Hotspot 2.0 Specification

Version 2.0

This document is the specification for the
Wi-Fi Alliance Wi-Fi CERTIFIED Passpoint™ (Release 2) program,
a solution for next generation Wi-Fi® hotspots.

Document History

Version	Date	Status	Comments
1.0	2014-08-08	Final	Public release version
1.1	2015-02-03	Final	- Editorial fixes to section 1.2 (References) - Update to Figure 83
1.2	2016-12-08	Final	Merged in: Hotspot_2-0_(R2)_Technical_Specification_v1-1-0-optional-omadm-draft2.docx Hotspot_2-0_(R2)_Technical_Specification_v1-1-0-sta-filtering.docx Updates for optional ICON processing
2.0	2019-02-13	Final	Renamed title and version.

Table of Contents

List of Figures

List of Tables

1. Overview

This document is the technical specification for Wi-Fi CERTIFIED Passpoint™ (Release 2), the Wi-Fi Alliance certification program that provides WPA2™ hotspot network access and online sign up. Hotspot 2.0 enables a secure, automatic connection experience for users and supports operator goals of leveraging Wi-Fi® technology for data offload of cellular networks.

1.1 Scope

The scope of the feature requirements is limited to that defined in this specification.

The Passpoint certification program plans to have multiple releases, the first two of which are:

- Release 1 – Network Selection and Security
- Release 2 – Network Selection and Security, and Online Signup and Policy Provisioning

This specification addresses Release 2 requirements. Release 2 is a superset of Release 1 and includes all feature requirements of Release 1.

1.2 References

The following referenced documents form a normative part of this specification to the extent specified herein. In the event of a conflict between this specification and the following referenced documents, the contents of this specification take precedence.

[1] Wi-Fi Peer-to-Peer (P2P) Technical Specification, Version 1.1, October 2010, http://www.wi-fi.org/discover-wi-fi/wi-fi-direct

[2] IEEE 802.11-2012, Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications, March 2012

[3] RFC-791, Internet Protocol, September 1981

[4] RFC-826, An Ethernet Address Resolution Protocol, November 1982

[5] RFC 1035, Domain Names - Implementation and Specification, Mockapetris, November 1987

[6] RFC-2460, Internet Protocol, Version 6 (IPv6) Specification, Deering and Hinden, December 1998

[7] RFC-2560, X.509 Internet Public Key Infrastructure Online Certificate Status Protocol - OCSP, Myers, Ankney et al, June 1999

[8] RFC-2616, Hypertext Transfer Protocol -- HTTP/1.1, Fielding, Gettys, Mogul, Frystyk, Masinter, Leach and Berners-Lee, June 1999

[9] RFC-2617, HTTP Authentication: Basic and Digest Access Authentication, Franks, Hallam-Baker, Hostetler, Lawrence, Leach, Luotonen, Stewart, June 1999

[10] RFC-2759, Microsoft PPP CHAP Extensions, Version 2, Zorn, January 2000

[11] RFC-2818, HTTP Over TLS, Rescorla, May 2000

[12] RFC-2985, PKCS #9: Selected Object Classes and Attribute Types Version 2.0, Nystrom, Kaliski, November 2000

[13] RFC-3447, Public-Key Cryptography Standards (PKCS) #1: RSA Cryptography Specifications Version 2.1, Jonsson and Kaliski, February 2003

[14] RFC-3629, UTF-8, A Transformation Format of ISO 10646, Yergeau, November 2003

[15] RFC-3709, Internet X.509 Public Key Infrastructure: Logotypes in X.509 Certificates, Santesson, Housley, Freeman, February 2004

[16] RFC-3748, Extensible Authentication Protocol (EAP), Aboba, Blunk, Vollbrecht, Carlson and Levkowetz, June 2004

[17] RFC 3986, Uniform Resource Identifier (URI): Generic Syntax, Berners-Lee, Fielding and Masinter, January 2005

[18] RFC-4034, Resource Records for DNS Security Extensions, Arends, Austein, Larson, Massey, and Rose, March 2005

[19] RFC-4035, Protocol Modifications for the Domain Name System Security Extensions, Arends, Austein, Larson, Massey and Rose, March 2005

[20] RFC-4186, Extensible Authentication Protocol Method for Global System for Mobile Communications (GSM) Subscriber Identity Modules (EAP-SIM), Haverinen and Salowey, January 2006

[21] RFC-4187, Extensible Authentication Protocol Method for 3rd Generation Authentication and Key Agreement (EAP-AKA), Arkko and Haverinen, January 2006

[22] RFC-4282, The Network Access Identifier, Aboba, Beadles, Arkko, and Eronen, December 2005

[23] RFC-4288, Media Type Specifications and Registration Procedures, Freed and Klensin, December 2005

[24] RFC-4861, Neighbor Discovery for IP version 6 (IPv6), November 2007

[25] RFC-5216, The EAP-TLS Authentication Protocol, Simon, Aboba and Hurst, March 2008

[26] RFC-5227, IPv4 Address Conflict Detection, Cheshire, July 2008

[27] RFC-5280, Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile, Cooper, May 2008

[28] RFC-5281, Extensible Authentication Protocol Tunneled Transport Layer Security, Authenticated Protocol Version 0 (EAP-TTLSv0), Funk and Blake-Wilson, August 2008

[29] RFC-5246, The Transport Layer Security (TLS) Protocol, Version 1.1, Dierks and Rescorla, April 2006

[30] RFC-5448, Improved Extensible Authentication Protocol Method for 3rd Generation Authentication and Key Agreement (EAP-AKA'), Arkko, Lehtovirta and Eronen, May 2009

[31] RFC-5746, Transport Layer Security (TLS) Renegotiation Indication Extension, Rescorla, Ray, Dispensa, Oskov, February 2010

[32] RFC-5967, The application/pkcs10 Media Type, Turner, August 2010

[33] RFC-6066, Transport Layer Security (TLS) Extensions: Extension Definitions, Eastlake, January 2011

[34] RFC-6204, Basic Requirements for IPv6 Customer Edge Routers, Singh et al, April 2011

[35] RFC-6961, The Transport Layer Security (TLS) Multiple Certificate Status Request Extension, Y. Pettersen, June 2013

[36] RFC-7030, Enrollment over Secure Transport, Pritikin, M. et al, June 10, 2013

[37] 3GPP TS 23.003, Technical Specification Group Core Network and Terminals; Numbering, addressing and identification, December 2011

[38] OMA Device Management Tree and Description, Version 1.2.1, OMA-TS-DM_TND-V1_2_1-20080617-A.pdf, June 2008

[39] Open Mobile Alliance (OMA) Device Management Protocol, Version 1.2, OMA-TS-DM_Protocol-V1_2_1-20080617-A.pdf, June 2008

[40] Open Mobile Alliance (OMA) Device Description Framework, Version 1.2, OMA-SUP-dtd_dm_ddf-V1_2-20070209-A.dtd, February 2007

[41] Open Mobile Alliance (OMA) Management Standardized Objects, Version 1.2.1, OMA-TS-DM_StdObj-V1_2_1-20080617-A.pdf, June 2008

