



## **RFC 105 - MEG Download**

Confidential, Draft

17 Pages

### **Abstract**

This document describes use cases as guidelines and specification for Data Download and Management service to the OSGi MEG server infrastructure environment. The service needs to work with existing data download and management implementations and specifications, i.e. on the existing infrastructure.

Copyright © OSGi Alliance 2005.

This contribution is made to the OSGi Alliance as MEMBER LICENSED MATERIALS pursuant to the terms of the OSGi Member Agreement and specifically the license rights and warranty disclaimers as set forth in Sections 3.2 and 12.1, respectively.

All company, brand and product names contained within this document may be trademarks that are the sole property of the respective owners.

The above notice must be included on all copies of this document that are made.

---

# 0 Document Information

---

## 0.1 Table of Contents

<b>0 Document Information .....</b>	<b>2</b>
0.1 Table of Contents .....	2
0.2 Terminology and Document Conventions .....	3
0.2.1 Definitions .....	3
0.3 Revision History .....	4
<b>1 Introduction .....</b>	<b>5</b>
<b>2 Application Domain .....</b>	<b>5</b>
<b>3 Problem Description .....</b>	<b>6</b>
<b>4 Requirements .....</b>	<b>6</b>
4.1 Basic Operation and Protocols .....	6
4.2 Off-line operation (PC Agent) .....	7
4.3 Sever-assisted Installation .....	7
4.4 Management Operation .....	7
4.5 Packaging .....	7
4.6 Software Component Discovery .....	7
4.7 Server Side Service Discovery .....	8
4.8 Operator and Device Characteristics .....	8
<b>5 Technical Solution .....</b>	<b>8</b>
5.1 OSGi specific extension into the OMA DLOTA 1.0 Download Descriptor - Normative .....	8
5.1.1 <i>environment</i> - additional Download Descriptor attribute .....	8
5.1.2 The OSGi specific schema .....	9
5.2 Download bindings- Informative .....	9
5.2.1 OMA DLOTA 1.0 binding .....	10
5.2.2 Binding to OMA SyncML DM 1.2 based OSGi Deployment Management Object .....	13
<b>6 Considered Alternatives .....</b>	<b>15</b>
<b>7 Security Considerations .....</b>	<b>16</b>
<b>8 Document Support .....</b>	<b>16</b>
8.1 References .....	16
8.2 Author's Address .....	16

8.3 Acronyms and Abbreviations.....	17
8.4 End of Document.....	17

## 0.2 Terminology and Document Conventions

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY" and "OPTIONAL" in this document are to be interpreted as described in [1].

### 0.2.1 Definitions

<b>Deployment Package (DP)</b>	An OSGi entity, the unit of deployment. Its packaging format is a JAR. It can be downloaded from a Web server.
<b>Download Descriptor (DD)</b>	Metadata about a <i>DP</i> and instructions how to download it. It allows the client device to decide if it has the capabilities to download and install it. It's specified in OMA DLOTA [3].
<b>Download Server</b>	A network infrastructure entity hosting <i>DPs</i> available for download using the OMA DLOTA protocol, e.g. Provisioning Server or Web Server with specialized Servlet. It is responsible for the download transaction from the server perspective. It handles download session management including actions triggered by the installation status report.
<b>Presentation Server</b>	A network infrastructure entity presenting a download service to the user. It is one of the possible discovery mechanisms, e.g. Web Server. The client device may browse a Web or WAP page at the presentation server and be redirected to the Download Server for the OMA Download transaction.
<b>Content Storage</b>	The physical location of the <i>DP</i> to be downloaded.
<b>Installation Notification</b>	A Status Report message from the client to the server. It indicates to the server that the Download Manager has successfully installed the <i>DP</i> , and it (to the best knowledge of the Download Manager) will be made available to the user. See OMA DLOTA [3] specification.
<b>Discovery Application</b>	A user agent in the device that discovers <i>DP</i> on behalf of the user. The End User discovers <i>DP</i> on the Web by using a Web browser or an application specifically created to find applications on dedicated Web sites. Email and MMS messages may contain Web addresses to <i>DP</i> available for downloading. These types of applications are collectively referred to as a <i>Discovery Application</i> .
<b>User UI</b>	User Interface, that provides the means of a dialog with the User.
<b>Download Manager</b>	A management entity in the device responsible for downloading a <i>DP</i> described by a <i>download descriptor</i> and forwarding it to the appropriate OSGi entity for further processing. It covers the

functionality specified for OMA Download Agent[3] and is the content handler of the OSGi DPs. It uses (directly or indirectly) the OSGi MEG Deployment API and is called by device DM framework to download a *DP* addressed. This management entity may be implemented inside or outside of the OSGi platform.

