OMA Device Management Bootstrap

Candidate Version 1.3 – 25 May 2010

Open Mobile Alliance

OMA-TS-DM_Bootstrap-V1_3-20100525-C

Contents

Figures

Tables

1. Scope

This document defines the Bootstrap Process, the process by which a Device is brought from a ‘clean’ state, to a state where it is capable to initiate a Management Session with a DM Server.

2. References

2.1 Normative References

[ACw7DM] "OMA DM w7 Application Characteristic". Working file in AC directory:
URL:http://www.openmobilealliance.org/tech/omna/dm-ac/ac_w7_dm-v1_0.txt

[C.S0023-B_v1.0] "Removable User Identity Module For Spread Spectrum Systems", 3GPP2 C.S0023-B version 1.0,
URL:http://www.3gpp2.org/Public_html/specs/C.S0023-B_v1.0_040426.pdf

[DMSecurity] "OMA Device Management Security, Version 1.2". Open Mobile Alliance™.
OMA-TS-DM_Security-V1_2.
URL:http://www.openmobilealliance.org

[DMSTDOBJ] "OMA Device Management Standardized Objects, Version 1.2". Open Mobile Alliance™. OMA-TS-DM_StdObj-V1_2.
URL:http://www.openmobilealliance.org

[DMTND] "OMA Device Management Tree and Description, Version 1.2". Open Mobile Alliance™. OMA-TS-DM_TND-V1_2.
URL:http://www.openmobilealliance.org

[DMTNDS] "OMA Device Management Tree and Description Serialization, Version 1.2". Open Mobile Alliance™.
OMA-TS-DM_TNDS-V1_2.
URL:http://www.openmobilealliance.org

[ERELDCP] "Enabler Release Definition for OMA Client Provisioning Specifications, version 1.1". Open Mobile Alliance ™. OMA-ERELD-ClientProvisioning-V1_1.
URL:http://www.openmobilealliance.org

[ERELDDM] "Enabler Release Definition for OMA Device Management Specifications, version 1.2". Open Mobile Alliance™. OMA-ERELD-DM-V1_2.
URL:http//:www.openmobilealliance.org

[IOPPROC] "OMA Interoperability Policy and Process", Version 1.1, Open Mobile Alliance™, OMA-IOP-Process-V1_1,
URL:http://www.openmobilealliance.org

[ISO7816-4] ISO/IEC 7816-4 (1995): "Information technology - Identification cards - Integrated circuit(s) cards with contacts - Part 4: Inter-industry commands for interchange".

[PKCS#15] PKCS #15 v1.1: Cryptographic Token Information Syntax Standard", RSA Laboratories, June 6, 2000.
URL: ftp://ftp.rsasecurity.com/pub/pkcs/pkcs-15/pkcs-15v1_1.pdf

[PROVBOOT] "Provisioning Bootstrap 1.1". Open Mobile Alliance™. OMA-WAP-ProvBoot-v1_1.
URL:http://www.openmobilealliance.org

[PROVCONT] "Provisioning Content 1.1". Open Mobile Alliance™. OMA-WAP-ProvCont-v1_1.
URL:http://www.openmobilealliance.org

[PROVSC] "Provisioning Smart Card Specification Version 1.1". Open Mobile Alliance™.
OMA-WAP-ProvSC-v1_1.
URL:http://www.openmobilealliance.org

[RFC2119] "Key words for use in RFCs to Indicate Requirement Levels", S. Bradner, March 1997,
URL:http://www.ietf.org/rfc/rfc2119.txt

[SCWS] "Enabler Release Definition for Smartcard-Web-Server", Open Mobile Alliance, OMA-ERELD_Smartcard_Web_Server-V1_1,
URL:http://www.openmobilealliance.org

[TS102.221] "Smart Cards; UICC-Terminal interface; Physical and logical characteristics", (ETSI TS 102 221 release 6),
URL:http://www.etsi.org/

[TS131.102] "Characteristics of the USIM application", (ETSI TS 131.102),
URL:http://www.etsi.org/

[TS151.011] "Specification of the Subscriber Identity Module - Mobile Equipment (SIM-ME) interface", (ETSI TS 151 011),
URL:http://www.etsi.org/

[WBXML1.1] "WAP Binary XML Content Format Specification", WAP Forum™. SPEC-WBXML-19990616.pdf. URL:http://www.openmobilealliance.org

[WBXML1.2] "WAP Binary XML Content Format Specification", WAP Forum™. WAP-154-WBXML. URL:http://www.openmobilealliance.org

[WBXML1.3] "WAP Binary XML Content Format Specification", WAP Forum™. WAP-192-WBXML. URL:http://www.openmobilealliance.org

2.2 Informative References

None.

3. Terminology and Conventions

3.1 Conventions

The key words “MUST”, “MUST NOT”, “REQUIRED”, “SHALL”, “SHALL NOT”, “SHOULD”, “SHOULD NOT”, “RECOMMENDED”, “MAY”, and “OPTIONAL” in this document are to be interpreted as described in [RFC2119].

All sections and appendixes, except “Scope” and “Introduction”, are normative, unless they are explicitly indicated to be informative.

3.2 Definitions

See the DM Tree and Description [DMTND] document for definitions of terms related to the management tree.

2G UICC	UICC activated in a 2G mode that has physical characteristics of UICC [TS102.221] but logical characteristics of SIM [151.011].
3G UICC	UICC activated in a 3G mode that has physical and logical characteristics of the UICC [TS102.221].
ADM	Access condition to an EF which is under the control of the authority which creates this file.
Application	The implementation of a well-defined and related set of functions that perform useful work on behalf of the user. It may consist of software and or hardware elements and associated user interfaces.
Application Identifier	A data element that identifies an application in a smartcard. An application identifier may contain a registered application provider number in which case it is a unique identification for the application. If it contains no application provider number, then this identification may be ambiguous.
Authentication	Authentication is the process of ascertaining the validity of either the DM Client or the DM Server’s identity.
Binary Files	Binary Files are equivalent to transparent files as described in [TS102.221].
Bootstrap Message	A message that is from a Management Authority to the DM Client outside the context of a DM Session
Bootstrap Process	The process of provisioning the DM client to a state where it is able to initiate a management session to a new DM server.
Card Issuer	The organization or entity that owns and provides a smartcard product.
Cardholder	The person or entity presenting a smartcard for uses.
Cardholder Verification	Also called the PIN. Typically a 4 to 8 digit number entered by the cardholder to verify that the cardholder is authorized to use the card.
Command	A message sent by the ME to the smartcard that initiates an action and solicits a response from the smartcard.
Content	Content means data delivered inside of OMA DM messages <Data>-elements.
Data Object Directory Files	Contain directories of data objects (not keys or certificates) ([PKCS#15], section 6.7) known to the PKCS#15 application.
Dedicated File	A file containing access conditions and, optionally, Elementary Files (EFs) or other Dedicated Files.
Device	The Device is, or is to become managed by one or more remote entities (DM Servers). A Device may have many characteristics, and many parameters may be made available for reading, writing, deleting and modifying by a DM Server.
DM Server	The DM Server is an entity that is responsible for maintaining one or more Devices, in whole or in part. Its role is to facilitate the easy maintenance of a Device.
Elementary File	A set of data units or records that share the same identifier. It cannot be a parent of another file.
File Identifier	A 2-byte binary value used to address a file on a smartcard.
Management Session	A continuous connection between the DM Client and the DM Server established for the purpose of carrying out one or more device management operations.

