# WSO2 IoT - 概述

[WSO2 IoT Server](https://wso2.com/iot)是一款全面的开源IoT解决方案，可帮助企业管理其移动和物联网（IoT）设备。它的所有功能都通过业界标准的Swagger注释的REST APIs展现出来。它允许设备制造商创建自己的设备类型并安全地注册和管理它们。它的设计旨在保护设备及其数据。它还提供分析功能来收集传感器数据，实时查看它们，识别模式并将其转换为响应式操作。您还可以扩展其功能以安全管理移动设备。下面的图1展示了WSO2的物联网架构。

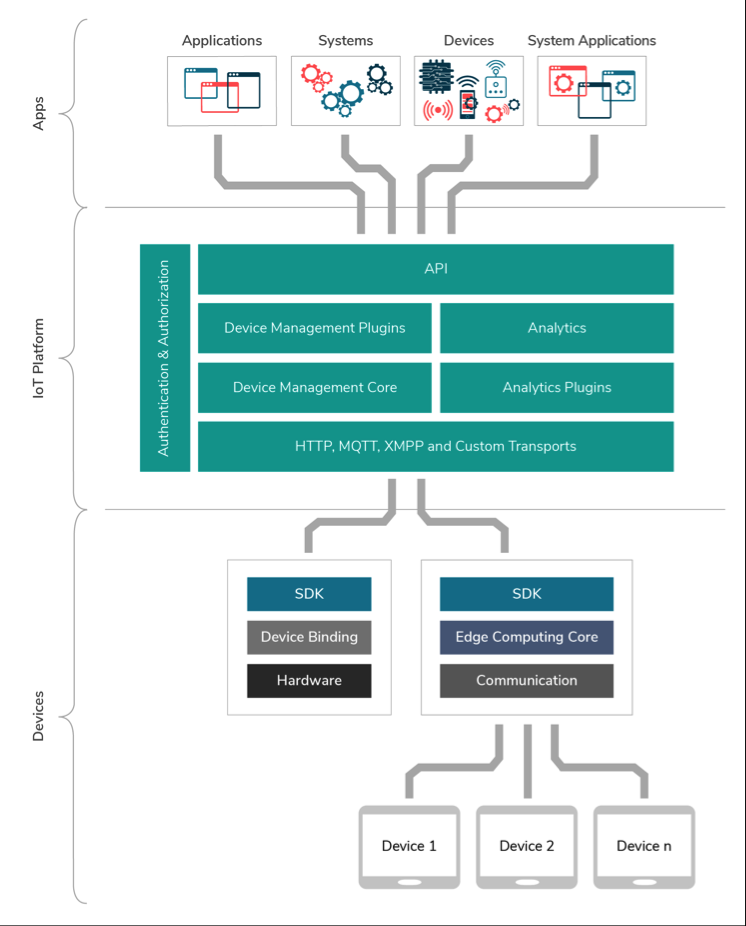


图1

WSO2物联网已经发展到今天的状态。它最初是作为一个管理移动设备的完整平台开始的，后来通过整合管理移动设备以及所有类型的物联网设备的功能，发展为更复杂的系统。最初的版本带有管理Android和iOS的能力，后来增加了Windows移动设备管理和应用程序管理。

