# （IoT）设备管理的可靠消息传递

## 介绍

提供任务关键型消息在任何设备管理框架中都扮演着重要的角色。 这些消息可以执行重要的配置更改和操作说明。 许多这些消息只需要交付一次。 一旦消息传递，获得反馈也起着重要作用，因为它是设备管理框架知道消息已经到达设备的唯一途径。

WSO2的物联网架构支持与流行的消息传递框架（例如APNS（Apple推送通知服务），FCM（基于消息的Google云消息传递），WNS（Windows通知服务）和MQTT）集成。 它还具有与其他许多此类服务集成的扩展功能。

本文讨论了设备管理的可靠消息传递中的一些重要概念以及具有上述消息传递服务的WSO2 IoT Server的特定实现模式。

## 什么是设备管理？

设备有很多种形式。它们与我们在日常工作中使用的计算机，手机，平板电脑，工厂中的机器，家庭中使用的电器，以及允许开发更大型机器的相对较小的硬件不同。无论物理外观或大小如何，设备管理都是这些设备的任何类型的常见需求。

设备管理是管理设备及其操作/功能的过程。它包括维护物理和/或虚拟设备，并包含各种管理任务和流程，以保持设备保持最新并遵守协议。

设备管理的一些典型示例包括：

1. 将驱动程序/固件和应用程序安装到设备上
2. 配置设备以遵守公司协议和策略
3. 实施安全措施以防止损害情况
4. 监测和分析设备数据以获得实时见解

为了使设备符合规则和法规，它必须以可靠的方式接收这些信息。这是可靠和智能的消息（或操作管理）发挥作用，以解决这一要求。

## 运行管理

### 什么是操作？

设备管理涉及遵守企业策略，安全协议以及升级/更新应用程序/固件。 为了实现这一点，设备必须以可读格式或机器语言接收相关信息。 这些相关信息可以通过许多不同的方式作为有效载荷发送到设备。 此外，这些信息可能包含某种类型的命令，指示设备如何行为或某种类型的配置使设备符合企业策略和安全协议。 这甚至可以包含具有所需细节的文件或系统/应用程序升级命令的位置。

### 运营管理的意义

一个值得注意的操作管理功能应该支持以下属性：

1. 无故障 - 无论网络和系统崩溃等外部因素可能导致的故障，都可以传递信息
2. 安全 - 这适用于消息发起，交付和持久性;整个流程必须得到保证
3. 诚信 - 确保只有预期的各方收到消息
4. 适应性 - 通过任何传输方式向任何设备发送任何类型的消息而无论其大小或内容的能力
5. 不重复 - 不会重复发送相同的消息
6. 顺序执行 - 添加到服务器的操作顺序与设备中的执行顺序相同
7. 可追溯性 - 谁发起，什么时候完成，响应是什么，设备状态如何变化
8. 可扩展性 - 随着设备数量的增加，系统性能不应降低，交付完成应该发生。系统应该水平和垂直缩放

## 操作剖析

操作分为两部分：

1. 操作标识符
2. 有效载荷

#### 操作标识符

操作标识符是一个独特的字符串，可以分别区分每个操作。这个独特的标识符依赖于设备类型和运行设备操作系统的感知。有时，操作标识符将用作在设备上执行的命令，例如Android设备中的“RING”。当设备接收到设备标识符开始执行响铃。这种类型的操作不带有操作有效载荷。在WSO2 IoT平台，操作标识符被指定为操作代码。

#### 有效荷载

在设备上需要配置/设置更改时使用操作有效负载。操作负载总是和操作标识符（操作代码）相关联，操作标识符将唯一地标识有效荷载的用途。例如，如果你需要向设备发送WIFI配置，我们在有效载荷中设置SSID和密码，然后作为操作标识符设置Wi-Fi。

#### 操作类型

从概念上讲，这些操作可以分为3种类型。分类是根据操作的语法进行评估的。这些不是绝对的规则，但可以参考以便于理解和实施。任何类型的内容必须作为操作的有效载荷来促进。

* 1. 命令操作
  2. 档案操作
  3. 策略运作

##### 命令操作

如上所述，命令操作仅包括操作标识符（操作码）。这些类型的操作被用来执行简单的功能，不需要设备中的任何有效载荷。例如，REBOOT是Android中的一项命令操作，不需要有效负载，只要设备获得该设备，它就会重新启动。

##### 档案操作

配置文件操作由单个操作标识符（操作代码）和有效负载组成。这些类型的操作被用来执行需要有效载荷的相对复杂的功能。例如，VPN是iOS中的一个配置文件操作，需要使用特定参数的有效负载。

