

# **Attachment 4-2-16**

## **WiMAX Forum<sup>®</sup> Network Architecture**

### **Universal Service Interface (USI)**

WiMAXAn Architecture for Internet+ Service Model

**WMF-T33-115-R015v01**

**Note:** This Document is reproduced without any modification with the consent of the WiMAX Forum®, which owns the copyright in them.





# **WiMAX Forum<sup>®</sup> Network Architecture**

Universal Services Interface (USI)

An Architecture for Internet+ Service Model

**WMF-T33-115-R015v01**

WiMAX Forum<sup>®</sup> Approved  
(2009-11-21)

**WiMAX Forum Proprietary**

**Copyright © 2007-2009 WiMAX Forum. All Rights Reserved.**

USI

**Copyright Notice, Use Restrictions, Disclaimer, and Limitation of Liability.**

Copyright 2007-2009 WiMAX Forum. All rights reserved.

The WiMAX Forum® owns the copyright in this document and reserves all rights herein. This document is available for download from the WiMAX Forum and may be duplicated for internal use, provided that all copies contain all proprietary notices and disclaimers included herein. Except for the foregoing, this document may not be duplicated, in whole or in part, or distributed without the express written authorization of the WiMAX Forum.

Use of this document is subject to the disclaimers and limitations described below. Use of this document constitutes acceptance of the following terms and conditions:

**THIS DOCUMENT IS PROVIDED “AS IS” AND WITHOUT WARRANTY OF ANY KIND. TO THE GREATEST EXTENT PERMITTED BY LAW, THE WiMAX FORUM DISCLAIMS ALL EXPRESS, IMPLIED AND STATUTORY WARRANTIES, INCLUDING, WITHOUT LIMITATION, THE IMPLIED WARRANTIES OF TITLE, NONINFRINGEMENT, MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE WiMAX FORUM DOES NOT WARRANT THAT THIS DOCUMENT IS COMPLETE OR WITHOUT ERROR AND DISCLAIMS ANY WARRANTIES TO THE CONTRARY.**

Any products or services provided using technology described in or implemented in connection with this document may be subject to various regulatory controls under the laws and regulations of various governments worldwide. The user is solely responsible for the compliance of its products and/or services with any such laws and regulations and for obtaining any and all required authorizations, permits, or licenses for its products and/or services as a result of such regulations within the applicable jurisdiction.

**NOTHING IN THIS DOCUMENT CREATES ANY WARRANTIES WHATSOEVER REGARDING THE APPLICABILITY OR NON-APPLICABILITY OF ANY SUCH LAWS OR REGULATIONS OR THE SUITABILITY OR NON-SUITABILITY OF ANY SUCH PRODUCT OR SERVICE FOR USE IN ANY JURISDICTION.**

**NOTHING IN THIS DOCUMENT CREATES ANY WARRANTIES WHATSOEVER REGARDING THE SUITABILITY OR NON-SUITABILITY OF A PRODUCT OR A SERVICE FOR CERTIFICATION UNDER ANY CERTIFICATION PROGRAM OF THE WiMAX FORUM OR ANY THIRD PARTY.**

The WiMAX Forum has not investigated or made an independent determination regarding title or noninfringement of any technologies that may be incorporated, described or referenced in this document. Use of this document or implementation of any technologies described or referenced herein may therefore infringe undisclosed third-party patent rights or other intellectual property rights. The user is solely responsible for making all assessments relating to title and noninfringement of any technology, standard, or specification referenced in this document and for obtaining appropriate authorization to use such technologies, technologies, standards, and specifications, including through the payment of any required license fees.

**NOTHING IN THIS DOCUMENT CREATES ANY WARRANTIES OF TITLE OR NONINFRINGEMENT WITH RESPECT TO ANY TECHNOLOGIES, STANDARDS OR SPECIFICATIONS REFERENCED OR INCORPORATED INTO THIS DOCUMENT.**

**IN NO EVENT SHALL THE WiMAX FORUM OR ANY MEMBER BE LIABLE TO THE USER OR TO A THIRD PARTY FOR ANY CLAIM ARISING FROM OR RELATING TO THE USE OF THIS DOCUMENT, INCLUDING, WITHOUT LIMITATION, A CLAIM THAT SUCH USE INFRINGES A THIRD PARTY’S INTELLECTUAL PROPERTY RIGHTS OR THAT IT FAILS TO COMPLY WITH APPLICABLE LAWS OR REGULATIONS. BY USE OF THIS DOCUMENT, THE USER WAIVES ANY SUCH CLAIM AGAINST THE WiMAX FORUM AND ITS MEMBERS RELATING TO THE USE OF THIS DOCUMENT.**

The WiMAX Forum reserves the right to modify or amend this document without notice and in its sole discretion. The user is solely responsible for determining whether this document has been superseded by a later version or a different document.

“WiMAX,” “Mobile WiMAX,” “Fixed WiMAX,” “WiMAX Forum,” “WiMAX Certified,” “WiMAX Forum Certified,” the WiMAX Forum logo and the WiMAX Forum Certified logo are trademarks of the WiMAX Forum. Third-party trademarks contained in this document are the property of their respective owners.

# TABLE OF CONTENTS

<b>REVISION HISTORY .....</b>	<b>10</b>
<b>1. DOCUMENT SCOPE .....</b>	<b>11</b>
<b>2. ABBREVIATIONS AND DEFINITIONS .....</b>	<b>12</b>
2.1 ABBREVIATIONS .....	12
2.2 TERMS & DEFINITIONS .....	12
<b>3. REFERENCES .....</b>	<b>13</b>
<b>4. USI REQUIREMENTS AND PRINCIPLES .....</b>	<b>14</b>
<b>5. USI NETWORK REFERENCE MODEL .....</b>	<b>15</b>
5.1 NON-ROAMING ARCHITECTURE .....	15
5.2 ROAMING ARCHITECTURE .....	17
5.2.1 Local USI Services .....	18
5.2.2 Home USI Services .....	18
<b>6. USI PROCEDURES .....</b>	<b>20</b>
6.1 USI SERVICE FLOWS (NON-ROAMING) .....	20
6.1.1 Direct USI service flow .....	20
6.1.2 Redirected USI Service Flow .....	21
6.2 USI SERVICE FLOWS (ROAMING) .....	22
6.2.1 Home USI services .....	22
6.2.2 Local USI service flows .....	23
6.3 USI IDENTITY .....	24
6.3.1 Long-Lived USI Identity .....	24
6.3.2 Short-Lived USI Identity .....	24
6.3.3 Usage of USI Identities .....	25
6.3.3.1 USI System discovery and User/MS's identification by iASP .....	25
6.3.4 USI Identity Allocation & Renewal .....	26
6.3.4.1 Direct Flow .....	26
6.3.4.2 MS/User Redirected Flow .....	26
6.4 USI REGISTRATION .....	27
6.4.1 USI Registration (Non-Roaming) .....	27
6.4.2 USI Registration (Roaming) .....	29
6.5 USI DEREGISTRATION .....	33
6.6 DISCOVERY PROCEDURES .....	34
6.6.1 Discovery of USI support in a CSN .....	34
6.6.2 Discovery (by iASP) of USI System Serving a User/MS .....	34
<b>7. USI SERVICES .....</b>	<b>37</b>
7.1 LOCATION RETRIEVAL .....	37
7.1.1 Determining MS Location in Non-Roaming Scenario .....	37
7.1.2 Determining MS Location in Roaming Scenarios .....	37
7.2 MS STATUS .....	39
7.2.1 Status Context .....	39
7.2.2 Determining MS Status in non-Roaming Scenario .....	40
7.2.3 Determining MS Status in Roaming Scenarios .....	41
7.3 IP ADDRESS DISCOVERY .....	42
7.3.1 IP Address Discovery in non-Roaming Scenario .....	42
7.3.2 IP Address Discovery in Roaming Scenarios .....	43
7.4 DYNAMIC MS CAPABILITY RETRIEVAL .....	44
7.4.1 Dynamic Capability Retrieval in non-Roaming Scenario .....	44
7.4.2 Dynamic Capability Retrieval in Roaming Scenarios .....	45

## USI

1	7.5	DYNAMIC QoS SUPPORT .....	47
2	7.5.1	Web Service Operations provided by USI System.....	47
3	7.5.1.1	USI QoS Session Creation .....	47
4	7.5.1.2	USI QoS Session Modification .....	48
5	7.5.1.3	USI QoS Session Termination .....	49
6	7.5.1.4	USI QoS Session Query .....	50
7	7.5.1.5	USI QoS Start Notification .....	52
8	7.5.1.6	USI QoS Stop Notification.....	52
9	7.5.2	Web Service Operations provided by iASP.....	53
10	7.5.2.1	Application Session Query.....	53
11	7.5.2.2	Event Notification .....	54
12	7.6	E-PAYMENT.....	56
13	7.7	LIVE STREAMING.....	57
14	7.7.1	Live Streaming Service types .....	57
15	7.7.2	Motivation for Live Streaming Delivery through USI.....	57
16	7.7.3	Live Streaming Interactions on U1 .....	57
17	7.7.4	Live Streaming Control on U1-Control .....	58
18	7.7.5	MS joining live streaming session.....	59
19	7.7.5.1	USI server has the context of the live streaming session from iASP.....	59
20	7.7.5.2	USI server does not have the context of the live streaming session from iASP .....	60
21	7.8	CONTENT PUSH .....	60
22	7.8.1	Content Push in Non-Roaming Scenario .....	60
23	7.8.2	Content Push in Roaming Scenarios.....	61
24	7.8.3	Direct MS-to-MS Content Push .....	63
25	7.8.4	Transport of Content from USI server to MS.....	64
26	8.	SECURITY ASPECTS.....	65
27	8.1	iASP AUTHENTICATION DURING PROCESSING OF USI REQUEST .....	65
28	8.1.1	iASP authentication and USI request authorization using digital signature .....	65
29	8.1.2	ASP authentication using bearer authentication .....	65
30	8.2	IDENTIFICATION AND AUTHENTICATION OF USER/MS .....	66
31	8.2.1	Methods of authentication .....	67
32	8.2.1.1	Identification and authentication by IP address.....	67
33	8.2.1.2	Identification and authentication using MS side USI certificate .....	67
34	8.2.1.3	Identification and authentication using MS/User identity and password.....	68
35	8.3	USI SECURITY TOKENS .....	68
36	8.3.1	Use of <AuthnRequest> issued by iASP to USI System.....	68
37	8.3.2	Use of <AuthnStatement> issued by USI System to iASP.....	69
38	8.3.3	Use of <AuthzDecisionStatement> issued by USI System.....	70
39	8.3.4	Use of <AuthzDecisionStatement> Issued by iASP to USI System.....	70
40	8.3.5	Use of <AuthzDecisionQuery> issued by iASP to USI System.....	70
41	9.	ACCOUNTING ASPECTS.....	71
42	9.1	REQUIREMENTS AND PRINCIPLES .....	71
43	9.2	USI ACCOUNTING PROCEDURES .....	71
44	9.3	CHARGING CORRELATION .....	72
45	10.	INTERFACE SPECIFICATIONS.....	73
46	10.1	GENERAL DESCRIPTION.....	73
47	10.1.1	SOAP Transport binding .....	73
48	10.1.1.1	USI Direct service flow .....	73
49	10.1.1.2	USI Redirect service flow.....	74
50	10.1.1.3	Error handling and error codes .....	75
51	10.2	DYNAMIC QoS .....	75
52	10.2.1	Parameter Definition .....	76
53	10.2.1.1	qoSSessionID .....	76
54	10.2.1.2	endUserID.....	76
55	10.2.1.3	endUserIPAddress .....	76

## USI

1	10.2.1.4	applicationChargingID .....	76
2	10.2.1.5	flowNumber .....	76
3	10.2.1.6	mediaType .....	76
4	10.2.1.7	codecData .....	76
5	10.2.1.8	qoSClass .....	77
6	10.2.1.9	reservationPriority .....	77
7	10.2.1.10	result .....	77
8	10.2.1.11	faultCode .....	77
9	10.2.1.12	qoSEvent .....	77
10	10.2.2	Type Definition .....	78
11	10.2.2.1	QoSFlowInfo Structure.....	78
12	10.2.2.2	FlowDescription Structure.....	78
13	10.2.2.3	QoSInformation Structure.....	79
14	10.2.2.4	FailedFlowInfo Structure.....	79
15	10.2.3	Message Definition for Web Service Operations .....	79
16	10.2.3.1	Web Service Operations provided by USI System .....	79
17	10.2.3.1.1	createQoSSession Request.....	79
18	10.2.3.1.2	createQoSSession Response .....	80
19	10.2.3.1.3	modifyQoSSession Request.....	80
20	10.2.3.1.4	modifyQoSSession Response .....	80
21	10.2.3.1.5	terminateQoSSession Request .....	81
22	10.2.3.1.6	terminateQoSSession Response.....	81
23	10.2.3.1.7	queryQoSSession Request .....	81
24	10.2.3.1.8	queryQoSSession Response.....	81
25	10.2.3.1.9	startQoSEventNotification Request .....	81
26	10.2.3.1.10	startQoSEventNotification Response.....	82
27	10.2.3.1.11	stopQoSEventNotification Request .....	82
28	10.2.3.1.12	stopQoSEventNotification Response.....	82
29	10.2.3.2	Web Service Operations provided by iASP .....	82
30	10.2.3.2.1	queryApplicationSession Request .....	82
31	10.2.3.2.2	queryApplicationSession Response .....	82
32	10.2.3.2.3	notifyQoSEvent Request .....	82
33	10.2.3.2.4	notifyQoSEvent Response .....	83
34	10.3	USI ACCOUNTING .....	83
35	10.3.1	Parameter Definition .....	83
36	10.3.1.1	aspChargingID.....	83
37	10.3.1.2	aspID .....	83
38	10.3.1.3	serviceType.....	83
39	10.3.1.4	acctRecordType .....	84
40	10.3.1.5	acctRecordNumber .....	84
41	10.3.1.6	acctSessionID .....	84
42	10.3.1.7	eventTimeStamp.....	84
43	10.3.1.8	nasIPAddress .....	84
44	10.3.1.9	interimCause .....	84
45	10.3.1.10	terminationCause .....	85
46	10.3.1.11	uplinkOctets.....	85
47	10.3.1.12	uplinkPackets.....	85
48	10.3.1.13	downlinkOctets .....	85
49	10.3.1.14	downlinkPackets.....	85
50	10.3.1.15	acctSessionTime .....	85
51	10.3.1.16	activeTime .....	85
52	10.3.1.17	msMACAddress .....	85
53	10.3.1.18	msIPAddress .....	85
54	10.3.1.19	sessionContinue .....	85
55	10.3.1.20	beginningOfSession.....	85
56	10.3.1.21	idleModeTransition .....	85
57	10.3.1.22	accessChargingID.....	85
58	10.3.1.23	napID .....	86
59	10.3.1.24	nspID .....	86
60	10.3.1.25	bsID .....	86
61	10.3.1.26	direction.....	86
62	10.3.1.27	result .....	86

## USI

1	10.3.1.28	faultCode .....	86
2	10.3.2	Type Definition .....	86
3	10.3.2.1	USIID Structure .....	86
4	10.3.2.2	GrantedQoS Structure .....	86
5	10.3.2.3	Flows Structure .....	88
6	10.3.2.4	AFCorrelationInformation Structure .....	88
7	10.3.3	Message Definition for Web Service Operations .....	89
8	10.3.3.1	Web Service Operations provided by iASP .....	89
9	10.3.3.1.1	notifyAccounting Request .....	89
10	10.3.3.1.2	notifyAccounting Response .....	90
11	10.4	E-PAYMENT SERVICE INTERFACE .....	90
12	10.4.1	E-Payment Authorization Request .....	90
13	10.4.1.1	<Subject> .....	91
14	10.4.1.2	<Action> .....	91
15	10.4.2	E-Payment Authorization Response .....	91
16	10.4.2.1	<Resource> .....	92
17	10.4.3	E-Payment Charge Request .....	92
18	10.4.3.1	<Evidence> .....	92
19	10.4.3.2	<Action> .....	92
20	10.4.4	E-Payment Charge Response .....	92
21	10.4.4.1	<Evidence> .....	92
22	10.4.5	E-Payment Structures .....	93
23	10.4.5.1	E-Payment Transaction Details .....	93
24	10.4.5.2	E-Payment Order Item .....	93
25	10.4.5.3	E-Payment Money .....	93
26	10.5	USER/MS ATTRIBUTES QUERY SERVICE .....	94
27	10.5.1	User/MS Attribute Request .....	94
28	10.5.1.1	<Attribute> .....	94
29	10.5.2	User/MS Attribute Response .....	94
30	10.5.2.1	<AttributeStatement> .....	94
31	10.5.2.2	<EncryptedAttribute> .....	95
32	10.5.2.3	<Attribute> .....	95
33	10.5.2.4	<AttributeValue> .....	95
34	10.5.2.5	Geo-Location Attribute .....	96
35	10.5.2.6	Civic-Location Attribute .....	96
36	10.5.2.7	WiMAX Location Measurements .....	96
37	10.5.2.8	Terminal Capabilities Attribute .....	96
38	10.5.2.9	MS Status Attribute .....	97
39	10.5.2.10	Personal Info Attribute .....	98
40	10.5.2.11	User Address Attribute .....	98
41	10.5.2.12	Medical Info. Attribute .....	99
42	10.5.2.13	ContactInfo Structure .....	100
43	10.6	CONTENT PUSH SERVICE .....	100
44	10.6.1	<ContentPushRequest> Message .....	100
45	10.7	LIVE STREAMING SERVICE .....	102
46	10.7.1	<LiveStreamingServiceGuide> structure .....	102
47	10.7.2	<LiveStreamingSecurityContext> structure .....	102
48	10.7.3	<LiveStreamingQoSContext> strcuture .....	103
49	10.8	SOAP HEADERS .....	103
50	10.8.1	Type Definition .....	103
51	10.8.1.1	USIRequestSecurityHeader Structure .....	103
52	10.8.1.2	USIResponseSecurityHeader Structure .....	103
53	10.9	SERVE MULTIPLE MESSAGES IN A SINGLE WEBSERVICE OPERATION .....	105
54	10.9.1	Parameter Definition .....	105
55	10.9.1.1	messageId .....	105
56	10.9.1.2	ipAddress .....	105
57	10.9.1.3	macAddress .....	105
58	10.9.1.4	commandId .....	106
59	10.9.1.5	timeStamp .....	106
60	10.9.2	Type Definition .....	106



## USI

1	10.9.2.1	wimaxUSIOperatorHeaderType Structure.....	106
2	10.9.2.2	wimaxUSIOperatorRequestType Structure .....	106
3	10.9.2.3	wimaxUSIOperatorRequestBodyType Structure .....	107
4	10.9.2.4	wimaxUSIOperatorRequestMessageType Structure.....	107
5	10.9.2.5	wimaxUSIOperatorResponseType Structure .....	108
6	10.9.2.6	wimaxUSIOperatorResponseBodyType Structure .....	109
7	10.9.2.7	wimaxUSIOperatorResponsetMessageType Structure .....	109
8	10.9.3	<i>Message Definition for Web Service Operations</i> .....	109
9	10.9.3.1	Web Service Operations provided by USI System .....	109
10	10.9.3.1.1	ProcessWimaxUSIOperator Request .....	109
11	10.9.3.1.2	ProcessWimaxUSIOperator Response.....	109
12	10.9.4	<i>iASP Type Definitions</i> .....	109
13	10.9.4.1	wimaxUSIAspHeaderType .....	109
14	10.9.4.2	wimaxUSIAspRequestType Structure .....	110
15	10.9.4.3	wimaxUSIAspRequestBodyType Structure.....	110
16	10.9.4.4	wimaxUSIAspRequestMessageType Structure .....	110
17	10.9.4.5	wimaxUSIAspResponseType Structure.....	110
18	10.9.4.6	wimaxUSIAspResponseBodyType Structure .....	111
19	10.9.4.7	wimaxUSIAspResponsetMessageType Structure.....	111
20	10.9.5	<i>Message Definition for iASP Web Service Operations</i> .....	111
21	10.9.5.1	Web Service Operations provided by iASP System .....	111
22	10.9.5.1.1	ProcessWimaxUSIAspRequest.....	111
23	10.9.5.1.2	ProcessWimaxUSIAsp Response .....	112
24	<b>11.</b>	<b>LAWFUL INTERCEPT .....</b>	<b>113</b>
25	<b>ANNEX A (INFORMATIVE): INTERWORKING WITH DM SERVER.....</b>		<b>114</b>
26	<b>ANNEX B (INFORMATIVE): INTERWORKING WITH POLICY &amp; CHARGING CONTROL (PCC)....</b>		<b>115</b>
27	<b>ANNEX C (INFORMATIVE): COMMERCIAL VOIP SUPPORT USING USI .....</b>		<b>116</b>
28	C.1.	KEY ASPECTS OF COMMERCIAL VOIP USING USI .....	116
29	C.2.	MS-ORIGINATED VOIP CALL ESTABLISHMENT .....	117
30	C.3.	MS-TERMINATED VOIP CALL ESTABLISHMENT .....	118
31	C.3	HANDOVER SUPPORT .....	119
32	C.4.	ACCOUNTING SUPPORT.....	119
33	C.5.	EMERGENCY SERVICES .....	119
34	<b>ANNEX D (INFORMATIVE): INTERWORKING WITH LOCATION BASED SERVICES (LBS).....</b>		<b>122</b>
35		<i>Determining MS Location in Non-Roaming Scenario .....</i>	<i>122</i>
36		<i>Determining MS Location in Roaming Scenarios.....</i>	<i>123</i>
37			
38			

## LIST OF FIGURES

FIGURE 1: USI NETWORK REFERENCE MODEL (NON-ROAMING).....	15
FIGURE 2: MAIN FUNCTIONALITIES OF THE USI SYSTEM.....	16
FIGURE 3: USI NETWORK REFERENCE MODEL (ROAMING) .....	17
FIGURE 4: LOCAL USI SERVICES IN ROAMING .....	18
FIGURE 5: HOME USI SERVICES IN ROAMING .....	19
FIGURE 6: DIRECT USI SERVICE FLOW .....	20
FIGURE 7: USER/MS REDIRECTED USI SERVICE FLOW .....	21
FIGURE 8: HOME USI DIRECT USI SERVICEFLOW .....	22
FIGURE 9: LOCAL AND HOME USI SERVICE FLOW (V-USI IS ACCESSED FIRST) .....	23
FIGURE 10: S-ID ALLOCATION OR RETREIVAL.....	26
FIGURE 11: S-ID ALLOCATION OR S-ID/L-ID RETREIVAL IN .....	27
FIGURE 12: USI REGISTRATION (NSP-TRIGGERED) .....	28
FIGURE 13: USI REGISTRATION (IASP AND/OR USER/MS TRIGGERED) .....	29
FIGURE 14: USI REGISTRATION FOR USI ROAMING (H-AAA TRIGGERED) .....	30
FIGURE 15: IASP TRIGGERED USI REGISTRATION FOR USI ROAMING (FOR LOCAL SERVICES) .....	31
FIGURE 16: NSP TRIGGERED LOCAL USI REGISTRATION FOR USI ROAMING .....	32
FIGURE 17: IASP AND/OR MS/USER TRIGGERED USI REGISTRATION FOR USI ROAMING (FOR HOME SERVICES).....	33
FIGURE 18: USI DE-REGISTRATION (NSP-TRIGGERED) .....	34
FIGURE 19: SERVING USI SYSTEM DISCOVERY IN REDIRECT USI SERVICE FLOW .....	35
FIGURE 20: DETERMINING MS LOCATION VIA USI (NON-ROAMING) .....	37
FIGURE 21: DETERMINING MS LOCATION VIA USI (LOCAL USI SERVICES IN ROAMING) .....	38
FIGURE 22: DETERMINING MS LOCATION VIA USI (HOME USI SERVICES IN ROAMING) .....	39
FIGURE 23: DETERMINING MS/USER STATUS VIA USI (NON-ROAMING).....	40
FIGURE 24: DETERMINING MS/USER STATUS VIA USI (LOCAL USI SERVICES IN ROAMING) .....	41
FIGURE 25: MS IP ADDRESS DISCOVERY (NON-ROAMING) .....	42
FIGURE 26: MS IP ADDRESS DISCOVERY (LOCAL USI SERVICES IN ROAMING) .....	43
FIGURE 27: DYNAMIC CAPABILITY RETRIEVAL (NON-ROAMING) .....	44
FIGURE 28: DYNAMIC CAPABILITY RETRIEVAL (LOCAL USI SERVICES IN ROAMING) .....	46
FIGURE 29: USI QoS SESSION CREATION .....	47
FIGURE 30: USI QoS SESSION MODIFICATION .....	49
FIGURE 31: USI QoS SESSION TERMINATION .....	50
FIGURE 32: USI QoS SESSION QUERY .....	51
FIGURE 33: USI QoS START NOTIFICATION.....	52
FIGURE 34: USI QoS STOP NOTIFICATION .....	53
FIGURE 35: APPLICATION SESSION QUERY.....	54
FIGURE 36: EVENT NOTIFICATION.....	55
FIGURE 37: E-PAYMENT PROCEDURE USING USI .....	56
FIGURE 38: LIVE STREAMING CONTROL FLOW ON U1-CONTROL.....	58
FIGURE 39: LIVE STREAMING DATA PATH SETUP ON U1-DATA (USI HAS IASP CONTEXT).....	59
FIGURE 40: LIVE STREAMING PATH SETUP ON U1-DATA (USI DOES NOT HAVE IASP CONTEXT) .....	60
FIGURE 41: CONTENT PUSH FLOW (NON-ROAMING).....	61
FIGURE 42: CONTENT PUSH FLOW (LOCAL USI SERVICES IN ROAMING) .....	62
FIGURE 43: CONTENT PUSH FLOW (HOME USI SERVICES IN ROAMING) .....	63
FIGURE 44: MS-TO-MS CONTENT PUSH FLOW (NON-ROAMING).....	64
FIGURE 45: IASP AUTHENTICATION USING AUTHORIZATION TOKEN .....	65
FIGURE 46: AUTHENTICATION OF USI USER/MS .....	66
FIGURE 47: HOME USI SERVICE FLOW, IN THE DIRECT FLOW .....	68
FIGURE 48: USI ACCOUNTING PROCEDURE.....	71
FIGURE 49: EXAMPLE USI REQUEST .....	73
FIGURE 50: EXAMPLE USI RESPONSE.....	74
FIGURE 51: EXAMPLE USI REQUEST/RESPONSE IN USI REDIRECT SERVICE FLOW.....	75

## LIST OF TABLES

2	TABLE 1: USER/MS IDENTIFICATION AND USI SYSTEM DISCOVERY BY IASP .....	25
3	TABLE 2: VALUES FOR <AUTHNCONTEXT> IN <AUTHNSTATEMENT> .....	67
4	TABLE 3: USE OF <AUTHNREQUEST> PARAMETERS .....	69
5	TABLE 4: NAME IDENTIFIER FORMAT IDENTIFIERS .....	69
6	TABLE 5: USE OF <AUTHNSTATEMENT> STATEMENT AND RELATED <ASSERTION> PARAMETERS .....	69
7	TABLE 6: USE OF < AUTHZDECISIONSTATEMENT> AND RELATED <ASSERTION> PARAMETERS .....	70
8	TABLE 7: USE OF <AUTHZDECISIONSTATEMENT> AND RELATED <ASSERTION> PARAMETERS .....	70
9	TABLE 8: <AUTHZDECISIONQUERY> PARAMETERS IN E-PAYMENT AUTHORIZATION REQUEST .....	91
10	TABLE 9: <ASSERTION> PARAMETERS USED IN E-PAYMENT AUTHORIZATION RESPONSE .....	91
11	TABLE 10: <AUTHZDECISIONSTATEMENT> PARAMETERS USED IN E-PAYMENT AUTHORIZATION RESPONSE .....	91
12	TABLE 11: <AUTHZDECISIONQUERY> PARAMETERS IN E-PAYMENT CHARGE REQUEST .....	92
13	TABLE 12: <EVIDENCE> PARAMETERS IN E-PAYMENT CHARGE REQUEST PARAMETERS .....	92
14	TABLE 13: <EVIDENCE> PARAMETERS IN E-PAYMENT CHARGE RESPONSE .....	93
15	TABLE 14: E-PAYMENT <TRANSACTIONDETAILSTYPE> PARAMETERS .....	93
16	TABLE 15: E-PAYMENT <ORDERITEMTYPE> PARAMETERS .....	93
17	TABLE 16: E-PAYMENT <MONEYTYPE> ATTRIBUTES .....	93
18	TABLE 17: <ATTRIBUTEQUERY> PARAMETERS IN A USER/MS ATTRIBUTE REQUEST .....	94
19	TABLE 18: <ASSERTION> PARAMETERS IN USER/MS ATTRIBUTE RESPONSE .....	94
20	TABLE 19: <ATTRIBUTESTATEMENT> PARAMETERS IN USER/MS ATTRIBUTE RESPONSE .....	95
21	TABLE 20: <ATTRIBUTE> ELEMENT'S TYPE AND CORRESPONDING <ATTRIBUTEVALUE> ELEMENT VALUE .....	95
22	TABLE 21: <ATTRIBUTE> ELEMENT'S TYPE AND CORRESPONDING <ATTRIBUTEVALUE> ELEMENT VALUE .....	96
23	TABLE 22: <ATTRIBUTE> ELEMENT'S TYPE AND CORRESPONDING <ATTRIBUTEVALUE> ELEMENT VALUE .....	97
24	TABLE 23: <ATTRIBUTE> ELEMENT'S TYPE AND CORRESPONDING <ATTRIBUTEVALUE> ELEMENT VALUE .....	98
25	TABLE 24: <ATTRIBUTE> ELEMENT'S TYPE AND CORRESPONDING <ATTRIBUTEVALUE> ELEMENT VALUE .....	98
26	TABLE 25: <ATTRIBUTE> ELEMENT'S TYPE AND CORRESPONDING <ATTRIBUTEVALUE> ELEMENT VALUE .....	99
27	TABLE 26: <ATTRIBUTE> ELEMENT'S TYPE AND CORRESPONDING <ATTRIBUTEVALUE> ELEMENT VALUE .....	99
28	TABLE 27: CONTACTINFO STRUCTURE .....	100
29	TABLE 28: <CONTENTPUSHREQUEST> MESSAGE ELEMENTS .....	100

---

## Revision History

November 6, 2009

Initial version of Release 1.5.

---

## 1. Document Scope

Universal Services Interface (USI) is a framework for specifying required WiMAX network interfaces towards trusted third party iASPs (SLA between SP and iASP MAY be required). These network interfaces SHOULD allow exposure of WiMAX network capabilities and mobile user information between the SP and iASP in a secure and controlled manner. USI is intended:

- To create a solution for service providers to generate additional revenue
- To expose network capability APIs to iASP for dynamic service creation (in a controlled manor for business purposes)
- To reuse network intelligence being built into WiMAX (e.g. LBS, MCBCS, PCC)

Exposing capabilities implemented in a WiMAX NSP covers two areas:

- *Defining the services and interfaces to be exposed* that represents the capabilities of the network to be offered to 3<sup>rd</sup> party-iASPs;
- *Enclosing the exposed services and interfaces in an Exposure Framework* for achieving a secure and controlled exposure (from the WiMAX network provider-SP point of view) for those interfaces

---

## 2. Abbreviations and Definitions

### 2.1 Abbreviations

HTTP: Hyper Text Transport Protocol  
ASP: Application Services Provider  
iASP: Internet Application Services Provider  
SLA: Service Level Agreement  
USI: Universal Services Interface  
WS: Web Services  
XML: Extensible Markup Language

### 2.2 Terms & Definitions

**Content Provider (CP):** The provider that owns the content (e.g. Walt Disney). iASP can be a content provider too

**Internet Advertiser (IA):** A customer of iASP that advertises through iASP (e.g. BMW)

**Internet Application Service Provider (iASP):** The provider that offers applications and aggregated contents on public Internet (e.g. Yahoo!, Google, E-Bay). An iASP MAY have further relationship with content providers and Internet advertisers

**Policy:** A combination of rules that defines how to manage and control access to resources. Policies can be related either to subscriber, iASP, Services or to a combination of them.

**Service Provider (SP):** The provider that operates and sells WiMAX services to the end user

**USI QoS Session:** A USI QoS Session is the QoS session defined per application per MS.

---

### 3. References

- [SPWG1.5] Recommendations and Requirements for Networks based on WiMAX Forum Certified Products – Release 1.5
- [NWGSTG2] WiMAX Forum, T32-001-R015v01, T32-002-R015v01, T32-003-R015v01, T32-004-R015v01, T32-005-R015v01, "Architecture Tenets, Reference Model and Reference Points" Part0 to Part3 and Abbreviations, Release 1.5
- [NWGSTG3] WiMAX Forum, T33-001-R015v01, "Detailed Protocols and Procedures, Base Specification", Release 1.5
- [XML] W3C Extensible Markup Language (XML) 1.0 (Second Edition), W3C Recommendation, Version 6-October-2000.
- [ROHC] WiMAX Forum T33-108-R015v01, "Architecture, detailed Protocols and Procedures, ROust Header Compression (ROHC) Support", Release 1.5
- [LBS] WiMAX Forum T33-110-R015v01, "Protocols and Procedures for Location Based Services", Release 1.5
- [SAMLBind] S. Cantor et al. *Bindings for the OASIS Security Assertion Markup Language (SAML) V2.0*. OASIS SSTC, March 2005. Document ID saml-bindings-2.0-os. See <http://www.oasis-open.org/committees/security/>.
- [SAMLConform] P. Mishra et al. *Conformance Requirements for the OASIS Security Assertion Markup Language (SAML) V2.0*. OASIS SSTC, March 2005. Document ID saml-conformance-2.0-os. See <http://www.oasis-open.org/committees/security/>.
- [SAMLCore] S. Cantor et al. *Assertions and Protocols for the OASIS Security Assertion Markup Language (SAML) V2.0*. OASIS SSTC, March 2005. Document ID saml-core-2.0-os. See <http://www.oasis-open.org/committees/security/>.
- [SAMLAssn-xsd] S. Cantor et al. SAML assertions schema. OASIS SSTC, March 2005. Document ID saml-schema-assertion-2.0. See <http://www.oasisopen.org/committees/security/>.
- [SAMLProt-xsd] S. Cantor et al. SAML protocols schema. OASIS SSTC, March 2005. Document ID saml-schema-protocol-2.0. See <http://www.oasisopen.org/committees/security/>.
- [SOAP11] D. Box et al. Simple Object Access Protocol (SOAP) 1.1. World Wide Web Consortium Note, May 2000. See <http://www.w3.org/TR/2000/NOTE-SOAP-20000508/>.
- [PCC] WiMAX Forum T33-109-R015v01, "Architecture, detailed Protocols and Procedures, Policy and Charging Control", Release 1.5
- [NALI] WiMAX Forum T33-107-R015v01, "Architecture, detailed Protocols and Procedures, WIMAX Lawful Intercept - NORTH AMERICAN REGION", Release 1.5

---

## 4. USI Requirements and Principles

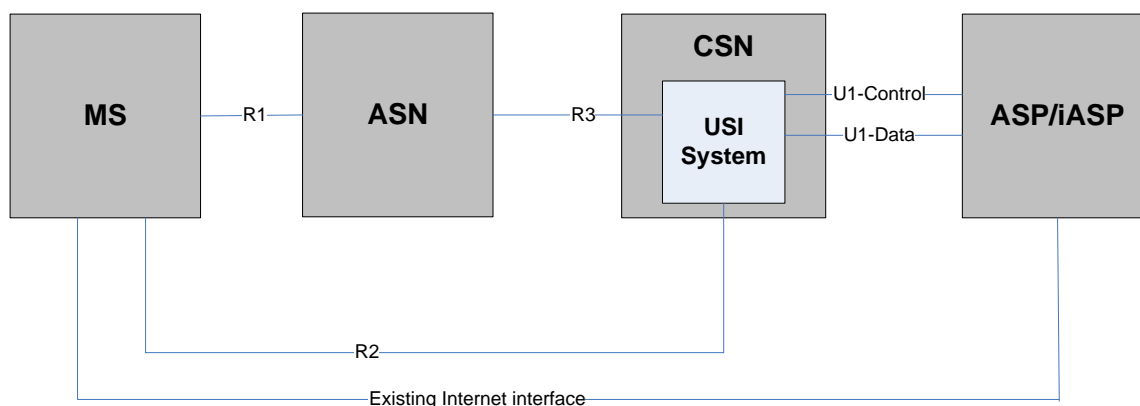
The USI requirements are presented in [SPWG1.5].



## 5. USI Network Reference Model

This section describes the end to end NWG architecture enhancements for USI support. Figure 1 shows an architecture overview of USI enhancement to WiMAX network architecture as defined in [NWGSTG2, NWGSTG3].

### 5.1 Non-Roaming Architecture



**Figure 1: USI Network Reference Model (Non-Roaming)**

As shown above, a new CSN network element called USI System is added to the architecture. USI System interfaces with iASP through the U1 interface. USI System can optionally interface with MCBSC Server, PCRF, AAA server/proxy, and location servers.

The U1 interface carries control traffic (e.g. user information such as location/presence, QoS parameters, and accounting) between the USI System and the iASP. R2 interface carries control messages between MS and the USI System (e.g. USI registration messages)

As shown the interfaces between the MS and the iASP is outside the scope of this specification. It is expected that existing Internet interfaces to be used for this interface.

The USI System makes available the WiMAX web services to 3<sup>rd</sup> party-iASP (either intra or extra domain with respect of the NSP) enriching them by functionality for their control (e.g. security, authentication, authorization, Service Level Agreement, policy management, accounting, end-users privacy, Identity Management).

