

Autor:

Auerswald GmbH & Co. KG

Version:

1.2

Datum:

18.06.2015

Versionsverfolgung

Version	Datum	Änderung	Verfasser
0.1	17.02.2014	Erste Version.	JWe
1.0	01.08.2014	Bilder und Texte angepasst	JWe
1.1	12.06.2015	COMfortel 3600 IP hinzugefügt	Jwe
1.2	18.06.2015	Provisioning von app-spezifischen Daten	JWe

Inhaltsverzeichnis

1 Gültigkeit.....	2
2 Überblick.....	2
2.1 XML und XML-Schema.....	2
2.2 Konfigurationsdaten.....	2
2.3 AppProvisioningService.....	3
2.4 Benutzer App.....	3
3 Sicherheit.....	4
4 Provisioning von app-spezifischen Daten	4

1 Gültigkeit

Dieses Dokument gilt für COMfortel 1400 IP und COMfortel 2600 IP ab Version 2.2A, sowie für COMfortel 3600 IP ab Version 2.2G.

2 Überblick

Provisionierung von Apps kann, besonders in Verbindung mit einem Geräteprovisioning die Verwaltung vieler Geräte deutlich vereinfachen.

Es ermöglicht die Installation, Aktualisierung und Deinstallation von Apps, definiert durch Konfigurationsquellen, die durch einen Systemverantwortlichen bereitgestellt werden können.

Der grundsätzliche Provisionierungsablauf teilt sich in 5 Schritte auf:

1. Konfigurationsdatei beschaffen (Provisioning)
2. Parsen der Datei, Erstellen eines Deltas zur bisherigen Konfiguration
3. Apps von Quellen laden, wenn nötig
4. Installation/Update der geladenen Apps
5. Deinstallation nicht mehr in der Konfiguration vorhandener Apps

Um die beschriebene Funktionalität flexibel abbilden zu können werden unterschiedliche Komponenten bereitgestellt:

- Eine XML-Datei inklusive XML-Schema, mit welchem die auf dem Telefon provisionierten Apps verwaltet werden können.
- Ein AppProvisioning-Service, zuständig für das Laden und Installieren von automatisch zu provisionierenden Apps
- Eine App, die benutzerdefinierte zu provisionierende Apps verwaltet und anzeigt

2.1 XML und XML-Schema

Eine App-Provisioning XML-Datei lässt sich durch das folgende XML-Schema beschreiben. Die XML-Datei enthält ein oder mehrere "*application*" Elemente, die durch eine URL, eine Version und ein Package eindeutig identifizierbar sind. Optionale Elemente für den Namen, eine Beschreibung und ein Icon ermöglichen eine übersichtliche Benutzerpräsentation.

2.2 Konfigurationsdaten

Die Konfigurationsdaten werden um zwei URL-Parameter erweitert: `autoInstallAppProvisioning` und eine `marketAppListUrl`. Diese enthalte jeweils eine URL, die auf eine nach dem in 2.1 beschriebenen Schema XML-Datei verweist.

Die in `autoInstallAppProvisioning` referenzierten Apps werden automatisch durch das Provisioning aktualisiert, während die in `marketAppListUrl` gelisteten Apps durch die Verwaltungsapp durch den Benutzer installiert werden können.

Neben dem Provisioning können die beiden Parameter auch durch ein normales Backup in das Telefon eingespielt werden. Da dieses Vorgehen aber das bereitstellen eines Servers voraussetzt, ist es für die Verwendung in Kombination mit Telefon-Provisionierung vorgesehen.

2.3 AppProvisioningService

Der AppProvisioningService wird aktiviert, wenn er durch den ProvisioningService oder durch eine manuelle Interaktion angestoßen wird. Geschieht dies, lädt der Service die URL, und erhält so eine Datei. Die Datei wird gegen das in Abschnitt 2.1 beschriebene XML-Schema validiert.

Es wird ein Delta erzeugt, um zu erkennen, welche Aktionen für die Pakete durchgeführt werden müssen. Mögliche Aktionen sind Keine, Installation, Update und Deinstallation.