[42] ISO 8601:2004, Data elements and interchange formats -- Information interchange -- Representation of dates and times, 2004

[43] Simple Object Access Protocol, SOAP Version 1.2 Part 1: Messaging Framework, W3C, Gudgin, Hadley, Mendelsohn, Moreau, Nielsen, June 2003

[44] ITU-T, E.212, Series E: Overall Network Operation, Telephone Service, Service Operation and Human Factors, November 1998

[45] Wi-Fi Alliance Online Sign-Up Certificate Policy Specification, Version 1.1, http://www.wi-fi.org/passpoint

[46] SP800-57, Recommendations for Key Management—Part 1: General, National Institute of Science and Technology (NIST), March 2007

[47] IANA EAP number assignments, http://www.iana.org/assignments/eap-numbers/eap-numbers.xml

[48] IANA vendor specific assignments, http://www.iana.org/assignments/enterprise-numbers

[49] Open Mobile Alliance (OMA) Device Management Representation Protocol, Approved Version 1.2.1, OMA-TS-DM_RepPro-V1_2_1-20080617-A.pdf, June 2008

[50] Subscription Provisioning Protocol (SPP) Web Services Description Language (WSDL) file, Version 1.0

[51] Subscription Provisioning Protocol (SPP), Version 1.0

[52] Wi-Fi Alliance Registration Protocol (REP), Version 1.0

[53] Open Mobile Alliance (OMA) Device Management Tree and Description Serialization, Version 1.2, OMA-TS-DM_TNDS-V1_2-20070209-A.pdf, February 2007

[54] Wi-Fi CERTIFIED Passpoint™ (Release 1) Operator Best Practices for AAA Interface Deployment Version 1.0.1, http://www.wi-fi.org/passpoint

[55] ISO 639-2, also see ISO-639 guidance, for further guidance on how to encode country codes

[56] ITU-T X.690, ISO/IEC 8825-1, Information technology – ASN.1 encoding rules: Specification of Basic Encoding Rules (BER), Canonical Encoding Rules (CER) and Distinguished Encoding Rules (DER), July 2002

[57] Wi-Fi CERTIFIED Passpoint™ (Release 2) Interoperability Test Plan, https://www.wi-fi.org/members/certifications-testing/test-plans

[58] IEEE 802.1ar 2009, http://standards.ieee.org/findstds/standard/802.1AR-2009.html

1.3 Definitions, abbreviations and acronyms

1.3.1 Definitions

The following definitions are applicable to this specification:

Access Network Query Protocol (ANQP) server: An advertisement server (see [2]) in the hotspot operator's network that contains ANQP elements or information that can be used to derive the required ANQP elements. The information in the ANQP server can be obtained by the access network query protocol. An ANQP server is a functional entity that supports proxy relationships with other ANQP servers. An ANQP server can be co-located with an AP or in an external device. Throughout this specification where the text describes an ANQP element as being provided by an AP, it is to be understood that the source of the message is an ANQP Server.

access point (AP): Device or set of devices that instantiate(s) the required IEEE 802.11 logical functions including RAN, security and authentication, as defined in IEEE 802.11-2012 [2]. Additional control, user and management plane functions can also be included. Note: the term applies to both single network element implementation and distributed implementation in which multiple network elements are involved (e.g., a radio access node and its controller).

Access state: A mobile device state when the mobile device has successfully associated and authenticated with the hotspot and can access the services for which the user has subscribed.

at least: Indicates the expectation that a leaf node (see section 9.1.1) could be much longer, and that support for a minimum length is needed (no upper bound is specified), e.g., "The mobile device shall support Uniform Resource Identifier (URI) strings at least 1023 octets long".

captive portal: A mechanism for Wi-Fi Hotspot network access in which a Hypertext Transfer Protocol (HTTP) request from a mobile device is redirected to a server for authentication.

Certificate Authority (CA): A collection of computer hardware, software and the people who operate it. The CA is known by two attributes: its name and its public key. The CA performs four basic CA functions:

1. Issues certificates (i.e., creates and signs the certificates).
2. Maintains certificate status information and issues certificate revocation lists (CRLs)
3. Publishes its current (unexpired) certificates and CRLs so users can obtain the information they need to implement security services.
4. Maintains archives of status information about the expired or revoked certificates it issued.

Discovery state: A mobile device state when the mobile device is scanning for APs with which to associate and for related information useful for network selection.

Gratuitous Address Resolution Protocol (ARP): An ARP request or reply message, transmitted to the broadcast destination medium access control (MAC) address, that is not normally needed according to the ARP [4], but is useful for other purposes, such as detecting duplicate Internet protocol (IP) address assignments [26] or notifying other hosts of a change of IP address. If such a message is converted to an individual destination MAC address, it is no longer considered to be a gratuitous ARP by this specification.

Home Service Provider (SP): An SP with which a mobile device has a subscription and associated credentials. The Home SP bills the user and authenticates the mobile device.

Hotspot: A site that offers public access to packet data services (e.g., the Internet) via a Wi-Fi access network (AN).

Hotspot Operator: The entity that is responsible for the operation of the hotspot.

Provisioning state: A mobile device state when the Wi-Fi infrastructure is establishing credential information and providing policy information to the mobile device. If the mobile device already has valid credentials for a given hotspot, this state is ephemeral.

Registration Authority (RA): A collection of computer hardware, software and the people who operate it. The RA is known by two attributes: its name and its public key. The RA is responsible for the verification of certificate contents for the Certificate Authority (CA).

Registration data: The data necessary to sign up for a subscription. Registration data typically include selection of a rate plan, terms and conditions, subscriber's contact information and payment information (e.g., credit card, bank account number).

Registration state: A mobile device state when the mobile device is setting up a new account with an SP or hotspot provider. If the mobile device already has valid credentials for a given hotspot, this state is ephemeral.

Service Provider (SP): An entity offering network services (from the perspective of the Hotspot Operator). SPs are represented in the Network Access Identifier (NAI) Realm, 3GPP Cellular Network (in the form of a list of public land mobile network identifiers (PLMN IDs)) or Roaming Consortium ANQP-elements.

subscription remediation: The process of fixing a problem in the subscriber's subscription. This includes provisioning new credentials to a mobile device (e.g., due to expiration), updating the PerProviderSubscription management object (MO) on a mobile device (e.g., because data need updates) or performing an online function to update the subscription (e.g., pay a delinquent bill). Note that in the latter example no new credentials/data are provisioned to the mobile device.

terms and conditions (T&C) data: The data necessary to accept terms and conditions for network access. The data typically include the subscriber's contact information and an acceptance indication.

