DEVICE PROVISIONING

In general, provisioning means "providing" or making something available.

The term is used in a variety of contexts in IT. For example,

- In **Grid Computing**, to provision is to activate a grid component, such as a server, array, or switch, so that it is available for use.
- In a **Storage Area Network** (SAN), storage provisioning is the process of assigning storage to optimize performance.
- In **Telecommunications** terminology, provisioning means providing a product or service, such as wiring or bandwidth.

Before knowing about MOBILE DEVICE PROVISIONING, let us understand a little about **Mobile Device Management (MDM)** :

- MDM is a **Framework** that control, monitor, and manage mobile devices
- Deployed across enterprises or service providers
- Provide these functions remotely:
  - Monitoring
  - Control
  - Manage
- iOS, Android and BlackBerry
  - Each provide their own MDM frameworks
  - Third-party vendors develop products that use the frameworks

DEVICE PROVISIONING

- Device provisioning is a process…
  … where a certificate is issued by the MDM Server for a specific device.

- The issued certificate contains device information that is obtained during the provisioning process.

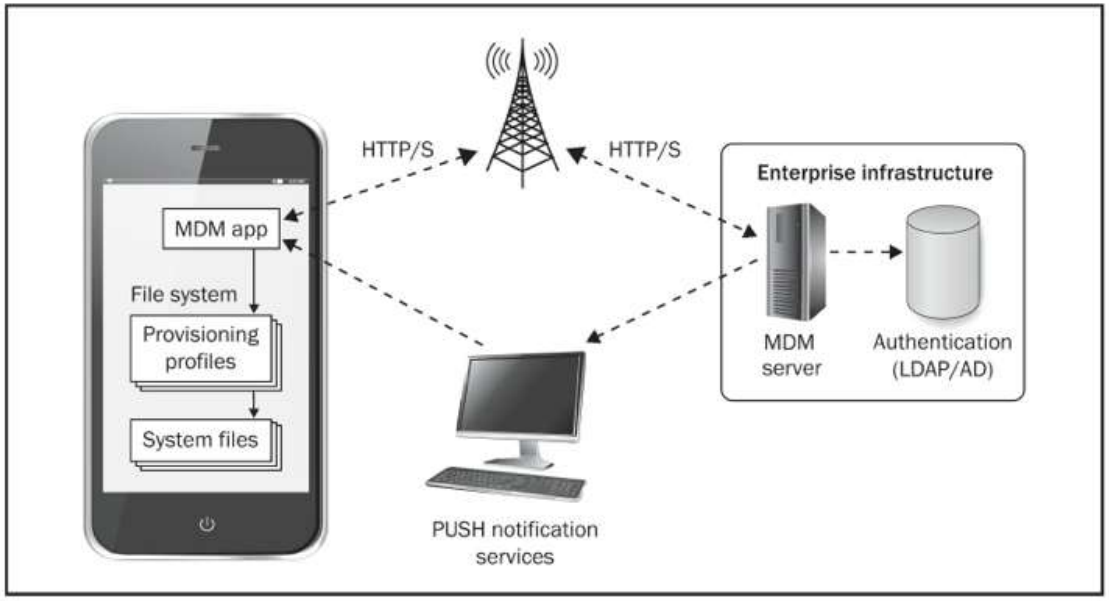
- In other words, Device Provisioning is the process of…

  … attaching a certificate to the Device Identity

- It deploy and enforce policies and restrictions on mobile devices

- Provides access to …

  … resources controlled by the MDM server



Device Identity is similar to user identity that is used to uniquely identify a specific device with the server.

The device ID is automatically generated by the client-side framework when requested by the MDM Server.

Device identity is essential for various features, such as:

a) Push notifications – You want to know which device you are sending the notification to.

b) Reports – You want to know how many devices are using your server.

