PacketCable™ 2.0

UE Provisioning Framework Specification

PKT-SP-UE-PROV-C01-140314

CLOSED

Notice

DISCLAIMER

This document is furnished on an "AS IS" basis and neither CableLabs nor its members provides any representation or warranty, express or implied, regarding the accuracy, completeness, noninfringement, or fitness for a particular purpose of this document, or any document referenced herein. Any use or reliance on the information or opinion in this document is at the risk of the user, and CableLabs and its members shall not be liable for any damage or injury incurred by any person arising out of the completeness, accuracy, or utility of any information or opinion contained in the document.

CableLabs reserves the right to revise this document for any reason including, but not limited to, changes in laws, regulations, or standards promulgated by various entities, technology advances, or changes in equipment design, manufacturing techniques, or operating procedures described, or referred to, herein.

This document is not to be construed to suggest that any affiliated company modify or change any of its products or procedures, nor does this document represent a commitment by CableLabs or any of its members to purchase any product whether or not it meets the characteristics described in the document. Unless granted in a separate written agreement from CableLabs, nothing contained herein shall be construed to confer any license or right to any intellectual property. This document is not to be construed as an endorsement of any product or company or as the adoption or promulgation of any guidelines, standards, or recommendations.

Document Status Sheet

Document Control Number:	PKT-SP-UE-PROV-C01-140314			
Document Title:	UE Provisioning Framework Specification			
Revision History:	I01 - Released September 5, 2008 I02 - Released May 27, 2010 C01 - Released March 14, 2014			
Date:	March 14, 2014			
Status:	~~Work in Progress~~	~~Draft~~	~~Issued~~	**Closed**
Distribution Restrictions:	~~Author Only~~	~~CL/Member~~	~~CL/ Member/ Vendor~~	**Public**

Key to Document Status Codes

Work in Progress	An incomplete document, designed to guide discussion and generate feedback that may include several alternative requirements for consideration.
Draft	A document in specification format considered largely complete, but lacking review by Members and vendors. Drafts are susceptible to substantial change during the review process.
Issued	A stable document, which has undergone rigorous member and vendor review and is suitable for product design and development, cross-vendor interoperability, and for certification testing.
Closed	A static document, reviewed, tested, validated, and closed to further engineering change requests to the specification through CableLabs.

Trademarks

Contents

Figures

Tables

1 SCOPE

1.1 Introduction and Purpose

This specification describes the general provisioning framework for PacketCable 2.0 User Equipment (UE). The framework defines the network and protocol requirements to configure and manage UEs, along with associated users and applications. This includes initial and incremental provisioning, configuration, management and event notification. The data model requirements of this framework are specified in a related document, the UE Provisioning Data Model Specification [PKT-UE-DATA]. For the provisioning and management of Embedded User Equipment (E-UE), please refer to the E-UE Provisioning Framework [PKT-EUE-PROV].

This specification also defines a provisioning framework for the PacketCable Core Network, consisting of a provisioning interface for the HSS (Home Service Subscriber) and IMS Application Server (AS).

1.2 Document Overview

This document is structured as follows:

- Section 2 - References
- Section 3 - Terms and Definitions
- Section 4 - Abbreviations and Acronyms
- Section 5 - Technical Overview: this is an informative section providing a description of the provisioning reference architecture, components, and interfaces.
- Section 6 - UE and IMS Components Provisioning Framework Requirements: this is a normative section detailing the framework requirements.
- Annex A - UE Management Events
- Annex B - Software Upgrade Trigger Message
- Appendix I - Sample UE Provisioning Flow
- Appendix II - Mapping of Event Severity For PacketCable And OMA DM

1.3 Requirements

Throughout this document, the words that are used to define the significance of particular requirements are capitalized. These words are:

"MUST"	This word means that the item is an absolute requirement of this specification.
"MUST NOT"	This phrase means that the item is an absolute prohibition of this specification.
"SHOULD"	This word means that there may exist valid reasons in particular circumstances to ignore this item, but the full implications should be understood and the case carefully weighed before choosing a different course.

"SHOULD NOT"	This phrase means that there may exist valid reasons in particular circumstances when the listed behavior is acceptable or even useful, but the full implications should be understood and the case carefully weighed before implementing any behavior described with this label.
"MAY"	This word means that this item is truly optional. One vendor may choose to include the item because a particular marketplace requires it or because it enhances the product, for example; another vendor may omit the same item.

2 REFERENCES

2.1 Normative References

In order to claim compliance with this specification, it is necessary to conform to the following standards and other works as indicated, in addition to the other requirements of this specification. Notwithstanding, intellectual property rights may be required to use or implement such normative references.

[CLAB-SM]	CableLabs Software Management Framework Specification, CL-SP-SM-I01-080708, July 8, 2008, Cable Television Laboratories, Inc.
[CLAB-WSRP]	CableLabs Web Services Recommended Practices, CL-SP-WSRP-I01-091023, October 23, 2009, Cable Television Laboratories, Inc.
[PKT-MEM1.5]	PacketCable 1.5 Management Event Mechanism Specification, PKT-SP-MEM1.5-I05-100527, May 27, 2010, Cable Television Laboratories, Inc.
[PKT-EUE-DATA]	PacketCable E-UE Provisioning Data Model Specification, PKT-SP-EUE-DATA-C01-140314, March 14, 2014, Cable Television Laboratories, Inc.
[PKT-EUE-PROV]	PacketCable E-UE Provisioning Framework Specification, PKT-SP-EUE-PROV-C01-140314, March 14, 2014, Cable Television Laboratories, Inc.
[PKT-UE-DATA]	PacketCable UE Provisioning Data Model Specification, PKT-SP-UE-DATA-C01-140314, March 14, 2014, Cable Television Laboratories, Inc.
[PKT-RST-UE-PROV]	PacketCable RST UE Provisioning Specification, PKT-SP-RST-UE-PROV-C01-140314, March 14, 2014, Cable Television Laboratories, Inc.
[OMA-CONNMO]	Standardized Connectivity Management Objects For use with OMA Device Management, OMS-DDS-DM_ConnMO-V1_0-20080415-D, April 15, 2008, Open Mobile Alliance (draft version).
[OMA-DCMO]	Device Capability Management Object, Version 1.0, OMA-TS-DCMO-V1_0-20080428-D, April 8, 2008, Open Mobile Alliance.
[OMA-DIAGMONTRAPMO]	DiagMon Trap Management Object, Version 1.0, OMA-TS-DiagMonTrapMO-V1_0-20070704-D, July 4, 2007, Open Mobile Alliance.
[OMA-DM]	Enabler Release Definition for OMA Device Management Draft Version 1.3, OMA-ERELD-DM-V1_3-20100211-D, February 11, 2010, Open Mobile Alliance.
[OMA-DMBOOT]	OMA Device Management Bootstrap, OMA-TS-DM_Bootstrap-V1_2-20070209-A, February 9, 2007, Open Mobile Alliance.
[OMA-DMFUMO]	Firmware Update Management Object, OMA-TS-DM-FUMO-V1_0-20070209-A, Version 1.0, February 9, 2007, Open Mobile Alliance.
[OMA-DMNOT]	OMA Device Management Notification Initiated Session, OMA-TS-DM_Notification-V1_2-20070209-A, February 9, 2007, Open Mobile Alliance.
[OMA-DMPRO]	OMA Device Management Protocol, OMA-TS-DM_Protocol-V1_2-20070209-A, February 9, 2007, Open Mobile Alliance.
[OMA-DMREPPRO]	OMA Device Management Representation Protocol, OMA-TS-DM_RepPro-V1_2-20070209-A, February 9, 2007, Open Mobile Alliance.
[OMA-DMSEC]	OMA Device Management Security, OMA-TS-DM_Security-V1_2-20070209-A, February 9, 2007, Open Mobile Alliance.
[OMA-DMSTDOBJ]	OMA Device Management Standardized Objects, OMA-TS-DM_StdObj-V1_2-20070209-A, February 9, 2007, Open Mobile Alliance.

[OMA-DMTND]	OMA Device Management Tree and Description, OMA-TS-DM_TND-V1_2-20070209-A, February 9, 2007, Open Mobile Alliance.
[OMA-DMTNDS]	OMA Device Management Tree and Description Serialization, OMA-TS-DM_TNDS-V1_2-20070209-A, February 9, 2007, Open Mobile Alliance.
[OMA-DPE]	Device Profile Evolution Technical Specification, Version 1.0, OMA-TS-DPE-V1_0-20080611-D, June 11, 2008, Open Mobile Alliance.
[OMA-OTAPUSH]	Push Over The Air, OMA-TS-PushOTA-V2_2-20080228-D, February 28, 2008, Open Mobile Alliance (draft version).
[OMA-PUSHAD]	Push Architecture, OMA-AD-Push-V2_2-20071002-C, October 2, 2007, Open Mobile Alliance (candidate version).
[OMA-PUSHSIPAD]	Push Using SIP Architecture, OMA-AD-SIP_Push-V1_0-20080505-D, May 5, 2008, Open Mobile Alliance (draft version).
[OMA-SIPPUSH]	Push using SIP, OMA-TS-SIP_Push-V1_0-20080416-D, April 16, 2008, Open Mobile Alliance (draft version).
[OMA-SYNCHTTP]	SyncML HTTP Binding, OMA-TS-SyncML_HTTPBinding-V1_2-20070221-A, February 21, 2007, Open Mobile Alliance.
[3GPP 24.167]	3GPP IMS Management Object, Stage 3, V7.4.0, March, 2008, 3rd Generation Partnership Project.
[PKT 24.229]	PacketCable SIP and SDP Stage 3 Specification 3GPP TS 24.229, PKT-SP-24.229-C01-140314, March 14, 2014, Cable Television Laboratories, Inc.
[PKT-ARCH-TR]	PacketCable Architecture Framework Technical Report, PKT-TR-ARCH-FRM-V06-090528, May 28, 2009, Cable Television Laboratories, Inc.
[3GPP 33.220]	3GPP Generic Authentication Architecture (GAA); Generic bootstrapping architecture (Release 7), V7.11.0, March 2008, 3rd Generation Partnership Project.
[3GPP 33.222]	3GPP Generic Authentication Architecture (GAA); Access to network application functions using Hypertext Transfer Protocol over Transport Layer Security (HTTPS) (Release 7), V7.3.0, December 2007, 3rd Generation Partnership Project.
[RFC 1034]	IETF RFC 1034, Domain Names - Concepts and Facilities, November 1987.
[RFC 1035]	IETF RFC 1035, Domain Names - Implementation and Specification, November 1987.
[RFC 1305]	IETF RFC 1305, Network Time Protocol (Version 3) Specification, Implementation and Analysis, March 1992.
[RFC 2131]	IETF RFC 2131, Dynamic Host Configuration Protocol, March 1997.
[RFC 2132]	IETF RFC 2132, DHCP Options and BOOTP Vendor Extensions, March 1997.
[RFC 2782]	IETF RFC 2782, A DNS RR for Specifying the Location of Services (DNS SRV), February 2000.
[RFC 2915]	IETF RFC 2915, The Naming Authority Pointer (NAPTR) DNS Resource Record, September 2000.
[RFC 3164]	IETF RFC 3164, The BSD Syslog Protocol, August 2001.
[RFC 3263]	IETF RFC 3263, Session Initiation Protocol (SIP): Locating SIP Servers, June 2002.
[RFC 3265]	IETF RFC 3265, Session Initiation Protocol (SIP)-Specific Event Notification, June 2002.
[RFC 3315]	IETF RFC 3315, Dynamic Host Configuration Protocol for IPv6 (DHCPv6), July 2003.
[RFC 3484]	IETF RFC 3484, Default Address Selection for Internet Protocol version 6 (IPv6), February 2003.
[RFC 3596]	IETF RFC 3596, DNS Extensions to Support IP Version 6, October 2003.

[RFC 3646] IETF RFC 3646, DNS Configuration Options for Dynamic Host Configuration Protocol for IPv6 (DHCPv6), December 2003.

[RFC 3680] IETF RFC 3680, A Session Initiation Protocol (SIP) Event Package for Registrations, March 2004.

[RFC 4330] IETF RFC 4330, Simple Network Time Protocol (SNTP) Version 4 for IPv4, IPv6 and OSI, January 2006.

[RFC 4346] IETF RFC 4346, The TLS Protocol Version 1.0, April 2006.

2.2 Informative References

This specification uses the following informative references.

[PKT-PRS] PacketCable Presence Specification, PKT-SP-PRS-C01-140314, March 14, 2014, Cable Television Laboratories, Inc.

[PKT-PROV1.5] PacketCable MTA Device Provisioning Specification, PKT-SP-PROV1.5-I04-090624, June 24, 2009, Cable Television Laboratories, Inc.

[OMA-DMDDF] OMA DM Device Description Framework DTD, Version 1.2, OMA-SUP-dtd_dm_ddf-V1_2-20070209-A, February 9, 2007, Open Mobile Alliance.

[OMA-PUSHPAP] Push Access Protocol, OMA-WAP-TS_PAP-V2_2-20071002-C, October 2, 2007, Open Mobile Alliance (candidate version).

[WBXML1.3] WAP Binary XML Content Format Specification, WAP-192-WBXML-20010725-a, July 25, 2001, WAP Forum.

[W3 XML] Extensible Markup Language (XML) 1.0 (Third Edition), W3C Recommendation 04, February 2004.

2.3 Reference Acquisition

- Cable Television Laboratories, Inc., 858 Coal Creek Circle, Louisville, CO 80027;
  Phone +1-303-661-9100; Fax +1-303-661-9199; http://www.cablelabs.com
- Internet Engineering Task Force (IETF) Secretariat, 48377 Fremont Blvd., Suite 117, Fremont, California 94538, USA, Phone: +1-510-492-4080, Fax: +1-510-492-4001, http://www.ietf.org
- Open Mobile Alliance (OMA), OMA Office, 4275 Executive Square, Suite 240, La Jolla, CA 92037, Fax +1-858-623-0743, Internet: http://www.openmobilealliance.com/
- 3rd Generation Partnership Project (3GPP), ETSI Mobile Competence Centre, 650 route des Lucioles, 06921 Sophia-Antipolis Cedex, France, Internet: http://www.3gpp.org/
- World Wide Web Consortium (W3C), www.w3.org

3 TERMS AND DEFINITIONS

This specification uses the following terms:

Bootstrap	A process by which a node acquires sufficient information to proceed with full configuration and provisioning.
Data Model	An abstract model that describes representation of data in a system.
DM Package #0	A notification message from a DM Server requesting a DM Client to initiate a management session. This out-of-band message is an optional part of any DM session because a DM Client can always initiate an unsolicited session.
IMS Components	The term used in this specification to refer to PacketCable Core Network components and Application Servers.
PacketCable Core Network	The part of the operator network providing SIP services and holding subscriber data, consisting of IMS components I-CSCF, S-CSCF, HSS, E-CSCF, and SLF.
Provisioning	The processes involved in the initialization of user attributes and resources to provide services to a User. This involves protocols, methodologies, and interfaces to network elements such as: Order Entry and Workflow Systems that carry out business processes, Operational Support Elements that handle network resources, Application Servers that offer services, and User Equipment that offer services, among others.
Push	A content delivery mechanism between a Push Initiator and a Push Client, originally developed for mobile devices.
Push Proxy Gateway	An agent that receives Push application content from a Push Initiator via Push Access Protocol, and delivers them to Push Client via Push Over-the-Air Protocol (Push OTA).
RESTful	Conforming to REST constraints.

4 ABBREVIATIONS AND ACRONYMS

This specification uses the following abbreviations:

AS	Application Server
DDF	OMA DM Device Description Framework
DM	OMA Device Management
DMAcc	Device Management Account management object
DTD	Document Type Definition
HTTP	Hypertext Transfer Protocol
IMS	IP Multimedia Subsystem
MO	OMA DM Managed Object definition
OMA	The Open Mobile Alliance
OMA DM	Open Mobile Alliance Device Management Protocol
REST	Representational State Transfer
SIP	Session Initiation Protocol
SOAP	Simple Object Access Protocol
TND	OMA DM Tree and Description
UE	User Equipment
W3C	World Wide Web Consortium
WAP	Wireless Application Protocol
WBXML	WAP Binary XML
WSDL	Web Services Description Language
XML	Extensible Markup Language
XSD	W3C XML Schema Definition Language

5 TECHNICAL OVERVIEW

PacketCable 2.0 is a CableLabs specification effort designed to support the convergence of voice, video, data, and mobility technologies. For more information about PacketCable 2.0, please refer to the PacketCable 2.0 Architecture Framework Technical Report [PKT-ARCH-TR].

This document is part of the PacketCable 2.0 set of specifications and technical reports that define the base architecture, network components, and protocol requirements necessary to meet the needs of various applications and services. This UE Provisioning Framework specification describes the base requirements and protocols for the configuration and management of:

- User Equipment (UE) in a PacketCable environment in support of UE, user, and application needs
- PacketCable functional network components, including core and application functional groups

Other specifications may describe how this framework is applied to the requirements of specific applications and devices.

This section provides a general overview of the UE Provisioning Framework and describes the high-level requirements, components and considerations.

5.1 User Equipment (UE)

PacketCable is based on SIP and IMS, and supports a wide variety of clients with varying characteristics and capabilities, including software and hardware-based, standalone and embedded devices (with other cable devices, e.g., DOCSIS Cable Modems). Consistent with IMS terminology, all PacketCable clients are called User Equipment (UE). For more information about UEs in PacketCable, please refer to the PacketCable Architecture Technical Report [PKT-ARCH-TR].