1.3.2 Abbreviations and Acronyms

Abbreviation	Definition
3GPP	3rd Generation Partnership Project
AAA	authentication, authorization and accounting
ACL	access control list
AKA	authentication and key agreement
AP	access point
AN	access network
ANA	Assigned Numbers Authority
ANQP	Access Network Query Protocol
ARP	Address Resolution Protocol
AS	Authentication Server
ASRA	additional step required for access
BSS	basic service set
BSSID	basic service set identifier
CA	Certificate Authority
CM	connection manager
CRL	certificate revocation list
CSR	certificate signing request
DDF	device description framework
DER	Distinguished Encoding Rules
DGAF	downstream group-addressed forwarding
DM	device management
DMAcc	device management account
DNS	domain name system
EAP	Extensible Authentication Protocol
EKU	extended key usage
ESP	Encapsulating Security Payload
ESS	extended service set
EST	Enrollment over Secure Transport (certificate)
FQDN	Fully Qualified Domain Name
GAS	generic advertisement service
GTK	group temporal key
HESSID	homogeneous extended service set (ESS) identifier
HS	Hotspot
HS2.0	Hotspot 2.0
HTTP	Hypertext Transfer Protocol
HTTPS	Hypertext Transfer Protocol Secure
IANA	Internet Assigned Numbers Authority
ICMP	Internet Control Message Protocol
ID	identifier
IEEE	Institute of Electrical and Electronics Engineers
IETF	Internet Engineering Task Force
IKE	Internet Key Exchange
IMSI	International Mobile Subscriber Identity
ITU-T	International Telecommunication Union Telecommunication Standardization Sector
IP	Internet Protocol
IPsec	Internet Protocol security
ISO	International Organization for Standardization
LMD	load measurement duration

MAC	medium access control
MCC	Mobile Country Code
MIME	Multipurpose Internet Mail Extensions
MNC	Mobile Network Code
MO	management object
MRD	marketing requirements document
MSDU	medium access control (MAC) service data unit
MSISDN	Mobile Subscriber Integrated Services Digital Network
NAI	Network Access Identifier
NAT	network address translation
ND&S	network discovery and selection
OCSP	Online Certificate Status Protocol
OI	organizational identifier
OMA	Open Mobile Alliance™
OSEN	online Sign Up (OSU) server-only authenticated layer 2 encryption network
OSENA	online sign up (OSU) server-only authenticated layer 2 encryption network association
OSU	online sign up
P2P	peer-to-peer (see [1])
PKI	Public Key Infrastructure
PLMN	public land mobile network
PMF	protected management frame
PPS MO	PerProviderSubscription management object
QoS	quality of service
RA	Registration Authority
RADIUS	Remote Authentication Dial In User Service
REP	registration protocol
RFC	Request for Comments
RSN	robust security network
RSNA	robust security network association
RSNE	Robust Security Network element
RSSI	receive signal strength indicator
SIM	subscriber identity module
SMI	Structure of Management Information
SOAP	Simple Object Access Protocol
SP	service provider
SPP	subscription provisioning protocol
SSH	secure shell
SSID	service set Identifier
T&C	terms and conditions
TKIP	temporal key integrity protocol
TLS	transport layer security
TTLS	tunneled transport layer security (TLS)
UDP	user datagram protocol
UI	user interface
URI	uniform resource identifier
URL	universal resource locator
USIM	universal subscriber identity module
VoIP	voice over Internet Protocol (IP)
VPN	virtual private network
WAN	wide area network
WEP	wired equivalent privacy
WLAN	wireless local area network
WNM	wireless network management
WPA2™	Wi-Fi Protected Access® version 2

2. Hotspot 2.0 Device, Operator and Service Provider requirements

Section 2.1 specifies the required capabilities for a Hotspot 2.0 (HS2.0) compliant access point (AP). Section 2.2 specifies the required capabilities for an HS2.0 compliant mobile device. Section 2.3 specifies the requirements applicable to Hotspot Operators, and section 2.4 specifies the requirements applicable to Service Providers (SPs).

To ensure that a mobile device always has an authentication method in common with Wi-Fi and SP infrastructure, the credential types and Extensible Authentication Protocol (EAP) methods identified in Table 1 shall be supported by HS2.0 equipment (see sections 2.1 and 2.2) and Service Providers (see section 2.4).

Table 1: Credential Types and EAP Methods

Credential Type	EAP Method
Certificate	EAP-TLS [25]
SIM/USIM	EAP-SIM [20], EAP-AKA [21], EAP-AKA' [30]
Username/Password (with server side certificates)	EAP-TTLS [28] with MSCHAPv2 [10]

Note: For the rest of this document “SIM” is used to refer to subscriber identity module / universal subscriber identity module (SIM/USIM) capability.

2.1 Required AP Capabilities

When an AP indicates support for HS2.0, it shall support the following capabilities:

- WPA2-Enterprise.
- All EAP methods listed in Table 1.
- The Interworking information element, including its Venue Info and Homogeneous Extended Service Set Identifier (HESSID) fields (defined in [2]); support for this element mandates support for the generic advertisement service (GAS).
- The Roaming Consortium information element; see [2].
- Setting the Interworking bit in the Extended Capabilities information element; see [2].
- The BSS Load element; see [2]. This element contains information on the current mobile device population and channel utilization in the BSS.
- The following ANQP-elements; see [2]:
    - Internet Protocol (IP) address type availability
    - Network Access Identifier (NAI) Realm
    - 3GPP Cellular Network
    - Domain Name
- The HS2.0 ANQP-elements (defined in section 4):
    - Hotspot (HS) Query List
    - HS Capability List
    - Operator Friendly Name
    - Wide Area Network (WAN) Metrics
    - Connection Capability
    - NAI Home Realm Query
    - Online Sign Up (OSU) Providers List
    - Icon Request
    - Icon Binary File

- The proxy address resolution protocol (ARP) service defined in section 5.3. The proxy ARP service has two purposes: 1) enabling the mobile device to remain in power save for longer periods of time, and 2) protecting against malicious behavior of an associated mobile device.
- When the value of the Access Network Type field in the Interworking element is either Free Public Network or Chargeable Public Network, all traffic inspection and filtering operate according to the procedures specified in section 5.1.
- The ability to disable downstream forwarding of group-addressed frames (i.e., multicast and broadcast frames) according to the procedures in section 5.2.
- The ability to disable P2P cross connect (see [1]) by advertising the P2P Manageability attribute with the Cross Connection Permitted field value 0.
- BSS Transition Management Request frame with the ESS Disassociation Imminent bit and Session Information universal resource locator (URL); see [2].
- The WNM-Notification Request Action frame (see section 3.2.1)
- ANQP responses up to a size of 65535 octets.
- The QoS mapping interworking services defined in subclause 10.24.9 of [2]. This includes the QoS Map Configure frame defined in [2].
- The ability to set the Additional Step Required for Access (ASRA) bit to 1, even though the value of dot11RSNAActivated may be true when a BSS advertises online sign up.
- The Country element in Beacon and Probe Response frames, except where prohibited by regulatory rules.
- The HS2.0 Release 2.0 AP shall be compliant with Release 1.
- Management Frame Protection per [2].
- RADIUS vendor-specific attributes, as defined in Annex D.

When an AP indicates support for HS2.0, it should have the following capability:

- Support for Remote Authentication Dial In User Service (RADIUS) attributes, as recommended in [54].