##### 策略运作

策略操作被压入服务中以将策略发送到设备。当设备必须符合企业的指导方针和协议时才使用此类操作。它的操作标识符将始终为“POLICY\_BUNDLE”，并且有效负载将包含某些需要在设备上执行的指令。

简单来说，策略是一组包含规则集的操作。操作集合意味着它包含与有效载荷不同的操作。大多数这些操作都是配置文件操作。例如，一系列操作可能包括Wi-Fi和VPN配置，设备限制和加密设置。有效负载将不包含任何不进行任何状态更改的操作，例如RING，REBOOT。

规则集定义哪些设备应该有这个策略。规则可以有不同的标准，例如设备类型，设备所有权（COPE，BYOD，CPE），设备组，用户和用户角色。这些规则规定了在设备应用策略时如何评估策略。在计算应该应用于设备的有效策略时，将根据策略规则评估设备元数据和通用信息。如果策略规则与设备元数据匹配，则选择该策略作为要在设备上应用的有效策略。因此，有效的策略作为策略操作被添加。

### 操作状态

当一个操作被添加到系统中时，它会经历一个可以称为操作生命周期的生命周期。

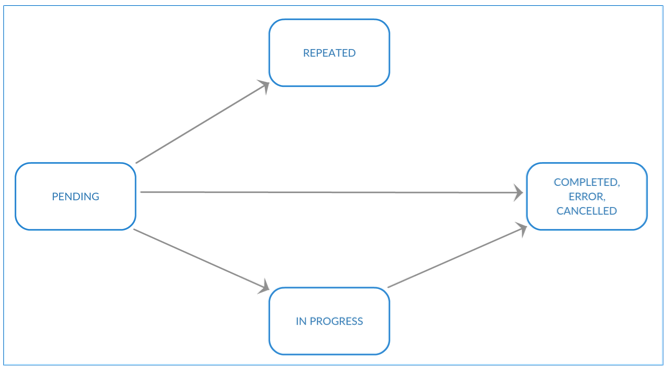


图1

如图1所示，操作始于“PENDING”状态。当设备接收到该操作并执行成功时，该操作将为“完成”。在获得设备操作后，如果执行不成功，则将其标记为“错误”。 “取消”状态在WSO2 IoT服务器中尚不可用，但很快会添加。这种状态是有益的，因为当设备从系统强制移除或由于某种类型的故障（网络故障，设备故障等）而变为不活动时，该设备的所有“待处理”操作将发音为“取消” 。

当设备获得操作但未完全完成时，将使用“正在进行”状态。在这种时候，这种状态很有用。例如，让我们考虑安装应用程序的操作。即使设备接收到该操作，在应用程序安装到设备中之前，它也不能标记为“已完成”。此外，可能需要几小时，直到设备从远程服务器完成应用程序下载。因此，在下载应用程序时，它将被标记为“正在进行”以指示操作仍在执行中。

当该特定类型的另一个操作仍处于未决状态时，添加操作时会关联“REPEATED”状态。该状态只能用于命令操作。较旧的操作将被标记为REPEATED。例如，如果在对特定设备执行另一个RING操作等待执行时添加RING（命令操作），则较旧的操作将更改为REPEATED状态。这种状态的原因是，当某些操作重复执行时，它不会改变设备的状态。这只会浪费资源。

### 如何添加操作

基本上，这些操作是通过暴露的API添加的。由于每种设备类型具有不同类型的操作，因此必须给出最常见的设备类型的具体实现，例如， Android有自己的API，iOS也有相同的API。

有一个基本操作管理组件，用于控制操作存储，选择准确的交付机制和计划交付时间等。它是以通用方式编写的，因此它理解操作的语法，但不了解语义。操作的语法是结构应该如何，语义是由各自的设备类型插件和API给出的。换句话说，操作管理核心理解操作的结构，并且它应该包括操作代码（操作标识符）和操作有效载荷，操作有效载荷对于命令操作是可选的。

随着操作代码从操作变为操作，操作有效负载也取决于设备类型。例如，在Android设备上关闭相机的操作与iOS不同，Android的VPN有效负载与iOS不同。在这些场景中，操作的语义由相应的设备类型插件和API（例如Android和iOS）提供。由于操作管理核心不了解有效负载，相应的设备类型API和插件会将操作转换为设备可读格式。对于Android来说，这只是将操作负载转换为JSON字符串，但对于iOS来说，它要复杂得多，因为它应该转换为Plist。