Message Authentication Code	A value computed based on a message hash and some form of shared secret. The MAC is transported outside the Bootstrap Message.
Object Directory File	The mandatory Object Directory File ([PKCS#15], section 6.2) consists of pointers to other EFs, each one containing a directory over PKCS#15 objects of a particular class (here and below, a "directory" means a list of objects).
Path	Concatenation of file identifiers without delimitation. The Path type is defined in [ISO7816-4] sub-clause 5.1.2. If the path starts with the MF identifier (0x3F00), it is an absolute path; otherwise it is a relative path. A relative path must start with the identifier of the current DF (or with the identifier '0x3FFF').
Record	A string of bytes within an EF handled as a single entity.
Smartcard	A device with an embedded microprocessor chip. A smartcard is used for storing data and performing typically security related (cryptographic) operations. A smartcard may be the SIM, the UICC, the R-UIM or a smartcard used in a secondary smartcard reader of a Device.
UICC	A physically secure device, an IC card (or smartcard), that can be inserted and removed from the Device. It may contain one or more applications [TS102.221].

3.3 Abbreviations

2G	Second generation network i.e. GSM
3G	Third generation network i.e. UMTS
ADF	Application Dedicated File
AID	Application Identifier
ALW	Always. Access condition indicating a given function is always accessible.
BIP	Bearer Independent Protocol
CHV	Cardholder Verification
DF	Dedicated File
DIR	Directory File
DM	Device Management
DODF	Data Object Directory Files
EF	Elementary File
FCP	File Control Parameter
IC	Integrated Circuit
ID	Identifier
MAC	Message Authentication Code
ME	Mobile Equipment
MF	Master File
ODF	Object Directory File
OID	Object Identifier
OMA	Open Mobile Alliance
PIN	Personal Identification Number
R-UIM	CDMA Removable User Identity Module
SIM	GSM Subscriber Identity Module
WAP	Wireless Application Protocol

4. Introduction

Other OMA DM specifications define how a management session is established and maintained. However, in order for a DM Client to be able to initiate a management session it must be provisioned with OMA DM settings.

Bootstrap is a process of provisioning the DM Client to a state where it is able to initiate a management session to a new DM Server. DM Clients that have already been bootstrapped can be further bootstrapped to enable the DM Client to initiate a Management Session to new DM Servers or may be rebootstrapped to update existing accounts.

5. Bootstrapping

5.1 Bootstrap scenarios

OMA DM Clients need to be able to function in diverse network environments and using a large set of protocols. This makes it hard to find a 'one size fits all' solution to the bootstrap problem. This section starts with the most basic requirements for bootstrap and continues to define three different processes for bootstrap

5.1.1 Requirements

An OMA DM solution capable of transforming an empty, clean Device into a state where it is able to initiate a management session needs to address these requirements.

- Re-use technology (WAP Push, HTTP Push)
- Tightly standardized and simple ⇒ Highly interoperable
- Self sufficient and complete
- Secure (signed and authenticated)
- Data format should be XML based
- Content mappable to OMA DM management objects
- Transport encoding should be [WBXML1.1], or [WBXML1.2], or [WBXML1.3]

5.1.2 Solutions

This document defines three different ways to perform the bootstrap process.

- Customized bootstrap
  Devices are loaded with OMA DM account and connectivity information at manufacture. Also referred to as factory bootstrap.
- Server initiated bootstrap
  DM Server sends out Bootstrap Message via some push mechanism, e.g. WAP Push or OBEX. DM Server needs to receive the Device address/phone number beforehand.
- Bootstrap from smartcard
  The smartcard is inserted in the Device and the DM Client is bootstrapped from the smartcard.

The DM Client MUST support at least one of these processes for each of the supported profiles (see section 5.2).

5.1.2.1 Customized bootstrap

This is a convenient way to bootstrap a Device from an end user perspective because the user does not have to do anything. In this scenario, an operator orders the Devices pre-configured from a device manufacturer. All the information about the operator's network and device management infrastructure is already in the Devices when they leave the factory. Another advantage of this method is that it is very secure. There is no need to transport sensitive commands and information, e.g. shared secrets, over the air. The method is however not very flexible and not all device manufacturers may provide this service. Not all Devices are sourced via the operator. In this scenario, either the DM Server or the DM Client initiates an OMA DM Management Session after user personalizes and bootstraps the DM Client.

Figure 1 gives an overview of this scenario.