**OSGi MEG Deployment Manager**

An OSGi management entity in the device responsible for the installation of an OSGi DP.

**Device DM framework**

The DM framework in the device represents all the functionalities needed to support device management protocol, maintenance of the DM tree and interactions between the device management related entities. It consists of OMA DM protocol adapter, OSGi DMT Admin and DM Deployment plug-in. DM adapters and plug-ins are not specified they are implementation dependent.

**Download Infrastructure (DI)**

The overall network infrastructure (service) that a client device is exposed to when it wants to select a *DP* and execute a download of it. A download infrastructure is typically constructed with the help of the abstract building blocks Presentation Server, Download Server and Content Storage.

**Device Management Server (DMS)**

A network based entity that issues device management commands to devices and correctly interprets responses sent from the devices.

### 0.3 Revision History

The last named individual in this history is currently responsible for this document.

Revision	Date	Comments
Initial	12.14. 2004	Created based on the first proposal discussed in Boca Raton. Magdolna Gerendai, Nokia, <a href="mailto:magdolna.gerendai@nokia.com">magdolna.gerendai@nokia.com</a>
0.2	12.16. 2004	Editorial changes, Andre Kruetzfeldt, Sun Microsystems, <a href="mailto:andre.kruetzfeldt@sun.com">andre.kruetzfeldt@sun.com</a>
0.3	01.19. 2004	Editorial changes, 5.3 update Magdolna Gerendai, Nokia, <a href="mailto:magdolna.gerendai@nokia.com">magdolna.gerendai@nokia.com</a>
0.4	03.18.2005	DMS initiated use case update and added reference to RFC 106 Magdolna Gerendai, Nokia, <a href="mailto:magdolna.gerendai@nokia.com">magdolna.gerendai@nokia.com</a>
0.5	04.08.2005	Re-working of use cases as bindings, removed Required protocols. Magdolna Gerendai, Nokia, <a href="mailto:magdolna.gerendai@nokia.com">magdolna.gerendai@nokia.com</a>

---

# 1 Introduction

---

This document describes use cases as guidelines for Deployment Package Download and Management service in the OSGi MEG infrastructure environment. The service needs to work with existing data download and management implementations and specifications as much as possible. This is the reason why most part of this document is informative that summarize the usage of the existing protocols in the different scenario. The normative part declares the extension to OMA DLOTA 1.0 with OSGi specific addendum. As download and management are important at both deployment/device management time and at run-time, this usage of this infrastructure service will need to transcend MEG workgroups.

---

## 2 Application Domain

---

This RFC discusses the “run-time” and “deployment time” needs for these download and management server side infrastructure sessions. It focuses in delivery of OSGi Deployment Packages and handling of management events within an operational infrastructure environment.

RFC 106, MEG Deploy Management Object specification [6] addresses client side download and deployment together with management object definition.

For components that need to perform download/management at runtime and/or as part of the deployment process, a common means of defining, registering, configuring and un-registering would be beneficial. This common means should support various means of download and management, including the finalized JSR 124, J2EE Client Provisioning Specification, and the ongoing JSR233 effort in the JCP for defining a J2EE Mobile Device Management and Monitoring Specification.

---

## 3 Problem Description

---

The use cases described in this document focuses on the client-server interactions that support the delivery and management of software components. The main areas are:

- Basic Operation and Protocols
- Off-line Operation (PC-Agent)
- Server-assisted Installation
- Management Operation
- Packaging
- Software Component Discovery
- Server Side Service Discovery
- Operator and Device Characteristics

---

## 4 Requirements

---

---

### 4.1 Basic Operation and Protocols

REQ-DL-01-01. It MUST be possible to control access to the deployment functionality by a flexible policy system. *For example it is possible to restrict, who can install what software over local API calls or what management server can update what software remotely.* (Corresponds to REQ-DEP-01-10)

REQ-DL-01-02. The download MUST support OMA DL 1.0 specification or higher. (Corresponds to REQ-DEP-03-11, REQ-DEP-03-12)

REQ-DL-01-03. The management MUST support OMA DM 1.1.2 specification or higher.