# 深入研究WSO2的物联网架构

下面的图2显示了WSO2的IoT产品的深入架构。两种架构图之间的主要区别在于，应用程序和设备都在“交互”层中表示。

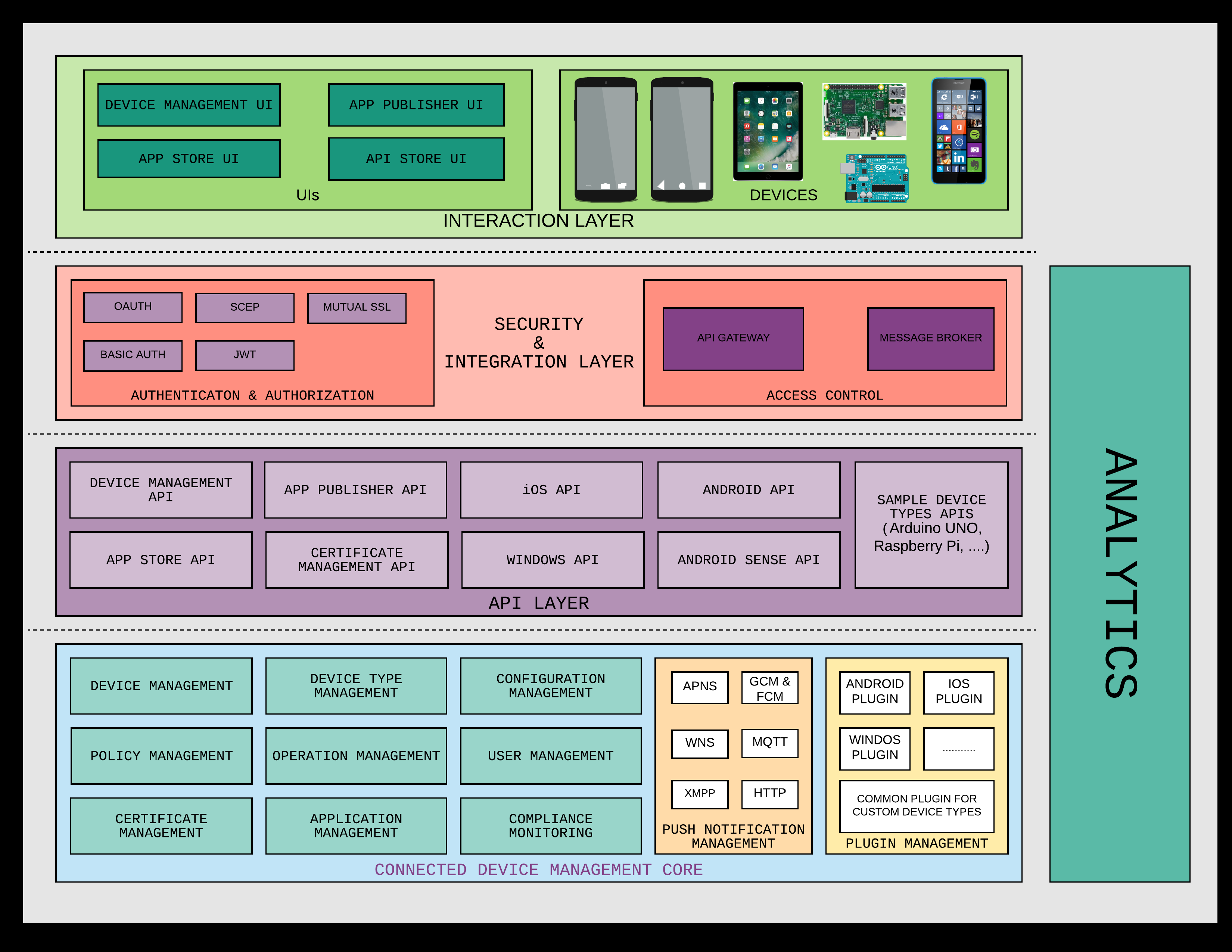


图2

如图2所示，整体架构可以分为以下4个层次。我们仔细看看下面的每一层。

1. 连接的设备管理核心（Connected Device Management Core）
2. API层（API Layer）
3. 安全和集成层（Security and Integration Layer）
4. 交互层（Interaction Layer）

# 连接的设备管理核心（CDM）

这是物联网服务器的核心。它管理和控制WSO2 IoT提供的所有功能。这充当了[WSO2 IoT平台](https://wso2.com/iot)的大脑。它设计有许多扩展点，因此其大部分功能都可以扩展和定制。但是，CDM内核已经发展成为一个更加强大和成熟的平台，任何设备类型都可以轻松集成。

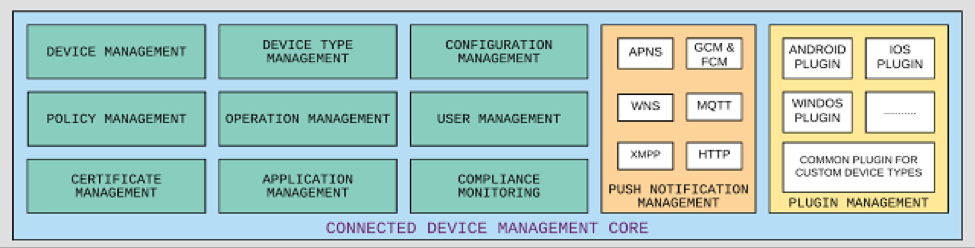


图3

如图3所示，CDM核心包含以下组件：

1. 设备管理（Device Management）
2. 设备类型管理（Device Type Management）
3. 策略管理（Policy Management）
4. 运行管理（Operation Management）
5. 应用管理（Application Management）
6. 配置管理（Configuration Management）
7. 证书管理（Certificate Management）
8. 用户管理（User Management）
9. 推送通知管理（Push Notification Management）
10. 插件管理（Plugin Management）
11. 合规性监测（Compliance Monitoring）