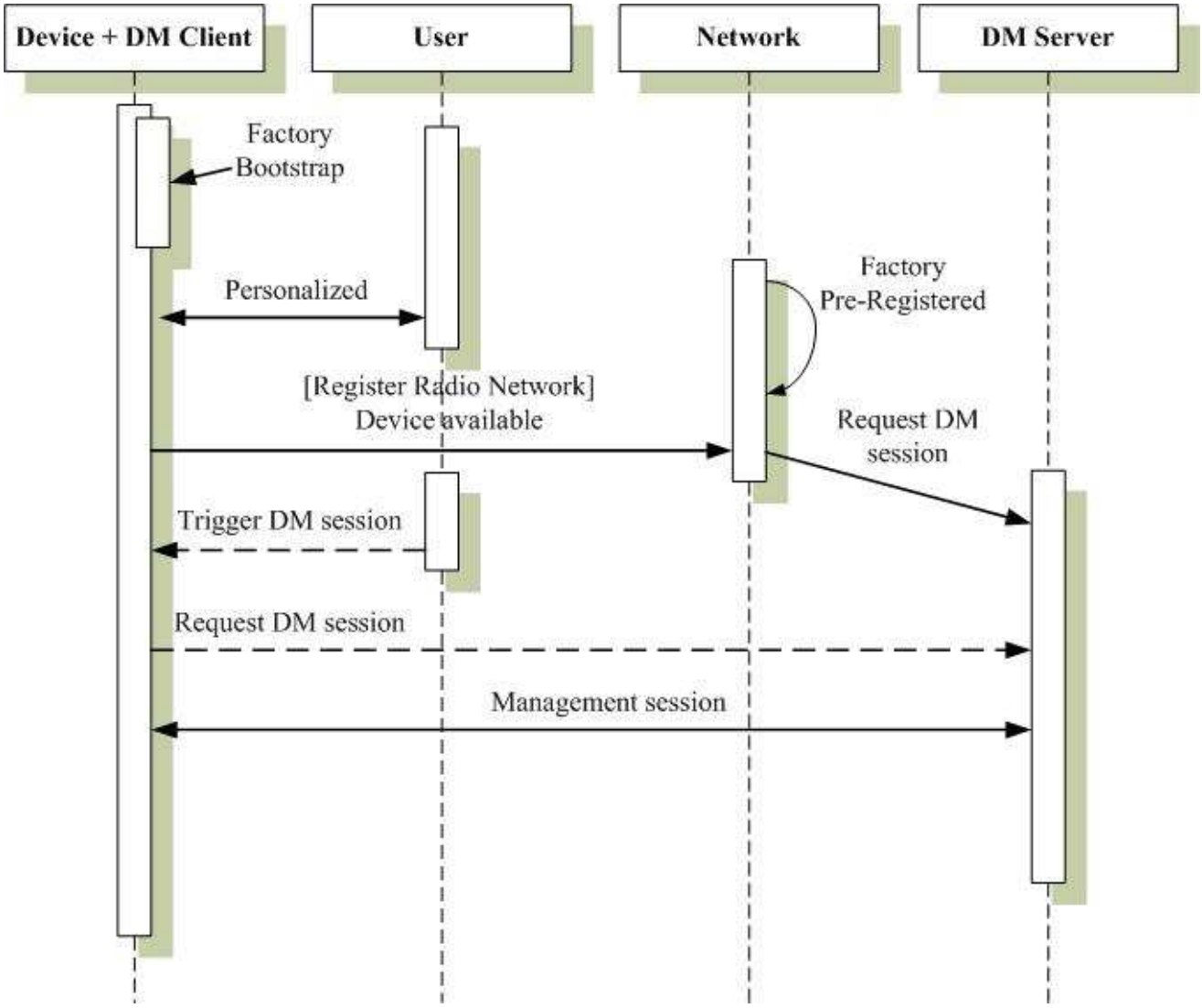

Figure 1: Customized bootstrap

5.1.2.1.1 Bootstrap from smartcard

This is a convenient way to bootstrap a Device from an end user perspective because the user does not have to do anything. In this scenario the DM Client is able to obtain the Bootstrap Message from the smartcard. There is no need to transport sensitive bootstrap commands and information, e.g. shared secrets, over the air. The smartcard is secure, ensuring that the Bootstrap Message is authorized. A Device supporting the smartcard can be bootstrapped for DM without necessarily being purchased from the operator. In this scenario, either the DM Server or the DM Client initiates a Management Session after DM Client bootstraps.

Figure 2 gives an overview of this scenario.

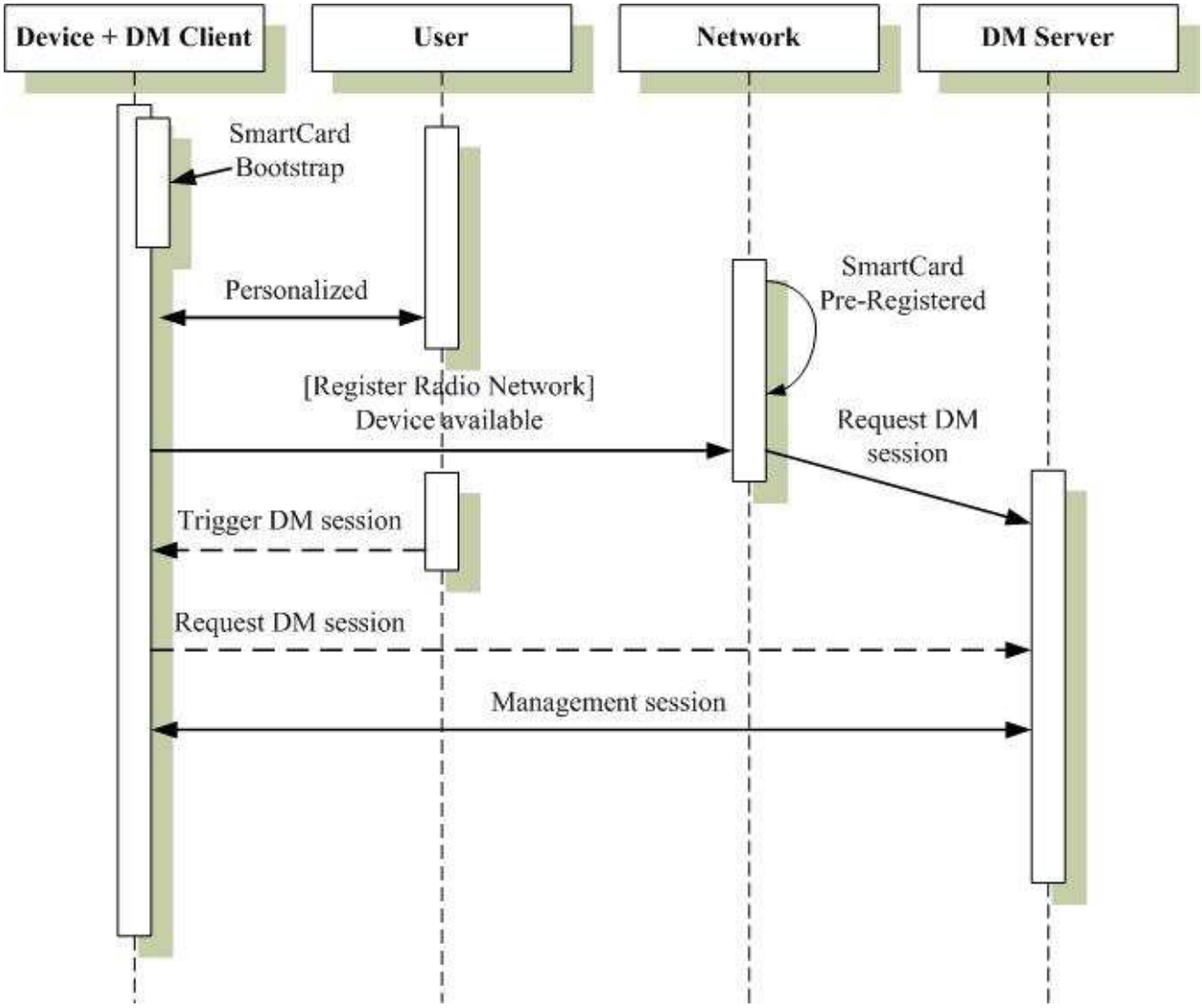

Figure 2: Bootstrap from smartcard

5.1.2.2 Server initiated bootstrap

In this scenario, the Devices leave the assembly line in a clean and empty state. Once a user acquires a Device and personalizes it, e.g. by inserting a SIM, the prerequisites for this process are in place. The problem is now to inform the DM Server of the identity, address or phone number of the device and this can be achieved in many ways.

- It could be done at the point-of-sales where a sales system ties in with the management system and delivers the information.
- It could be done through a self-service web site where the user enters her own phone number.
- It could be done by the network the first time the Device registers to the network. When this happens a trigger could be sent from the core network to the DM Server with the number used by the Device.
- It could be done with a voice prompt system where the user is prompted to key in her phone number.

Regardless of how the phone number or Device address reaches the DM Server, the DM Server is now in a position where it can send out a Bootstrap Message. This message, whose structure and content are defined in this document, contains enough

information for the DM Client to be able to initiate a management session with the device management server that sent out the Bootstrap Message.

It is important that DM Clients accept Bootstrap Messages only from authorized servers [DMSecurity].

Figure 3 gives an overview of this scenario.