The figure below depicts, as a graphical representation, which are the main functionalities of the USI System. This functional decomposition is informational and does not mandate any implementation of the USI System.

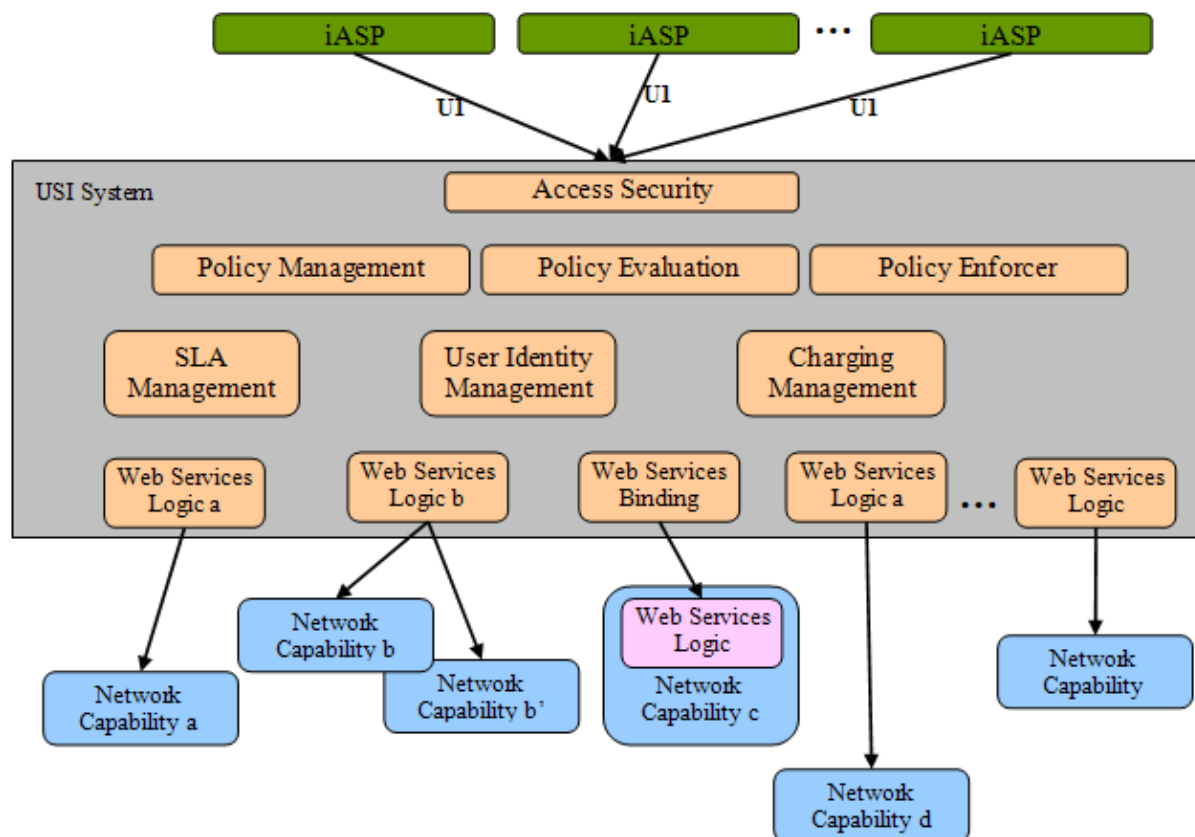
Figure 1 also shows the decomposition of the U1 reference point into U1-Control and U1-Data.

U1-Data interface on the downlink carries the data from the iASP, which is intended for the MS; such data MAY then be delivered by the USI System to the MS by using the unicast or multicast data paths in the NAP and NSP and is out of scope of the USI specification. On the uplink, the U1-Data interface MAY carry data traffic of the MS from the USI to the iASP; the data MAY be delivered from the MS to the USI

## USI

System via the unicast or multicast data paths in the NAP and NSP and is out of scope of the USI specification.

U1-Control interface is meant for carrying all the signaling and control related exchanges between the iASP and the USI System.



**Figure 2: Main functionalities of the USI System**

The interaction between internal USI functionalities is out of scope of this specification.

The *Access Security* component is in charge of protecting the NSP. This topic is specified in Section 5.

When an iASP requests to use WiMAX web services, the USI System applies some rules (i.e. policies) that define how to manage and control access to those WiMAX web services.

- Authentication Policy: the iASP needs to be authenticated (the authentication can be based on e.g. IP address, password, certificate);
- Authorization Policy: the iASP needs to have the authorization to make a specific request (the authorization can be based on e.g. subscription or free-access,)

Examples of policies related to the Services are:

- Service usage Policy: The iASP has some constraints on the WiMAX web services usage, such as time usage, duration, frequency, black/white list, privacy indication, service invocation result restriction, session parameter, priority level;
- SLA policies: The iASP needs to respect a SLA agreed with the NSP.

## USI

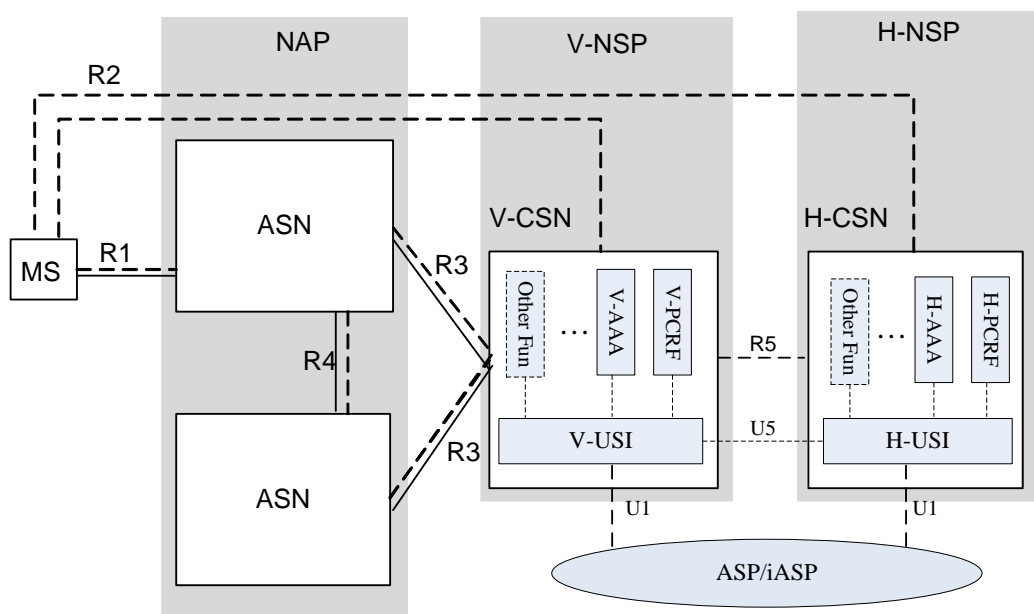
Examples of policies related to the Subscriber are:

- Identity hiding: the Subscriber did not authorize the iASP to see his/her real identity so the USI System has to apply identity hiding.

The USI System provides to iASPs several WebServices, where each WebService represents a functionality to be exposed, such as Terminal Capabilities, Terminal Location. In order to expose the functionality, a WebService can interact at the lower level with one ("Web Services Logic a" in Figure 2) or more network resources: this is depicted as "Web Services Logic b" and "Web Services Logic b'" respectively in Figure 2. Another case to be taken into account, is the case when a network resource has already a Web Service interface: in this case, a Web Service binding allows to expose that functionality (i.e. the already existing Web Service) in the context of the USI System exploiting all the functionalities that the USI System can provide ("Web Services Binding" in Figure 2).

## 5.2 Roaming Architecture

The Network Reference Model for USI Roaming is as following:



**Figure 3: USI Network Reference Model (Roaming)**

Where the Visited USI (V-USI) is the USI System located in the visited NSP and the Home USI (H-USI) is the USI System located in the home NSP.

A new reference point U5 is introduced for USI roaming. The U5 reference point is a logical connection between the V-USI and the H-USI. The primary purpose of the U5 is to transport the USI information and signalling between H-USI and V-USI when MS roams to the V-NSP. The functionality of the U5 reference point includes:

- Transmitting the USI registration signaling between H-USI and V-USI;
- Transmitting the USI Identity authentication signaling between H-USI and V-USI;

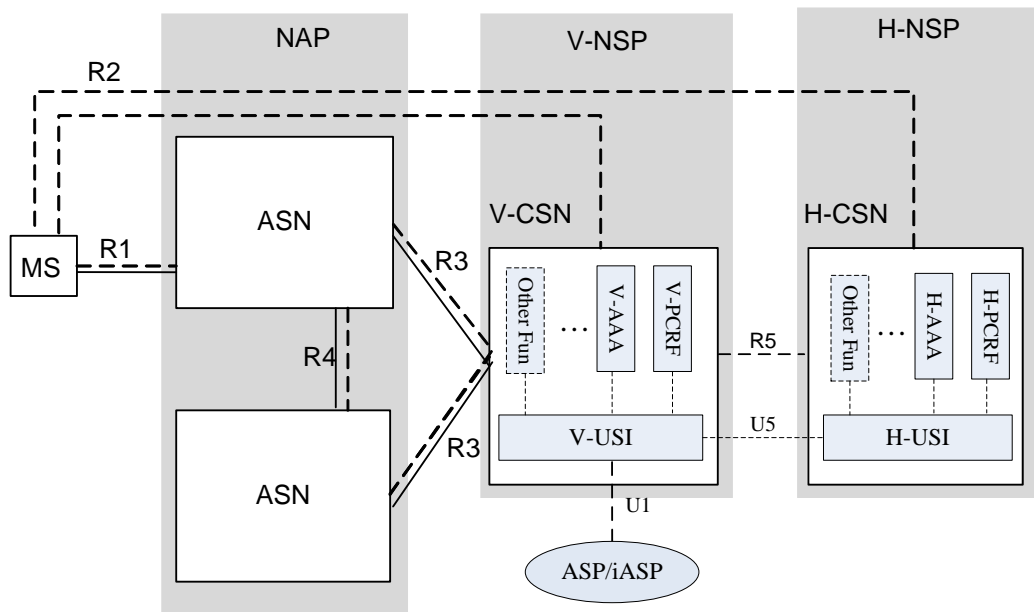
## USI

- Transmitting the USI service calling signaling between H-USI and V-USI;
- Transmitting the USI service related accounting information between H-USI and V-USI;

If iASP has an interface with both H-USI and V-USI, either H-USI or V-USI can be used as the interface to call the network functions, which one SHALL be used depends on the specific service or the agreement between H-NSP and V-NSP; For the local service e.g. MCBSC, QoS inquiry, the V-USI is preferred.

For the case that service provided by iASP is via V-USI calling, part or all of the user's USI profile SHALL be transferred to V-USI from H-USI for the V-USI SHALL recognize and authenticate the user's USI ID.

### 5.2.1 Local USI Services

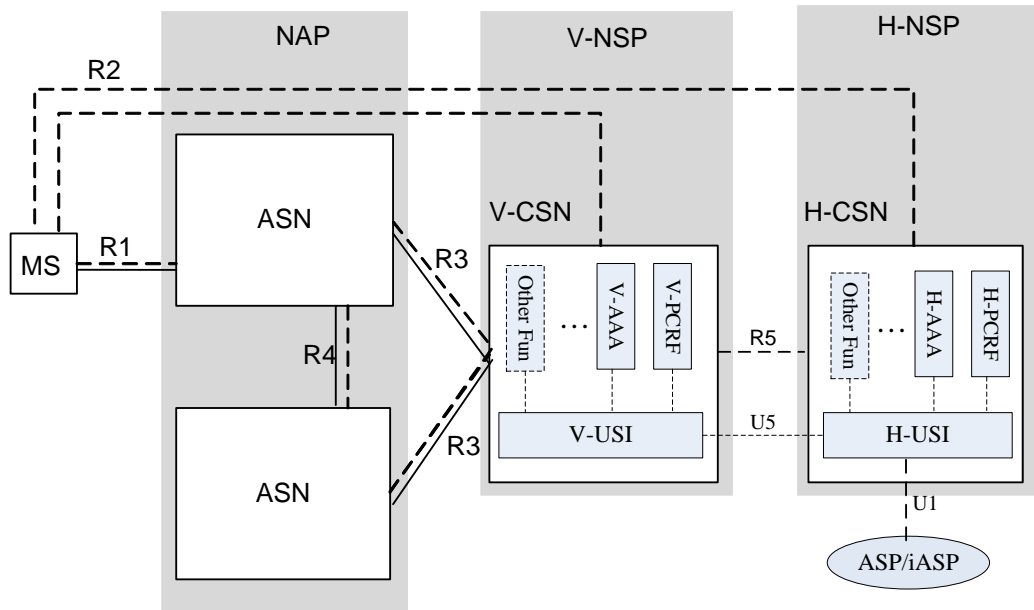


**Figure 4: Local USI Services in Roaming**

In this scenario, the iASP sends the service request to the V-USI through U1 interface. The U5 interface is necessary in this scenario because a) the H-USI does not have a USI SLA with the iASP, or b) the H-USI has a USI SLA with the iASP, but the iASP is required to use the local USI..

### 5.2.2 Home USI Services

## USI



**Figure 5: Home USI Services in Roaming**

In the Scenario 2 the iASP sends the service request to the H-USI through U1 interface. The U5 interface is necessary in this scenario because a) the V-USI does not have a USI SLA with the iASP, or b) the V-USI has a USI SLA with the iASP, but the iASP is required to use the home USI.

## 6. USI Procedures

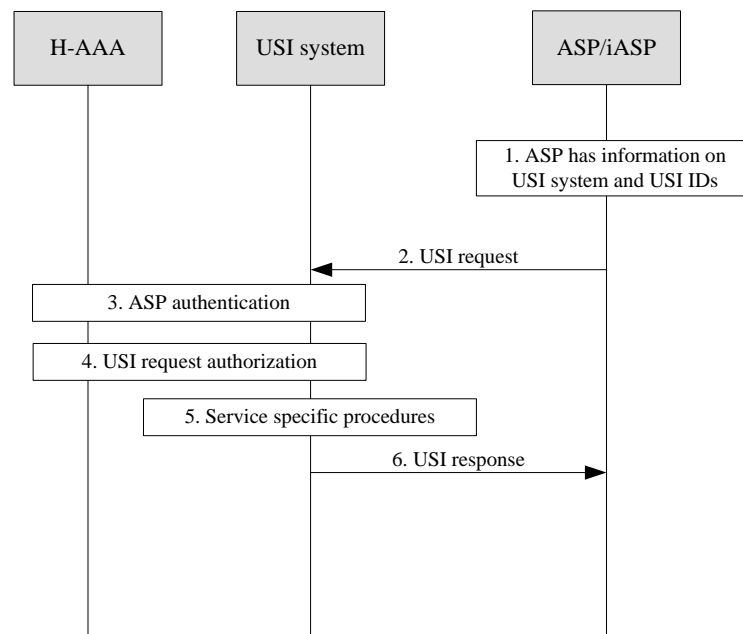
### 6.1 USI Service Flows (Non-Roaming)

The USI service is delivered by means of a USI request – USI response pair. A USI request is submitted to USI System and a USI response carries the service response.

This section describes the possible USI services flows where a USI request can be fully serviced by the USI System's CSN. Cases where USI service involves components from other CSNs and/or roaming MS(s), are described in section 6.2

#### 6.1.1 Direct USI service flow

If iASP has all the required information to submit the USI request, the iASP MAY use the flow described in Figure 6 to submit a USI request.



**Figure 6: Direct USI Service Flow**

1. iASP has prior knowledge of the USI System address and any required USI IDs. This MAY be, for example, because the iASP has previously used the User/MS redirected USI serviceflow.
2. iASP submits a USI request to USI System
3. The iASP MAY be authenticated by USI System.
4. The USI request is then authorized by USI System prior to actual service delivery. The step 3 and 4 can be combined to reduce the message exchange between USI and AAA server.
5. Following USI request authentication and authorization, the USI System conducts service specific procedures. These procedures are detailed in section 0. For each specific service, the service specific procedures MAY involve other components from USI System's own CSN and possibly one or more MS

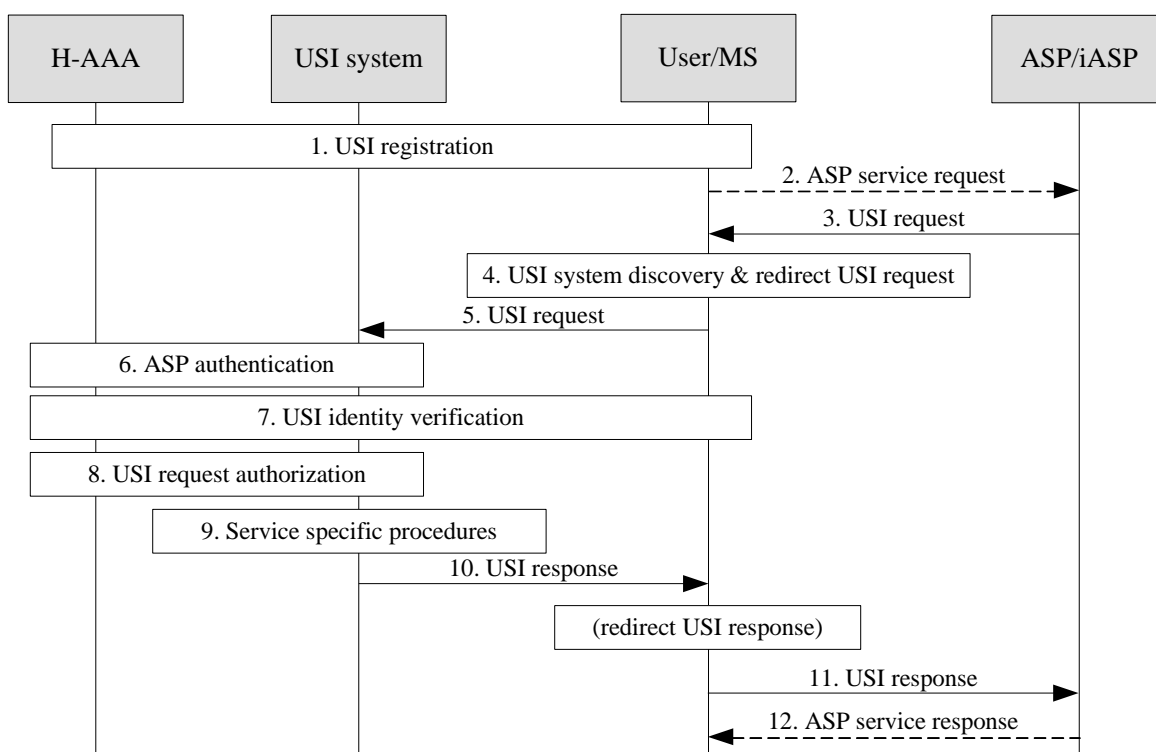
## USI

6. A USI response is then delivered back to the iASP

### 6.1.2 Redirected USI Service Flow

User/MS redirected USI service flow if iASP requires MS assistance in detecting the serving USI System, or iASP requires User/MS authentication or USI ID allocation, the iASP MAY use flow described in Figure 7.

NOTE: The flow is designed to work on MS side even with a standard web browser.



**Figure 7: User/MS Redirected USI Service Flow**

1. User/MS is registered in USI System
2. User/MS (MAY be a standard Web browser) submits an iASP service request to iASP
3. iASP constructs a USI request and sends USI request to User/MS with instructions to redirect to USI System
4. User/MS optionally performs the USI System discovery procedures as described in Section 6.6.2. This step takes place based on what iASP requires.
5. User/MS redirects the USI request to the serving USI System.
6. The iASP MAY be authenticated by USI System. This step is a part of the procedure described in Section **Error! Reference source not found.** and requires iASP to provide information.

## USI

7. .Optionally, the User/MS identity is detected and verified by USI System as described in Section **Error! Reference source not found..**

8. USI System checks for authorization to perform USI request prior to actual service delivery. The step 6,7 and 8 can be combined together to reduce the message exchange between USI and AAA server

9. Following USI request authentication and authorization, the USI System conducts the service specific procedures, these procedures are detailed in Section 0 for each specific service, The service specific procedures MAY involve other CSN components from USI System's own CSN and MAY also involve one or more MS. This is specific to the USI service and is not shown here.

10. USI System delivers a USI response back to the User/MS with instructions to redirect response to iASP. The USI response optionally contains the USI identity of the User/MS and a proof of verification of this USI identity from USI server.

11. User/MS redirects the USI response to the iASP

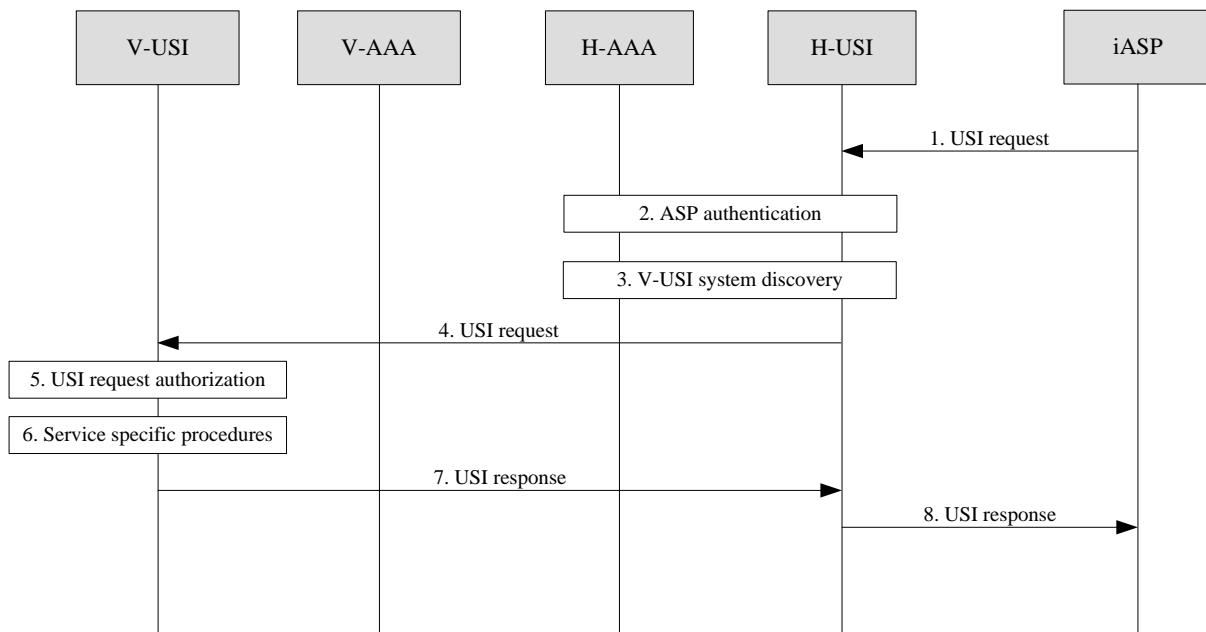
12. iASP delivers response to the User/MS

It is up to iASP to decide whether it uses the Direct or User/MS redirected USI service flow. In addition, the USI request in the redirect service flow can also reference the 'current User/MS', without explicitly specifying the User/MS identity, which MAY not be known at the time.

## 6.2 USI Service Flows (Roaming)

### 6.2.1 Home USI services

In case of a roaming MS, when a USI request is submitted to the USI System of the home NSP (called H-USI), that needs service from the USI System of the MS visited network, called V-USI, the flow is described in Figure 8.



**Figure 8: Home USI Direct USI ServiceFlow**



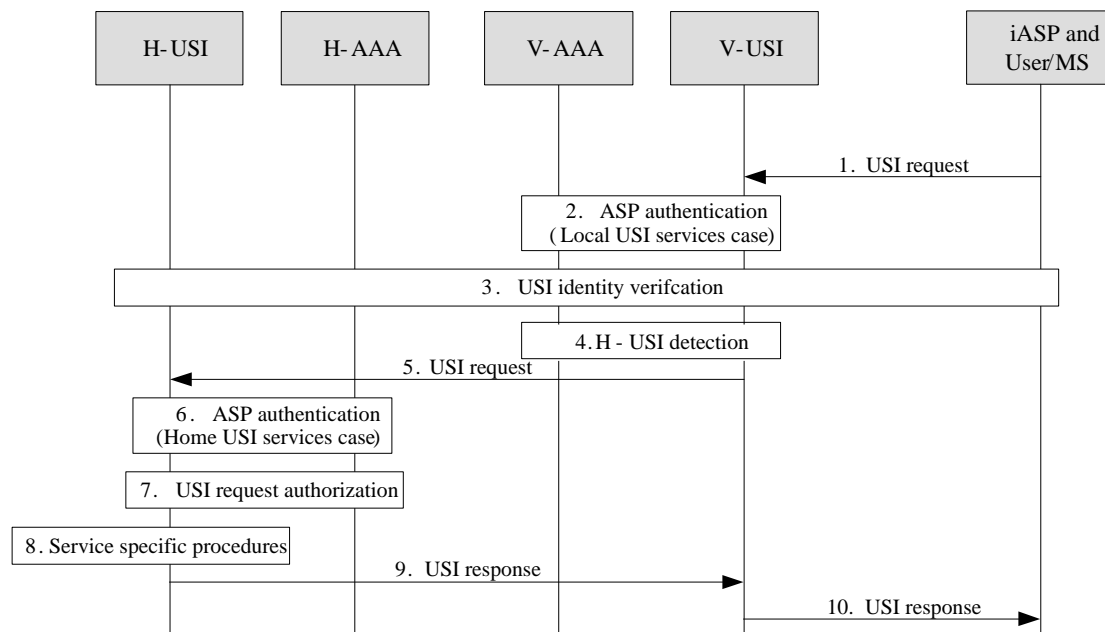
## USI

1. iASP and/or User MS submits a USI request to H-USI System as per section 6.1 (Direct USI service flow)
2. H-USI, optionally in conjunction with H-AAA authenticates iASP
3. Since MS is roaming, and USI request must be served by a foreign USI System (the V-USI), the H-USI detects the V-USI, optionally assisted by AAA
4. Then USI request is directed to V-USI carrying authorization from H-USI to V-USI regarding the request and iASP.
5. V-USI System checks for authorization to perform USI request prior to actual service delivery.
6. V-USI System conducts service specific procedures, These procedures are detailed in Section 0
7. V-USI System delivers a USI response back H-USI
8. H-USI System delivers the USI response back to the iASP or User/MS

### 6.2.2 Local USI service flows

In case of a roaming MS, when a USI request is submitted to the USI System of the home V-NSP (called V-USI), that needs service from the home USI System of the MS, called H-USI, the flow is described in

Figure 9 for both Direct USI service flow and User/MS redirected USI service flows.



**Figure 9: Local and Home USI service flow (V-USI is accessed first)**

The flow shown in Figure 9:

1. USI request is submitted to V-USI System, directly by iASP (Direct flow) or via User/MS using the User/MS redirected flow
2. If the iASP is known to the V-USI (local USI services case), the iASP is authenticated by V-USI System, possibly in conjunction with AAA server.

## USI

3. Optionally, the User/MS identity is detected and verified by V-USI System as described in Section **Error! Reference source not found.** In conjunction with V-AAA, H-AAA and H-USI. This step takes place based on what iASP requires.
4. Since USI request must be serviced by H-USI, V-USI System assisted by V-AAA detects the H-USI System which will be serving the USI request.
5. V-USI submits a USI request to H-USI System
6. If the iASP is known to the H-USI (home USI services case), the iASP is authenticated by H-USI System, possibly in conjunction with V-AAA server.
7. H-USI System and H-AAA authorizes the USI service request
8. H-USI System conducts service specific procedures, These procedures are detailed in Section 7 for each specific service,
9. H-USI System return USI response to V-USI System
10. V-USI System returns USI response to caller

### 6.3 USI Identity

USI identity is the means by which the MS and its serving NSP are identified. To ensure user-privacy and security purposes, user pseudo-identity SHALL be used for user identity. The user pseudo identity MAY be short-lived (e.g. hours, days, months) and MAY be allocated by the NSP. Additionally, a long-lived user identity MAY be used. A USI identity of User/MS is an identification that SHALL include the user pseudo identity and the serving NSP realm. USI identity MAY be used according to serving NSP policy.

The USI identity SHALL be a network access identifier (NAI) as defined [RFC2486]:

[user-identity@usi.NSP-identity](#)

The realm part of the USI identity SHALL be the FQDN of the USI System as appears in the Internet public DNS.

#### 6.3.1 Long-Lived USI Identity

To support Direct USI service flow, a long-lived user USI identity (L-ID) MAY be used. L-ID is allocated to the user by the Home NSP. And the user identity part of L-ID SHALL be unique in this NSP. Once an L-ID is generated, it is known to the user or kept in MS, and it is kept in USI System. User/MS MAY provide iASP its L-ID through application layer interaction, such as iASP registration or URL, and iASP MAY keep this user's L-ID as a part of the user's profile. For example, the L-ID MAY be appended to the URL:

<http://www.asp-name.com/search?q=restaurant&usi-identity=joe@usi.serviceprovider.com>

#### 6.3.2 Short-Lived USI Identity

To ensure user-privacy and security, iASP SHALL use a short-lived USI identity (S-ID). S-ID MAY be generated by USI System during USI request, during User/MS initiated flows or when an iASP submits USI request to USI System using a user's L-ID as described in 6.1. Every generated S-ID SHOULD be randomly generated using a globally unique identifier algorithm. The S-ID has a lifetime, and USI System SHOULD keep the mapping of this S-ID and its associated L-ID in the lifetime of the S-ID. When an S-ID expires, USI System will delete the old S-ID and MAY generate a new S-ID and substitute the old one. It is up to the policy whether to generate a new S-ID or not.

## USI

When an iASP contacts USI System with a user's L-ID, USI System will inform the iASP of the S-ID associated with that L-ID. And iASP MAY keep this S-ID for the associated L-ID until the S-ID's timeout. Before that information, USI System SHOULD check if such an S-ID exists. If not, USI System SHOULD generate an S-ID for the associated L-ID provided in the request of iASP.

For roaming cases, for local USI services, the V-USI System allocates the S-ID using the V-NSP realm. For the roaming home USI services, the H-USI System allocates the S-ID using H-NSP realm.

### 6.3.3 Usage of USI Identities

This section provides some usage guidelines of L-ID and S-ID. L-ID is a semi-permanent USI identity that allows the iASP to initiate USI transaction. The iASP which has obtained user's L-ID MAY combine the L-ID with user's iASP ID. When the iASP initiates a USI transaction targeted to a user in the case when the iASP doesn't know the user's S-ID, the L-ID is used. When an iASP contacts the USI System with a user's L-ID, the USI System will generate an S-ID for the user, and then inform this S-ID to the iASP. After that, the iASP will use the new S-ID for the subsequent USI transactions until the S-ID times out. The descriptions of the two IDs are summarized as following:

	Definition	Scope	Allocation	Authentication
L-ID	Supporting Direct USI transaction	MS、NSP、iASP	By NSP to MS at service agreement	By request by iASP indicated in USI Request
S-ID	Supporting privacy and limited access	NSP、iASP	By NSP to iASP at USI Request	By request by iASP indicated in USI Request

#### 6.3.3.1 USI System discovery and User/MS's identification by iASP

Table 1 lists the possible cases and corresponding solutions to USI System discovery and obtaining the User/MS's USI ID.

**Table 1: User/MS Identification and USI System discovery by iASP**

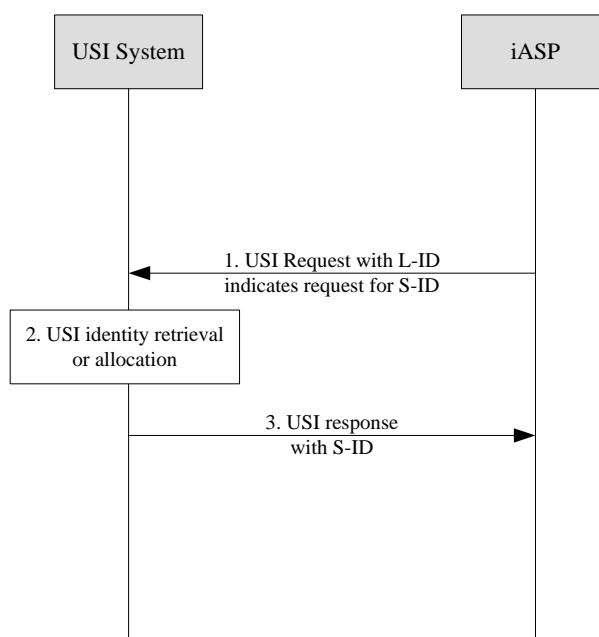
Case	Serving USI System Known	USI Identity Known	Solutions
1	No	No	<ol style="list-style-type: none"> <li>iASP uses redirect flow, requests USI ID (as described in Section 6.3.4) and USI System detection (as described in Section 6.6.2)</li> <li>iASP obtains USI ID by out-of-band means (e.g. direct user input), then uses iASP initiated flow with serving USI or to H-USI</li> </ol>
2	Yes	No	<ol style="list-style-type: none"> <li>iASP uses redirect flow, requests USI ID (as described in Section 6.3.4) and USI System detection (as described in Section 6.6.2)</li> <li>iASP obtains USI ID by out-of-band means (e.g. direct user input), then uses iASP initiated flow with serving USI or to H-USI</li> </ol>
3	No	Yes	<ol style="list-style-type: none"> <li>1. iASP uses iASP initiated flow with H-USI</li> <li>2. iASP uses User/MS initiated flow with USI System detection (for serving USI), or H-USI</li> </ol>
4	Yes	Yes	<ol style="list-style-type: none"> <li>iASP uses iASP initiated flow with H-USI or serving USI in case of roaming</li> <li>iASP uses User/MS initiated flow with USI System detection (for serving USI), or H-USI</li> </ol>

## USI

**6.3.4 USI Identity Allocation & Renewal**

L-ID or S-ID allocation can be requested at any time, by submitting a USI Request. To do so, the USI Request must contain a <AuthnRequest>. The allocated USI ID is contained in the USI Response.

It is up to the USI System to decide whether a new pseudo-identity SHOULD be generated (e.g. in case of expired S-ID) or if an existing pseudo-identity is returned based upon operator and user policy.

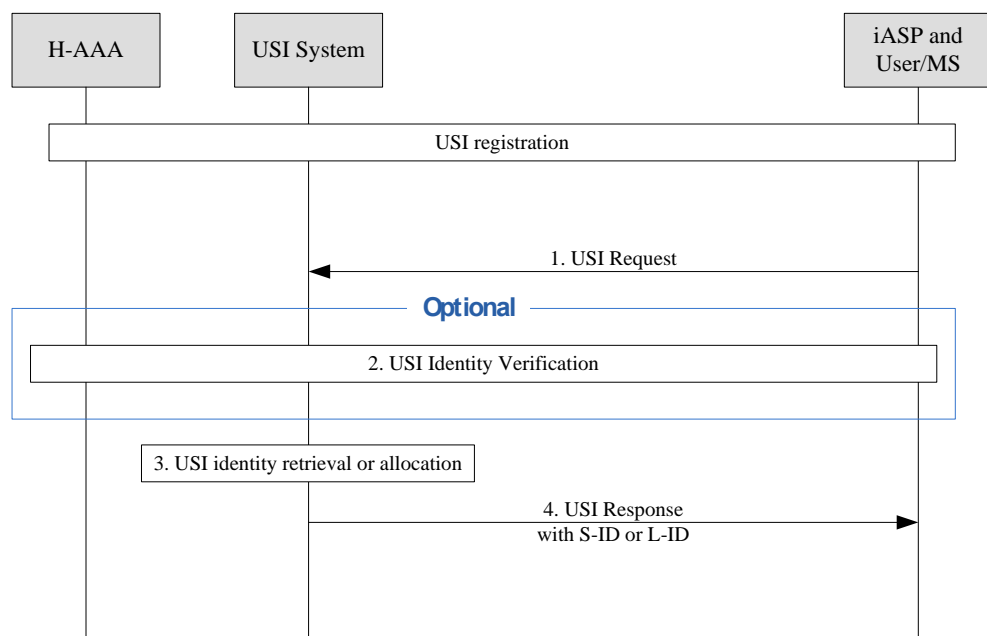
**6.3.4.1 Direct Flow**

**Figure 10: S-ID Allocation or Retrieval**

1. The iASP submits a USI Request to the USI System, optionally with L-ID (if present) and MAY explicit request for S-ID
2. The USI System decides whether to allocate a new S-ID or reissue an existing S-ID to the iASP.
3. The USI System communicates the assigned S-ID to the iASP in the USI Response

**6.3.4.2 MS/User Redirected Flow**

## USI



**Figure 11: S-ID Allocation or S-ID/L-ID Retrieval in**

1. The User/MS requests the USI ID from the USI System, by submitting a USI Request with optional explicit request for L-ID or S-ID.
2. The USI System MAY do a verification of the identity of User/MS before retrieving or allocating a USI ID. This procedure is defined in section **Error! Reference source not found..**
3. The USI System decides whether to allocate a new USI S-ID or reissue an existing USI S-ID. L-ID MAY also be retrieved during this procedure.
4. The USI System communicates the assigned USI ID in the USI Response

Since the USI ID MAY be shortlived, the USI S-ID MAY be updated to enhance the user privacy and security. The procedure for the USI ID update is similar to the procedure for identity allocation or retrieval described in this section. In this case, however, the USI System will perform stages 2, 3 and 4 even without the explicit request (<AuthnRequest>) from User/MS in the USI request at stage 1.

