# 1、思路

1、安卓设备安装app，然后app运行都需要发起一次请求Authenticate和TokenUpdate，

在ios中，唤醒设备需要token和pushmagic参数，不知道拿这两个参数做了什么然后得出是要把消息推送给哪台设备，设备收到消息就根据描述文件的ServerURL发起请求获取命令。所以，如果我们要实现由apns（android push notification service）主动推送消息给app，那应该是要保留TokenUpdate的方法的，如果是由app轮询服务器的话，那token和pushmagic似乎没有用处，应该就不执行这个方法了。

Authenticate的通讯仿照ios：

*/\*  
 PUT /your/url HTTP/1.1  
 Host: www.yourhostname.com  
 Content-Length: 1234  
 Content-Type: application/x-apple-aspen-mdm-checkin //****自己制定的协议，这个没有必要了*** *\* \*/*

**<?xml version="1.0" encoding="UTF-8"?>**  
 **<!DOCTYPE plist PUBLIC "-//Apple//DTD PLIST 1.0//EN" "http://www.apple.com/DTDs/PropertyList-1.0.dtd">**  
 **<plist version="1.0">**  
 **<dict>**  
 **<key>MessageType</key>**  
 **<string>Authenticate</string>**  
 **<key>Topic</key>**  
 **<string>com.apple.mgmt.External.75446d48-bba4-484e-8f34-eae167fa61c4</string>**  
 **<key>UDID</key>**  
 **<string>** android\_id **</string>**  
 **<key>OSVersion</key><string>OSVersion</string>** *//****ios可以额外提交这些数据，不知道android可以提交什么，反正就是往这里补充，因为这些是要保存到数据库的，最好能提供*****<key>BuildVersion</key><string>BuildVersion</string>**  
 **<key>ProductName</key><string>ProductName</string>**  
 **<key>SerialNumber</key><string>SerialNumber</string>**  
 **<key>IMEI</key><string>IMEI</string>**  
 **<key>MEID</key><string>MEID</string>**  
 **</dict>**  
 **</plist>**

Topic是设备监听的主题，如果不是使用推送的方式，Topic好像也没有用处。还有我们完全可以使用json的方式来处理，由于之前ios的处理xml已经有类库解析，所以使用xml也可以。

TokenUpdate的通讯仿照ios：

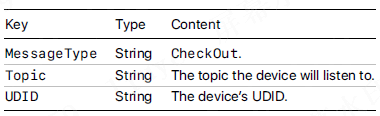
**<?xml version="1.0" encoding="UTF-8"?>**  
 **<!DOCTYPE plist PUBLIC "-//Apple//DTD PLIST 1.0//EN" "http://www.apple.com/DTDs/PropertyList-1.0.dtd">**  
 **<plist version="1.0">**  
 **<dict>**  
 **<key>MessageType</key>**  
 **<string>TokenUpdate</string>**  
 **<key>Topic</key>**  
 **<string>com.apple.mgmt.External.75446d48-bba4-484e-8f34-eae167fa61c4</string>**  
 **<key>UDID</key>**  
 **<string>** android\_id **</string>**  
 **<key>Token</key>**