REQ-DL-01-04. The download MUST support OMA CP 1.1 specification or higher.

REQ-DL-01-05. The management MUST support OMA DS 1.1.2 or higher. (Corresponds to REQ-DEP-03-10)

## 4.2 Off-line operation (PC Agent)

REQ-DL-02-01. It MUST be possible to install an UDP or an UDL from a PC without a management server being involved, such as from a CD-ROM drive or over the Internet. (Corresponds to REQ-DEP-02-01 and REQ-DEP-02-02)

REQ-DL-02-02. It MUST be possible to access management functionalities over a local API. *For example it is possible to write a manager that allows installation of software components, provided that the invoking user-application has installation rights.*

## 4.3 Sever-assisted Installation

REQ-DL-03-01. It MAY be possible to install/update UDPs and UDLs on the user's request. The user selects the component to download, and the delivery happens automatically. (Corresponds to REQ-DEP-03-02)

REQ-DL-03-02. If a server (management / provisioning) initiates the installation/update of an UDP or UDL independently of the user, device MAY require user confirmation before initiating download. (Corresponds to REQ-DEP-03-04)

REQ-DL-03-03. It MUST be possible that a server can assemble the contents of an UDL or UDP package with parts (i.e. binaries, settings, resources) custom-selected for a specific device. For example, the relevant application parts may be selected based on the device capability, pre-installed software, etc. (Corresponds to REQ-DEP-03-05)

## 4.4 Management Operation

REQ-DL-04-01. Management model MUST be supported where the secure communication between the remote manager and the mobile device can be established. (Corresponds to REQ-DEP-03-06)

REQ-DL-04-02. The management server SHOULD support dependency resolution to initiate the acquiring and installation of a missing component. (Corresponds to REQ-DEP-03-07)

REQ-DL-04-03. It MUST be possible to customize the management process by scripts, add-ons, etc

## 4.5 Packaging

REQ-DL-05-01. It MUST be possible to package all data (code, settings, resources, other component-specific data) into one file including stocking information that can be expanded into individual files on the device. (Corresponds to REQ-DEP-05-01)

REQ-DL-05-02. It MUST be possible to digitally sign UDLs and UDPs to check their authenticity. *For example it is possible to add company X's signature to the package so that the user or the management system can be sure of the package's origin.* (Corresponds to REQ-DEP-05-03)

## 4.6 Software Component Discovery

REQ-DL-06-01. It MUST be possible to list available software components from the specified content store by device. (Corresponds to REQ-DEP-06-01)

## 4.7 Server Side Service Discovery

REQ-DL-07-01. It MAY be possible to subscribe for a specified software component if the content store supports a subscription model. (Corresponds to REQ-DEP-06-02)

REQ-DL-07-02. It MUST be possible to unsubscribe for a subscribed software component if the content store supports a subscription model and the specified software component is provided by the selected content store. (Corresponds to REQ-DEP-06-03)

REQ-DL-07-03. The management server MUST be able to map management services to content store components.

REQ-DL-07-04. A download of a managed software component MUST trigger the appropriate management server action.

## 4.8 Operator and Device Characteristics

REQ-DL-08-01. It MUST be possible to attach an operator-specific subscriber identifier (e.g. IMSI) to a management server service. (Corresponds to REQ-DEP-07-01)

REQ-DL-08-02. It SHOULD be possible to attach an operator identification code to a management server service. (Corresponds to REQ-DEP-07-02)

REQ-DL-08-03. It MAY be possible to attach a device identification code (e.g. IMEI) to a management server service. (Corresponds to REQ-DEP-07-03)

---

# 5 Technical Solution

---

## 5.1 OSGi specific extension into the OMA DLOTA 1.0 Download Descriptor - Normative

The OMA DLOTA [3] Download Descriptor is extendible. Extensions can be made to the Download Descriptor by defining the extension data in a separate namespace. That way, extension names will not collide with the standard metadata.