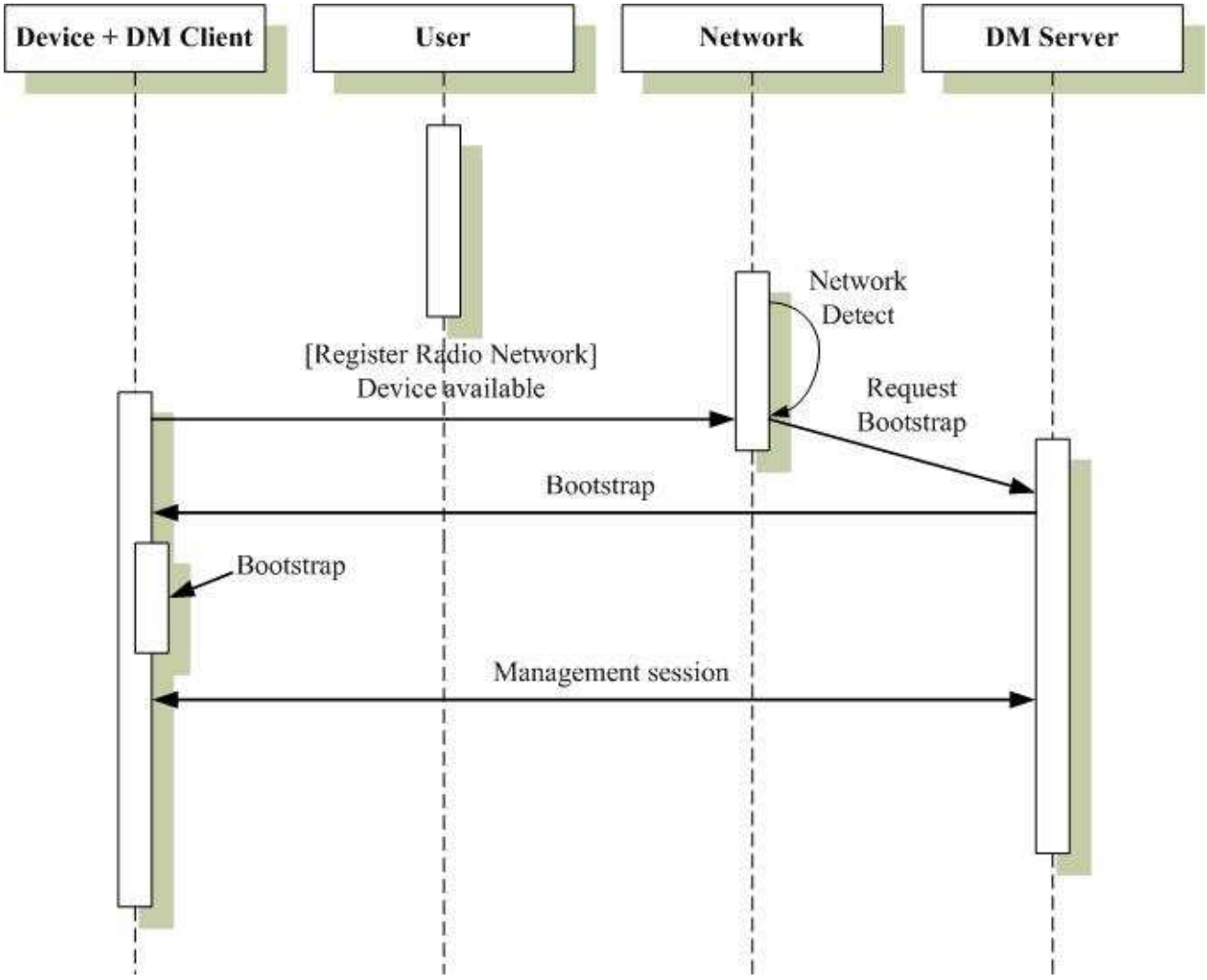

Figure 3: Server initiated bootstrap

5.2 Bootstrap profiles

OMA DM has been designed to meet the management requirements of many different types of devices. For some of these device types there already exists a bootstrap or provisioning mechanism. In these cases OMA DM leverages the existing mechanisms so that backwards compatibility and simple deployment can be achieved. To define how different kinds of devices can be bootstrapped and to specify how OMA DM leverages existing standards this document introduces the concept of bootstrap profiles. Each profile defines its own security, transport and data format. The Device Management Profile MUST be supported. Any other particular profile MAY be supported.

Currently two profiles are planned, but as interest in OMA DM grows and usage of it increases more profiles can be added. The two profiles are:

OMA Client Provisioning

This profile specifies alignment of two existing enablers – OMA Client Provisioning [ERELDCP] and OMA Device Management [ERELDDM]. The profile defines how the information provisioned using OMA Client Provisioning can be transferred to the management tree specified in the OMA Device Management. In this profile at least the mapping of w7 (DM account) information to the management tree needs to be supported, but other provisioning information can also be mapped to the management tree.

OMA Device Management

This profile defines how the OMA Device Management [ERELDDM] can be used for bootstrapping.

5.3 OMA Client Provisioning Profile

OMA Client Provisioning enabler [ERELDCP] is designed to provision the initial configuration information to Devices, and can be used with OMA Device Management enabler [ERELDDM] to subsequently add, update, delete and retrieve all kind of data. The chapter specifies the mapping of the Client Provisioning information to the Device Management management tree in a way that later management for the provisioned parameters is possible in case both Client Provisioning and Device Management enablers are supported by the Device.

The content of the provisioning message is based on the OMA Provisioning Content Specification [PROVCONT]. In order to enable the usage of the OMA Provisioning Content Specification within the OMA Device Management framework, the DM application registration document w7 [ACw7DM] is released by DM group to provide information how the APPLICATION characteristic in OMA Provisioning content [PROVCONT] is used to provision OMA Device Management enabler [ERELDDM] parameters.

5.3.1 Transports

Bootstrapping using OMA Client Provisioning profile is done as defined in the OMA Client Provisioning Bootstrap specification [PROVBOOT].

5.3.2 Mapping Characteristic Data to the Management Tree

When Device receives Client Provisioning document the DM Client creates a management object for each application characteristic in the DM management tree. Management object can have two different types of name space identifiers (Property Name described in [DMTND]) - One where the name is already given in the DDF [DMTND] and another where the name is dynamic separating the instances of the child nodes (See Figure 4: Example Management Object and Name Identifiers).

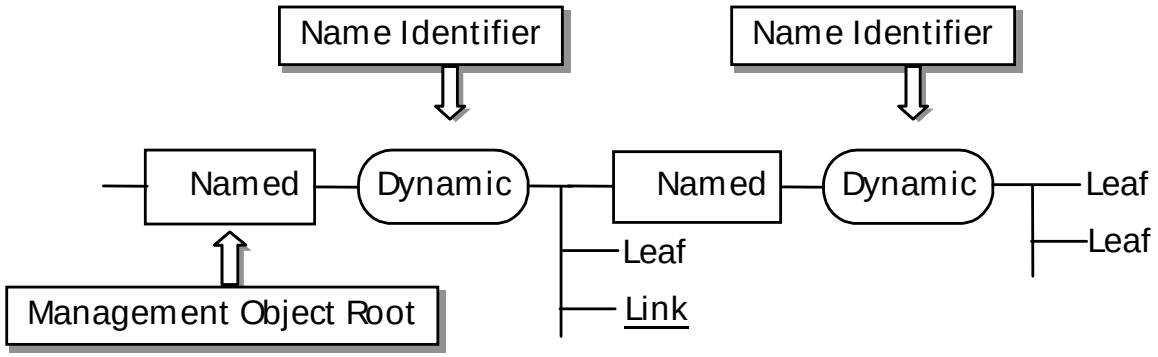

Figure 4: Example Management Object and Name Identifiers

The name identifiers for named nodes are already given in the management object DDF. Also, the parameter mapping between Client Provisioning parameters and Management Object parameters MAY be specified in the Management Object

specification. In addition a general rule that SHOULD be followed to map named information between Client Provisioning APPLICATION characteristic and standardized Connectivity Management Object template structure is given in Appendix C.