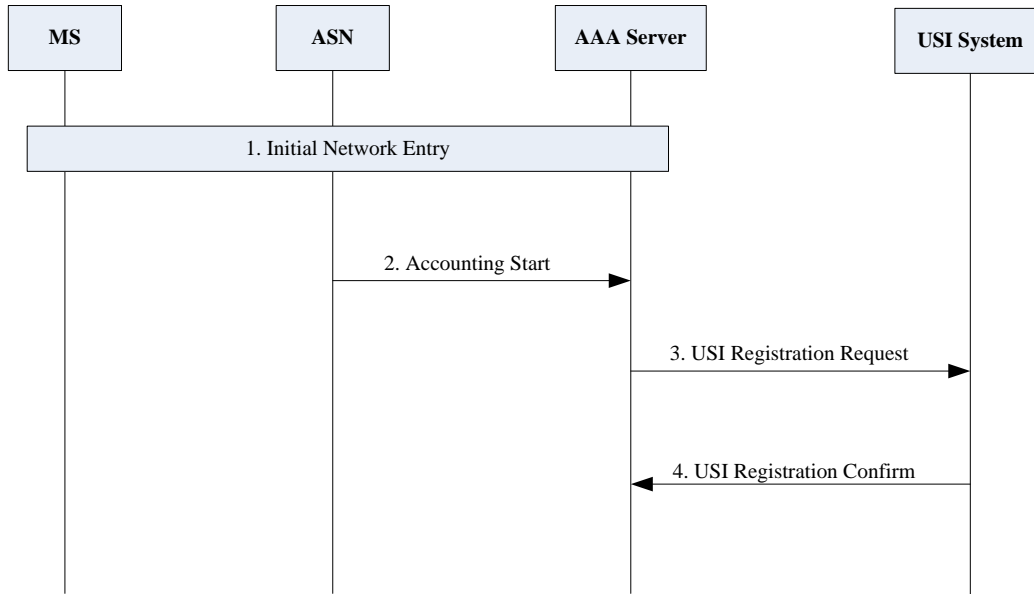
## 6.4 USI Registration

USI registration is a procedure where the presence of an MS (subscriber) and its relevant context (called a 'USI context') is registered in the USI System. The MS (subscriber) context is used by the USI System to exchange, authorize, and enable USI services.

### 6.4.1 USI Registration (Non-Roaming)

The USI registration procedure is illustrated in figure below.

## USI

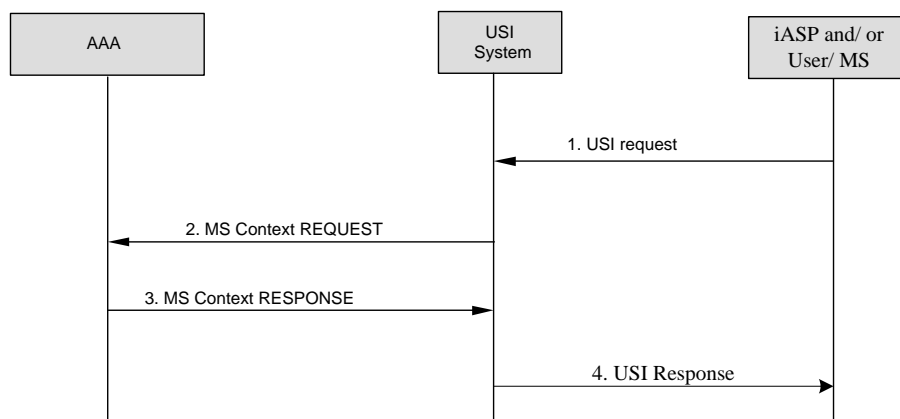


**Figure 12: USI Registration (NSP-Triggered)**

1. Initial Network Entry (or re-entry) is performed [NWGSTG3]. At this step, the MS is challenged and authenticated by the AAA Server that is aware of the MS capabilities and subscription to USI services.
2. The AAA Accounting Start informs the AAA Server of the MS IP address and start of session.
3. Upon successful Initial Network Entry, the AAA Server sends the USI REGISTRATION REQUEST to the USI System. The USI REGISTRATION REQUEST includes the MS context with the following information: the MS identity (MAC/IP Address), USI L-ID, Subscribed Capabilities (i.e. the capabilities that the USI System can enable based on user subscription, the MS and/or network capability).
4. The USI System registers the MS and stores the MS context and MAY send the USI REGISTRATION CONFIRM after successful registration confirming successful USI registration to the AAA Server.

It SHALL also be possible to trigger the registration process when iASP requests via U1 interface to avoid unnecessary registration (as shown in Figure 13).

## USI



**Figure 13: USI Registration (iASP and/or User/MS Triggered)**

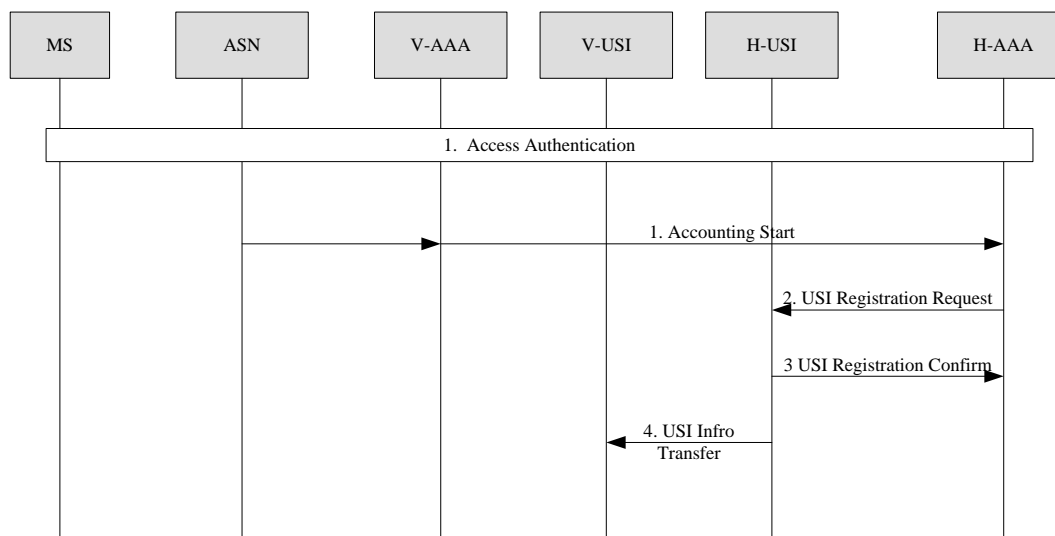
1. USI System receives a USI trigger from the iASP and/or User/MS (in case of redirected flow)
2. If USI System does not have the MS/User context for which iASP requested USI service, the USI System sends USI CONTEXT REQUEST to AAA Server. This message contains the MS/User Identify. The USI System MAY authenticate the MS/User and/or the iASP.
3. If the MS has already entered the network, the AAA Server responds with the USI CONTEXT RESPONSE to the USI System. The USI CONTEXT RESPONSE includes the MS context with the following information: the MS identity, Subscribed Capabilities (i.e. the capabilities that the USI System can enable based on user subscription, the MS and/or network capability). The USI System SHALL store this information for a set period of time.
4. The USI processes the USI Service request and replies the iASP and/or User/MS with the USI Response message.

The interface between USI System and AAA Server is outside the scope of USI work item.

#### 6.4.2 USI Registration (Roaming)

The USI Registration can be triggered by either NSP or iASP. The USI Registration triggered by Home AAA is as following:

## USI

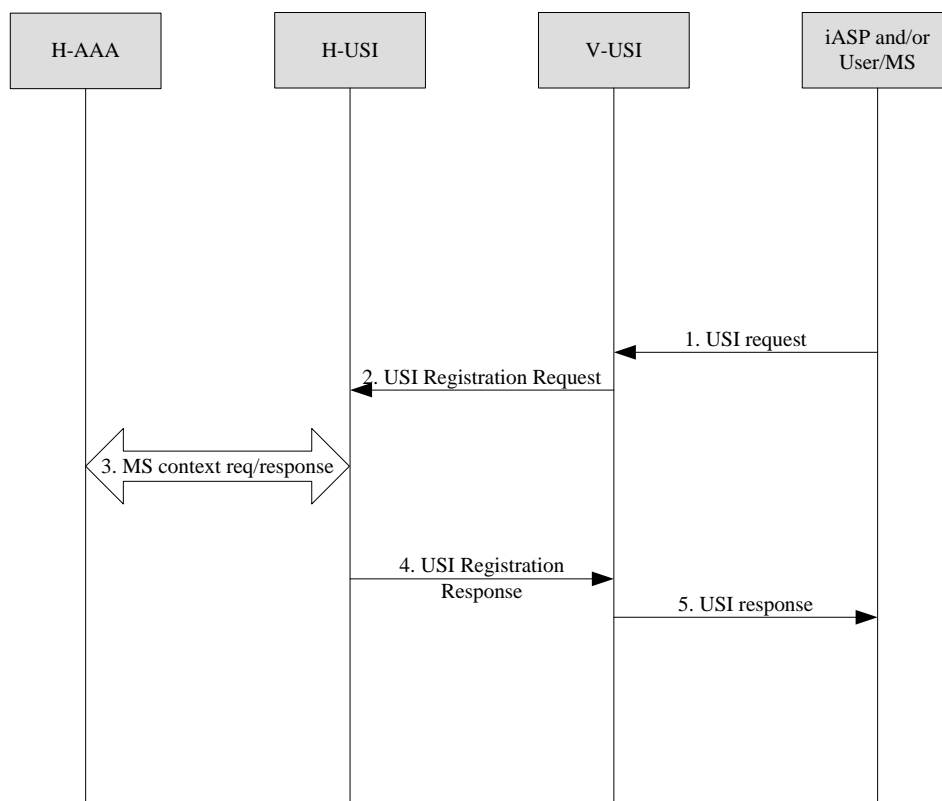


**Figure 14: USI Registration for USI Roaming (H-AAA triggered)**

1. Initial Network Entry (or re-entry) is performed. At this step, the MS is challenged and authenticated by the H-AAA Server that is aware of the MS capabilities and subscription to USI services.
  2. Upon successful Initial Network Entry, the H-AAA Server sends the USI Registration Request to the H-USI System. The USI Registration Request includes the MS context with the following information: the MS identity, for the MS identity, the long-lived USI ID is suggested to use if the MS has one before the USI registration procedure, other MS identity can be used as well, e.g. IP address, Subscribed Capabilities (i.e. the capabilities that the USI System can enable based on user subscription, the MS and/or network capability);
  3. The H-USI System registers the MS and stores the MS context. The H-USI MAY send the USI Registration Confirm after successful registration confirming successful USI registration to the AAA Server. A Short-lived USI ID is generated by H-USI as the user's temporary identity in the following USI service process to ensure user privacy and security. The V-USI realm information SHOULD be embodied in this Short-lived USI ID to assist the H-USI and iASP (Especially for Scenario 1) to locate the V-USI;
  4. The H-USI transfers part or all of the user's USI profile or registration information including the generated Short-lived USI ID to V-USI. The H-USI system could discover the IP address of the V-USI using a lookup table
- It SHALL also be possible to trigger the registration process when iASP requests via U1 interface to avoid unnecessary registration.
- For local services, the USI registration triggered by iASP the registration is as following:



## USI



**Figure 15: iASP triggered USI Registration for USI Roaming (For Local Services)**

1. The USI Service is triggered by iASP (and/or may be redirected by the MS/User) due to the application layer communication between MS and iASP. The iASP initiates a USI Service Request message and sends the USI Service Request message to the V-USI.
2. The V-USI sends a USI Registration Request message to the user's H-USI System over the U5 reference point.
3. The H-USI checks the user's subscribed USI service profile with H-AAA and stores the MS context after USI registration. The V-USI System can also register the MS locally if V-USI has the MS context, else V-USI shall communicate with V-AAA and/or H-AAA to retrieve the MS context.
4. The H-USI replies the V-USI with USI Registration Response message over the U5 reference point. Part or all of the user's USI profile or registration information including the generated Short-lived USI ID SHALL be included in the USI Registration Response message.
5. The V-USI process the USI Service according to the USI Service Request message received from iASP and replies the iASP with the USI Service Response message.

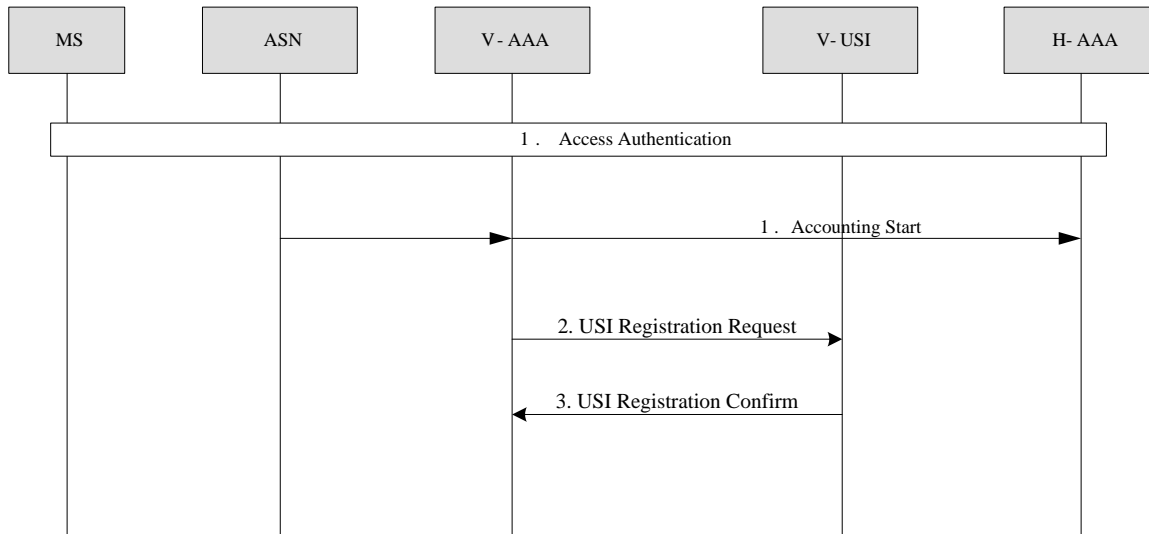
Local USI Registration at V-USI shall be supported, which can

1. Accelerate the process of Local USI Service calling
2. Provide more Local USI service than the Home USI to user
3. Be applicable for the scenario where Visited NSP has a USI system while the Home NSP have no USI system, for which the USI service calling SHALL be based on R5.

## USI

Whether the Local USI Registration at V-USI is allowed to provide more local USI services by V-USI depends on the USI Roaming agreement between Home and Visited NSP.

The procedure of Local USI registration at V-USI triggered by V-AAA is as following,

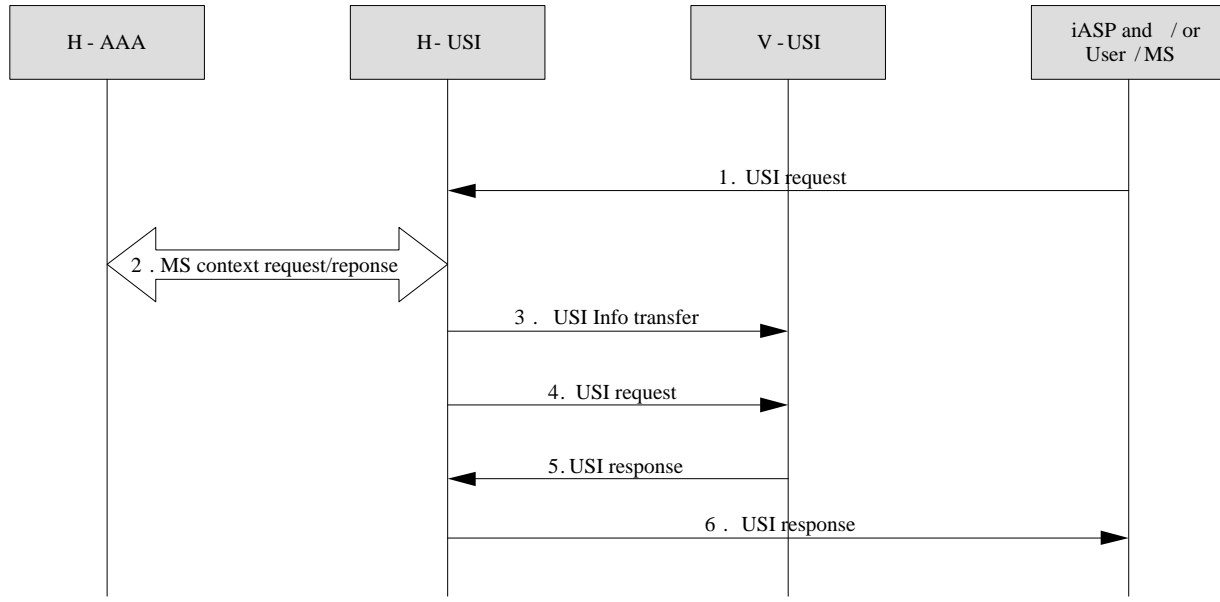


**Figure 16: NSP triggered Local USI Registration for USI Roaming**

1. Initial Network Entry (or re-entry) is performed. At this step, the MS is challenged and authenticated by the H-AAA Server via V-AAA. The V-AAA shall obtain the MS context from H-AAA during initial access if the local USI registration is supported.
2. Upon successful Initial Network Entry, the V-AAA Server initiates the local USI registration by sending a USI Registration Request to the V-USI. The USI Registration Request includes the MS context with the following information: the MS identity, for the MS identity, the long-lived USI ID is suggested to use if the MS has one before the USI registration procedure, other MS identity can be used as well, e.g. IP address, Subscribed Capabilities (i.e. the capabilities that the USI System can enable based on user subscription, the MS and/or network capability). The V-AAA shall retrieve the MS context from H-AAA by sending a MS context request message if it doesn't have the MS context.
3. The V-USI System registers the MS locally and stores the MS context. The V-USI MAY send the USI Registration Confirm after successful registration confirming successful USI registration to the V-AAA Server.

For Home Services, the USI registration triggered by iASP the registration is as following:

## USI



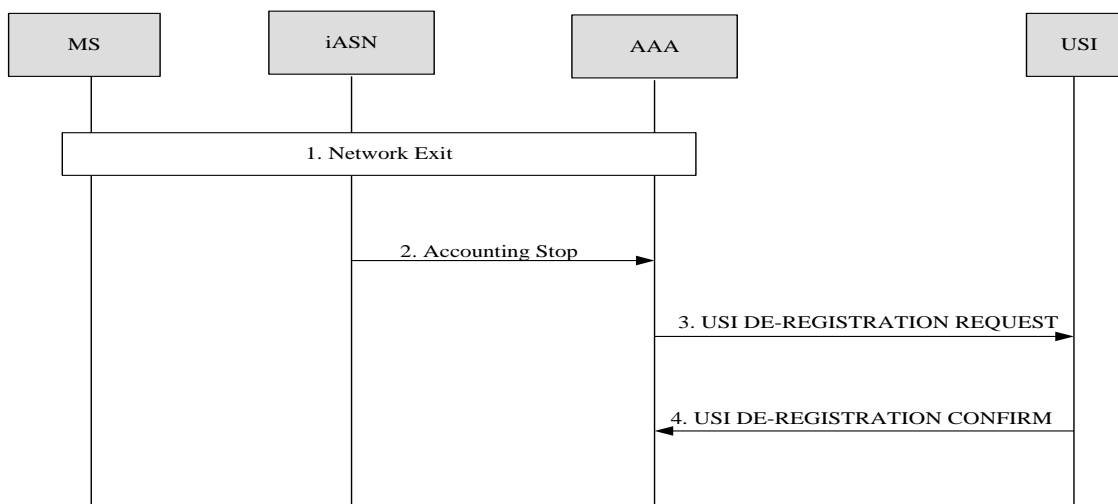
**Figure 17: iASP and/or MS/User triggered USI Registration for USI Roaming (For Home Services)**

1. The USI Service is triggered by iASP (and/or may be redirected by the MS/User) due to the application layer communication between MS and iASP. The iASP initiates the USI Service Request message and sends the USI Service Request message to user's H-USI System over U1 reference point.
2. The H-USI checks the user's subscribed USI service profile with H-AAA and stores the MS context after USI registration.
3. The H-USI transfers part or all of the user's USI profile or registration information including the required USI IDs to V-USI.
4. USI request is relayed to V-USI
5. The V-USI replies with USI Response message to H-USI.
6. The H-USI replies with USI Response message to iASP over the U1 reference point.

## 6.5 USI Deregistration

USI de-registration is a procedure where the presence of an MS (subscriber) and its relevant context is removed (de-registered) from the USI System. The USI de-registration procedure is illustrated in Figure 18.

## USI



**Figure 18: USI De-Registration (NSP-Triggered)**

1. Network Exist is performed [NWGSTG3]. The AAA Server is involved with the network exit procedure and receives the AAA Accounting Stop (release indication).
  2. The AAA Server sends the USI DE-REGISTRATION REQUEST to the USI System. The USI DE-REGISTRATION REQUEST includes the MS identity.
  3. The USI System sends the USI DE-REGISTRATION RESPONSE to the AAA Server and removes the MS context.
- The MS context MAY also be removed by timer expiration in the USI System.

## 6.6 Discovery procedures

### 6.6.1 Discovery of USI support in a CSN

This procedure is used by iASP or NSP as required, for detecting USI support in a CSN of (another) NSP.

A USI System's FQDN must appear in global DNS network. Assuming the NSP realm is known (e.g. from USI ID), a DNS query for USI <NSP-realm> can be made to retrieve the USI System address. Failure indicates the NSP does not support USI.

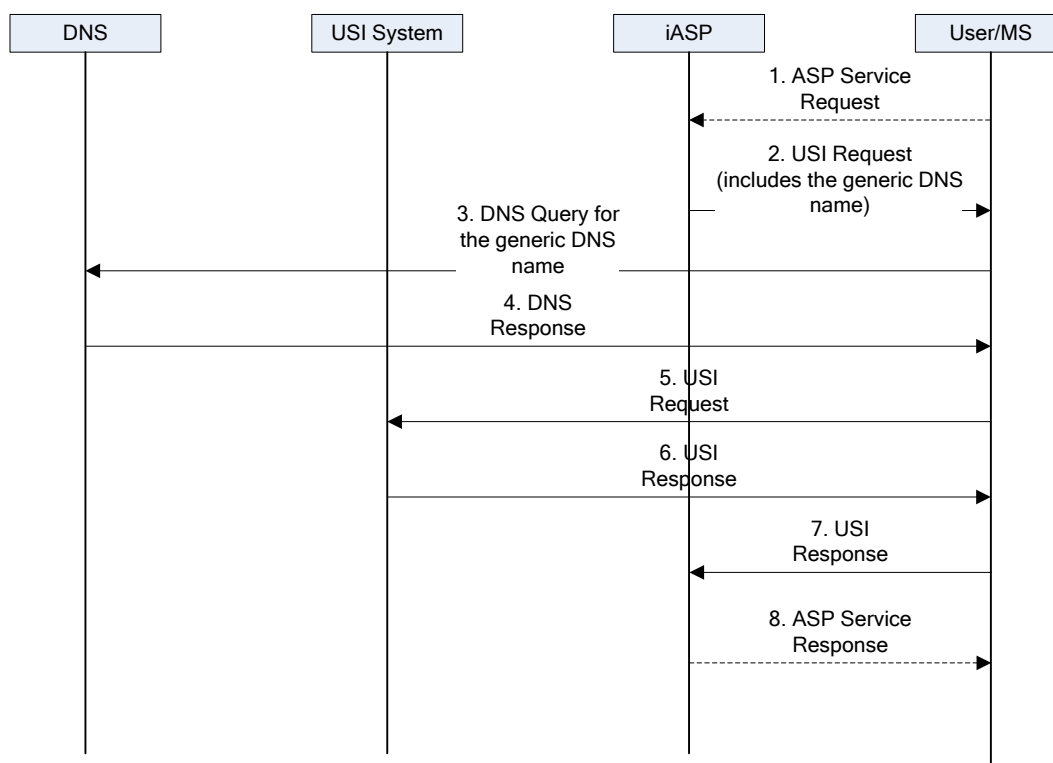
### 6.6.2 Discovery (by iASP) of USI System Serving a User/MS

This procedure is used during User/MS redirected USI flow. iASP can submit a USI request without prior knowledge to the currently serving (Home or Visited) USI System of a User/MS. This is done as described in Figure 19. The iASP substitutes the address of the USI System with a well-known, non-fully qualified domain name of wimaxusi. The DNS server address of MS must be configured by the NSP during network entry to be a DNS server which correctly resolves this domain name to the IP address of the serving USI System for that MS.

The iASP can request the USI response which contains the address of the serving USI System, for use in submitting subsequent USI requests.

## USI

1



2

**Figure 19: Serving USI System discovery in redirect USI service flow**

3

1. The User/MS submits an iASP service request.
2. The iASP responds with a USI request containing the generic DNS name.
3. The MS does a DNS query to resolve the DNS name.
4. The generic DNS name is resolved by serving DNS to the serving USI System's address.
5. The USI request is submitted to the serving USI System.
6. The serving USI system conducts the service specific procedures (as detailed in Section 7), and then responses to User/MS.
7. MS sends a USI service response back to iASP.
8. iASP responses to the User/MS.

Note: Step 3 and 4 may be excluded if MS knows the serving USI system address before step 3.

15

16

17

1  
2

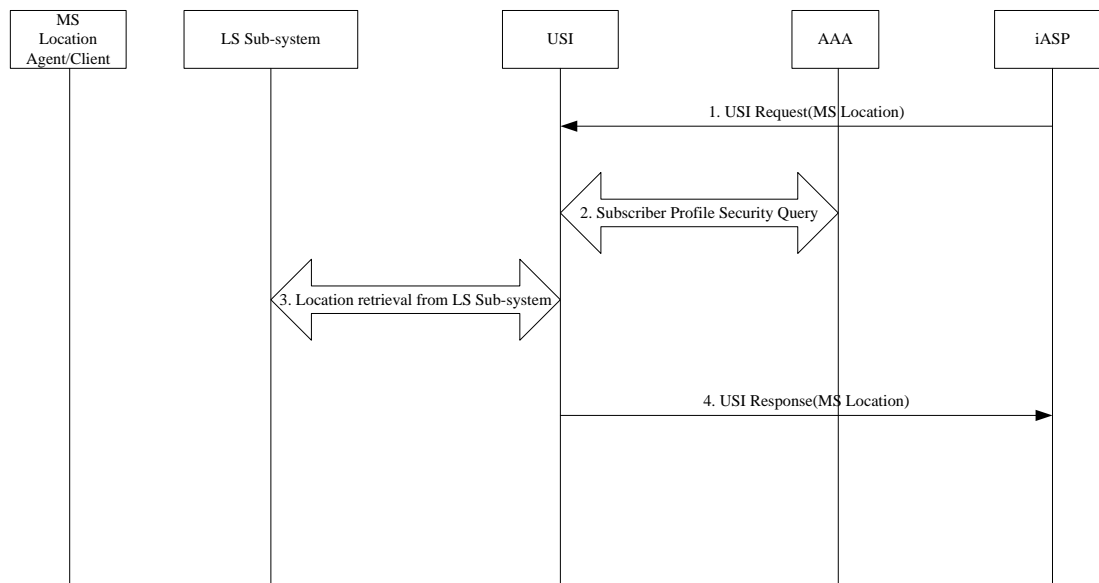
## 7. USI Services

### 7.1 Location Retrieval

LBS can be easily enabled via USI by co-locating the Location Server (LS) within the USI System. Here the iASPs can request the MS location from the USI System during a session with the MS. Based on the MS location, the iASPs MAY be able to provide value added service to the user.

#### 7.1.1 Determining MS Location in Non-Roaming Scenario

Figure 20 shows the location determination for the MS from the USI perspective. The iASP could request the location of the MS from the USI System. The USI System in turn authenticates the request for location. Then it proceeds to determine the location of the MS using the LS Sub-System (as described in [LBS]). The USI System then provides the location of the MS to the iASP and MAY perform an accounting update.

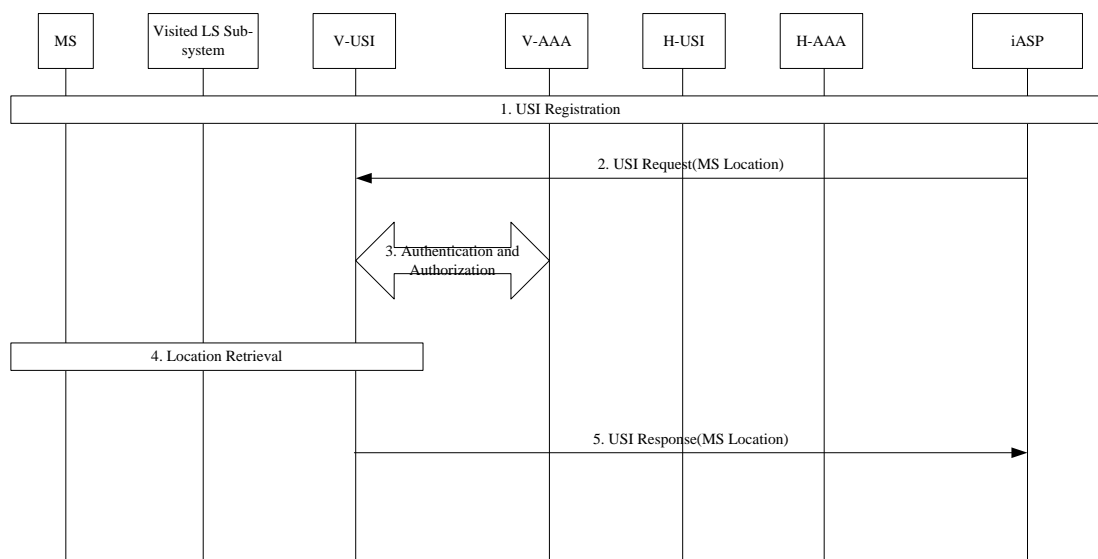


**Figure 20: Determining MS location via USI (non-Roaming)**

#### 7.1.2 Determining MS Location in Roaming Scenarios

For the roaming case of Local USI Services in Roaming, the procedure of determining MS location is as following:

## USI

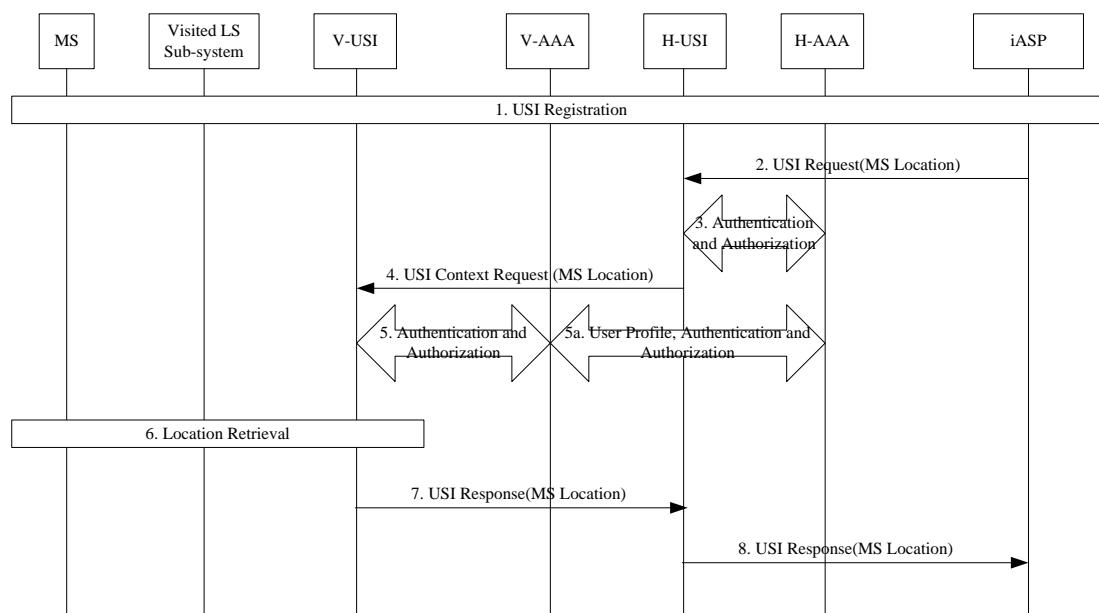


**Figure 21: Determining MS location via USI (Local USI Services in Roaming)**

1. USI Registration procedure. The USI registration is triggered by iASP or network. The user's USI profile or registration information is delivered to V-USI by H-USI after the successful USI registration.
  2. iASP sends USI context request message to V-USI to get MS's location information. The MS's USI ID, e.g. short-lived USI ID or long-lived USI ID SHALL be included in this message to be used to identify the MS, and the short-lived USI ID SHOULD be used in priority if both USI ID are available.
  3. The V-USI triggers the Visited Location service to get the requested location information. The location determination will be done as specified [LBS]. This step may include authentication and authorization.
  4. V-USI replies the iASP with the MS location result by a USI context response message.
- For the roaming case of Home USI Services in Roaming, the procedure of determining MS location is as following:



## USI



**Figure 22: Determining MS location via USI (Home USI Services in Roaming)**

1. USI Registration procedure. The USI registration is triggered by iASP or network. The user's USI profile or registration information is delivered to V-USI by H-USI after the successful USI registration.
2. iASP sends USI context request message to H-USI to get MS's location information. The MS's USI ID, e.g. short-lived USI ID or long-lived USI ID SHALL be included in this message to be used to identify the MS, and the short-lived USI ID SHOULD be used in priority if both USI ID are available.
3. The H-USI should contact the H-AAA to authenticate the incoming USI request.
4. The H-USI determines that the MS is in Visited network, the H-USI forwards the USI context request to the V-USI to request the MS's location information.
5. The incoming location request may be authenticated with the V-AAA. The V-AAA may need to contact the H-AAA for authentication or user profile as shown in step 5a. The steps 5 and 5a are optional procedures.
6. The V-USI contacts the Visited LS Sub-system to get the requested location information. The location determination will be done as specified in the LBS specification [LBS].
7. V-USI replies the H-USI with the MS location result by a USI context response message.
8. The H-USI forwards the MS location result to iASP.

## 7.2 MS Status

USI can be used by the iASPs to judge the user/device status and provide value add services based on the MS status information.

### 7.2.1 Status Context

Status context contains information regarding the power state of the device as well as whether the user is signed onto the NSP.

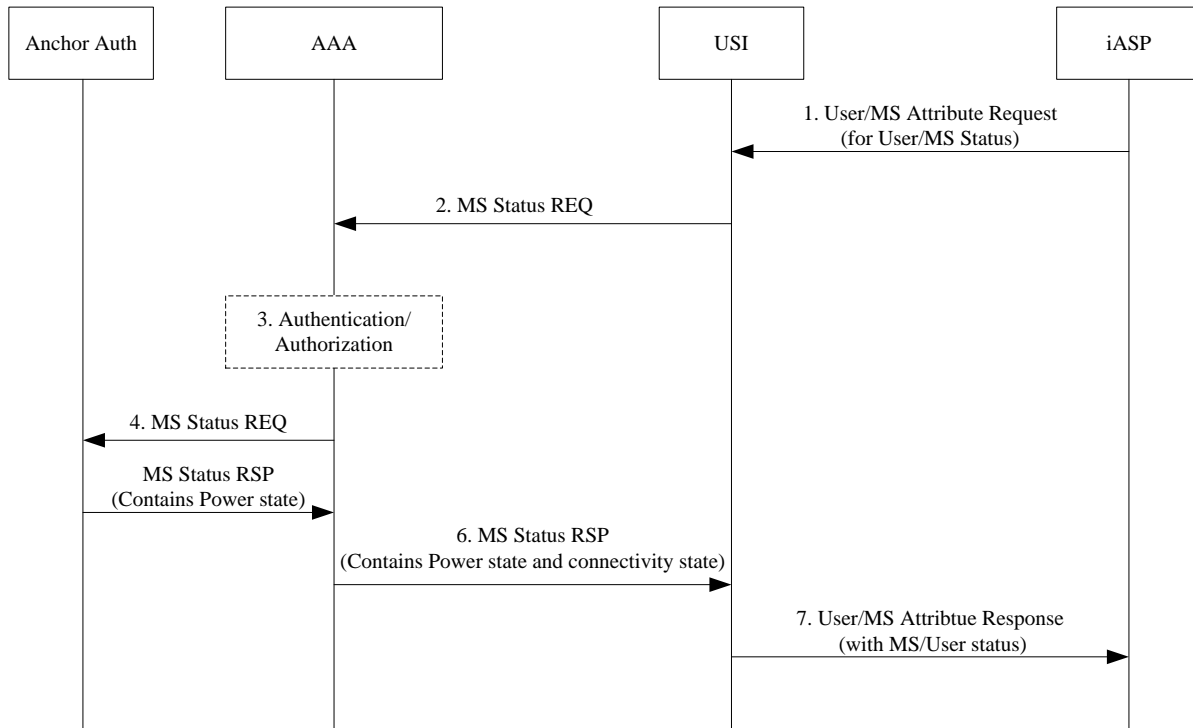
Specifically the user status context SHALL contain:

## USI

- a) User Status aka Connectivity state of the user: the valid values are - CONNECTED or NOT\_CONNECTED
- b) MS status aka Power state of the device: the valid values are - IDLE or ACTIVE

### 7.2.2 Determining MS Status in non-Roaming Scenario

The iASP could request the status of the MS/user from the USI System. Figure 23 shows the status determination for the MS/user from the USI perspective.