### 5.1.1 *environment* - additional Download Descriptor attribute

The JAR content to be downloaded may be processed by special rules of the environment targeted. The *environment* element indicates the specific environment via its XML namespace. This element has an optional



XML attribute named “*envtype*”, which is used to select the appropriate background content handler to process it. The OSGi specific value of this *envtype* attribute is defined below.

<b>Name</b>	environment
<b>Definition</b>	Container for identification of the environment agent, which processes the JAR content. Extension introduced by OSGi.
<b>Status</b>	Download Descriptor: Optional. User Agent: Optional
<b>Datatype</b>	A complex type with an optional “ <i>envtype</i> ” XML attribute as URI.
<b>Refinement</b>	Extension introduced by OSGi.
<b>Comment</b>	The <i>envtype</i> attribute value identifies the content handler of the JAR package.

## 5.1.2 The OSGi specific schema

```
<?xml version="1.0" encoding="UTF-8"?>
<xsd:schema targetNamespace="http://www.osgi.org/xmlns/dd/v.1.0" xmlns:
md="http://www.osgi.org/xmlns/dd/v.1.0" xmlns:xsd="http://www.w3.org/2001/XMLSchema"
elementFormDefault="qualified">
  <xsd:element name="environment" >
    <xsd:complexType>
      <xsd:attribute name="envtype" type="xsd:anyURI" use="required"/>
    </xsd:complexType>
  </xsd:element>
</xsd:schema>
```

The **environment** element indicates that the content is targeted to the OSGi platform.

The **envtype** attribute has the “http://www.osgi.org/xmlns/dd/DP” value indicating the OSGi Deployment Package format. This value is used to select the Download Manager as a content handler for OSGi DP for further processing. The DLOTA DD descriptor refers to the mime type of the JAR package (application/java-archive).

### 5.1.2.1 Example of a DLOTA 1.0 DD of an OSGi Deployment Package

```
<?xml version="1.0" encoding="UTF-8"?>
<media xmlns="http://www.openmobilealliance.org/xmlns/dd"
xmlns:dd="http://www.osgi.org/xmlns/dd/v.1.0" >
  <objectURI>http://acme.com/management/bundle.jar</objectURI>
  <size>1234</size>
  <type>application/java-archive</type>
  <dd:environment envtype="http://www.osgi.org/xmlns/dd/DP" />
</media>
```

## 5.2 Download bindings- Informative

The following chapters describe two download related bindings for OSGi MEG framework: OMA DLOTA 1.0 [3] protocol for content (DP) download and OSGi MEG Deployment Management Object protocol framework [6] when a Device Management Server initiates a download/installation management action. They summarize the usage of these protocols in the different scenario and provide an overview of the desired functionalities based on the OMA protocols.

## 5.2.1 OMA DLOTA 1.0 binding

### 5.2.1.1 User initiated download

#### Description

This use case describes the basic functionality provided by OMA DLOTA [3]. User is browsing with a Discovery Application on his/her device and selects a link to download an OSGi application (Deployment Package). Download Server (Infrastructure) must create the Download Descriptor. The Download Server may need device specific information (e.g. as provided via UAProf or HTTP request headers) to accomplish this.

The server delivers the descriptor to the device. The Download Descriptor may be delivered to the User rather than downloaded, e.g. via a messaging protocol, removable media, etc. Download Manager is called for DD processing. Download Manager and User are able to make informed decisions before initiating the download (e.g. Do I have enough storage space for this content? Do I want to pay for this content?) The device shows a dialog to the user for accepting the download of the DP. The dialog contains e.g. name, vendor and size of the application. If User accepts it the download starts. Download Manager passes the package to the OSGi MEG Deployment Manager for installation.

After the installation the Download Server has optionally received an installation notification indicating the download was a success or failure via an InstallNotify message sent and the User is informed about the outcome of the download/installation.