Each UE may support multiple applications and multiple users, where each user may or may not be a user of each application. A UE supporting a hypothetical application *XX* may be referred to as an *XX* UE, and requirements specific to configuration and management of application *XX* across all UEs will be described an application-specific provisioning specification document. Similarly, there may be specific types of UEs, such as an E-DVA, which have unique provisioning requirements in support of all or certain applications, and which will have specific provisioning documents, or sections of application provisioning documents, as well.

5.2 UE Provisioning Considerations

5.2.1 Objective

The UE Provisioning Framework is intended to specify procedures and protocols to support the provisioning, configuration, and management requirements of embedded and standalone UEs within the PacketCable architecture framework. A general overview of the goals of UE Provisioning is described in [PKT-ARCH-TR]. These include IP network initialization, initial configuration, incremental configuration, data model definition, configuration retrieval, event reporting, diagnostics, and retrieval of discovered information, statistics, and UE status.

The UE Provisioning Framework must support both UE provisioning and layered User provisioning, where the relationship of Users to UEs is many to many, and Users, or their applications, may be associated with different Operators than that of the UE itself.

5.2.2 Design Assumptions and Limitations

The UE Provisioning Framework assumes a native IP environment. This is in contrast to special environments, such as WAP, where non-IP protocols are used. As such, the uses of OMA protocols in this specification require bindings and transports that function in an Internet context in conformance with IETF standards. At the same time, the need for compatibility with common use and implementation of OMA protocols means that deviations from and extensions to the specific protocols used must be avoided wherever possible, and OMA standard data models must be employed where possible and carefully extended where necessary. This should allow interworking with non-IP environments using OMA through back-office interfaces, and possibly through shared back-office systems.

The devices and soft-client UEs addressed by this framework are expected to connect to the Operator from a variety of locations, not limited to the Cable Operator's access network, and potentially behind NAT devices and firewalls that create hurdles for direct communication. These clients may be of a variety of types, from public kiosks to PC software, from set-top-boxes to mobile devices. Because of the variety of clients and applications expected, this framework attempts to remain general and avoids special requirements and behaviors of special devices and environments. Follow-on specifications with more specific target environments, client, or applications may add to or modify the requirements of this specification as needed, within their own constrains of compatibility.

5.2.3 IP Network Environments

This specification differentiates the IP network environments in which a UE offers service into *controlled* and *uncontrolled* environments.

When the local IP network in which a UE operates is under the control of the PacketCable Operator, the UE is considered to operate in a Controlled Environment. An example of a UE operating in a controlled environment is a PacketCable Embedded Digital Voice Adapter (E-DVA) operating in an IP network that is controlled by the Cable Operator.

When the local IP network in which a UE operates is not under the control of the Operator, the UE is said to function in an Uncontrolled Environment. An example of a UE in an uncontrolled environment is a UE operating in a public IP network not controlled by the Operator.

The term 'control' in this section refers to the Operator's ability to control the IP configuration information available in the network, e.g., via DHCP (independent of how the IP address is allocated). UEs operating in controlled environments may use somewhat simplified provisioning solutions, relative to UEs in uncontrolled environments. Examples include learning the Operator domain name, DHCP-assisted P-CSCF discovery, and UE bootstrapping.

This specification supports UEs using IPv4, IPv6, or both.

5.2.4 Provisioning Models

This specification defines two provisioning models, "pre-configured" and "dynamic configuration", enabling the PacketCable Operators to choose the deployment model.

The "pre-configured" provisioning model requires UEs to be pre-configured with some Operator information prior to establishing contact with the PacketCable Operator's network. This will generally include the Operator domain name, a unique device identifier, and some credentials. The "pre-configured" UE may still require further bootstrap configuration to obtain all credentials and to interact with the provisioning framework.

The "dynamic configuration" provisioning model allows UEs to have no pre-configured information to establish contact with a PacketCable Operator's network.

The dynamic configuration provisioning model is not addressed in this version of the UE Provisioning specification.

5.3 UE Provisioning

The UE Provisioning Framework is based on the Open Mobile Alliance Device Management specification [OMA-DM]. Section 5.4 provides an overview of OMA DM. The OMA DM protocol provides a rich set of configuration and management operations against an extensible, tree-oriented management data model. In IP-based environments, such as PacketCable, OMA DM is an HTTP-based protocol. Management sessions are client-initiated using HTTP requests to the DM server, and may be server-initiated through the delivery of a notification message. Server-initiated notification uses OMA Push mechanisms. PacketCable will deliver notifications using SIP-based Push through the PacketCable Core Network, to support NAT traversal in IP environments.

Client event notification is supported through the OMA DM Generic Alert mechanism within an OMA DM session. Secure software download is based on the OMA DM Firmware Update Managed Object specification [OMA-DMFUMO] with PacketCable extensions.

Figure 1 below shows the protocol layering of OMA Device Management and the associated OM DM Notification message for server-initiated sessions.

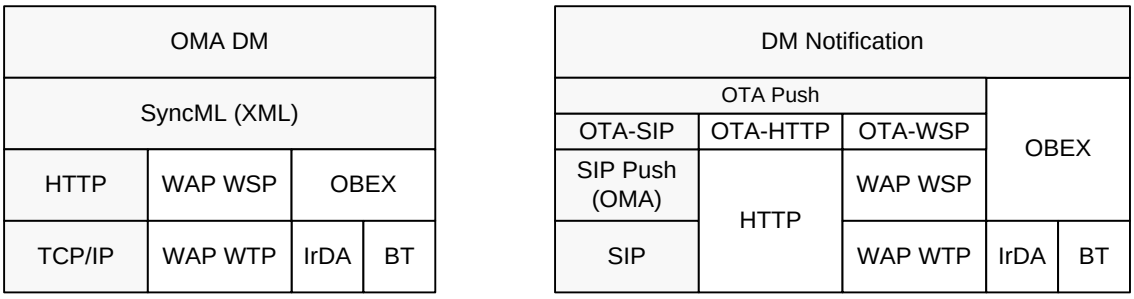

Figure 1 - OMA DM Protocol Layering

The UE Provisioning Framework encompasses the requirements to provision and manage the UE device, the applications and services supported by the UE, and the management of the possible interactions of the different users with the UE, application, and services. Table 1 summarizes the high-level management features supported by the UE. The management features are grouped by Device, Users, and Application data components (see UE Management Model Components section in [PKT-UE-DATA]), and the traditional management functions: Fault, Configuration, Accounting, Performance (including monitoring), and Security.

Table 1 - Management Features Requirements for UE

Features	Management Functional Area	UE Data Component	Description
UE Bootstrapping	Configuration	UE device	Initial information required for the UE perform the Provisioning flow
Device IP Connectivity	Configuration	UE Device	The status of the device IP connectivity configuration
UE Capabilities	Configuration	UE Device	List of UE capabilities
User Private and Public Identities	Configuration	User, UE Device	Configuration of IP Multimedia Public and Private Identities
IMS/SIP network Setup	Configuration	User, UE device	Includes the necessary information to participate in the IMS/SIP infrastructure
PacketCable Presence	Configuration	User	Configuration of Presence Application. See [PKT-PRS].

Features	Management Functional Area	UE Data Component	Description
DM Server Access Control	Configuration, Security	UE device	Security and Access Control related to the OMA DM client and sever
IP Family Selection	Configuration, Fault	UE device	Device selection method of IPv4/IPv6 Addresses
Software Management	Configuration, Fault	UE device	Firmware upgrades management
Residential SIP Telephony	Configuration, Fault, Performance	UE device User, Applications	Listed here as an example of UE Data applications not covered in this specification. See [PKT-RST-UE-PROV].
Events	Configuration, Fault, Performance	UE device, User, Applications	Reporting of event and exceptions

5.3.1 Relationship to E-UE Provisioning Framework

The E-UE and UE Provisioning Framework are different and mutually exclusive. In the E-UE Provisioning specification, SIP registration occurs after the E-UE provisioning flow is complete. In UE Provisioning, SIP registration is part of the overall provisioning flow because SIP is used to enable management. SIP registration must succeed for the UE to complete provisioning. Further, these specifications use different protocols for configuration and management, and have different assumptions about the network environment. It is possible that a device supports both the UE and E-UE Provisioning Frameworks, and it needs to determine whether to initiate one provisioning flow or the other. Such a device needs to select the desired provisioning flow based on bootstrapping information; however there might be mechanisms to switch to the other provisioning framework. The need for such mechanisms is left undefined in this document.

5.4 OMA DM and SIP Push Overview

This section provides an informative overview of the OMA DM and OMA Push protocols.

5.4.1 OMA DM: Introduction

OMA DM consists of two stages. Stage 1 is a bootstrapping stage, and is discussed elsewhere in this document. Stage Two is the process by which the device and services within the device are managed through use of the OMA DM protocol.

The OMA DM protocol makes use of the SyncML Representation Protocol (as defined by [OMA-DMREPPRO]) and the OMA DM specifications. The SyncML Representation Protocol is an XML-based representation protocol that specifies an XML DTD to allow the representation of all the information required to perform synchronization or device management, including data, metadata, and commands. The OMA DM Specifications specify how SyncML messages conforming to the XML DTD are exchanged in order to allow a device management client and server to exchange additions, deletes, updates, gets, and other commands in a secure fashion.

Each device that supports OMA DM contains a management tree. The management tree organizes all available management objects in the device as a hierarchical tree structure where all nodes can be uniquely addressed with a URI. Nodes are the entities that can be manipulated by management actions carried over the OMA DM protocol. Each node has properties (name, type, etc.) associated with it.

A Management Object (MO) is a sub-tree of the management tree that is intended to be a logical collection of nodes that are related in some way. While in many cases an MO is structured as an object with multiple parameters, MOs

may also define complex hierarchies, with multi-instanced sub-trees, optional elements, and specific nodes provided for vendor extension.

The SyncML Representation and DM protocols are transport-independent. Each SyncML package is completely self-contained, and could in principle be carried by any transport. The initial bindings specified by OMA DM are HTTP, WSP, and OBEX. In IP-based environments, such as PacketCable, OMA DM makes use of HTTP as the transport protocol. The SyncML Representation defines both native XML and WBXML (WAP Binary XML) encodings for protocol messaging, and OMA DM requires the support of both formats (see [W3 XML] and [WBXML1.3]).

5.4.2 OMA DM: Architecture

Figure 2 illustrates the logical elements of the OMA DM architecture that are applicable to PacketCable and the interfaces that are defined.

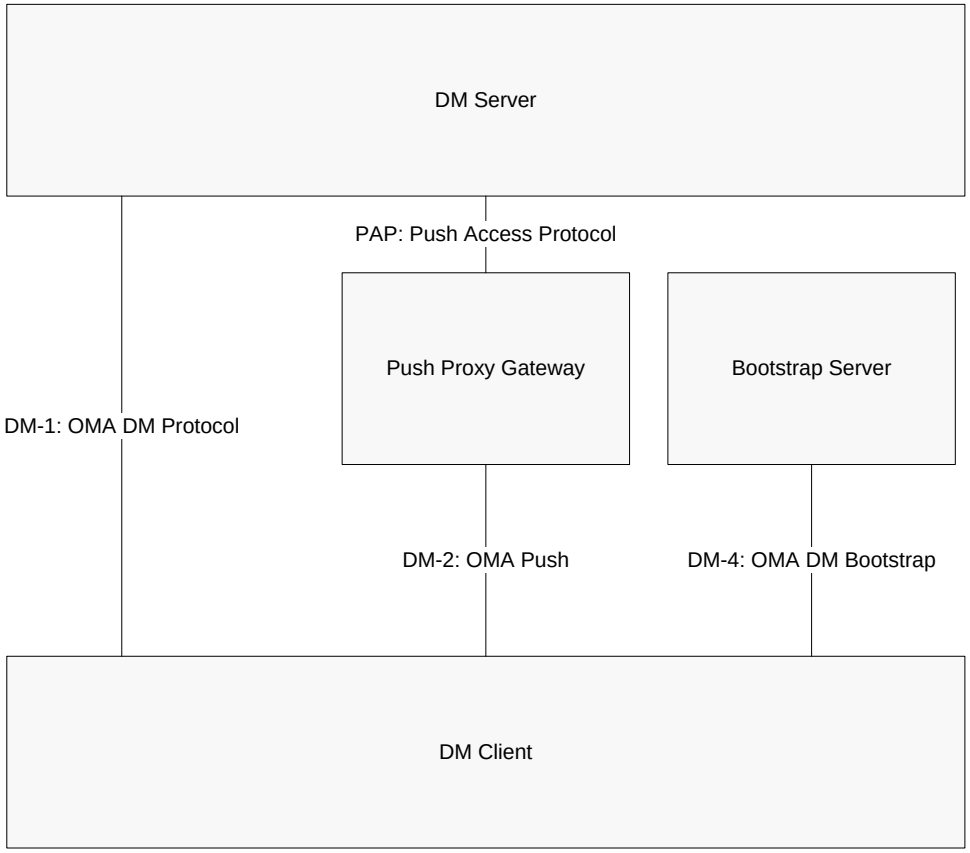

Figure 2 - OMA DM Architecture and Interfaces

5.4.2.1 OMA DM Architecture Elements

DM Server

An OMA Device Management Server is an entity integrated into a Device Management System that focuses on communicating with a device and/or other server(s) in order to provide management services. Further, it conforms to all the defined OMA DM server side requirements.

The requirements for an OMA DM Server include the protocol requirements in [OMA-DMPRO], the DM usage of the SyncML representation protocol requirements in [OMA-DMREPPRO], the security requirements in [OMA-

DMSEC], the OMA DM device management tree requirements in [OMA-DMTND], the standard OMA DM objects defined in [OMA-DMSTDOBJ], and the device description framework requirements in [OMA-DMDDF].

Additionally, an OMA DM server supports both the bootstrap (see [OMA-DMBOOT]) and notification (see [OMA-DMNOT]) mechanisms defined by OMA DM.

DM Client

An OMA DM Device Management Client is an entity that conforms to all the defined OMA DM client side requirements.

The specific client requirements include the protocol requirements in [OMA-DMPRO], the DM usage of the SyncML representation protocol requirements in [OMA-DMREPPRO], the security requirements in [OMA-DMSEC], the OMA DM device management tree requirements in [OMA-DMTND], the standard OMA DM objects defined in [OMA-DMSTDOBJ], and the device description framework requirements in [OMA-DMDDF].

Additionally an OMA DM client supports both the bootstrap (see [OMA-DMBOOT]) and server side notification (see [OMA-DMNOT]) requirements defined by OMA DM.

Push Proxy Gateway (PPG)

The Push Proxy Gateway is an entity defined by the OMA Push (see [OMA-PUSHAD]) enabler. The PPG is responsible for delivering push content to the client. In doing so, it potentially may need to translate the client address provided by the Push Initiator (PI) into a format understood by the mobile network, transform the push content to adapt it to the client's capabilities, store the content if the client is currently unavailable, etc. The PPG does more than deliver messages. For example, it may notify the Push Initiator about the final outcome of a push submission and optionally handle cancellation, replace, or client capability requests from the Push Initiator.

In the context of OMA DM, the PPG facilitates the use of OMA Push for the delivery of client notifications for DM session establishment, as described below.

Within the PacketCable use of OMA DM the Push Initiator is the DM Server. The PPG is a logical function that may be co-located with the DM Server.

Bootstrap Server

The Bootstrap Server is an entity whose role is to send bootstrap information to the OMA DM Client via a defined mechanism. In addition to basic connectivity information, device and User Application settings can also be configured during the bootstrap process.

Note that the Bootstrap Server is a logical function that may be co-located with the DM Server.

The OMA DM Bootstrap server conforms to the requirements in [OMA-DMBOOT].

5.4.2.2 OMA DM Architecture Interfaces

DM-1

DM-1 is the interface between the OMA DM Client and the OMA DM Server and is realized by the OMA DM protocol defined by [OMA-DMPRO].

DM-2

DM-2 is the interface between a PPG and an OMA DM client where the client provides OMA Push client capabilities. This interface is used to deliver OMA DM Package #0 from a DM Server to a DM Client to trigger client establishment of a DM session.

Multiple bindings are supported for DM-2 within OMA including WAP and HTTP. For the purposes of PacketCable 2.0 DM-2 will be realized by the Session Initiation Protocol (SIP) in accordance with [OMA-SIPPUSH].

DM-4

DM-4 is the interface between a Bootstrap Server and an OMA DM client. DM-4 is realized by OMA DM conveying an appropriate bootstrap MO; however, bootstrap has unique properties in that a Bootstrap Message is a one-off message that is not part of an existing DM session, and further no response is sent on reception of the message.

PAP

The Push Access Protocol is defined by [OMA-PUSHAD] to convey requests from a Push Initiator to a PPG. It is defined in [OMA-PUSHPAP] making use of HTTP as a transport. For the purposes of PacketCable 2.0, PAP may be used to initiate an OMA DM session by requesting that the PPG send Package #0 to the OMA DM client.

5.4.3 OMA DM: Management Session Establishment

OMA DM sessions are established from the client to the server. As stated within the PacketCable architecture, such client established sessions will make use of HTTP as the transport protocol; thus re-using the HTTP bindings already defined by OMA.