当从UI或相应的API添加操作时，它将在数据库中创建一个条目。此条目是独立于平台的，因此通常以JSON字符串形式存储。

### 设备如何接收操作？

除非设备是虚拟的，否则设备总是通过网络接收操作。 为了让设备获得操作，代理应该运行在可以转换从服务器发送的内容的设备上。 该代理可以是运行在设备硬件上的固件或运行在固件上的软件代理。

如上所述，当添加操作时，将其保存在服务器的数据库中以保证可靠性，并且应该使用可靠和智能的技术将该操作传送到设备。

以下是关于设备如何以可靠方式接收操作的三种机制，如上所述：

1. 设备轮询（轮询机制）
2. 设备通知（通知机制）
3. 设备唤醒（唤醒机制）

#### 轮询机制

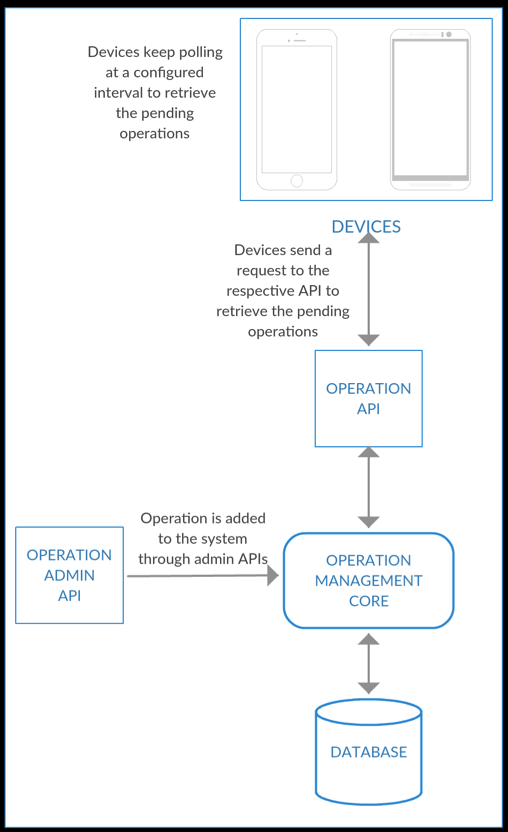


图2

如图2所示，设备以配置的时间间隔轮询服务器。 当操作被添加到系统时，它将处于挂起状态。 当设备轮询时，它会发送一个请求以从系统中检索挂起的操作。 这个请求是一个反向HTTP调用。 这意味着请求由与前一个操作的执行相关的负载组成。 换句话说，先前检索到的操作的响应与下一个请求一起获得，以获得未决操作。