When an AP indicates support for HS2.0, it may have the following capabilities:

- Support for the following HS2.0 ANQP-element (defined in section 4):
    - Operating Class Indication

When an AP indicates support for HS2.0, it shall not use the following IEEE 802.11 security protocols:

- Temporal key integrity protocol (TKIP)
- Wired equivalent privacy (WEP)

2.2 Required Mobile Device Capabilities

When a mobile device associates to a BSS and includes the HS2.0 element in the (Re)Association Request frame, the mobile device shall support the following capabilities:

- WPA2-Enterprise.
- If the device has SIM/USIM credentials, it shall support all credential types and associated EAP methods listed in Table 1.
- If the device does not have SIM/USIM credentials, it shall support certificates and username/password credential types and their associated EAP methods listed in Table 1.
- The Interworking information element including the Venue Info and HESSID fields [2].
- The Roaming Consortium information element [2].
- Setting the Interworking bit in the Extended Capabilities information element [2].
- The BSS Load element; see [2]. This element contains information on the current mobile device population and channel utilization in the BSS.
- Filtering of frames encrypted using the group temporal key (GTK), according to the procedures in section 6.
- The following ANQP-elements; see [2]:

    - Network Authentication Type
    - Roaming Consortium
    - NAI Realm
    - 3GPP Cellular Network (only required for the mobile device having SIM credentials)
    - Domain Name
- The HS2.0 ANQP-elements (defined in section 4):
    - HS Query list
    - HS Capability list
    - Operator Friendly Name
    - WAN Metrics
    - Connection Capability
    - OSU Providers list
- Online Sign Up (see sections 7 and 8) and subscription provisioning using the SOAP XML protocol per section 8. This includes support for the PerProviderSubscription management object (MO), the DevDetail Wi-Fi Extension MO (see section 9) and EST (see section 7.6).
- The QoS mapping interworking services defined in section 10.24.9 of [2]. This includes the QoS Map Configure frame defined in [2].
- ANQP responses up to a size of 65535 octets.
- The mobile device procedures specified in section 6.
- The procedure in section 7.3.3.2, to validate the AAA Server certificate (when using EAP methods which employ a server certificate) when connecting to an HS2.0 network.
- The capability to determine time in order to validate certificate time and date requirements.
- The capability to be provisioned with and use multiple PerProviderSubscription MOs.
- Management Frame Protection per [2].
- If the device is a Release 2 mobile device, it shall be compliant with HS2.0 Release 1.
- The capability to handle BSS Transition Management Request, including support for the ESS Disassociation Imminent bit and the Session Information URL; see [2].

When a mobile device indicates support for HS2.0, it should support the following capabilities:

- The following ANQP-elements; see [2]:
    - Venue Name
    - IP Address Type Availability.

When a mobile device indicates support for HS2.0, it may support the following capabilities:

- The HS2.0 ANQP-elements (defined in section 4):
    - NAI Home Realm Query
    - Operating Class Indication
    - Parsing the standard tags embedded by OSU servers and responding with standard, embedded tags, as defined in Annex E.
- The use of OMA DM protocol per section 8 for Online Sign Up (see sections 7 and 8) and subscription provisioning.

When a mobile device indicates support for HS2.0 and the mobile device possesses a display, the following capabilities apply:

- The user interface (UI) may be capable of displaying an HS2.0 indicator. How the indicator is displayed is implementation dependent.
- The UI may indicate the credential type used by the mobile device to connect to the AP (i.e., EAP-SIM/EAP-AKA/ EAP-AKA', certificate, username/password).
- The UI shall indicate whether link layer security is in use.
- The UI may be capable of displaying an OSU Provider Icon, see section 4.8. If so, the following ANQP element shall be supported; if not, support for these ANQP elements is not required:
    - Icon Request

    - Icon Binary File
- When the UI is capable of displaying an OSU Provider Icon, support for the Icon type of image/png with an icon size up to 65535 octets is required. Note: this is the maximum size that can be transported in an ANQP message.

When a mobile device indicates support for HS2.0, it shall not use the following IEEE 802.11 security protocols:

- Temporal key integrity protocol (TKIP)
- Wired equivalent privacy (WEP)

2.3 Requirements for Hotspot Operators

Hotspot Operators shall, at a minimum, configure an HS2.0 hotspot as defined in this subsection. If any of these required capabilities is not configured on an AP, then the HS2.0 Indication shall not be included in Beacon and Probe Response frames:

- Roaming Consortium element, see [2]:
    - Registering for an OI is mandatory for large hotspot operators (e.g., national or regional operators) and optional for smaller operators (e.g., hotels). Hotspot Operators that have an OI shall include it in the Roaming Consortium element.
- The Country element's (see [2]) two-digit ISO 3166-1 country code shall be set to the value for the country in which the hotspot is located. The Country element shall be included in Beacon and Probe Response frames, unless prohibited by regulatory rules or the hotspot's location is unknown (e.g., the hotspot is on an airplane).
- The following ANQP-elements (see [2]):
    - Network Authentication Type
    - Roaming Consortium, for Hotspots that have roaming relationships with one or more SPs possessing a roaming consortium OI.
    - NAI Realm, for Hotspots that have roaming relationships with one or more SPs identified by their realms. Note: some SPs allow users to supply their own username and realm (e.g., email address for a social media account) making it impractical to configure all the necessary realms; SPs in this situation may find it convenient to be identified by their roaming consortium OI rather than by their realms.
    - 3GPP Cellular Network, for Hotspots having roaming relationships with cellular operators
    - Domain Name using FQDNs
- The following HS2.0 ANQP-elements (defined in section 4):
    - HS Query list
    - HS Capability list
    - Operator Friendly Name
    - NAI Home Realm Query.

Hotspot Operators should configure the following capabilities at HS2.0 compliant hotspots:

- The following ANQP-elements (see [2]):
    - Venue Name
    - IP Address Type Availability information
- The HS2.0 ANQP-elements (defined in section 4):
    - WAN Metrics (see section 4.4)
    - Connection Capability (see section 4.5)
    - Operating Class Indication (see section 4.7)
    - Hotspot Operators supporting online sign up shall configure the following ANQP-elements:
        - OSU Providers List
        - Icon Request
        - Icon Binary File

All Hotspot Operators shall acknowledge the credentials and EAP methods in Table 1 as a satisfactory security basis on which to establish roaming relationships with other Service Providers.

The HS2.0 compliant operator shall not employ features in their infrastructure that causes DNS Security Extensions (see [18], [19]) to be inoperable or that requires DNS Security Extensions to be disabled in a mobile device. The use of DNS redirection for Captive Portals is an example of such a behavior.

2.4 Requirements for Service Providers

Service Providers (SPs) shall support the following capabilities:

- An SP that does not have a SIM/USIM infrastructure shall support at least one of the following:
  - Username/password
  - Certificate credentials and their associated EAP method (see Table 1).
- An SP having SIM/USIM infrastructure shall support SIM/USIM credentials and their associated EAP methods and shall support at least one of the following:
  - Username/password
  - Certificate credentials and their associated EAP method (see Table 1).
- All SPs shall acknowledge the credentials and EAP methods in Table 1 as a satisfactory security basis on which to establish roaming relationships with other Service Providers (SPs) or Hotspot Operators.

