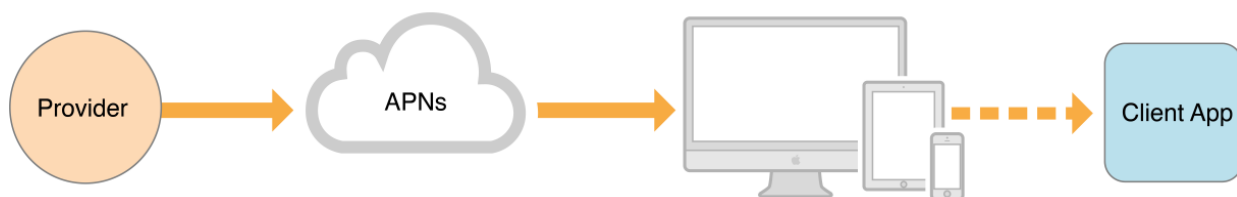


阅读Apple Notification的心得体会

Apple的Notification主要分类本地通知和远端通知(remote Notification)。其中本地通知主要是通过系统层面的控制来展示到前端界面，而Remote Notification需要APP这边推送通知的平台接入苹果统一的推送服务 *Apple Push Notification service* (APNs) 下面是一个简单的示意图



这张图上面的Provider是开发者自己需要部署的消息推送后台，Provider按照约定接入APNs以后，就能够使用部署的后台进行远端消息的推送。

为了推送一条消息，Provider需要经过一个如下的流程：

1. 按照Apple官方给出的[推送格式规则](#)构筑内容，并构建一个包含通知内容的JSON字典；
2. 将通知内容,全局唯一token和其他需要传递的信息放进HTTP请求。
3. 最后将这个包含了令牌证书或者密码凭证的通过一个持久安全的通道发送给APNs。

关于APNs的一点新的体会

这里主要谈谈这次完整了解APNs相关服务后，发现之前没有正确理解的问题。APNs提供的服务包括QoS，存储和转发，合并通知。QoS不用说了，是保证通信安全的一个标准。而这里的存储和转发有一点细节的问题，在Provider已经将Notification发送给APNs时，如果APNs发现目标用户设备处于下线状态，则会将这条通知存储一段时间，在用户处于上线转台的时候立马推送给用户。这里的存储有一个特性，该服务只会存储最新的通知，这就导致了对一个处于下线状态的用户再次推送通知会导致之前的通知(下线状态收到的)会被丢弃

掉。同时，如果超过一定的时间，APNs会把之前保存的通知全部丢弃。APNs还提供了合并多调通知的功能。具体的操作方法是在发送HTTP请求的时候，有一个名为 *apns-collapse-id* 的字段，APNs会自动将该字段取值相同的Notification合并成一条再推送给设备。

