权回调接口，完成TVS的授权功能。

当采用第三方账号授权方式且无手机APP参与流程的时候，ClientID生成需要按照以下格式完成："ThirdParty","",厂商id,厂商token,"", "",ProductID,dsn。

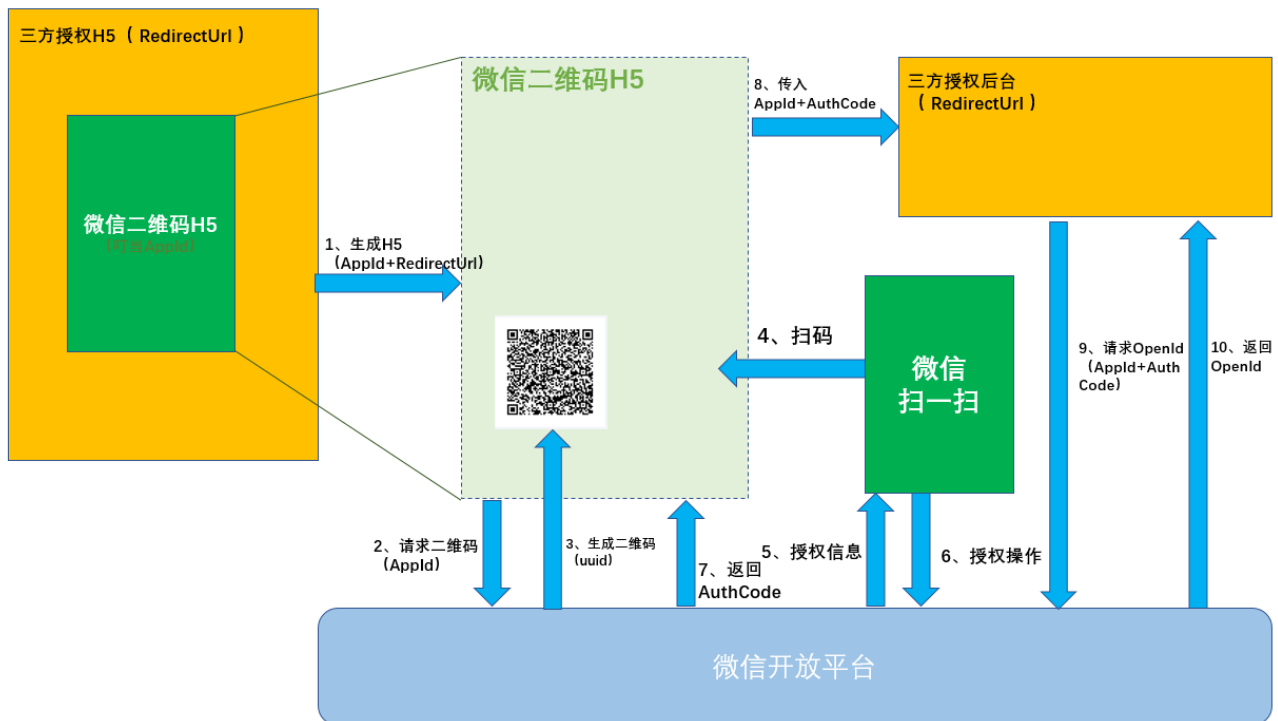
当采用访客账号，即无手机APP参与流程时，请使用带productID和dsn参数的ClientID生成方法完成全逻辑。

8.3.2 第二步：TVSToken的获取

通过授权或刷票获取的最新的TVSToken需要缓存起来，访问服务的时候这是重要凭证信息。

8.4 音乐授权

其原理就是在设备上弹出音乐二维码，通过手Q或微信扫码方式来进行授权，以微信为例其流程图如下。



注意：由于屏幕尺寸的差异，目前仅支持标准的屏幕分辨率的有屏方案，如非标准屏幕需要使用QQ音乐授权功能，请采用更加通用的云小微无屏授权方案或联系云小微客服。

8.4.1 第一步：TVSToken获取

通过设备端授权接口，使用ClientID来换取真正用来访问服务的TVSToken

8.4.2 第二步：接入音乐登录二维码页面

8.4.2.1 Android智能设备

接入tvswab sdk，将上一步获取到的TVSToken作为参数传入loadURL方法中，并打开音乐URL：<https://ddsdk.html5.qq.com/v2/opt/music/login>

8.4.2.2 Linux智能设备

集成浏览器，打开音乐URL：<https://ddsdk.html5.qq.com/v2/opt/music/login>，并将上一步获取到的TVSToken设置到cookie中。