SPs should support the following capabilities:

- For Home SPs, Online Sign Up (see section 8) and/or subscription provisioning using the SOAP XML (SPP) protocol. This includes support for the PerProviderSubscription MO (see section 9.1). In addition, SPs should support the registration process [52], as defined in Annex E.
- Deploy AAA servers that support the RADIUS attributes recommended in Annex D.

SPs may support the following capabilities:

- For Home SPs, the use of OMA DM protocol for Online Sign Up (see section 8) and/or subscription provisioning

3. Element and frame definitions

All reserved fields in these definitions have the value 0 on transmission and are ignored upon reception. In addition, little endian encoding is used for multi-byte fields and subfields.

3.1 Element definitions

3.1.1 HS2.0 Indication element

The HS2.0 Indication element enables the AP and the mobile device to indicate that they are HS2.0 capable and that they operate at the level of security required by this specification. This element uses the vendor-specific information element (see subclause 8.4.2.28 of [2]).

The HS2.0 AP shall include an HS2.0 Indication element in Beacon and Probe Response frames. The HS2.0 Indication element is ordered relative to other elements in the Beacon and Probe Response frames, as defined in Table 8-20 and Table 8-27, respectively, of [2]. Figure 1 shows the format of the HS2.0 Indication element.

When the mobile device supports HS2.0 capabilities, it shall include the HS2.0 Indication element in its (Re)Association Request frames. The ordering of this element in (Re)Association Request frame shall be in accordance with Table 8-22 and Table 8-24, respectively, of [2].

Implementations shall parse the HS2.0 Indication element according to the rules in subclause 9.24.8 of [2]. For this element, the Length field is defined below and is not drawn from Table 8-54 in [2].

	Element ID	Length	OI	Type	Hotspot Configuration	PPS MO ID (optional)	ANQP Domain ID (optional)
Octets:	1	1	3	1	1	2	2

Figure 1: HS2.0 Indication element format

The Element ID is a 1-octet field whose value is set to 221, the value for vendor specific information elements (see Table 7-26 in [2]).

The Length field is a 1-octet field whose value is set to 5 or 7. In HS2.0 Release 2 either the PPS MO ID field or the ANQP Domain ID field (these are mutually exclusive fields) is included in the HS2.0 Indication element. See the text below.

The OI is a 3-octet field and is defined in subclause 8.4.2.71.5 of [2]. As used by the Wi-Fi Alliance, the content of the OI field is set to the value 0x 50 6F 9A.

The Type field is a 1-octet field set to the value 0x10.

The format of the Hotspot Configuration field is shown in Figure 2.

	B0	B1	B2	B3	B4 B7
	DGAF Disabled	PPS MO ID Present	ANQP Domain ID Present	Reserved	Release Number
Bits:	1	1	1	1	4

Figure 2: Hotspot Configuration field format

The value of the Downstream Group-Addressed Forwarding (DGAF) Disabled bit is set to 1 when the AP is not forwarding downstream group-addressed frames (see section 5.2). Otherwise, it is set to 0.

When the HS2.0 Indication element is included in a (Re)Association Request frame transmitted from a mobile device the value of the DGAF Disabled bit is set to 0.

Note: a mobile device can use the DGAF Disabled subfield to alert a local IGMP client that multicast transmissions are unavailable or to indicate that the client will be accessed in a different manner (for instance, using DMS; see [2]).

The value of the PPS MO ID Present bit is set to 1 when the PPS MO ID field is present in the HS2.0 indication element; otherwise it is set to 0.

The value of the ANQP Domain ID Present subfield is set to 1 when the ANQP Domain ID field is present in the HS2.0 Indication element; otherwise it is set to 0.

The Release Number is a four bit subfield that identifies the HS2.0 release capability. The encoding of the Release Number value is shown in Table 2. When the HS2.0 Indication element is included in Beacon or Probe Response frames, it indicates the AP's HS2.0 capability. When the HS2.0 Indication element is included in a (Re)Association Request frame, it indicates the mobile device's HS2.0 capability; this capability will be used by the mobile device after successful association. Each AP and mobile device shall set the Release Number in accordance with its own capabilities, with the following exception: the mobile device shall not use a Release Number value in a (Re)Association Request frame greater than the Release Number value advertised by the AP to which the (Re)Association Request frame is sent.

Table 2: Release Number definition

Meaning	Value
Release 1	0
Release 2	1
Reserved	2 – 15

The PPS MO ID field is two octets long and contains the current version of the PerProviderSubscription MO that is provisioned to the mobile device. Figure 3 shows the format of the PPS MO ID field. When a mobile device transmits the HS2.0 Indication element in a (Re)Association Request frame, the device sets the value of the PPS MO ID field to the value of the PerProviderSubscription/UpdateIdentifier (see section 9.1). If the mobile device has not been provisioned with a PerProviderSubscription MO or if the PerProviderSubscription MO has become corrupted or reset to its default values, the value of the PPS MO ID field is 0. The AP does not include this field when it transmits Beacon or Probe Response frames that contain the HS2.0 Indication element.

A mobile device might have more than one PerProviderSubscription MO. In that case the mobile device uses the UpdateIdentifier value from the PerProviderSubscription MO instance, which contains the credential selected for use in the next EAP [16] authentication exchange.

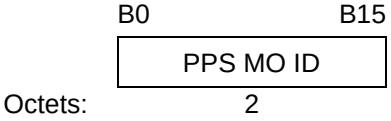


Figure 3: PPS MO ID field format

The ANQP Domain ID field is two octets long and is included when the HS2.0 Indication element is in Beacon and Probe Response frames transmitted by the AP. This field contains the AP's ANQP domain identifier. The ANQP Domain ID field is not included in the HS2.0 Indication element when this element is transmitted by the mobile device in (Re)Association Request frames. Figure 4 shows the format of the ANQP Domain ID field.

All APs in the same ESS that share a common nonzero value of ANQP Domain ID shall have identical ANQP information for the ANQP-elements and Hotspot 2.0 Vendor Specific ANQP elements in the following list:

- Venue Name ANQP-element
- Network Authentication Type ANQP-element
- Roaming Consortium ANQP-element
- IP Address Type Availability ANQP-element
- NAI Realm ANQP-element
- 3GPP Cellular Network ANQP-element
- Domain Name ANQP-element
- HS Capability list
- Operator Friendly Name
- WAN Metrics
- Connection Capability
- Operating Class Indication
- OSU Providers list
- Icon Binary File.

In the list above all of the elements with "ANQP-element" in their names are defined in [2]. The others are defined in this technical specification.

The AP whose ANQP domain identifier value is 0 either uses unique ANQP information in one or more of its ANQP-elements or Hotspot 2.0 vendor specific ANQP-elements, or has not been implemented with a means of determining whether its ANQP information is unique.

The mobile device should use the ANQP Domain ID field to limit the APs, to which it transmits ANQP queries, thereby optimizing the use of the medium.