The DM Client gives the name identifiers for dynamic nodes that are separating the instances of the child nodes. Though the format of name identifiers for these dynamic nodes is implementation specific, a client MAY assign numeric identifiers starting from '1' and increasing by one every time. In this case and when there is priority specified in the Provisioning Content document the rank SHOULD reflect that.

5.3.3 Management Object Location in the Management Tree

Newly created management object location in the management tree is decided by the DM Client. However, it MUST be placed following the published DDF of the management tree so that the server is able to know where to find the provisioned information.

5.3.4 Management Object Access Rights

All provisioning information mapped from the Provisioning Content [PROVCONT] document to management tree MUST be granted Get, Replace and Delete ACL rights to the ServerID specified in the w7 APPLICATION characteristic provisioned inside Provisioning Content message. The management authority owning the ServerID may modify this ACL in a subsequent DM session.

In case w7 APPLICATION characteristic is not part of the provisioning message the Device receiving the message and mapping the information to the DM management tree MUST NOT give the access rights to these parameters to the improper management authority.

5.3.5 Special Behaviors

5.3.5.1 Smart Card Provisioning

In case Smart Card contains the provisioning information as specified in the [PROVSC], the Device SHOULD detect the removal and/or change of the Smart Card. When the Smart Card is removed and/or changed, the DM Client SHOULD remove all the provisioned management object information (originated from the Smartcard) from the DM management tree.

5.3.6 Device Management, Access Point and Proxy Information

Devices supporting both Client Provisioning and Device Management MUST be able to map w7 (Device Management account) and NAPDEF (if supported) and PROXY (if supported) characteristics information to the DM management tree. The mapping of the named nodes is specified in [DMSTDOBJ]. An explicit mapping of w7 to DMAcc is provided in Appendix C of [DMSTDOBJ], a general mapping of application characteristics can be found in Appendix C of this document.

The DM Client MUST give the names for the dynamic nodes as described in Section 5.3.2.

5.3.7 Other Client Provisioning information

Devices supporting both Client Provisioning and Device Management MAY decide to map other information provisioned in the Client Provisioning message to the DM management tree. In case a specific mechanism is described in the Management Object document that mapping MUST be followed.

5.4 OMA Device Management Profile

The content of the Bootstrap Message is a standard OMA DM message. DM Clients MUST support embedded WBXML encoded TNDS objects and normal TNDS objects [DMTNDS] and MUST support the Inbox. In order to be bootstrapped successfully the DM client requires both DM account information and connectivity information. It is RECOMMENDED to use standardized connectivity MOs to represent the connectivity information.

5.4.1 Transport

Any transport MAY be used to send the Bootstrap Message to the DM Client. Security appropriate for the transport used MUST be employed. If there is no appropriate transport security, transport neutral security MUST be employed. See the security document for further information [DMSecurity].

5.4.2 Management tree ACL and bootstrap

The policy that the Device consults to decide if a Bootstrap Message will be accepted is outside the scope of this specification. If a Bootstrap Message is accepted it MUST be processed according to the conditions described in section 5.4.5.

5.4.3 Management Object Access Rights

When a Bootstrap Message adds new TNDS objects - any ACL values that are to be set for these objects MUST be included in the TNDS data as ACL property data for the applicable nodes.

5.4.4 Bootstrap Message Content

The content of a Bootstrap Message is a normal DM Message. However, it is a special package in many ways since it is not part of an ongoing OMA DM session but rather a one-time message. Hence, many of the elements needed to manage the session are superfluous in this context, but they must still be included so that the message may be processed by the normal OMA DM client.

A Bootstrap Message MUST set the values for the DMAcc management object defined in [DMSTDOBJ]. Other values (such as connectivity settings) MAY also be set.

A DM Server MUST put the value of one of the DMAcc's AppAddr/<x>/Addr nodes in SyncHDR/Source/LocURI.

OMA DM bootstrap message MUST be [WBXML1.1], or [WBXML1.2], or [WBXML1.3] encoded.

A DM Server SHOULD use the XML-encryption [XMLENC] mechanism on parts of a Bootstrap Message when the message contains confidential information. In that case, only the data that is confidential SHOULD be encrypted.

DM Servers MUST NOT expect any response message for a Bootstrap Message. An implicit acknowledgement of successful processing of a Bootstrap Message can be concluded when the client connects to the server for the first Management Session.

A DM Server SHOULD use the XML-signature [XMLSIGN] mechanism on a Bootstrap Message. A DM Server SHOULD use the XML-encryption [XMLENC] mechanism on a Bootstrap Message when the message contains confidential information.

See section 5.8.2.2 of the DM Security document [DMSecurity] for more information on XML-signature and XML-encryption applicability to a Bootstrap Message.

5.4.5 Processing of the Bootstrap

A Bootstrap Message is processed just like a normal DM Message, except that a response message MUST NOT be sent back.

The DM Client MAY rename a new MO. In the case of the Connectivity MO the DM Client SHOULD also rename the values of the corresponding connectivity references to the new name for all MO's encoded within the same TNDS object.

When a TNDS object contains a MO where connectivity references are linked to a Connectivity or Proxy MO that also are included in the same TNDS object, then the values of those connectivity references MAY contain a URI that starts with "./Inbox". In that case the URI MUST have the value of "./Inbox/" plus the URI of that Connectivity MO's location in the same TNDS object.

This is an example of a TNDS object where only part of the TNDS object is shown:

```
<MgmtTree>
```

```
•    <VerDTD>1.2</VerDTD>
•    <Node>
•        <NodeName>OperatorX</NodeName> <!--- DM Account MO --->
•        <RTProperties>
•            <Format>
•                <node/>
•            </Format>
•            <Type><DDFName>org.openmobilealliance/1.0/w7</DDFName></Type>
•        </RTProperties>
•        <Node>
•            <NodeName>PrefConRef</NodeName>
             <RTProperties>
                 <Format>
                     <chr/>
                 </Format>
                 <Type><MIME>text/plain</MIME></Type>
             </RTProperties>
             <Value>./Inbox/Internet</Value>
         </Node>

         …
•        <NodeName>Internet</NodeName>  <!--- Connectivity MO --->
•        <RTProperties>
•            <Format>
•                <node/>
•            </Format>
•            <Type><DDFName>org.openmobilealliance/1.0/ConnMO</DDFName></Type>
•        </RTProperties>

         …

•             </Node>
•</MgmtTree>I
```

If a DM Client encounters an item with a URI of the EXT sub-tree that it is not prepared to handle, the DM Client MAY ignore that item so that the message may succeed.