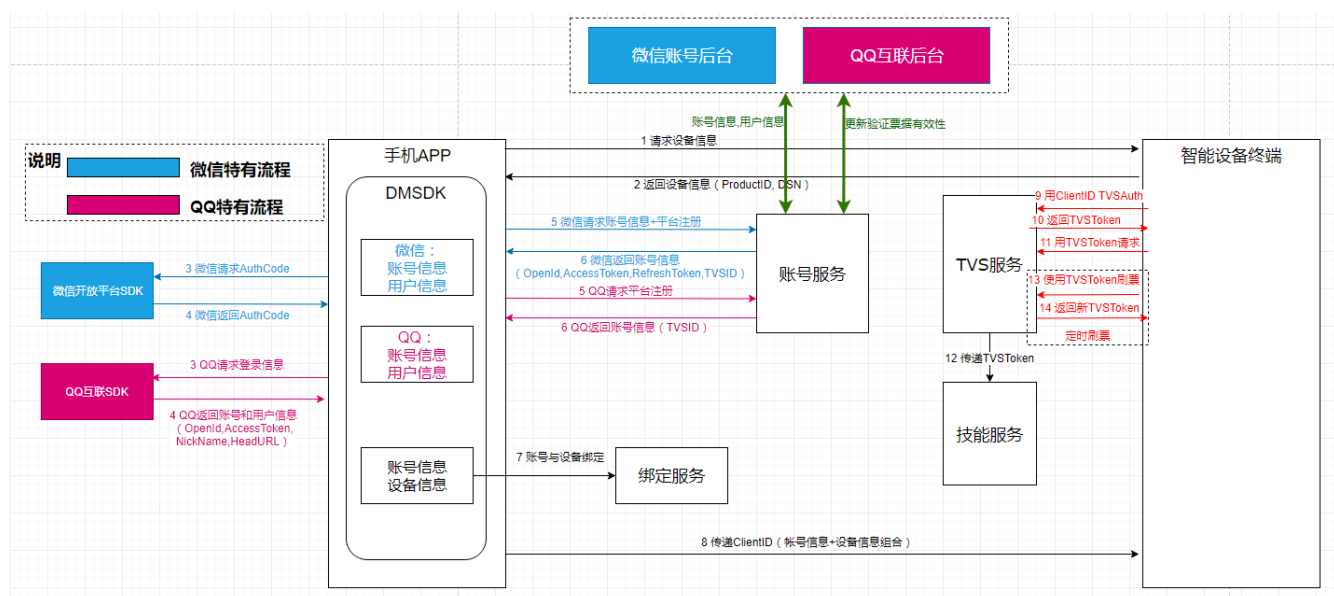


8.4.3 第三步：使用微信或手Q扫码登录

选择登录方式，然后使用微信或手Q的扫一扫功能，扫码登录，完成音乐账号授权。

9 有屏方案（TVSAPI）

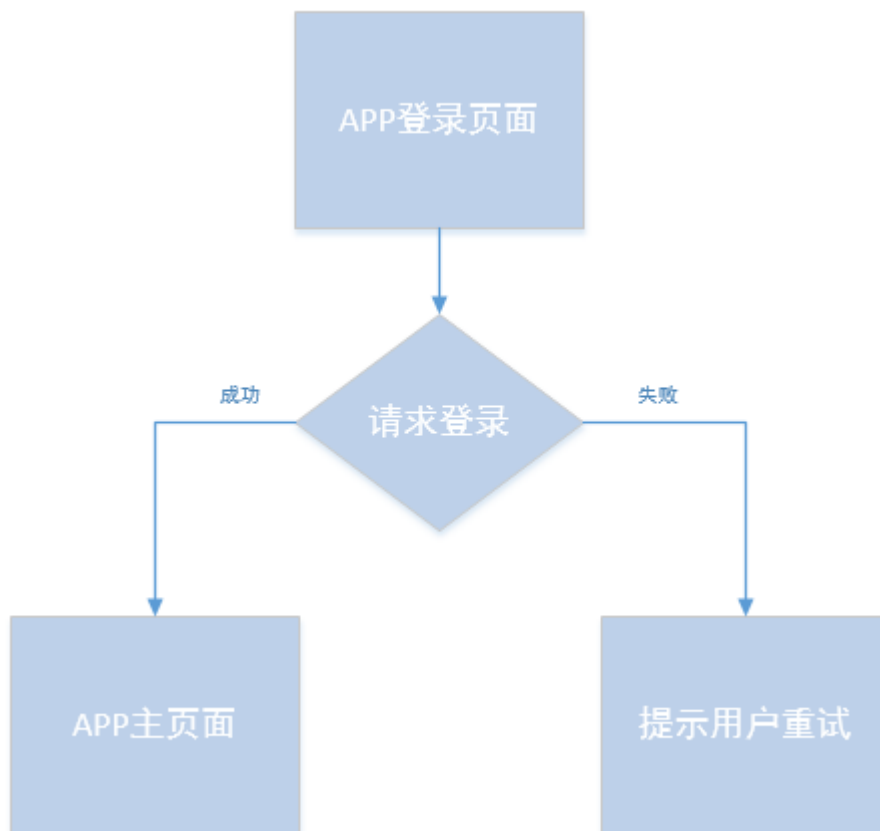
9.1 ER流程图



9.2 手机端

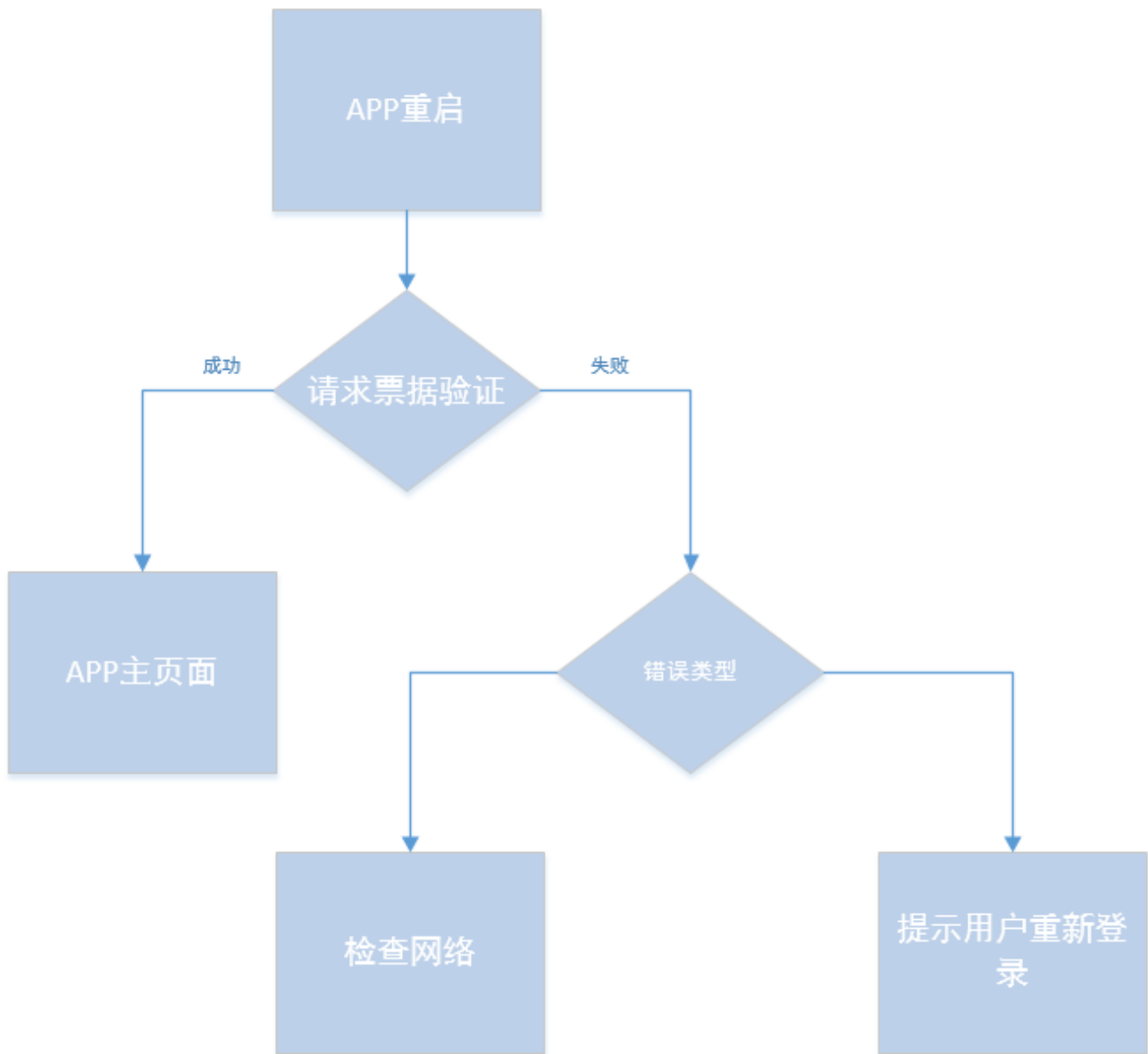
9.2.1 第一步：账号登录

手机APP在初始化状态下，应该是无登录态的，DMSDK中提供了针对微信和QQ的授权接口tvLogin可以通过拉起微信或手Q的授权页面来给厂商APP提供授权能力，根据回调成功或失败接口来进行下一步操作，例如成功后带上登录信息和用户信息进入下一个页面，失败后就通过强提示功能提示用户重新登录。



9.2.2 第二步：账号验证

手机APP在有登录态的场景下，进入后，必须调用requestTokenVerify进行票据验证。成功后带上最新的登录信息和用户信息进行正常操作，失败后通过强提示功能：清除本地的账号信息缓存并跳转到登录页面要求用户重新登录

