Key Concepts

Let's take a look at some concepts and terminology that you need to know in order to follow the use cases.

- **Devices**
  - Device Ownership
  - Device Types
  - Device Groups
- **Mobile Applications**
- **Configurations**
- **Operations**
- **Policies**
- **User Management**
  - User Roles
  - Users
- **APIs**
- **Extensions**
  - Device Type Extensions
  - Device Agents
  - Transport Extensions
  - Authentication Extensions
  - Analytics Extensions
  - UI Extensions
- **Security**
  - Application-level Security
    - Encryption
    - Authentication
    - Authorization
    - Certificates
    - Tokens
    - Scopes
    - Single Sign-On
    - Role-based Access Control
  - Transport-level Security
- **User Interfaces**

**Devices**

A device is a physical computing unit capable of achieving one or multiple tasks. WSO2 IoT Server enables organizations to enroll, secure, manage, and monitor devices, irrespective of the mobile operator, service provider, or the organization.

**Device Ownership**

In some corporate environments, mobile devices are used to carry out organizational tasks such as email access. These devices are categorized into two main groups based on the ownership:

- **Bring Your Own Device (BYOD):** These devices are owned by the employee and managed by the employer. They are subject to policies and conventions enforced by the employer.
- **Corporate Owned, Personally Enabled (COPE):** These devices are owned and managed by the employer.

**Device Types**

Devices are subdivided into two main groups based on the usage:
• **Mobile devices**: These are handheld devices that are usually used for day-to-day ordinary activities such as making phone calls, sending emails, and setting up alarms. WSO2 IoT Server supports managing Android, iOS and Windows mobile devices.

  - For a quick hands-on experience see [Mobile Device and App Management](#).
  - You can also try the available samples for [Android](#), [iOS](#), and [Windows](#). If you do not have an Android device to enroll with WSO2 IoT Server, see [Android Virtual Device](#).
  - For detailed information on managing mobile devices, see the following sections:
    - Working with Android Devices
    - Working with iOS Devices
    - Working with Windows Devices.

• **IoT devices**: These devices are specifically created to function in a connected environment via the internet. They can collect data via embedded sensors and exchange them with other devices. WSO2 IoT Server supports managing Android Sense, Arduino, Raspberry Pi and custom IoT device types.

  - For a quick hands-on experience, see [Enterprise IoT solution](#).
  - You can also try the available samples for [Android Sense](#), [Arduino](#), and [Raspberry Pi](#). If the available IoT device types do not meet your requirement, see [Creating a New Device Type](#) and [Device Manufacturer Guide](#).

**Device Groups**

WSO2 IoT Server allows you to group multiple enrolled devices in order to monitor multiple devices in one go.

[Back to Top]

**Mobile Applications**

A mobile application is a software application specifically created to run on mobile devices. WSO2 IoT Server enables managing Android, iOS, and Windows mobile applications.

WSO2 IoT Server supports the following two UIs to help **Mobile App Creators/Publishers** manage mobile applications:

- **App Publisher**: This UI enables you to create and manage mobile applications.
- **App Store**: This UI enables you to install and update mobile applications on mobile devices. It also comes with social features such as rating and liking that help Mobile App Creators to understand the popularity and usability of their mobile applications.

For more information on mobile application management:

- For a quick hands-on experience see [Publishing Applications](#).
- You can also try the tutorials to create [Android](#), [iOS](#), and [Web Clip](#) applications, to create versions, and to install all them.
- For detailed information on managing mobile applications, see [Managing Mobile Applications](#)

[Back to Top]

**Configurations**

WSO2 IoT Server enables two kinds of configurations:

- **Preliminary configurations**: These configurations are required to manage mobile devices via WSO2 IoT Server. For more information on preliminary configurations, see the following sections:
  - Product Administration
  - Working with Android Devices
  - Working with iOS Devices

Copyright © WSO2 Inc. 2015-2017
• **Working with Windows Devices**

• **Secondary configurations**: These configurations enable WSO2 IoT Server to integrate with other applications. For more information, see the following sections:

  - Configuring WSO2 IoT Server with WSO2 Data Analytics Server
  - Configuring WSO2 IoT Server with WSO2 Message Broker
  - Configuring WSO2 IoT Server with WSO2 API Manager
  - Configuring WSO2 IoT Server with WSO2 Identity Server
  - Configuring WSO2 IoT Server with third-party MQTT broker

To manage IoT devices supporting HTTP, preliminary configurations are sufficient. To manage other IoT devices you need to perform secondary configurations as well.

---

**Operations**

Each device supports a set of operations depending on its platform (i.e., Android, iOS, Windows), such as screen lock, device unlock, and device reboot. WSO2 IoT Server facilitates these operations to be performed remotely via the **Device Management Console**.