## 设备管理

该部分负责向服务器注册/取消注册设备，管理设备信息（如位置，已安装的应用程序，设备统计信息和运行时信息，如设备内存及其使用情况）。创建/删除设备组并将设备分配给组也属于此组件。

## 设备类型管理

[设备类型管理](https://docs.wso2.com/display/IoTS310/Key+Concepts#KeyConcepts-DeviceTypes)组件具有添加/删除新设备类型的功能，并添加到[WSO2 IoT Server 3.1.0](https://wso2.com/iot)版本中。这使您可以描述设备功能，并生成设备操作和设备详细信息。

## 策略管理和合规性监测

[策略](https://docs.wso2.com/display/IoTS310/Policy+Management)确保属于用户的设备符合公司规则和法规。策略将最大限度地控制设备并降低企业数据的风险。如果设备不符合给定的策略，服务器将收到任何纠正措施的通知，例如重新执行相同的策略。该策略可能包括限制（如相机禁用）以及配置（如VPN或Wi-Fi）。

## 运行管理

[WSO2 IoT Server](https://wso2.com/iot)通过[操作](https://docs.wso2.com/display/IoTS310/Key+Concepts" \l "KeyConcepts-Operations)控制设备。任何消息，配置，限制，策略或命令都会作为操作发送到设备。这些操作要么通过通知机制推送到设备，要么在配置的时间轮询设备。一旦设备执行相应的操作，结果也会发送到服务器并存储在数据库中。

## 应用管理

[应用程序管理](https://docs.wso2.com/display/IoTS310/Mobile+Application+Management)是移动设备管理（MDM）的主要部分。WSO2 IoT提供了添加应用程序并通过应用程序发布者控制台上传apk / ipa文件的方式和手段。此外，使用WSO2 IoT Server注册设备的用户可以从应用商店安装应用程序。应用发布商和商店有能力从Google商店或Apple App Store添加公共应用。您可以在连接到WSO2 IoT Server的设备上安装Web剪辑。另外，管理员在多台设备上安装应用程序的企业注册也可以通过应用商店完成。

## 配置管理

[配置管理](https://docs.wso2.com/display/IoTS310/General+Platform+Configurations)用于添加开始注册设备所需的先决条件和许可协议。在注册iOS设备之前，您需要配置先决条件，例如plist文件和APNS证书。您需要在完成iOS设备注册过程之前添加用户需要接受的许可协议。

## 证书管理

[证书管理](https://docs.wso2.com/display/IoTS310/Certificate+Management)组件具有简单证书注册协议（SCEP）协议的实现，该协议支持通过相互SSL配置注册设备。使用SCEP，您需要提供用于设备认证和授权的SSL证书。在注册过程中，设备会生成证书并将其作为证书签名请求（CSR）与服务器共享。

## 用户管理

[用户管理](https://docs.wso2.com/display/IoTS310/Key+Concepts#KeyConcepts-UserManagement)与[WSO2 IoT](https://wso2.com/iot)捆绑在一起，便于在不同级别管理和控制用户帐户和角色。

WSO2 IoT的用户存储配置为以以下任何一种模式运行：

1. 用户存储以读/写模式运行 - 在读/写模式下，WSO2 IoT读取/写入用户存储
2. 用户存储以只读模式运行 - 在只读模式下，WSO2 IoT保证它不会修改用户存储中的任何数据。WSO2 IoT维护数据库中的角色和权限，但它可以从配置的用户存储中读取用户/角色

WSO2 IoT的使用具有以下特点：

1. 单用户商店的概念可以是外部的，也可以是内部的
2. 能够在公司的LDAP用户存储上以只读/读写模式运行
3. 能够以读/写模式使用Active Directory目录服务（AD DS）和Active Directory轻型目录服务（AD LDS）
4. 角色可以包含来自外部用户存储的用户
5. 改进了外部用户存储的配置功能
6. 能够从LDAP / Active Directory用户存储读取角色
7. 实施WSO2 IoT控制台的管理权限