Anschliessend werden wenn nötig die Applikationen geladen und die Aktion wird ausgeführt.

2.4 Benutzer App

Die App stellt eine grafische Oberfläche für eine in Abschnitt 2.1 definierte XML-Datei dar. Der Benutzer kann eine URL angeben, von der die Datei geladen werden soll. Diese wird in der Konfigurationsdaten gespeichert. Liegt bei Start der App bereits eine URL vor, wird diese geladen, validiert, geparsed und dargestellt.

Dem Benutzer werden nur neue und aktualisierte Apps im Market angezeigt, so dass er diese installieren und aktualisieren kann. Die Deinstallation wird in diesem Fall durch Android Boardmittel durchgeführt.

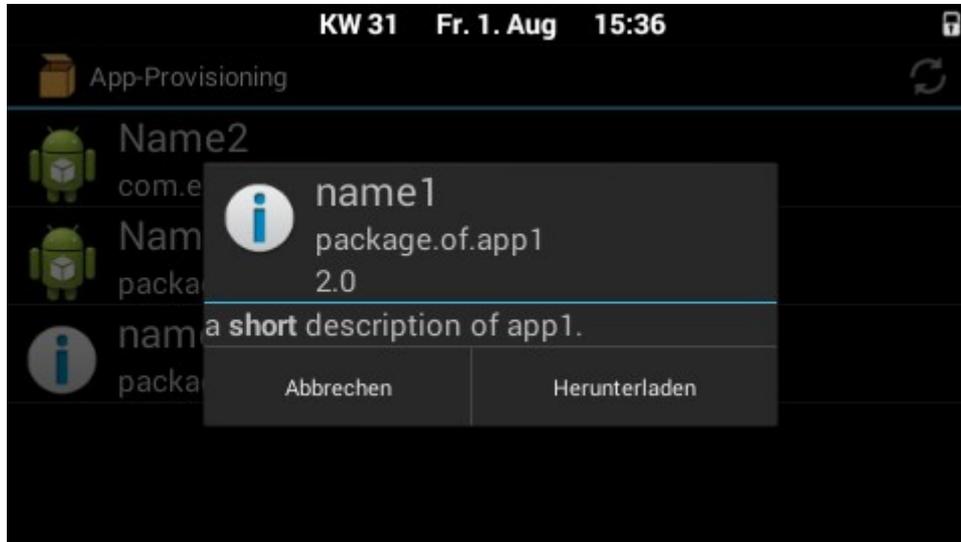
2.4.1 UI

Der Ablauf der Benutzerinteraktion stellt sich wie folgt dar:

1. Start der App



2. Nach Auswählen einer App wird ein Detail- und Downloaddialog dargestellt:



3. Der Benutzer drückt auf Herunterladen. Dies startet den Download, und wenn dieser durchgeführt wurde, wird eine Notification erzeugt, die den Benutzer zum Installationsdialog führt, oder die Datei löscht.

3 Sicherheit

Die URLs können mittels https adressiert werden. Die hier verfügbaren Sicherheitsmechanismen sind nutzbar.

4 Provisioning von app-spezifischen Daten

Um über das Provisioning Daten für Apps bereitstellen zu können, existiert ein ContentProvider. Dieser stellt Key-Value Paare zur Verfügung. Diese können über den Abschnitt `<appData/>` in die Provisioningdaten eingebettet werden. Die dafür benötigte Syntax ist dem `auer_settings.xsd` Schema zu entnehmen.

Um auf diese Daten zuzugreifen, wurde eine Klasse `de.auerswald.appdata.AppDataContract` in das `function_key.jar` eingefügt. Diese Klasse enthält die folgenden Konstanten:

Konstante	Bedeutung
URI	Content URI um auf den Provider zuzugreifen
TYPE	Content Type
KEY	Name des Key Feldes
VALUE	Name des Value Felde
ID	Name des ID Feldes

Der Zugriff auf diese Daten ist nicht eingeschränkt, daher sollten nur Daten über diesen Weg verteilt werden, die öffentlich sind.