#### 通知机制

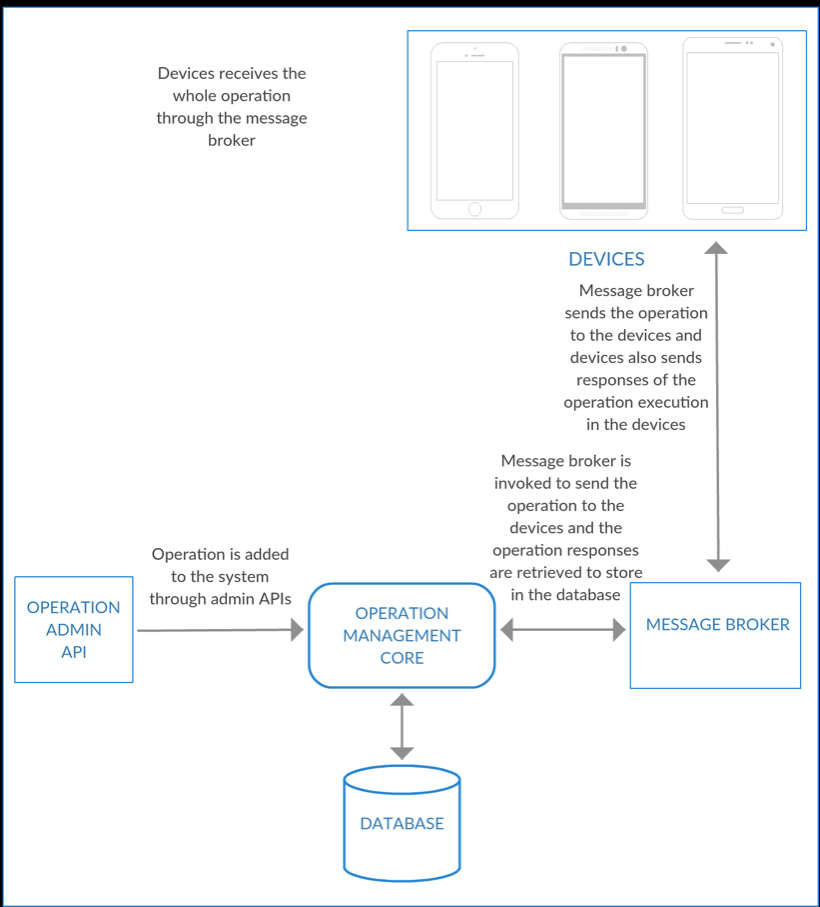


图3

如图3所示，操作通过推送机制从IoT服务器传送到设备。 在这种情况下，我们操纵了一个通过安全通道连接到设备的消息代理。 将操作添加到IoT服务器时，它将整个操作发送到消息代理; 消息代理又将操作发送到设备。 当设备完成操作的执行时，它将响应发送回消息代理，然后将响应推送给IoT服务器。

#### 唤醒机制

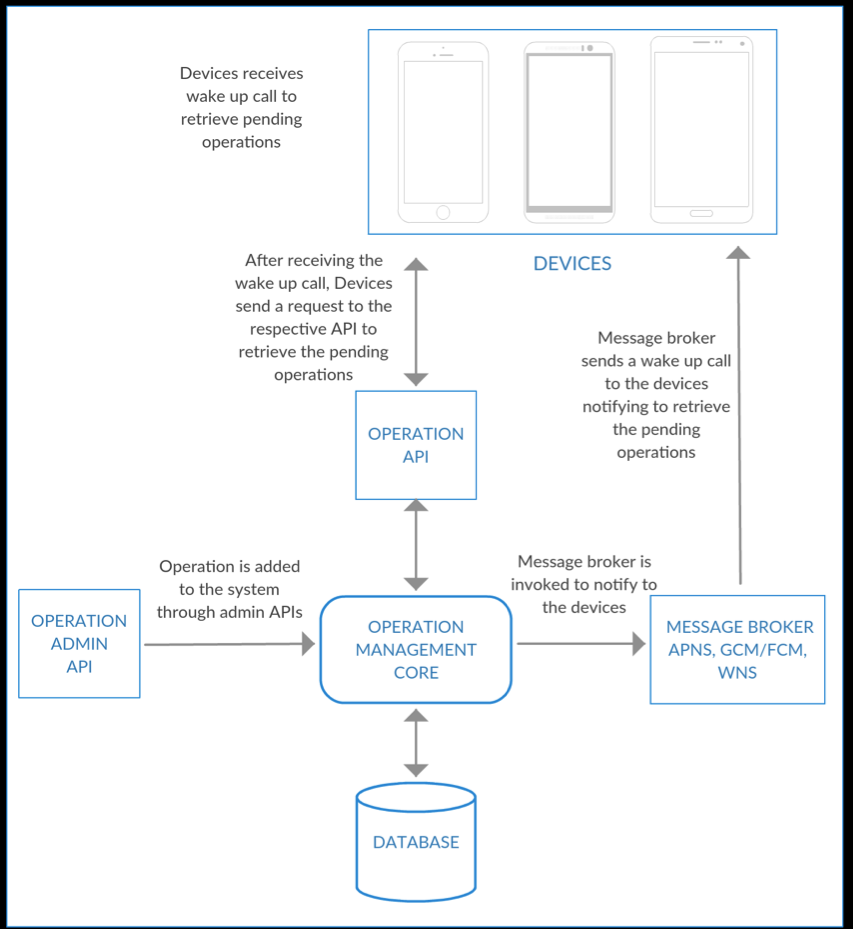


图4

由于某些消息代理的限制，必须使用唤醒通知机制。这是以前的机制，轮询和通知的组合。某些消息代理（如APNS（Apple推送通知服务），FCM（Firebase云消息传递））对消息大小有限制。因此，IoT服务器不能使用这些消息代理将操作发送到设备。

因此，当向IoT服务器添加操作时，会调用消息代理以通知设备有关挂起的操作。这是对设备的唤醒呼叫，并且一旦收到设备，设备就开始发送请求以接收挂起的操作。来自设备的这个请求也是一个反向的HTTP调用，因为它在相同的调用中发送先前操作的响应。

### 通知方法

如上所述，设备如何接收操作，通知和唤醒机制都使用推送通知方法来发送和接收来自设备和服务器的消息。在通知机制中，WSO2 IoT平台具有在MQTT协议中实现的消息代理，该消息代理用于通过消息代理在服务器和设备之间进行通信。

WSO2 IoT服务器支持的其他通知机制是APNS，GCM / FCM，WNS和HTTP。这些用于唤醒机制来调用设备来检索挂起的操作。如果需要，WSO2 IoT Server可通过扩展点来增加新的推送通知机制。如果企业需要编写自己的自定义设备类型和自定义推送通知机制，则WSO2 IoT Server是理想的选择。