The network however, can (and in the case of PacketCable, will) request that the client initiates a DM session due to network-detected events such as a change in provisioned data or simply detection of a newly registered PacketCable UE instance. This is done via delivery of a notification (known as Package #0) from the server to the client. As within the PacketCable Architecture, the OMA DM server is logically an IMS Application Server, the PacketCable infrastructure will be used to deliver server-to-client notifications over SIP via the MESSAGE method, thus leveraging the NAT traversal capabilities provided by the PacketCable core. This is in accordance with the OMA SIP Push enabler (reference [OMA-SIPPUSH]).

5.5 UE Provisioning Model

A high-level conceptual diagram of all the UE provisioning components and provisioning flows is indicated in Figure 3.

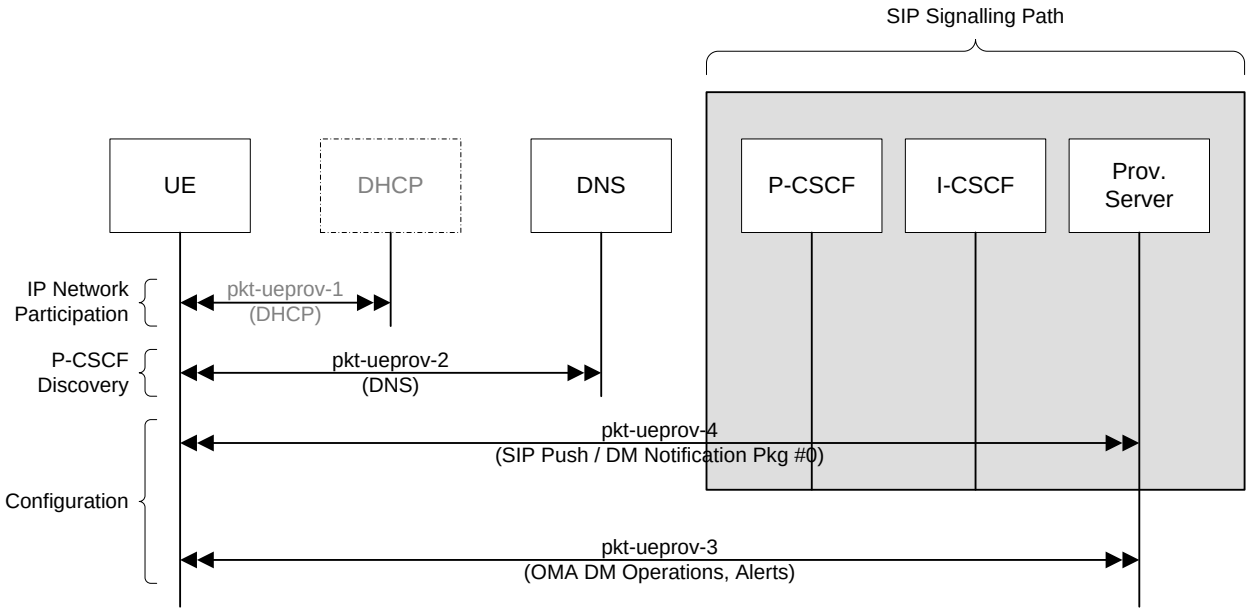

Figure 3 - PacketCable UE Provisioning Flow (conceptual)

The UE may interact with a DHCP server for address assignment and network configuration, which may not be under the control of the Operator. The UE interacts with a DNS server for P-CSCF discovery and other service discovery and server name resolution purposes. Interaction with the Provisioning Server is mostly direct, using OMA-DM over HTTP. OMA-DM provides initial UE configuration, incremental changes, collection of statistics, initiation of diagnostics, and reporting of events. When the Provisioning Server needs to initiate an OMA-DM session, it uses the SIP signaling path to send a DM Notification message to the UE using the SIP Push. Further detail of the roles of the network elements is provided below.

5.5.1 Network Elements

DHCP Server

The DHCP server may provide IP address assignment and network configuration parameters to the UE. It may provide DNS server addresses in support of locating other network elements. Because this framework is designed to support UEs in uncontrolled environments, the DHCP server is an optional component with few requirements other than basic DHCP service, as is typically provided by various Operators, enterprises, and home gateway devices. Its purpose is to assist the UE in IP network participation and access to network resources.

While there are DHCP options defined to provide SIP Servers names or addresses, and these are referenced in 3GPP IMS [PKT 24.229], the UE Provisioning Framework does not utilize those options for general UE provisioning, since all UEs cannot be assured of finding relevant information though those DHCP options. Instead, non-DHCP methods may be necessary to bootstrap the information needed to locate P-CSCF servers.

DNS Server

The DNS server provides Domain Name System name resolution services to resolve network element host names (FQDNs) to IP addresses, to resolve domain-based service names to host and port information (e.g., DNS SRV records), to provide namespace mapping between SIP URIs and SIP server connectivity information (e.g., DNS NAPTR records), etc.

Provisioning Server

The Provisioning Server is the focus of UE configuration and management operations, which are based on the OMA Device Management protocol. It provides configuration management, event collection, statistics gathering, firmware management, diagnostics execution, and troubleshooting support. All these operations are based on the OMA DM protocol, in which the Provisioning Server acts as a DM/HTTP server. Because OMA DM sessions are transient, and always based on client-initiated connections, the Provisioning Server embeds a DM Notification function used to trigger sessions from the client. The UE Provisioning Framework employs SIP Push protocols for delivery of DM Notifications, so the Provisioning Server also acts logically as an Application Server within the 3GPP IMS architecture framework.

Time Server

In the overall UE Provisioning process, there are a number of operations that depend on the successful acquisition and knowledge by the UE of the current time of the day. Verification of server certificates and other security credentials is often dependent on the time of the day. The event messages generated by UEs, and carrying UE related information, also require an accurate sense of time in enabling the Operator the correct synchronization between various back office components (network, security, billing, etc.). In all such operations, the time of the day acquisition becomes critical in ensuring the UE's ability to provide PacketCable 2.0 services.

5.5.2 UE Provisioning Framework Interfaces

The UE Provisioning Framework defines the interfaces depicted in Figure 4. The requirements of these interfaces are detailed in Section 6.2.

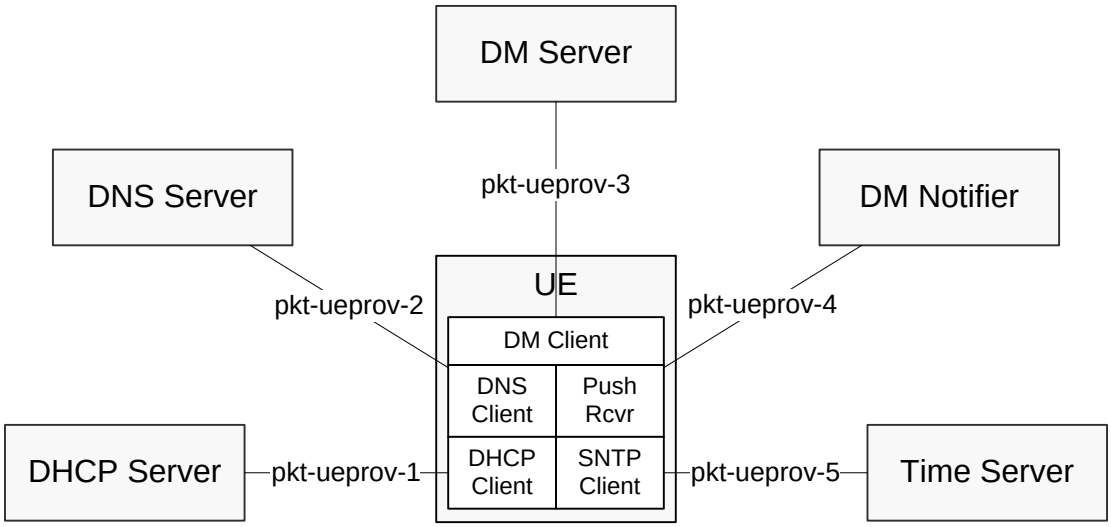

Figure 4 - UE Provisioning Interfaces (logical view)

In Figure 4 above, the pkt-ueprov-3 interface corresponds to DM-1 of Figure 2 shown previously, and pkt-ueprov-4 corresponds to DM-2, optionally combined with PAP. The pkt-ueprov-3 interface uses OMA-DM over HTTP. The pkt-ueprov-4 interface uses SIP Push via the SIP MESSAGE method.

5.6 UE Provisioning Data Models

The PacketCable UE Provisioning data model allows for a many-to-many relationship among users, devices, and applications. This is done by partitioning the data model logically between the UE itself, the users of the UE, and the applications provided to those users. See [PKT-UE-DATA] for more details.

This framework has data model requirements based on the OMA Device Management Tree and Description specification (DM TND, see [OMA-DMTND]) in order to support the OMA DM protocol. DM TND defines the abstract semantics of the management tree, which is a hierarchical, tree-oriented data model, composed of interior and leaf nodes, with provision for flexible type and format designation, access control, and extensibility. The DM protocol provides a variety of operators that create, read, modify and delete individual nodes or whole sub-trees in bulk.

In addition to specifying the data model abstraction, OMA DM TND also defines a markup language to describe its data models, called the Device Description Framework (DDF). This markup language is specified by the Device Description Framework XML DTD [OMA-DMDDF]. The DDF language is used to describe both common data models and device-specific data models. It provides for description of tree organization and composition from multiple other data models, and specification of default information.

The PacketCable UE Provisioning Data Model specification [PKT-UE-DATA] specifies a series of base data models to be supported by all UEs. While this specification mostly defines new PacketCable data models, it also defines PacketCable extensions to some existing data models defined by OMA.

5.7 UE Bootstrapping

Bootstrapping is a process that provides the information necessary for establishing connectivity with the Operator management system and PacketCable Core Network, e.g., device/UE identity, Operator identity and credentials. Examples of bootstrapping mechanisms include 3GPP ISIM SmartCards containing OMA DM bootstrapping documents, and protocol-based mechanisms based in device certificates, such as used in E-UE provisioning. In relation to OMA DM, a principal task of bootstrapping is to provision the device with a DM Account Management Object (DMAcc MO, see [OMA-DMSTDOBJ]), which describes addressing and authentication information for a DM server instance.

The specification of bootstrapping methods is out of scope for this version of the specification. Different bootstrapping mechanisms may be defined for specific UE types in device-specific ways. Bootstrapping will generally correspond to the Device Management Profile requirements of [OMA-DMBOOT], with some exceptions. Specific requirements of any bootstrapping method are defined in Section 6.5. This describes the minimal information that is assumed to be present on the UE prior to the provisioning process defined in this document.

5.8 PacketCable Core Network Components Provisioning

The PacketCable Core Network (Core) consists of the functional components required to provide SIP services and subscriber data. Application Servers (AS) provide specialized applications and services to users, and access the SIP facilities provided by the Core components for proper operation and interworking. See more details about the Core in PacketCable Architecture Technical Report [PKT-ARCH-TR].

This section describes a provisioning framework for the IMS Core components and ASs (referred as IMS Components thereafter) with the basic objective of provisioning user and application data from the OSS/BSS down to the particular subsystem (e.g., HSS).

5.8.1 Assumptions and Limitations

IMS components reside in the operator side compared to UEs that are facing the MSO access network. Therefore, the management of the IMS components is different of the network access devices like UEs. Primarily those differences are related to bootstrapping process, number of devices on the network, operating system, security models associated with the management of those devices, etc.

The following aspects are in scope of this provisioning framework:

- Management protocol for Provisioning and Configuration
- Security Model for the Management protocol

The following aspects are outside of the scope of this provisioning framework:

- Device Bootstrapping
- Access Control and Security configuration for management
- Fault and performance management
- Accounting

5.8.2 Core and Application Server Provisioning Model

The IMS Core and Application Server provisioning model is defined to be dynamic in nature. As the network evolves (e.g., new devices, users, and applications are added, removed, and updated overtime) the original configuration needs to be modified to support those changes. It is the responsibility of each functional aspect of the service or application offered to the users to define the persistence and retention of the datasets being provisioning to accommodate the business needs. The protocols defined in this section are seen as atomic operations with the purpose of creating, updating, deleting, and retrieving portions of the configuration and provisioning of datasets based on their underline object models.

5.8.3 IMS Core Components Provisioning Interface Architecture

This specification defines the usage of Web Services as the primary provisioning interface and follows the CableLabs Web services Recommended Practices [CLAB-WSRP].

5.8.3.1 Web Services

The messaging processing of the IMS Core Components provisioning interface is SOAP per [CLAB-WSRP]. RESTful architecture could be later defined if deemed necessary. The data model considerations of [CLAB-WSRP] separates the web services messages (defined in WSDL notation) from the underline data types in the form of XML Schema definitions (XSD). Moreover, the UE Data model approach [PKT-UE-DATA] separates the semantics from the syntax of the data elements by defining the data requirements in a protocol independent manner. Therefore, both [CLAB-WSRP] and [PKT-UE-DATA] are inline allowing the introduction of web services as extensions to existing object models (see [PKT-UE-DATA]).

Figure 5 shows the Web Services architecture defined for the IMS Components.

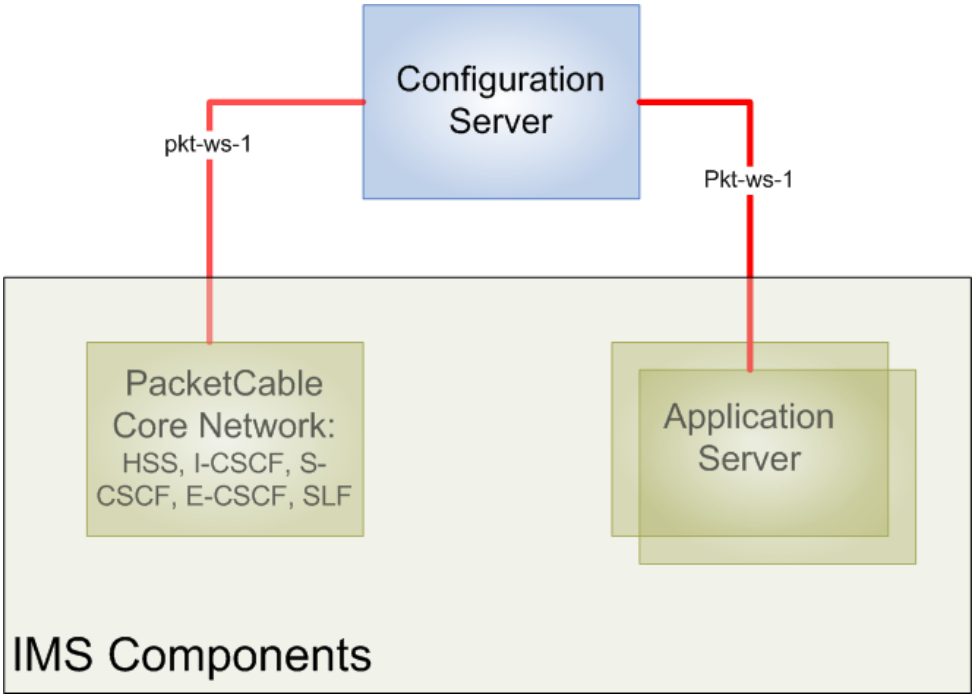

Figure 5 - Web Services Interfaces for IMS Components

Configuration Server

An Extension to the OSS Configuration Server [PKT-ARCH-TR] for the configuration of IMS Components.

PacketCable Core Network

As per [PKT-ARCH-TR].

Application Server

As per [PKT-ARCH-TR].

5.8.3.2 Web Services Interfaces

WS-1 is the interface between the OSS Configuration Server and the IMS Components following the requirements and design guidelines in [CLAB-WSRP].

5.8.3.3 Security

Web Services privacy and authentication are defined in the security section of [CLAB-WSRP]. This specification clarifies the requirements within the PacketCable constraints.

5.9 UE Provisioning Security

UE Bootstrapping provides credentials and configuration for SIP signaling security and OMA DM security. These credentials may or may not be related, and it is the Operator's choice as to whether the DM Server will employ separate credentials, and possibly a separate password store or different AAA mechanism.

OMA Device Management allows for security at the transport layer (TLS), at the HTTP protocol layer, as well as at the DM protocol layer. The security that will be used is based on what is configured in the UE's DMAcc MO settings corresponding to the DM server (this may configure multiple alternatives), and then negotiated between the DM Client and DM Server, according to [OMA-DMSEC]. Mutual authentication is a requirement of OMA DM.

The DMAcc information provided through UE Bootstrapping may indicate that the 3GPP Generic Bootstrapping Architecture [3GPP 33.220] is to be used in conjunction with the associated DM server, allowing DM authentication to leverage existing IMS credentials.

The DM Notification Package #0 over SIP Push is both protected by the security established with the PacketCable Core Network for SIP communication, as well as being internally secured by a keyed MD5 digest. It communicates only the identifier of the DM server to contact, and does not specify the contact URL, which must already be known to the UE.

For DM operations, the UE applies access control to the management tree based on ACL properties of nodes in the tree. The ACL properties describe specific operation permissions for each allowed DM server. This allows multiple DM servers to have different read or write access to parts of the UE's management tree.

6 UE AND IMS COMPONENTS PROVISIONING FRAMEWORK

This section presents the normative requirements for the PacketCable UE Provisioning Framework based on the Open Mobile Alliance Device Management protocol. This includes interfaces, data models, and requirements of specific logical or network elements.