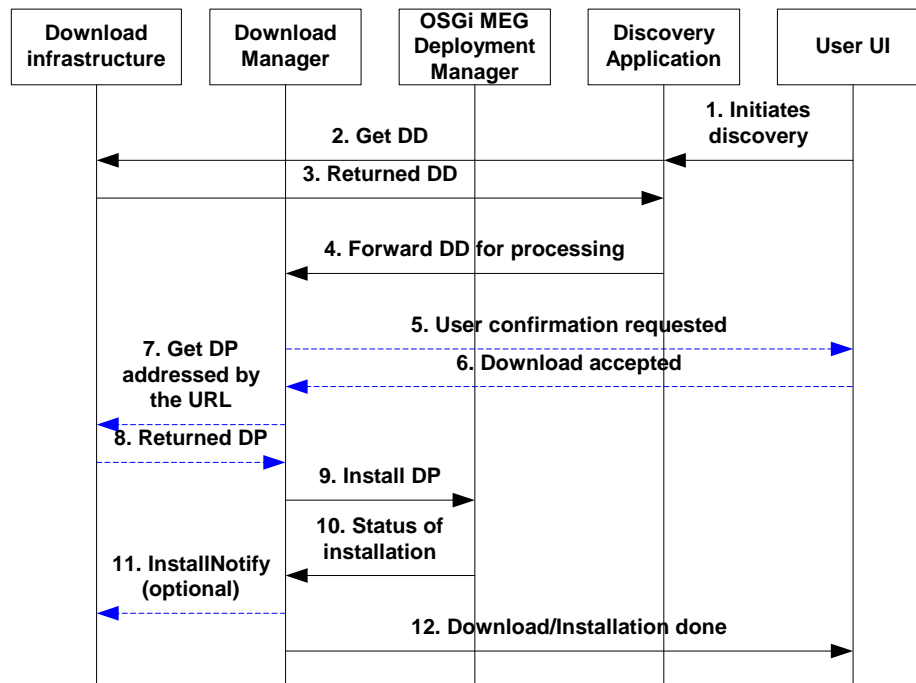


Figure 1. User initiated download

#### Notation:

--- Indicates OMA DLOTA protocol interactions.

#### Normal Flow

1. User starts a Discovery Application, e.g. a browser. The selected link points to the DLOTA DD of a Deployment Package.

2-3. The Browser downloads the Download Descriptor.

4. Download manager as the content handler is called.

5-6. User confirmation for the download/installation is requested.

7-8. OSGi package downloaded to the device.

9-10. DP is passed to the OSGi MEG Deployment Manager for installation. The status is returned.

11. If the installation is successful, and the server requested notification in DD via InstallNotify URI, the notification is sent and the semantic of the notification is the indication of the successful download and installation.

12. User is informed about the outcome of the action.

"The confirmation of a successful download operation (installation notification) is particularly useful in deployments where some kind of pay-per-transaction business model is used." See OMA DLOTA [3].

### *5.2.1.2 Server push from Download Server*

#### Description

Download of a DP is initiated via a push message (see Push message syntax [5.2.1.3]) containing the URL of the DD. The Download Server should have a capability to push the message through a Push Proxy Gateway. The push message is forwarded to the appropriate application, e.g. Browser.. From this point on the use case continues according to 5.2.1.1 step 2.

Download Manager and the User are able to make informed decisions before initiating the download (e.g. Do I have enough storage space for this content? Do I want to pay for this content?) The device shows a dialog to the user for accepting the download of the DP. The dialog contains e.g. name, vendor and size of the application. If User accepts it the download starts. Download Manager passes the package to the OSGi MEG Deployment Manager for installation.

After the installation the Download Server has optionally received an installation notification indicating the download was a success or failure via an InstallNotify message sent and the User is informed about the outcome of the download/installation.