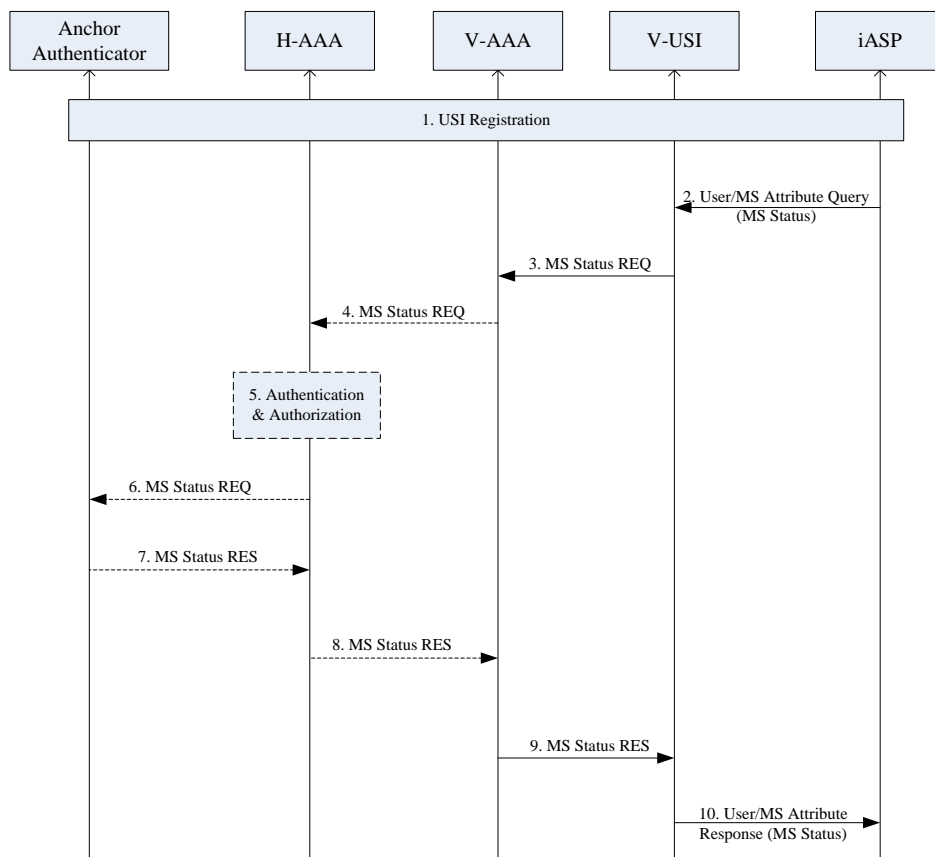
**Figure 23: Determining MS/User Status via USI (Non-Roaming)**

1. The iASP sends User/MS Attribute Request for the MS status to the USI System for providing value added services.
2. The USI System contacts the AAA server for obtaining the status information
3. AAA server authenticates the request
4. AAA server knows the Anchor Authenticator of the MS. It contacts the Anchor Authenticator to obtain the power state of the MS. AAA server is expected to know the connectivity state of the user. This step is optional if the AAA server already knows the power state of the MS
5. Anchor Auth responds back with the power state of the MS to the AAA server. The Anchor auth is expected to know the power state of the MS. This step is optional if the AAA server already knows the power state of the MS through prior AAA updates
6. AAA server responds to the USI with the status data containing the MS and user status
7. The USI System responds back to the iASP with User/MS Attribute Response

## USI

**7.2.3 Determining MS Status in Roaming Scenarios**

For the roaming case of Local USI Services in Roaming, the procedure of determining MS Status is as following:



**Figure 24: Determining MS/User Status via USI (Local USI Services in Roaming)**

1. USI Registration procedure. The USI registration is triggered by iASP or network. The user's USI profile or registration information is delivered to V-USI by H-USI after the successful USI registration.
2. The iASP requests the MS status from the V-USI by the USI context request message. The MS's USI ID, e.g. short-lived USI ID or long-lived USI ID SHALL be included in this message to be used to identify the MS, and the short-lived USI ID SHOULD be used in priority if both USI ID are available.
3. The V-USI contacts the V-AAA server for obtaining the status information. If the V-AAA has already knows the MS's status, V-AAA may authenticate the request and then respond to V-USI with MS Status RSP by step 9, step 4-8 is skipped, else step 4 goes on.
4. The V-AAA forwards the MS Status REQ to H-AAA.
5. H-AAA server authenticates the request.
6. H-AAA server knows the Anchor Authenticator of the MS. It contacts the Anchor Authenticator to obtain the power state of the MS. This step is optional if the AAA server already knows the power state of the MS.

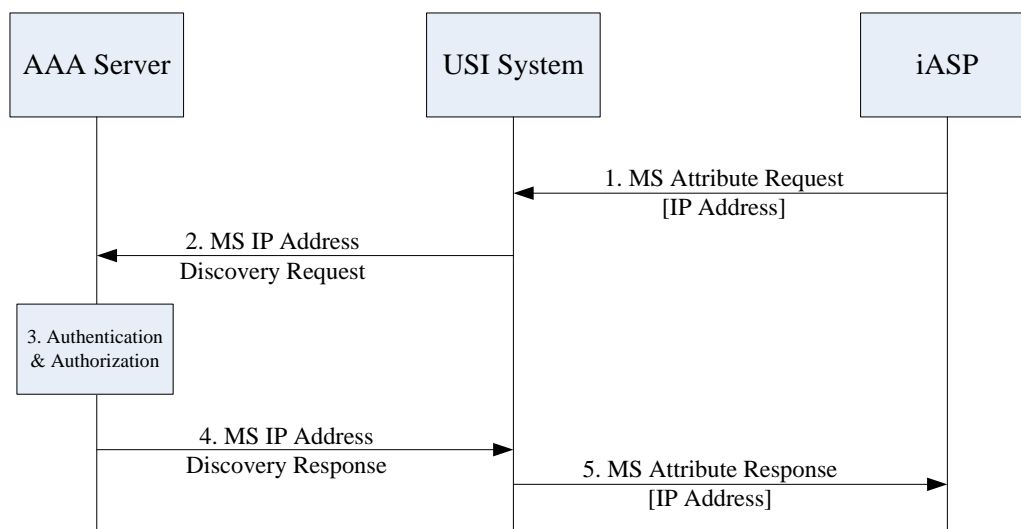
## USI

7. Anchor Auth responds with the power state of the MS to the H-AAA server. The Anchor Auth is expected to know the power state of the MS. This step is optional if the H-AAA server already knows the power state of the MS through prior AAA updates.
  8. H-AAA server responds the V-AAA with the status data containing the MS and user status.
  9. V-AAA forwards the status data containing the MS and user status to V-USI.
  10. V-USI responds to the iASP with the user and MS status data.
- For the roaming case of Home USI Services in roaming, the procedure of determining MS Status SHOULD be terminated within home network without the involvement of V-USI as the non-roaming scenario since the information of MS status is generally maintained in the home network.

## 7.3 IP Address Discovery

### 7.3.1 IP Address Discovery in non-Roaming Scenario

IP Address Discovery is a service facilitating an iASP to request MS IP address from MS's NSP. When the iASP is unaware of the MS's current IP address, the iASP can retrieve it by asking the USI server of MS's NSP with the MS's USI identity.



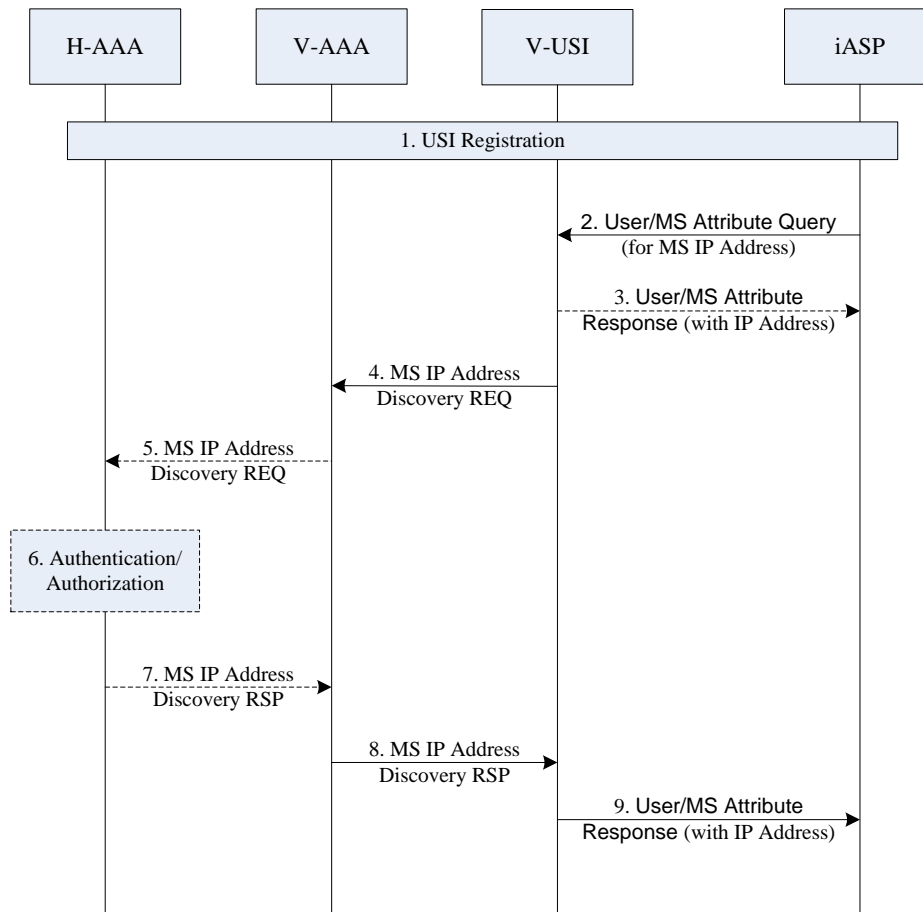
**Figure 25: MS IP Address Discovery (Non-Roaming)**

1. User/MS Attribute Request is used by the iASP to request the MS's current IP address from the USI System.
2. The USI system contacts the AAA with IP Address Discovery REQ.
3. Optionally, AAA may authenticate the IP Address Discovery REQ.
4. The AAA acquires the MS IP Address and responds to the USI with IP Address Discovery RSP.
5. The USI System responds to the iASP with the MS IP Address. User/MS Attribute Response is used by the USI System to respond to the IP Address Discovery Request from the iASP.

## USI

**7.3.2 IP Address Discovery in Roaming Scenarios**

For the roaming case of Local USI Services in Roaming, the procedure of MS IP Address Discovery is as following:



**Figure 26: MS IP Address Discovery (Local USI Services in Roaming)**

1. USI Registration procedure. The USI registration is triggered by iASP or network. The user's USI profile or registration information is delivered to V-USI by H-USI after the successful USI registration.
2. The iASP requests the MS IP Address from the V-USI by User/MS Attribute Request message. The MS's USI ID, e.g. short-lived USI ID or long-lived USI ID SHALL be included in this message to be used to identify the MS, if the Direct USI flow is used, and the short-lived USI ID SHOULD be used in priority if both USI ID are available.
3. The V-USI MAY authenticate the request and then respond to the iASP with the MS IP Address if V-USI already knows the MS IP address. In this case, steps 4-8 are not required.
4. V-USI contacts the V-AAA server for obtaining the MS IP Address. If the V-AAA has already knows the MS's IP Address, V-AAA MAY authenticate the request and then respond to V-USI with MS IP Address Discovery RSP by step 8, step 5-7 is skipped, else step 5 goes on.
5. The V-AAA forwards the IP Address Discovery Request to H-AAA.
6. H-AAA server authenticates the request.

## USI

7. H-AAA server responds to the V-AAA with the MS IP Address.

8. V-AAA forwards the MS IP Address to V-USI System.

9. The V-USI System responds to the iASP with the MS IP Address.

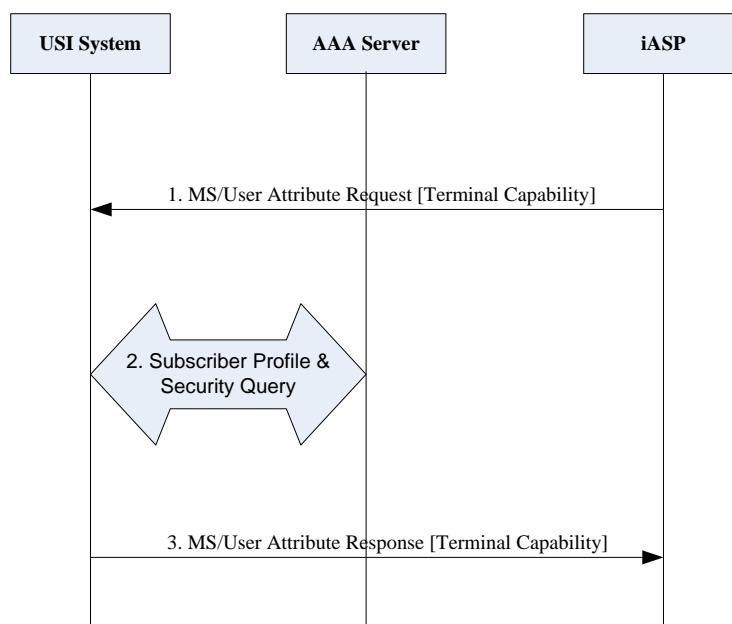
For the roaming case of Home USI Services in Roaming, the procedure of IP Address Discovery SHOULD be terminated within home network without the involvement of V-USI as the non-roaming scenario since the information of MS IP Address is generally maintained in the home network.

## 7.4 Dynamic MS Capability Retrieval

Once iASP has the device capability information of a device, it can manipulate data best fit to the device. The information MAY include the vendor, type, and model of the device, OS version, memory configuration, display characteristics, to name a few.

The way for the USI System to obtain and maintain the information is outside the scope of this spec, which MAY vary depending on the information types.

### 7.4.1 Dynamic Capability Retrieval in non-Roaming Scenario



**Figure 27: Dynamic Capability Retrieval (non-Roaming)**

- Over the U1 interface, an authenticated and authorized iASP sends a Device Information Request to the USI System for certain set of device capabilities of a specific device. Then, the USI System responds with a MS/User Attribute Request [Terminal Capability] to the iASP containing the requested type-value pair(s).

NOTE: Device Identity in the following primitives is subject to change depending on the decision made for the USI identification. In order to make Device Identity available to iASP, a certain way needs to be defined to deliver or pre-configure it to iASP prior to the MS/User Attribute Request.

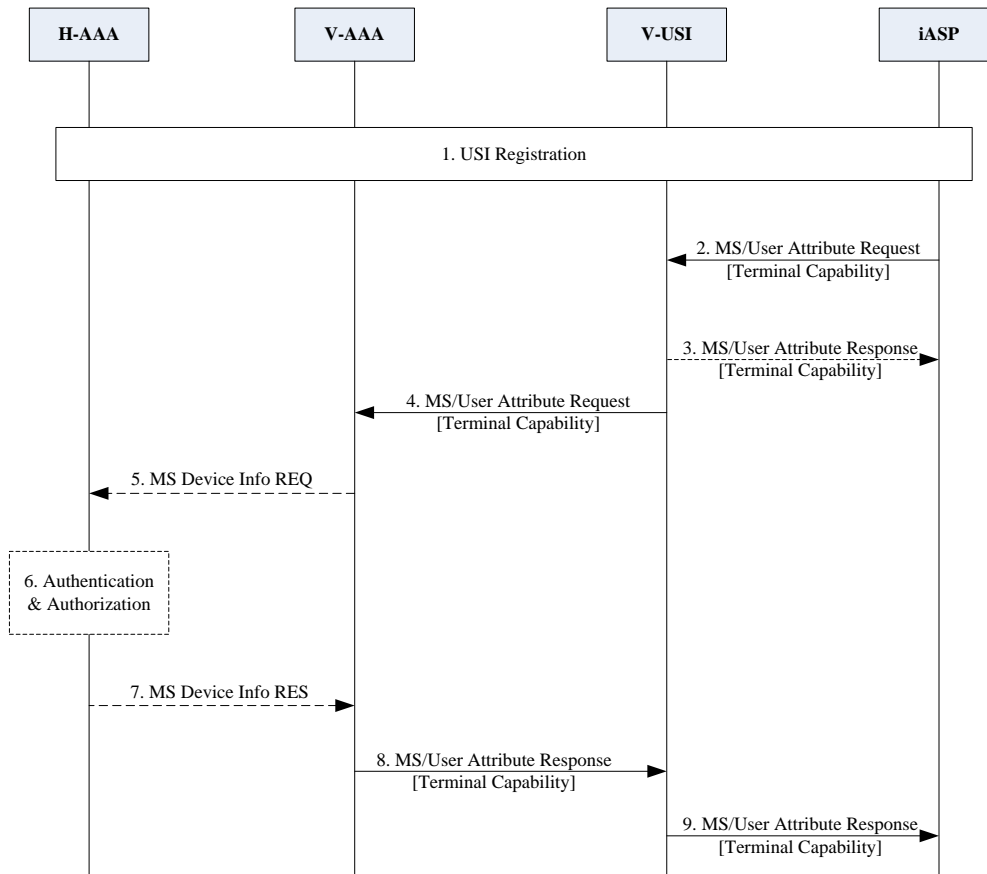
## USI

2. MS/User Attribute Request [Terminal Capability] is used by the iASP to request the USI System to retrieve certain device capability information of a specific device. The MS/User Attribute Request [Terminal Capability] includes the following Information Elements;
  - **USI Identity**, is an identity that is associated with the device for which information is being queried.
  - **iASP Identity**, which identifies the iASP that requests the device capability information.
  - **Requested Information Type(s)**, which identify one or more device capabilities requested by the iASP.
3. MS/User Attribute Response [Terminal Capability] is used by the USI System to respond to the MS/User Attribute Request [Terminal Capability] to inform the iASP of the requested device capability information. Device Information Response delivers at least the following Information Element;
  - **USI Identity**, which is the same Device Identity sent in the MS/User Attribute Request [Terminal Capability].
  - **Requested Information Type-Value Pair(s)** which corresponds to the type and value pair of the requested device capabilities

#### 7.4.2 Dynamic Capability Retrieval in Roaming Scenarios

For the roaming case where the USI services are provided by the local USI System, the procedure of MS Device Information Retrieval is as following:

## USI



**Figure 28: Dynamic Capability Retrieval (Local USI Services in Roaming)**

1. USI Registration procedure. The USI registration is triggered by iASP or network. The user's USI profile or registration information is delivered to V-USI by H-USI after the successful USI registration.
2. The iASP requests the MS Device Information from the V-USI by the MS/User Attribute Request [Terminal Capability] message. The MS's USI ID, e.g. short-lived USI ID or long-lived USI ID SHALL be included in this message to be used to identify the MS, and the short-lived USI ID SHOULD be used in priority if both USI ID are available.
3. The V-USI MAY authenticate the request and then respond to the iASP with the MS/User Attribute Response [Terminal Capability] if V-USI already knows the Terminal Capability Information which might be transferred to V-USI by H-USI as part of the USI registration information after the MS registers with H-USI or based on previous inquiry record, else Step 4 goes on.
4. V-USI contacts the V-AAA server for obtaining the Terminal Capability Information. If the V-AAA has already knows the Terminal Capability Information, V-AAA responds to V-USI with MS/User Attribute Response [Terminal Capability] by step 8, step 5-7 is skipped, else step 5 goes on..
5. The V-AAA forwards the MS/User Attribute Request [Terminal Capability] to H-AAA.
6. H-AAA server authenticates the request.
7. After subscriber profile and security query, HAAA reply V-AAA with Terminal Capability Information.



## USI

8. V-AASP forwards the MS Device Information to V-USI.

9. V-USI responds to the iASP with the Terminal Capability Information.

For the roaming case where the USI services are provided by the home USI System, the procedure of MS Device Information Retrieval SHOULD be terminated within home network without the involvement of V-USI as the non-roaming scenario since Terminal Capability Information is generally maintained in the home network.

## 7.5 Dynamic QoS Support

Driven by iASP, QoS for application is requested via U1 interface at any time and can be changed dynamically. Dynamic QoS or on-demand QoS through the USI is supported based on WiMAX Dynamic QoS Subsystem. Whenever there is a QoS request from iASPs, the USI System interwork with Dynamic QoS Subsystem. Consequently, the target terminal(s) will add, change, or delete dynamic service flows through the DSx message exchanges with the base station. Appropriate policy is maintained by the user profile and NSP policies. The appropriate accounting functions will segregate the proper messages to respective entities.

### 7.5.1 Web Service Operations provided by USI System

In this subsection, Web service operations provided by USI System are described with detailed procedures. These operations are used to create, modify, terminate, and query USI QoS Sessions.

#### 7.5.1.1 USI QoS Session Creation

When an iASP decides to provide a dynamic QoS for a terminal (e.g., because the user asked a video clip with a QoS option), it will initiate a USI QoS Session creation by using the “QoS\_Session\_Creation” Web Service operation of USI System which will trigger IP-CAN session modification. Figure 29 shows the procedure of the USI QoS Session creation.

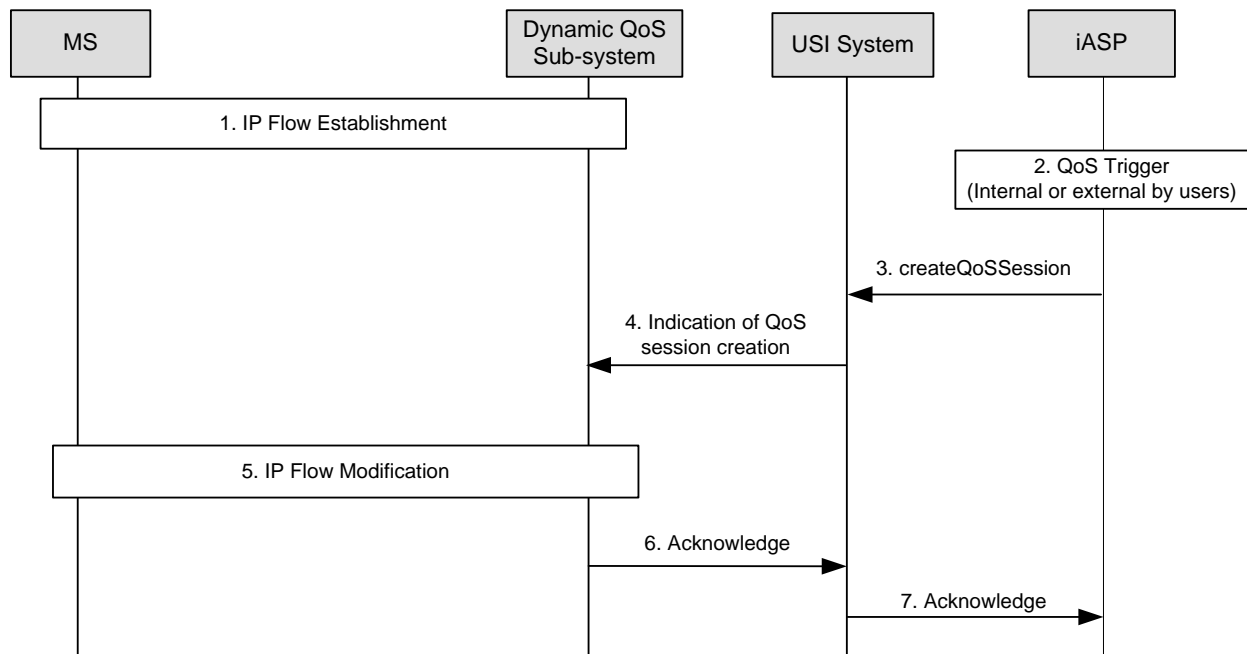


Figure 29: USI QoS Session Creation

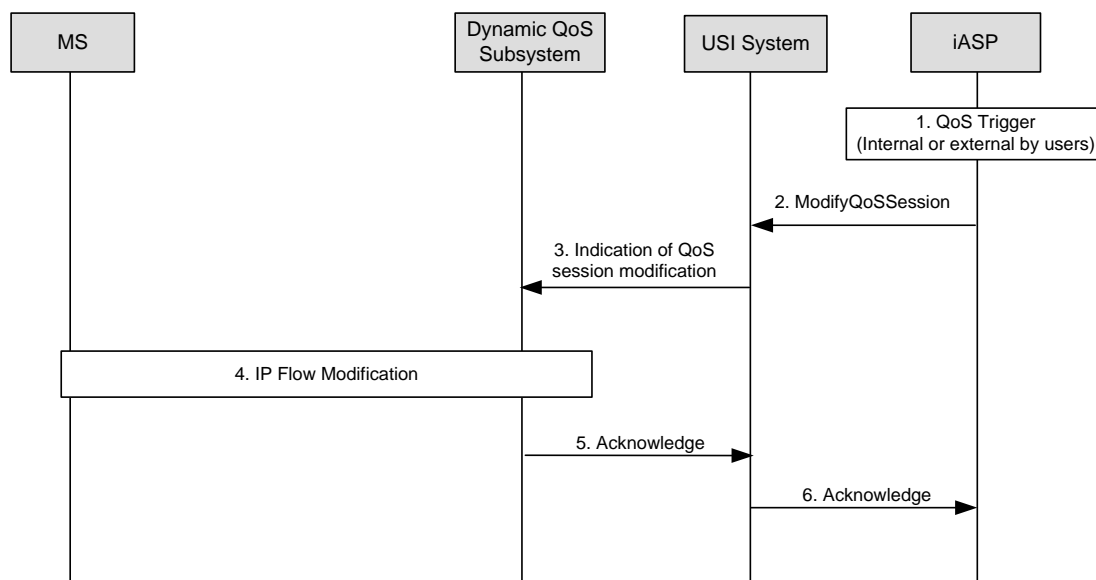
## USI

1. Initiated by the IP address allocation to the terminal, an IP-CAN session is established and the relevant PCC procedures are completed.
2. The iASP receives an internal or external trigger to set-up of a new application session for QoS. For example, the user requests a video clip with a guaranteed QoS option from the Web server of the iASP. Optionally, without user interaction, a trigger event in the iASP system MAY cause the iASP system to determine that a new QoS service is required for the user, e.g., timed QoS service.
3. The iASP requests a new USI QoS Session for the user via U1 interface. This request SHOULD include the following information:
  - a. The end user identifier for which the QoS is being request, and optionally the iASP identifier.
  - b. The QoS feature identifier and QoS attributes.
  - c. Application charging identifier, if required.
4. The USI System is required to establish a USI QoS Session and provides relevant service information to the PCC system.
5. The PCRF makes the authorization and policy decision based on subscription-related information stored in the PCRF during the IP-CAN session establishment. The IP-CAN session modification procedure or IP-CAN Bearer establishment procedure is performed as specified in [PCC]
6. The PCC system stores the service information and responds with an Acknowledgement to the USI System.
7. The USI System also stores the information and sends ACK to the iASP including the result of the request, the unique USI QoS session request identifier, and the current value of the configurable QoS parameters. The iASP system MAY show the result of the QoS request to the user.

**7.5.1.2 USI QoS Session Modification**

A USI QoS Session can be modified when a user (or iASP itself) wants to change level of QoS or flow information, or add media flows. The “QoS\_Session\_Modification” Web Service operation is used for this purpose. Figure 30 is applicable if a USI QoS Session is being modified by the iASP.

## USI



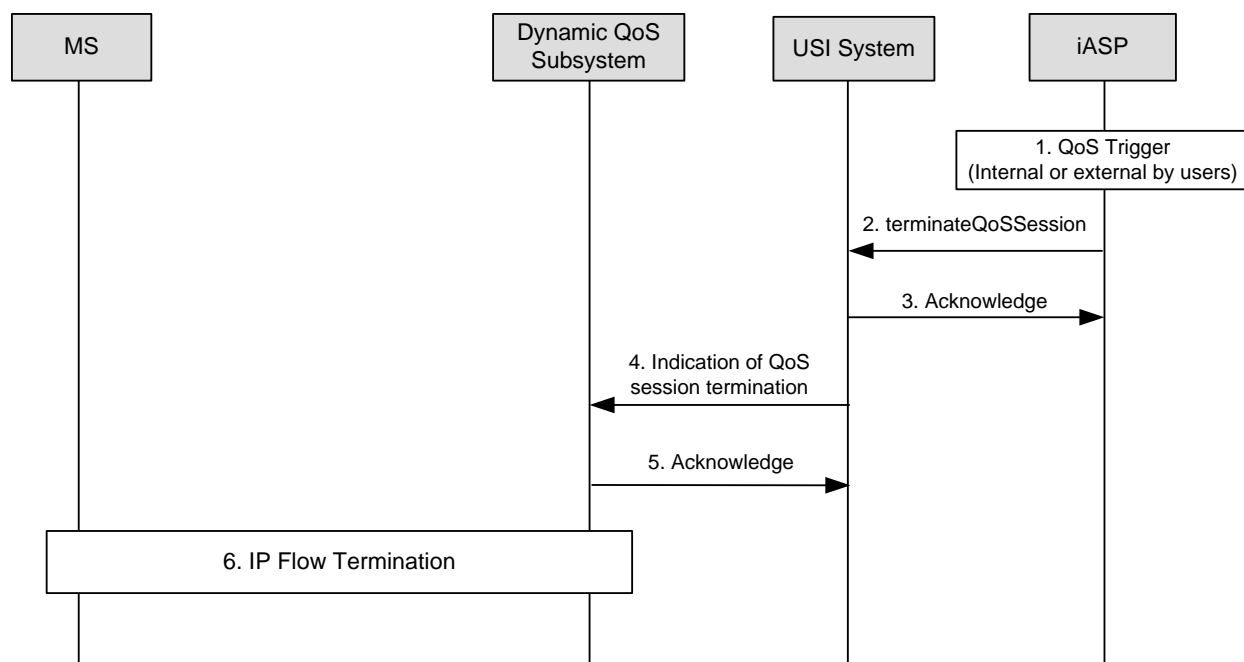
**Figure 30: USI QoS Session Modification**

1. The iASP receives an internal or external trigger to change the previous application session for QoS. For example, the user requests the better quality for the video clip being played from 128kbps to 384kbps. Optionally, without user interaction, a trigger event in the iASP system MAY cause the iASP system to determine a change of the QoS service is required for the user, e.g., adding an advertisement video clip to the QoS application session.
2. The iASP requests a change of the previous USI QoS session for the user via U1 interface. This request SHOULD include the unique USI QoS request identifier and specify the new values for the QoS attributes.
3. The USI System indicates that the USI QoS Session is being modified and provides relevant service information to the PCC system.
4. The IP-CAN session modification procedure is performed as defined in [PCC].
5. The PCC system stores the service information and responds with an Acknowledgement to the USI System.
6. The USI System also stores the information and sends an ACK to the iASP with the result of the request for the configurable QoS parameters. The iASP system MAY show the result of the QoS request to the user.

### 7.5.1.3 USI QoS Session Termination

A USI QoS Session is terminated when the “QoS\_Session\_Termination” Web Service operation in the USI System is executed by the iASP and results in termination of PCC IP-CAN bearer(s). This can be triggered by users or the iASP system itself, for example, because a user wants to stop QoS service from the iASP. (Generally, if all QoS flows associated with a USI QoS Session are being removed, iASP decides to terminate the USI QoS Session.). The following figure shows the procedure of USI QoS Session termination.

## USI

**Figure 31: USI QoS Session Termination**

1. The iASP receives an internal or external trigger for a session release.
2. The iASP sends a USI QoS Session termination request, `QoS_Session_Termination`, to the USI System to request the removal of the session. This request SHOULD include the unique USI QoS request identifier.
3. The USI System removes information related to the session and acknowledges to the iASP.
4. The USI System indicates to the PCC system that the USI QoS Session is being terminated.
5. The PCRF identifies the affected IP-CAN sessions where PCC rule(s) for the IP flow(s) of this USI QoS Session are installed. These PCC rules need to be removed. The PCC system modifies IP-CAN session and acknowledges to the USI System.
6. The PCRF starts the IP-CAN session modification resulting in the IP-CAN bearer termination as defined by [PCC]

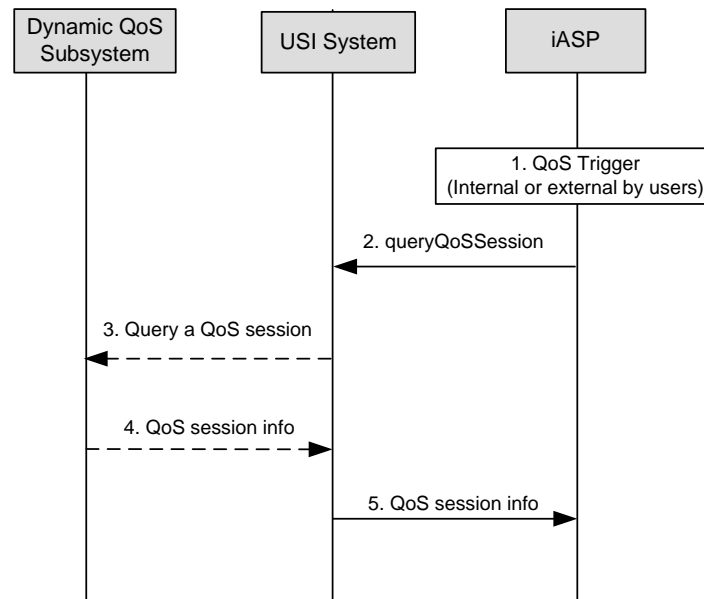
NOTE: In USI QoS Session termination procedure, acknowledgment is sent without waiting for the subsequent requests and responses. This is because possible retrying for termination function is a role of the receiver. For example, the user is not requested to retry for stopping the service.

**7.5.1.4 USI QoS Session Query**

When an iASP needs to check whether or not the previously created USI QoS Session is active and is properly provided, the “`QoS_Session_Query`” Web Service operation is utilized. This procedure is particularly useful when changes made by either iASP or WiMAX could not be applied to the peer due to the disconnection between USI System and iASP. States mismatch can also happen if an iASP implements only a Web Service client and thus it cannot receive any notification from the USI System for network failure (e.g., deletion of service flows during a handover). The following figure depicts the procedure of USI QoS Session query.

## USI

1



2

3

**Figure 32: USI QoS Session Query**

4

- 5 1. The iASP receives an internal or external trigger to query whether a USI QoS Session is
- 6 active or not and whether there have been any changes. For example, after the iASP system
- 7 restarted, it MAY want to ask the USI System the status of USI QoS Sessions previously
- 8 active.
- 9 2. The iASP sends a USI QoS Session query request with the USI QoS request identifier
- 10 and/or the User Identity to the USI System.
- 11 3. Optionally, if the USI System does not maintain up-to-date information for USI QoS Sessions,
- 12 e.g., because the USI System acts as only a client for the PCRF and thus it cannot receive
- 13 any events from the PCRF, it forwards the query to the PCRF.
- 14 4. The PCRF identifies the associated policy rules with the USI QoS Session and/or session
- 15 information regarding the identified user and sends information for the USI QoS Session to
- 16 the USI System.
- 17 5. The USI System stores the received information and forwards it to the iASP.
- 18

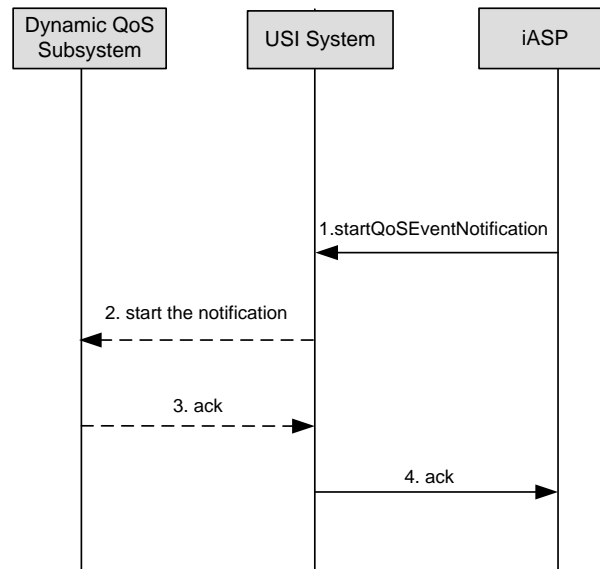
## USI

**7.5.1.5 USI QoS Start Notification**

USI QoS Start Notification procedure is initiated by the iASP to inform the USI of its interest in receiving notifications related to certain events related to a specific user or to a list of user. The information the iASP has to provide with the “QoSStartNotification” invocation is at least:

- An end user identifier or list of end user identifiers
- iASP endpoint interfaces where the iASP like to receive notification events
- The events that the iASP is interested to.

Then the USI will return to iASP the result of the request (ack), and a correlator to match the notification



**Figure 33: USI QoS Start Notification**

**7.5.1.6 USI QoS Stop Notification**

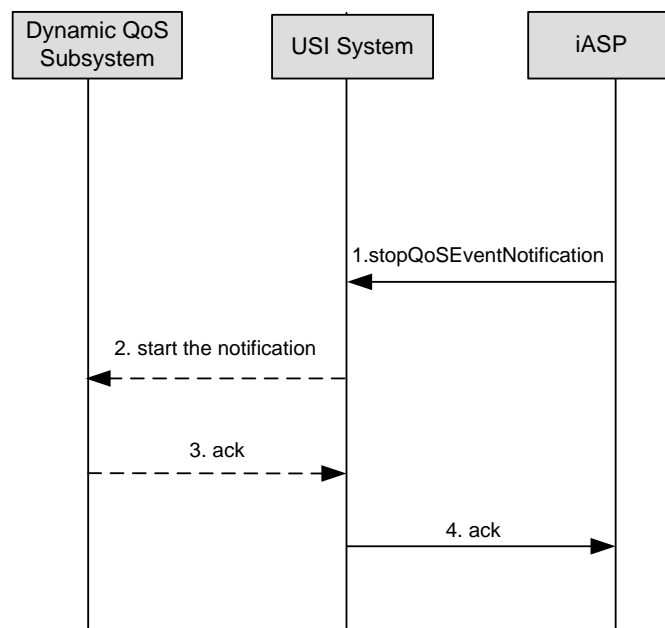
The USI QoS Stop Notification procedure is initiated by the iASP to stop receiving notifications regarding events related to a particular USI QoS session.