### 操作响应

当设备完成操作时，它可能会产生响应。因此，当设备请求下一个挂起操作时，它会将前一操作的响应发送给有效负载。例如，如果执行的操作是DEVICE\_INFO，那么哪个设备有义务发送设备的信息，包括实时和静态的响应。

### 具有操作响应的分析

操作响应是实现系统状态最重要的部分，因为它们提供有关设备位置，状态，固件/应用程序状态等重要信息。 WSO2 IoT Server提供了这些功能来分析实时和批处理模式下的信息。这有助于总结数据以便对其中包含的信息作出结论。此外，它会以毫秒为单位通过实时分析来折中情况。

### 为Android Powered设备实施可靠消息传递

考虑到上述概念，实现了基于Android的设备管理。它支持用于移动设备管理的轮询和唤醒机制的操作管理。对于支持Android的物联网设备，WSO2 IoT平台支持所有通知机制，以实现服务器和设备之间的可靠通信。

在Android中，我们实施了处理设备注册和通信的自定义代理。该代理负责执行设备中的操作，监视，更新固件和应用程序。

#### 使用Android轮询

轮询机制主要用于带有WSO2 IoT Server的Android移动设备。 Android代理默认配置为每分钟接近WSO2 IoT Server。设备注册时提供服务器URL和其他相关配置。

Android设备在安全的REST端点上继续轮询以获取未决操作。在每次请求时，设备都会发送上次操作执行的结果。因此，这个请求是一个反向的HTTP调用。

#### 通过Android唤醒

由于Android代理中的默认设置配置为进行轮询，因此可以在注册时将其更改为唤醒机制。通过这种配置，服务器到设备的通信通过代理进行。

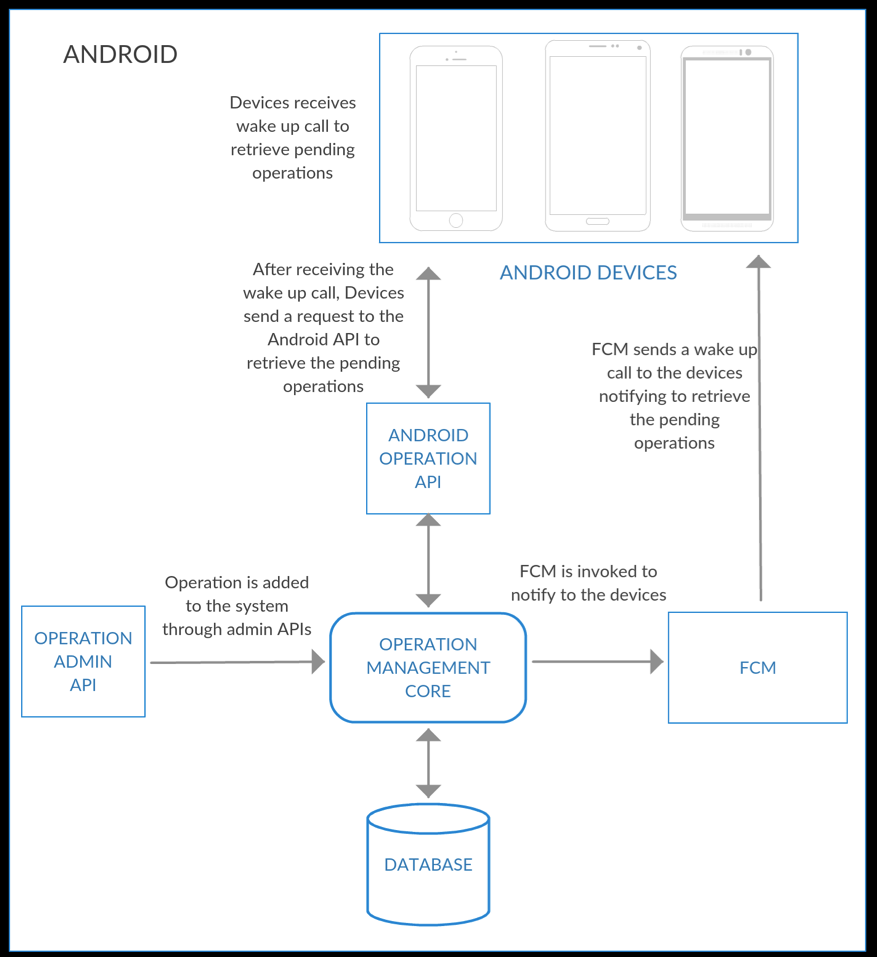


图5

当针对Android设备添加操作时，它会记录在数据库中。 同时，向基于火情的云消息传递（FCM）服务器发送通知以提醒设备，然后设备将调用预定义的安全REST端点来检索未决操作。 这也是一个反向HTTP调用，它接收与前一个操作的执行结果相关的负载。