6.1 UE Functional Architecture

Figure 6 represents the network components and interface interfaces that form the UE Provisioning Framework.

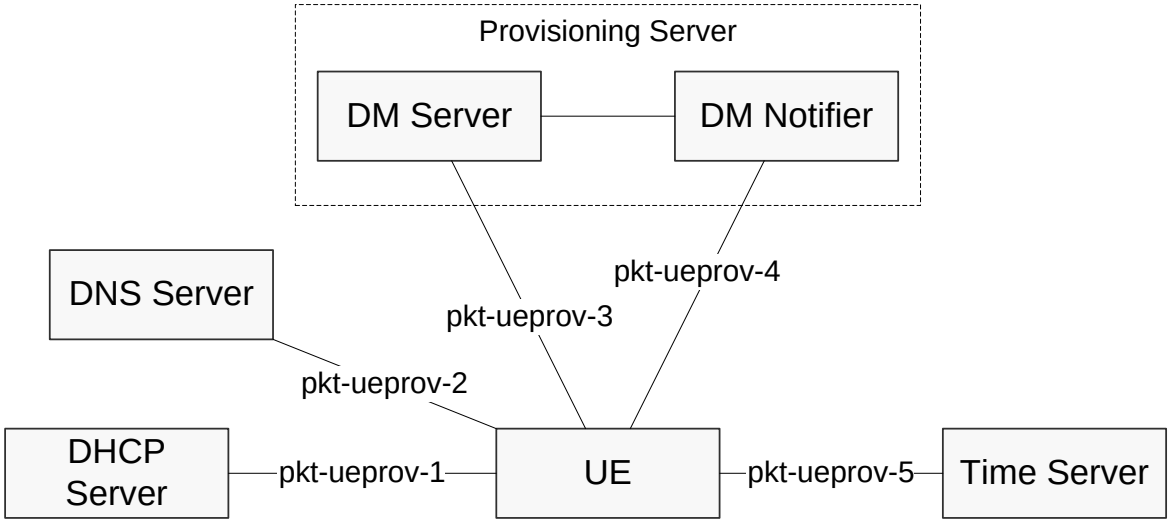

Figure 6 - UE Provisioning Components and Interfaces

6.2 UE Provisioning Interfaces

6.2.1 pkt-ueprov-1

The pkt-ueprov-1 interface reference point corresponds to the protocol exchanges between the UE, acting in the role of a DHCP client, and the DHCP server. This reference point facilitates the dynamic distribution of IP addresses and IP configuration information. The procedures relevant to pkt-ueprov-1 are defined by the IETF protocol specifications for DHCPv4 and DHCPv6, respectively, for IPv4 and IPv6 networks.

DHCP clients and DHCP servers implementing the pkt-ueprov-1 reference point for IPv4 MUST conform to [RFC 2131] and [RFC 2132].

DHCP clients and DHCP servers implementing the pkt-ueprov-1 reference point for IPv6 MUST conform to [RFC 3315] and [RFC 3646].

The pkt-ueprov-1 interface is used by UEs that do not obtain IP address and network configuration through other means (i.e., static configuration, IPv6 stateless address auto-configuration, or, in the case of soft client UEs, from the underlying operating system.) The primary purpose of pkt-ueprov-1 in support of this specification is the assignment of IP addresses and the configuration of DNS server addresses. Specific UE requirements for pkt-ueprov-1 are described in Section 6.3.1. Specific DHCP server requirements for pkt-ueprov-1 are described in Section 6.3.2.

6.2.2 pkt-ueprov-2

The pkt-ueprov-2 interface reference point corresponds to the protocol exchange between the UE, acting in the role of a DNS client and the DNS server. DNS clients and DNS servers implementing the pkt-ueprov-2 reference point MUST conform to the requirements in [RFC 1034], [RFC 1035], [RFC 2782], [RFC 2915], and for IPv6 use, [RFC 3596].

The pkt-ueprov-2 interface is to be used by the UE to locate OMA DM Servers, P-CSCF servers, and other servers, when those are not specifically configured by IP address.

6.2.3 pkt-ueprov-3

The pkt-ueprov-3 interface reference point corresponds to the protocol exchange between the UE, acting in the role of an OMA Device Management (DM) client, and the Provisioning Server, acting in the role of an OMA Device Management server. This reference point uses the OMA Device Management protocol described in [OMA-DM], transported over HTTP. It provides a protocol for device configuration, management, and event reporting.

DM clients and DM servers implementing the pkt-ueprov-3 reference point MUST conform to the OMA DM high-level protocol requirements, described in [OMA-DMPRO], the underlying representation protocol requirements, described in [OMA-DMREPPRO], the additional security requirements described in [OMA-DMSEC], and the HTTP binding requirements, described in [OMA-SYNCHTTP]. Additionally, DM clients and DM servers MUST conform to sections 5 and 6 of [OMA-DMNOT] for the implementation of DM Notification Initiated Sessions using the pkt-ueprov-4 reference point below.

6.2.4 pkt-ueprov-4

The pkt-ueprov-4 interface reference point corresponds to the protocol exchange between the UE, acting in the role of a DM Notification Receiver, and the Provisioning server, acting in the role of a DM Notifier. This interface allows the Provisioning Server to notify the UE when a DM session is needed, and provides a mechanism for delivery of the General Notification Trigger Message (Package #0). To be consistent with other OMA DM contexts, this message is delivered using the OMA Push OTA Protocol. While this interface model uses key components of the OMA Push architecture, this specification does not require the entire OMA Push architecture, and specifies a specific and limited use of OMA Push.

The underlying protocol of pkt-ueprov-4 is Push OTA Protocol over SIP (OTA-SIP), which uses SIP-based Push (SIP Push) [OMA-PUSHSIPAD]. Only the MESSAGE method of SIP Push delivery is used by this specification for DM Notification message delivery. All SIP messaging MUST be sent via the PacketCable Core Network. The DM Notification Receiver side of pkt-ueprov-4 corresponds to the P-1 reference point (see [OMA-PUSHSIPAD]) and the DM Notifier side corresponds to the P-2 reference point.

The DM Notification Receiver is a subset of the Push Receiver Agent defined in [OMA-PUSHSIPAD]. The DM Notification Receiver MUST conform to the terminal (Push Client) side requirements of OTA-SIP as specified in [OMA-OTAPUSH] for Registration and the MESSAGE method. The DM Notification Receiver MUST conform to the Push Receiver Agent requirements of SIP Push as specified in [OMA-SIPPUSH] for Registration and the MESSAGE method.

The DM Notifier MAY act as a Push Initiator, using the Push Access Protocol to send DM Notification Package #0 to the UE via a Push Proxy Gateway (see [OMA-PUSHAD]), which supports the Push Sender Agent function. Alternatively, the DM Notifier MAY embed a limited Push Proxy Gateway by directly implementing the Push Sender Agent function defined in [OMA-PUSHSIPAD].

The DM Notifier MUST act as either a Push Initiator or a Push Sender Agent according to the following requirements:

- The DM Notifier or PPG acting as a Push Sender Agent MUST conform to the PPG side requirements of OTA-SIP as specified in [OMA-OTAPUSH] for Registration and the MESSAGE method.
- The DM Notifier acting as a Push Sender Agent MUST do so in conformance with the requirements of SIP Push as specified in [OMA-SIPPUSH] for Registration and the MESSAGE method.
- DM Notifier MUST support the Sync ML Device Management Application-Id 'x-wap-application:syncml.dm', which corresponds to the delivery of DM Notification Package #0.

The DM Notification Receiver MUST also support the Sync ML Device Management application corresponding to the delivery of DM Notification Package #0.

6.2.5 pkt-ueprov-5

The pkt-ueprov-5 reference point corresponds to the protocol interchange between the UE, acting in the role of a Simple Network Time Protocol (SNTP) client, and the Time server, acting as a Network Time Protocol (NTP) or SNTP server.

The procedures for the pkt-ueprov-5 reference point are defined by the SNTP specified in [RFC 4330] and the NTP protocol specified in [RFC 1305]. It is to be noted that both NTP and SNTP servers can support SNTP clients, and the differences are more to do with how they themselves determine the current time.

SNTP clients MUST conform to the requirements in [RFC 4330].

6.3 UE Provisioning Components

This section describes the network components that implement the interface interfaces in Section 6.2 and the associated requirements.

6.3.1 User Equipment (UE)

The UE is a PacketCable User Equipment component, which may be a standalone hardware device, an eSAFE device embedded with a Cable Modem, or a software application hosted by end-user equipment (i.e., a soft-client on a PC). The requirements of this specification apply to all UE types implementing the UE Provisioning Framework. Other specifications may impose additional requirements for specific types of devices, or modify these requirements as necessary for special environments or application.

- The UE MUST implement the pkt-ueprov-3 reference point as a DM Client.
- The UE MUST implement the pkt-ueprov-4 reference point as a DM Notification Receiver.
- The UE MUST support the OMA DM Tree and Description Serialization encoding and decoding requirements as specified in [OMA-DMTNDS].

6.3.1.1 Network Participation Requirements

The UE MAY support operation in either an IPv4 or IPv6 network environment, or in both environments. Depending on the IP network environment supported, the UE MUST support the corresponding DHCP client requirements of the pkt-ueprov-1 reference point defined in Section 6.2.1, unless the UE is a soft-client application dependent on the services of the host operating system. A UE that supports both IPv4 and IPv6 operation MUST comply with the requirements for IP address family selection in Section 6.4.

The UE MUST obtain network parameters (IP address and other network configuration) using the DHCP according to the pkt-ueprov-1 reference point, unless pre-configured or obtained by other means.

If the UE has obtained an IP address using a non-DHCP process, but has not obtained the network configuration parameters (e.g., DNS Servers), it MUST use the pkt-ueprov-1 reference point to request additional parameters from the DHCP server using the protocols specified in pkt-ueprov-1.

A UE supporting IPv4 operation and using DHCP for network configuration MUST support the use of the DHCP Domain Name Servers option 6 as defined in [RFC 2132] to learn the addresses of local DNS recursive name servers. The UE MUST request option 6 via the Parameter Request List option 55 in any DHCPDISCOVER, DHCPREQUEST, or DHCPINFORM messages sent to the DHCPv4 server.

A UE supporting IPv6 operation and using DHCPv6 for network configuration MUST support the use of the DHCPv6 DNS Recursive Name Server option 23 as defined in [RFC 3646] to learn the addresses of local DNS recursive name servers. The UE MUST request option 23 via the Option Request Option 6 in any DHCPv6 Solicit, Request, Renew, or Information-Request messages sent to the DHCPv6 server.

If the UE has pre-configured DNS server information, it SHOULD combine the information from both sources, giving priority to the DHCP-provided information.

The UE MUST implement the client side of the pkt-ueprov-2 reference point according to Section 6.2.2.

A UE MAY implement the SNTP client side of the pkt-ueprov-5 reference point, in order to acquire the current time of the day using the Simple Network Time Protocol (SNTP) as defined in [RFC 4330]. If acquiring the time of the day, the UE MUST act as an SNTP Client.

6.3.2 DHCP Server

In uncontrolled environments, the DHCP server used by the UE is not likely to be within the administrative domain of the Cable Operator. In this case, the DHCP server is outside the scope of the requirements of this document, though the basic requirements of DHCP address assignment and configuration of DNS server information are likely to be met by any DHCP server encountered by the UE.

When operating as part of a controlled environment, the DHCP server MUST conform to the server side requirements of the pkt-ueprov-1 reference point.

6.3.3 DNS Server

In uncontrolled environments, the DNS server used by the UE may or may not be one within the administrative domain of the Cable Operator. While the UE may be configured to use the operator's DNS server, it may be configured, or even required, due to network restrictions, to use a local DNS resolver. Any DNS server not under the operator's control is outside the scope of the requirements of this document, though the basic requirements of standards-compliant DNS service and support for SRV and NAPTR records are likely to be met by any DNS server providing Internet DNS name resolution.

When under the control of the Cable Operator, the DNS server MUST conform to the server side requirements of the pkt-ueprov-2 reference point.

6.3.4 Provisioning Server

The Provisioning Server combines both the DM Notifier and the DM Server functions. There may be multiple instances of the Provisioning Server component for different purposes. For example, there may be one Provisioning Server for UE configuration, and another for User configuration. There may be Provisioning Servers used in limited roles, such as for event collection or statistics gathering.

The Provisioning Server MUST conform to the server side requirements of the pkt-ueprov-3 reference point as specified in Section 6.2.3.

The Provisioning Server MUST implement the pkt-ueprov-4 reference point acting as a DM Notifier, according to Section 6.2.4. The Provisioning Server SHOULD implement the limited Push Sender Agent role of pkt-ueprov-4. However, it MAY implement a full Push Proxy Gateway supporting OTA-SIP. It MAY act as a Push Initiator, depending on a separate PPG acting as a Push Sender Agent, and MUST act as a Push Initiator if it does not implement a PPG/Push Sender Agent.

6.3.5 Time Server

The Time Server MUST implement the server side of the pkt-ueprov-5 reference point. The Time Server SHOULD function as an NTP server and MAY function as an SNTP server.

6.4 IP Address Family Selection

A UE capable of supporting both IPv4 and IPv6 needs to determine which address family (IP version) to use for operation with the PacketCable network elements defined in this and other PacketCable specifications. The PacketCable 2.0 Architecture Framework [PKT-ARCH-TR] specifies that a single IP version be selected by the UE for operations and management. When a UE is capable of supporting IPv4 and IPv6, it MUST perform address family selection as follows:

- For UEs that control their address acquisition process, attempt acquisition of IPv4 and IPv6 addresses using either DHCP, stateless address auto-configuration (IPv6), or static IP address configuration.
- If the UE successfully obtains both an IPv4 and IPv6 address, then the UE will use [RFC 3484] to determine the address family that will be used for bootstrapping, P-CSCF discovery, and SIP registration.
- Until the UE registers successfully, it MAY use any acquired IP address for SIP and non-SIP communications.

Once the UE has succeeded in SIP registration, it MUST adhere to the following requirements:

- If the UE successfully registered with one IP address, it will use the same IP address for all other SIP and non-SIP communication related to PacketCable operation and management, such as OMA DM.
- If the UE successfully registered with more than one IP address, it can use any of the IP addresses it used for registration for SIP and non-SIP communications related to PacketCable operation and management, such as OMA DM.

Figure 7 provides a flowchart for IP network configuration of a UE that is in control of its address acquisition (typically a standalone or embedded hardware UE).

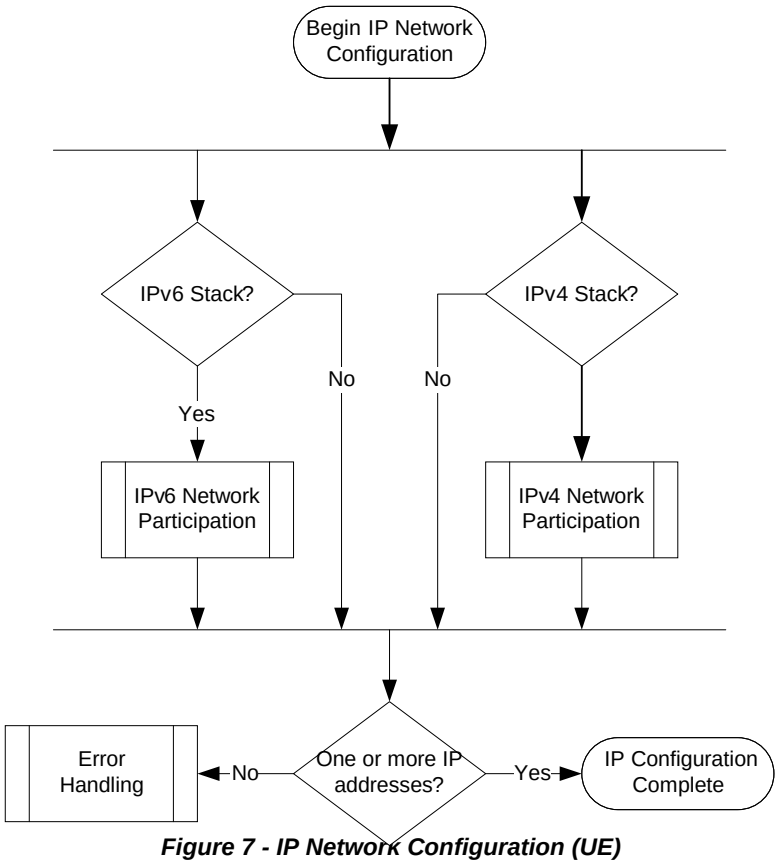

Figure 7 - IP Network Configuration (UE)

6.5 UE Bootstrapping

No specific bootstrapping mechanism is required or specified by this version of the document. Various mechanisms may be specified in the future for specific types of UEs, and an optional common method may also be specified. However, bootstrapping itself is generally dependent on some initial data, which will vary from mobile handsets, to soft-clients, to hardware embedded UEs. The UE bootstrapping mechanism MUST meet the Device Management Profile requirements of [OMA-DMBOOT]. However, the UE bootstrap content MAY be XML rather than WBXML.

Regardless of the bootstrapping mechanism used, the UE must have certain information in order to proceed with the general provisioning flows described in this document. Accordingly, the UE MUST securely obtain the data listed in Table 2, through either pre-configuration or a bootstrapping process.