APNs的安全措施

为了保证APNs服务的可靠性，APNs强制使用端到端服务，加密验证以及双层认证。双层认证分别为：

- 连接认证
 - Provider-APNs连接认证：

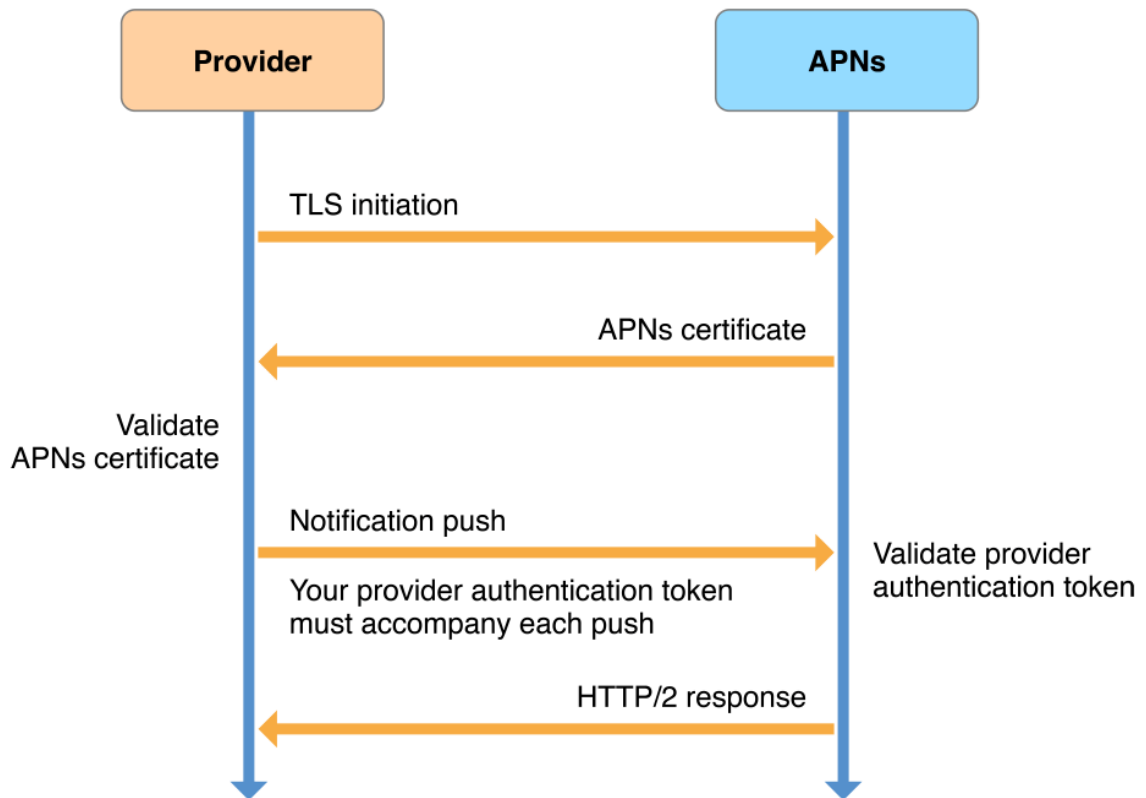
Provider和APNs想建立可靠连接，需要Provider遵循SSL证书或者token授权证书。这两种证书都可以在[Apple开发者官网上](#)进行申请。
 - APNs-设备连接认证：

设备的token信息是一个封闭的NSData数据，只有APNs可以解析其内容。每一个APP在注册到APNs的时候都会获得一个唯一的token用于通信，在之后的每一次推送通知的请求中都需要携带这个token作为标志。
- 设备令牌认证