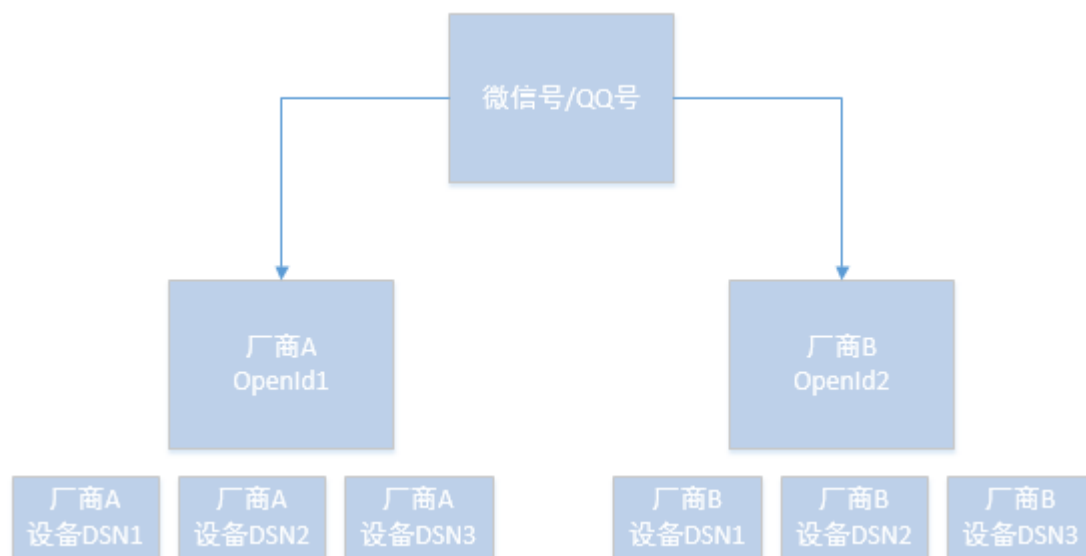


9.2.3 第三步：账号设备绑定

需要完成账号信息与设备信息的绑定，对于支付、IoT、闹钟提醒、推送等场景需要校验云端绑定态。

对于绑定，解绑时填写的字段有一定的要求，具体可详见DMSDK API文档。

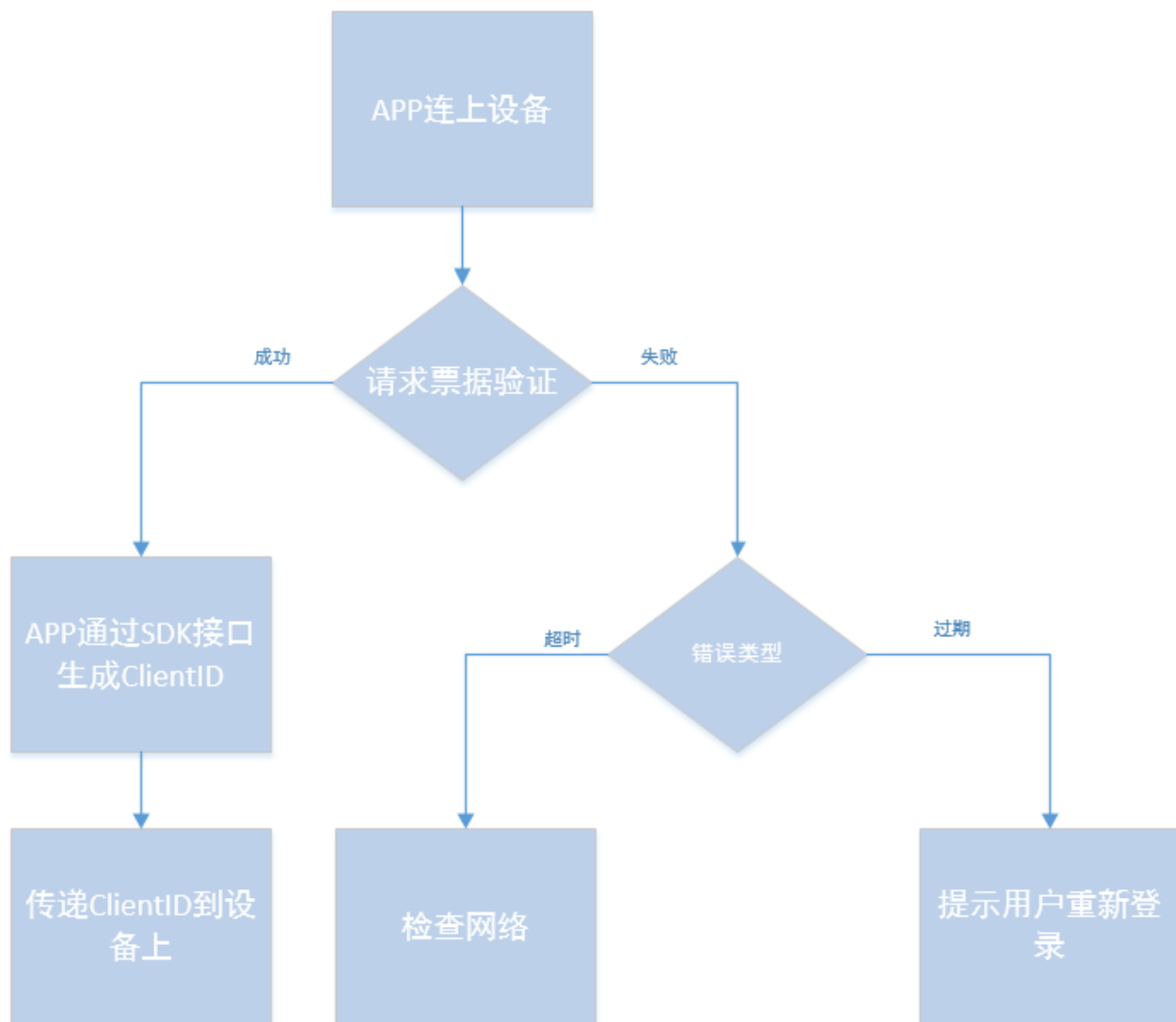
如果是非微信或QQ授权类型的，请按照DMSDK API文档，传入第三方账号字段进行绑定、解绑。



9.2.4 第四步：授权信息生成

授权信息ClientID是手机到设备的需要传递的唯一账号数据信息，并且只有刷票成功才可以去生成ClientID。

如果是非微信或QQ授权类型的，请按照DMSDK API文档，传入第三方账号字段进行ClientID的获取。



9.2.5 第五步：授权信息ClientID传递

使用蓝牙、局域网方式，或通用设备通信协议完成，该部分为厂商自行实现

9.3 设备端

```
// TVSAPI接入方式正式环境授权/验票API  
https://tvs.htm15.qq.com/auth/o2/token  
// TVSAPI接入方式体验环境授权/验票API  
https://tvsexp.htm15.qq.com/auth/o2/token  
// TVSAPI接入方式测试环境授权/验票API  
https://tvstest.htm15.qq.com/auth/o2/token
```

9.3.1 第一步：授权

当采用第三方账号授权方式且无手机APP参与流程的时候，ClientID生成需要按照以下格式完成："ThirdParty","",厂商id,厂商token,"", "",ProductID,dsn。

当采用访客账号，即无手机APP参与流程时，请使用以下编码格式完成ClientID生成。

```
//终端的ProductId
string strProductId;
//终端的Dsn
string strDsn;
//拼接文本
string strPlain = strProductId + strDsn;
/*生成终端ClientId
"ENCRYPT:0001," , "0001" , "MD5"是魔数,不能改变 (包括大小写)
md5表示采用MD5算法,生成32字节的字符串
upper表示将字符串转换成大写
*/
string strClientId = "ENCRYPT:0001," + upper(md5(upper(md5(strPlain + "0001"))) + "MD5")) +
"," + strProductId + "," + strDsn;
```

9.3.1.1 接口描述

设备上拿到手机传过来的ClientId,去TVS后台进行鉴权操作,换取真正用于服务合法访问的TVSToken,对于无屏设备来说TVS鉴权失败的话,就要进行重试操作,或是以语音方式提示用户检查网络,对于有屏设备来说失败的话,也是需要重试操作,或是以界面UI的形式拉起网络设置页面,让用户检查当前设备连接的网络是否正常。