After successfully processing the Bootstrap Message, the OMA DM Client SHOULD automatically initiate a Management Session to any DM Server configured in the Bootstrap Message at the next practical opportunity, subject to restrictions and configuration in the DMAcc of each bootstrapped server (i.e., when network connectivity and other factors would allow such a connection).

If the Bootstrap Message contains a MO that the DM Client does not support, the DM Client MAY ignore this MO, so that the message may succeed.

If the Bootstrap Message contains multiple versions of a MO, the DM Client SHOULD use the latest version of that MO that it supports and ignore the other versions, so that the message may succeed.

5.4.6 Smartcard

If the Device supports a smartcard, the DM Client MUST support detection, retrieval, and processing of Bootstrap Message from the smartcard as described in Appendix D. The DM Client MAY include configurable security policy to disable smartcard bootstrap functions. If the smartcard bootstrap function is enabled (i.e. no security policy is implemented or security policy does not disable smartcard bootstrap) and the smartcard has not been rejected by the device (for example, because of a SIM-locking mechanism), the DM Client SHALL retrieve the Bootstrap Message from the smartcard when the device is switched on and apply it to the device configuration.

The DM Client SHOULD check that the bootstrap data for all DM Servers previously bootstrapped from the smartcard are still available from the smartcard when the device is switched on; if not, the information for any DM Servers that were

previously bootstrapped from the smartcard but are no longer stored on the smartcard SHOULD be removed from the Device Management tree.

5.4.7 Bootstrap via HTTPS Get

If the Device supports the HTTPS protocol, the DM Client MAY retrieve a Bootstrap Message from a URL by following these steps:

1. The DM Client performs a HTTPS Get to a DM Server using a URL.
2. The DM Server returns the Bootstrap Message to the DM Client or indicates the Bootstrap Message is not available (e.g. returns error code 404).
3. If the Bootstrap Message is returned to the DM Client, upon successful verification of the Bootstrap Message, the DM Client processes the Bootstrap Message as normal.

NOTE: Further work on URL needs to be added.

If the Device supports HTTPS protocol and [SCWS], the DM Client MAY retrieve a bootstrap message by following the above steps and using the following absolute URL: “https://{SCWS@}/OMA/DM/Bootstrap.xml”, where {SCWS@} depends on the transport and IP version supported as shown in the following table:

Transport	IP version	{SCWS@}
BIP (Note 1)	IPv4	127.0.0.1:4116
	IPv6	0.0.0.0.0.0.0.1:4116
TCP/IP	IPv4	localuicc:443
	IPv6	localuicc:443

(Note 1) The DM Client MAY use “localhost” host name instead of loopback address “127.0.0.1” for IPv4 or “0.0.0.0.0.0.0.1” for IPv6.

Appendix A. (Informative)

A.1 Approved Version History

Reference	Date	Description
N/A	N/A	No prior 1.3 version.

A.2 Draft/Candidate Version 1.3 History

Document Identifier	Date	Sections	Description
Draft Versions OMA-TS-DM_Bootstrap-V1_3	15 Oct 2008	All	Baseline to v1.3 using OMA-TS-DM_Bootstrap-V1_2.
	21 May 2009	2.1, 5.4	Applied OMA-DM-DM13-2009-0023-CR_Bootstrap_Security.
	01 Jun 209	5.4.7	Applied OMA-DM-DM13-CR_URL_Bootstrap OMA-DM-2008-0158R01-CR_Mandatory_Bootstrap.
	06 Jul 2009	2.1, 5.4.4	Applied OMA-DM-DM13-2009-0030-CR_Remove_Bootstrap_XML_Security.
	08 Sep 2009	All	Applied OMA-DM-DM13-2009-0064R02-CR_SC_bootstrap_patch.
	17 Nov 2009	All	Applied OMA-DM-DM13-2009-0077R03-CR_Bootstrap_Cleanup OMA-DM-DM13-2009-0084R02-CR_Bootstrap_is_Mandatory
	10 Dec 2009	5.4, B	Applied OMA-DM-DM13-2009-0116R02-CR_Bootstrap_WBXML
	29 Dec 2009	All	Corrected grammar, seplling and other clerical errors.
	11 Jan 2009	All	More spelling errors corrected, applied OMA-DM-DM13-2009-0132-CR_Bootstrap_Figure_Clarification.
	04 Feb 2010	All	Applied OMA-DM-DM13-2010-0008R01-CR_Bootstrap_bugfix OMA-DM-DM13-2010-0029R01-CR_Boot_missing_SCR OMA-DM-DM13-2010-0031R01-CR_Boot_clarification OMA-DM-DM13-2010-0028R01-CR_Bootstrap_Fixes_Again.
	10 Feb 2010	All	General editorial clean-up by DSO.
	23 Mar 2010	5.4.7	Changed blue text to black.
	14 Apr 2010	5.1.1	Editorial change to remove “SIP Push”.
	26 Apr 2010	All	Creation by DSO of a clean version without change marks.
	05 May 2010	5.4.5, 5.4.7	Formatting of an example Formatting of a note.
CandidateVersion OMA-TS-DM_Bootstrap-V1_3	25 May 2010	N/A	Status changed to Candidate by TP Ref # OMA-TP-2010-0221- INP_DM_V1.3_ERP_and_ETR_for_Candidate_approval

Appendix B. Static Conformance Requirements (Normative)

The notation used in this appendix is specified in [IOPPROC].

B.1 SCR for Bootstrap Client

Item	Function	Reference	Status	Requirement
DM-BOOT-C-001	Support for at least one bootstrap mechanism.	Section 5.1.2	M	
DM-BOOT-C-002	Support for OMA Client Provisioning Profile	Section 5.3	O	ProvBoot:ProvBoot-B-C-001 AND DM-BOOT-C-004
DM-BOOT-C-003	Support for OMA Device Management Profile	Section 5.4	M	
DM-BOOT-C-004	Provisioning Content granted Get, Replace and Delete ACL rights to ServerID in w7.	Section 5.3.4	O	
DM-BOOT-C-005	Support for OMA Client Provisioning Profile AND OMA Device Management	Section 5.3.6	O	DM-BOOT-C-006
DM-BOOT-C-006	Map w7, NAPDEF (if supported) and PROXY (if supported) to management tree.	Section 5.3.6	O	
DM-BOOT-C-007	Device supports a Smartcard.	Section 5.4.6	O	DM-BOOT-C-007
DM-BOOT-C-008	DM Client is capable of detecting, retrieving, and processing DM Profile bootstrap data from the Smartcard.	Section 5.4.6	O	
DM-BOOT-C-009	Smartcard bootstrap function is enabled by DM client and the smartcard has not been rejected by the device.	Section 5.4.6	O	DM-BOOT-C-010 AND DM-BOOT-C-013
DM-BOOT-C-010	Device retrieves bootstrap data from the Smartcard and applies it to the device configuration.	Section 5.4.6	O	
DM-BOOT-C-011	Support for embedded WBXML encoded TNDS objects and normal TNDS objects.	Section 5.4	M	
DM-BOOT-C-012	Support for Inbox.	Section 5.4	M	
DM-BOOT-C-013	DM Client removes the DM Server's information from the Device Management Tree if they are no longer stored on the smartcard when the	Section 5.4.6	O	