Provisioning Types :

1. No provisioning
2. Auto provisioning
3. Custom provisioning

1. **No Provisioning**

- **No provisioning** is appropriate for development environments.
- Using **No provisioning** means that the provisioning process is not triggered (requested) by the MDM Server.
- The application obtains the device ID and sends it to the server as-it-is.
- The server does not validate whether this device is allowed to communicate with it.
- The certificate is not issued and not requested at any stage.
- No-provisioning is the default setting for mobile applications.

2. Auto Provisioning

- **Auto provisioning** is an automated one-time process during which a certificate is issued by the MDM Server and sent to the client application.
- **Auto provisioning** is triggered by a server when it requests a provisioned device identity.
- The application obtains the device ID and starts an automated provisioning process.
- The server collects the supplied device information and issues a certificate.
- The certificate is issued to any device that requests it, therefore **Auto provisioning** makes sense only when it is used after a successful application authenticity check.

3. Custom provisioning

- Custom device provisioning is an extension of auto device provisioning.
- With Custom device provisioning you can validate:
    - Certificate Signing Request during initial provisioning flow.
    - Certificate during every application start.

Provisioning Profile

- Provisioning Profile is installed on the device by the MDM client
- It is an XML or text file
- It specifies configuration and provisioning information for the mobile device
- It may be
  - Plain-text
  - Signed
  - Encrypted & signed

iOS

- MDM server sends provisioning profiles through Apple's Mail client (ActiveSync) or the MDM app installed on the device
- Mobile device stores the profiles at
  - `/private/var/mobile/Library/ConfigurationProfiles`
- Stored as XML files (plists) with **.stub** file extensions

Example Provisioning Profile

- Plist files with .stub extensions

```
iPhone:/private/var/mobile/Library/ConfigurationProfiles root# ls -l *.stub
-rw-r--r-- 1 mobile mobile  2516 Apr 17  2012
4598b7ba178f96bae7864be9b88a1545bc3296eaa+800194199.stub
-rw-r--r-- 1 mobile mobile  7533 Oct 17  2011 com_apple_attwifi+3369864630.stub
-rw-r--r-- 1 mobile mobile 35057 Jan  6 10:36
com_good_iphone_policy+1281327003.stub
-rw-r--r-- 1 mobile mobile  2962 Dec  8  2011
f9ba36a2a2360ede0d588fe242bfdbc7cd12c169a+28338739.stub
```

Sample of iOS Provisioning Profile

```
<!DOCTYPE plist PUBLIC "-//Apple//DTD PLIST 1.0//EN"
"http://www.apple.com/DTDs/PropertyList-1.0.dtd">
<plist version="1.0">
<dict>
        .
        .
        <dict>
                <key>MCProfileIsRemovalStub</key>
                <true/>
                <key>PayloadContent</key>
                <dict>
                        <key>ConfirmInstallation</key>
                        <false/>
                        <key>DeviceAttributes</key>
                        <array>
                                <string>UDID</string>
                                <string>IMEI</string>
                                <string>ICCID</string>
                                <string>VERSION</string>
                                <string>PRODUCT</string>
                        </array>
                        <key>EnrollmentIdentityPersistentID</key>
                        <data>
                        aWRudXXXXXXXXXXg
                        </data>
                        <key>URL</key>
                        <string>https://www.xyz.com/abc.do</string>
```

Sample of iOS Provisioning Profile

```
        </dict>
        <key>PayloadDescription</key>
        <string>Install to enroll to encrypted profile service.</string>
        <key>PayloadDisplayName</key>
        <string>iPhone - Security Profile</string>
        <key>PayloadType</key>
        <string>Profile Service</string>
        <key>PayloadUUID</key>
        <string>xxxxx-xxx-xxxx-xxxx-xxxxxxxx</string>
        <key>ProductVersion</key>
        <string>5.1.1</string>
        <key>ProfileData</key>


key>ProfileTrustLevel</key>
        <integer>2</integer>
        <key>ProfileWasEncrypted</key>
        <false/>
        <key>ProfileWasSigned</key>
        <true/>
        <key>ProfileWasTrusted</key>
        <true/>
        <key>SignerCerts</key>
```