URL : POST <https://tvs.html5.qq.com/auth/o2/token>

9.3.1.2 请求数据

```
{
  "grant_type": "authorization_code",
  "code": "{{STRING}}",
  "redirect_uri": "{{STRING}}",
  "client_id": "{{STRING}}",
  "code_verifier": "{{STRING}}"
}
```

Parameter	Type	必选	描述
grant_type	string	Yes	请求类型
code	string	Yes	获取票据授权Code
redirect_uri	string	Yes	重定向URL
client_id	string	Yes	设备ID
code_verifier	string	Yes	获取票据校验码

9.3.1.3 返回参数

响应状态码说明

(1) HTTP 200且数据完整

终端授权成功,接下来终端发起建立下行通道请求

(2) 其他情况(包括无响应)

终端提示用户重新通过APP登录，并重新发起授权请求

响应数据

```
{
  "access_token": "{{STRING}}",
  "refresh_token": "{{STRING}}",
  "token_type": "bearer",
  "expires_in": LONG
}
```

Parameter	Type	必选	描述
access_token	string	Yes	访问票据
refresh_token	string	Yes	刷票票据
token_type	string	Yes	访问票据类型
expires_in	long	Yes	访问票据有效时间

9.3.2 第二步：定时刷票

9.3.2.1 接口描述

使用返回的过期时间，在终端启动定时器，定时执行刷票操作，防止使用过期的票据请求失效。

URL：POST https://tvs.html5.qq.com/auth/o2/token

9.3.2.2 请求参数

```
{
  "grant_type": "refresh_token",
  "client_id": "{{STRING}}",
  "refresh_token": "{{STRING}}"
}
```

Parameter	Type	必选	描述
grant_type	string	Yes	请求类型
client_id	string	Yes	设备ID
refresh_token	string	Yes	刷票票据

9.3.2.3 返回参数

响应状态码说明

(1) HTTP 200且数据完整

终端刷票成功，接下来终端发起建立下行通道请求

(2) 其他情况(包括无响应)

终端重试(次数由终端控制)，若重试后仍然不成功，则提示用户重新通过APP登录，并重新发起授权请求

响应数据

```
{
  "access_token": "{{STRING}}",
  "refresh_token": "{{STRING}}",
  "token_type": "bearer",
  "expires_in": LONG
}
```

Parameter	Type	必选	描述
access_token	string	Yes	访问票据
refresh_token	string	Yes	刷票票据
token_type	string	Yes	访问票据类型
expires_in	long	Yes	访问票据有效时间