The information the iASP has to provide with the “QoS\_Stop\_Notification” invocation is at least:

- The identifier used to identify the notification registration

Then, the USI returns to iASP the result of the request (ack).

## USI



**Figure 34: USI QoS Stop Notification**

1. The iASP requests to stop a previously requested notification registration
2. The USI System could interact with QoS subsystem in order to start the event notification
3. The QoS subsystem send back an ack,
4. The USI send an ack with the result of the start notification operation.

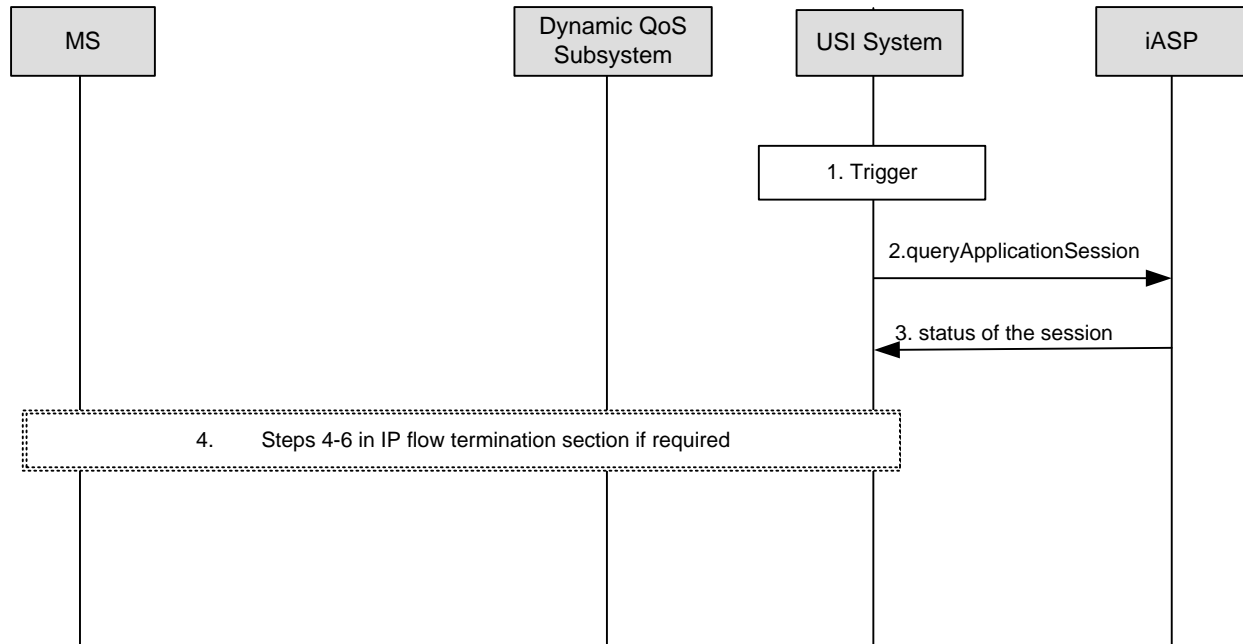
## 7.5.2 Web Service Operations provided by iASP

In this subsection, Web service operations provided by iASP are described with detailed procedures. These operations are used to notify network events (e.g., terminal disconnection) and to query the status of application sessions. These operations are only applicable if the iASP acts as a server as well as a client in the Web service interface.

### 7.5.2.1 Application Session Query

When an USI System needs to check whether or not an application session is still active, the “Application\_Session\_Query” Web Service operation is executed. This service is particularly useful when changes made by iASP could not be applied to the USI System due to the disconnection between USI System and iASP. For example, if the USI System misses a USI QoS Session termination request, resources allocated to the USI QoS Session will not be released and so charged. This procedure is only possible when the iASP implements such web service operations as a server as shown in Figure 35.

## USI



**Figure 35: Application Session Query**

1. The USI System receives an internal or external (from the PCRF) trigger to query whether an application session of the iASP is active or not. For example, the USI System MAY want to ask the iASP system the status of the application session which USI QoS Session has been active for 24 hours.
2. The USI System sends an application session query request with the USI QoS Session identifier to the iASP.
3. The iASP identifies and checks the corresponding application session, and sends the USI System a response with activity of the session.
4. Optionally, if the application session is unknown (i.e., already closed), the USI System initiates the termination of the USI QoS Session and this procedure is the same as Steps 4-6 in Figure 31.

#### 7.5.2.2 Event Notification

If there happens any change or failure for service flows associated with a USI QoS Session and the event is detected by USI System, it notifies the corresponding iASP to reflect the change. The “Event Notification” Web Service operation is utilized for this purpose. This procedure is only possible when the iASP implements such web service operations as a server as shown in Figure 36.



## USI

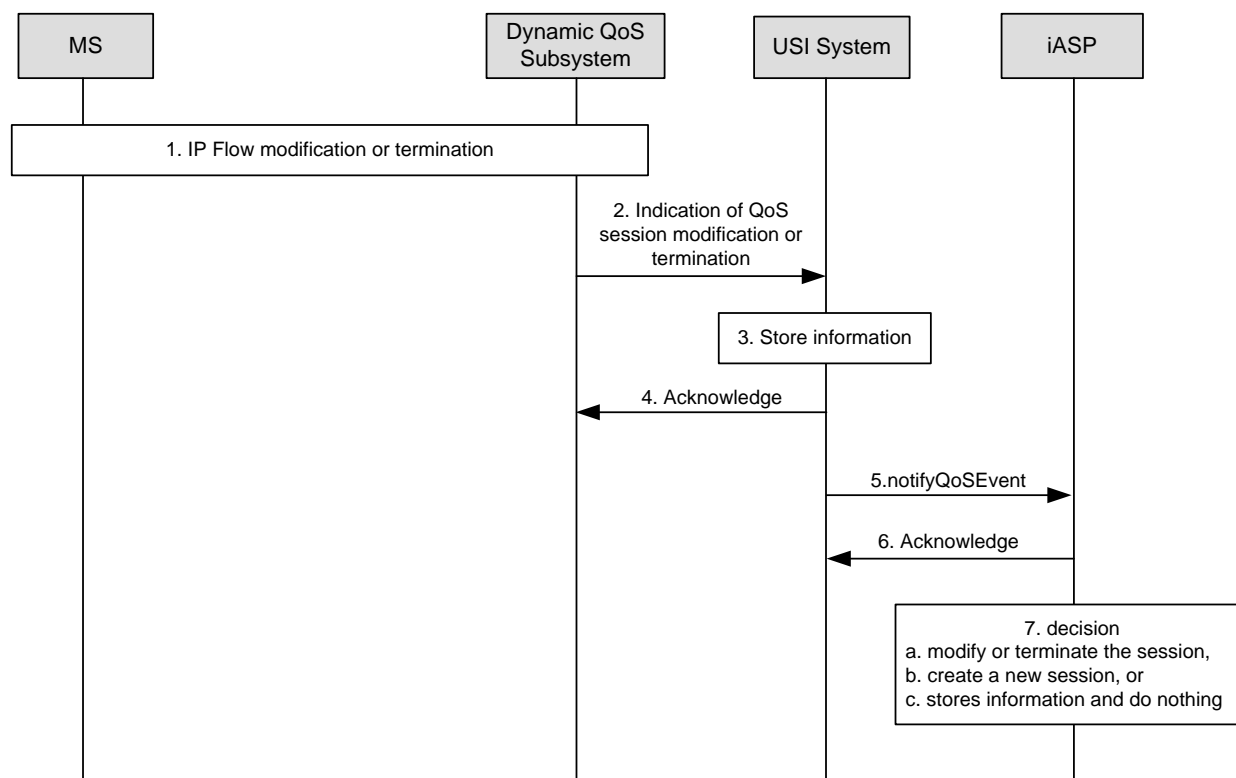


Figure 36: Event Notification

1. IP-CAN session modification or termination occurs initiated by the user, or the PCC system. For example, one or more service flows provisioned to the terminal can be removed or its bandwidth can be reduced, or even the terminal can be disconnected during a handover.
2. The PCC system identifies the affected USI QoS Sessions due to the IP-CAN session modification or termination and MAY indicate such events to the USI System.
3. The USI System stores the received information and identifies the affected iASP application sessions.
4. The USI System acknowledges to the PCC system.
5. The USI System sends an event notification request to the iASP. This event can be the removal of service flow(s), changes of QoS, disconnection of the terminal, and so on. The USI System SHOULD include the unique USI QoS session identifier and specify the event.
6. The iASP acknowledges to the USI System.
7. The iASP decides what to do for the application session. Depending on the characteristics of the application and the event received, it MAY modify or terminate the associated USI QoS Session(s), or create a new USI QoS Session as it was.

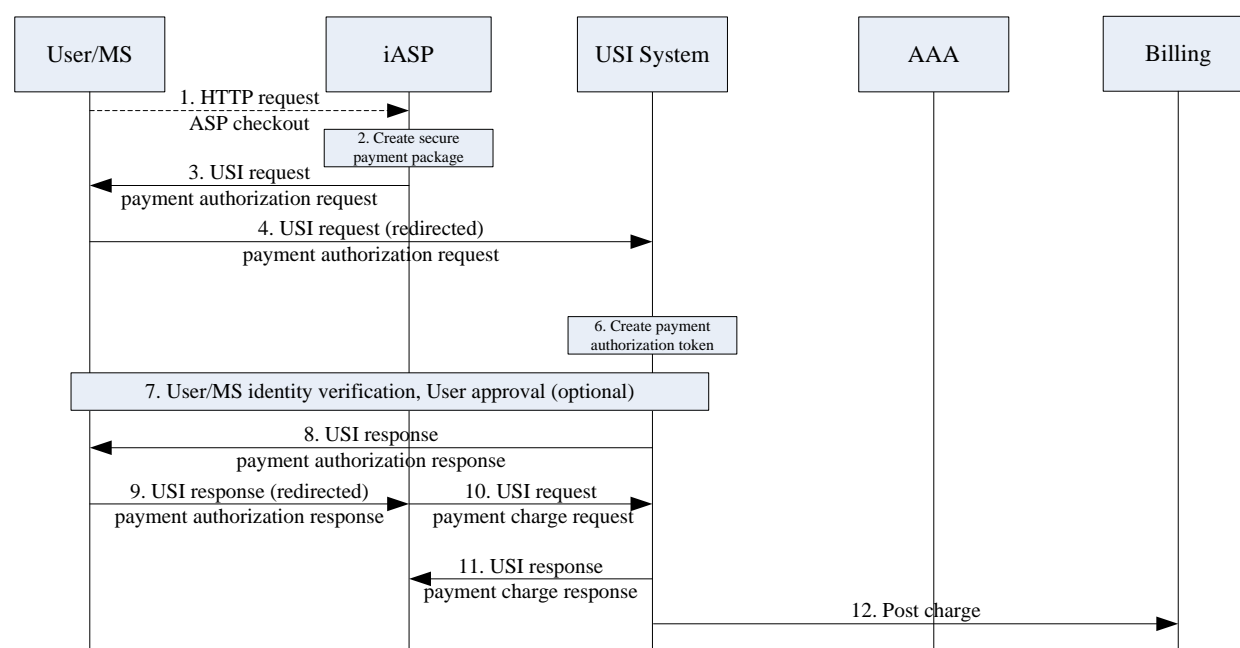
NOTE: For step 7: It SHOULD be noted that, in case of communication application like “Click to VoIP” where two or more users are involved, modification or termination of a single USI QoS Session would impact the other party’s USI QoS Session(s) which are only known by the application provider.

## USI

## 7.6 E-Payment

E-Payment is an optional feature that enables the user on a USI-enabled WiMAX network to send secure electronic payment to any iASP. This electronic payment could be posted into the user's monthly access charges from his/her WiMAX service provider. This technology could be used to enable several use cases where a user on a small mobile device (with limited input/output capability) could purchase airline tickets online and have the charges posted on his mobile bill. Another example is exchange of electronic money between friends (e.g. to split the restaurant check). In this process, the service provider (his designated host of the service) could collect a small percentage of the transaction as the service fee.

Figure 37 defines the electronic payment (e-payment) USI procedure. The E-Payment flow is composed of two parts: The first is transaction authorization and user identification and approval, which is done by submitting an E-Payment Authorization Request using the User/MS redirected flow, while the second part is handling the charging operation by submission of an E-Payment Charge Request using direct the flow.



**Figure 37: E-Payment Procedure Using USI**

1. The user triggers an e-payment transaction with the iASP, using means outside scope of USI.
2. The iASP requests a payment authorization request (token) from the USI System by submitting a USI request. The request MAY be signed by iASP.
- 3-4. The USI request is redirected to the USI System by User/MS using a secure communication (e.g. HTTPS).
5. USI System communicate with Billing and AAA servers to authorize the transaction based on user profile
  - The AAA and/or Billing server verify the transaction is authorized based on the subscriber's profile. As a minimum, the following must be verified:
    1. User is subscribed to the USI E-Payment service
    2. User profile allows for financial transaction (amount, etc) matching with the payment transaction details.

## USI

3. The iASP is known and authorized to use USI E-Payment service.

6. If processing is successful, the AAA and/or Billing server generates an accept response that is returned to USI System. USI System proceeds to create a Payment Authorization Token. The Payment Authorization Token parameters are described in 10.4.2. In particular the token contains a one time use transaction identifier. When the transaction identifier is used for charging it is flushed from the USI System. The transaction identifier prevents repeated charging requests by the iASP

7. USI System verifies (authenticates) the User/MS identity as described in **Error! Reference source not found.** and optionally obtains a user's final confirmation for the transaction.

NOTE: This step MAY take place before step 4. The USI server MAY refuse to accommodate specific User/MS authentication methods, which MAY be considered unsuitable for a specific E-Payment transaction.

8. The Payment Authorization Response (Token) is sent back to the User/MS via the same secure communication tunnel between the MS and the USI Server formed in step 4.

9. The Payment Authorization Response (Token) is redirected to the iASP by User/MS

10. The iASP can now send Payment Charge Request as described in 10.4.3. The payment charge request only takes place after goods are shipped-to or received-by the user. Multiple charging requests MAY happen for the same transaction identifier if partial charging is required (e.g. when part of goods is delivered).

11. Upon receipt of this message, the USI System can look up the transaction identifier validity. If the transaction number presented does not exist, the e-payment process will be rejected. If the token number is valid, the USI System looks up the USI User Identity associated with the transaction number and constructs the Charge Response which includes parameters specified in 10.4.4.

12. The USI Server posts a charge to the billing system for the user.

## 7.7 Live Streaming

### 7.7.1 Live Streaming Service types

Two broad categories of live streaming services are possible over WiMAX.

- a) Live streaming for/by the WiMAX operator: service specifically tailored for/by the NSP for a WiMAX end host. This is a typical walled garden approach.
- b) Live streaming on the internet: service NOT tailored for a WiMAX end host. This can be targeted and delivered through USI.

### 7.7.2 Motivation for Live Streaming Delivery through USI

Several live streaming and live PPV services available on the internet today. Most of these services though multicast in nature, are often delivered as multi-unicast to the end hosts. This leads to inefficient use of last mile bandwidth and especially does not scale well with wireless last miles

Hence, we need an efficient mechanism to bring all of live streaming services on the internet to the WiMAX end host. Enabling internet based live streaming through USI can solve this problem of efficiency on the wireless link since the USI System can be the focal point of such live streaming traffic from the internet and can deliver such traffic through the existing live streaming data path within the NSP/NAP.

### 7.7.3 Live Streaming Interactions on U1

The following interactions are expected on the U1 interface:

## USI

a) Live streaming control/signaling information on U1-Control

b) Live streaming data information on U1-Data

### 7.7.4 Live Streaming Control on U1-Control

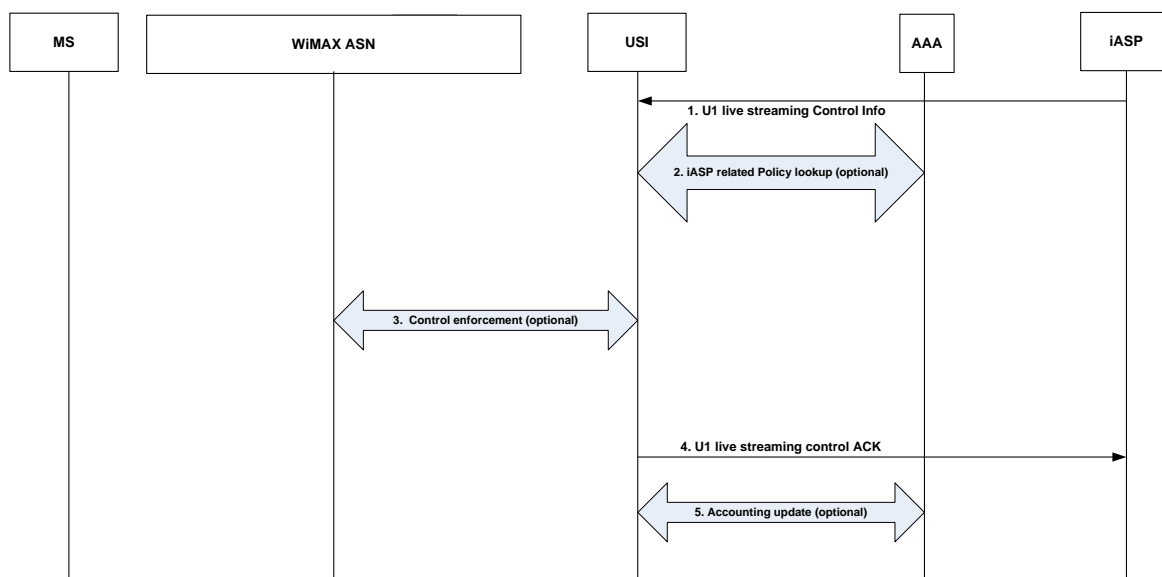
Following are the key control information on U1-Control

a) Service guide delivery (iASP → USI) or request (USI → iASP): this contains the service/program guide of the iASP. This is an optional procedure as the iASP MAY choose not to deliver the service guide apriori to USI

b) Security context delivery (iASP → USI) or request (USI → iASP): example: DRM enabled or not; if yes, DRM related context exchange.

c) QoS requirements for the service (iASP → USI) or request (USI → iASP). E.g.: Favorable Peak/Avg rates; tolerable latencies. This is an optional procedure.

The Live Streaming control flow on U1 is shown below.



**Figure 38: Live Streaming Control Flow on U1-Control**

1. The iASP signals the live streaming related control information (e.g.: service guide/QoS information etc) on U1. Alternatively the USI System can request such control information from iASP in a prior step for which the iASP can respond as shown in step-1.

2. The USI System MAY lookup related policy information from the AAA with regards to the iASP. This is an optional procedure and could involve operations like authenticating the iASP, authenticating the control information on U1 is within policy etc.

3. The control MAY be enforced within the NAP and/or NSP (for example QoS enforcement as decided by MCBSC subteam)

4. The USI System responds to the iASP with an ACK

## USI

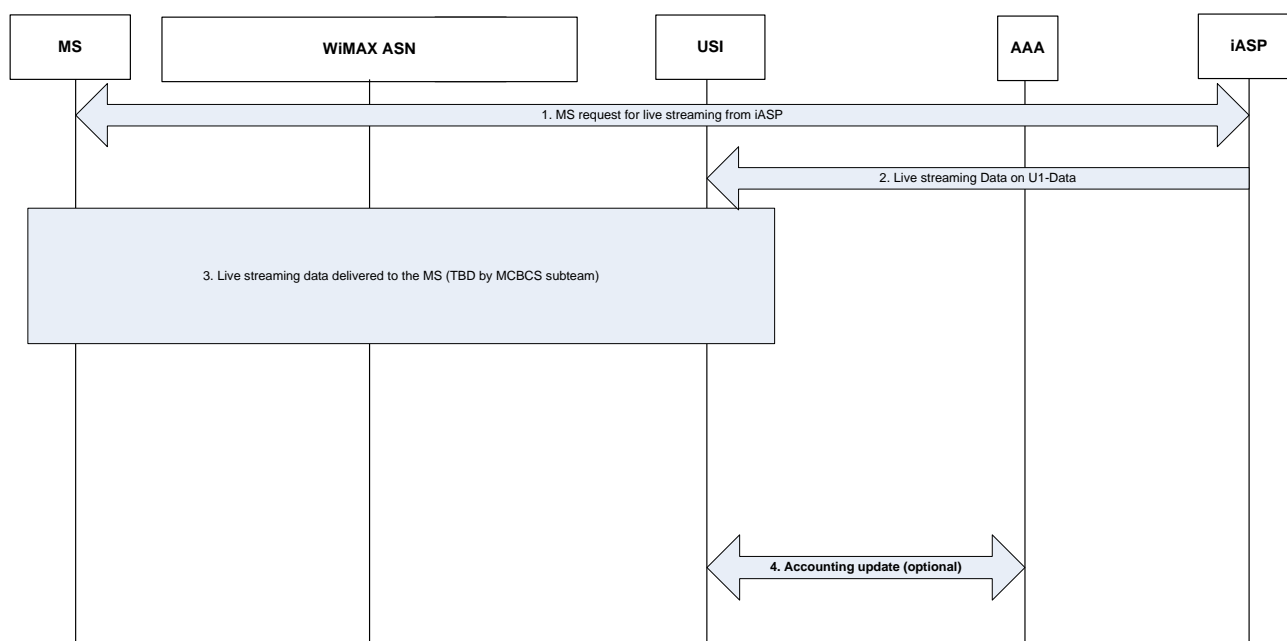
5. The USI System MAY perform an accounting update with regard to the iASP.

### 7.7.5 MS joining live streaming session

Two cases are considered here:

- a) The USI server has the context of the live streaming session from the iASP
- b) The USI server does not have the context of the live streaming session for iASP

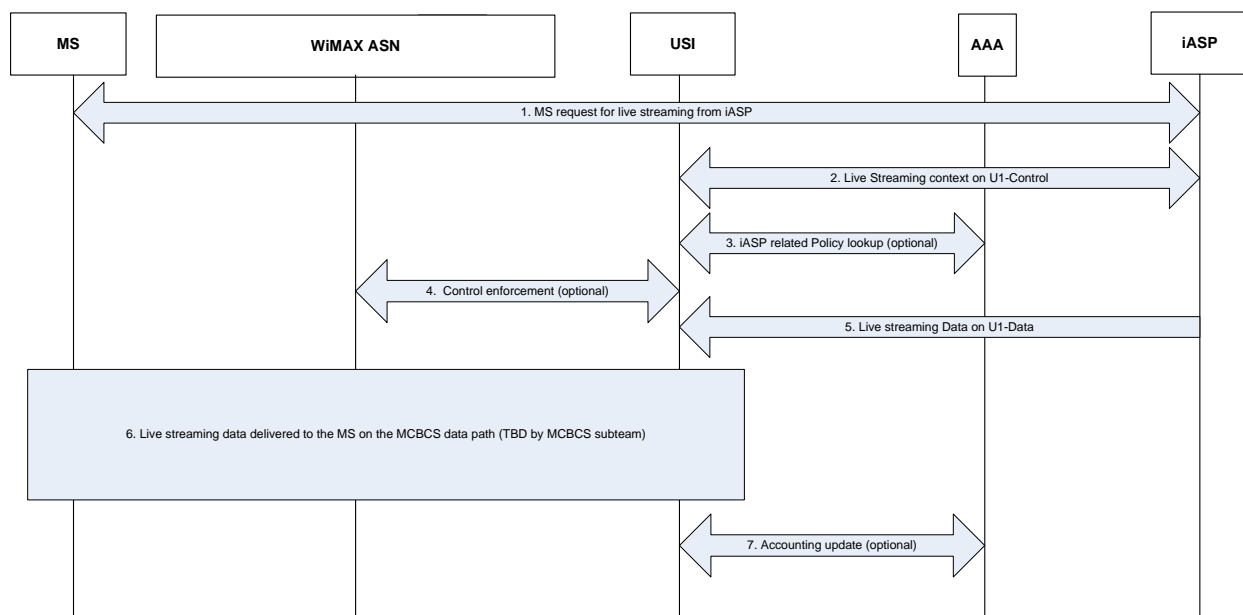
#### 7.7.5.1 USI server has the context of the live streaming session from iASP



**Figure 39: Live Streaming Data Path Setup on U1-Data (USI has iASP context)**

1. MS sends HTTP request for a particular multicast session to the iASP and iASP responds with an ACK indicating that the multicast data will be available via USI. This is out of scope of USI.
2. USI server has the iASP context already. Hence, iASP directly sends the live streaming data to the live streaming server in the USI System on U1-Data
3. Live streaming data is delivered to the MS via the MCBCS data path that achieves better efficiency (To be defined by the MCBCS work in NWG) – as opposed to a non-efficient multi-unicast data path that would be used if this live streaming data came directly from the internet to the MS.
4. The USI System MAY perform an accounting update with regard to the MS and/or the iASP (optional)

## USI

**7.7.5.2 USI server does not have the context of the live streaming session from iASP****Figure 40: Live Streaming Path Setup on U1-Data (USI does not have iASP context)**

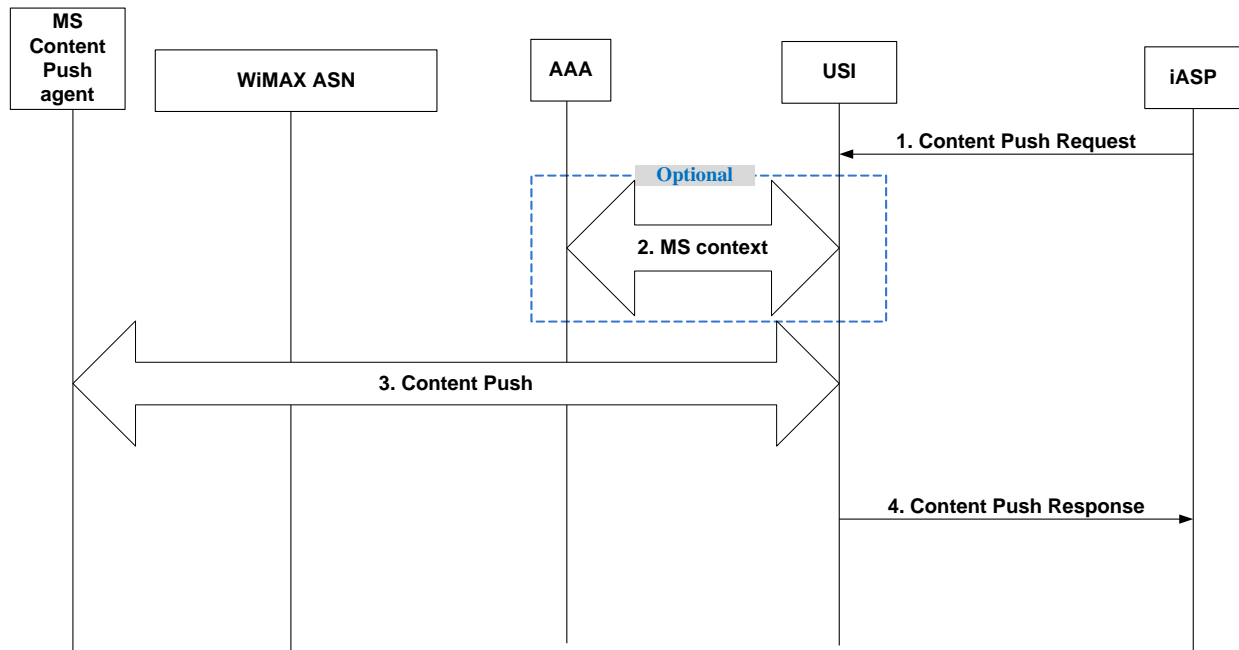
1. MS sends HTTP request for a particular multicast session to the iASP and iASP responds with an ACK indicating that the multicast data will be available via USI. This is out of scope of USI.
2. iASP contacts the USI server to deliver the live stream context.
3. USI MAY perform iASP related policy lookup with AAA (optional)
4. USI MAY enforce the control information in the serving ASN (optional)
5. iASP now sends the live streaming data to the live streaming server in the USI System on U1-Data
6. Live streaming data is delivered to the MS via the MCBCS data path that achieves better efficiency (To be defined by the MCBCS work in NWG) – as opposed to a non-efficient multi-unicast data path that would be used if this live streaming data came directly from the internet to the MS.
7. The USI System MAY perform an accounting update with regard to the MS and/or the iASP (optional)

**7.8 Content Push**

Content Push service enables iASP originated push of data into the MS, in a secure, controlled manner

**7.8.1 Content Push in Non-Roaming Scenario**

Figure 41 shows the flow for content pushing service from the USI perspective.



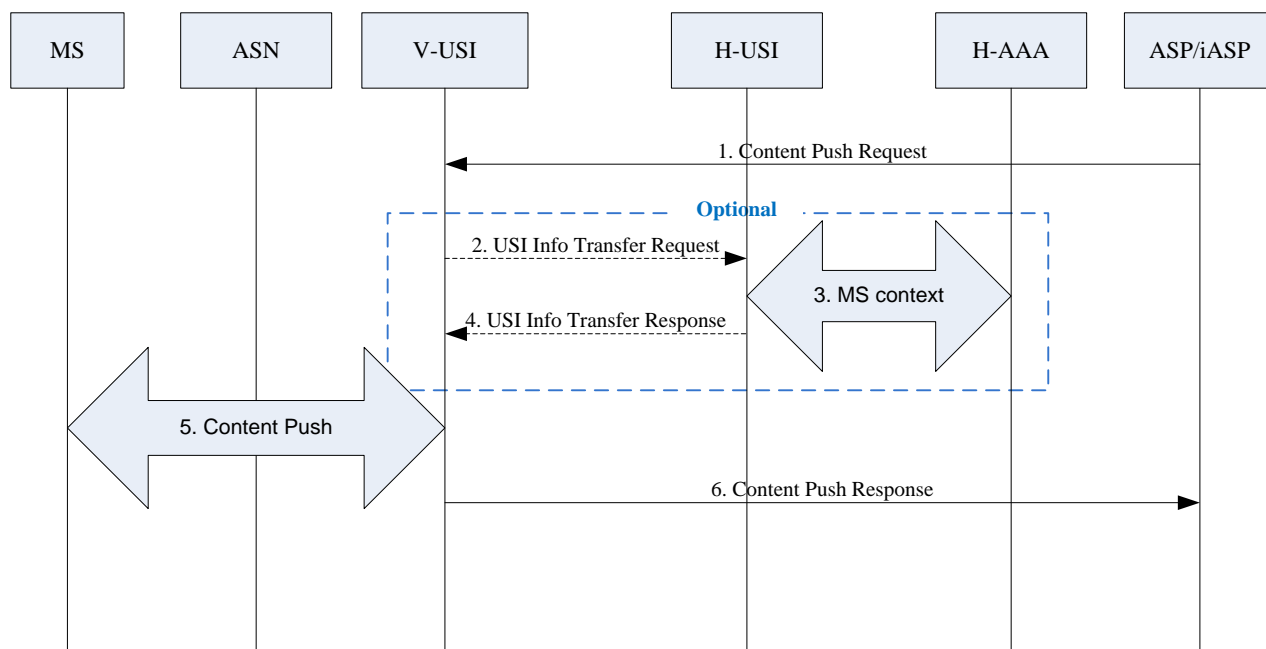
**Figure 41: Content Push flow (non-Roaming)**

1. ASP submits a Content Push Request to the USI System
2. The USI System optionally interact with AAA to retrieve and enforce policies for Content Push request,
3. If the request authorization is successful, the USI System pushes the content (data) to the MS's Content Push agent as described in 7.8.4
4. The USI System returns a Content Push Response to iASP with result code

## 7.8.2 Content Push in Roaming Scenarios

For the roaming case of Local USI Services in Roaming, the procedure of determining MS location is as following:

## USI



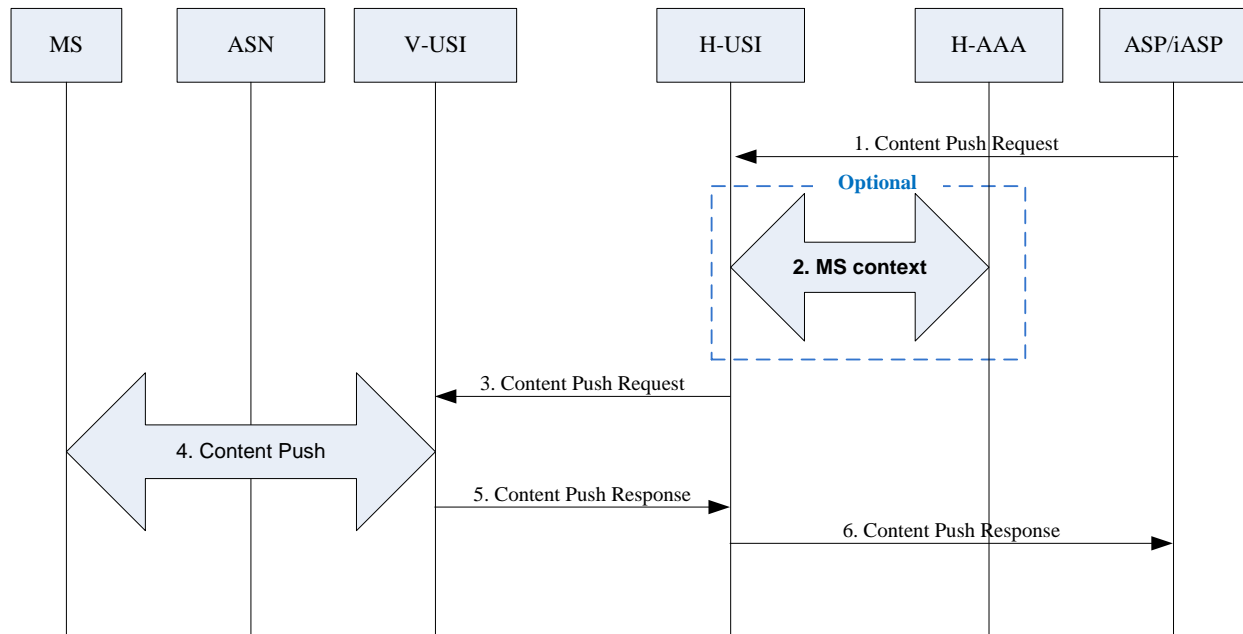
**Figure 42: Content Push flow (Local USI Services in Roaming)**

1. iASP submits a Content Push Request to Visited USI System
2. If the Visited USI System does not have a suitable authorization for the Content Push service request and/or the User/MS identity, then it contacts the Home USI System with a USI Info Transfer Request, and steps 3 and 4 take place.
3. The Home USI System interacts with AAA to authorize the Content Push request
4. The Home USI System return authorization and other required information to Visited USI System.
5. The Visited USI System pushes the content (data) into the MS's Content Push agent
6. The Visited USI System returns a Content Push Response to iASP, including a result code

For the roaming case of Home USI Services in Roaming, the procedure of determining MS location is as following:



## USI



**Figure 43: Content Push flow (Home USI Services in Roaming)**

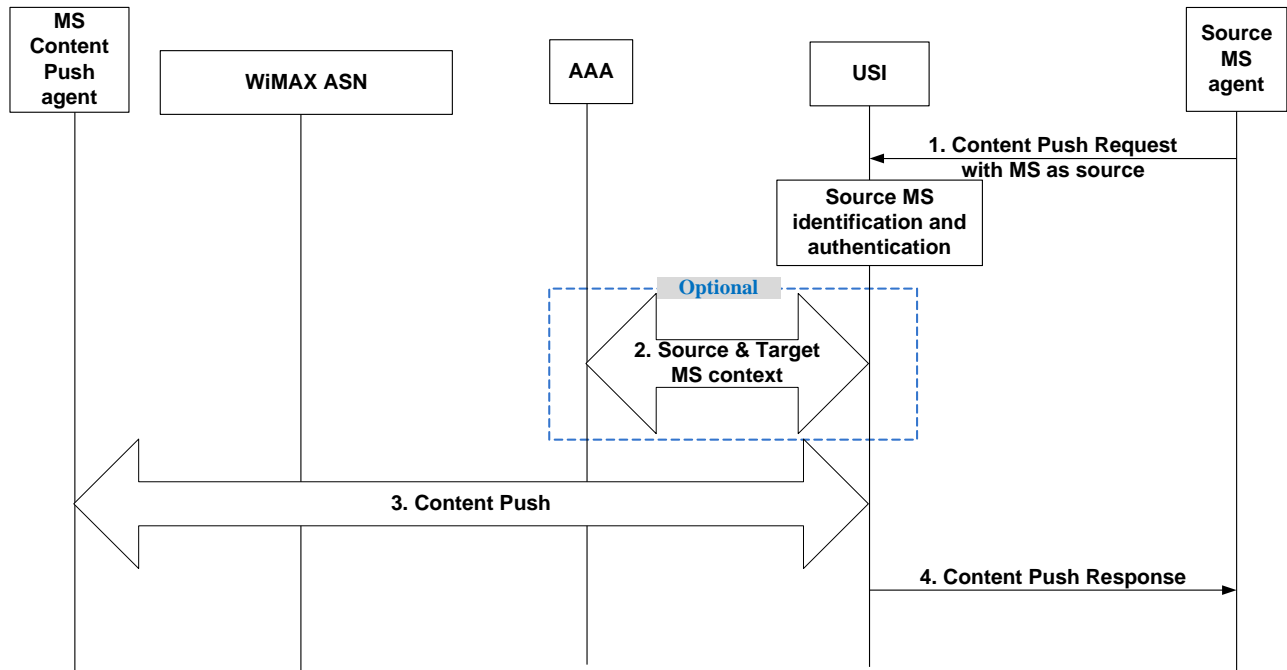
1. iASP submits a Content Push Request to Home USI System.
2. The Home USI System authorizes the Content Push request. This may optionally require interaction with Home AAA.
3. The Home USI System forwards the USI Content Push Request and authorization to the Visited USI System.
4. The Visited USI System pushes the content (data) into the MS's Content Push agent
5. The Visited USI System returns a Content Push Response to Home USI System, including a result code
6. The Home USI System returns a Content Push Response to iASP with result code

### 7.8.3 Direct MS-to-MS Content Push

Figure 44 shows it is possible for MS to use the Content Push service directly, without a need for an iASP mediator service. The flow is similar to the iASP initiated Content Push flow, with following changes:

1. The MS must provide its own identity instead of that of iASP.
2. The authorization check must include both Source MS and Target MS profiles and policies.