Provider-APNs可靠性

Provider与APNs之间的通信可靠性可以用两种方案来实现。分别是基于token的连接认证和基于证书的连接认证。

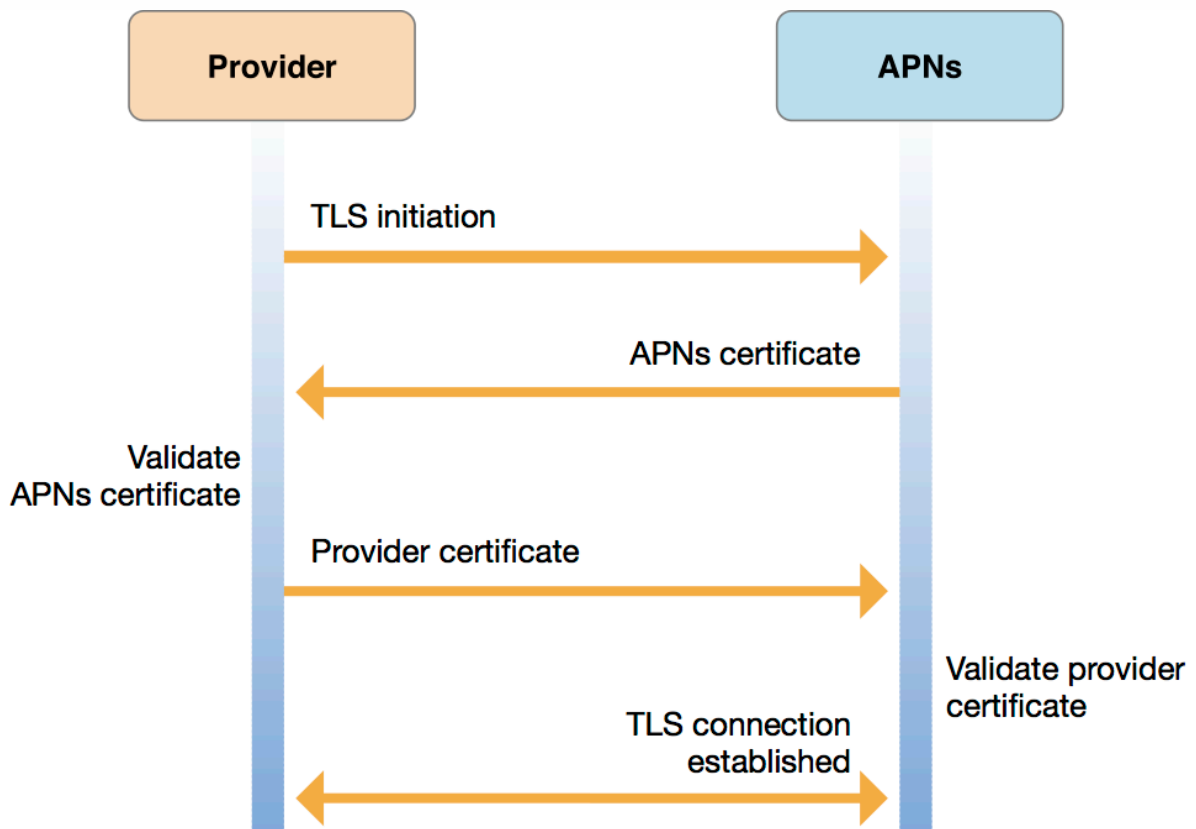
1. 基于token的连接认证



简单解释下上图中的通信流程：

1. Provider通过TLS向APNs请求建立一个安全可靠的链接；
2. APNs返回给Provider一个APNs证书，标识该Provider是有效的。到这一步为止，连接已经建立完成；
3. Provider发送通知推送请求，请求里面需要携带token；
4. APNs验证Provider发送过来的token，回应推送请求。