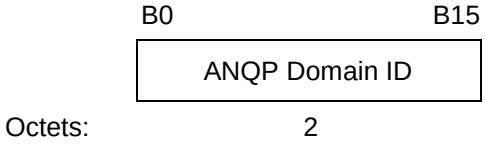


Figure 4: ANQP Domain ID field format

3.1.2 OSU Server-only authenticated layer 2 Encryption Network element

The OSU Server-only authenticated layer 2 Encryption Network (OSEN) element is used to advertise and select an OSEN capable network. The OSEN element has the same format as the RSNE except header fields, which are defined for the OSEN element below. All the remaining fields are the same as the RSNE payload and convey the same meaning.

	Element ID (221)	Length	OI	Type	Group Data Cipher Suite	Pairwise Cipher Count	Pairwise Cipher Suite List
Octets	1	1	3	1	4	2	4 x m

	AKM Suite Count	AKM Suite List	RSN Capabilities	PMKID Count	PMKID List	Group Management Cipher Suite
Octets	2	4 x n	2	2	S*16	4

Figure 5: OSEN element format

The Element ID is a 1-octet field whose value is set to 221, the value for vendor specific information elements (see Table 7-26 in [2]).

The Length is a 1-octet field whose value is set to sum of 4 plus the length of the contents of the OSEN element payload. Payload refers to all the subfields after Type field.

The Organizational Identifier (OI) is a 3-octet field and is defined in subclause 8.4.2.71.5 of [2]. The value of the OI field is set to 0x 50 6F 9A, as used by the Wi-Fi Alliance.

The Type is a 1-octet field set to the value 0x12.

The Group Data Cipher Suite field is four octets long and can contain any suite value allowed in an RSN. However, since DGAF is disabled for an OSEN, this value has no effect.

The Pairwise Cipher Suite Count field is two octets long and contains the number of cipher suites that are in the following Pairwise Cipher Suite List field in the OSEN element.

The Pairwise Cipher Suite List field contains information about any pairwise cipher suite allowed in an RSN.

The value of the AKM Suite Count field is the number of AKMs that are part of the AKM Suite List field.

The AKM Suite List field contains information about any AKM that is defined in section 3.1.3 to be an anonymous client AKM.

The RSN Capabilities field is two octets long and contains the same values as the RSN Capabilities field defined in subclause 8.4.2.27.4 of [2].

The PMKID Count and PMKID List fields contain the same values as the fields with those names in the RSN defined in subclause 8.4.2.27.5 of [2].

The Group Management Cipher Suite field is 4 octets long and can contain any value that is allowed in an RSN by this specification; however, because DGAF is disabled for an OSEN, this value has no effect.

3.1.3 WFA anonymous client 802.1X AKM

Table 3 describes the WFA anonymous client 802.1X AKM; extending the set of AKMs defined in IEEE 802.11-2012 subclause 8.4.2.27.3 of [2].

Table 3: Wi-Fi Alliance AKM values

OUI	Suite Type	Meaning		
		Authentication Type	Key Management Type	Key derivation Type
50-6F-9A	1	Anonymous client authentication negotiated over IEEE 802.1X with SHA256 key derivation, as defined in subclause 11.6 of [2]. Note: PMKSA caching is not permitted with this AKM.	RSNA key management with SHA256 key derivation, as defined in subclause 11.6 of [2].	Defined in subclause 11.6.1.7.2 of [2].

When a device uses the OSEN AKM, the integrity algorithm shall be AES-128-CMAC, the size of the MIC shall be 16, and the key wrap algorithm shall be the NIST AES key wrap.

3.2 Frame definitions

3.2.1 WNM-Notification Request frames

When the AP transmits a WNM-Notification frame containing one or more of the following subelements, it shall employ IEEE 802.11 protected management frame (PMF) procedures:

- Subscription Remediation subelement
- Deauthentication Imminent Notice subelement.

The mobile device may discard any unprotected WNM-Notification Request frame that it receives.

3.2.1.1 Subscription Remediation subelement

After the AAA server notifies (using a RADIUS Access-Accept message) an AP that subscription remediation is needed, the AP forwards that information to a mobile device in a Subscription Remediation subelement sent in a WNM-Notification Request frame.

When subscription remediation is needed, the AP transmits the WNM-Notification Request frame carrying the Subscription Remediation subelement immediately following successful completion of the 4-way handshake. Rules for using the WNM-Notification Request frame are specified in subclause 10.23.16 of [2].

The mobile device may have to wait until an IP address is obtained before contacting the host identified by the URL (if present) in the WNM-Notification Request frame.

Note: the WNM-Notification Response frame is not used in this case.

The AP notifies the client device using:

- The WNM-Notification Request frame (see subclauses 8.5.14.28 and 10.23.16 of [2])
- The value 1 in the WNM-Notification Request frame's Type field
- A subelement field with a vendor specific extension (WFA) and Subtype value for subscription remediation.

Figure 6 shows the format of the Subscription Remediation subelement.

	Subelement ID	Length	OI	Type	Server URL Length	Server URL (optional)	Server Method (optional)
Octets:	1	1	3	1	1	variable	1

Figure 6: Subscription Remediation subelement format

The Subelement ID is a 1-octet field whose value is equal to 221 (the value for vendor-specific sub elements (see Table 8-257 in [2])).

The Length is a 1-octet field whose value is set to sum of 6 plus the length of the contents of the Server URL field, if both the Server URL and Server Method fields are present. If both the Server URL field and the Server Method fields are not present, the value of the Length field is set to 5. Note: the Server URL field and the Server Method field are either both present or both absent.

The OI is a 3-octet field and is defined in subclause 8.4.2.28 of [2]. The value of the OI field is set to 0x 50 6F 9A, as used by the Wi-Fi Alliance.

The Type is a 1-octet field set to value 0x00.

The Server URL Length is a 1-octet field whose value is set to the length of the contents of the Server URL field. The value in the Server URL Length field is set to 0 if the Server URL field is not present.

The Server URL field is a variable length field that contains the URL of the subscription remediation server that the mobile device should contact for the purposes of remediating the user's subscription.

The Method is a 1-octet field and contains a value that identifies the protocol supported by the subscription server. Table 10 lists the set of available values. The mobile device uses this provisioning protocol when it uses the Server URL to identify the subscription server with which to initiate the remediation process.

3.2.1.2 Deauthentication Imminent Notice subelement

An AP transmits the Deauthentication Imminent Notice subelement in the WNM-Notification Request frame to a mobile device to indicate that the mobile device is about to be deauthenticated from the Wi-Fi AN. The frame also provides instructions on the time required to elapse before the AAA server will permit the mobile to be successfully reauthenticated on the same BSS or ESS. Optionally, the deauthentication reason, if provided by the AAA server, may be displayed by the mobile device on its user interface. The AAA server providing the reason shall provide a URL for a resource which supplies human-readable text in the human language of the user.

If deauthentication of the device is needed, the AP transmits a WNM-Notification Request frame carrying a Deauthentication Imminent Notice subelement following the successful completion of the 4-way handshake. Rules for using the WNM-Notification Request frame are specified in 10.23.16 of [2].