Table 2 - Required Bootstrap Information

Data	Description
DMAcc MO for UE Provisioning and Management	The DMAcc MO MUST conform to [OMA-DMSTDOBJ]. A single DMAcc MO MUST be present, corresponding to the DM Server used for UE-level provisioning. If multiple DMAcc MOs are present, the UE must know which single DMAcc instance is intended for UE Provisioning. Requirements for this DMAcc MO: • The <X>/ServerID node MUST contain the server identifier used in Pkg#0 notifications from the UE-level Provisioning Server (see [OMA-DMNOT]). • The <X>/AppAddr information MUST specify the URL or IP address of the UE-level Provisioning Server. • The <X>/AppAuth information MUST specify the appropriate authentication techniques and credentials for use with UE-level Provisioning Server. • The <X>/PrefConRef SHOULD be present, unless only one connectoid is configured, in which case the <X>/ToConRef/<X>/ConRef SHOULD link to that connectoid.
DMAcc PacketCable extension (optional)	As specified in [PKT-UE-DATA], the PacketCable extension to DMAcc provides additional settings specific to PacketCable 2. This data SHOULD be provided by UE Bootstrapping, but default behavior will occur in its absence. When present, this data MUST include: • <X>/Ext/Pktc2/Boot/Init/DMInitConnect, which describes whether the device will immediately initiate an OMA DM HTTP connection (beginning with Pkg #1) on device reset. • <X>/Ext/Pktc2/Boot/Init/AuthExt, which indicates extended OMA DM authentication behavior, such as the use of GBA.
UE IMS Identity and Connectivity Data	This MAY be in the form of the 3GPP IMS MO [3GPP 24.167] or equivalent data. The data MUST include: • UE's Private Identity used for SIP REGISTER • UE's home network domain name. This data SHOULD include a UE Public Identity for use in SIP communication with the UE, unless the Private Identity is used for this purpose. This data SHOULD include P-CSCF FQDN or IP address, unless the UE home network domain name is to be used for P-CSCF discovery
UE IMS Credentials	The UE MUST have IMS Credentials corresponding to the IMS Private Identity used by the UE for SIP REGISTER.
BSF Location	The UE MAY be configured with the location of a GBA BSF server, or MAY be configured to locate a BSF through the home network domain.
Connectivity Data	The UE MUST have some connectivity information, such as an instance of a Connectivity MO [OMA-CONNMO] or other appropriate connectivity configuration. This is referenced by the DMAcc MO, and by referenced by the 3GPP IMS MO, if present.

Any correspondence between the identities and credentials used by the UE for SIP REGISTER and for OMA DM authentication are outside the scope of this specification, but maybe determined by operator policy or DM Server implementation requirements.

6.6 UE Provisioning Security

The core requirements for the UE and Provisioning Server with respect to OMA DM security, are specified in the OMA Device Management Protocol specification, and the OMA Device Management Security specification [OMA-DMSEC]. The configuration of authentication mechanisms, identities, and credentials to be used with a DM Server instance are conveyed in the DMAcc MO (see [OMA-DMSTDOBJ]).

While DMAcc supports indicating the use of transport-level security (using <X>/AppAuth/<X>/AAuthType = 'TRANSPORT'), it does not specify the configuration of credentials/certificates for transport-level security. When establishing a DM session to a server where the AAuthType is configured as 'TRANSPORT', the UE MUST follow this procedure:

- Use TLS (see [RFC 4346]) when establishing a establish the DM session.
- If the corresponding AppAuth/<X>/AAuthName contains a value that matches the CN attribute of the SubjectName field of a UE certificate, use this certificate for client authentication when requested by DM server.
- Authenticate the DM server by validating the server certificate, and ensuring that the CN attribute of the SubjectName field corresponds to the host portion of the URI, or to configured IP address, in the AppAddr/<X>/Addr node used to contact this server.

6.6.1 GBA Usage

The UE SHOULD support the use of the 3GPP Generic Bootstrapping Architecture [3GPP 33.220] to establish credentials for use with OMA DM. The need for GBA is indicated through the PacketCable extension to the DMAcc MO. If the DMAcc MO for a particular DM server contains the <X>/Ext/Pktc2/Boot/Init/AuthExt node with the value '3GPP-GBA', then a UE that supports GBA MUST employ the GBA protocol to obtain a shared key for use with DM server authentication. The UE implementing GBA MUST conform to the requirements for use of GBA with HTTP as specified in [3GPP 33.222].

In order to use the GBA protocol, the UE must locate a Bootstrapping Function Server (BSF). The UE may be configured with the exact hostname(s) or IP address(es) of the BSF for an Operator, either through bootstrapping or through (or modified by) subsequent provisioning or configuration operations. If the BSF name is configured, the UE MUST use DNS to resolve this name to one or more IPv4 or IPv6 addresses, depending on the IP network environment. Alternatively, the UE may be configured to locate the BSF using DNS, based on the Operator domain name. When so configured, the UE MUST locate the BSF through DNS SRV records (see [RFC 2782]) using the service identifier 'gba-pktc' and the protocol identifier 'tcp'.

The credentials used by the UE for GBA MUST be SIP credentials that match the AppAuth/<X>/AAuthName specified in the DMAcc MO being used. When the AAuthName is contains a SIP identity of the form “user@domain”, the domain name MUST be used to locate the BSF information from the BSF Object (see [PKT-UE-DATA].) If the AAuthName does not include a domain, the UE default SIP credentials are used, and the BSF information is located based on the UE default operator domain.

6.7 UE Provisioning Flows

UE Provisioning involves UE Bootstrapping, IP network participation, P-CSCF discovery (if not supplied by bootstrapping), SIP Registration, optional SIP Subscription, DM Notification, and finally DM session establishment.

The portion of the UE Provisioning Flow specified in this document assumes certain initial conditions:

- UE Bootstrapping has occurred, meeting the requirements specified above.
- Network participation has occurred, providing IP connectivity.
- P-CSCF discovery, or configuration via bootstrapping.

The general UE Provisioning flow is shown in Figure 8. The steps are depicted abstractly, both to reduce detail and to allow for realization via different protocols. While this flow presents UE Provisioning, the same flow may apply to User provisioning for devices configured with additional user identities. An overview of the flow is shown in Figure 8.

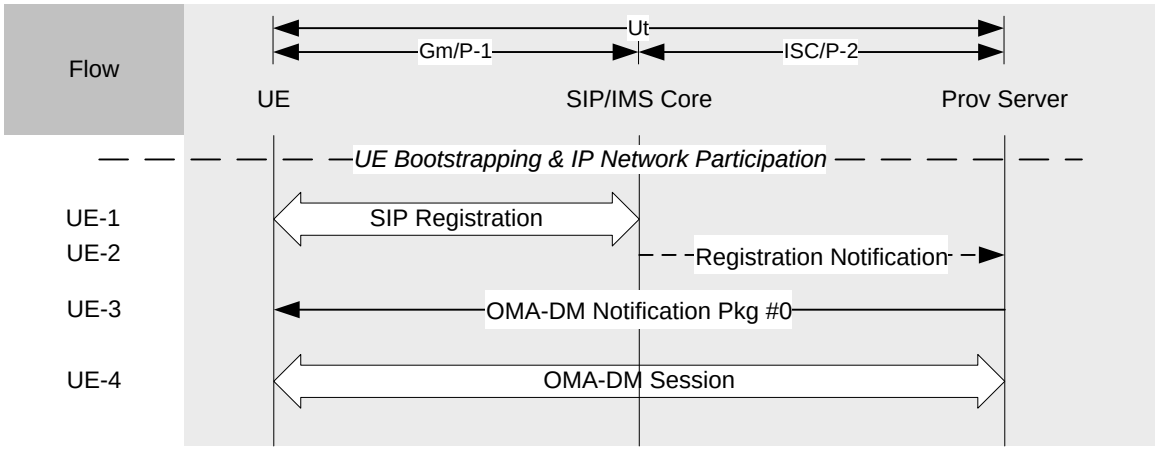

Figure 8 - UE Provisioning Flow Overview

The flow begins with SIP Registration at UE-1. This is a basic requirement of PacketCable, specified in [PKT 24.229]. SIP registration is required in the UE Provisioning Flow in support of the SIP Push mechanism that will be used for DM Notification to deliver DM Package #0.

The Provisioning Server may have acted to receive notification of SIP Registration, either as an AS receiving third-party REGISTER, or via the "reg" event package (see [PKT 24.229]). If so, the Provisioning Server will be informed, at UE-2, of the registered public user identity, or the entire implicit registration set, depending on the chosen mechanism.

The UE-3 step depicts an OMA DM Notification Package #0 message. This may occur immediately, as shown, or may occur after some period of time, when the Provisioning Server is prepared to provision the UE. This notification is carried in a SIP MESSAGE request.

The UE-4 step depicts an OMA DM session. This is an HTTP-based communication session, initiated by the UE acting as an HTTP client. It may be triggered by the receipt of OMA DM Package #0 (UE-3), or it may be initiated by the UE without waiting for DM Notification, due to the bootstrapping configuration of <X>/Ext/Pktc2/Boot/Init/DMInitConnect. The bootstrapping configuration should be reflective of the operator's intent to deliver OMA DM Package #0 during UE Provisioning.

Steps UE-3 and UE-4 may be repeated any number of times in the future for incremental management and provisioning. UE-4 may occur at any time in the future as required by the UE, based on a configured polling interval or based on the need to send event notifications or other status, without being solicited by DM Notification Package #0 (UE-3).

The requirements of the UE Provisioning flow steps are described in Table 3, and show the UE Provisioning flow logic.

Table 3 - UE Provisioning Flow Steps

Step	Description
UE-1	SIP Registration by the UE MUST follow the requirements of [PKT 24.229], using Private Identity and credentials provided by the bootstrap mechanism. Additionally, the UE MUST follow the additional registration requirements specified in [OMA-SIPPUSH] and the OTA-SIP section of [OMA-OTAPUSH].
UE-2	If the Provisioning Server chooses to monitor SIP Registrations, the Provisioning Server MUST follow the procedures of [PKT 24.229], section 5.7.1.1, optionally including use of the "reg" event package according to [RFC 3680]. The Provisioning Server MUST monitor SIP Registration unless the operator chooses to disable such monitoring. Monitoring might be disabled because bootstrapping configuration has ensured that all UEs are configured to initiate a DM Session autonomously during provisioning.
UE-3	A message containing DM Notification Package #0, requesting a DM session, MAY be sent immediately following SIP Registration (UE-1), or at any future time. The UE MUST process the DM Notification content according to [OMA-DMNOT].
UE-4	Following successful SIP Registration (UE-1), the UE MUST request an immediate OMA DM session, if configured to do so via the PacketCable DMAcc extension DMInitConnect object. If the UE receives DM Notification Pkg #0 (UE-3) at any time from a configured DM server, the UE MUST immediately initiate an OMA session to the indicated DM server according to the corresponding DMAcc MO configuration, unless the UE is already in a DM session with that server. When establishing a DM Session, the UE MUST conform to the requirements of [OMA-DMPRO]. When the AppAddr of the relevant DMAcc entry specifies the use of the HTTP protocol, the OMA DM Session MUST use the HTTP binding as specified in [OMA-SYNCHTTP]. The OMA DM session MUST proceed according to [OMA-DMPRO] until the Provisioning Server terminates the session.

Figure 9 shows the same abstract provisioning flow steps in the context of bootstrapping and IP network configuration. It demonstrates the effect of the bootstrap configuration setting, which controls whether the UE initiates an immediate DM session.

After UE configuration, the UE will have information about specific users that are configured on the UE. These User identities may be marked as active and mapped to endpoints on a UE device, or may be activated on-demand through a user-interaction that activates a specific user-profile. When a User identity is activated on the UE, the UE MUST perform SIP Registration with the corresponding IMPU/IMPI, including the indication of SIP Push support as specified in [OMA-OTAPUSH].

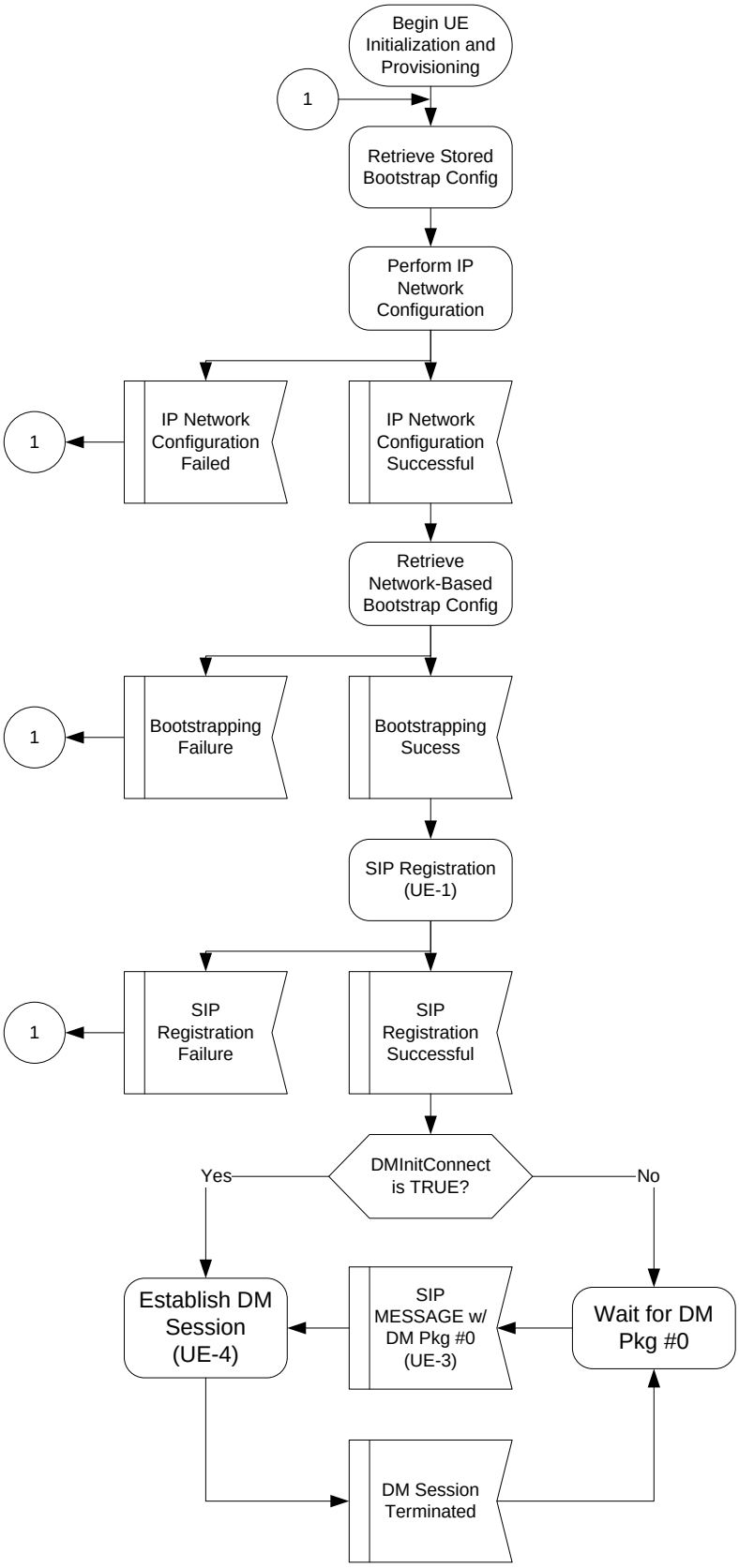

Figure 9 - UE Provisioning Flow

6.8 UE Management

This sections details OMA DM Protocol requirements and management requirements per conventional Network Management functions.

6.8.1 OMA DM Protocol Requirements for UEs

This section describes UE requirements in the implementation of the OMA DM protocol, while maintaining a minimal set of requirements to be OMA DM interoperable.

The UE uses the OMA DM protocol [OMA-DM] for management. Management sessions may be initiated by the UE, to report events or alerts, or initiated by the Provisioning Server, to poll for statistics or perform troubleshooting.

Server-initiated management sessions are triggered by sending DM Notification Package #0 to the UE over the pkt-ueprov-4 interface. This is realized in the delivery of SIP Push content to the UE for the DM Notification Push application. The Package #0 content includes the Server-ID of the requesting server, and causes the UE to initiate a DM session, opening an HTTP connection to the notifying server and sending DM Package #1.

Client-initiated management sessions are internally triggered due to events or configuration, and begin with DM Package #1 over HTTP. The DM server that is contacted is determined by the configuration of the events or alerts. The UE may also initiate management sessions to report on the delayed result of a diagnostic or other asynchronous or scheduled operation.

The UE DATA Specification defines two types of requirements: Protocol independent Object Models that represents the semantics and syntax of the UE managed features; and OMA DM MOs. The UE specifications will reuse existing OMA DM MOs when possible and define UE specific Object Models and MOs when necessary.

The UE MUST support the semantics of the Object Models requirements in [PKT-UE-DATA].

The UE MUST implement the OMA MOs defined or referenced in [PKT-UE-DATA].

The Provisioning Server MUST implement the Object Model semantics requirements and OMA MOs protocol implementation in order to manage the UE devices.

The UE MUST support the Alert Code Command (inbound and outbound) to communicate with the Provisioning Server about the state of protocol communications and transactions as stated in [OMA-DMPRO].

A special case of Alert command is the Generic Alert that uses the event code 1226 [OMA-DMPRO]. The support of Generic Alert is specified in Section 6.8.3.