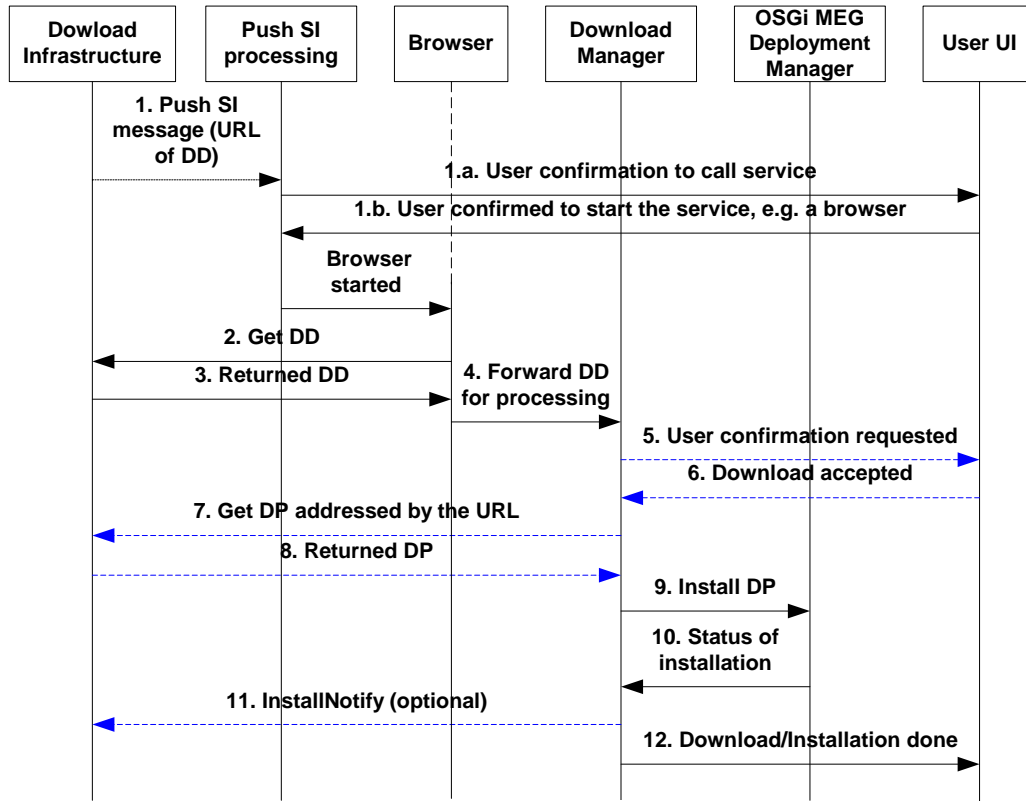


Figure 2. Server push via DLOTA

#### Notation:

— — — Indicates OMA DLOTA protocol interactions.

#### Normal Flow

1. The Download Server pushes the URL of the Download Descriptor to the device. See also Push message syntax in 5.2.1.3.

1.a,b. User confirmation is needed to start the requested service (e.g. browser). The User acknowledged service is launched.

2-3. Application, e.g. a browser downloads the Download Descriptor.

4. Download Manager, as the content handler is called.

5-6. User confirmation for the download/installation is requested.

7-8. OSGi package downloaded to the device.

9-10. DP is passed to the OSGi MEG Deployment Manager for installation. The status is returned.

11. If the installation is successful, and the server requested notification in DD via InstallNotify URI, the notification is sent and the semantic of the notification is the indication of the successful download and installation.

12. User is informed about the outcome of the action.

### 5.2.1.3 Push message syntax

Push message is a WAP Push message having Service Indication content type see [7]. Its content is described in XML. The content format is defined via DTD. When an SI is presented to the user, he or she may choose to load the service indicated by the SI.

The *href* attribute of the *indication* element contains the URL of the DD to be downloaded. All of the other attributes and the *info* element are optional and handled according to the standard.

Example:

```
<?xml version="1.0"?>
  <!DOCTYPE si PUBLIC "-//WAPFORUM//DTD SI 1.0//EN"
    "http://www.wapforum.org/DTD/si.dtd">
  <si>
    <indication href="http://www.download.server.com/tetrisgame.dd"/>
  </si>
```

## 5.2.2 Binding to OMA SyncML DM 1.2 based OSGi Deployment Management Object

### 5.2.2.1 Server push from Device Management Server

Description

Device Management Server sends a Notification for Session Initiation (Alert) to the device. If the device has an OMA DM session already, this step is not needed. User Interaction is controlled by the push message. If User Interaction is needed, DM framework prompts the user about the initiation of a DM session. Device Management Server adds a dynamic node under the Download node and sets the DLOTA Download Descriptor URI of the Deployment Package, the ID and EnvType via Add management commands and requests an Exec command on the Download/<d>/Operations/DownloadAndInstallAndActivate node.

DM framework passes the URL of the DLOTA DD to the Download Manager. The DD or the actual DP (that happens to be a JAR) is on separate server called Content storage. Download Manager gets the DLOTA DD and with the User is able to make informed decisions before initiating the download (e.g. Do I have enough storage space for this content? Do I want to pay for this content?) The device shows a dialog to the user for accepting the download of the DP. The dialog contains e.g. name, vendor and size of the application. If User accepts it the download starts. The downloaded DP is installed via the appropriate MEG Deployment API method invocation. Content Server has optionally received an installation notification indicating the download was a success or failure via an InstallNotify message sent.