## 推送通知管理

默认情况下，[WSO2 IoT Server](https://wso2.com/iot)已经为MQTT，XMPP，FCM和APNS实现了推送通知提供程序。由于WSO2 IoT Server旨在引入新的[推送通知机制](https://docs.wso2.com/display/IoTS310/Adding+a+Push+Notification+Provider)，因此任何用户都可以将自己的推送通知库编写为Java扩展并将其安装在WSO2 IoT Server上。

当需要将消息（操作）发送到设备时，推送通知将发挥作用。如上所述，操作发送到设备有三种方式。

1. 轮询
2. 用消息推送通知（操作）
3. 通过唤醒呼叫推送通知

目前，WSO2 IoT Server支持以下推送通知机制：

1. APNS
2. GCM/ FCM
3. WNS
4. MQTT
5. XMPP
6. HTTP

### 轮询

设备将以预设频率触发到配置终端的网络呼叫。该网络调用将由WSO2 IoT服务器接收，挂起的操作列表由服务器发送到设备。一旦设备接收到这些操作，它将执行它们并存储结果，直到下一次网络通话被触发。一旦触发，设备将在相同的网络调用中发送前一个操作列表的结果，以获得下一组待定操作。这与反向HTTP调用类似。一个开箱即用的Android代理以这种方式工作。

### 用消息推送通知（操作）

只要操作启动，该机制就会将整个消息（操作）发送到设备。[WSO2 IoT Server](https://wso2.com/iot)将操作推送到消息代理，代理又将消息发送到设备。设备执行操作后，设备会将响应发送回消息代理，并将响应发送到WSO2 IoT Server。Android感应设备类型以这种方式工作。

### 通过唤醒呼叫推送通知

当相应的消息代理具有关于消息（操作）有效载荷的大小的限制时，使用该推送通知机制。 启动操作时，WSO2 IoT Server会向相应的消息代理发送唤醒消息，然后它会将此消息发送给设备。 然后，设备将通过HTTP向WSO2 IoT服务器发起网络呼叫以检索挂起的操作。 一旦设备接收到挂起的操作，它将存储执行结果，直到下一次唤醒呼叫发生。 当发生唤醒呼叫时，设备将在相同的网络呼叫中发送先前的操作结果以检索下一个未决操作。[APNS](https://docs.wso2.com/display/IoTS310/iOS+Notification+Method)，[GCM/FCM](https://docs.wso2.com/display/IoTS310/Android+Notification+Methods)以这种方式工作。启动操作时，WSO2 IoT Server会调用APNS（如果设备是iOS），将唤醒呼叫发送到设备。 一旦iOS设备收到唤醒信息，它就会调用IoT服务器的预配置端点来接收下一个未决操作。

## 插件管理

[WSO2 IoT Server](https://wso2.com/iot)被设计为可插拔架构。 当引入新的设备类型时，这种可插拔的体系结构很有用。 当用户需要添加自定义设备类型（如果设备很简单）时，添加插件并不是强制性的。 没有任何外部插件被引入IoT服务器，大多数物联网设备都被支持。 但是如果你的设备需要更复杂的操作，编写一个JAVA插件可能是最合适的方法。

WSO2 IoT Servr可以引入3种新的设备类型：

1. [使用设备类型服务和API](https://docs.wso2.com/display/IoTS310/Creating+a+New+Device+Type+via+APIs)
2. [通过设备类型描述符](https://docs.wso2.com/display/IoTS310/Writing+Device+Type+via+the+Template)
3. [通过用给定的接口编写Java扩展](https://docs.wso2.com/display/IoTS310/Writing+Device+Plugins+via+Java+Code)