Item	Function	Reference	Status	Requirement
	device is switched on.			
DM-BOOT-C-014	Device supports SCWS and HTTPS	Section 5.4.7	O	DM-BOOT-C-014
DM-BOOT-C-015	Device retrieves Bootstrap Message from Smartcard URL using HTTPS Get	Section 5.4.7	O	
DM-BOOT-C-016	Support for WBXML encoded DM Bootstrap Messages.	Section 5.4	M	

B.2 SCR for Bootstrap Server

Item	Function	Reference	Status	Requirement
DM-BOOT-S-001	Support for OMA Client Provisioning Profile	Section 5.3	O	
DM-BOOT-S-002	Support for OMA Device Management Profile	Section 5.4	M	
DM-BOOT-S-003	Encode DM Bootstrap Message into WBXML.	Section 5.4	M	
DM-BOOT-S-004	Support for embedded WBXML encoded TNDS objects and normal (non-WBXML-encoded) TNDS objects.	Section 5.4	M	

Appendix C. General Mapping

In the below table the Provisioning Content APPLICATION characteristic correspondence is shown to the Structure Template for Application Connectivity Management Object.

APPLICATION CHARACTERISTIC INFORMATION	STRUCTURE TEMPLATE FOR APPLICATION CONNECTIVITY MANAGEMENT OBJECT
APPID	AppID
PROVIDER-ID	ProviderID
NAME	Name
AACCEPT	AAccept
APROTOCOL	AProtocol
TO-PROXY	PrefConDef, if multiple ToConRef/<X>/ConRef
TO-NAPID	PrefConDef, if multiple ToConRef/<X>/ConRef
ADDR	PrefAddr
APPADDR/ADDR	AppAddr/<X>/Addr
APPADDR/ADDRTYPE	AppAddr/<X>/AddrType
APPADDR/PORT/PORTNBR	AppAddr/<X>/Port/<X>/PortNbr
APPADDR/PORT/SERVICE	AppAddr/<X>/Port/<X>/Service/<X>/Service
APPAUTH/AAUTHLEVEL	AppAuth/<X>/AAuthLevel
APPAUTH/AAUTHTYPE	AppAuth/<X>/AAuthType
APPAUTH/AAUTHNAME	AppAuth/<X>/AAuthName
APPAUTH/AAUTHSECRET	AppAuth/<X>/AAuthSecret
APPAUTH/AAUTHDATA	AppAuth/<X>/AAuthData
RESOURCE/URI	Resource/<X>/URI
RESOURCE/NAME	Resource/<X>/Name
RESOURCE/AACCEPT	Resource/<X>/AAccept
RESOURCE/AAUTHTYPE	Resource/<X>/AAuthType
RESOURCE/AAUTHNAME	Resource/<X>/AAuthName
RESOURCE/AAUTHSECRET	Resource/<X>/AAuthSecret
RESOURCE/AAUTHDATA	Resource/<X>/AAuthData
RESOURCE/STARTPAGE	N/A

Table 1: General Mapping

Appendix D. Storage of DM Bootstrap Message on the Smartcard (Normative)

We can sort out three main types of smartcards used for wireless telecom networks, characterised by their physical and logical characteristics:

- SIM smartcards platforms [TS151.011]
- UICC smartcards platforms [TS102.221]
- R-UIM smartcards platform [C.S0023-B_v1.0]

This section aims at specifying the storage mechanism of Bootstrap Message on such smartcard platform type.

For the purposes of this document the R-UIM is to be treated according to the rules defined for the SIM.

D.1 File structure

The information format is based on [PKCS#15] specification. The Bootstrap Message is located under the PKCS#15 directory allowing the card issuer to decide the identifiers and the file locations. The smartcard operations that are relevant include:

- Application selection
- Cardholder verification
- File access (select file, read, write)

The [PKCS#15] specification defines a set of files. Within the PKCS#15 application, the starting point to access these files is the Object Directory File (ODF). The EF(ODF) contains pointers to other directory files. These directory files contain information on different types of objects (authentication objects (PIN), data objects, etc). For the purpose of Bootstrap Message, EF (ODF) MUST contain the record describing the DODF-bootstrap. The EF(ODF) is described in section D.4.1 and [PKCS#15].

EF(ODF) contains pointers to one or more Data Object Directory Files (DODF) in priority order (i.e. the first DODF has the highest priority). Each DODF is regarded as the directory of data objects known to the PKCS#15 application. For the purposes of DM bootstrapping, EF(DODF-bootstrap) contains pointer to the Bootstrap Message, namely DM_Bootstrap File. The EF(DODF-bootstrap) is described in section D.4.2 and [PKCS#15].

The provisioning files are stored as PKCS#15 opaque data objects.

The support of smartcard Bootstrap Message will be indicated to the ME's user agent, by the presence in the EF DIR (see [TS102.221]) of an application template as defined here after.

The RECOMMENDED format of EF(DIR) is a linear fixed record in order to be in line with [TS102.221].

EF (DIR) MUST contain the application template used for a PKCS#15 application as defined in [PKCS15]. Application template MUST consist of Application identifier (tag 0x4F) and Path (tag 0x51) information.

The EF(ODF) and EF(DODF-bootstrap) MUST be used by the ME to determine the path of the DM_Bootstrap file.

UICC smartcard platforms can support two modes of activation: 2G and 3G. UICC smartcard platform activated in a 2G mode has the logical characteristics of the SIM smartcard platform [TS151.011]. In that case, smartcard operations for accessing the Bootstrap Message conform to the ones defined for the SIM as specified in section D.2.

UICC smartcard platform activated in a 3G mode has the physical and logical characteristics according to [TS102.221]. In that case, smartcard operations for accessing the Bootstrap Message are specified in section D.3.

D.2 Bootstrap Message on SIM or UICC activated in 2G mode

D.2.1 Access to the file structure

To select the PKCS15 application, the Device MUST evaluate the PKCS#15 application template present in the EF (DIR), then the Device MUST use the indirect selection method as defined in [TS151.011] to select the application.

D.2.2 Files Overview

The file structure for the Bootstrap Message within the SIM smartcard is described below.

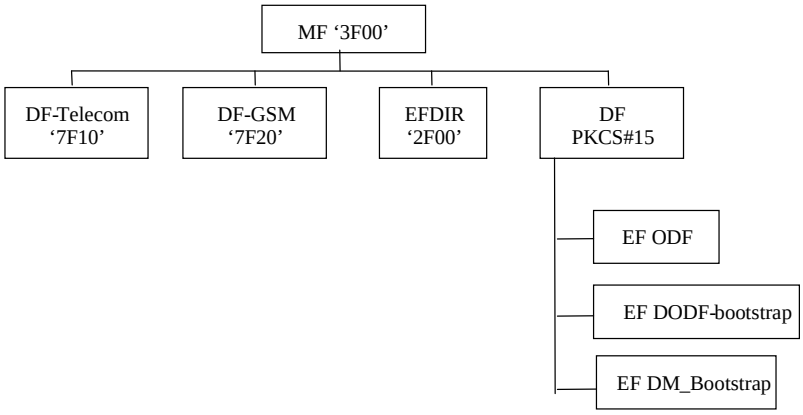

Figure 5: File structure for Bootstrap Message on SIM smartcard or 2G UICC

D.2.3 Access Method

SIM commands Read Binary and Update Binary, as defined in [TS151.011], are used to access the Bootstrap Message.