6.8.2 UE Configuration Management Requirements

The UE MUST support the mandatory OMA DM protocol commands specified in [OMA-DMREPPRO].

The command Exec provides an extensible way to introduce asynchronous tasks in the operation of the device; however, Exec operation is optional by [OMA-DMREPPRO]. Most traditional UE operations are inherited from previous PacketCable specifications. In the absence of Exec commands those specifications performed SET operations (equivalent to OMA DM 'Replace' operations), e.g., device reset, or enabling a feature, etc. Whenever OMA 'Replace' operations could cover the required functionality, this specification and the UE Data specification will use traditional set operations (i.e., 'Replace' command) instead of 'Exec'.

In general there are cases where OMA DM optional commands such as 'Exec', 'Atomic', and 'Copy' are required by the OMA specified MOs. In those cases the Access will be in conformance to the MOs, otherwise specified in the UE specifications.

OMA DM guidelines include the definition of MOs with minimal access of read (OMA DM 'Get'). The UE specifications may need to specify a different type of access depending on the UE management needs.

The OMA DM Protocol provides capabilities to perform device initial provisioning and incremental provisioning.

The Object models need to indicate the persistent storage of the configurable elements in order to provide a clear management policy in the provisioning system (see [PKT-UE-DATA] for more details).

6.8.3 Fault Management Function

This section defines the UE requirements for fault management including event reporting, diagnostics, and troubleshooting.

The UE MUST support the management requirements for event reporting defined in this specification and the object model and MOs specified in [PKT-UE-DATA]. The UE MAY support other diagnostic features defined by OMA DM not covered in this specification.

6.8.3.1 Event Notification Requirements

The UE MAY support PacketCable Events and/or Vendor-specific Events.

Core PacketCable-specific events are defined in this document. Additional application-specific events may be defined by other PacketCable documents. Vendor-specific events are left to vendor implementation and are out of scope for this specification.

Each Event has an associated Event ID as described in the next sub-section. PacketCable-Specific events are identical if their EventIDs are identical. The PacketCable-Specific EventIDs are specified by the PacketCable Specifications, including this specification. For each particular vendor, Vendor-specific events are identical if the corresponding Event IDs are identical. The Vendor-specific EventIDs are defined by particular vendors and are out of scope for this specification.

Example:

- Two or more PacketCable Events with the same Event ID (say 4000970100) are considered to be identical irrespective of the description or other parameters.
- Two or more Vendor-Specific Events, from the same vendor (say XYZ) with the same Event ID (say 10) are considered to be identical irrespective of the description or other parameters.

For identical events occurring consecutively, the UE MAY choose to store only a single event. In such a case, the event description recorded MUST reflect the most recent event.

Aside from the procedures defined in this document, event recording MUST conform to the requirements of [PKT-UE-DATA]. The Event Descriptions MUST NOT be longer than 127 characters.

6.8.3.1.1 Event ID Assignments

The EventID is a 32-bit unsigned integer.

PacketCable-specific EventIDs MUST be defined in the range of 0x80000000 (decimal 2,147,483,648) to 0xFFFFFFFF (decimal 4,294,967,295).

Vendor-specific EventIDs MUST be defined in the range of 0x00000000 (decimal 0) to 0x7FFFFFFF (decimal 2,147,483,647).

Vendor-specific EventIDs MUST be unique for a particular vendor's enterprise number.

6.8.3.1.2 PacketCable Management Event Format

The format of a PacketCable Management Event is made up of the following information:

- Event Counter - indicator of event sequence
- Event Time - time of occurrence
- Event severity - severity of condition as defined in the Management Event Severities sub-section below. Severities for legacy events may be obtained through translation with help from Appendix II.
- Event Enterprise number - Vendor-specific enterprise number
- Event ID - determines event function
- Event Text - describes the event in human readable form
- FQDN/Endpoint ID - describes the device FQDN and the specific endpoint associated with the event

6.8.3.1.3 Management Events

Core UE Management Events are defined in Annex A of this document. Additional UE Events may be defined in other application-specific documents. Previous PacketCable specifications defined similar events; but, with a different format or severity value than that defined by this specification. Appendix II of this document provides a mapping and method of using legacy PacketCable events within this framework.

6.8.3.1.3.1 Management Event Severities

Each event is assigned an initial (default) PacketCable Severity. The definitions for the Severity follow those used by the OMA DM Generic Alert Mark parameter [OMA-DMPRO]. The following levels are allowed: fatal, critical, minor, warning, informational, harmless, and indeterminate. The order indicates the importance level with "critical" as the most important and indeterminate as least important.

If events need to be cleared, they MUST be cleared by other events.

6.8.3.1.3.2 Changing Default Event Severities

The default event severity MUST be changeable to a different value for each given event via the MEM interface.

6.8.3.1.4 Management Event Reporting Mechanism

The UE supports a set of defined events, as well as allows for vendor-defined events. It maintains a log of recent events and makes these data available through the MEM MO [PKT-UE-DATA]. Unsolicited events can be reported to a management entity through the use of the Generic Alert and the configuration provided by the DiagMonTrapMO [OMA-DIAGMONTRAPMO]. Additionally the SYSLOG method may be used to send SYSLOG messages.

6.8.3.1.4.1 Notification Mechanism

The notification mechanism for each event MUST be programmable via the MEM MO.

The UE MUST support the sending of each event to one or more of the event notifications listed below.

The notification mechanism definitions are as follows:

- local: The event is stored locally on the device in which it is generated. The event can be retrieved via polling from the MEM MO interface.
- alert: The event is sent via the Generic Alert mechanism defined in this document and by the DiagMonMO, see [PKT-UE-DATA] for details on this OMA DM UE requirements.
- syslog: The event is sent to the SYSLOG server. The use of syslog is optional and may not be supported by the UE. If the UE does not support SYSLOG, the UE MAY ignore this notification mechanism setting.
- none: No reporting action is taken, this is the equivalent of disabling the event. If "none" is specified, the other notification mechanism choices MUST be ignored.

6.8.3.1.4.2 Local Log of Events

The UE MUST support local logging of events. The local log MUST be accessed via the MEM MO defined in [PKT-UE-DATA]. A vendor may provide alternative access procedures.

The UE MAY implement local logging either in volatile memory, non-volatile memory, or both. The UE MUST clear the events not intended to be in non-volatile memory at UE re-initialization. The UE MUST add new events to the local log after the previously persisted log entries in a manner that the retrieval process of the log will present events sorted from the older to the latest.

The UE MUST clear both the local-volatile and local-nonvolatile logs when commanded through the MEM MO.

6.8.3.1.4.3 Generic Alerts of Events

The UE MUST use the OMA DM Generic Alert [OMA-DMPRO] to send unsolicited messages to a DM Server.

This specification defines two types of Generic Alerts:

- OMA Standard Notifications are characterized for a Meta/Type value of 'Reversed-Domain-Name:org.openmobilealliance.dm.diagmon.trap', and an Item/Data equal to the registered Trap ID, see [OMA-DIAGMONTRAPMO].
- PacketCable-defined UE client events contain a Meta Type value of 'Reversed-Domain-Name:com.cablelabs.pkt.oma.dm.ue-events' and an Item/Data as defined in Table 6.

The reason for these two types of events is the need to support standard OMA Generic Alerts and PacketCable-specific events. The PacketCable-specific events carry extra information in the event Description (see Annex A), while the OMA generic alerts do not allow that information in the Item/Data element.

The UE MAY support OMA registered Generic Alert notifications that follows the rules of [OMA-DIAGMONTRAPMO]. The UE MUST support PacketCable-defined UE client as defined in Table 6.

The UE MUST support the mechanisms to configure the sending of OMA Standard notifications and PacketCable-defined UE clients events as defined in [PKT-UE-DATA].

6.8.3.1.4.4 SYSLOG Event Access

The SYSLOG method of accessing events involves sending the events to a SYSLOG server via the UDP protocol to the UDP SYSLOG port as defined in [RFC 3164]. UE support for SYSLOG is optional.

The UE MAY support the use of SYSLOG messages for event reporting. If SYSLOG is supported, the UE MUST comply with the format and requirements of this section. If supported, the UE MUST follow the event data format as defined in Section 6.8.3.1.5.

All Syslog messages sent by a UE MUST comply with the following requirements:

- The UE MUST use UDP as the transport mechanism with 514 as the destination port as defined in section 2 of the BSD syslog protocol [RFC 3164].
- The UE SHOULD use port 514 as the source port, as recommended in section 2 of SNMP applications [RFC 3164].
- The UE MUST comply with the Packet Format and Contents as defined in section 4 of [RFC 3164] as applicable to the origination of the message, and use the format as described in the following sub-section.

6.8.3.1.5 Event Message Format

Event messages sent to external systems such as Generic Alerts or syslog must encode the information payload as defined for syslog messages defined in [RFC 3164] with the variations and requirements listed below.

6.8.3.1.5.1 PRI Part of a Syslog Packet

For the PRI part defined in section 4.1.1 of [RFC 3164], the UE MUST use the 'Numerical Code' 16; corresponding to the 'Facility' local use 0 (local0).

The severity is the severity as indicated in the definition of the Event message (0-7).

The 'Priority Code' is as defined in section 4.1 of [RFC 3164] and ranges between 128 and 135 for PacketCable.

6.8.3.1.5.2 MSG Part of a Syslog Packet

The UE MUST include the following components:

TIMESTAMP, HOSTNAME, TAG, and the CONTEXT.

Where:

- TIMESTAMP is the time recorded by the UE; this MUST reflect the time in UTC as obtained from the Cable Modem.
- HOSTNAME MUST be the UE Public Identity of the UE.
- The TAG field MUST be set to the string 'UE', without the quotes.
- The PID field MUST be implemented and used as an 'Event Type Identifier'. The value MUST be 'PACKETCABLE' for all PacketCable-defined Event Messages.
- A vendor-specific unique identifier for vendor-defined Event Messages. While the vendor-specific choices are out of scope of this specification, a vendor MUST use the same unique identifier for all messages originating from a device).
- The CONTEXT part of the message MUST be formatted as follows:

  <eventID><correlationID> Description

  Where:

  - eventID MUST be the Event ID defined for each Event Message enclosed within angular braces.
  - correlationID MUST be set to 0 and is kept for format consistency with previous PacketCable specifications.
  - Description MUST be the description associated for the particular event as stored in the Management Event MO.

Example 1:

PROV-EV-1 is a PacketCable-defined 'Event', defined as follows:

Table 4 - Example PacketCable-defined Event

Event Name	Event Priority	Default Display String	PacketCable EventID	Comments
PROV-EV-4	critical	"Not all UE and User profiles were configured; may also contain warning(s) and error(s)"	4000970003	Not all the UE and User profiles were downloaded. Additionally, one or more of the downloaded profiles may also contain errors or warnings.

- The Event Priority value for "critical" is 2; hence the 'Priority Code' is 130. The value is calculated by multiplying the Facility code times eight and adding the priority per [RFC 3164].
- Since this is a PacketCable-defined event, the 'Event Type Identifier' is 'PACKETCABLE'.
- The defined Event ID is 4000970003 and assuming the default string has not been changed, the associated text is "Not all UE and User profiles were configured; may also contain warning(s) and error(s)".
- Assume the hostname to be user@domainname.com.

Thus, the event, if triggered, will be sent as the following SYSLOG message:

<130>Jan 1 09:00:00 user@domainname.com UE: [PACKETCABLE]:<4000970003><0>
Not all UE and User profiles were configured; may also contain warning(s) and error(s)".

Example 2:

Assume the following hypothetical vendor-specific event is defined by vendor 'XYZ Inc', with vendor ID 'XYZ'.

Table 5 - Example Vendor-specific Event

Event Name	Event Priority	Display String	Vendor Specific EventID	Comments
XYZ-EV-1	warning	"AC Power Failure; running on battery"	10	AC Power Failure occurred and the device is running on battery power.

- The Event Priority for warning is 4 and hence the 'Priority Code' is 132.
- Vendor ID is 'XYZ', as stated in the example.
- The defined Event ID is 10 and the display string as indicated is: 'AC Power Failure; running on battery'.
- Assume the hostname to be user@domainname.com.

Thus, the event, if triggered will be sent as the following SYSLOG message:

<132> Jan 11 21:04:03 user@domainname.com UE: [XYZ]:<10><0>AC Power Failure; running on battery.

6.8.3.1.6 PacketCable Event Notification Alert Format

The UE MUST format the Generic Alert [OMA-DMPRO] as specified in this section and summarized in Table 6.

Table 6 - Generic Alert Fields

Generic Alert Message Element	Value	Observations
Alert/CmdId	Any number	Per [OMA-DMPRO]
Alert/Data	1226	Per [OMA-DMPRO]
Alert/Correlator	Optional	Vendor-specific per [OMA-DMPRO]
Alert/Item/Source/LocURI	Not used	
Alert/Item/Meta/Type	'Reversed-Domain-Name: com.cablelabs.pkt.oma.dm.ue-events'	Reversed domain-name com.cablelabs.pkt.oma.dm.ue-event
Alert/Item/Meta/Format	Chr	Per [OMA-DMPRO]
Alert/Item/Meta/Mark	Fatal | critical | minor | warning | informational | harmless and indeterminate.	Per [OMA-DMPRO]
Alert/Item/Data	Event payload	Per Section 6.8.3.1.5

The UE MUST encode the following Generic Alert Message Elements as:

- Alert/Data: 1226 (always for generic alert). The Generic Alert: 1226.
- Alert/Correlator: Only used in case of asynchronous responses to Exec commands [DM-Protocol].
- Alert/Item/Source/LocURI: The UE SHOULD NOT provide a value for this element.
- Alert/Item/Meta/Type: Reversed domain-name com.cablelabs.pkt.oma.dm.ue-events.
- Alert/Item/Meta/Mark: The severity of the Event and MUST be included. Allowable values include: fatal, critical, minor, warning, informational, harmless, and indeterminate. The order indicates the importance level with critical as the most important and indeterminate as the least important.
- Alert/Item/Data: The text defined in Section 6.8.3.1.1.

6.8.3.2 Resetting or Initializing the RST UE

The Resetting or Initializing the UE is a process that brings all UE components and data elements to their initial state, followed by the execution of the procedures required for UE to initiate the UE functionality. The UE resetting or initializing can be done by different means, for example by using the management interface, or by power cycling the particular RST UE. When being reset or initialized, a UE MUST adhere to the following requirements:

- Report the "Resetting Event",
- Gracefully terminate all related RST services currently active.

6.8.4 Security Management Function

The OMA DM security mechanisms are described in Section 5.9. This section describes requirements for Access Control List (ACL) configuration of the UE management requirements.

The UE MUST support ACLs as specified in [OMA-DM]. Note that users may manipulate device and applications data. OMA DM does not provide access control other than using the OMA DM interfaces. Therefore, and unless otherwise specified, the UE MUST not allow the modification of UE configuration data via non-OMA DM interfaces. Certain applications can still provide user configuration capabilities indirectly, such as configuring the service via an operator hosted application and pushing the user information via DM servers with the appropriate ACL configuration.

6.8.5 Performance Management Function

The UE MUST reset its counters after client hard reset or operating system re-initialization. Note that if the OMA DM client is reinitialized, it does not necessarily means that the UE application or other resources will have reinitialized their counters.

The UE MUST provide rollover capabilities for counters. It means the device MUST reset a counter to zero when it reaches the maximum value. Note that counters do not define range constraints, thus, rollover occurs when all bits of the data type are set to one and the value is increased by a value equal or greater than one.

6.8.6 Accounting Management Function

There are no accounting Management requirements for the UE.

6.8.7 Software Management Function

The UE MUST support the client software upgrade requirements as specified in [CLAB-SM] and Annex B.

6.8.7.1 Requirements

The UE can support monolithic firmware image or software components. The UE MUST accept the Trigger Message defined in Annex B. However, the UE MUST check only for well-formed Trigger Messages. The UE MUST ignore unknown trigger message elements.

The UE supports the OMA DevInfo MO, and the DevDetail MO (see [PKT-UE-DATA]), in order to provide the management system with information about hardware, firmware, and software of the UE. If the UE supports software modules, the UE MUST report the Device Capabilities property (Group/Attribute/Property) Software/Others/InstalledSoftwareList defined in [OMA-DPE] in the OMA Device Capabilities MO [OMA-DCMO].

Table 7 summarizes the list of MOs that a software manager is expecting to retrieve from the UE for software upgrades management.

Table 7 - UE Hardware and Software Information

OMA Element	OMA MO	Reference
./DevInfo/Man	DevInfo MO	[OMA-DMSTDOBJ]
./DevInfo/Mod	DevInfo MO	[OMA-DMSTDOBJ]
./DevInfo/DmV	DevInfo MO	[OMA-DMSTDOBJ]
./DevDetail/DevTyp	DevDetail MO	[OMA-DMSTDOBJ]
./DevDetail/OEM	DevDetail MO	[OMA-DMSTDOBJ]
./DevDetail/FwV	DevDetail MO	[OMA-DMSTDOBJ]
./DevDetail/SwV	DevDetail MO	[OMA-DMSTDOBJ]
./DevDetail/HwV	DevDetail MO	[OMA-DMSTDOBJ]
<*>/Group = 'Software <*>/Property = 'InstalledSoftwareList'	Device Capability MO	[OMA-DCMO]

The UE MUST support HTTP for the retrieval of the UE firmware and/or software modules. The UE follows the security guidelines defined in [CLAB-SM]. The UE MUST use the authentication credentials of the DMAcc MO instance pointed by the TransportSpecificControlInfo/AuthServerID element of the SwModuleControlData in the Trigger Message when downloading the UE software image from the ModuleFileURI.