# API层

这是设备管理功能的核心级别，大多数设备通信协议在[WSO2 IoT Server中作为REST API](https://docs.wso2.com/display/IoTS310/Getting+Started+with+APIs)公开。所有这些API都配备了行业标准Swagger注释，因此可以轻松完成存根或客户端生成。使用设备类型相关的API，因为每种设备类型以不同的方式进行通信和控制。例如，与iOS使用的相比，Android使用不同的消息格式来控制设备。 iOS有自己的协议来与iOS设备可以理解的设备和自己的消息格式进行通信。 WSO2 IoT Server操作核心的实现方式不需要知道发送到设备的消息类型，消息格式和传送机制。这解释了设备类型相关的API，可以以设备可以理解的方式翻译和解释操作。 WSO2 IoT服务器提供简单的设备消息协议，例如JSON，XML，DM SyncML（也称为开放移动联盟设备管理或OMA DM XML）以及用于iOS设备管理的配置文件协议。

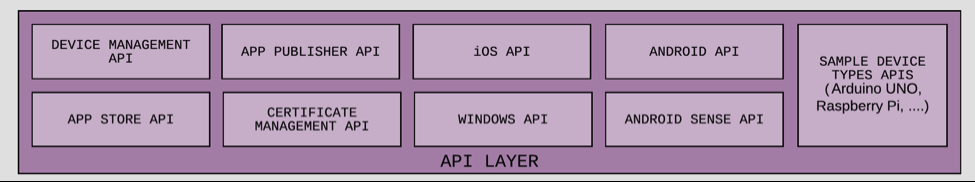


图4

图4显示了WSO2 IoT Server中最常用的REST API，它提供了一系列服务，如下所示。 应该指出的是，这些能力不仅限于下面列出的能力，而且包括更广泛的频谱; 下面的列表包含最常用的功能。

1. [设备管理API](https://docs.wso2.com/display/IoTS300/swagger/?url=https://docs.wso2.com/display/IoTS300/swagger/apis/device-mgt.json)
2. 运行管理
3. 设备组CRUD操作
4. 将设备分配给组
5. 策略管理
6. 用户管理
7. 设备管理
8. 设备信息更新和检索
9. 用户管理
10. [App Publisher API](https://docs.wso2.com/display/IOTS310/iot-server/publisher/)
11. 添加/发布应用程序
12. 为应用程序发布配置工作流程
13. [App Store API](https://docs.wso2.com/display/IOTS310/iot-server/store/)
14. 安装应用程序到自己的设备
15. 安装应用程序到企业设备
16. 卸载应用程序
17. [Android API](https://docs.wso2.com/display/IoTS300/swagger/?url=https://docs.wso2.com/display/IoTS300/swagger/apis/android.json)
18. 注册Android设备
19. 添加操作
20. [iOS API](https://docs.wso2.com/display/IoTS300/swagger/?url=https://docs.wso2.com/display/IoTS300/swagger/apis/iOS.json/)
21. 注册iOS设备
22. 添加操作
23. [Windows API](https://docs.wso2.com/display/IoTS300/swagger/?url=https://docs.wso2.com/display/IoTS300/swagger/apis/windows.json)
24. 注册Windows设备
25. 添加操作

# 安全和集成层

物联网安全和集成在任何物联网解决方案中扮演着非常关键的角色。[WSO2 IoT Server](https://wso2.com/iot)旨在为安全性提供最高优先级。当WSO2物联网服务器用于企业时，它存储用户数据和设备数据。设备存储有关业务的非常机密的信息，因此保护和保护物联网服务器访问控制和相关设备是一项关键要求。此外，将物联网服务器与其他应用程序和用户界面集成在一起，并将其功能安全地展现给外界至关重要。因此，集成在实施物联网服务器时也起着关键作用。

WSO2 IoT Server提供以下安全协议来验证/授权用户和设备：

1. OAuth
2. Basic Auth
3. Mutual SSL
4. SCEP
5. JWT
6. Scopes

启动操作时会发生两级授权。首先，它检查用户是否有权访问提供的API。接下来，它会检查用户是否有权访问或执行给定设备上的操作。如果用户是设备的所有者或设备管理员，或者设备与用户使用设备组共享，则用户可以在设备上执行操

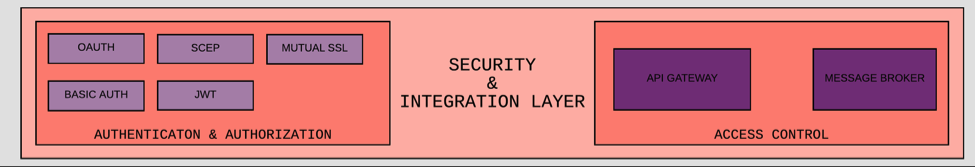


图5

集成通过两个组件进行。 第一个是API网关，第二个是消息代理。 API网关处理HTTP请求和响应的方式与消息代理处理设备和服务器之间的MQTT消息的方式相同。 鉴于IoT服务器支持插件，您可以插入XMPP服务器或COAP服务器，以根据需要通过新的访问控制与设备进行通信。