A Home SP uses the Deauthentication Imminent Notice to inform the mobile that, although EAP authentication succeeded, it was not or is no longer authorized for service at the time and/or location where the notice was received, or there was a temporary condition in the network which necessitated deauthentication (e.g., congestion in the Wi-Fi AN or congestion on a mobile core network element). The mobile device that receives this notice shall not attempt to reauthenticate to the same BSS or ESS until the expiration of the reauthentication delay.

The WNM-Notification Request action frame is used to indicate to the mobile device that deauthentication is needed. When deauthentication is needed, the AAA server notifies the AP via the RADIUS Access-Accept message or the Disconnect-Request message (see Annex D).

The AP notifies the mobile device using:

- The WNM-Notification Request frame (see [2], subclauses 8.5.14.28 and 10.23.16)
- The value 1 in the WNM-Notification Request frame's Type field
- A subelement field with a vendor specific extension (WFA) and Subtype value for deauthentication imminent notice

Since this message is received after successful completion of the four-way handshake, the credentials and security settings used by the mobile device to access the Wi-Fi network have already been properly provisioned and configured. Therefore, the mobile device should not interpret receipt of this message as an indication of a credential problem. The credentials might remain valid for use at other APs, as indicated by the Deauth Reason Code.

Figure 7 shows the format of the Deauthentication Imminent Notice subelement.

	Subelement ID	Length	OI	Type	De-Auth Reason Code	Re-Auth Delay	Reason URL Length	Reason URL (optional)
Octets:	1	1	3	1	1	2	1	variable

Figure 7: Deauthentication Imminent Notice subelement format

The Subelement ID is a 1-octet field whose value is equal to 221 (the value for vendor-specific subelements (see Table 8-257 in [2])).

The Length is a 1-octet field whose value is set to 8 plus the length of the contents of the Reason URL field.

The OI is a 3-octet field and is defined in subclause 8.4.2.28 of [2]. The value of the OI field is set to 0x 50 6F 9A, as used by the Wi-Fi Alliance.

The Type is a 1-octet field set to the value 0x01.

The De-Auth Reason Code is a 1-octet field whose value is selected from Table 4.

Table 4: De-Auth Reason Code definition

Meaning	Value
User's subscription does not allow or no longer allows access at this BSS or is temporarily not available at this BSS.	0
User's subscription does not allow or no longer allows access at this ESS or is temporarily not available at this ESS.	1
Reserved	2 to 255

The Re-Auth Delay is a 2-octet field whose value is set to the delay in seconds that a mobile device shall wait before attempting reassociation to the same BSS or ESS (as indicated in the De-Auth Reason Code field). A Re-Auth Delay field value of 0 means the delay value is chosen by the mobile device. The Re-Auth Delay starts when the WNM-Notification Request frame with a Deauthentication Imminent Notice subelement is successfully received by the mobile device.

The Reason URL Length is a 1-octet field whose value is set to the length of the contents of the Reason URL field. The value of the Reason URL Length field is set to 0 if the Reason URL field is not present.

The Reason URL field contains a URL, formatted in accordance with [15], which provides a webpage explaining why the mobile device was not authorized (or is no longer authorized) and will be deauthenticated from the BSS or ESS. It is implementation dependent whether the mobile device launches a browser to this URL to display the information on its user interface.

If the HS2.0 Deauthentication Request attribute (see section D.1.4) is received in RADIUS Access-Accept, the AP should transmit the WNM-Notification Request frame as soon as

practicable after the 4-way handshake has completed. In this case the AP does not cache the PMKSA.

The AP may allow sufficient time (for example, 1 minute) for the mobile device to retrieve the resource identified by the Reason URL field before deauthenticating the device.

4. Hotspot 2.0 ANQP-elements

The values of all reserved fields shall be 0 on transmission and ignored upon reception. In addition, little endian encoding is used for multi-octet fields and subfields.

When processing a received frame implementations shall ignore all unknown or reserved HS2.0 ANQP-elements (identified by the Subtype field, see Table 5) and shall parse the remaining frame body, when one exists, for additional ANQP-elements that have recognizable Info ID and/or Subtype values.

The Hotspot 2.0 (HS2.0) ANQP-elements provide additional functionality to the IEEE 802.11 ANQP-elements that support HS2.0 features. The HS2.0 ANQP-elements are formatted as defined by the ANQP vendor-specific element; see subclause 8.4.4.8 of [2], copied in Figure 8.

The Type, Subtype, Reserved and Payload fields comprise the vendor specific content, as shown in Figure 8-414 of [2].

	Info ID	Length	OI	Type	Subtype	Reserved	Payload
Octets:	2	2	3	1	1	1	variable

Figure 8: HS2.0 ANQP-element format

The Info ID is a 2-octet field whose value is set to 56797 (the value for the ANQP vendor-specific element) (see Table 8-184 in [2]).

The Length is a 2-octet field whose value is set to 6 plus the length of the contents of the Payload field.

The OI is a 3-octet field and is defined in subclause 8.4.2.71.5 of [2]. The OI field is set to the value 0x 50 6F 9A, as used by the Wi-Fi Alliance.

The Type is a 1-octet field set to the value 0x11.

The Subtype is a 1-octet field whose value identifies the HS2.0 ANQP-element. The possible Subtype field values are defined in Table 5.

The Reserved is a 1-octet field and is included in the ANQP-element in order to word align the ANQP-element's header.

The Payload field has variable length and contains information specific to the HS2.0 ANQP-element, as defined in sections 4.1 to 4.10.

Table 5: HS2.0 ANQP-element Subtype definition

ANQP-element Name	Subtype Value	Description (section)	Extensible
Reserved	0	n/a	
HS Query list	1	4.1	
HS Capability list	2	4.2	
Operator Friendly Name	3	4.3	
WAN Metrics	4	4.4	
Connection Capability	5	4.5	
NAI Home Realm Query	6	4.6	
Operating Class Indication	7	4.7	
OSU Providers list	8	4.8	Yes
Reserved	9		
Icon Request	10	4.9	
Icon Binary File	11	4.10	
Reserved	12-255	n/a	

A “Yes” in the Extensible column of an HS2.0 ANQP-element listed in Table 5 indicates that the ANQP-element might be extended in future revisions of this specification. When parsing an extensible HS2.0 ANQP-element, an implementation shall discard any part of the element beyond the fields defined in this release of the specification and shall otherwise process the HS2.0 ANQP-element as though this truncated ANQP-element had been received.

The usage of the HS2.0 ANQP-elements is shown in Table 6.