At the completion of the installation process DM framework notifies the DMS server of the resulting status through a subsequent client or server initiated OMA DM session to assure that the management server is informed of the execution result.

DMS maintains the state of the individual devices and may request the actual state of the device via Inventory query before (and/or after) the download and installation.

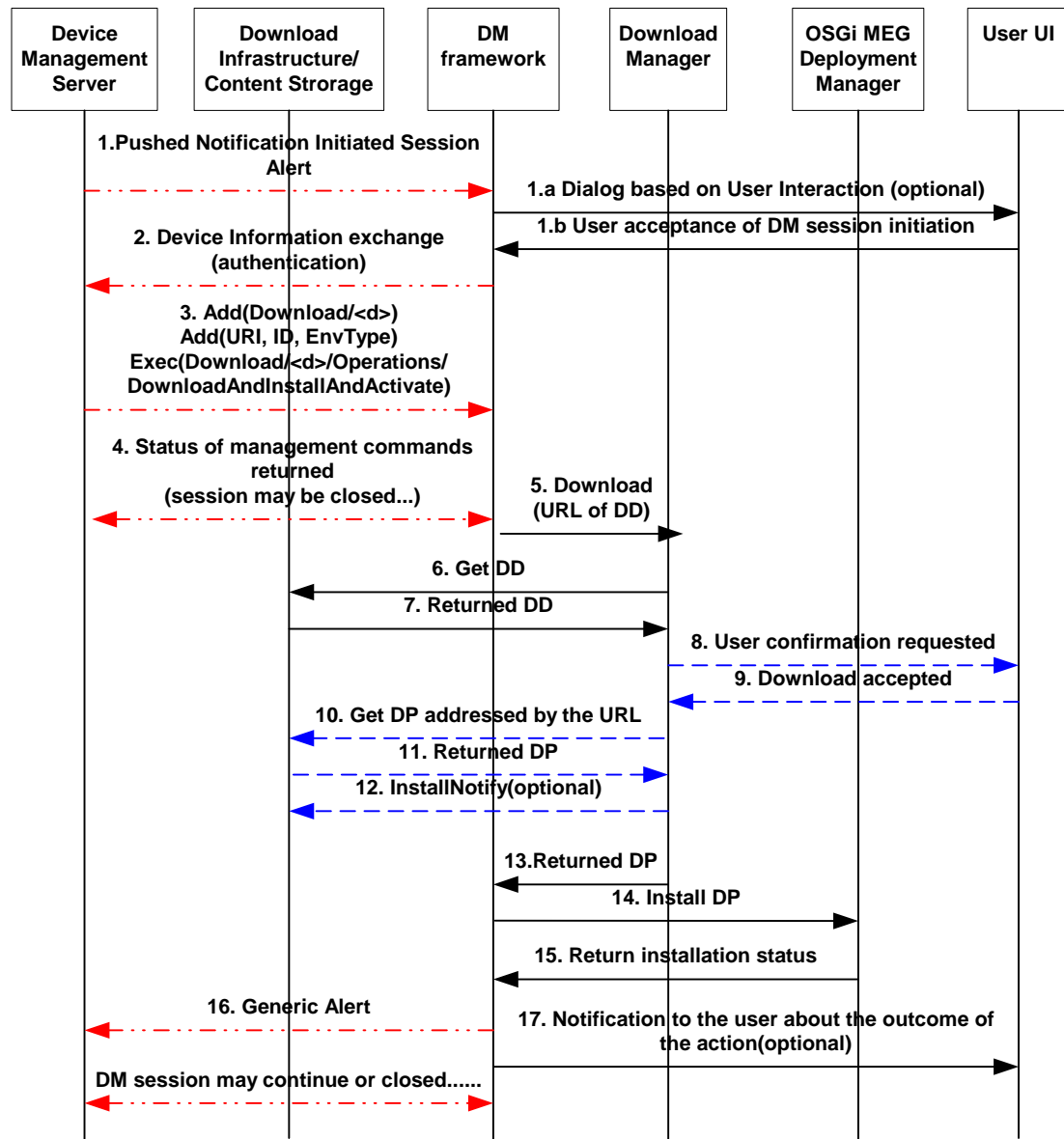


Figure 3. Server push via OMA DM

Notation:

--- Shows OMA DLOTA protocol interactions.