2. 基于证书的连接认证

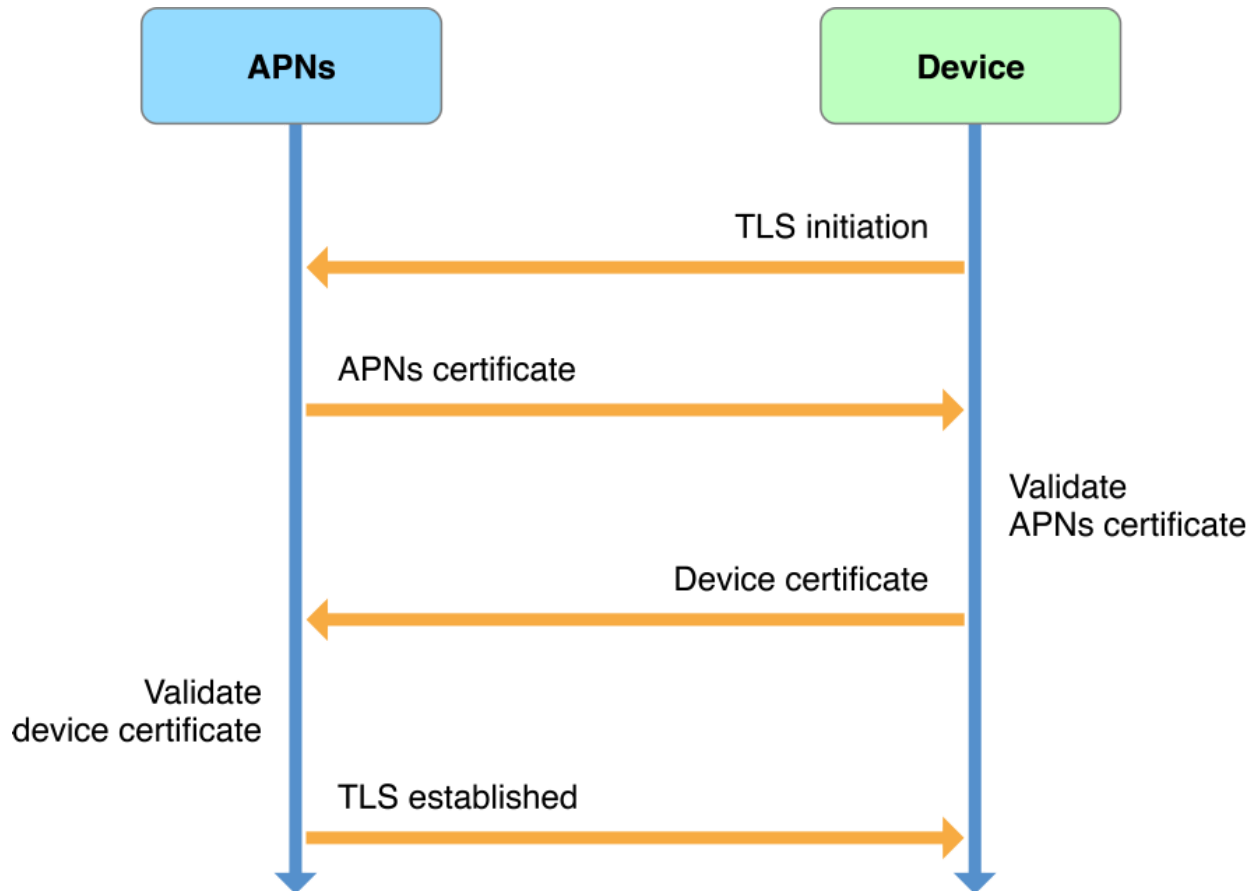


大致的流程：

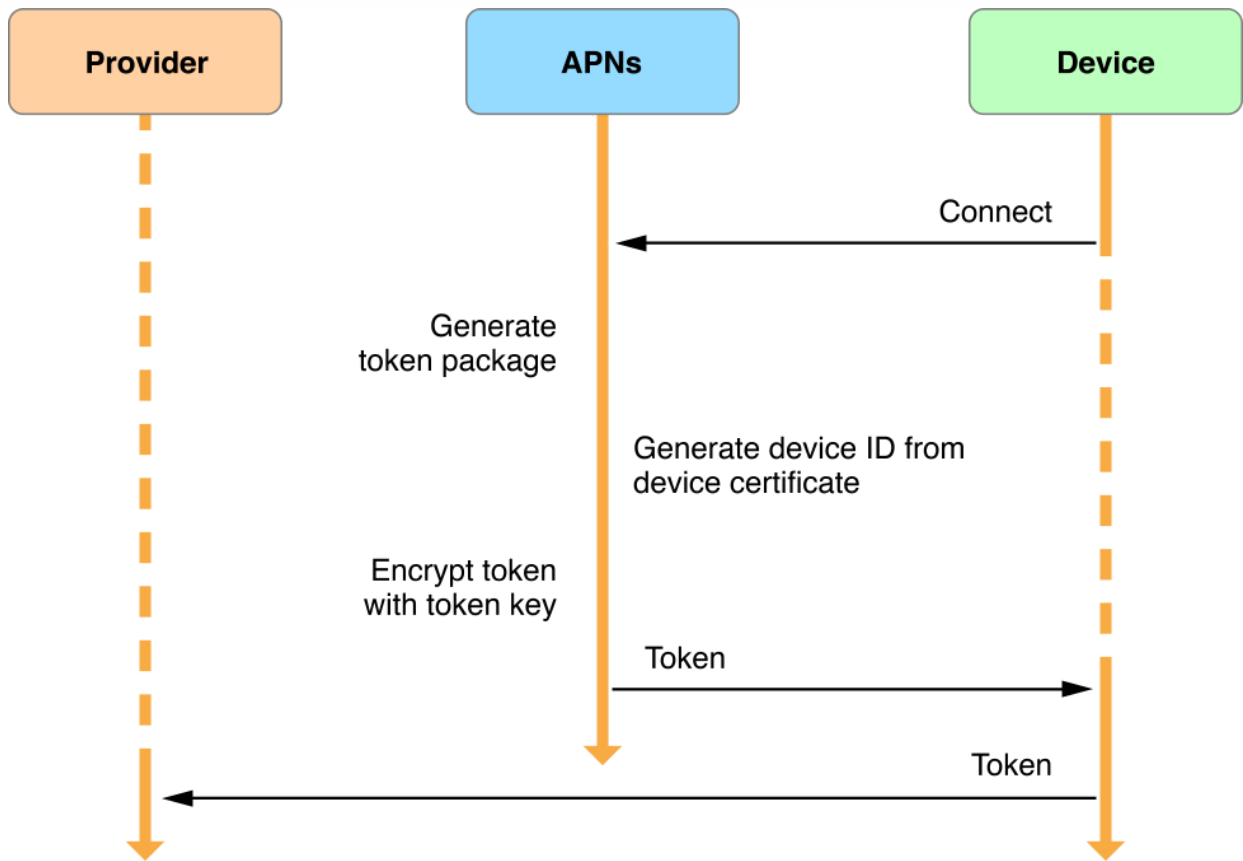
1. Provider通过TLS向APNs请求建立一个安全可靠的链接;
2. APNs返回一个证书，连接建立完成；
3. Provider发送一个*Apple-provisioned provider certificate*(可以通过开发者网站申请)给APNs；
4. APNs对发过来的*Apple-provisioned provider certificate*进行验证，确定跟APNs通信的Provider是可靠有效的，因此建立连接。