## USI



**Figure 44: MS-to-MS Content Push flow (Non-Roaming)**

#### 7.8.4 Transport of Content from USI server to MS

This specification defines one way of delivering the Content from USI server to MS:

The entire <ContentPushRequest> structure is transported over a UDP message using port [Add Port defined by IANA]. The content is not encrypted beyond whatever the WiMAX network provides to IP/UDP packets sent to the MS. The USI requestor may further encrypt the content as required by the particular application.

Due to network limitations on size of UDP packets, content senders should use the <ContentURL> for large content transfer.

## 8. Security Aspects

USI SHALL protect NSP against security attacks (e.g. denial of service, reply). iASP and NSP SHALL be able to mutually authenticate each other. USI interface (i.e. U1) SHALL ensure privacy, integrity, and non-repudiation of the exchanged messages.

### 8.1 iASP authentication during processing of USI request

At times, iASP's identity must be authenticated by NSP. iASP may place an authorization token into the USI request, allowing the NSP to prove which iASP has authorized the USI request.

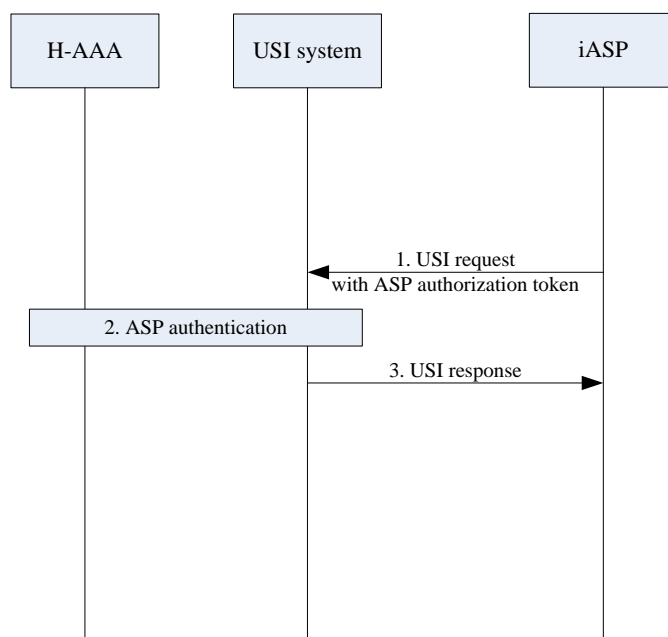


Figure 45: iASP authentication using authorization token

There are two methods of providing authentication information the iASP can use, described in this section.

#### 8.1.1 iASP authentication and USI request authorization using digital signature

The authorization token contains a digest or digital signature of USI request or parts thereof and is signed by iASP using an iASP certificate with known CA (e.g. WiMAX forum).

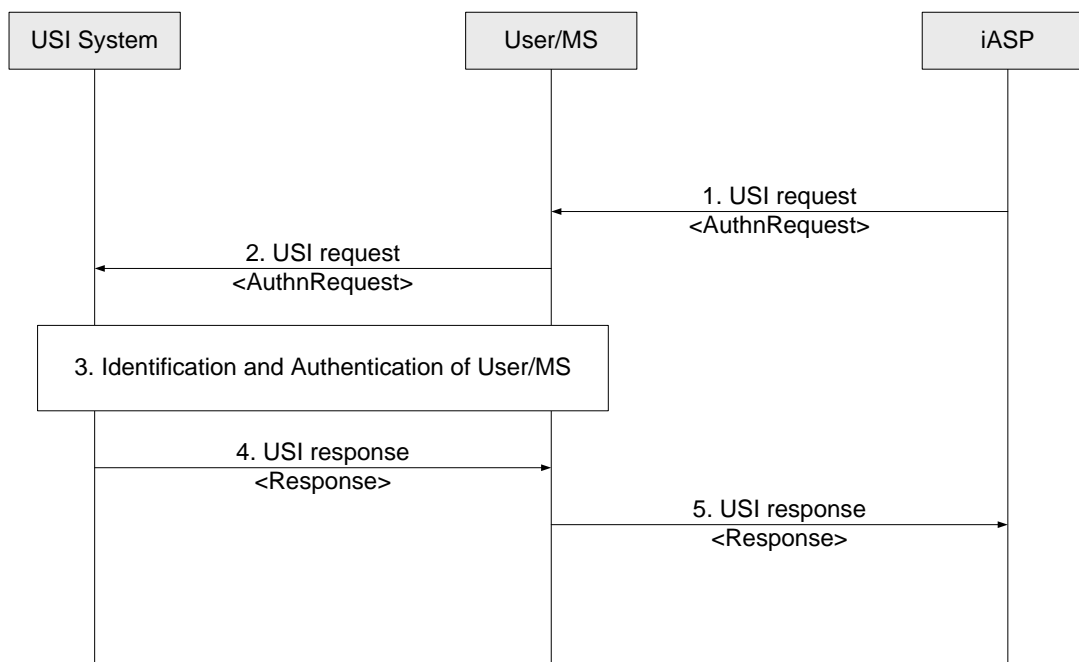
#### 8.1.2 ASP authentication using bearer authentication

The authorization token specifies “bearer” method of authentication, meaning the iASP is authenticated by USI system using HTTP client side authentication. This is applicable only to direct iASP-USI communication,

## USI

**8.2 Identification and Authentication of User/MS**

To prevent the MS from providing a fraud ID (e.g. other person's USI ID) to the iASP, the iASP MUST have a way to validate the USI ID. The authorization is illustrated in the following diagram.



**Figure 46: Authentication of USI User/MS**

1. iASP uses the User/MS redirected flow described in Section 6.1.2. The USI request contains a request from iASP to detect and authenticate the User/MS identity. Since it is up to the USI System/NSP to decide for the desired authentication method, iASP SHALL NOT specify the desired method(s) of authentication. The <RequestedAuthnContext> SAML element in the <AuthnRequest> sent by the iASP in the USI Request SHALL be ignored by the USI System if included. 2. The USI request is submitted by User/MS to the USI System, without iASP involvement 3. The USI System selects and performs an authentication method, based on the NSP policy and/or the user subscription. For a particular authentication method (e.g. ID/PWD based authentication), the USI System may also redirect the USI request to a proper system that handles the User/MS authentication (e.g. web portal supporting ID/PWD authentication).

4. USI System completes the USI request flow. If the identification/authentication succeeded in step 3, the User/MS USI identity and a proof of its verification (from USI System) are placed in the USI response by means of <AuthnStatement> SAML element. If a failure happened in step 3, the USI system returns a <Response> with <StatusCode> SAML element indicating the failure.

5. User/MS redirects the USI response to the iASP, together with the iASP service request (not shown in this diagram).

## 8.2.1 Methods of authentication

**Error! Reference source not found.** lists the possible User/MS authentication methods. The selection of authentication method is based on the NSP policy and/or the user subscription, and in addition, an authentication policy check and enforcement are done by USI System (in conjunction with AAA) for the specific USI service call during step 8 in section 6.1.2.

**Table 2: Values for <AuthnContext> in <AuthnStatement>**

Value of <AuthnContext>	Meaning	Optional/Mandatory
urn:wmf:usi:authncontext:IP	See 8.2.1.1	Mandatory for USI System N/A for MS (supported by default).
urn:wmf:usi:authncontext:certificate	See 8.2.1.2	Optional for USI System Optional for MS Mandatory for specific services as denoted in Section 8.1
urn:wmf:usi:authncontext:IdPassword	See 8.2.1.3	Optional for both USI System and MS

### 8.2.1.1 Identification and authentication by IP address

In this method, the USI System simply looks-up the source IP address of the User/MS (AAA MAY be used in the process). This method MAY be employed for identification-only purpose (no authentication guaranteed) or a weak-security identification and authentication of User/MS, depending on value of <ForceAuth> (see below).

Security considerations: To ensure the security for this method, the iASN gateway of USI System must implement an egress filter, to ensure that source IP address of outbound IP packets matches the expected MAC address. Filters must also be implemented in CSN to ensure the USI request arrives from the iASN, and not directly from Internet. Even so, the level of security of this authentication method MAY not be sufficient for some type of services. In this case, the iASP MAY decide either to provide USI services or to provide non-USI services or to return an error..

This method of authentication is mandatory for USI server, and is supported by all MS. This method is used if the policy of NSP for the USI service requested does not require another, more secure, method of authentication.

When this method is selected for the User/MS authentication, unless iASP explicitly specifies a <ForceAuth> value 'False', the USI System MUST fail the authentication if security measures mentioned above are not implemented in the CSN.

NOTE: This method of authentication does not work if User/MS is using a VPN. If VPN is used, USI System MAY decide to employ other authentication method or the identification and authentication process fails.

### 8.2.1.2 Identification and authentication using MS side USI certificate

This method is optional for USI server, and is optionally supported by the MS.

In step 3 in the Figure 12, USI System forces client side authentication for the TLS session. MS is required to supply a certificate proving its identity.

The provisioning of this certificate is outside the scope of this release of specification.

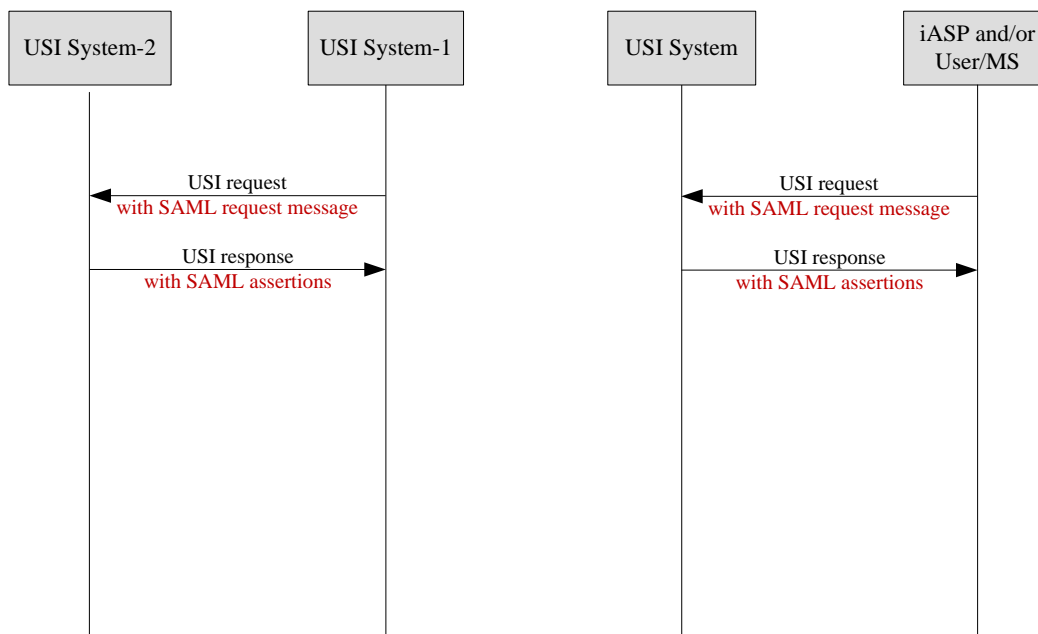
## USI

**8.2.1.3 Identification and authentication using MS/User identity and password**

This method is optional for both USI System and the MS. The MS/User identity and credential for this authentication method may be the long-lived identity and related password. It's up to the NSP policy whether to use the L-ID or to generate another operator specific identity for this authentication method.

**8.3 USI Security Tokens**

USI messages carry authentication and/or authorization information in form of SAML [SAMLCore] messages and assertions (tokens). This is shown in Figure 47.



**Figure 47: Home USI service flow, in the Direct Flow**

SAML messages [SAMLCore] are used to make the SAML-defined requests and return appropriate responses. SAML assertions carry statements about a principal that an asserting party claims to be true. A SAML assertion is typically carried between parties in a SAML protocol response message. The general format of SAML messages and assertions are defined in [SAMLCore].

This section defines a set of constraints on the use of a general SAML requests, assertions and statements for the USI environment. As such, this USI profiling of SAML MAY constrain optionality, require the use of specific SAML elements or functionality (for example, attributes, conditions, or bindings), and in other respects define the processing rules to be followed by actors (e.g. USI System or iASP).

**8.3.1 Use of <AuthnRequest> issued by iASP to USI System**

This SAML message is attached to a USI service request by iASP, in order to request the USI System authenticates the MS or end user and return result to iASP, by means of <AuthnStatement>. iASP MAY supply USI ID, in which case the MS is authenticated against this USI ID, or MAY not supply any USI ID, in which case the current User/MS is authenticated and its USI ID (S-ID or L-ID) is delivered back to the iASP inside the <AuthnStatement>

**Table 3: Use of <AuthnRequest> parameters**

Parameter	SAML element used	Comments
ASP identity	<Issuer>	Mandatory if L-ID requested
User/MS USI ID to authenticate	<Subject>	Optional.
L-ID/S-ID request indication	<NameIDPolicy>	Optional – see Table 4
Signature of iASP	<ds:Signature>	Optional
Whether authentication can be done based on previous authentication, such as the network entry + IP address	ForceAuthn	Optional Indicates whether to force actual authentication with MS

**Table 4: Name Identifier Format Identifiers**

Value	Meaning
urn:oasis:names:tc:SAML:2.0:nameid-format:persistent	The USI identity is L-ID
urn:oasis:names:tc:SAML:2.0:nameid-format:transient	The USI identity is S-ID

### 8.3.2 Use of <AuthnStatement> issued by USI System to iASP

This SAML statement inside a SAML <Assertion> is attached to a USI service response by USI System, usually in response to a request from iASP. The current User/MS is authenticated and the authentication result as well as the authenticated USI ID is delivered inside the <AuthnStatement>.

**Table 5: Use of <AuthnStatement> statement and related <Assertion> parameters**

Parameter	SAML element	Comments
Identity of USI System	<Issuer>	Mandatory
The USI user's ID	<Subject>	Mandatory Also contains indication of L-ID/S-ID. (See Table 4: Name Identifier Format Identifiers)
Authentication method used	<AuthnContext>	Mandatory URI values specify the authentication contexts. (e.g. URI specifying EAP-TTLS authentication) <b>See Error! Reference source not found. in Section 8.2.1: Values for &lt;AuthnContext&gt; in &lt;AuthnStatement&gt;</b>
Signature of USI System	<ds:Signature>	Optional

## USI

**8.3.3 Use of <AuthzDecisionStatement> issued by USI System**

A SAML <AuthzDecisionStatement> statement is placed in a SAML <Assertion> attached to a USI service response by the USI System, to allow a particular USI request to be serviced with another USI System with which the iASP does not have formal trust relationship.

**Table 6: Use of <AuthzDecisionStatement> and related <Assertion> parameters**

Parameter	SAML Element	Comments
Identity of authorizing USI System	<Issuer>	Mandatory
Reference to request	<Resource>	Optional Specified if a specific request is authorized
Identity of target USI System(s)	<Conditions>	Optional Specified if this is authorization for particular USI System(s)
The target USI user(s) or MS(s) ID(s)	<Conditions>	Optional Specified if this is a generic authorization for <u>anyone</u> to submit certain USI requests for these targets
ASP identity/authorization for the USI request	<Subject>	Mandatory
Signature of USI System	<ds:Signature>	Optional

**8.3.4 Use of <AuthzDecisionStatement> Issued by iASP to USI System**

A SAML <AuthzDecisionStatement> statement is placed in a SAML <Assertion> attached to a USI service request issued by iASP, to allow USI System to verify the iASP's identity. The SAML <AuthzDecisionStatement> relates to part or the entire USI request.

**Table 7: Use of <AuthzDecisionStatement> and related <Assertion> parameters**

Parameter	SAML element	Comments
Identity of authorizing iASP	<Issuer>	Mandatory
Reference to request	<Resource>	Mandatory
Signature of iASP	<ds:Signature>	Optional

**8.3.5 Use of <AuthzDecisionQuery> issued by iASP to USI System**

This SAML message is attached to a USI service request by iASP, in order to request authorization by the USI System of the iASP to a service of a foreign USI System.



## 9. Accounting Aspects

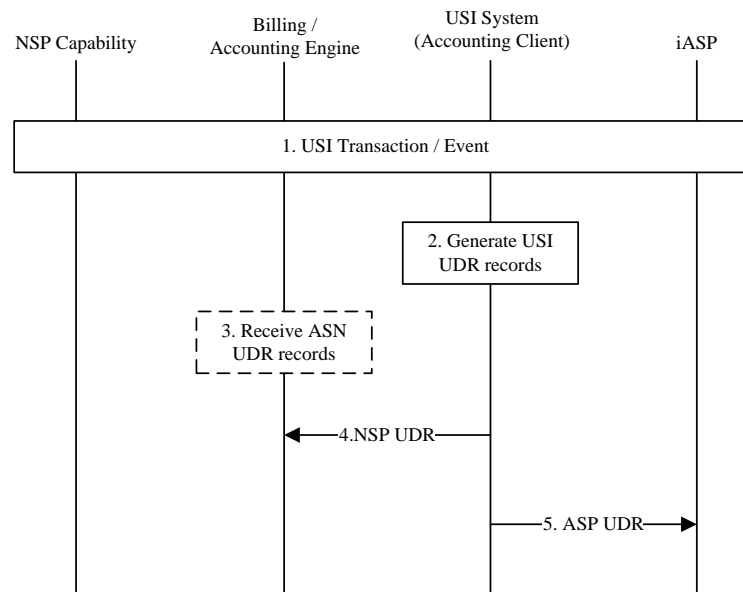
The USI System SHALL support an accounting client function which is responsible for collecting the accounting information about the USI service triggered by iASP. The USI System SHALL generate UDRs based on the collected accounting information and transmits them to the NSP billing system (i.e. AAA/OCS/OFCS) and/or the iASP.

### 9.1 Requirements and Principles

The following requirements and principles are applied to USI accounting:

- The USI accounting supports offline charging.
- The USI accounting supports event based charging that is separate from the traffic bearer accounting.
- The USI accounting supports flow based charging if the USI service results in creation/ modification/ termination of a packet data flow. (e.g. Dynamic QoS support)
- The USI accounting record SHALL be able to provide per iASP or per iASP + USI User granularity.

### 9.2 USI Accounting Procedures



**Figure 48: USI Accounting Procedure**

1. The USI System completes a transition or event. The USI transaction MAY involve other NSP entity.
2. USI Accounting Client records the UDR information, including iASP ID, Service Type, and optionally USI ID of the Service requestor.
3. In case of the flow based charging, the Billing / Accounting Engine (AAA) receives the ASN UDR from the ASN as a result of the USI transaction or event.

## USI

- 1        4. The USI Accounting Client in the USI System reports the NSP UDR to the NSP Billing and  
2        Accounting Engine. The NSP UDR includes the USI UDR information generated in the step 2.
- 3        5. The USI Accounting Client in the USI System MAY report the iASP UDR to the iASP. If the USI  
4        System received the ASN UDR in the step 3, the iASP UDR includes the ASN UDR information  
5        as well as the USI UDR information.

### 6    **9.3 Charging Correlation**

7    The USI System SHALL provide a mechanism for charging correlation between iASP and NSP.

8    On each USI request, the iASP provides an iASP charging ID to the USI System. The Accounting client  
9    generates a USI charging ID based on the iASP charging ID and transmits it to the NSP. Based on the  
10   USI charging ID, NSP generates an access charging ID for the NSP Service that triggered by the USI  
11   request.

12   The USI UDR SHOULD include the iASP charging ID, the access charging ID, and the USI charging ID,  
13   to indicate the AAA/OFCS perform charging correlation.

---

## 10. Interface Specifications

### 10.1 General Description

In order to support different services, different sets of Web Service operations are defined over U1 reference point. For each method defined in the interface, the response to the request is instantaneous and contains the status of the request.

#### 10.1.1 SOAP Transport binding

##### 10.1.1.1 USI Direct service flow

USI Request and Response are transported inside a SOAP envelop over HTTP(S) as specified in [SOAP11].

The SAML SOAP binding [SAMLBind] defines how to send and receive the SAML based tokens.

Simple USI Requests and Responses are simply placed inside the SOAP body.

The POST USI MUST be the recipient's HTTP endpoint for the protocol.

The SOAPAction header MUST be the constant "urn:wmf:usirequest"

```
POST /USIService HTTP/1.1
Host: www.example.com
Content-Type: text/xml
Content-Length: nnn
SOAPAction: urn:wmf:usirequest
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
  <SOAP-ENV:Body>
    <saml:AttributeQuery
      <ds:Signature> ... </ds:Signature>
      <saml:Subject>
        ...
      </saml:Subject>
    </saml:AttributeQuery>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

**Figure 49: Example USI Request**

## USI

```

HTTP/1.1 200 OK
Content-Type: text/xml
Content-Length: nnnn
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
<SOAP-ENV:Body>
  <saml:Assertion>
    <saml:AttributeStatement>
      <ds:Signature> ... </ds:Signature>
      <saml:Subject>
        ...
      </saml:Subject>
    </saml:AttributeStatement>
  </saml:Assertion>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>

```

**Figure 50: Example USI Response**

#### 10.1.1.2 USI Redirect service flow

The USI Redirect service flow uses the same SOAP based USI Request/Response structure as the USI direct service flow, with following change: The SOAP envelope is placed into an HTML form control and transmitted using the HTTP POST method as specified here:

The SOAP is form-encoded by applying the base-64 encoding rules to the XML representation of the message and placing the result in a hidden form control within a form as defined by [HTML401] Section 17. The HTML document MUST adhere to the XHTML specification, [XHTML]. The base64-encoded value MAY be line-wrapped at a reasonable length in accordance with common practice.

The action attribute of the form MUST be the recipient's HTTP endpoint for the protocol or profile using this binding to which the SAML message is to be delivered. The method attribute MUST be "POST".

If the message is a USI Request, then the form control MUST be named USIRequest. If the message is a USI Response, then the form control MUST be named USIResponse. Any additional form controls or presentation MAY be included but MUST NOT be required in order for the recipient to process the message.

If a SAML "RelayState" value [SAMLBind] is required to accompany a SAML protocol message, it MUST be placed in an additional hidden form control named RelayState within the same form with the USI message, as defined by [SAMLBind].

**Figure 51: Example USI Request/Response in USI Redirect service flow**

In response to a USI request in a SOAP message, the USI server MUST return either a USI response element within the body of another SOAP message or generate a SOAP fault.

The USI server MUST NOT send a fault code or other error messages to the sending party. Instead, USI response with status code is used.

The USI server SHOULD return a <SOAP-ENV:Faultstring> element containing an informative message. This specification does not specify any normative text. Sending parties MUST NOT rely on specific contents in the <SOAP-ENV:Faultstring> element.

If, on the other hand, the USI Server cannot, for some reason, process the SOAP message, it should return a SOAP fault code. SOAP fault codes MUST NOT be sent for errors related to the USI service itself (e.g. as a signal that subject is not authorized to access the USI service)

Web service operations for dynamic QoS support in USI are defined as follows:

```
createQoSSession
modifyQoSSession
terminateQoSSession
queryQoSSession
startQoSEventNotification
stopQoSEventNotification
```

## USI

(2) Web service operations provided by iASP

queryApplicationSession

notifyQoSEvent

Each operation or method belongs to one of two different categories based on who provides the services.

## 10.2.1 Parameter Definition

### 10.2.1.1 qoSSessionID

The qoSSessionID is of type xsd:string, and it identifies a QoS session per application per MS in USI. This identifier is generated by the USI System.

### 10.2.1.2 endUserID

The endUserID is of type xsd:string, and it contains a user pseudo-identity that can be used for user identification in USI. Refer to USI identity defined in 7.1.

### 10.2.1.3 endUserIPAddress

The endUserIPAddress is type of xsd:string, and it contains user's IP address which is acquired by the iASP. The UserIPAddress SHALL appear either source or destination address in every IP flow of the USI QoS session.

### 10.2.1.4 applicationChargingID

The applicationChargingID is of type xsd:string, contains the application level charging identifier that is sent by an application of iASP. This information MAY be used for charging correlation.

### 10.2.1.5 flowNumber

The flowNumber is of type xsd:int, and it contains the ordinal number of the IP flow. It is used for flow identification within a USI QoS session.

### 10.2.1.6 mediaType

The mediaType is of type xsd:string, and it determines the media type of a session component. The media types indicate the type of media in the same way as the SDP media types with the same names defined in RFC 4566. The following values are defined:

AUDIO

VIDEO

DATA

APPLICATION

CONTROL

TEXT

MESSAGE

OTHER

### 10.2.1.7 codecData

The codecData is of type xsd:string, and it contains codec related information known at the application of the iASP. The codecData MAY only be applicable to SIP-based VoIP applications as its value consists of SDP lines.

**10.2.1.8 qoSClass**

The qoSClass is of type xsd:int, and it identifies the particular service between the iASP and USI System that the application service session belongs to. The qosClass is uniquely defined between the iASP and NSP per Service Level Agreements (SLA). Each qosClass defines specific QoS information in the USI System for a given iASP. It MAY be used to complete the QoS authorization with application specific default settings in the WiMAX Dynamic QoS Subsystem. Following two tables provide examples of qosClass mapping to QoS information for two arbitrary iASP. As can be seen, the qosClass definition is unique only within the USI System for a given NSP-iASP relationship.

iASP1	Guaranteed Min Rate UL	Guaranteed Min Rate DL	Max UL Rate	Max DL Rate	Packet Interval	Max Packet Size
Class 0	N/A	N/A	N/A	N/A	N/A	N/A
Class 1	50 Kb/s	400 Kb/s	200 Kb/s	1 Mb/s	10 ms	512 Oct
Class 2	100 Kb/s	800 Kb/s	2 Mb/s	2 Mb/s	10 ms	875 Oct
Class 3	700 Kb/s	300 Kb/s	1.5 Mb/s	500 Kb/s	5 ms	763 Oct

iASP2	Guaranteed Min Rate UL	Guaranteed Min Rate DL	Max UL Rate	Max DL Rate	Packet Interval	Max Packet Size
Class 0	N/A	N/A	N/A	N/A	N/A	N/A
Class 1	700 Kb/s	300 Kb/s	1.5 Kb/s	500 Mb/s	15 ms	700 Oct
Class 2	200 Kb/s	600 Kb/s	400 Mb/s	2 Mb/s	10 ms	805 Oct

**10.2.1.9 reservationPriority**

The reservationPriority is of type xsd:int, and is used to assign a priority to the IP flow of the media. Values from 0 to 7 are defined where 0 is the lowest level of priority.

**10.2.1.10 result**

The result is of type xsd:boolean, and it indicates whether the result of the request is successful or not.

**10.2.1.11 faultCode**

The faultCode is of type xsd:string, and it indicates the reason for the failure of the result.

**10.2.1.12 qoSEvent**

The qoSEvent is of type xsd:enumeration and it specifies the events that MAY occur on any active USI QoS session on the user connection(s). The QoSEvent is defined with the following values:

ABNORMAL\_CONNECTION\_TERMINATION (0)

End user connection(s) terminated abnormally because of a fault in the network (e.g., handoff failure) causing all the QoS reservations that were active on the connection(s) to be released as well.

USI

**NORMAL\_CONNECTION\_TERMINATION (1)**

End user connection(s) terminated normally – e.g. user(s) have logged off – causing all QoS reservations active on the connection(s) to be automatically released.

**QoS\_RELEASED (2)**

QoS reservation that was active on an end user connection has been released because the QoS service flow is removed, but user traffic is being served through another service flow (e.g., initial service flow).

**10.2.2 Type Definition****10.2.2.1 QoSFlowInfo Structure**

The QoSFlowInfo contains service information for a single media component within an application session (e.g., a VoIP session). The service information MAY be based on information exchanged between the application and the application session client in the MS. This information MAY be used by the PCC (Dynamic QoS Subsystem) to determine authorized QoS and IP flow classifiers for bearer authorization and PCC rule selection.

Parameter	Type	Occurrence	Description / Clause defined
flowNumber	xsd:int	1	10.2.1.5
flowDescription	FlowDescription	1-2	each for either uplink or downlink flow. In case of bi-directional IP flow, the flowDescription will appear two times. 10.2.2.2
qoSInformation	QoSInformation	1	10.2.2.3
mediaType	xsd:string	0-1	10.2.1.6
codecData	xsd:string	0-1	10.2.1.7
reservationPriority	xsd:int	0-1	10.2.1.9

**10.2.2.2 FlowDescription Structure**

The FlowDescription defines a packet filter for an IP flow with the following information.

Parameter	Type	Occurrence	Description
direction	xsd:string	1	UPLINK or DOWNLINK
sourceIPAddress	xsd:string	1	Source IP address
sourcePort	xsd:int	0-1	if omitted, any source port is allowed
destinationIPAddress	xsd:string	1	Destination IP address
destinationPort	xsd:int	0-1	if omitted, any destination port is allowed
protocol	xsd:string	0-1	UDP or TCP if omitted, any protocol is allowed



### 10.2.2.3 QoSInformation Structure

The QoSInformation contains QoS related information that is being applied to the IP flow. The QoSInformation includes the following parameters.

Parameter	Type	Occurrence	Description / Clause defined
qoSClass	xsd: int	1	10.2.1.8. If following parameters are also present, the qoSClass is ignored
maxBandwidthUL	xsd:int	0-1	Refer to [PCC]
maxBandwidthDL	xsd:int	0-1	Refer to [PCC]
guaranteedBitrateUL	xsd:int	0-1	Refer to [PCC]
guaranteedBitrateDL	xsd:int	0-1	Refer to [PCC]
packetInterval	xsd:int	0-1	Refer to [PCC]
packetSize	xsd:int	0-1	Refer to [PCC]

### 10.2.2.4 FailedFlowInfo Structure

The FailedFlowInfo contains information on the IP flow which has caused the error.

Parameter	Type	Occurrence	Description / Clause defined
flowNumber	xsd:int	1	identifies which flow has encountered an error 10.2.1.5
flowDescription	FlowDescription	0-2	SHOULD be the same as the requested one, otherwise it SHALL be ignored 10.2.2.2
qoSInformation	QoSInformation	1	This information can be used to indicate the reduced QoS reservation. 10.2.2.3
faultCode	xsd:string	0-1	flow-specific error code 10.2.1.11

## 10.2.3 Message Definition for Web Service Operations

### 10.2.3.1 Web Service Operations provided by USI System

#### 10.2.3.1.1 createQoSSession Request

Parameter	Type	Occurrence	Description / Clause defined
endUserID	xsd:anyURI	0-1	At least one of the user identities (i.e., endUserID and UserIPAddress) SHALL appear. 10.2.1.2
endUserIPAddress	xsd:string	0-1	10.2.1.3
applicationChargingID	xsd:string	0-1	10.2.1.4
qoSFlowInfo	QoSFlowInfo	1+	Every IP flow SHALL contain MS's IP

USI

Parameter	Type	Occurrence	Description / Clause defined
			address in its source or destination IP address. 10.2.2.1

#### 10.2.3.1.2 createQoSSession Response

Parameter	Type	Occurrence	Description / Clause defined
qoSSessionID	xsd:string	1	A newly generated identifier for this USI QoS session. 10.2.1.1
result	xsd:boolean	1	10.2.1.10
faultCode	xsd:string	0-1	This error code is not specific to an IP flow. (e.g., UNKNOWN_USER) 10.2.1.11
failedFlowInfo	FailedFlowInfo	0+	10.2.2.4

#### 10.2.3.1.3 modifyQoSSession Request

Parameter	Type	Occurrence	Description / Clause defined
qoSSessionID	xsd:string	1	The identifier generated by the USI server in response to the original QoSSessionCreation operation. 10.2.1.1
qoSFlowInfo	QoSFlowInfo	1+	The IP flows to be modified. The IP flows can be added, removed, or changed. A new flowNumber SHALL be used to add a new QoS IP flow. The IP flows which are not specified but previously provisioned are remained unchanged. 10.2.2.1

#### 10.2.3.1.4 modifyQoSSession Response

Parameter	Type	Occurrence	Description / Clause defined
result	xsd:boolean	1	10.2.1.10
faultCode	xsd:string	0-1	This error code is not specific to an IP flow. 10.2.1.11
failedFlowInfo	FailedFlowInfo	0+	10.2.2.4

**10.2.3.1.5 terminateQoSSession Request**

Parameter	Type	Occurrence	Description / Clause defined
qoSSessionID	xsd:string	1	The identifier generated by the USI server in response to the original QoSSessionCreation operation. 10.2.1.1

**10.2.3.1.6 terminateQoSSession Response**

Parameter	Type	Occurrence	Description / Clause defined
result	xsd:boolean	1	10.2.1.10
faultCode	xsd:string	0-1	10.2.1.11
failedFlowInfo	FailedFlowInfo	0+	10.2.2.4

**10.2.3.1.7 queryQoSSession Request**

Parameter	Type	Occurrence	Description / Clause defined
qoSSessionID	xsd:string	1	The identifier generated by the USI server in response to the original QoSSessionCreation operation. 10.2.1.1

**10.2.3.1.8 queryQoSSession Response**

Parameter	Type	Occurrence	Description / Clause defined
result	xsd:Boolean	1	The value TRUE indicates that the USI QoS session is active at the USI server. 10.2.1.10
faultCode	xsd:string	0-1	10.2.1.11
qoSFlowInfo	QoSFlowInfo	0+	does not occur when the result is false 10.2.2.1

**10.2.3.1.9 startQoSEventNotification Request**

Parameter	Type	Occurrence	Description / Clause defined
endUserID	xsd:anyURI	1+	Identifies one or more end users for which the iASP wants to receive QoS notification events. 10.2.1.2
qoSEvent	xsd:enumeration	1+	One or more QoS events the iASP wants to receive. 10.2.1.12

**10.2.3.1.10 startQoSEventNotification Response**

Parameter	Type	Occurrence	Description / Clause defined
result	xsd:boolean	1	10.2.1.10
correlator	xsd:string	0-1	Identifies the notification registration. To be defined also in the parameter definition section.

**10.2.3.1.11 stopQoSEventNotification Request**

Parameter	Type	Occurrence	Description / Clause defined
correlator	xsd:string	1	Indicates notification registration to be stopped.

**10.2.3.1.12 stopQoSEventNotification Response**

Parameter	Type	Occurrence	Description / Clause defined
result	xsd:boolean	1	10.2.1.10

**10.2.3.2 Web Service Operations provided by iASP****10.2.3.2.1 queryApplicationSession Request**

Parameter	Type	Occurrence	Description / Clause defined
qoSSessionID	xsd:string	1	The USI QoS session the USI server wants to check if it is still maintained at the iASP side. 10.2.1.1

**10.2.3.2.2 queryApplicationSession Response**

Parameter	Type	Occurrence	Description / Clause defined
result	xsd:boolean	1	The value of TRUE indicates that the USI QoS session is active at the iASP side. 10.2.1.10
faultCode	xsd:string	0-1	10.2.1.11
qoSFlowInfo	QoSFlowInfo	0+	does not occur when the result is FALSE 10.2.2.1

**10.2.3.2.3 notifyQoSEvent Request**

Parameter	Type	Occurrence	Description / Clause defined
qoSSessionID	xsd:string	1	The USI QoS session the event is related to. 10.2.1.1

## USI

Parameter	Type	Occurrence	Description / Clause defined
qoSEvent	xsd:enumeration	1	10.2.1.12
flowNumber	xsd:int	0+	If the event is not session-level but applicable to specific flow(s). 10.2.1.5

**10.2.3.2.4 notifyQoSEvent Response**

Parameter	Type	Occurrence	Description / Clause defined
result	xsd:boolean	1	10.2.1.10

**10.3 USI Accounting**

Web service operations for the USI accounting are defined as follows:

(1) Web service operations provided by USI System

(None)

(2) Web service operations provided by iASP

notifyAccounting

**10.3.1 Parameter Definition****10.3.1.1 aspChargingID**

The aspChargingID is of type xsd:string and contains the application level charging identifier that is generated by an iASP. This information MAY be used for charging correlation.

**10.3.1.2 aspID**

The aspID is of type xsd:string and identifies an iASP that requested a USI service.

**10.3.1.3 serviceType**

The serviceType is type of xsd:int and identifies a particular USI service between the iASP and USI System. The serviceType is uniquely defined between the iASP and NSP per Service Level Agreements (SLA). Each serviceType defines a specific USI service in the USI System for a given iASP. Following table shows an example of serviceType mapping to the USI services for an arbitrary iASP.

serviceType	USI Service
1	MS Location Retrieval
2	MS Status Retrieval
3	MS IP Address Discovery
4	Device Capability Retrieval

USI

serviceType	USI Service
5	Dynamic QoS Support
6	E-Payment

#### 10.3.1.4 acctRecordType

The acctRecordType is type of xsd:enumeration and indicates the record type as follows:

EVENT\_RECORD 1

An Accounting Event Record is used to indicate that a one-time event has occurred (meaning that the start and end of the event are simultaneous). This record contains all information relevant to the service, and is the only record of the service.

START\_RECORD 2

An Accounting Start, Interim, and Stop Records are used to indicate that a service of a measurable length has been given. An Accounting Start Record is used to initiate an accounting session, and contains accounting information that is relevant to the initiation of the session.

INTERIM\_RECORD 3

An Accounting Interim Record contains cumulative accounting information for an existing accounting session.

STOP\_RECORD 4

An Accounting Stop Record is sent to terminate an accounting session and contains cumulative accounting information relevant to the existing session.

#### 10.3.1.5 acctRecordNumber

The acctRecordNumber is of type xsd:int and identifies this record within one session. An easy way to generate this parameter is to set the value to 0 for records of type EVENT\_RECORD and START\_RECORD, and set the value to 1 for the first INTERIM\_RECORD, 2 for the second, and so on until the value for STOP\_RECORD is one more than for the last INTERIM\_RECORD. This parameter is only used for the flow based charging.

#### 10.3.1.6 acctSessionID

The acctSessionID is of type xsd:string and is used to match Start, Interim, and Stop. It is generated by the accounting client and is unique per start/stop pair.

#### 10.3.1.7 eventTimeStamp

The eventTimeStamp is type of xsd:int and indicates the time that this accounting event occurred, in seconds since January 1, 1970 00:00 UTC.

#### 10.3.1.8 nasIPAddress

The nasIPAddress is of type xsd:string and indicates the IP address of the serving NAS.

#### 10.3.1.9 interimCause

The interimCause is of type xsd:enumeration and is used to indicate the reason why the accounting interim message was generated by the accounting client. The following values are defined:

INTERIM\_INTERVAL 1

Interim message was generated by the accounting interim interval timer.

IDLE\_MODE\_TRANSITION 2

Interim message was generated upon the idle mode transition.

#### **10.3.1.10 terminationCause**

The terminationCause is of type xsd:enumeration and it indicates why the flow has been stopped.

#### **10.3.1.11 uplinkOctets**

The uplinkOctets is type of xsd:int and contains the total number of octets in IP packets sent by the user. Counted after de-compression and de-fragmentation at the accounting agent.

#### **10.3.1.12 uplinkPackets**

The uplinkPackets is type of xsd:int and contains the total number of IP packets sent by the user. Counted after de-compression and de-fragmentation at the accounting agent.

#### **10.3.1.13 downlinkOctets**

The downlinkOctets is type of xsd:int and contains the total number of octets in IP packets sent to the user, as received at the accounting agent from the IP network (i.e. prior to any compression and/or fragmentation).

#### **10.3.1.14 downlinkPackets**

The downlinkPackets is type of xsd:int and contains the total number of IP packets sent to the user, as received at the accounting agent from the IP network (i.e. prior to any compression and/or fragmentation).

#### **10.3.1.15 acctSessionTime**

The acctSessionTime is type of xsd:int and contains the number of seconds the flow was active.

#### **10.3.1.16 activeTime**

The activeTime is type of xsd:int and contains the number of seconds in which the MS is active as opposed to idle mode.

#### **10.3.1.17 msMACAddress**

The msMACAddress is of type xsd:string and contains the MAC Address of the MS. The format of this parameter is like the following: 1A-2B-3C-4D-5E-6F.

#### **10.3.1.18 msIPAddress**

The msIPAddress is of type xsd:string and contains the IP Address of the MS.

#### **10.3.1.19 sessionContinue**

The sessionContinue is of type xsd:boolean and if TRUE, it indicates that the stop is immediately followed by a start. If this parameter is FALSE or missing it means that this is the final stop.

#### **10.3.1.20 beginningOfSession**

The beginningOfSession is of type xsd:boolean and if TRUE, it indicates that a new flow is starting. If this parameter is FALSE or missing, this is a continuation of a previous flow.

#### **10.3.1.21 idleModeTransition**

The idleModeTransition is type of xsd:enumeration and indicates idle mode entry(1) or exit(0).

#### **10.3.1.22 accessChargingID**

The accessChargingID is of type xsd:string and identifies a packet data flow. This parameter is assigned to be unique for each flow in an IP session and remains constant through all handover scenarios.

USI

**10.3.1.23 napID**

The napID is type of xsd:string and contains the operator ID of the NAP.

**10.3.1.24 nsplD**

The nsplD is type of xsd:string and contains the operator ID of the NSP.

**10.3.1.25 bsID**

The bsID is type of xsd:string and contains the value of BSID, which uniquely identifies a base station that is serving the MS at the time the UDR is generated.

**10.3.1.26 direction**

The direction is type of xsd:enumeration and indicates the direction of the packet data flow. The following values are defined:

UPLINK	1
DOWNLINK	2
BIDIRECTIONAL	3

**10.3.1.27 result**

The result is of type xsd:boolean, and it indicates whether the result of the request is successful or not.

**10.3.1.28 faultCode**

The faultCode is of type xsd:string, and it indicates the reason for the failure of the result.

**10.3.2 Type Definition**

*<This section defines complex Type (structure type)>*

**10.3.2.1 USIID Structure**

The USIID contains at least one of the L-ID and the S-ID.

Parameter	Type	Occurrence	Description / Clause defined
longLivedID	xsd:string	0-1	7.1.1
shortLivedID	xsd:string	0-1	7.1.2

**10.3.2.2 GrantedQoS Structure**

The GrantedQoS describes over the air QoS parameters that are associated with a flow.

Parameter	Type	Occurrence	Description
qosID	xsd:int	1	A unique ID for this QoS specification in this packet. The ID is used in the Service Flow Descriptor attribute to reference a specific QoS Spec.
globalServiceClassName	xsd:string	0-1	This parameter represents the Global Service Class Name as defined in IEEE802.16e.



Parameter	Type	Occurrence	Description
serviceClassName	xsd:string	0-1	This parameter represents the Service Class Name as defined in IEEE802.16e.
scheduleType	xsd:enumeration	1	The parameter specifies the Uplink Granted Scheduling Type as defined in IEEE802.16e. Octet enumeration with the following values defined: 0 = Reserved 1 = Reserved 2 = Best Effort 3 = nrtPS 4 = rtPS 5 = Extended rtPS 6 = UGS 7 – 255 = Reserved
trafficPriority	xsd:int	0-1	The value of this parameter specifies the priority assigned to a service flow.
maximumSustainedTrafficRate	xsd:int	0-1	This parameter defines the peak information rate of the service. The rate is expressed in bits per second and pertains to the SDUs at the input to the system.
minimumReservedTrafficRate	xsd:int	0-1	Represents the Minimum Reserved Traffic Rate as defined in IEEE802.16e. This parameter specifies the minimum rate reserved for this service flow.
maximumTrafficBurst	xsd:int	0-1	Represents the Maximum Traffic Burst as defined in IEEE802.16e.
toleratedJitter	xsd:int	0-1	Represents the Tolerated Jitter as defined in IEEE802.16e.
maximumLatency	xsd:int	0-1	Represents the Maximum Latency as defined in IEEE802.16e.
reducedResourcesCode	xsd:enumeration	0-1	This code indicates that the requesting entity will accept reduced resources if the requested resources are not available.
mediaFlowType	xsd:enumeration	0-1	Describes the application type, used as a hint in admission decisions, for instance, VoIP, video, PTT, gaming, etc. The first octet of the string represents an enumeration with the following values: 0 = Reserved 1 = Voice over IP 2 = Robust Browser 3 = Secure Browser/ VPN 4 = Streaming video on demand 5 = Streaming live TV

USI

Parameter	Type	Occurrence	Description
			6 = Music and Photo Download 7 = Multi-player gaming 8 = Location-based services 9 = Text and Audio Books with Graphics 10 = Video Conversation 11 = Message 12 = Control 13 = Data 14 – 255 = Reserved
unsolicitedPollingInterval	xsd:int	0-1	The value of this parameter specifies the maximal nominal interval between successive polling grants opportunities for this Service Flow.
mediaFlowDescription	xsd:int	0-1	This is a variable length string having SDP information . The <SDP string> is encoded as specified in IETF RFC 2327.
transmissionPolicy	xsd:int	0-1	The parameter indicates the transmission policy of a service flow.

### 10.3.2.3 Flows Structure

The Flows indicates IP flows via their flow identifiers.

Parameter	Type	Occurrence	Description / Clause defined
mediaComponentNumber	xsd:int	1	indicates the media component number.
flowNumber	xsd:int	0+	If no flowNumber exists, the Flows structure refers to all flows matching the media component number.

### 10.3.2.4 AFCorrelationInformation Structure

The AFCorrelationInformation contains aspChargingID and associated flow identifiers.

Parameter	Type	Occurrence	Description / Clause defined
aspChargingID	xsd:string	1	identifies which flow has encountered an error 10.3.1.1
Flows	Flows	1+	SHOULD be the same as the requested one, otherwise it SHALL be ignored 10.3.2.3

### 10.3.3 Message Definition for Web Service Operations

#### 10.3.3.1 Web Service Operations provided by iASP

##### 10.3.3.1.1 notifyAccounting Request

Parameter	Type	Occurrence		Description / Clause defined
		Event Based	Flow Based	
aspChargingID	xsd:string	1	1	10.3.1.1
aspID	xsd:string	1	1	10.3.1.2
usiID	USIID	1	1	10.3.2.1
msIPAddress	xsd:string	0-1	1	10.3.1.18
serviceType	xsd:int	1	1	10.3.1.3
eventTimeStamp	xsd:int	1	1	10.3.1.7
acctRecordType	xsd:enumeration	1	1	10.3.1.4
acctRecordNumber	xsd:int	0	1	10.3.1.5
acctSessionID	xsd:string	0	1	10.3.1.6
nasIPAddress	xsd:string	0	0-1	10.3.1.8
interimCause	xsd:enumeration	0	0-1	10.3.1.9
terminationCause	xsd:enumeration	0	0-1	10.3.1.10
uplinkOctets	xsd:int	0	0-1	10.3.1.11
uplinkPackets	xsd:int	0	0-1	10.3.1.12
downlinkOctets	xsd:int	0	0-1	10.3.1.13
downlinkPackets	xsd:int	0	0-1	10.3.1.14
acctSessionTime	xsd:int	0	0-1	10.3.1.15
activeTime	xsd:int	0	0-1	10.3.1.16
msMACAddress	xsd:string	0	0-1	10.3.1.17
sessionContinue	xsd:boolean	0	0-1	10.3.1.19
beginningOfSession	xsd:boolean	0	0-1	10.3.1.20
idleModeTransition	xsd:enumeration	0	0-1	10.3.1.21
accessChargingID	xsd:string	0	0-1	10.3.1.22
napID	xsd:string	0	0-1	10.3.1.23
nsplID	xsd:string	0	0-1	10.3.1.24
bsID	xsd:string	0	0-1	10.3.1.25
Direction	xsd:enumeration	0	0-1	10.3.1.26
uplinkFlowDescription	FlowDescription	0	0-1	10.2.2.2 Specifies a packet filter for an

USI

Parameter	Type	Occurrence		Description / Clause defined
		Event Based	Flow Based	
				uplink service flow.
downlinkFlowDescription	FlowDescription	0	0-1	10.2.2.2 Specifies a packet filter for a downlink service flow.
uplinkGrantedQoS	GrantedQoS	0	0-1	10.3.2.2 Specifies uplink QoS granted to the MS.
downlinkGrantedQoS	GrantedQoS	0	0-1	10.3.2.2 Specifies downlink QoS granted to the MS.
QoSInformation	QoSInformation	0	0-1	10.2.2.3
AFCorrelationInformation	AFCorrelationInformation	0	0-1	10.3.2.4

1

2 **10.3.3.1.2 notifyAccounting Response**

Parameter	Type	Occurrence		Description / Clause defined
		Event Based	Flow Based	
aspChargingID	xsd:string	1	1	10.3.1.1
acctRecordType	xsd:enumeration	1	1	10.3.1.4
acctRecordNumber	xsd:int	0	1	10.3.1.5
acctSessionID	xsd:string	0	1	10.3.1.6
accessChargingID	xsd:string	0	0-1	10.3.1.22
result	xsd:Boolean	1	1	10.3.1.27 Indicates if the notifyAccounting operation is successful or not.
faultCode	xsd:int	0-1	0-1	10.3.1.28 Present if the result parameter is FALSE.

3

4 **10.4 E-Payment service interface**5 **10.4.1 E-Payment Authorization Request**

6 E-Payment Authorization Request is a SAML <AuthzDecisionQuery> message. The parameters are  
7 shown in Table 8.

**Table 8: <AuthzDecisionQuery> parameters in E-Payment Authorization Request**

Parameter	Type	Occurrence	Description / Clause defined
<Subject>	saml:SubjectType	0-1	The identity of the user (to be verified).
<Action>	saml:ActionType	1	The type of this USI request. See 10.4.1.1
<Evidence>	usi:TransactionDetailsType	1+	The e-payment transaction details. See 10.4.5

**10.4.1.1 <Subject>**

This optional parameter holds the USI identity (L-ID or S-ID) of the User/MS. If specified, the NSP must fail the e-payment authorization if the User/MS does not verify to have this USI identity.

**10.4.1.2 <Action>**

This parameter defines that the SAML authorization query is a E-Payment Authorization Request. The value of this parameter must be **urn:wmf:usi:request:paymentauthrequest**

**10.4.2 E-Payment Authorization Response**

E-Payment Authorization Response uses a SAML **<Assertion>** containing a SAML **<AuthzDecisionStatement>**. The values of both structures are described in Table 9 and Table 10, respectively.

**Table 9: <Assertion> parameters used in E-Payment Authorization Response**

Parameter	Type	Occurrence	Description / Clause defined
<Subject>	xsd:int	1	The verified identity of the User/MS
<Issuer>	saml:IssuerType	1	The identity of the USI System
<Condition>		1	
<ds:Signature>	<ds:Signature>	1	USI integrity

**Table 10: <AuthzDecisionStatement> parameters used in E-Payment Authorization Response**

Parameter	Type	Occurrence	Description / Clause defined
<Decision>	saml:DecisionType	1	as specified in [SAMLCore]: 'Permit' if authorization is granted, otherwise 'Deny'
<Evidence>	usi:TransactionDetailsType	1+	The e-payment transaction details. See 10.4.5
<StatusDetail>	xsd:string	0-1	Text describing in human language more about the decision
<Resource>	xsd:string	1	USI System transaction identifier. See

## USI

Parameter	Type	Occurrence	Description / Clause defined
			10.4.2.1

**10.4.2.1 <Resource>**

This is a unique random URI value identifying the particular e-payment transaction within the USI System.  
This parameter is internal to the USI System and opaque to iASP and MS.

**10.4.3 E-Payment Charge Request**

E-Payment Charge Request is a SAML **<AuthzDecisionQuery>** message as defined in [SAMLCore].

**Table 11: <AuthzDecisionQuery> parameters in E-Payment Charge Request**

Parameter	Type	Occurrence	Description / Clause defined
<Subject>	saml:SubjectType	0-1	The identity of the user (to be verified).
<Action>	saml:ActionType	1	The type of this USI request. See
<Evidence>	usi:TransactionDetailsType	1+	The e-payment transaction details. See 10.4.5

**10.4.3.1 <Evidence>**

The <Evidence> element is filled as shown in Table 12.

**Table 12: <Evidence> parameters in E-Payment Charge Request parameters**

Parameter	Type	Occurrence	Description / Clause defined
<order-total>	usi:MoneyType	1	E-Payment Charge Confirm
<amount>, attribute "currency"	xsd:string	1	Currency as defined by ISO-4217 3-letter codes
< order-number>	xsd:string	1	USI System transaction identifier. See 10.4.2.1

-

**10.4.3.2 <Action>**

This parameter defines that the SAML authorization query is a E-Payment Authorization Request. The value of this parameter must be **urn:wmf:usi:request:paymentchargerequest**

**10.4.4 E-Payment Charge Response**

E-Payment Charge Response uses a SAML **<Assertion>** containing a SAML **<AuthzDecisionStatement>**. The values of both structures are described in.

**10.4.4.1 <Evidence>**

The <Evidence> element is filled as shown in Table 13.

USI

**Table 13: <Evidence> parameters in E-Payment Charge Response**

Parameter	Type	Occurrence	Description / Clause defined
<order-total>	usi:MoneyType	1	How much to charge
<order-number>	xsd:string	1	USI System transaction identifier. See 10.4.2.1

**10.4.5 E-Payment Structures****10.4.5.1 E-Payment Transaction Details**

E-Payment Transaction Details information uses XML structure with parameters as shown as shown in Table 14.

**Table 14: E-Payment <TransactionDetailsType> parameters**

Parameter	Type	Occurrence	Description / Clause defined
<item>	usi:OrderItemType	1+	Order Item details; see 10.4.5.2
<order-total>	usi:MoneyType	1	Total amount of transaction
< buyer-id>	xsd:long	0-1	Identity of the iASP user (not USI user)

**10.4.5.2 E-Payment Order Item****Table 15: E-Payment <OrderItemType> parameters**

Parameter	Type	Occurrence	Description / Clause defined
<item-name>	xsd:string	1	Short name
<item-description>	xsd:string	0-1	Detailed description
<unit-price>	usi:MoneyType	1	How much does one item cost. See 10.4.5.3
<quantity>	xsd:double	1	How many items
< merchant-item-id>	xsd:string	0-1	Merchant specific P/N (part number)

**10.4.5.3 E-Payment Money**

<Money> is a <xsd:decimal> with attributes as shown below

**Table 16: E-Payment <MoneyType> attributes**

Attribute	Type	Occurrence	Description / Clause defined
attribute "currency"	xsd:string	0-1	Currency as defined by ISO-4217 3-letter codes Default is USD

## 10.5 User/MS Attributes Query service

This service allows for query and response as to the User/MS or Subscriber attributes. It is a unified mandatory interface which supports basic services such as Determining MS Location, MS Status, IP Address Discovery and Dynamic Capability Retrieval. Several attributes can be queried and retrieved in one service call. It is possible for USI System to provide the attributes in either plain text or encrypted form.

### 10.5.1 User/MS Attribute Request

A User/MS Attribute Request is a SAML **<AttributeQuery>** message as defined in [SAMLCore]. The specific parameters used are shown in Table 17.

**Table 17: <AttributeQuery> parameters in a User/MS Attribute Request**

Parameter (SAML element)	Type	Occurrence	Description / Clause defined
<Issuer>	saml:IssuerType	1	Identity of NSP
<Subject>	saml:SubjectType	1	The USI user's ID
<Attribute>	saml:AttributeType	1+	Requested information. See 10.5.1.1

#### 10.5.1.1 <Attribute>

The <Attribute> element in a User/MS Attribute Request contains the request attribute, as defined in 10.5.2.4 in the "name" XML attribute of <Attribute> element.

### 10.5.2 User/MS Attribute Response

A Subscriber Attribute Response is a SAML **<Assertion>** as defined by [SAMLCore] containing a SAML <AttributeStatement>. The parameters used in the <Assertion> are defined in Table 18.

**Table 18: <Assertion> parameters in User/MS Attribute Response**

Parameter (SAML element)	Type	Occurrence	Description / Clause defined
<Issuer>	saml:IssuerType	1	Identity of NSP
<Subject>	saml:SubjectType	1	The USI user's ID
<AttributeStatement>	saml:AttributeStatementType	1	a SAML <b>&lt;AttributeStatement&gt;</b> as shown in Table 19
<ds:Signature>	<ds:Signature>	1	Signature of USI System

#### 10.5.2.1 <AttributeStatement>

The <AttributeStatement> element holds one or more <Attribute> elements



**Table 19: <AttributeStatement> parameters in User/MS Attribute Response**

Parameter (SAML element)	Type	Occurrence	Description / Clause defined
<Attribute>	saml:AttributeType	0+	10.5.2.3
<EncryptedAttribute>	saml:AttributeType	0+	Attribute Encrypted

**10.5.2.2 <EncryptedAttribute>**

The <EncryptedAttribute> element represents a <Attribute> element as defined in 10.5.2.3 in encrypted fashion, as defined by [SAMLCore].

**10.5.2.3 <Attribute>**

An <Attribute> element in a User/MS Attribute Response contains the attribute's type in the "name" and value as defined in 10.5.2.4.

**10.5.2.4 <AttributeValue>**

<AttributeValue> element holds the requested information.

**Table 20: <Attribute> element's type and corresponding <AttributeValue> element value**

XML attribute "Name" of <Attribute>	Type of data contained in <AttributeValue> in the USI response	Occ	Description / Clause defined
urn:wmf:usi:mobilestation:geolocation	10.5.2.5	1	10.5.2.5