9.3.3 第三步：启动刷票

在设备端有票据登录态的场景下，执行刷票操作，防止使用过期的票据请求失效。

URL：POST https://tvs.htm15.qq.com/auth/o2/token

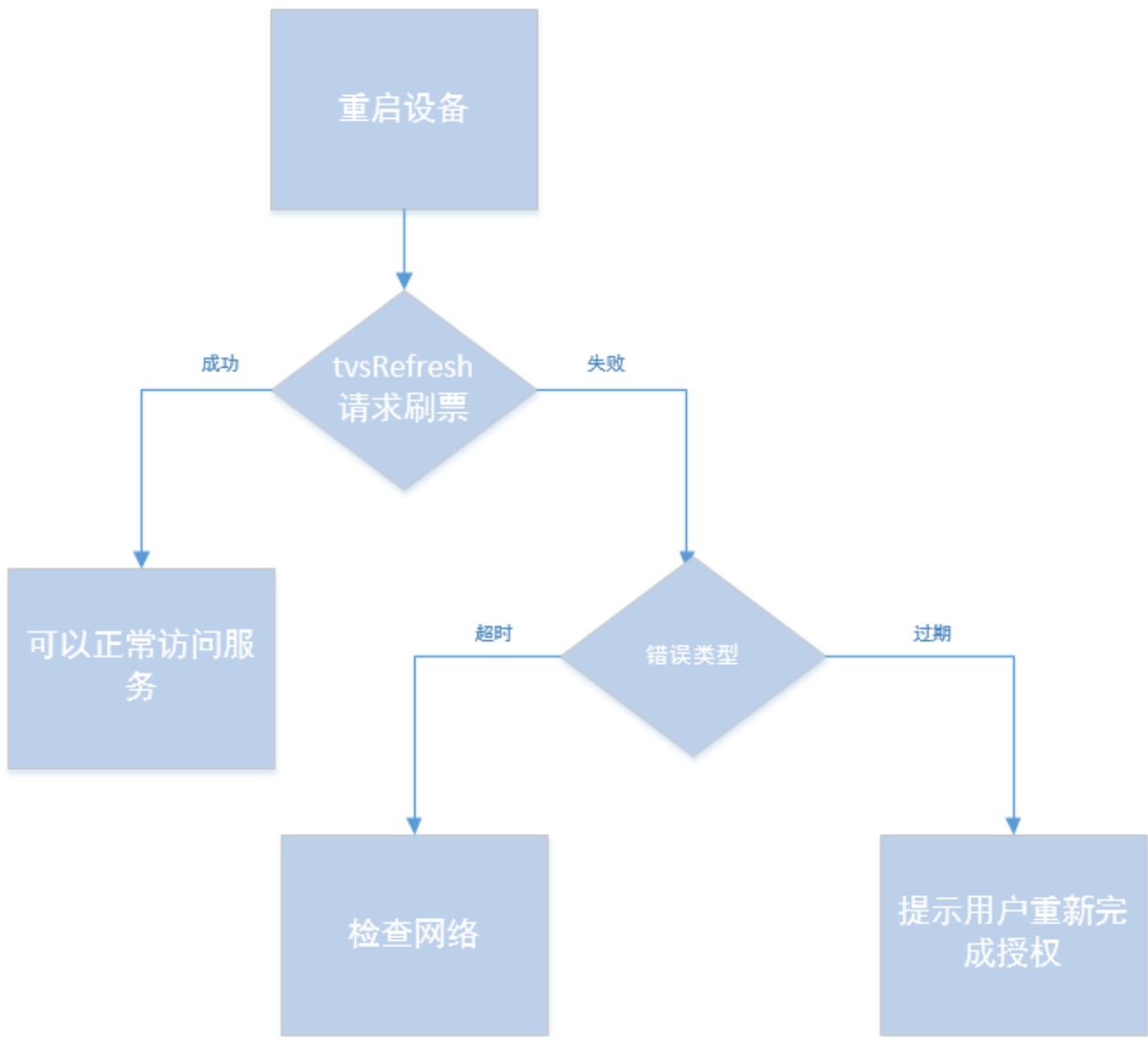
接口描述与参数同定时刷票

9.3.4 第四步：网络切换刷票

在设备端开机状态下网络状态从无到有的场景下，执行刷票操作，防止使用过期的票据请求失效。

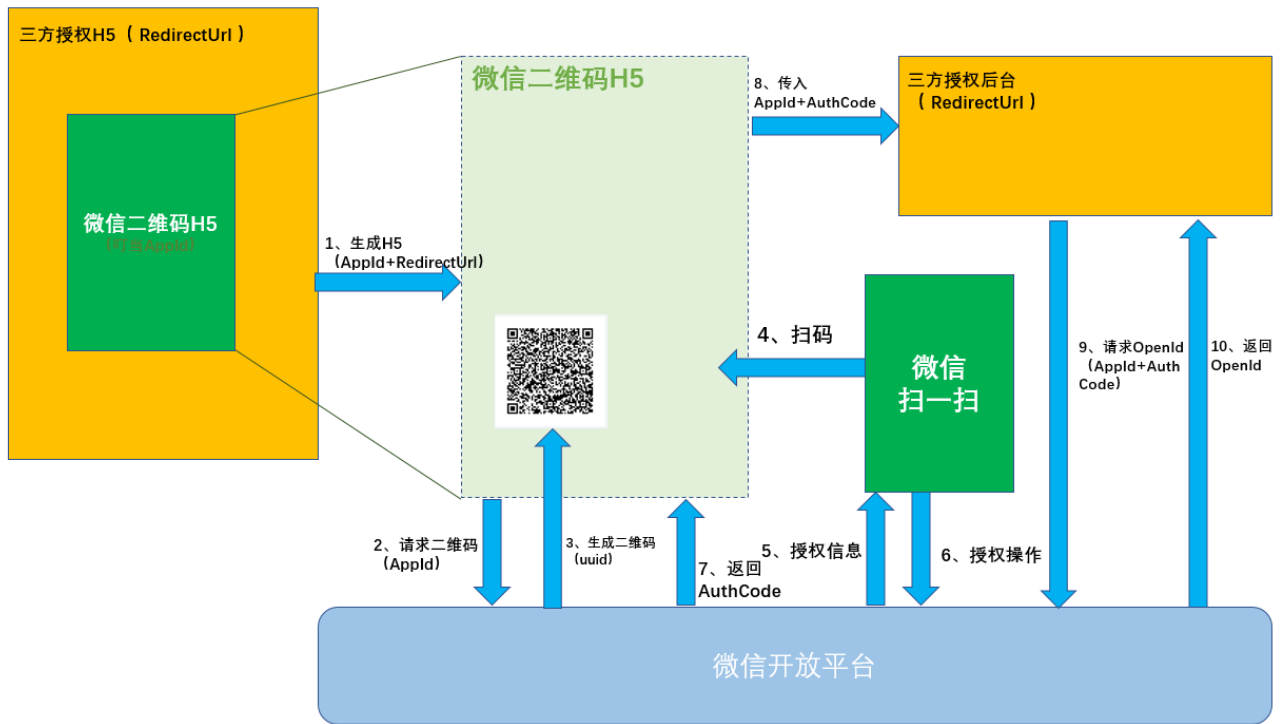
URL：POST https://tvs.htm15.qq.com/auth/o2/token

接口描述与参数同定时刷票



9.4 音乐授权

其原理就是在设备上弹出音乐二维码，通过手Q或微信扫码方式来进行授权，以微信为例其流程图如下。



注意：由于屏幕尺寸的差异，目前仅支持标准的屏幕分辨率的有屏方案，如非标准屏幕需要使用QQ音乐授权功能，请采用更加通用的云小微无屏授权方案或联系云小微客服。

9.4.1 第一步：TVSToken获取

通过设备端授权接口，使用ClientID来换取真正用来访问服务的TVSToken

9.4.2 第二步：接入音乐登录二维码页面

9.4.2.1 Android智能设备

接入tvswab sdk，将上一步获取到的TVSToken作为参数传入loadURL方法中，并打开音乐URL：<https://ddsdk.html5.qq.com/v2/opt/music/login>

9.4.2.2 Linux智能设备

集成浏览器，打开音乐URL：<https://ddsdk.html5.qq.com/v2/opt/music/login>，并将上一步获取到的TVSToken设置到cookie中。