#### 通知Android

除了采用Android移动设备的标准企业移动管理（EMM）解决方案外，WSO2 IoT Server还拥有所有组件，可将Android移动设备作为物联网传感器使用户从单点评估其物联网产品。 这个特定的Android代理将Android移动设备变成一组传感器，显然使设备可以作为温度计，气压计，速度计等运行。所有这些事件都通过代理发送到分析服务器，以进行实时和智能分析。

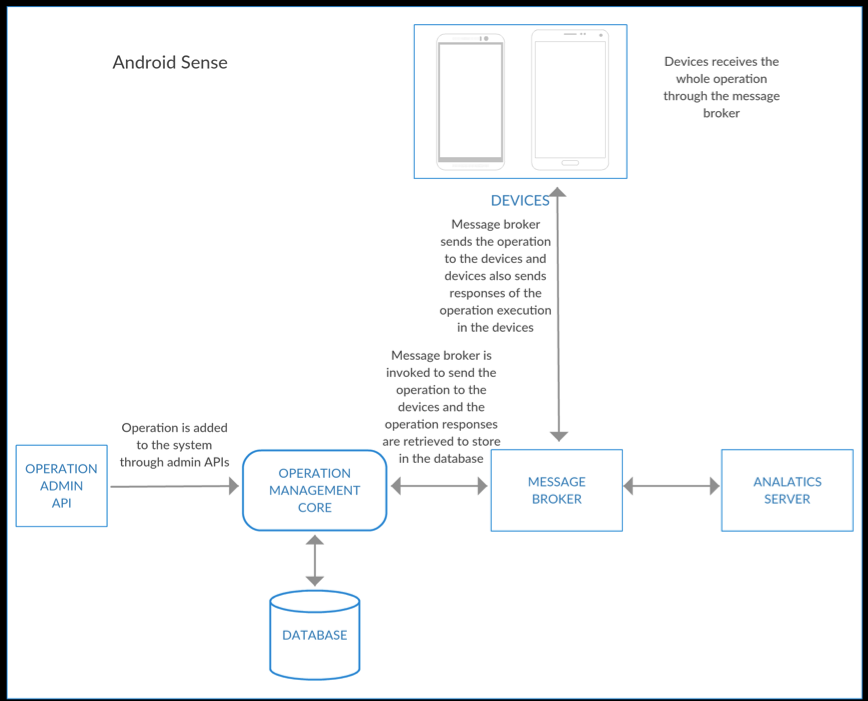


图6

如图6所示，当操作管理核心通过管理API添加操作时，操作将被存储在数据库中。 同时，消息代理将被警告将操作发送到设备。 当设备接收到操作时，它将执行操作并将结果发送回消息代理。 消息代理然后将响应发回给IoT服务器。 这是标准的操作流程。

但是，由于此设备有权从其传感器感测和汇集数据，因此所生成的所有事件也将通过消息代理发送到分析服务器。 分析服务器可以实时和批量模式处理，汇总和分析数据; 因此，所有这些事件都通过智能分析过程发送，以制定决策制定模式，欺诈检测和异常检测以及报告生成。

### 为iOS设备实现可靠消息传递

#### 用iOS唤醒

WSO2 IoT Server支持带有唤醒机制的iOS设备管理。 如图7所示，当通过管理API添加操作时，它被隐藏在数据库中。 同时，它向Apple推送通知服务器发送唤醒消息，然后将其传递给iOS设备。 只要设备收到唤醒通知，它就会建立到IoT服务器的连接以检索挂起的操作。