urn:wmf:usi:mobilestation:speed	xsd:float	1	Reported in meters per second
urn:wmf:usi:mobilestation:direction	xsd:float	1	The direction towards which User/MS is moving, in degrees (0-360), (0=360=True North).
urn:wmf:usi:mobilestation:heading	xsd:float	1	Which direction the User/MS is facing, in degrees (0-360), (0=360=True North)
urn:wmf:usi:mobilestation:altitude	xsd:float	1	The <alt> element is used to indicate the altitude of a User/MS in meters
urn:wmf:usi:mobilestation:civic-location	10.5.2.6	1	contains the position of a Principal in the format of a street address
urn:wmf:usi:mobilestation:location-measurements	10.5.2.7	1	10.5.2.7

urn:wmf:usi:mobilestation:terminalcapabilities urn:wmf:usi:mobilestation:terminalcapabilities:*	7.4	1+	10.5.2.8
urn:wmf:usi:mobilestation:msstatus urn:wmf:usi:mobilestation:msstatus:*	7.2	1	10.5.2.9
urn:wmf:usi:mobilestation:ipaddress	xsd:string	1	The IP address in standard xxx.xxx.xxx.xxx decimal format
urn:wmf:usi:user:personalinfo	10.5.2.10	1	10.5.2.10
urn:wmf:usi:user:address	10.5.2.11	1	10.5.2.11
urn:wmf:usi:user:medicalinfo	10.5.2.12	1	10.5.2.12
urn:wmf:usi:user:buddylist	xsd:string	1	Contains the comma separated list of LIDs of the user's buddies.

#### 10.5.2.5 Geo-Location Attribute

Geodetic location object as defined in [LBS].

#### 10.5.2.6 Civic-Location Attribute

Civic location object as defined in [LBS].

#### 10.5.2.7 WiMAX Location Measurements

WiMAX Scan object and WiMAX BSID as defined in [LBS]

#### 10.5.2.8 Terminal Capabilities Attribute

The terminal capabilities query is used to retrieve information about the terminal itself and its capabilities. The information is returned in one or more <Attribute> elements.

**Table 21: <Attribute> element's type and corresponding <AttributeValue> element value**

XML attribute "Name" of <Attribute>	Type of data contained in <AttributeValue> in the USI response	Occ	Description / Clause defined
urn:wmf:usi:mobilestation:terminalcapabilities	Multiple <Attribute> elements are returned as specified in the rows below	Multiple as described in this table	This is a query for all terminal capabilities.
urn:wmf:usi:mobilestation:terminalcapabilities:ipcaps	XML object	1	XML object with the information as specified in NWG OTA spec for OMA DM node

## USI

			WiMAX/DevCap/IPCap
urn:wmf:usi:mobilestation: terminalcapabilities:terminalequipment	XML object	1	XML object with the information as specified in NWG OTA spec for OMA DM node WiMAX/TerminalEquipment
urn:wmf:usi:mobilestation: terminalcapabilities:wimaxdevice	XML object	1	XML object with the information as specified in NWG OTA spec for OMA DM node /DevInfo
urn:wmf:usi:mobilestation: terminalcapabilities:currentbandwidth	xsd:int	1	Current bandwidth, in Mbps 0 if below 0.5Mbps
urn:wmf:usi:mobilestation: terminalcapabilities:maxbandwidth	xsd:int	1	Current bandwidth, in Mbps 0 if below 0.5Mbps
urn:wmf:usi:mobilestation:UAProfURL	xsd:string	1	URL link to a User Agent Profile as defined by OTA spec node WiMAX/DevCap/UAProf

1

2 **10.5.2.9 MS Status Attribute**3 **Table 22: <Attribute> element's type and corresponding <AttributeValue> element value**

XML attribute "Name" of <Attribute>	Type of data contained in <AttributeValue> in the USI response	Occ	Description / Clause defined
urn:wmf:usi:mobilestation:msstatus	Multiple <Attribute> elements are returned as specified in the rows below	Multiple as described in this table	This is a query for all terminal capabilities.
urn:wmf:usi:mobilestation:msstatus:modem	xsd:string (containing a URN)	1	Table 23
urn:wmf:usi:mobilestation:msstatus:visitedNSP realm	xsd:string	0-1	Realm of visited NSP NOTE: Is an empty string when not roaming
urn:wmf:usi:mobilestation:msstatus:homeNSP realm	xsd:string	1	Realm of home NSP

urn:wmf:usi:mobilestation:msstatus:servingNA Prealm	xsd:string	1	Realm of serving NAP
--	------------	---	-------------------------

**Table 23: <Attribute> element's type and corresponding <AttributeValue> element value**

WiMAX modem/device status	Value in the <AttributeValue> field
Online	urn:wmf:usi:mobilestation:msstatus:modem:online
Hotlined due to some reason	urn:wmf:usi:mobilestation:msstatus:modem:blocked
Roaming (not hotlined)	urn:wmf:usi:mobilestation:msstatus:modem:roaming
Not on a WiMAX network	urn:wmf:usi:mobilestation:msstatus:modem:offline
Device that is not subscriber and is not currently on the network (hotlined)	urn:wmf:usi:mobilestation:msstatus:modem:unknown

#### 10.5.2.10 Personal Info Attribute

The personal information query is used to retrieve personal information about the user. The information is returned in one or more <Attribute> elements.

**Table 24: <Attribute> element's type and corresponding <AttributeValue> element value**

XML attribute "Name" of <Attribute>	Type of data contained in <AttributeValue> in the USI response	Occ	Description / Clause defined
urn:wmf:usi:user:personalinfo	Multiple <Attribute> elements are returned as specified in the rows below	Multiple as described in this table	This is a query for all personal info.
urn:wmf:usi:user:personalinfo:name	xsd:string	1	Name of the user.
urn:wmf:usi:user:personalinfo:dob	xsd:string	1	Date of birth of the user.
urn:wmf:usi:user:personalinfo:sex	xsd:string	1	Sex of the user.

#### 10.5.2.11 User Address Attribute

The user's address information query is used to retrieve address information of the user. The information is returned in one or more <Attribute> elements.

## USI

**Table 25: <Attribute> element's type and corresponding <AttributeValue> element value**

XML attribute "Name" of <Attribute>	Type of data contained in <AttributeValue> in the USI response	Occ	Description / Clause defined
urn:wmf:usi:user:address	Multiple <Attribute> elements are returned as specified in the rows below	Multiple as described in this table	This is a query for all addresses of user.
urn:wmf:usi:user:address:billing	ContactInfo	1	Billing address of the user. 10.5.2.13
urn:wmf:usi: user:address:work	ContactInfo	0-1	Work address of the user. 10.5.2.13
urn:wmf:usi: user:address:home	ContactInfo	0-1	Home address of the user. 10.5.2.13

**10.5.2.12 Medical Info. Attribute**

The medical information query is used to retrieve medical information of the user. The information is returned in one or more <Attribute> elements.

**Table 26: <Attribute> element's type and corresponding <AttributeValue> element value**

XML attribute "Name" of <Attribute>	Type of data contained in <AttributeValue> in the USI response	Occ	Description / Clause defined
Urn:wmf:usi:user:medicalinfo	Multiple <Attribute> elements are returned as specified in the rows below	Multiple as described in this table	This is a query for all medical information.
Urn:wmf:usi:user:medicalinfo:emergencycontact	ContactInfo	1-2	Emergency information of the user. 10.5.2.13
Urn:wmf:usi:user:medicalinfo:doctorinfo	ContactInfo	1	Doctor information of the user. 10.5.2.13

## USI

Urn:wmf:usi:user:medicalinfo:insuranceinfo	ContactInfo	1 +	Insurance information of the user. 10.5.2.13
Urn:wmf:usi:user:medicalinfo:pharmacyinfo	ContactInfo	0-1	Pharmacy information of the user. 10.5.2.13
Urn:wmf:usi:user:medicalinfo:medicalcondition	xsd:string	1	Medical condition of the user.

### 10.5.2.13 ContactInfo Structure

**Table 27: ContactInfo Structure**

Parameter	Type	Occurrence	Description / Clause defined
name	xsd:string	1	Name of the contact.
address	xsd:string	1	Address of the contact.
phone	xsd:string	0-1	Phone number of the contact.
email	xsd:string	0-1	E-mail address of the contact.

## 10.6 Content Push service

This service allows delivery of a single message or document to the MS. The message or document is delivered to a Content Push agent function in the MS

### 10.6.1 <ContentPushRequest> Message

The specific parameters used are shown in Table 28.

**Table 28: <ContentPushRequest> message elements**

XML Element	Type	Occurrence	Description / Clause defined
<Target>	USIID	1	The identity of the (target) User/MS to whom the content is pushed.
<Requestor>	xsd:string	1	The identity of the iASP
<PortType>	xsd:string	1	'Application' if the content is to be delivered to the MS's OS based applications  'Device' if the content is to be delivered to the WiMAX modem internally
<Port>	xsd:int	1	Identity of OS based application, or identity of internal WiMAX application/function identifier within WiMAX modem

## USI

<Content>	B64 encoded content	0-1	The information to push.
<ContentURL>	xsd:string	0-1	A URL (location pointer) to the information.  If <ContentURL> is used, <Content> should not be present. The URL string is sent to the MS, and it is up to the MS to download the actual information.
<ContentType>	xsd:string	1	MIME type associated with <Content> or <ContentURL>

1

2

3

## 10.7 Live Streaming service

Following data structures are delivered on U1-control for live streaming

### 10.7.1 <LiveStreamingServiceGuide> structure

IE	Type	Occurrence	M/O	Notes
<ProgramInformation>		N times (N = number of channels in the service guide)	M	Include one or more Program ID.
<ProgramID>	string	1	M	
<ContentInfo>			M	Contains MBS contents information in the nested IEs.
<ContentIP>	String	1	O	If the content is public or free, this can be included.
<ContentType>	String	1	M	Indicate the service type.
<DeliveryStartTime>	Time (min, hr, day, month, year)	1	M	Indicate MBS burst transfer start time at base station.
<DeliveryEndTime>	Time (min, hr, day, month, year)	1	M	Indicate MBS burst transfer end time at base station.

### 10.7.2 <LiveStreamingSecurityContext> structure

IE	Type	Occurrence	M/O	Notes
Program Information		N times (N = number of channels in the service guide)	M	Include one or more Program ID.
<ProgramID>	String	1	M	
<SecurityEnabledFlag>	Bool	1	M	Yes or no
<EncryptionMechanism>	String	0-1	O	Only if security is enabled for the program ID
<SupportedKeyExchangeProtocol>	String	0-1	O	Only if security is enabled for the program ID



### 10.7.3 <LiveStreamingQoSContext> structure

IE	Type	Occurrence	M/O	Notes
<ProgramInformation>		N times (N = number of channels in the service guide)	M	Include one or more Program ID.
<ProgramID>	String	1	M	
<PeakRate>	Int	0-1	O	
<SustainedRate>	Int	0-1	O	
<TolerablePlaybackLatency>	Int	0-1	O	

## 10.8 SOAP Headers

As mentioned in Section 8 (Security Aspects), all request and response messages defined in Section 10.1 through Section 10.6 can **optionally** contain SOAP Headers that will include the various USI Security Tokens. These UI Security Tokens will be encapsulated in the data structures as defined in Section 10.8.1

### 10.8.1 Type Definition

#### 10.8.1.1 USIRequestSecurityHeader Structure

The USIRequestSecurityHeader contains at least one of the AuthnRequest or Assertion structure as defined in Section 8 (Security Aspects) for a request message.

Element	Type	Occurrence	Description / Clause defined
AuthnRequest	samlp:AuthnRequest	0-1	Section 8
Assertion	Saml:Assertion	0-1	Section 8

This structure may be contained in the SOAP Header for a request message.

#### 10.8.1.2 USIResponseSecurityHeader Structure

The USIResponseSecurityHeader contains Assertion structure as defined in Section 8 (Security Aspects) for a response message.

Element	Type	Occurrence	Description / Clause defined
Assertion	saml:Assertion	0-1	Section 8

USI

- 1 This structure may be contained in the SOAP Header for a response message.
- 2
- 3

## 10.9 Serve multiple messages in a single WebService Operation

The various web service operations defined in Section 10.1 through Section 10.6 can be classified as being provided by an Operator (USI System) or by an iASP:

### (a) Web service operations provided by USI System

createQoSSession

modifyQoSSession

terminateQoSSession

queryQoSSession

startQoSEventNotification

stopQoSEventNotification

ePaymentAuthz

ePaymentCharge

AttributeQuery

ContentPush

### (b) Web service operations provided by iASP

queryApplicationSession

notifyQoSEvent

notifyAccounting

These individual operations can also be clubbed together in a single web operation provided by the respective USI System or iASP. Multiple requests can be submitted to receive multiple responses, synchronously.

However, if the optional SOAP Header contains USI Security Token, all the requests are processed in context of the given USI Security Token.

## 10.9.1 Parameter Definition

### 10.9.1.1 messageId

The messageId is of type xsd:long, and it uniquely identifies a set of multiple-requests for a caller of the web service operation.

### 10.9.1.2 ipAddress

The ipAddress is of type xsd:string, and it is the IP Address of the caller of the web service operation.

### 10.9.1.3 macAddress

The macAddress is of type xsd:string, and it is the MAC Address of the caller of the web service operation.

USI

**10.9.1.4 commandId**

The commandId is of type xsd:long, and it uniquely identifies a request within a set of multiple-requests for a caller of the web service operation.

**10.9.1.5 timeStamp**

The timeStamp is of type xsd:dateTime, and it contains the timestamp of when the request or response was created.

**10.9.2 Type Definition**

<This section defines complexType (structure type)>

**10.9.2.1 wimaxUSIOperatorHeaderType Structure**

The wimaxUSIOperatorHeaderType identifies the set of multiple web service operation requests.

Element	Type	Occurrence	Description
messageId	xsd:long	1	10.9.1.1
ipAddress	xsd:string	0-1	10.9.1.2
macAddress	xsd:string	0-1	10.9.1.3

**10.9.2.2 wimaxUSIOperatorRequestType Structure**

The wimaxUSIOperatorRequestType contains a **single** Web Service Operation request provided by USI System (Section 10.8 a).

Element	Type	Occurrence	Description / Clause defined
commandId	xsd:long	1	10.9.1.4
timeStamp	Xsd:dateTime	1	10.9.1.5
createQoSSessionRequest	createQoSSessionRequest	0-1	10.2.3.1.1
modifyQoSSessionRequest	modifyQoSSessionRequest	0-1	10.2.3.1.3
terminateQoSSessionRequest	terminateQoSSessionRequest	0-1	10.2.3.1.5
queryQoSSessionRequest	queryQoSSessionRequest	0-1	10.2.3.1.7

## USI

Element	Type	Occurrence	Description / Clause defined
	uest		
startQoSEventNotificationRequest	startQoSEventNotificationRequest	0-1	10.2.3.1.9
stopQoSEventNotificationRequest	stopQoSEventNotificationRequest	0-1	10.2.3.1.11
ePaymentAuthzRequest	ePaymentAuthzRequest	0-1	10.4.1
ePaymentChargeRequest	ePaymentChargeRequest	0-1	10.4.3
AttributeQueryRequest	AttributeQueryRequest	0-1	10.5.1
ContentPushRequest	ContentPushRequest	0-1	10.6.1

**10.9.2.3 wimaxUSIOperatorRequestBodyType Structure**

The wimaxUSIOperatorRequestBodyType contains 1 or more wimaxUSIOperatorRequestType.

Element	Type	Occurrence	Description
Request	wimaxUSIOperatorRequestType	1-Unbounded	10.9.2.2

**10.9.2.4 wimaxUSIOperatorRequestMessageType Structure**

The wimaxUSIOperatorHeaderType identifies the set of multiple web service operation requests contained in wimaxUSIOperatorRequestBodyType.

Element	Type	Occurrence	Description
Header	wimaxUSIOperatorHeaderType	1	10.9.2.1
Body	wimaxUSIOperatorRequestBodyType	1	10.9.2.3

## USI

**10.9.2.5 wimaxUSIOperatorResponseType Structure**

The wimaxUSIOperatorResponseType contains a **single** Web Service Operation response provided by USI System (Section 10.8 a).

Element	Type	Occurrence	Description / Clause defined
commandId	xsd:long	1	10.9.1.4 – should match the corresponding request commanded
timestamp	Xsd:dateTime	1	10.9.1.5
createQoSSessionResponse	createQoSSessionResponse	0-1	10.2.3.1.2
modifyQoSSessionResponse	modifyQoSSessionResponse	0-1	10.2.3.1.4
terminateQoSSessionResponse	terminateQoSSessionResponse	0-1	10.2.3.1.6
queryQoSSessionResponse	queryQoSSessionResponse	0-1	10.2.3.1.8
startQoSEventNotificationResponse	startQoSEventNotificationResponse	0-1	10.2.3.1.10
stopQoSEventNotificationResponse	stopQoSEventNotificationResponse	0-1	10.2.3.1.12
ePaymentAuthzResponse	ePaymentAuthzResponse	0-1	10.4.2
ePaymentChargeResponse	ePaymentChargeResponse	0-1	10.4.4
AttributeQueryResponse	AttributeQueryResponse	0-1	10.5.2
ContentPushResponse	ContentPushRequest	0-1	10.6.2

**10.9.2.6 wimaxUSIOperatorResponseBodyType Structure**

The wimaxUSIOperatorResponseBodyType contains 1 or more wimaxUSIOperatorResponseType.

Element	Type	Occurrence	Description
Response	wimaxUSIOperatorResponseType	1- Unbounded	10.9.2.5 <a href="#">wimaxUSIOperatorRequestType_Structure</a>

**10.9.2.7 wimaxUSIOperatorResponseSetMessageType Structure**

The wimaxUSIOperatorHeaderType identifies the set of multiple web service operation requests contained in wimaxUSIOperatorRequestBodyType.

Element	Type	Occurrence	Description
Header	wimaxUSIOperatorHeaderType	1	10.9.2.1
Body	wimaxUSIOperatorResponseBodyType	1	10.9.2.6

**10.9.3 Message Definition for Web Service Operations****10.9.3.1 Web Service Operations provided by USI System****10.9.3.1.1 ProcessWimaxUSIOperator Request**

Parameter	Type	Occurrence	Description / Clause defined
wimaxUSIOperatorRequestMessage	wimaxUSIOperatorRequestMessageType	1	10.9.2.6 10.9.2.4

**10.9.3.1.2 ProcessWimaxUSIOperator Response**

Parameter	Type	Occurrence	Description / Clause defined
wimaxUSIOperatorResponseMessage	wimaxUSIOperatorResponseMessageType	1	10.9.2.7

**10.9.4 iASP Type Definitions****10.9.4.1 wimaxUSIAspHeaderType**

The wimaxUSIAspHeaderType identifies the set of multiple web service operation requests.

Element	Type	Occurrence	Description
messageId	xsd:long	1	10.9.1.1

## USI

Element	Type	Occurrence	Description
ipAddress	xsd:string	0-1	10.9.1.2
macAddress	xsd:string	0-1	10.9.1.3

10.9.1.1 10.9.1.2 10.9.1.3 10.9.1.4 10.9.1.5

#### 10.9.4.2 wimaxUSIAspRequestType Structure

The wimaxUSIAspRequestType contains a single Web Service Operation request provided by iASP System (Section 10.8 b).

Element	Type	Occurrence	Description / Clause defined
commandId	xsd:long	1	10.9.1.4
timeStamp	Xsd:dateTime	1	10.9.1.5
queryApplicationSessionRequest	queryApplicationSessionRequest	0-1	10.2.3.2.1
notifyQoSEventRequest	notifyQoSEventRequest	0-1	10.2.3.2.3
notifyAccountingRequest	notifyAccountingRequest	0-1	10.3.3.1.1

#### 10.9.4.3 wimaxUSIAspRequestBodyType Structure

The wimaxUSIAspRequestBodyType contains 1 or more wimaxUSIAspRequestType.

Element	Type	Occurrence	Description
Request	wimaxUSIAspRequestType	1-Unbounded	<a href="#">10.8.4.2</a>

#### 10.9.4.4 wimaxUSIAspRequestMessageType Structure

The wimaxUSIAspHeaderType identifies the set of multiple web service operation requests contained in wimaxUSIAspRequestBodyType.

Element	Type	Occurrence	Description
Header	wimaxUSIAspHeaderType	1	10.9.4.310.9.4.1
Body	wimaxUSIAspRequestBodyType	1	10.9.4.3

#### 10.9.4.5 wimaxUSIAspResponseType Structure

The wimaxUSIAspResponseType contains a single Web Service Operation response provided by USI System (Section 10.8 a).



## USI

Element	Type	Occurrence	Description / Clause defined
commanded	xsd:long	1	10.8.1.4 – should match the corresponding request commanded
timestamp	xsd:dateTime	1	10.9.1.510.9.4.5
queryApplicationSessionResponse	queryApplicationSessionResponse	0-1	10.2.3.2.2
notifyQoSEventResponse	notifyQoSEventResponse	0-1	10.2.3.2.4
notifyAccountingResponse	notifyAccountingResponse	0-1	10.3.3.1.2

**10.9.4.6 wimaxUSIAspResponseBodyType Structure**

The wimaxUSIAspResponseBodyType contains 1 or more wimaxUSIAspResponseType.

Element	Type	Occurrence	Description
Response	wimaxUSIAspResponseType	1- Unbounded	10.9.4.5 <a href="#">wimaxUSIOperatorRequestType_Struct</a>

**10.9.4.7 wimaxUSIAspResponsesetMessageType Structure**

The wimaxUSIAspHeaderType identifies the set of multiple web service operation requests contained in wimaxUSIAspRequestBodyType.

Element	Type	Occurrence	Description
Header	wimaxUSIAspHeaderType	1	10.9.2.110.9.2.6
Body	wimaxUSIAspResponseBodyType	1	10.9.2.6

**10.9.5 Message Definition for iASP Web Service Operations****10.9.5.1 Web Service Operations provided by iASP System****10.9.5.1.1 ProcessWimaxUSIAspRequest**

Parameter	Type	Occurrence	Description / Clause defined
wimaxUSIAspRequestMessage	wimaxUSIAspRequestMessageType	1	10.9.4.410.9.4.7

USI

1 **10.9.5.1.2 ProcessWimaxUSIAsp Response**

Parameter	Type	Occurrence	Description / Clause defined
wimaxUSIAspResponseMessage	wimaxUSIAspResponseType	1	10.9.4.7

---

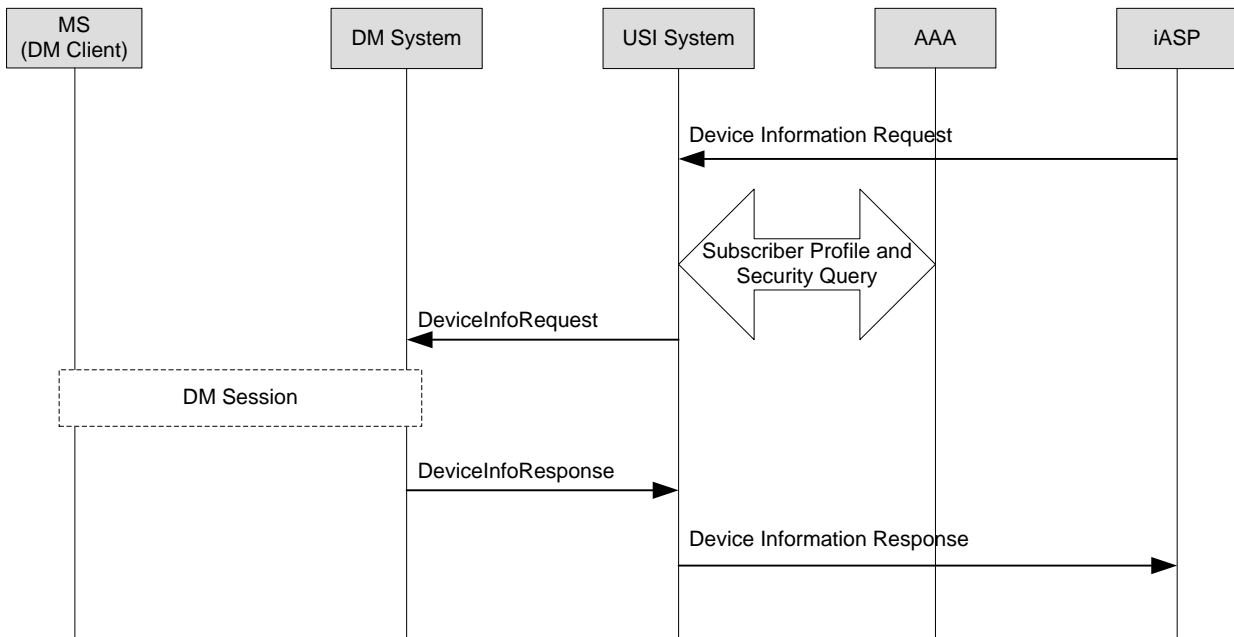
1 **11. Lawful Intercept**

- 2 The lawful Intercept aspects and details are specified in [NALI].

## Annex A (Informative): Interworking with DM server

This is an example flow of obtaining device capability information from Device Management System. Since the DM Server in WiMAX network already keeps certain types of information as a form of Management Object (MO), the USI server can provide the information to iASP by retrieving the related MOs from DM server as shown in Figure A.1.

Responding to a Device Information Request from an authenticated and authorized iASP over the U1 interface, the USI Server MAY request the related information to the DM System. Depending on the type of the capabilities, the DM Server MAY retrieve the stored values or query to the DM client in the MS to obtain up-to-date values. After gathering the related values, the DM Server replies the type-value pair of the capabilities and, in turn, the USI System sends a Device Information Response to the iASP.

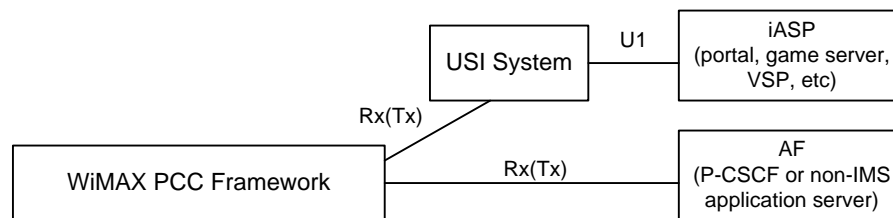


**Figure A.1: USI Interworking with DM Server**

## Annex B (Informative): Interworking with Policy & Charging Control (PCC)

USI System MAY optionally interwork with WiMAX PCC framework as the Dynamic QoS Subsystem. Figure B.1 shows how an USI System can interwork with a PCC system. The interface between USI System and PCC system is outside the scope of this document. Such an interface can be based on Rx(Tx) when interworking with the existing PCRF of the WiMAX PCC framework.

NOTE: In this case the capability of the WiMAX network defined by the parameters of U1 interface MAY be restricted by those of Rx(Tx) interface.



**Figure B.1: USI System Interworking with WiMAX PCC Framework**

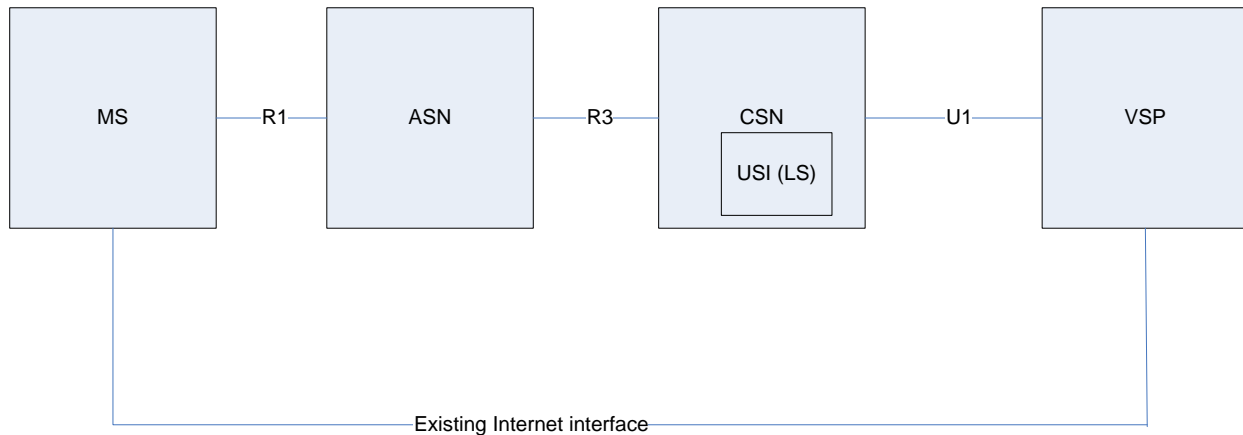
## Annex C (Informative): Commercial VoIP Support Using USI

Typically VoIP has been an application that is deployed by the network service provider itself.

With the changing landscape of VoIP, several VoIP service providers have started providing VoIP for users on the internet. However, these types of VoIP services are not nearly “commercial VoIP” with regards to the WiMAX end user, due to the following reasons:

- a) QoS on the WiMAX access link is neither negotiated nor guaranteed.
- b) Emergency calling support is typically not available.
- c) Lawful intercept capabilities are not provided to the Law enforcement agencies

However, using USI, we SHOULD be able to provide “commercial VoIP” services from third party VoIP service providers (VSP) with high QoS to the WiMAX and other aspects typically available in a commercial VoIP service. Figure C.1 shows the architecture of commercial VoIP using USI.



**Figure C.1: Commercial VoIP Support using USI**

The key interface is U1 – which is the interface between the VoIP service provider (VSP) and the Network service Provider (NSP) aka CSN.

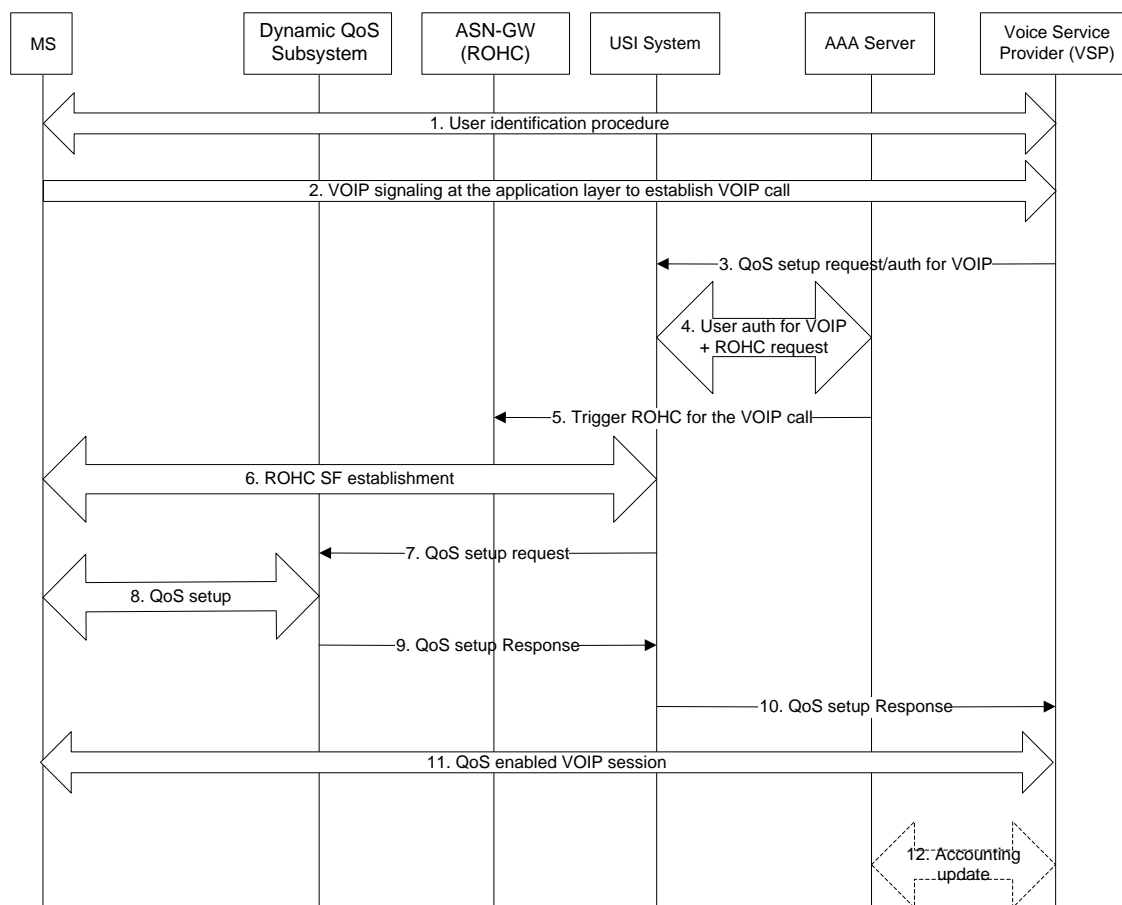
### C.1. Key aspects of Commercial VoIP using USI

The key aspects for a USI enabled VoIP service is as follows:

- a) call establishment – MS originated
- b) call establishment – MS terminated.
- c) Handover of the VoIP call with the user moves.
- d) Charging/billing
- e) Emergency services
- f) Lawful intercept.

## C.2. MS-Originated VoIP Call Establishment

The call establishment is shown in Figure C.2. ROHC MAY be used to optimize the data path for the VoIP call.



**Figure C.2: MS-Originated VoIP Call Establishment**

Step-1: the User identification is performed upon the registration of the MS.

Step-2: at some point in time after user identification, the MS signals to the VSP at the application layer to establish a VoIP call.

Step-3: the VSP authorizes the request with the USI in the NSP. As part of this step, it also requests proper QoS to be set up for the VoIP call

Step-4: USI contacts the AAA server for authorization for the VoIP and ROHC request

Step-5: the AAA triggers the ROHC function in the ASN-GW for establishing the ROHC enabled SF.

Step-6: the ROHC enabled SF is created as specified in the NWG ROHC specification [ROHC]

Step-7: Upon successful authorization, the USI requests the QoS to be setup for the VoIP call from the Dynamic QoS sub-system

## USI

- Step-8: QoS setup happens via the Dynamic QoS sub-system using the currently defined QoS primitives
- Step-9 & 10: successful setup of the QoS is sent from the Dynamic QoS sub- to the USI and onto VSP
- Step-11: QoS enabled, ROHC compressed VoIP call is established
- Step-12: Accounting update is performed.
- NOTE: Steps [5,6] and [7,8,9] MAY happen in parallel

### C.3. MS-Terminated VoIP Call Establishment

In this architecture, when the MS needs to be paged by the VSP for a VoIP call, the paging SHALL happen transparent to the USI in the event that the VSP has the IP address of the MS in idle mode, as defined in NWG specifications.

In the event that IP address of the idle MS is not available at the VSP (potentially due to time outs or other reasons), the VSP SHALL contact the USI with the pseudo identity of the MS. The USI SHALL lookup the MS and contact the Anchor DPF of the MS in the ASN to initiate the paging operations. Once the Anchor DPF has the data availability indication for the MS, the paging operations are the same as in NWG specifications.

Furthermore ROHC MAY be used to optimize the data path for the VoIP call.

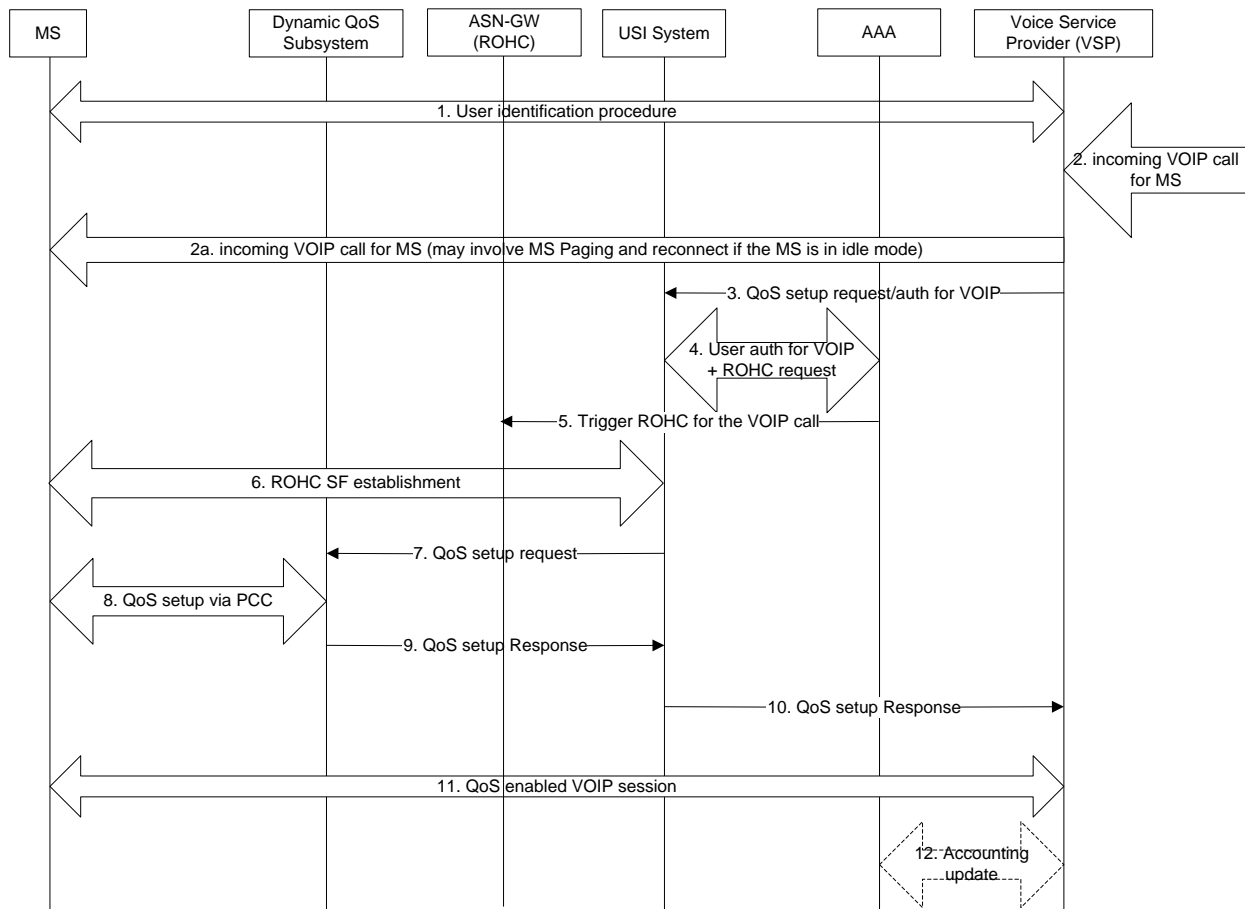


Figure C.3: MS-Terminated VoIP Call Establishment



## USI

The call establishment is shown in Figure C.3. Here it is assumed that the VSP has the latest IP address of the MS.

Step-1: the User identification is performed upon the registration of the MS.

Step-2: at some point in time after user identification, the VSP receives a call for the MS. In step 2a, the VSP contacts the MS to set up this call, using the MS's IP address. As part of this step, if the MS is in idle state, paging MAY be involved, in a manner that is transparent to the VSP, as detailed in the NWG specification.

Step-3: the VSP authorizes the request with the USI in the NSP. As part of this step, it also requests proper QoS to be set up for the VoIP call

Step-4: USI contacts the AAA server for authorization for the VoIP and ROHC request.

Step-5: the AAA, triggers the ROHC function in the ASN-GW for establishing the ROHC enabled SF.

Step-6: the ROHC enabled SF is created as specified in the NWG ROHC specification [ROHC]

Step-7: Upon successful authorization, the USI requests the QoS to be setup for the VoIP call from the PCC system.

Step-8: QoS setup happens via the PCC system using the currently defined QoS primitives

Step-9&10: successful setup of the QoS is sent from the PCC to the USI and onto VSP

Step-10: VoIP call is established

Step-11: Accounting update is performed.

NOTE: Steps [5,6] and [7,8,9] MAY happen in parallel

### C.3 Handover Support

Handovers are handled transparent to the iASP as specified in NWG specifications.

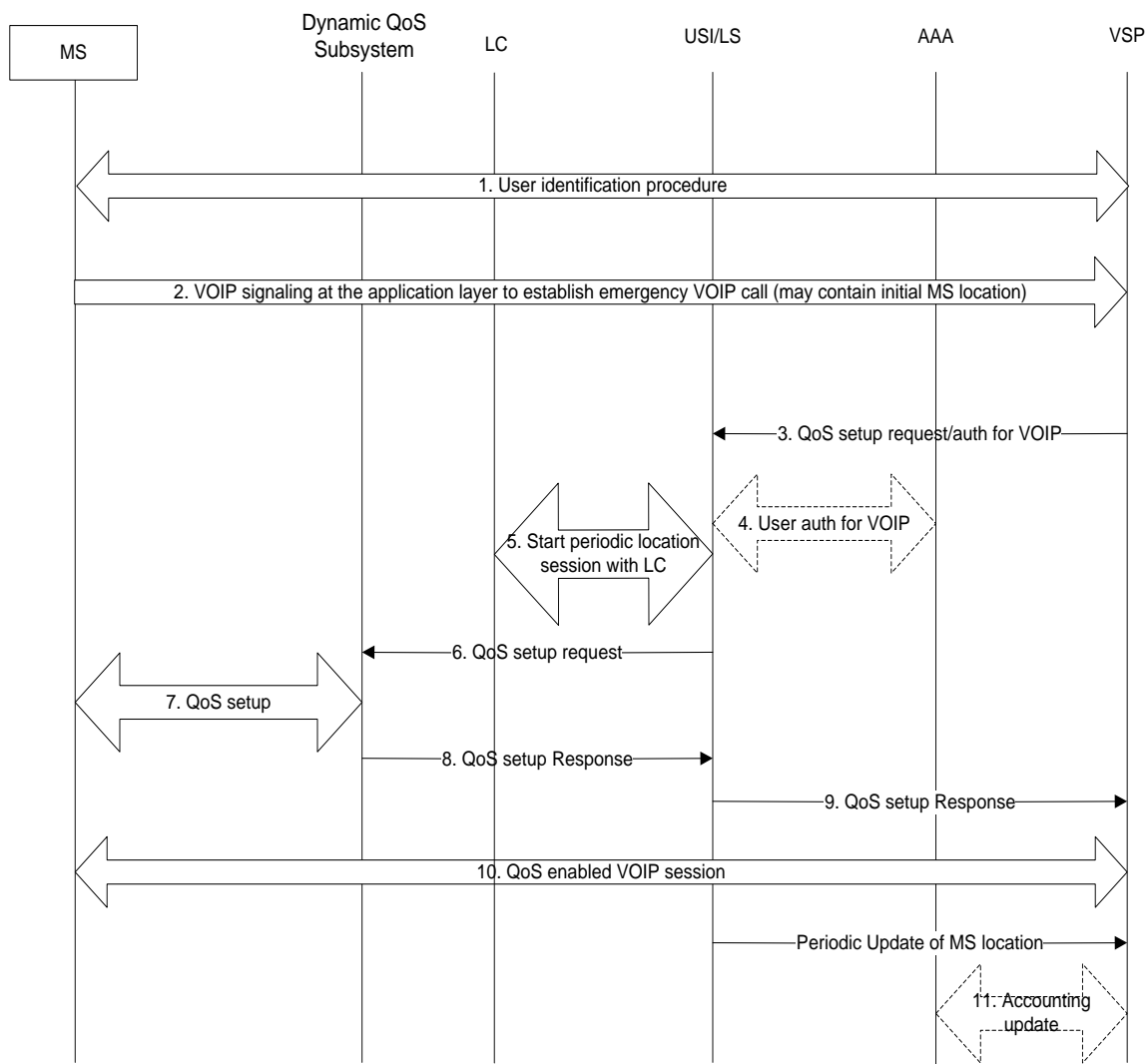
### C.4. Accounting Support

The charging aspects are specified in Section 9.

### C.5. Emergency Services

The call flow for the MS making an emergency call is as shown in Figure C.4. Here the VSP interfaces and routes to the PSAP based on the MS location. In this scenario, the VSP requires not just the QoS for the VoIP call, but also the MS location.

## USI



**Figure C.4: Handling Emergency Services**

Step-1: the User identification is performed upon the registration of the MS.

Step-2: at some point in time after user identification, the MS signals to the VSP at the application layer to establish an emergency VoIP call. This message will contain the "emergency indication". It MAY also contain the initial location of the MS.

Step-3: the VSP authorizes the request with the USI in the NSP.

Step-4: USI contacts the AAA server for authorization. This step is optional and depends upon respective national and regional regulations.

Step-5: the Periodic location session for the MS is stated with the location controller of the MS.

Step-6,7,8,9: QoS setup for the emergency call as in the previous case via the PCC system.

Step 10: VoIP call is established. During the VoIP call, the MS location MAY be periodically updated.

Step 11: Accounting update is performed.

USI

1  
2  
3  
4  
5  
6  
7  
8

## **C.6. Lawful intercept**

Lawful intercept is handled as part of the LI specification [NALI].

Essentially for the commercial VoIP service, the LIA (LI agent) or the intercept access point SHALL be present in the VSP. The LIS (LI server) or the delivery function can be present either in the VSP or NSP.

## Annex D (Informative): Interworking with Location Based Services (LBS)

LBS can be easily enabled via USI by co-locating the Location Server (LS) within the USI System. However, the deployments may have a standalone LS that is not collocated with the USI System. This annex addresses such a scenario where interworking is enabled between the LS and USI.

### Determining MS Location in Non-Roaming Scenario

Figure D1 shows the location determination for the MS from the USI perspective when LS is not collocated with the USI System.

Step 1: The iASP could request the location of the MS from the USI System.

Step 2: The USI System passes the incoming location request to the LS

Step 3: The LS in turn authenticates the request for location. This is an optional procedure.

Step 4: If the location request can be properly authenticated, then the LS proceeds to determine the location of the MS from the ASN (as described in [LBS]).

Step 5: The LS calculates the location of the MS and passes it on to the USI System.

Step 6: The USI System then provides the location of the MS to the iASP and optionally performs an accounting update.

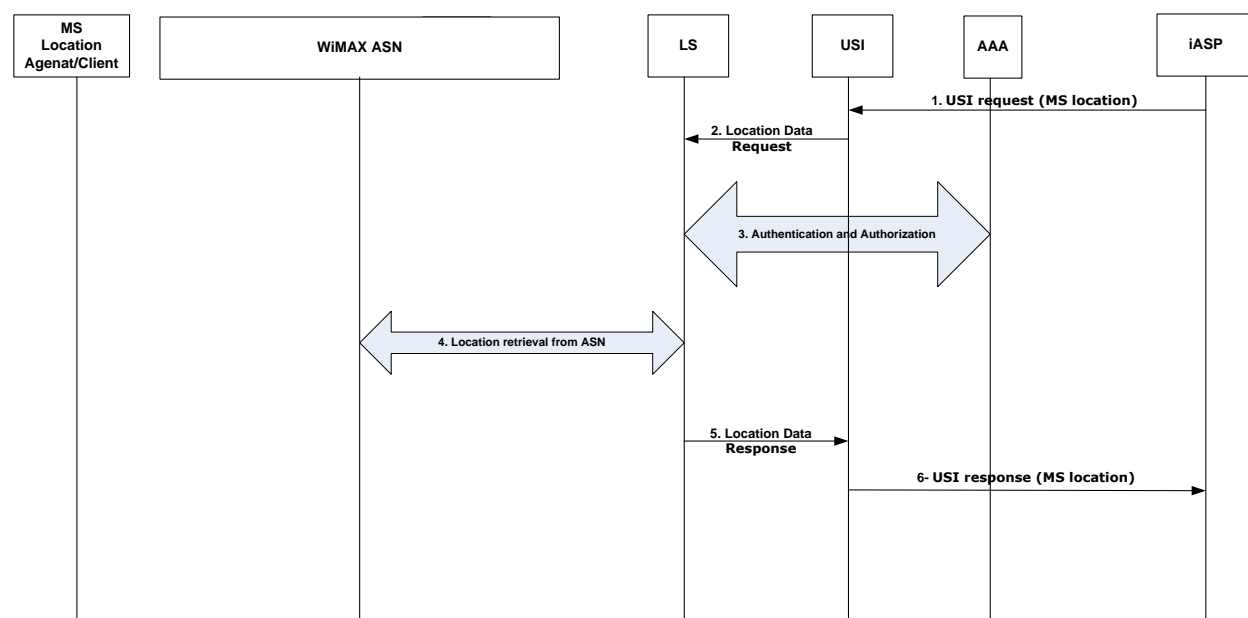
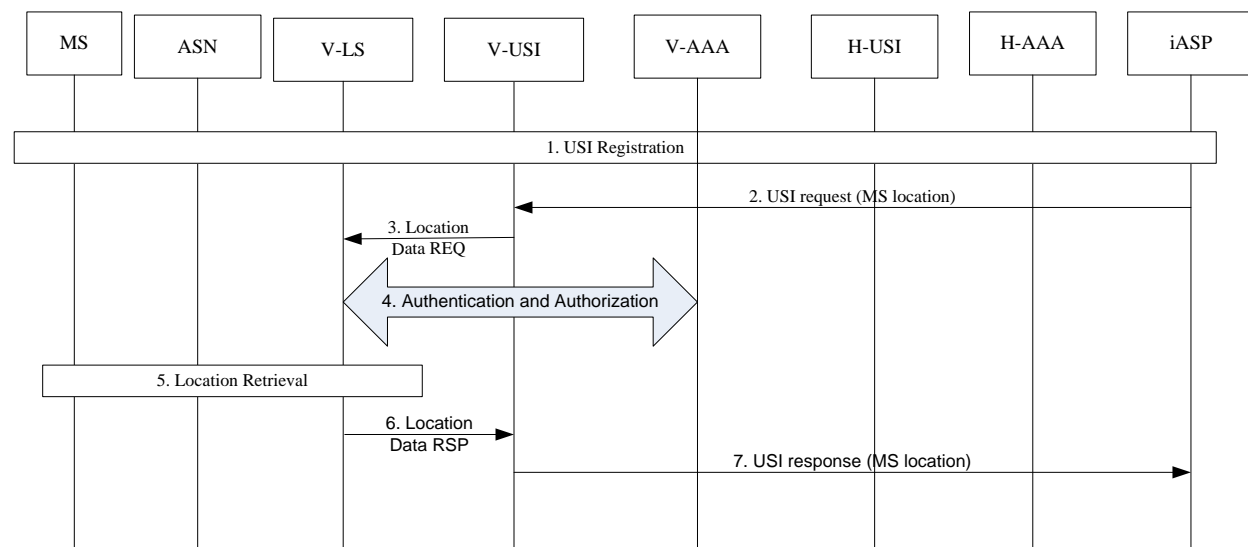


Figure D.1: Determining MS location via USI (non-Roaming)

## USI

**Determining MS Location in Roaming Scenarios**

For the roaming case of Local USI Services in Roaming, the procedure of determining MS location is as following:

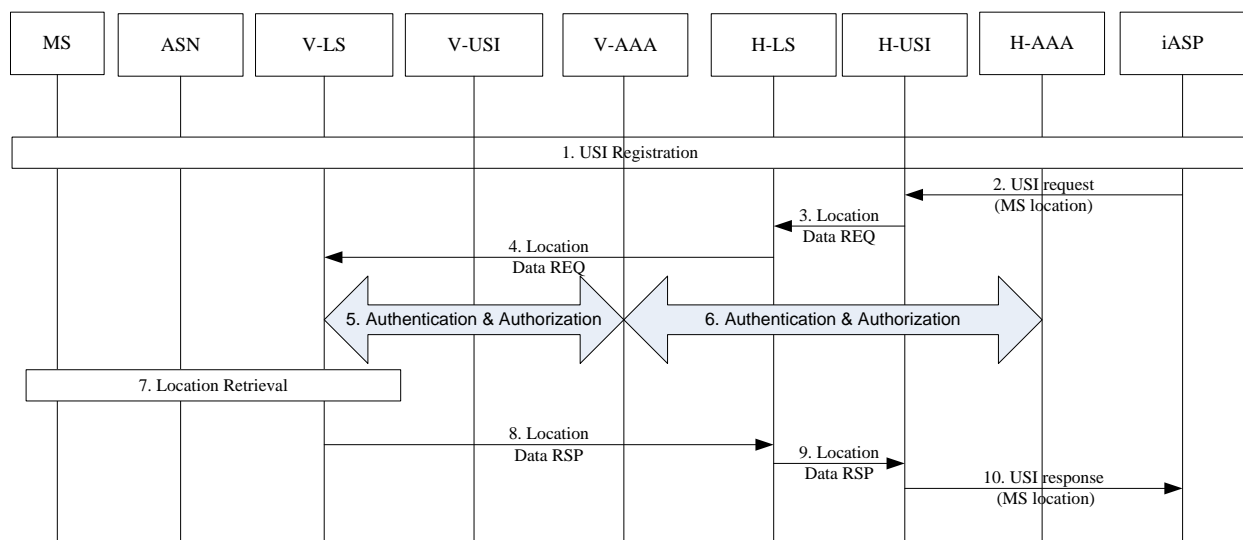


**Figure D.2: Determining MS location via USI (Local USI Services in Roaming)**

1. USI Registration procedure. The USI registration is triggered by iASP or network. The user's USI profile or registration information is delivered to V-USI by H-USI after the successful USI registration.
2. iASP sends USI context request message to V-USI to get MS's location information. The MS's USI ID, e.g. short-lived USI ID or long-lived USI ID SHALL be included in this message to be used to identify the MS, and the short-lived USI ID SHOULD be used in priority if both USI ID are available.
3. The V-USI triggers the Visited Location server (V-LS) to get the requested location information. The location determination will be done as specified in [LBS].
4. The V-LS authenticates the incoming location request with the V-AAS. This is an optional procedure.
5. If the location request can be properly authenticated, then the V-LS proceeds to determine the location of the MS from the ASN (as described in [LBS]).
6. The V-LS calculates the location of the MS and passes it on to the V-USI System.
7. V-USI responds to the iASP with the MS location via the USI context response message.

For the roaming case of Home USI Services in Roaming, the procedure of determining MS location is as following:

## USI



**Figure D.3: Determining MS location via USI (Home USI Services in Roaming)**

1. USI Registration procedure. The USI registration is triggered by iASP or network. The user's USI profile or registration information is delivered to V-USI by H-USI after the successful USI registration.
2. iASP sends USI context request message to H-USI to get MS's location information. The MS's USI ID, e.g. short-lived USI ID or long-lived USI ID SHALL be included in this message to be used to identify the MS, and the short-lived USI ID SHOULD be used in priority if both USI ID are available.
3. H-USI forwards the location to H-LS.
4. The H-LS determines that the MS is in Visited network, and forwards the incoming location Data REQ to the V-LS.
5. The V-LS authenticates the incoming location request with the V-AAS. This is an optional procedure.
6. The V-AAS may further authenticate the location request with H-AAS. This is an optional procedure.
7. If the location request can be properly authenticated, then the V-LS proceeds to determine the location of the MS from the ASN (as described in the LBS specification for WiMAX).
8. The V-LS calculates the location of the MS and passes it on to the H-LS.
9. The H-LS passes the location onto the H-USI
10. The H-USI forwards the MS location result to iASP.