请选择QQ音乐的登录方式

微信登录

QQ登录

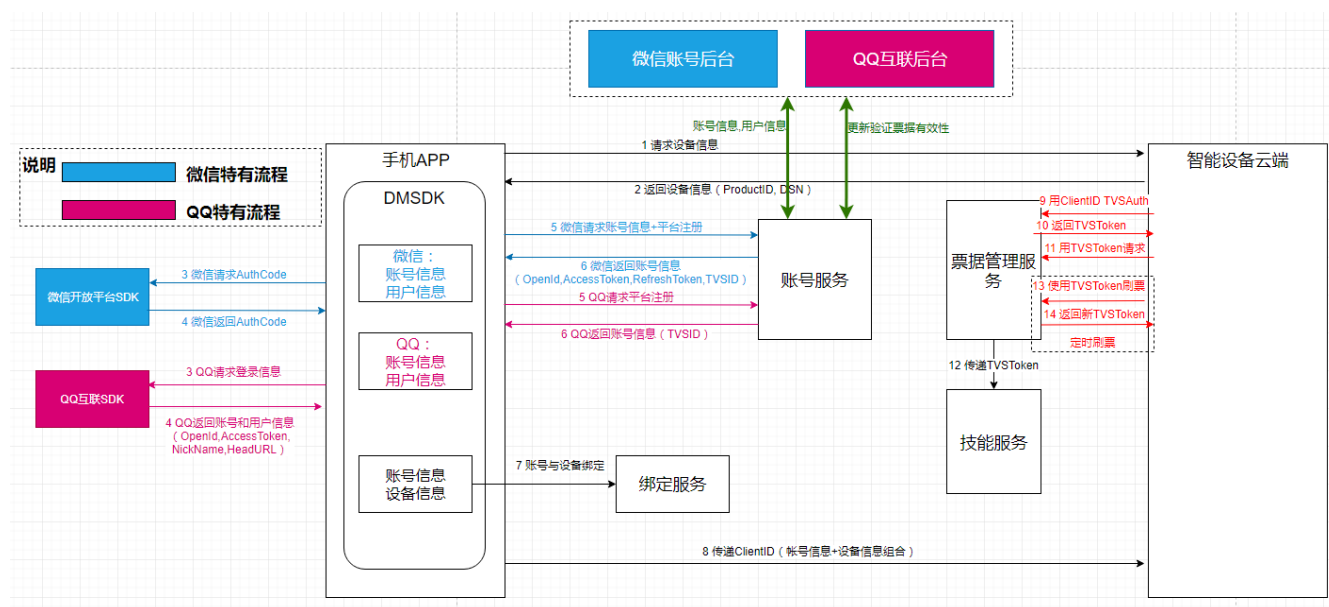
为了保护您的账号，微信登录有效期为30天，QQ登录有效期为90天

9.4.3 第三步：使用微信或手Q扫码登录

选择登录方式，然后使用微信或手Q的扫一扫功能，扫码登录，完成音乐账号授权。

10 有屏方案（基础API）

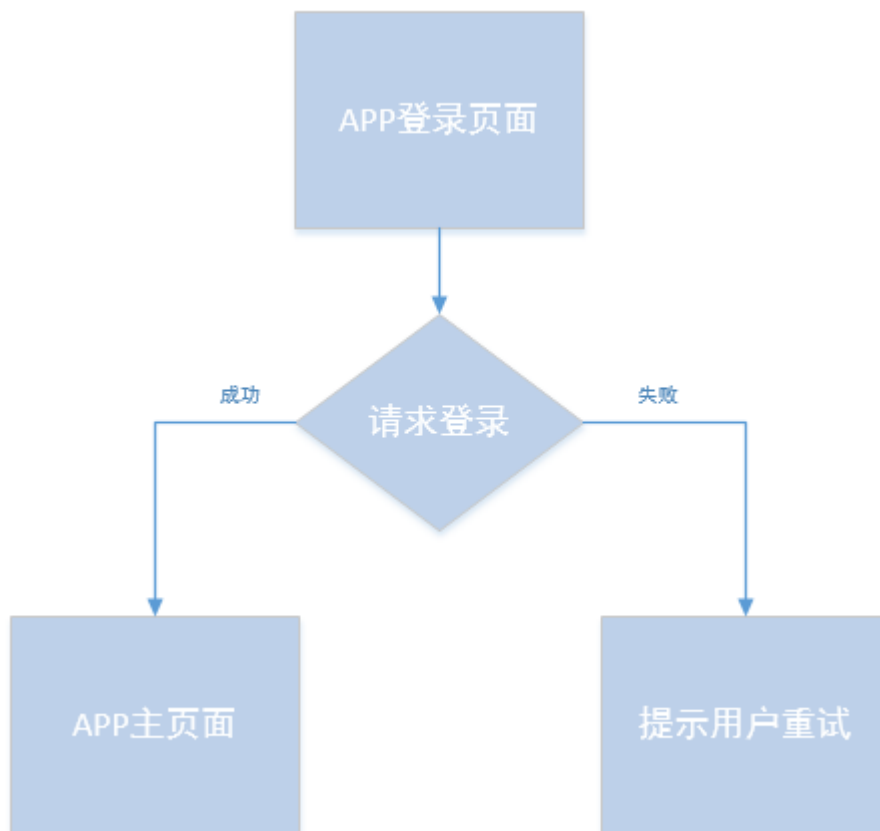
10.1 ER流程图



10.2 手机端

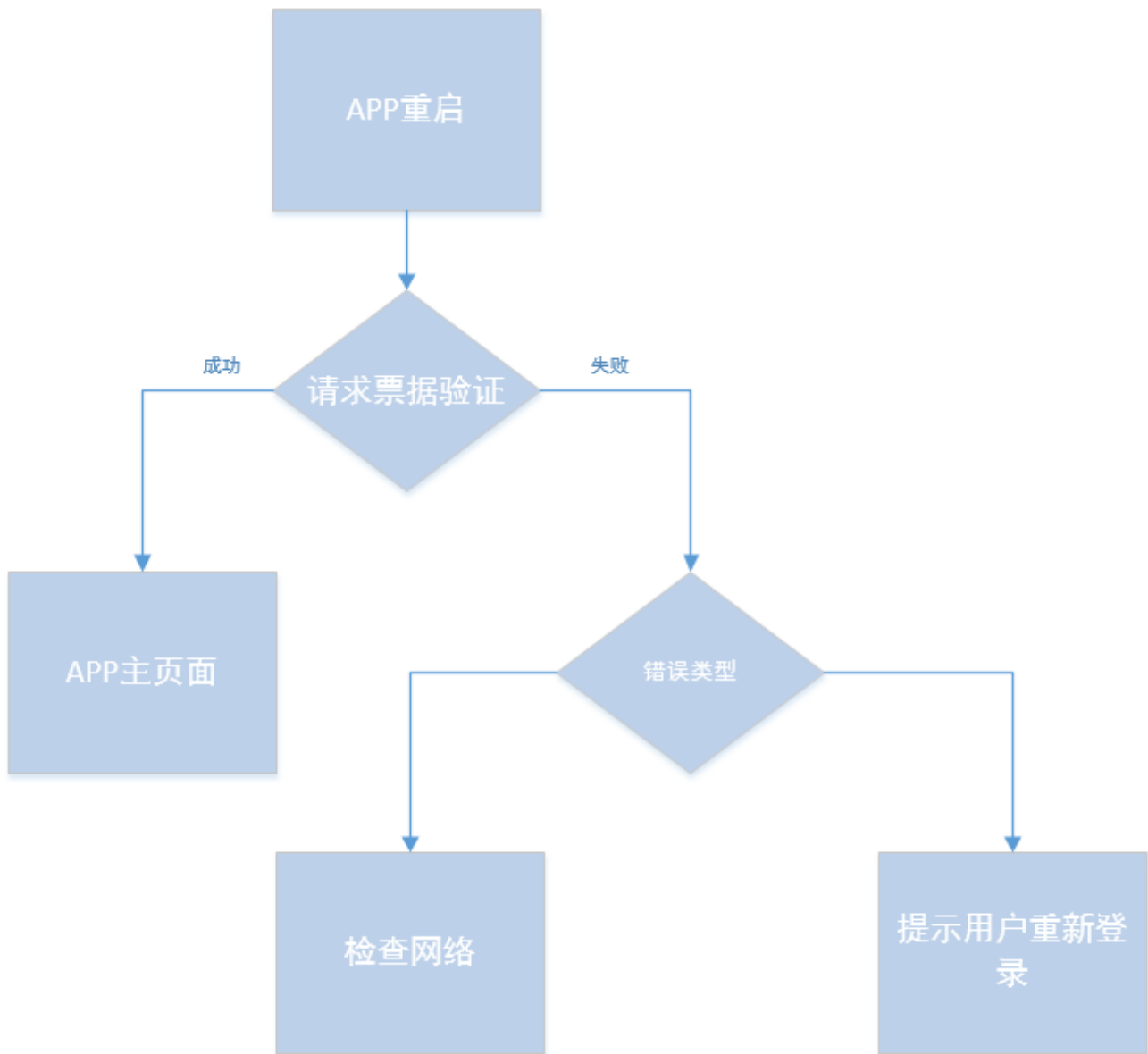
10.2.1 第一步：账号登录

手机APP在初始化状态下，应该是无登录态的，DMSDK中提供了针对微信和QQ的授权接口tvLogin可以通过拉起微信或手Q的授权页面来给厂商APP提供授权能力，根据回调成功或失败接口来进行下一步操作，例如成功后带上登录信息和用户信息进入下一个页面，失败后就通过强提示功能提示用户重新登录。



10.2.2 第二步：账号验证

手机APP在有登录态的场景下，进入后，必须调用requestTokenVerify进行票据验证。成功后带上最新的登录信息和用户信息进行正常操作，失败后通过强提示功能：清除本地的账号信息缓存并跳转到登录页面要求用户重新登录