APNs-设备连接认证与设备token

需要注意的一点是：APNs和每一个设备的认真连接是自动建立的。每一个设备都一份属于自己的加密证书和一个私钥。通信流程大致如下图所示：



建立的流程比较简单，就不在赘述了。在TLS连接建立完成后。在当前设备上所有的APP都能够通过注册到APNs的方式来获取属于当前APP的特定token用于远端推送通知。在获取到设备token之后，APP需要将获得的Token发送给相关的Provider。Provider(通知后台)需要token来跟APNs和目标设备通信。 token在三端之间的传递流程如下图所示：

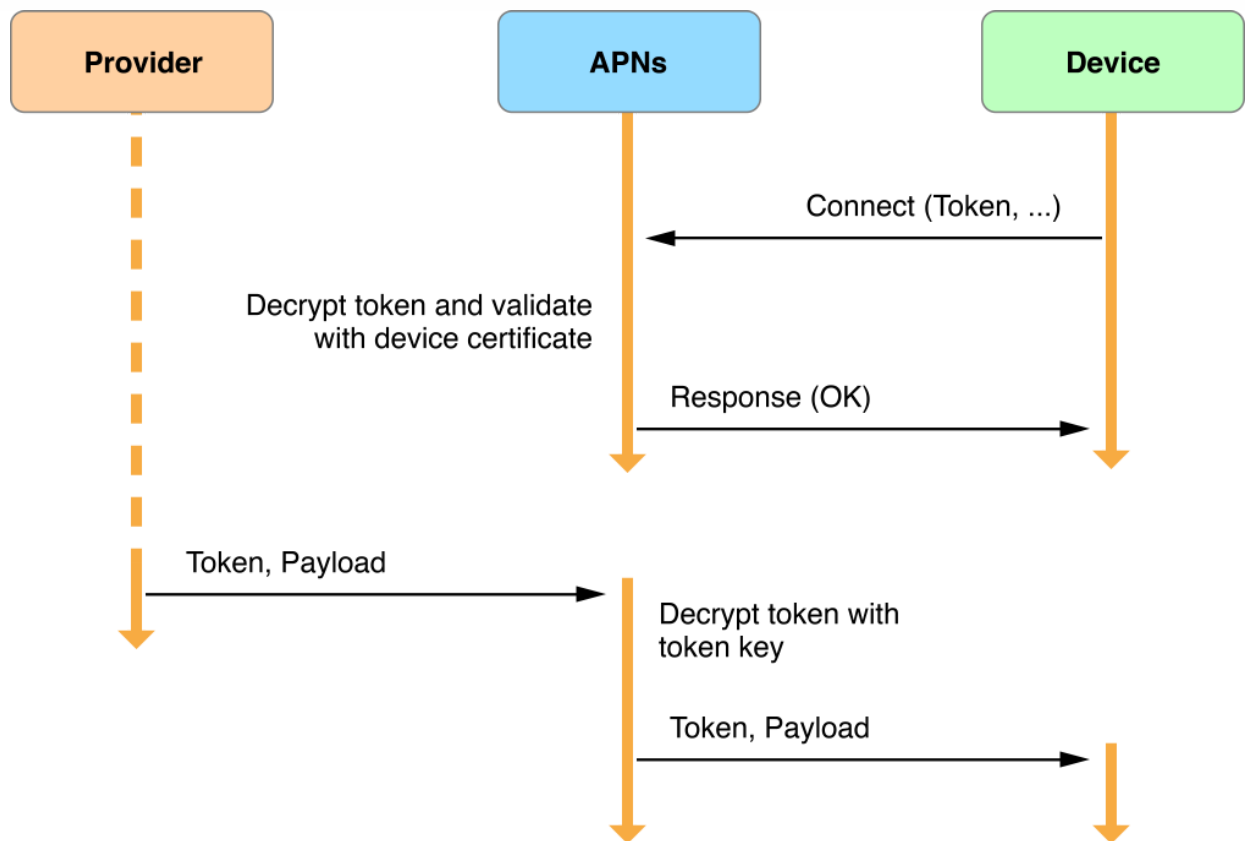


1. 首先将APP注册到APNs上。如果当前的APP已经注册过，并且APP对应的Token没有发生改变的话，系统会快速返回之前的Token并直接跳到第4步。
2. 如果需要新建一个设备Token，APNs通过设备的证书里面包含信息生成一个新的Token。然后将其发送给对应的设备。
3. 通过代码获取到APP对应的Token，调用

```
- (void)application:(NSApplication *)application  
didRegisterForRemoteNotificationsWithDeviceToken:  
(NSData *)deviceToken;
```