**<string>Token</string>**  
 **<key>PushMagic</key>**

**<string>PushMagic</string>** *//****这个值由设备生成*****<key>UnlockToken</key>**

**<string>UnlockToken</string>**  
*// <key>AwaitingConfiguration</key><true>* **</dict>**  
 **</plist>**

CheckOut不举例了，参数如下，Topic看是否需要用到

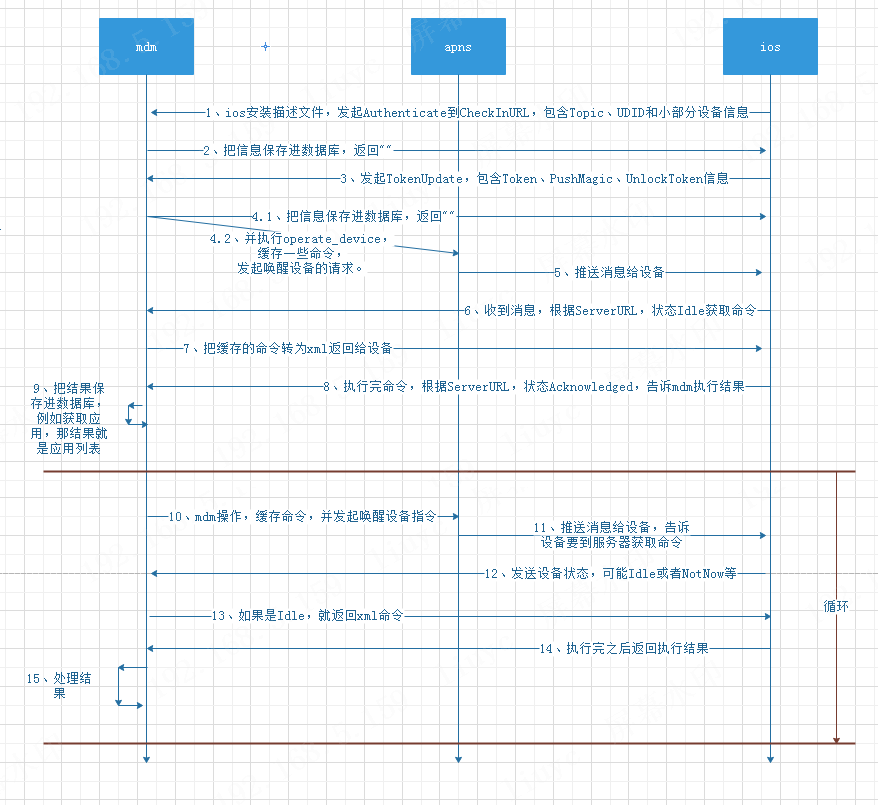


2、根据目前ios mdm的代码，执行完Authenticate和TokenUpdate，就已经把一部分的信息记录到数据库了，如序列号、token、pushmagic、系统版本、IMEI等等，然后mdm在执行TokenUpdate处理结果的时候，还会缓存一些命令（数组的方式），如获取设备信息、获取设备的非系统应用列表、获取设备的托管应用列表，然后唤醒设备，apns收到通知，就推送消息给设备，设备就根据描述文件的ServerURL进行请求，如果设备是Idle状态，就收到mdm缓存的命令（数组转xml）并执行，执行完返回执行结果到ServerURL，所以，设备信息表和设备应用表都会保存好了对应设备的一些信息。

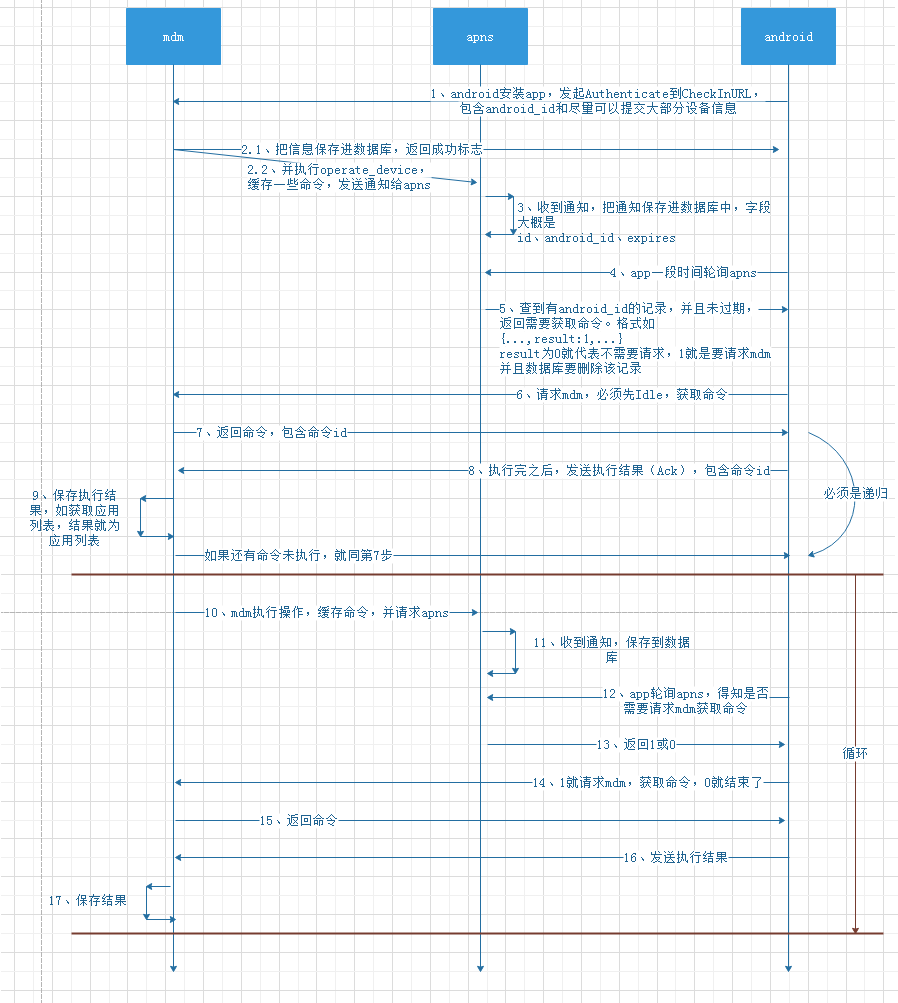
android mdm按上述模仿ios mdm，设备app发起Authenticate到mdm中，执行完之后，设备信息表保存了一部分信息，app根据执行结果继续发起TokenUpdate请求，保存token和pushmagic等信息（目前不知道怎么使用TokenUpdate，也可以不发送吧），并且在执行TokenUpdate过程中，缓存一些命令，如获取设备详细信息。然后就唤醒设备，唤醒方式是mdm发送请求到android push notification service（参数正常情况应该是token和pushmagic，我们这里参数改为android\_id），apns收到请求之后，同时也知道是哪台设备，要么使用推送的方式，要么使用轮询的方式，总之就是为了让设备知道准备请求mdm获取命令。假如采用轮询的方式，那么apns可以把信息记录在数据库中，可能包含id、android\_id、expires字段，app轮询apns查到数据库有自己的android\_id记录，并且没过期，就请求mdm获取命令了。然后假如手机不在网络，expires过期了，就不获取命令了，相当于唤醒失败的情况。