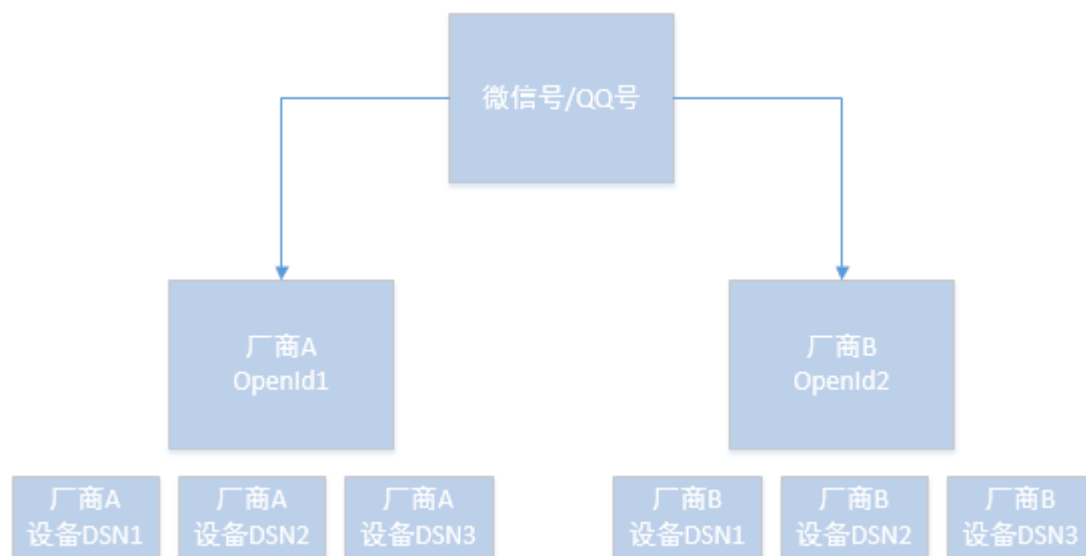


10.2.3 第三步：账号设备绑定

需要完成账号信息与设备信息的绑定，对于支付、IoT、闹钟提醒、推送等场景需要校验云端绑定态。

对于绑定，解绑时填写的字段有一定的要求，具体可详见DMSDK API文档。

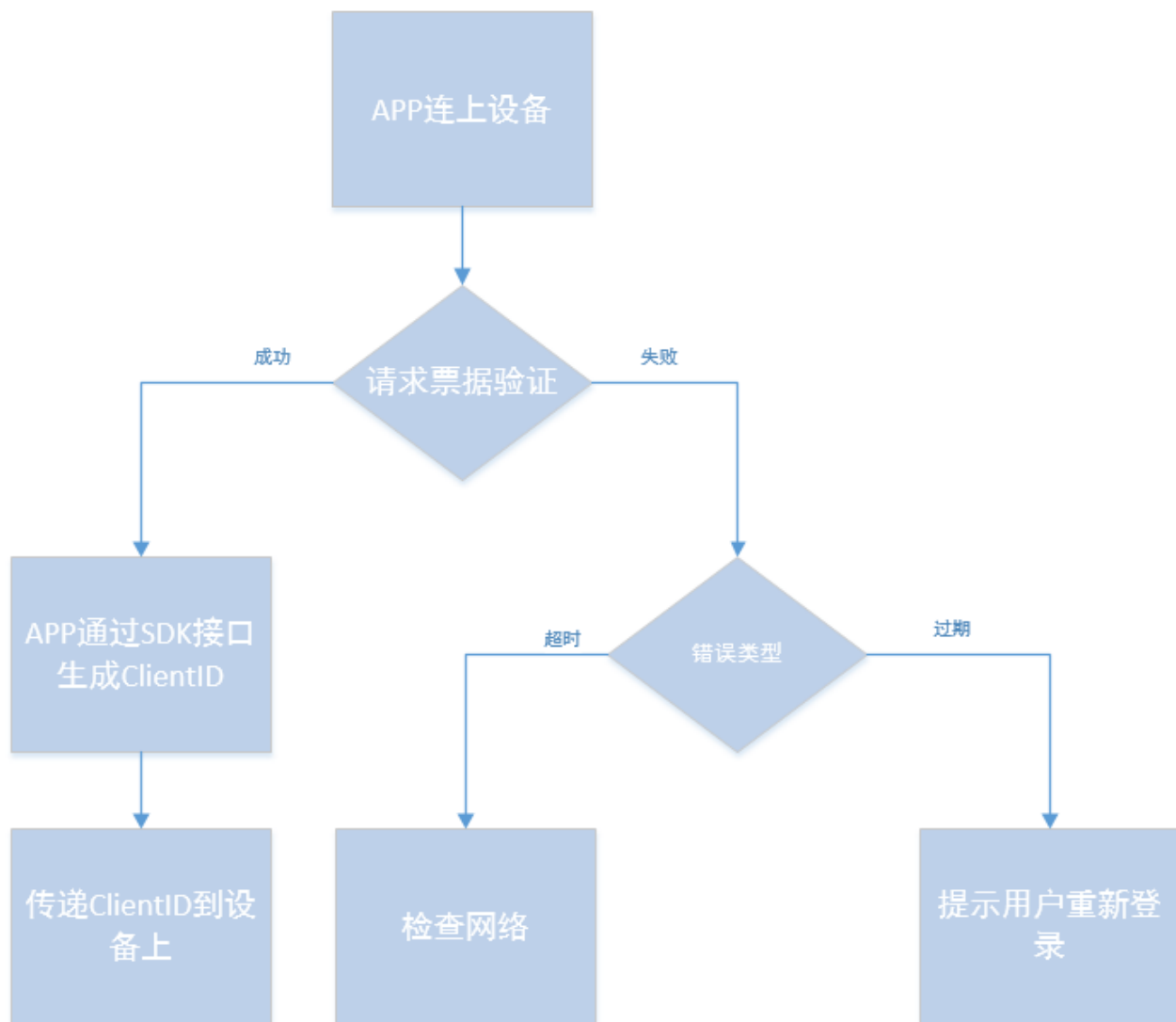
如果是非微信或QQ授权类型的，请按照DMSDK API文档，传入第三方账号字段进行绑定、解绑。



10.2.4 第四步：授权信息生成

授权信息ClientID是手机到设备的需要传递的唯一账号数据信息，并且只有刷票成功才可以去生成ClientID。

如果是非微信或QQ授权类型的，请按照DMSDK API文档，传入第三方账号字段进行ClientID的获取。



10.2.5 第五步：授权信息ClientID传递

使用蓝牙、局域网方式，或通用设备通信协议完成，该部分为厂商自行实现

10.3 厂商云端

```
// 基础API方式正式环境授权API
https://aiwx.html5.qq.com/api/v1/account/authorize
// 基础API方式正式环境验票API
https://aiwx.html5.qq.com/api/v1/account/refresh

// 基础API方式体验环境授权API
https://aiwx.html5.qq.com/exapi/v1/account/authorize
// 基础API方式体验环境验票API
https://aiwx.html5.qq.com/exapi/v1/account/refresh

// 基础API方式测试环境授权API
https://aiwx.html5.qq.com/testapi/v1/account/authorize
// 基础API方式测试环境验票API
https://aiwx.html5.qq.com/testapi/v1/account/refresh
```

10.3.1 第一步：授权

10.3.1.1 接口描述

当采用第三方账号授权方式且无手机APP参与流程的时候，ClientID生成需要按照以下格式完成："ThirdParty","",厂商id,厂商token","",ProductID,dsn。

当采用访客账号，即无手机APP参与流程时，请使用以下编码格式完成ClientID生成。

```
//终端的ProductId
string strProductId;
//终端的Dsn
string strDsn;
//拼接文本
string strPlain = strProductId + strDsn;
/*生成终端ClientID
"ENCRYPT:0001","0001","MD5"是魔数，不能改变（包括大小写）
md5表示采用MD5算法，生成32字节的字符串
upper表示将字符串转换成大写
*/
string strClientId = "ENCRYPT:0001," + upper(md5(upper(md5(strPlain + "0001"))) + "MD5")) +
"," + strProductId + "," + strDsn;
```

提供终端ClientId票据授权接口。

调用时机：设备侧第一次拿到手机端传过来的ClientID，需要调用本接口换取对应的 tvsRefreshToken 和 authorization，tvsRefreshToken 用于刷票接口，authorization 内部有账号与设备信息，调用其他API请求接口时，把 authorization 填入 header.user.authorization。如果填入 header.user.authorization，那么 header.guid,header.user 其他字段都不需要填写。

10.3.1.2 请求参数

URL：POST https://aiwx.html5.qq.com/api/v1/account/authorize

```
{
  "header": {
    "qua": "{{STRING}}",
    "ip": "8.8.8.8",
    "device": {
      "network": "4G"
    }
  },
  "payload": {
    "clientId": "{{STRING}}"
  }
}
```