Operations can be performed by the following roles:

- **Device Admin**: Users with this role can perform operations on multiple devices that are under their control.
- **Device Owner**: Users with this role can perform operations on their own devices.

To view the list of operations supported for the Android, iOS, and Windows devices, see [Supported Operations for Mobile Devices](#).

---

**Policies**

A policy is a set of configurations enforced on a device, that influences the device functionality. Policies are able to control the settings on devices, inform the user when the device is not responding as expected and much more. For example, you can disable the camera on a mobile device via a policy.

Policies can be created and applied to devices by the following user roles:

- **Device Admin**: Users with this role can create and enforce policies on multiple devices, under their control, and monitor policy compliance. This behavior is more relevant to mobile device admins in a corporate environment.
- **Device Owner**: Users with this role can create and enforce policies on their own devices. This behavior is more relevant to IoT device owners, but depending on your organizational policies and procedures it can apply to mobile device owners as well.

In WSO2 IoT Server, a collection of policies is called a **profile**. Policy profiles allow you to apply multiple policies to a device collectively. WSO2 IoT Server has predefined policies in place to manage mobile devices and supports creating custom policies for IoT devices.

Let's take a look at how a policy is enforced on a device:

- **Step 1: Filtering based on the Platform (device type)** Policies are filtered based on the mobile platform to match each policy with the platform of the registered device.
- **Step 2: Filtering based on the device ownership type** Next, the policies are filtered based on the device ownership type (i.e., BYOD or COPE) to match with the device ownership type of the registered device.
- **Step 3: Filtering based on the user role or name** The policies are filtered again to match the device owners username or role.
• **Step 4: Enforcing the policy** Finally, the policy having the highest priority out of the pool of filtered policies is enforced on the registered device.

For more information on creating and applying policies, see [Policy Management](#).

---

**User Management**

You can create user accounts, create roles, assign permissions, and manage the devices of the users in your organization using WSO2 IoT Server.

**User Roles**

There are several user categories that are serviced by WSO2 IoT Server:

- **Device Owner**: These users own devices that need to be managed via WSO2 IoT Server.
- **Device Creator**: These users register devices that need to be managed via WSO2 IoT Server. Depending on the organizational structure, this role might be played by a Device Admin.
- **Device Admin**: These users perform administrative tasks related to WSO2 IoT Server, such as user management, configuring security, and installing features. For more information, see [Product Administration](#).
- **Device Manufacturer**: These users create innovative device types that need to be managed by WSO2 IoT Server. For more information, see [Device Manufacturer Guide](#).
- **Mobile App Creator**: These users create mobile applications using WSO2 IoT Server's App Publisher. Depending on the organizational structure, this role might be played by a Mobile App Publisher. For more information on mobile app creation see the following sections:
  - Creating an Android Application
  - Creating an iOS Application
  - Creating a Web Clip
  - Creating a New Version of the Mobile Application
- **Mobile App Publisher**: These users publish the mobile applications created by Mobile App Creators to the App Store.

For more information user role management, see [Managing Roles](#).

**Users**

WSO2 IoT Server enables creating and managing users in your organization and assigning **User Roles** to them. You can create users manually or by integrating WSO2 IoT Server with an existing user store. For more information on user management, see [Managing Users](#).

---

**APIs**

An Application Programming Interfaces (API) is a way of exposing software functionality without revealing its implementation. APIs enable software applications to interact with each other and exchange data. Following are the list of APIs WSO2 IoT Server supports:

- **Device Management APIs**: These APIs expose the device management functionality associated with WSO2 IoT Server [Device Management Console](#). You can also use them to facilitate device management functionality through a third-party UI as well.
- **Device APIs**: These APIs ensure communication between devices and the WSO2 IoT Server.
- **App Management APIs**: These APIs expose app publishing and app portal functionality associated with WSO2 IoT Server [App Publisher](#) and [App Store](#) respectively. You can also use them to facilitate app publishing and app portal functionality through third-party UIs as well.
- **API Management APIs**: These APIs expose API publishing and API portal functionality associated with WSO2 IoT Server.
Certificate Management APIs: These APIs implement Simple Certificate Enrollment Protocol (SCEP) so that WSO2 IoT Server can authenticate and authorize devices with SSL certificates.

For more information, see Device Management REST APIs.

Extensions

WSO2 IoT Server enables creating and managing new devices types that are not available by default. This helps Device Manufacturers to manage custom device types via WSO2 IoT Server.

There are two steps you need to follow to start managing a new device type:

- **Step 01**: Create the new device in WSO2 IoT Server via Device Extensions and install the Device Agent.
- **Step 02**: Implement other necessary configurations required to establish the communication between the new devices and WSO2 IoT Server via Transport Extensions, Authentication Extensions, Analytics Extensions and UI Extensions.