D.2.4 Access Conditions

The Device is informed of the access conditions of the Bootstrap Message by evaluating the "private" and "modifiable" flags in the corresponding DODF-bootstrap files structure. When one of these flags is set cardholder verification is required. The CHV1 MUST be verified as defined in [TS151.011] when the "private" or "modifiable" flags are set.

Access conditions for files are proposed in the section D.4.

D.2.5 Requirements on the SIM or 2G UICC

To retrieve the Bootstrap Message from the SIM or 2G UICC, the Device MUST perform the following steps:

- Read EF (DIR) to evaluate the PKCS#15 application template and find the file identifier (and path of the PKCS#15 DF),
- Select PKCS#15 DF (indirect selection), as defined in [TS151.011],
- Read ODF,
- Read DODF-bootstrap to locate the DM_Bootstrap file,
- Read the DM_Bootstrap file.

D.3 Bootstrap Message on UICC Activated in 3G Mode

D.3.1 Access to the file structure

To select the PKCS#15 application, the Device:

- MUST evaluate the PKCS#15 application template – i.e. PKCS#15 AID - present in the EF (DIR),
- MUST open a logical channel using MANAGE CHANNEL command as specified in [TS102.221],
- MUST select the PKCS#15 ADF using the PKCS#15 AID as parameter of the SELECT command, using direct application selection as defined in [TS102.221].

DM_Bootstrap file will be located under the PKCS#15 ADF.

D.3.2 Files Overview

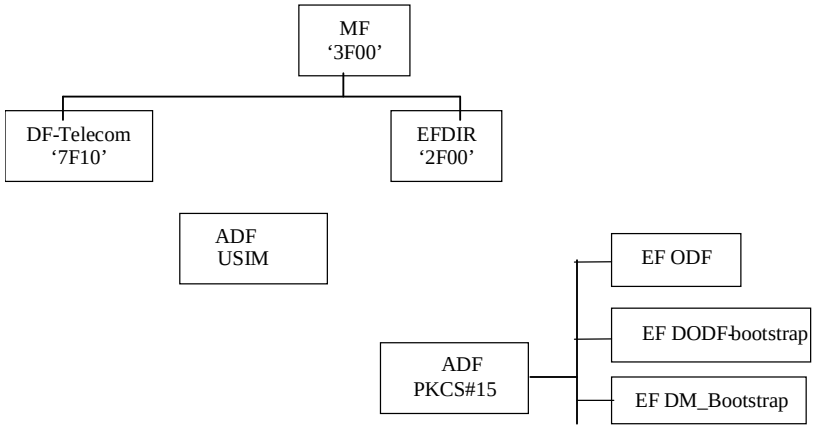

Figure 6: File structure for Bootstrap Message on 3G UICC

D.3.3 Access Method

UICC commands Read Binary and Update Binary, as defined in [TS102 221], are used to access bootstrap data.

D.3.4 Access Conditions

The Device is informed of the access conditions of provisioning files by evaluating the "private" and "modifiable" flags in the corresponding DODF-bootstrap files structure.

In the case where one of the above mentioned flag is set, cardholder verification is required. The Device must evaluate the PIN references that must be verified as defined in [TS102.221] i.e. evaluate the FCP.

Access conditions for files are proposed in section D.4.

D.3.5 Requirements on the 3G UICC

To retrieve the Bootstrap Message from the 3G UICC, the Device MUST perform the following steps:

- Select PKCS#15 file structure as specified in D.3.1.
- Read ODF to locate the DODF-bootstrap,
- Read DODF-bootstrap to locate the DM_Bootstrap file,
- Read the DM_Bootstrap file

D.4 Files Description

All files defined are binary files as defined in [TS102.221]. These files are read and updated using commands related to the application they belong to.

D.4.1 Object Directory File, EF ODF

The mandatory Object Directory File (ODF) ([PKCS#15], section 5.5.1) contains pointers to other EFs, each one containing a directory of PKCS#15 objects of a particular class (e.g. DODF-bootstrap). The File ID is specified in [PKCS#15]. The card issuer decides the file size. The EF (ODF) can be read but it MUST NOT be modifiable by the user.

In the case of SIM or UICC, the EF (ODF) is described below:

Identifier: default 0x5031, see [PKCS#15]	Structure: Binary	Mandatory
File size: decided by the card issuer	Update activity: low	
Access Conditions: READ ALW UPDATE ADM INVALIDATE ADM REHABILITATE ADM		
Description		
See [PKCS#15]		

D.4.2 Bootstrap Data Object Directory File, EF DODF-bootstrap

This Data Object Directory File provisioning contains directories of provisioning data objects ([PKCS#15], section 6.7) known to the PKCS#15 application.

The File ID is described in the EF (ODF). The file size depends on the number of provisioning objects stored in the smartcard. Thus, the card issuer decides the file size.

Identifier: 0x6420, See ODF	Structure: Binary	Mandatory
File size: decided by the card issuer	Update activity: low	

Access Conditions: READ ALW or CHV1 (SIM, See section D.2) or Universal / application / Local PIN (UICC, See section D.3) UPDATE ADM INVALIDATE ADM REHABILITATE ADM
Description
See hereafter and [PKCS#15]

The EF (DODF-bootstrap) MUST contain information on provisioning objects:

- Readable label describing the provisioning document (`CommonObjectAttributes.label`). The ME could display this label to the user.
- Flags indicating whether the provisioning document is private (i.e., is protected with a PIN) and/or modifiable (`CommonObjectAttributes.flags`). The card issuer decides whether or not a file is private (it does not need to be if it does not contain any sensitive information)
- Object identifier indicating a DMboostrap object and the type of the provisioning object (`CommonDataObjectAttributes.applicationOID`)
- Pointer to the contents of the provisioning document (Path.path)

The EF(DODF-bootstrap) MUST contain the types of provisioning documents (indicated using object identifiers) to be used by the ME. The Bootstrap type is described hereafter.

A dedicated OID is required and defined for each provisioning file:

- `Bootstrap OID = { joint-isu-itu-t(2) international-organizations(23) wap(43) oma-dm(7) dm-bootstrap(1)}`

The ME MUST use the OID to distinguish the DODF-bootstrap from any other DODF. The EF(DODF-bootstrap) can be read but it MUST NOT be modifiable by the user.

D.4.3 EF DM_Bootstrap

Only the card issuer can modify EF DM_Bootstrap

Setting all bytes to 'FF' initialises EF DM_Bootstrap.

Identifier: See DODF	Structure: Binary	Optional
File size: decided by the card issuer	Update activity: low	
Access Conditions: READ ALW or CHV1 (SIM, See section D.2) or Universal / application / Local PIN (UICC, See section D.3) UPDATE ADM INVALIDATE ADM REHABILITATE ADM		
Description		
Contains a Bootstrap Message		