Note that DM Server credentials are used indirectly to perform the Software Server-Client authentication and this server is not necessarily an OMA DM compliant Server; The UE MUST support at least one of the following DMAcc Authentication Levels: CLCRED, SRVCRED, and HTTP. The UE MUST support at least one of the following authentication types: HTTP-BASIC, HTTP-DIGEST, BASIC, DIGEST, HMAC, X509.

6.8.7.2 Client software Upgrade Process

The UE uses the OMA DM FUMO MO [OMA-DMFUMO] to trigger the software upgrade by a DM Server.

The UE procedure is different to the standard FUMO procedures as the UE is instructed to download a Trigger Message File defined in Annex B and not the binary file itself. However it is convenient to use the same FUMO MO for that purpose as defined below:

The UE MUST support the FUMO MO ./downloadAndUpdate node to trigger the software upgrade. The UE MUST start downloading the Trigger Message file set in the ./downloadAndUpdate/pkgURL immediately after a command exec sent to the UE includes a ./downloadAndUpdate node. The UE MUST be able to recognize a Trigger Message instead of a binary image file by the Content-Type text/xml in the HTTP packets. The UE MUST be able to process the content of the Trigger Message stored, and schedule the firmware upgrade according to the content of the TriggerControlData, instead of an immediate firmware upgrade as the standard FUMO procedure.

The UE MAY support the other firmware upgrade procedures described in [OMA-DMFUMO].

6.9 Additional Requirements

6.9.1 UE PacketCable Capabilities reporting

The reporting of PacketCable capabilities by the UE is an important aspect of proper UE configuration. To support capabilities reporting, the DevInfo managed object [OMA-DMSTDOBJ] is extended to include PacketCable device capabilities information. The definition of this extension is specified in [PKT-UE-DATA]. The UE Capabilities are defined in Annex C. When the UE sends DM Package #1 to initiate a DM session, it MUST always send the DevInfo MO within a Replace command, according to [OMA-DMPRO]. The UE MUST include the PacketCable extension to DevInfo within Package #1.

6.9.2 P-CSCF Discovery

The exact hostname(s) or IP address(es) of the P-CSCF MAY be configured on the UE, either through bootstrapping or through (or modified by) subsequent provisioning or configuration operations. When configured with this information, the UE MUST use the configured data. If the P-CSCF name is configured, the UE MUST use DNS to resolve this name to one or more IPv4 or IPv6 addresses, depending on the IP network environment.

This framework specification makes no requirement on the use of DHCP for P-CSCF discovery. Unless permitted or required by an overriding specification, the UE MUST NOT perform DHCP-based P-CSCF discovery as specified in section 9.2.1(c)(I) of [PKT 24.229].

The UE MAY be configured to perform discovery of P-CSCF names and addresses using DNS NAPTR and SRV resource records, starting from a configured domain name corresponding to a proxy. When so configured, the UE MUST perform such discovery according to section 4.1 of [RFC 3263], starting with the URL "sip:<proxy>", where '<proxy>' represents the configured proxy domain name.

6.9.3 Battery Backup

UEs supporting Battery Backup MUST support the requirements of the PacketCable Battery Backup MO, as specified in [PKT-UE-DATA]. Additionally, UEs MUST use the identifier "UE" when required within the context of Battery Backup (e.g., reporting values of UPS-attached devices).

6.10 IMS Components Functional Architecture

Figure 10 represents the components and interfaces for the IMS Components configuration.

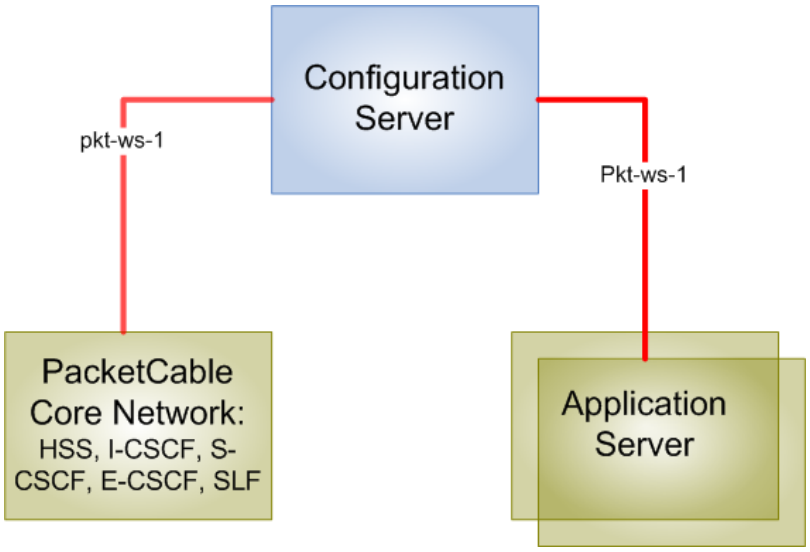

Figure 10 - IMS Components and Configuration Interfaces

6.11 IMS Components Provisioning Interfaces

6.11.1 pkt-ws-1

The pkt-ws-1 interface reference point corresponds to the protocol exchange between the IMS Components and the Configuration Server. The pkt-ws-1 consists of a message processing model and Service Contract in the form of WSDLs. This specification only defines message processing model requirements. Each WS configuration and provisioning feature defines its own requirements in the form of WSDL requirements.

The pkt-ws-1 follows the [CLAB-WSRP] requirements to support HTTP SOAP 1.2 Bindings for all the operations defined within the WSDL. Other bindings maybe optionally supported.

6.12 IMS Components Web Services Interface Elements

The Configuration Server MAY support the pkt-ws-1 reference point interface.

The IMS Components MAY support the pkt-ws-1 reference point interface.

6.13 IMS Components Provisioning Interface Security

If the Configuration Server supports the pkt-ws-1 reference point interface, the Configuration Server MUST support at least HTTP Basic authentication, and traffic encryption with SSL 3.0 and TLS 1.0 as defined in [CLAB-WSRP].

If the IMS Component supports the pkt-ws-1 reference point interface, the IMS Component MUST support at least HTTP Basic authentication, and traffic encryption with SSL 3.0 at the minimum and TLS 1.0 optionally as defined in [CLAB-WSRP].

Annex A UE Management Events (Normative)

A.1 UE Provisioning Events

The UE MUST support the events listed in Table 8.

Table 8 - UE Provisioning Events

Event Name	Default Severity for Event	Default Display String	Packet-Cable EventID	Comments
UE configuration				
PROV-UE-1	**informational**	"Successful acceptance of all UE and User profiles"	4000970000	All the UE and User profiles were successfully downloaded and accepted without any warnings or errors that were reported.
PROV-UE-2	**warning**	"UE and User profiles configured, warnings reported"	4000970001	All the UE and User profiles were successfully downloaded and some warnings were reported, but no errors.
PROV-UE-3	**minor**	"UE and User profiles configured; contains error(s) and warning(s)"	4000970002	All the UE and User profiles were successfully downloaded, but one or more of them contained errors and possible warnings.
PROV-UE-4	**critical**	"Not all UE and User profiles were configured; may also contain warning(s) and error(s)"	4000970003	Not all the UE and User profiles were downloaded. Additionally, one or more of the downloaded profiles may also contain errors or warnings.
UE Registration				
EUE-EV-1	As per [PKT-EUE-DATA]			Notes: - The reference to pktcEUEUsrIMPUSigSecurity corresponds to the UE attribute SigSecurity of the IMPU object defined in [PKT-UE-DATA]. - eUE context of the event applies to UE in this specification.
EUE-EV-2	As per [PKT-EUE-DATA]			Note: eUE context of the event applies to UE in this specification.
UE Software Management				
UE-SM-1	**warning**	Software Download Failed: <reason description>	4000970100	Reason description can include HTTP error Response, retries limit reached, etc.
UE-SM-2	**warning**	Software Downloaded Failed Verification: <reason description>	4000970101	Reason description can include Invalid File image, Time limit to activate the new image expired, etc.
UE-SM-3	**warning**	SwAuthenticationFailure: <reason description>	4000970102	Reason description can include e.g., CVC validation failures, etc.
UE-SM-4	**harmless**	SW activation delayed due user activity: <activity>	4000970103	Activity includes e.g., RST activity, PacketCable gaming application, etc.
UE-SM-5	**harmless**	SW activation Completed	4000970104	
RST UE				
			400098xxxx	Reserved for RST UE, see [PKT-RST-UE-PROV].

A.2 UE Powering Events

If the UE supports Battery Backup, the UE MUST support the Powering Events defined in [PKT-MEM1.5].

Annex B Software Upgrade Trigger Message (Normative)

This Annex defines the PacketCable UE Software Management Trigger Message format in conformance with [CLAB-SM] and the restrictions indicated below.

B.1 Software Upgrade Trigger Message Object Model Definitions

B.1.1 Software Upgrade Trigger Message Requirements

The UE MUST support the PacketCable UE Software Management Trigger Message requirements defined in [CLAB-SM].

The UE MUST implement the syntax of the Trigger Message as defined in B.3 with the following considerations.

B.1.1.1 *TriggerDownloadType Element*

The UE MUST support the pull mode described in [CLAB-SM].

The UE MUST at least process Trigger Messages with 'TriggerDownloadType' equal to 'pull'. The required configuration for the 'push' method is outside of this specification.

B.1.1.2 *TriggerControlDataVersion*

The UE MUST support and process the Trigger Message with an element 'TriggerControlDataVersion' identical to any of the values the 'UE Software Upgrade Trigger Versions' capability, (Annex C). Note that the Trigger Message includes the version in the Trigger Message XML document, as well as in the TriggerControlDataVersion element.

The UE MUST ignore unknown data elements in the Trigger Message.

B.1.1.3 *SwModuleControlData*

B.1.1.3.1 TargetDevice

The UE MUST process only Trigger Messages that contain in the 'TargetDevice' element the value 'UE and the value associated with the UE registration IMPU.

B.1.1.3.2 DownloadTime

The UE MUST initiate the download of the Trigger Message immediately if the element 'DownloadTime' is not present, or if it has a value prior to the current time (e.g., epoc Time).

The UE MUST attempt up to 3 retries to download the UE image file.

B.1.1.3.3 ActivationTime

Delay in seconds to initiate the software activation after successful download. The UE MUST ignore the element 'ActivationTime/Value' and 'ActivationTime/Delay' when the attribute 'ActivationTime/Control' is 'immediately' or 'uponCriticalServices'.

When the element 'ActivationTime/Control' in the Trigger Message is 'uponCriticalServices', the UE MUST delay the new software image activation for a period of time 'ActivationTime/Delay' or the default value of 15 minutes (900 seconds if not included in the Trigger Message), where non-critical services have been executed by the UE users.

The UE MUST delay the activation up to 4 times the value in 'ActivationTime/Delay' when 'ActivationTime/Control' is equal to 'uponCriticalServices'.

B.1.1.3.4 ActivationAfterModule

THE UE MAY support the element 'ActivateAfterModules' with vendor proprietary extensions.

B.1.1.3.5 ModuleFileURI

See [CLAB-SM].

B.1.1.3.6 ModuleFileSize

See [CLAB-SM].

B.1.1.3.7 HashType

The UE MUST support SHA-1 and SHA-256 as the hash algorithms to check the integrity of the Trigger Message.

B.1.1.3.8 ModuleFileHash

The Hash is calculated by using the content of the trigger message without the ModuleFileHash element. Note that outside the ModuleFileHash, trailing characters such as spaces or carrier returns introduced by the hashing function can't be detected by the client. Therefore, implementers should avoid those conditions.

If the 'ModuleFileHash' element is present in the Trigger Message, the UE MUST use the ModuleFileHash element value to verify the integrity of the downloaded image.

B.1.1.3.9 ValidationCVC

The UE MUST perform Code Verification Certificate (CVC) verification as defined in [CLAB-SM].

B.1.2 Error Conditions

The UE is not required to validate the Trigger Message against the XML Schema definition. However, the UE MUST discard the Trigger process and log an error in case the device found an error while processing the Trigger Message document. (See Annex A for details on UE Software Management Errors.)

B.2 Software Upgrade Trigger Message Object Model Definitions

Figure 11 shows the UE Object model for the software Management Trigger Message.

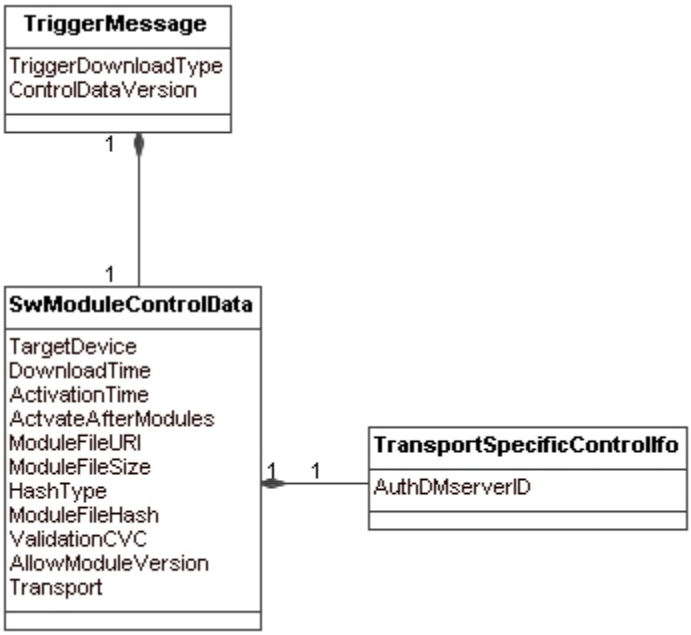

Figure 11 - UE Software Management Trigger Message Object Model

B.3 Software Upgrade Trigger Message XML Schema definition

The UE MUST support the schema definition.

```
<?xml version="1.0"?>
<schema xmlns="http://www.w3.org/2001/XMLSchema"
targetNamespace="http://www.cablelabs.com/namespaces/PacketCable/2.0/xsd/ue/CL-SM-
TRIGGER-MESSAGE" xmlns:CL-
SM="http://www.cablelabs.com/namespaces/PacketCable/2.0/xsd/ue/CL-SM-TRIGGER-MESSAGE"
        version="1.0" elementFormDefault="qualified"
attributeFormDefault="unqualified" >
   <element name="TriggerDownloadType" default="pull">
      <simpleType>
         <restriction base="string">
            <enumeration value="pull"/>
            <enumeration value="push"/>
         </restriction>
      </simpleType>
   </element>
   <element name="ControlDataVersion" type="string"/>

   <element name="TargetDevice" type="string" abstract="false"/>
   <element name="DownloadTime" type="dateTime"/>

   <element name="Control">
      <simpleType>
         <restriction base="string">
            <enumeration value="immediately"/>
            <enumeration value="dateAndtimeIndicated"/>
            <enumeration value="uponCriticalServices"/>
         </restriction>
      </simpleType>
   </element>
   <element name="Value" type="dateTime"/>
   <element name="Delay" type="int"/>

   <element name="ActivationTime">
```

```
      <complexType>
         <sequence>
            <element ref="CL-SM:Control"/>
            <element ref="CL-SM:Value" minOccurs="0"/>
            <element ref="CL-SM:Delay" minOccurs="0"/>
         </sequence>
      </complexType>
   </element>
   <element name="ActivateAfterModules" type="IDREFS"/>
   <element name="ModuleFileURI" type="anyURI"/>
   <element name="ModuleFileSize" type="int"/>
   <element name="HashType">
      <simpleType>
         <restriction base="string">
            <enumeration value="SHA-1"/>
            <enumeration value="SHA-256"/>
         </restriction>
      </simpleType>
   </element>
   <element name="ModuleFilehash" type="hexBinary"/>
   <element name="ValidationCVC" type="hexBinary"/>
   <element name="AllowModuleVersion">
      <simpleType>
         <restriction base="string">
            <enumeration value="older"/>
            <enumeration value="same"/>
            <enumeration value="new"/>
         </restriction>
      </simpleType>
   </element>

   <element name="pull" type="string"/>
   <element name="push">
      <simpleType>
         <restriction base="string">
            <enumeration value="dsm-cc"/>
            <enumeration value="dsg"/>
            <enumeration value="mpeg"/>
            <enumeration value="ipmulticast"/>
         </restriction>
      </simpleType>
   </element>

   <element name="Transport">
      <complexType>
         <choice>
            <element ref="CL-SM:pull"/>
            <element ref="CL-SM:push" minOccurs="0" maxOccurs="1"/>
         </choice>
      </complexType>
   </element>
   <element name="TransportSpecificControlInfo">
      <complexType>
         <sequence>
            <element ref="CL-SM:AuthServerID"/>
         </sequence>
      </complexType>
   </element>
   <element name="AuthServerID" type="string"/>



   <element name="SwModuleControlData">
      <complexType>
         <sequence>
            <element ref="CL-SM:TargetDevice"/>
            <element ref="CL-SM:DownloadTime" minOccurs="0"/>
```

```
            <element ref="CL-SM:ActivationTime"/>
            <element ref="CL-SM:ActivateAfterModules"/>
            <element ref="CL-SM:ModuleFileURI"/>
            <element ref="CL-SM:ModuleFileSize" minOccurs="0"/>
            <element ref="CL-SM:HashType"/>
            <element ref="CL-SM:ModuleFilehash"/>
            <element ref="CL-SM:ValidationCVC"/>
            <element ref="CL-SM:AllowModuleVersion" minOccurs="1"/>
            <element ref="CL-SM:Transport"/>
            <element ref="CL-SM:TransportSpecificControlInfo"/>
         </sequence>
      </complexType>
   </element>

   <complexType name="TriggerMessage">
      <sequence>

         <element ref="CL-SM:TriggerDownloadType"/>
         <element ref="CL-SM:ControlDataVersion"/>
         <element ref="CL-SM:SwModuleControlData"/>
      </sequence>
   </complexType>
</schema>
```

Annex C UE Capabilities (Normative)

Table 9 summarizes the PacketCable capabilities that apply to UEs.