由于所有管理服务都是作为REST API实施的，因此它们通过API网关安全公开进行集成。 WSO2 IoT Server还将API存储功能作为REST API提供。 用户只需调用API存储终端即可找到所有可用的API。

# 交互层

根据它们与服务器的交互，此层可以分为两个部分。 如图6所示，这两个段都在同一图层中表示。 但是，由于它们与服务器的交互方式不同，因此最好单独标识它们。

1. 设备到服务器的交互
2. 应用程序和外部系统与服务器交互

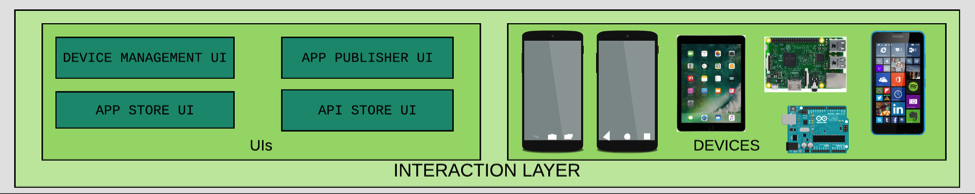


图6

## 设备到服务器交互

这管理着设备如何与物联网平台连接。 设备可以使用不同的通信协议来发送和接收数据，并使用不同的消息格式进行通信。 由于大多数设备类型都支持HTTP通信，因此MQTT和XMPP通信协议也在即装即用的IoT服务器中引入。 针对iOS的APNS，针对Android的GCM / FCM和针对Windows的WNS被添加，因为它们是移动设备通信所需的。 所有通过HTTP调用都是通过API网关安全处理的。

## 应用程序/外部系统与服务器的交互

随着WSO2 IoT服务器公开其所有功能，与外部系统和应用程序的集成非常简单。 可以从任何设备访问安全公开的API，因此系统应用程序也可以扩展。 这些API不仅受到保护，而且在与WSO2 Identity Server连接时还提供了不同的身份验证机制，例如SSO，SAML和XACML。 WSO2 IoT服务器由四个主要的UI控制台组成：

1. 设备管理控制台
2. 应用发布者控制台
3. 应用商店控制台
4. API Store Console（可选）

# 分析

[分析](https://docs.wso2.com/display/IoTS310/WSO2+IoT+Server+Analytics)是设备管理的关键部分。 WSO2的物联网平台不仅可以提供实时分析和批量分析， 机器学习在WSO2 IoT Server中也是一项突出的功能。 WSO2 IoT平台提供智能分析功能，可以检测异常情况并触发即时操作。 它提供了流式分析功能，复杂事件处理和机器学习功能，可帮助您了解事件，映射其影响，识别模式，并在数毫秒内作出实时反应。

## 实时分析

WSO2 IoT Server提供了每秒处理超过百万个事件的功能。这将有助于实时分析设备数据并根据需求做出决策。预测分析支持对未来事件做出正确的业务决策。实时分析可帮助即时刷新仪表板。这可以用来检测异常并立即采取纠正措施。

## 批量分析

WSO2 IoT Server支持对收集的数据进行批处理分析，以显示汇总和总结的视图。汇总的数据也存储在RDBMS数据库中，因此其他应用程序可以使用它来支持公司的生态系统。

## 边缘分析

您可以在设备上运行事件处理功能，与WSO2 IoT Server通信并管理物联网应用所需的边缘计算功能。复杂的事件处理库“Siddhi”可以与任何设备代理打包在一起，以大大增强解决方案的功能，并使您能够在边缘设备上运行实时或批量查询。

# 设备和SDK

WSO2 IoT Server提供SDK支持，可将您的设备与WSO2 IoT平台建立和连接。 由于所有功能都通过API公开，因此SDK支持可以更轻松地创建在设备上运行的新代理应用程序。 SDK本身支持连接设备和服务器的所有基本功能。