... Indicates OMA DM protocol interactions.

Normal flow

1. Notification Initiated Session Alert OMA DM command is pushed to the device. The Push message may be delivered via a Push Proxy Gateway or DMS may directly communicate with the device if it's able to operate as a PPG. If there is a DM session already between the device and DMS, the session initiation steps (1) aren't needed.

1a,b. The <ui-mode> field of the pushed message indicates the User Interaction Mode requested. This user action may be informative only or may require the user confirmation of the DM session initiation.

2-4. Device DM framework initiates an OMA DM session with the DMS. As a minimum set of selection criteria, DevInfo parameters are sent to the server. This is mandatory for each OMA DM management session. The server sends the DM session management commands to create a dynamic node under the Download node and deliver the URI of the DLOTA DD of the OSGi DP, the ID and EnvType values and the execution request for the download and installation. Parties may close the DM session at this point.

5. Device DM framework calls Download Manager to download the OSGi DP. The URL of the DLOTA DD is passed.

6-9. After downloading the DLOTA DD some checks (e.g. capability) are performed. User confirmation is requested and the size of the package investigated. If the outcome of any of the investigation is negative, further steps for download will be aborted and the reason returned to the DM framework.

10-11. The OSGi content is downloaded.

12. If the installNotifyURI attribute in the Download Descriptor has been explicitly used, the Download Manager informs the status of the download to the server indicated in the installNotifyURI field of the descriptor. This only implies that the OSGi DP has been successfully downloaded. It does not imply that it has been processed.

13. Download Manager returns the downloaded DP.

14-15. Device DM framework installs the DP via OSGi MEG Deployment API and gets back the status of the installation operation.

16. Device DM framework notifies the DMS server of the resulting status through a (subsequent client or server initiated) OMA DM session. The result is provided to the DMS server via a Generic Alert notification. This DM session can continue to carry other DM related management commands from any side or closed if no other actions needed.

17. User is notified about the outcome of the action (optional).

---

## 6 Considered Alternatives

---

*For posterity, record the design alternatives that were considered but rejected along with the reason for rejection. This is especially important for external/earlier solutions that were deemed not applicable.*

---

## 7 Security Considerations

---

*Description of all known vulnerabilities this may either introduce or address as well as scenarios of how the weaknesses could be circumvented.*

---

## 8 Document Support

---

### 8.1 References

- [1]. Bradner, S., Key words for use in RFCs to Indicate Requirement Levels, RFC2119, March 1997.
- [2]. Software Requirements & Specifications. Michael Jackson. ISBN 0-201-87712-0
- [3]. Generic Content Download Over The Air, OMA DLOTA 1.0,  
[http://www.openmobilealliance.org/release\\_program/docs/Download/v1.0-20040625/OMA-Download-OTA-V1\\_0-20040625-A.pdf](http://www.openmobilealliance.org/release_program/docs/Download/v1.0-20040625/OMA-Download-OTA-V1_0-20040625-A.pdf)
- [4]. OMA Client Provisioning V1, [http://www.openmobilealliance.org/release\\_program/cp\\_v11.html](http://www.openmobilealliance.org/release_program/cp_v11.html)
- [5]. "OMA Device Management Protocol, Version 1.2". Open Mobile Alliance™. OMA-TS-DM-Protocol-V1\_2\_0,
- [6]. MEG Deployment Management Object specification, OSGi RFC 106
- [7]. WAP Push, Service Indication Specification,  
<http://www.openmobilealliance.org/tech/affiliates/wap/wapindex.html#previous>
- [8]. User Agent Profiling (UAProf) Specification,  
<http://www.openmobilealliance.org/tech/affiliates/wap/wapindex.html#previous>

---

### 8.2 Author's Address



Name	
Company	
Address	
Voice	
e-mail	

---

### 8.3 Acronyms and Abbreviations

JCP – Java Community Process

JSR – Java Specification Request

OMA – Open Mobile Alliance

---

### 8.4 End of Document