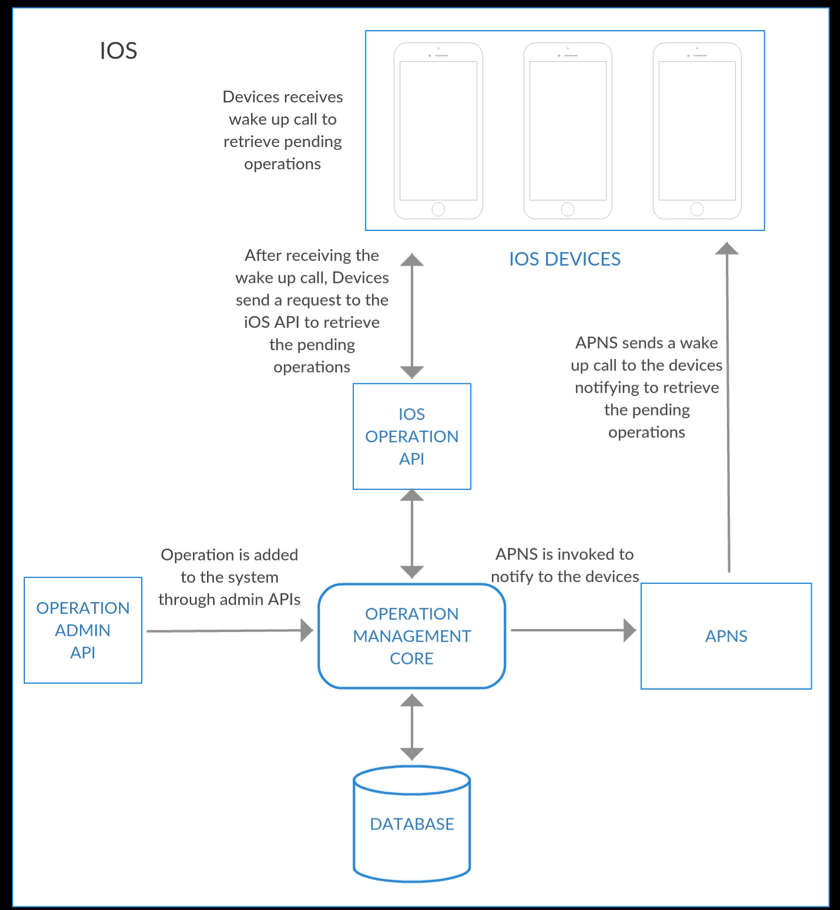


图7

iOS设备具有内置的代理，用于大多数集成在操作系统中的操作及其设计。这个内置的代理与我们在Android系统之上构建的Android代理不同。由于Android代理可以在一次调用中检索所有操作，所以iOS代理的行为有所不同。每个待处理操作堆栈都会针对该设备进行一次调用。如果有10个未决操作，则需要大约11个请求才能完成整个过程。这里，前10个请求用于逐个检索所有操作。当没有可用的操作时，它需要一个400响应代码来停止来自设备的请求流。这就是iOS代理由Apple设计的方式，因此第11个请求将从WSO2 IoT Server收到400响应代码。

#### 通知iOS

由于内置iOS代理处理所有主要功能，因此有几项操作无法完成。检索设备位置是其中之一，因为内置的iOS代理不支持它。因此，WSO2 IoT Server为这些排除操作提供了一个自定义iOS代理。此自定义代理分别支持DEVICE\_LOCATION，NOTIFICATION和RING操作。

如图8所示，当通过管理API添加操作时，将记录存储在数据库中，同时将操作发送到APNS服务器。然后，APNS服务器将其交付给iOS设备。该交付由设备中安装的自定义代理接收，并将执行该操作。操作执行的结果通过与每个操作相关的公开API发送到WSO2 IoT Server。

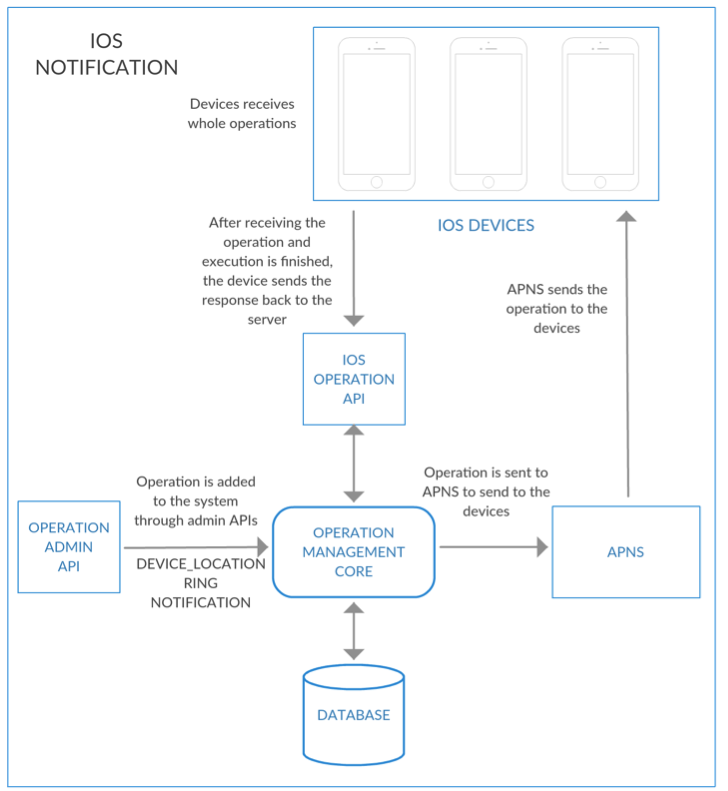


图8

没有任何上述操作被添加到未决操作列表中，因为它们只被内置代理接收。这些操作在其生命周期中采用不同的路径。只要这些操作被添加到服务器并传送到APNS服务，它们就被标记为已完成。但是这些操作可能需要一些时间来接收操作响应，因为不能保证APNS服务器何时将这些操作传送到设备。