The UE reports the PacketCable capabilities in the DevInfo extension defined in [PKT-UE-DATA].

Table 9 - UE Capabilities

Type	Reference
5.1 PacketCable Version	[PKT-EUE-PROV]
5.2 … 5.25	Not supported. Reserved for [PKT-EUE-PROV] and/or [PKT-PROV1.5].
5.26 Battery Backup Support	C.1.2
5.27 … 5.35	Reserved for [PKT-RST-UE-PROV].

C.1.1 PacketCable Capabilities from other non-UE specifications

The UE MUST support the capability type 5.1 PacketCable version as defined in [PKT-EUE-PROV].

C.1.2 Battery Backup Support

The UE MUST report the Battery Backup Support capability.

Type	Length	Values	Comment	Default Value
5.26	1	0	Not Supported	N/A
		1	Supported	

Appendix I Sample UE Provisioning Flow (Informative)

The diagram below shows an example of the provisioning flow for a UE using the SIP MESSAGE method of SIP PUSH. This has been simplified to omit authentication challenges within both SIP and OMA DM.

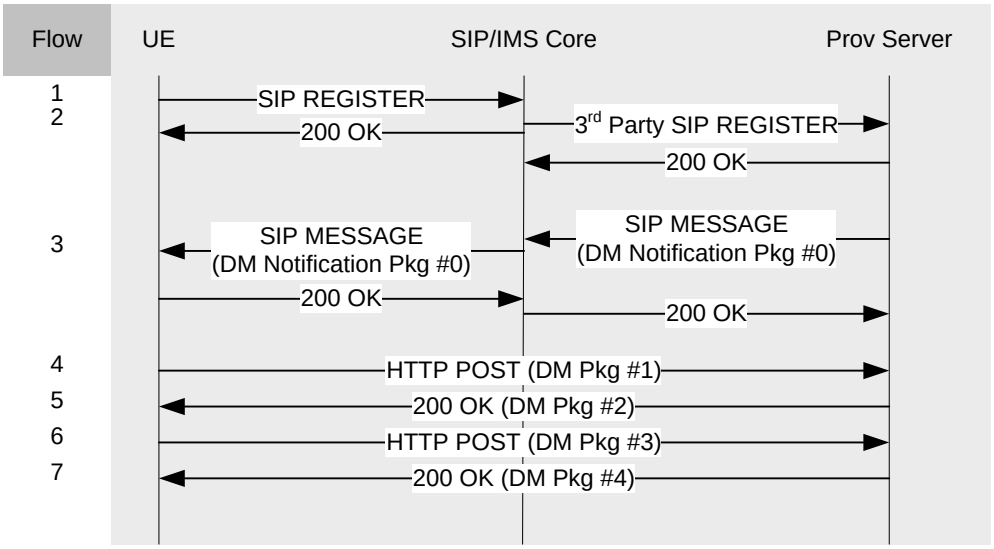

Figure 12 - Example Provisioning Flow

Step 1: The UE sends a SIP REGISTER request to the P-CSCF to register the IMPU associated the UE. In support of DM Notification over OTA-SIP Push, the REGISTER message 'Contact' header includes the following feature tags:

- OMA Push IMC Communication Service Identifier (ICSI):

  `+g.3gpp.app_ref="urn%3Aurn-xxx%3A3gpp-service.ims.icsi.omapush"`

- DM Notification push application identifier:

  `+g.oma.pusheventapp="x-wap-application:syncml.dm"`

- Indication of MESSAGE method support (because MESSAGE will be used for SIP Push):

  `methods="MESSAGE"`

Step 2: If the Provisioning Server has been configured in the Filter Criteria provided by the HSS and executed by the S-CSCF as an AS that is to receive third-party REGISTER, it will receive a copy of the REGISTER request for each matching service profile in the implicit registration set. This will make it aware of the registered public identity from the UE, and the support for DM notification over SIP Push.

Any additional public identities in a single service profile will not be communicated in this REGISTER message. The DM Server must subscribe to the 'reg' event package for the service profile to receive information about the whole implicit registration set for a service profile.

Step 3: The Provisioning Server sends a MESSAGE request to the UE. This includes indication of the DM Notification push application in the 'Accept-Contact' header:

`Accept-Contact:*;+g.oma.pusheventapp="x-wap-application:syncml.dm"`

The 'Content-Type' in the MESSAGE body corresponds to the OMA Push content-type:

```
Content-Type: application/vnd.oma.push
```

Within the OMA Push content, additional headers defined by OMA Push describe the inner content. In this case, the inner content corresponds to DM Notification:

```
Content-Type: application/vnd.syncml.notification
```

The DM Notification content is a binary content type. It contains an MD5 digest authenticator, employing the server-identifier and server password, as well as a session identifier and server identifier. The UE uses the server-identifier in the DM Notification message to locate the DMAcc entry corresponding to the notifying server, and then uses the password (AAuthSecret) and nonce (AAuthData) information in the DMAcc entry to validate the notification message.

Step 4: The UE initiates an HTTP or HTTPS connection to the URI or IP address specified in the DMAcc MO for the requesting server, using the interface connectivity information referenced by the <X>/ToConRef subtrees of the DMAcc entry. Depending on the authentication requirements of the DMAcc entry for this DM server, the UE may provide transport-level client authentication, and may perform server authentication at the transport level.

At the HTTP level, the UE sends an HTTP POST to the path specified in the DMAcc URI. The Request URI of the POST may be '*' if the DMAcc entry does not provide a URI path for the DM server. The body of the POST contains a SyncML message, using XML or WBXML, with the content-type set appropriately. This first HTTP message contains Package #1 according to [OMA-DMPRO]. It identifies the client to the server, provides the DevInfo MO content describing the client, identifies whether the session is client- or server-initiated, and informs the server of any client-generated alert.

The sample below shows most of a typical Package #1 for a server-initiated session. In this case it contains OMA DM application-level client authentication (the <Cred> tag in the SyncML content).

```
POST ./mgmt-server HTTP/1.1
Host: http://www.operator.net
Accept: application/vnd.syncml.dm+xml
Accept-Charset: utf-8
Accept-Encodings: chunked
Accept-Language: en-US
Content-Type: application/vnd.syncml.dm+xml; charset="utf-8"
Content-Length: XXXX
User-Agent: Vendor.Example.Net ZZZ-Device v3.4
Cache-Control: no-store
Transfer-Encoding: chunked

<SyncML xmlns='SYNCML:SYNCML1.2'>
  <SyncHdr>
    <VerDTD>1.2</VerDTD>
    <VerProto>DM/1.2</VerProto>
    <SessionID>1</SessionID>
    <MsgID>1</MsgID>
    <Target><LocURI>http://www.operator.net/mgmt-server</LocURI></Target>
    <Source><LocURI>sip:device-aor@operator.net</LocURI></Source>
    <Cred>  <!-- Client credentials -->
      <Meta>
      <Type xmlns='syncml:metinf'>syncml:auth-basic</Type>
      <Format xmlns='syncml:metinf'>b64</Format>
      </Meta>
      <Data> <!-- base64 formatting of userid:password --> </Data>
    </Cred>
  </SyncHdr>
  <SyncBody>
    <Alert>
      <CmdID>1</CmdID>
      <Data>1200</Data> <!-- Server-initiated session -->
    </Alert>
    <Replace>
```

```
        <CmdID>3</CmdID>
        <Item>
          <Source><LocURI>./DevInfo/DevID</LocURI></Source>
          <Meta>
            <Type xmlns='syncml:metinf'>chr</Type>
            <Format xmlns='syncml:metinf'>text/plain</Format>
          </Meta>
          <Data>sip:device-aor@operator.net</Data>
        </Item>
        <Item>
          <Source><LocURI>./DevInfo/Man</LocURI></Source>
          <Meta>
            <Type xmlns='syncml:metinf'>chr</Type>
            <Format xmlns='syncml:metinf'>text/plain</Format>
          </Meta>
          <Data>Vendor.Example.Net, Inc.</Data>
        </Item>
        <Item>
          <Source><LocURI>./DevInfo/Mod</LocURI></Source>
          <Meta>
            <Type xmlns='syncml:metinf'>chr</Type>
            <Format xmlns='syncml:metinf'>text/plain</Format>
          </Meta>
          <Data>ZZZ Device</Data>
        </Item>

        ...

      </Replace>
      <Final/>
    </SyncBody>
</SyncML>
```

Step 5: The DM Server responds to the UE with an HTTP 200 OK message, which contains DM Package #2. This message provides server identity and will optionally authenticate the server to the client if this was not done at the transport level. If the client has not yet been authenticated, and did not offer credentials in Package #1, then Package #2 may challenge the client for authentication, which causes the client to revert to sending Package #1 again with credentials.

The example below shows the DM server providing its credentials to authenticate to the client, acknowledging the successful authentication of the client to the server, acknowledging the client Alert that indicated that the session was server-initiated, and acknowledging the client's Replace command that provided DevInfo. Following the Status elements that contain these various acknowledgements there is a Replace and a Get command to modify and retrieve some (invented) nodes.

```
HTTP/1.1 200 OK
Content-Type: application/vnd.syncml.dm+xml; charset="utf-8"
Content-Length: XXXX

<SyncML xmlns='SYNCML:SYNCML1.2'>
  <SyncHdr>
    <VerDTD>1.2</VerDTD>
    <VerProto>DM/1.2</VerProto>
    <SessionID>1</SessionID>
    <MsgID>1</MsgID>
    <Target><LocURI>sip:device-aor@operator.net</LocURI></Target>
    <Source><LocURI>http://www.operator.net/mgmt-server</LocURI></Source>
    <Cred> <!-- Server credentials -->
      <Meta>
        <Type xmlns="syncml:metinf">syncml:auth-basic</Type>
        <Format xmlns='syncml:metinf'>b64</Format>
      </Meta>
      <Data> <!-- base64 formatting of userid:password --> </Data>
    </Cred>
```

```
  </SyncHdr>
  <SyncBody>
    <Status>
      <MsgRef>1</MsgRef><CmdRef>0</CmdRef>
      <Cmd>SyncHdr</Cmd>
      <CmdID>6</CmdID>
      <TargetRef>http://www.operator.net/mgmt-server</TargetRef>
      <SourceRef>sip:device-aor@operator.net</SourceRef>
      <!-- Authenticated for the session -->
      <Data>212</Data>
    </Status>
    <Status>
      <MsgRef>1</MsgRef><CmdRef>1</CmdRef>
      <CmdID>7</CmdID>
      <Cmd>Alert</Cmd>
      <Data>200</Data><!-- Alert OK -->
    </Status>
    <Status>
      <MsgRef>1</MsgRef><CmdRef>3</CmdRef>
      <CmdID>8</CmdID>
      <Cmd>Replace</Cmd>
      <Data>200</Data><!-- Replace OK -->
    </Status>
    <Replace>
      <CmdID>2</CmdID>
      <Item>
        <Source><LocURI>./XXX/YY/Count</LocURI></Source>
        <Meta>
          <Type xmlns="syncml:metinf">int</Type>
          <Format xmlns="syncml:metinf">text/plain</Format>
        </Meta>
        <Data>10</Data>
      </Item>
    </Replace>
    <Get>
      <CmdID>4</CmdID>
      <Item>
        <Target><LocURI>./XX/YY/Status</LocURI></Target>
      </Item>
    </Get>
    <Final/>
  </SyncBody>
</SyncML>
```

Step 6: The UE responds to the DM Server with a new HTTP POST request containing DM Package #3. This contains the status and results of any commands sent in Package #2, and could also contain some new client-generated alert.

In this example, the UE communicates successful server authentication and successful status to the Replace and Get commands. The UE also returns the result of the Get.

```
POST ./mgmt-server HTTP/1.1
Host: http://www.operator.net
Accept: application/vnd.syncml.dm+xml
Accept-Charset: utf-8
Accept-Encodings: chunked
Accept-Language: en-US
Content-Type: application/vnd.syncml.dm+xml; charset="utf-8"
Content-Length: XXXX
User-Agent: Vendor.Example.Net ZZZ-Device v3.4
Cache-Control: no-store
Transfer-Encoding: chunked

<SyncML xmlns='SYNCML:SYNCML1.2'>
  <SyncHdr>
```

```
      <VerDTD>1.2</VerDTD>
      <VerProto>DM/1.2</VerProto>
      <SessionID>1</SessionID>
      <MsgID>2</MsgID>
      <Target><LocURI>http://www.operator.net/mgmt-server</LocURI></Target>
      <Source><LocURI>sip:device-aor@operator.net</LocURI></Source>
    </SyncHdr>
    <SyncBody>
      <Status>
        <MsgRef>1</MsgRef>
        <CmdID>1</CmdID>
        <CmdRef>0</CmdRef>
        <Cmd>SyncHdr</Cmd>
        <Data>212</Data> <!-- SyncHdr accepted -->
      </Status>
      <Status>
        <MsgRef>1</MsgRef>
        <CmdRef>2</CmdRef>
        <CmdID>2</CmdID>
        <Cmd>Replace</Cmd>
        <TargetRef>>./XX/YY/Count</TargetRef>
        <Data>200</Data> <!-- Replace OK -->
      </Status>
      <Status>
        <MsgRef>1</MsgRef>
        <CmdRef>4</CmdRef>
        <CmdID>3</CmdID>
        <Cmd>Get</Cmd>
        <TargetRef>>./XX/YY/Status</TargetRef>
        <Data>200</Data> <!-- Get OK -->
      </Status>
      <!-- Results for the Get -->
      <Results>
        <MsgRef>1</MsgRef>
        <CmdRef>4</CmdRef>
        <CmdID>4</CmdID>
        <Item>
          <Source><LocURI>./XX/YY/Status</LocURI></Source>
          <Data>operational</Data>
        </Item>
      </Results>
      <Final/>
    </SyncBody>
  </SyncML>
```

Step 7: The DM Server responds to the UE with another HTTP 200 OK response. This response contains DM Package #4. This could contain further server command for the client. A series of exchanges of Package #4 and Package #3 can continue until the server is finished issuing commands and decides to close the session.

In this example, the server chooses to terminate the session because it has no further actions to take.

```
HTTP/1.1 200 OK
Content-Type: application/vnd.syncml.dm+xml; charset="utf-8"
Content-Length: XXXX

<SyncML xmlns='SYNCML:SYNCML1.2'>
  <SyncHdr>
    <VerDTD>1.2</VerDTD>
    <VerProto>DM/1.2</VerProto>
    <SessionID>1</SessionID>
    <MsgID>2</MsgID>
    <Target><LocURI>sip:device-aor@operator.net</LocURI></Target>
    <Source><LocURI>http://www.operator.net/mgmt-server</LocURI></Source>
  </SyncHdr>
  <SyncBody>
    <Status>
      <MsgRef>2</MsgRef><CmdRef>0</CmdRef>
      <Cmd>SyncHdr</Cmd>
      <CmdID>1</CmdID>
      <Data>200</Data>
    </Status>
    <Final/>
  </SyncBody>
<SyncML>
```

Appendix II Mapping of Event Severity for PacketCable and OMA DM (Informative)

The OMA-DM Generic Alert uses Severities defined by [OMA-DMPRO]. PacketCable specifications use Severities originally defined in the syslog specification. Use this section to explain how Legacy Events may be mapped to Generic Alert events. The Event Level is used to calculate the syslog message priority value in Section 6.8.3.1.5. Note that this mapping is equivalent, preserving the priority of prior versions of the PacketCable specifications and this specification.

Table 10 - OMA DM and PacketCable Severity Mappings

Event Level	OMA-DM Severity	Legacy Severity
0	fatal	emergency
1	alert	alert
2	critical	critical
3	minor	error
4	warning	warning
5	informational	info
6	harmless	notice
7	indeterminate	debug

Appendix III Acknowledgements

CableLabs wishes to thank the PacketCable Provisioning focus team participants for various contributions and efforts that led to the development of this specification. Specifically, the following individuals are thanked for their direct contributions (alphabetical by company name):

Eugene Nechamkin (Broadcom)

Thomas Clack (Broadcom)

Lakshmi Raman (CableLabs)

Sumanth Channabasappa (CableLabs)

Josh Littlefield (Cisco)

Donald Joong (Ericsson)

Mark Trayer (Samsung)

Special thanks are extended to Josh Littlefield for being the primary author of this specification.

Eduardo Cardona and the PacketCable Architects, CableLabs, Inc

Appendix IV Revision History

The following Engineering Change Notice was incorporated in PKT-SP-UE-PROV-I02-100527

ECN	ECN Date	Summary
UE-PROV-N-09.0622-4	4/5/2010	Provisioning Requirements for AS, HSS Provisioning Interface and UE OMA-D updated References