# 消息流

WSO2 IoT Server消息流可以分为两部分：

1. 设备到服务器消息流
2. 应用程序到服务器消息流

## 设备到服务器消息流

设备到服务器的消息流如图7所示。WSO2 IoT Server将消息发送给代理，APNS，FCM，GCM和WNS作为虚拟代理进行工作。 MQTT和XMPP是在IoT服务器和设备之间交换消息的实际代理。 由于消息大小的限制，APNS，FCM，GCM和WNS用于通知和唤醒设备。 实际有效负载不通过它们发送到设备。 一旦设备接收到唤醒呼叫，它就会调用IoT服务器来检索实际有效负载。 API网关是WSO2 IoT平台的一部分。

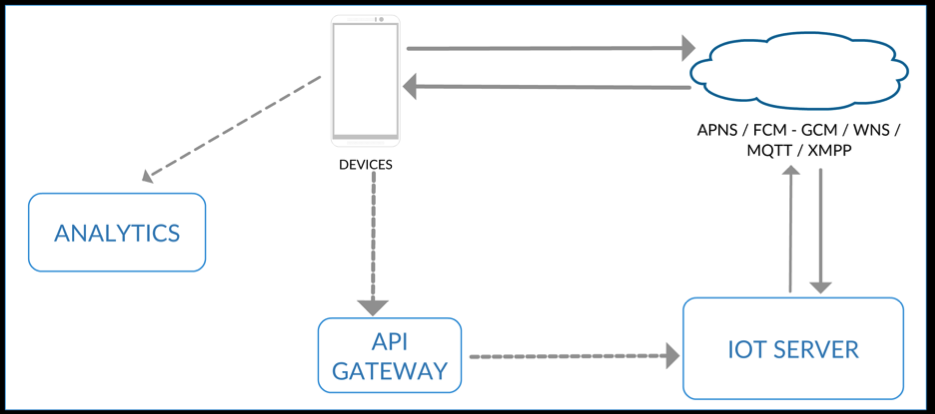


图7

## 应用程序到服务器消息流

外部/系统应用程序通过API网关连接到IoT服务器。 请注意，API网关是WSO2 IoT平台的一部分。

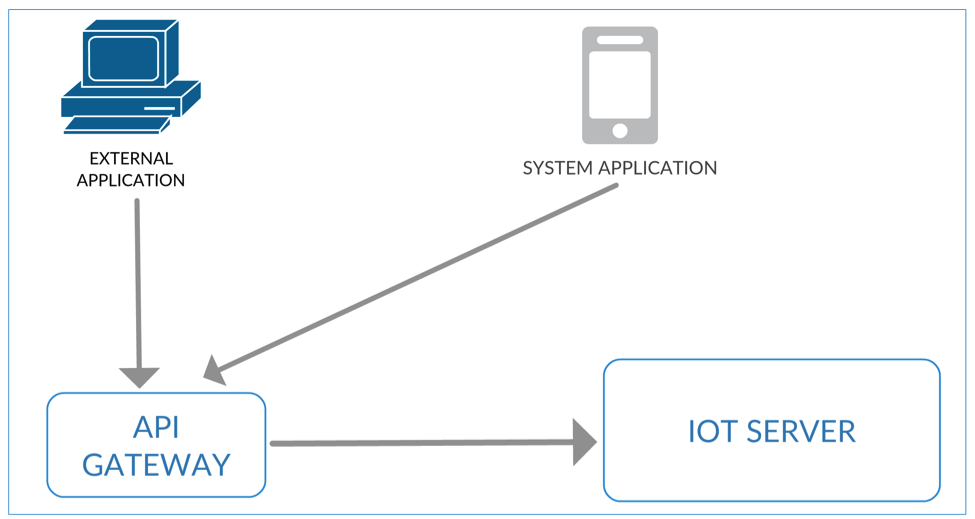


图8

# 总结

WSO2 IoT为需要更好的移动设备管理功能的用户以及开发新设备的OEM供应商提供了优势。 由于它是免费且开放源代码的，所以只需点击几下鼠标即可立即下载完整的功能集并部署到服务器上。 其主要优势包括其完全集成的物联网安全性，行业标准Swagger注释的REST API以及实时，批处理和边缘分析，从而将该平台与其他任何物联网产品区分开来。 从本质上讲，WSO2 IoT Server是一个涵盖物联网领域所有内容的完整平台。