参数名	类型	是否必选	描述
header	-	是	请求头
header.qua	string	是	设备及应用信息，详细说明见 附录-QUA字段说明
header.ip	string	否	终端IP，仅限云服务使用。终端直接调用本节课不要填这个字段。
header.device	-	否	
header.device.network	string	否	网络类型：4G / 3G / 2G / Wi-Fi
payload	-	是	请求内容
payload.clientId	-	是	DMSDK ClientId

10.3.1.3 返回参数

```
{
  "header": {
    "retCode": 0,
    "errMsg": "{{STRING}}"
  },
  "payload": {
    "tvsRefreshToken": "{{STRING}}",
    "authorization": "{{STRING}}",
    "expiredTimeInSeconds": {{INT}}
  }
}
```

Header Parameters

参数名	类型	是否必选	描述
header	object	Yes	头部
header.retCode	long	Yes	返回码。如果错误码不等于0且大于-1000000，可以认为票据无效。
header.errMsg	string	Yes	错误消息

Payload Parameters

参数名	类型	是否必选	描述
<code>payload</code>	<code>object</code>	Yes	负载
<code>payload.tvRefreshToken</code>	<code>string</code>	Yes	tvRefreshToken，刷票用
<code>payload.authorization</code>	<code>string</code>	Yes	账户信息。用在其他请求的 <code>header.user.authorization</code> 中。
<code>payload.expiredTimeInSeconds</code>	<code>int</code>	Yes	票据过期时间 (秒)。到达过期时间，需要调用7.8刷票接口重新刷票。

10.3.2 第二步：定时刷票

10.3.2.1 接口描述

提供终端刷票接口。根据授权返回的 `payload.expiredTimeInSeconds` 字段，启动云端定时刷票

10.3.2.2 请求参数

URL：POST `https://aiwx.html5.qq.com/api/v1/account/refresh`

```
{
  "header": {
    "qua": "{{STRING}}",
    "ip": "8.8.8.8",
    "device": {
      "network": "4G"
    }
  },
  "payload": {
    "tvRefreshToken": "{{STRING}}"
  }
}
```

参数名	类型	是否必选	描述
header	-	是	请求头
header.qua	string	是	设备及应用信息，详细说明见 附录-QUA字段说明
header.ip	string	否	终端IP，仅限云服务使用。终端直接调用本接口不要填这个字段。
header.device	-	否	
header.device.network	string	否	网络类型：4G / 3G / 2G / wi-Fi
payload	-	是	请求内容
payload.tvsRefreshToken	-	是	TVS刷票Token

10.3.2.3 返回参数

```
{
  "header": {
    "retCode": 0,
    "errMsg": "{{STRING}}"
  },
  "payload": {
    "tvsRefreshToken": "{{STRING}}",
    "authorization": "{{STRING}}",
    "expiredTimeInSeconds": {{INT}}
  }
}
```

Header Parameters

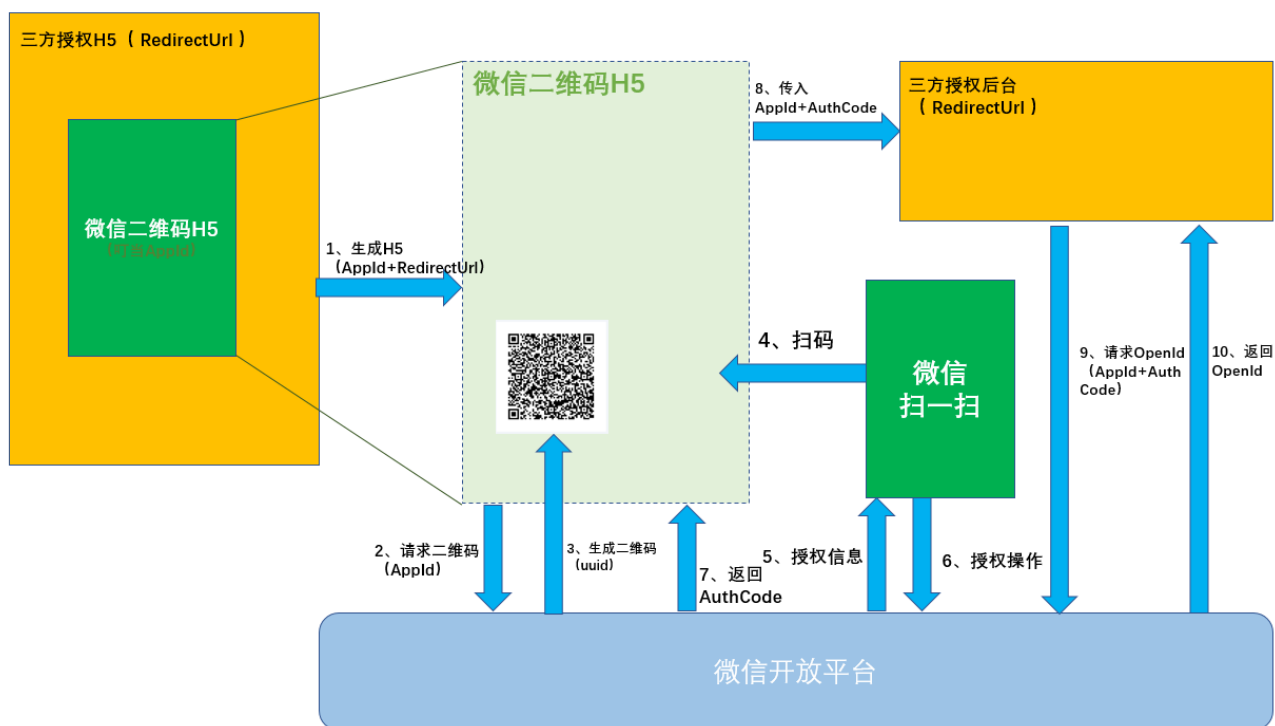
参数名	类型	是否必选	描述
header	object	Yes	头部
header.retCode	long	Yes	返回码。如果错误码不等于0且大于-1000000，可以认为票据无效。
header.errMsg	string	Yes	错误消息

Payload Parameters

参数名	类型	是否必选	描述
<code>payload</code>	<code>object</code>	Yes	负载
<code>payload.tvRefreshToken</code>	<code>string</code>	Yes	最新tvRefreshToken，下次刷票使用。
<code>payload.authorization</code>	<code>string</code>	Yes	最新账户信息。每次请求7.1-7.6的接口，都需要用最新账户信息。
<code>payload.expiredTimeInSeconds</code>	<code>int</code>	Yes	过期时间 (秒)。到达过期时间，需要重新刷票。

10.4 音乐授权

其原理就是在设备上弹出音乐二维码，通过手Q或微信扫码方式来进行授权，以微信为例其流程图如下。



注意：由于屏幕尺寸的差异，目前仅支持标准的屏幕分辨率的有屏方案，如非标准屏幕需要使用QQ音乐授权功能，请采用更加通用的云小微无屏授权方案或联系云小微客服。

10.4.1 第一步：TVSToken获取

通过设备端授权接口，使用ClientID来换取真正用来访问服务的TVSToken

10.4.2 第二步：接入音乐登录二维码页面

10.4.2.1 Android智能设备

接入tvswed sdk，将上一步获取到的TVSToken作为参数传入loadURL方法中，并打开音乐URL：<https://ddsdk.html5.qq.com/v2/opt/music/login>

10.4.2.2 Linux智能设备

集成浏览器，打开音乐URL：<https://ddsdk.html5.qq.com/v2/opt/music/login>，并将上一步获取到的TVSToken设置到cookie中。



10.4.3 第三步：使用微信或手Q扫码登录

选择登录方式，然后使用微信或手Q的扫一扫功能，扫码登录，完成音乐账号授权。