4. APP将拿到的Token通过HTTP请求发送给Provider

在整个消息推送流程中Provider到设备的路径



远端通知具体内容

每一个远端通知都有一个保存的内容的负载，其负载大小的容量随通知的类型而定。

系统要求是通常类型的通知内容，大小不能超过4KB；语音形式的内容(VoIP)，大小不能超过5KB。

(如果发送的请求消息体容量超过限制，APNs会拒绝接受这条通知。此外，通知消息本身是不会有保护或者加密措施的，因此关于客户的敏感信息最好不要放在推送通知里面进行传递。)

一个典型的通知内容JSON

```
{
  "aps" : { "alert" : "Message received from Bob" },
  "acme2" : [ "bang", "whiz" ]
}
//aps字段是通知中最重要的字段，该字段用于确定接收通知的系统该如何提醒用户。
//acme2里面存放的是自定义数据。
```

下面是一个复杂一点的通知体

```
{
  "aps" : {
    "alert" : "You got your emails.", //通知的
    title
    "badge" : 9, //将APP的
    icon 右上角的红色提醒标记为9
    "sound" : "bingbong.aiff" //播放名为
    bingbong的提示音
  },
  "acme1" : "bar",
  "acme2" : 42
}
```

构筑一个后台更新通知

后台更新通知通过定期唤醒您的应用程序，以便在后台刷新其数据，从而改善用户体验。这种后台通知既可以显示的提醒用户，也可以静默地在后台完成工作。那么如何使用这种通知呢？

具体的做法是：

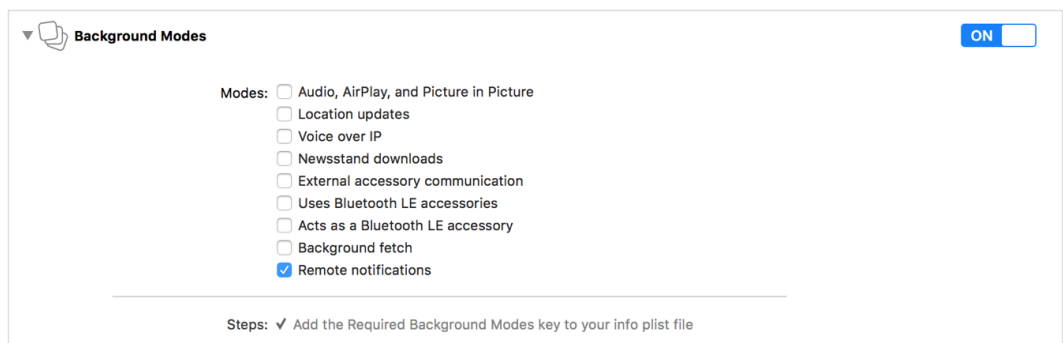
1. 在aps字段中添加content-available字段，并且设置其值为1;

2. 系统在接到后台通知后，会通过

`application:didReceiveRemoteNotification:fetchCompletionHandler:` 通知APP。使用该方法来进行界面的更新操作。

3. 在后台处理远程通知需要开发者将相应的背景模式添加到APP的配置中：

- 选中工程文件；
- 选中对应的Target；
- 选中Capabilities选项；
- 打开Background Modes；
- 选中 Remote notification模式；



背景通知的范例：

```
{
  "aps" : {
    "content-available" : 1
  },
  "acme1" : "bar",
  "acme2" : 42
}
```

本地化远端通知(多语言问题)

本地化远端通知有两种方式：

1. 从后台服务器上提供本地化的内容；
2. 在APP bundle里面存放一个本地化字符串；

其中第一条比较简单，就不多做解释了，就是语言问题放在服务器后台去处理。

第二种简单介绍下，在本地创建一个 `Localizable.strings` 文件，比如写上这么一句：

```
GAME_PLAY_REQUEST_FORMAT" = "%@ and %@ have invited you  
to play Monopoly";
```

搭配下面这个通知体：

```
{  
    "aps" : {  
        "alert" : {  
            "loc-key" : "GAME_PLAY_REQUEST_FORMAT",  
            "loc-args" : [ "Jenna", "Frank"]  
        }  
    }  
} //会根据loc-key来找到对应的本地文案进行替换。
```

就能够实现远端通知搭配本地保存的Strings进行最终显示。