Device Type Extensions

There are three methods that can be used to create a new device type:

- **API based**: You can create a new device type using the available REST APIs. For more information, see Creating a New Device Type via APIs.
- **XML based**: WSO2 IoT Server maintains a template with common elements (e.g. attributes related to license management) generic to all device types. You can create a new device type by extending this template. For more information, see Writing Device Plugins via the Template.
- **Code based**: This enables creating a new device type by writing JAVA code. For more information, see Writing Device Type Plugins via Java Code and Creating a New Device Type via the Maven Archetype.

Device Agents

Device agent is a software program installed on the hardware device that enables communication between the hardware device and WSO2 IoT Server. For more information, see Writing Device Agents.

Transport Extensions

Transport extensions enable you to establish a new communication channel between a device and WSO2 IoT Server. For more information, see Writing Transport Extensions.

Authentication Extensions

By default WSO2 IoT Server supports OAuth, basic auth, mutual SSL and certificate-based authentication mechanisms. If the new device types require some other authentication mechanism, authentication extensions can be used for this purpose.

Analytics Extensions

Analytics extensions help you to integrate with the Analytics profile of WSO2 IoT Server, so that device sensor data of a new device type can be channeled to the Analytics profile. For more information, see Writing Analytics and Monitoring Devices Using Location Based Services.

UI Extensions

This helps you to customize UIs of the new device type. For more information, see Writing UI Extensions.

Security
Security refers to the means through which computer systems are protected from damage and disruption without being compromised to risks and vulnerabilities. WSO2 IoT Server implements security at the application level and transport level.

**Application-level Security**

Application-level security refers to the security requirements at the application level. Following is a list of concepts related to application-level security:

**Encryption**

Encryption is the process of translating/encoding data/messages (plaintext) using an algorithm (cipher) into a secret code (cipher text) that can only be accessed by authorized entities with a secret key or a password.

**Authentication**

Authentication is the process used to distinctly identify a certain entity using the following factors:

- **Knowledge factor**: This is something the user knows, e.g., password, PIN, and security question.
- **Ownership factor**: This is something the user has, e.g., identity card, mobile phone, and security token.
- **Inherence factor**: This is something the user is/does, i.e., biometrics.

Authentication is implemented in either of the following forms:

- **Single-factor authentication**: This mechanism utilizes a single factor to authenticate an entity.
- **Two-factor authentication**: This mechanism utilizes two factors to authenticate an entity, e.g., password and security token.
- **Multi-factor authentication**: This mechanism utilizes more than two factors to authenticate an entity.

WSO2 IoT Server uses OAuth, Basic Auth, JWT, and mutual SSL for authentication.

**Authorization**

Authorization is the process via which an entity is granted permission to access to another entity, e.g., data, resources, system. In general, authorization takes places subsequent to authentication. WSO2 IoT Server uses Role-based access control (RBAC) and scopes to implement authorization.

**Certificates**

A certificate (also known as SSL certificate) is an encryption tool issued by a trusted certification authority (CA) that encrypts data transmitted between a client and a server. WSO2 IoT Server uses Simple Certificate Enrollment Protocol (SCEP) to securely enroll and authenticate iOS devices by creating a certificate for each device. For more information, see Certificate-based Authentication and Certificate Management.

**Tokens**

A token is a credential created by an authentication server that grants an entity to access protected resources. WSO2 IoT Server uses tokens to identify devices and their ability to access protected resources. For more information, see Token-Based Authentication and Generating the Access Token.

**Scopes**

Scopes define the permission model that enables invoking an API. For more information, see Getting the Scope Details of an API and Device Management API Scopes.

**Single Sign-On**

Single sign-on (SSO) enables users to provide their credentials once and obtain access to multiple applications. A user who has already signed in to an application is not prompted for credentials to access other applications until that session terminates.
Once signed in to an application, users are not prompted for their credentials to access other applications until the session terminates. WSO2 IoT Server enables SSO for its web applications, i.e., Device Management Console, API Store, App Publisher, and App Store.

**Role-based Access Control**

Role-based access control (RBAC) is a type of access control that restricts access to authorized users based on their role.

**Transport-level Security**

Transport-level security (TLS) is a mechanism that secures internet and intranet communications. WSO2 IoT Server uses mutual SSL, certificates, and keystores to implement transport-level security.

**User Interfaces**

WSO2 IoT Server comes with the following user interfaces (UIs):

- **Device Management Console**: This UI facilitates all the administrative tasks pertaining to WSO2 IoT Server.
- **API Store**: This UI displays all the APIs associated with WSO2 IoT Server.
- **App Publisher**: This UI enables you to create and manage mobile applications.
- **App Store**: This UI enables you to install and update mobile applications on mobile devices. It also comes with social features such as rating and liking that help Mobile App Creators to understand the popularity and usability of their mobile applications.