# 2、详细设计

IOS示意图：



Android示意图：（去掉了TokenUpdate，因为不知道怎么使用）



针对android的示意图，详细设计每一步的通讯

（1）假如仍然是使用xml的格式，那么android发起Authenticate的时候，大致格式为（也可以使用json格式）：

**<?xml version="1.0" encoding="UTF-8"?>**  
 **<!DOCTYPE plist PUBLIC "-//Apple//DTD PLIST 1.0//EN" "http://www.apple.com/DTDs/PropertyList-1.0.dtd">**  
 **<plist version="1.0">**  
 **<dict>**  
 **<key>MessageType</key>**  
 **<string>Authenticate</string>**  
 **<key>UDID</key>**  
 **<string>** android\_id **</string>**  
 **<key>OSVersion</key><string>OSVersion</string>** *//****ios可以额外提交这些数据，不知道android可以提交什么，反正就是往这里补充，因为这些是要保存到数据库的，最好能提供*****<key>BuildVersion</key><string>BuildVersion</string>**  
 **<key>ProductName</key><string>ProductName</string>**  
 **<key>SerialNumber</key><string>SerialNumber</string>**  
 **<key>IMEI</key><string>IMEI</string>**  
 **<key>MEID</key><string>MEID</string>**

**。。。。更多数据**  
 **</dict>**  
 **</plist>**

（2.1）在ios那边成功失败都是返回空字符串，因为设备那边并不关心你的mdm是否发生了错误。针对安卓设备的话，我们可以根据不同的返回值，在app中显示执行结果，使用json格式吧，因为不知道android那边方不方便处理plist格式，plist格式是这样的：

<key>a</key>

<string>aaa</string>

<key>b</key>

<string>bbb</string>

在安卓那边解析出来有可能是{"key":["a","b"],"string":["aaa","bbb"]}就会有问题，就是说不清楚容不容易解析，所有这里暂时定为json格式。

（2.2）缓存命令的操作同原来的ios mdm就可以了，把数组存起来，然后把android\_id发送给apns。

（3）apns保存数据到命令表。apns至少两个表，一个设备表，一个命令表，设备表用于checkin、checkout。

（4）app轮询apns，提供android\_id参数（可使用json或plist格式），因为apns是php代码，已经有解析plist的类库。

（5）查到有android\_id的记录，并且未过期，返回result:1，过期或者没有就返回result:0。暂时定为json格式，因为不知道android是否能解析plist数据。

（6）收到result:1，请求mdm。

（7）从缓存中获取命令（包含CommandUUID），返回命令给设备app。暂定json格式。

（8）执行命令，把执行结果发送到mdm（也包含CommandUUID）。

（9）mdm保存执行结果到数据库，如果此时还有未执行的命令，就继续返回命令。