此外，上述操作使用JSON作为有效负载消息类型，而内置代理支持的其他操作使用专有消息类型（application / x-apple-aspen-mdm，application / x-apple-aspen-mdm-checkin ）。另外，内置代理使用基于SSL的相互认证和授权机制，而定制代理使用相同的OAuth。

### 为Windows设备实现可靠消息传递

WSO2 IoT Server支持将Windows设备管理作为其核心功能，并且与iOS设备的管理方式有些相似。 Windows设备带有一个内置的软件代理，用于管理完整的管理功能。该代理是支持SyncML（同步标记语言）作为设备和服务器之间的通信协议的OMA DM（开放移动联盟）（设备管理）客户端。与iOS不同，Windows设备目前没有定制的WSO2代理。

#### Windows设备支持轮询和唤醒机制。

##### 使用Windows轮询

Windows轮询与Android轮询机制相同。添加操作时，会将其记录发送到数据库。在设备注册时，它会收到一个管理轮询间隔的配置。因此，设备以配置的时间间隔开始轮询并接收上述操作。下一个轮询请求将包含此操作的执行结果，无论它是否成功或抛出错误。

##### 用Windows唤醒

Windows还支持唤醒通知机制，该机制向设备发送消息以检索挂起的操作。

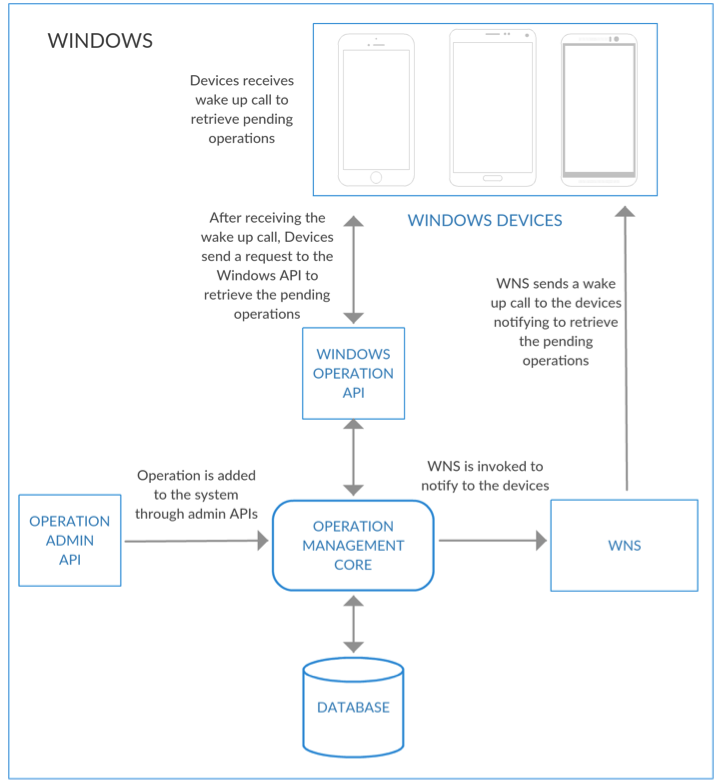


图9

如图9所示，用户通过管理API添加一项操作，并将其记录在数据库中。 同时，启动WNS通知适配器向设备发送唤醒呼叫。 当WNS（Windows通知服务）收到消息时，它会中继到设备。 一旦设备接收到唤醒呼叫，它就开始调用预配置的端点来检索挂起的操作，并且每个请求都会在其有效负载中获取先前执行的操作的结果。

## 结论

WSO2 IoT平台是一款全面的开源产品，可通过智能可靠的消息传递实施涵盖设备管理; 它可以管理具有企业级功能的设备和应用程序。 WSO2 IoT Server已经与众所周知的消息传递框架（如APNS，FCM，WNS和MQTT）集成，并具有故障安全机制并采用高度安全的方式。 它也有许多扩展点，可以方便地插入任何新的传送协议和实现。