Table 6: HS2.0 ANQP-element usage

ANQP-element Name	ANQP-Element (subclause)	ANQP-element type	AP	Mobile Device
HS Query List	4.1	Q	R	T
HS Capability List	4.2	S	T	R
Operator Friendly Name	4.3	S	T	R
WAN Metrics	4.4	S	T	R
Connection Capability	4.5	S	T	R
NAI Home Realm Query	4.6	Q	R	T
Operating Class Indication	4.7	S	T	R
OSU Providers List	4.8	S	T	R
Icon Request	4.9	Q	R	T
Icon Binary File	4.10	S	T	R
Symbols Q element is an ANQP Query S element is an ANQP Response T ANQP-element may be transmitted by MAC entity R ANQP-element may be received by MAC entity				

4.1 HS Query List element

The HS Query list provides a list of identifiers of HS2.0 ANQP-elements for which the requesting mobile device is querying in an HS ANQP query. The HS Query List ANQP-element is included in a GAS Query Request frame. The HS Query List ANQP-element shall be used in a GAS Query Request to request HS2.0 ANQP-elements. Both the ANQP Query List and the HS2.0 Query List may be included in a single GAS Query Request (see the example query in Annex B).

The format of the HS Query List payload is provided in Figure 9.

	HS2.0 Element Subtype #1	HS2.0 Element Subtype #2 (optional)	…	HS2.0 Element Subtype #n (optional)
Octets:	1	0 or 1	…	0 or 1

Figure 9: HS Query List ANQP-element payload format

The value of each HS2.0 element Subtype field is a Subtype drawn from Table 5. A Subtype is included in the HS Query List element to indicate that the mobile device performing the GAS Query Request is requesting that the HS2.0 element corresponding to that Subtype be returned in the GAS Query Response. The Subtypes included in the HS Query List element shall be ordered by monotonically increasing Subtype value.

4.2 HS Capability List element

The HS Capability List element provides a list of information/capabilities that have been configured on an AP. The HS Capability List element is returned in response to a GAS Query Request. When a mobile device discovers an HS2.0 AP, the mobile device may assume that mandatory HS2.0 ANQP-elements are supported by the HS2.0 AP. Support for this HS ANQP-element is mandatory (see section 2), but its usage is optional.

The format of the HS Capability List payload is provided in Figure 10.

	HS Capability #1	HS Capability #2 (optional)	…	HS Capability #n (optional)
Octets:	1	0 or 1	…	0 or 1

Figure 10: HS Capability List payload format

Each HS Capability field value is a Subtype drawn from Table 5. If included in the HS Capability list response, it indicates that a query request for that Subtype will return the requested HS element. The Subtype for HS Capability list is always included in the HS Capability list returned in a GAS Query Response. The list includes no duplicate Subtypes. The Subtypes returned in the HS Capability list are ordered by increasing Subtype value.

4.3 Operator Friendly Name element

The Operator Friendly Name element provides zero or more operator names who are operating the IEEE 802.11 AN i.e., the Hotspot Operator. The format of the Operator Friendly Name element Payload is shown in Figure 11. If more than one Operator Name Duple field is included, the fields shall represent the same operator name in different human languages.

	Operator Name Duple #1 (optional)	Operator Name Duple #2 (optional)	…	Operator Name Duple #n (optional)
Octets:	variable	variable	…	variable

Figure 11: Operator Friendly Name element payload format

The format of the Operator Name Duple field is shown in Figure 12.

	Length	Language Code	Operator Name
Octets:	1	3	variable

Figure 12: Operator Name Duple field

The Length is a 1-octet field whose value is equal to sum of 3 plus the number of octets in the Operator Name field.

The Language Code is a 3-octet ISO-14962-1997 encoded string field that defines the language used in the Operator Name field. The Language Code field value is a two or three character language code selected from ISO-639 [55]. A two character language code value has 0 (“null” in ISO-14962-1997) appended to make it 3 octets long.

The Operator Name is a variable length UTF-8 formatted field containing the operator's name. The maximum length of this field is 252 octets. UTF-8 format is defined in IETF RFC 3629 [14].

Note: the format of the Operator Name Duple field is identical to the Venue Name Duple field definition shown in Figure 8-407 in [2], except that the Operator Name replaces the Venue Name in the Venue Name subfield.

4.4 WAN Metrics element

The WAN Metrics element provides information about the WAN link connecting an IEEE 802.11 AN and the Internet. Transmission characteristics such as the speed of the WAN connection to the Internet are included. The format of the WAN Metrics element Payload is shown in Figure 13.

	WAN Info	Downlink Speed	Uplink Speed	Downlink Load	Uplink Load	LMD
Octets:	1	4	4	1	1	2

Figure 13: WAN Metrics element payload format

The format of the WAN Info field is shown in Figure 14.

Bits:	B0 B1	B2 B3	B4 B7
Link Status	Symmetric Link	At Capacity	Reserved

Figure 14: WAN Info field format

The Link Status is 2 bits long field and is set to reflect the status of the WAN link. The value of the Link Status subfield shall be one of the non-reserved values in Table 7.

Table 7: Link Status definition

Meaning	Value
Reserved	0
Link up	1
Link down	2
Link in test state	3

The value of the Symmetric Link bit is set to 1 if the WAN link has the same link speed in the uplink and downlink direction and is set to 0 if the uplink speed is different from the downlink speed.

The value of the At Capacity bit is set to 1 if the WAN link is currently operating at its maximum capacity and no additional mobile devices will be allowed to associate to the AP; otherwise it is set to 0.

The value of the Reserved field is set to 0 in transmitted frames and is ignored in received frames.

The Downlink Speed is a 4-octet positive integer whose value is an estimate of the WAN Backhaul link current downlink speed in kilobits per second. For backhaul links that do not vary in

speed or those for which no accurate estimation can be made, this attribute contains the nominal speed. The maximum value reported by this field is 4,294,967,296 kbps (approximately 4.2Tbit/s); if the backhaul downlink speed is greater than this value, the maximum value is reported. The downlink speed value is set to zero when the downlink speed is unknown.

The Uplink Speed is a 4-octet positive integer whose value is an estimate of the WAN Backhaul link's current uplink speed in kilobits per second. For backhaul links that do not vary in speed or those for which no accurate estimation can be made, this field contains the nominal speed. The maximum value reported in this field is 4,294,967,296 kbps (approximately 4.2Tbit/s); if the backhaul uplink speed is greater than this value, the maximum value is reported. The uplink speed value is set to zero when the uplink speed is unknown.

The Downlink Load is a 1-octet positive integer representing the current percentage loading of the downlink WAN connection, scaled linearly with 255 representing 100%, as measured over an interval the duration of which is reported in Load Measurement Duration. In cases where the downlink load is unknown to the AP, the value is set to zero. The algorithm to calculate downlink load is implementation dependent.

The Uplink Load is a 1-octet positive integer representing the current percentage loading of the uplink WAN connection, scaled linearly with 255 representing 100%, as measured over an interval, the duration of which is reported in Load Measurement Duration. In cases where the uplink load is unknown to the AP, the value is set to zero. The algorithm to calculate uplink load is implementation dependent.

The LMD (Load Measurement Duration) field is a 2-octet positive integer representing the duration over which the Downlink Load and Uplink Load have been measured, in tenths of a second. When the actual load measurement duration is greater than the maximum value, the maximum value will be reported. The value of the LMD field is set to 0 when neither the uplink nor downlink load can be computed. When the uplink and downlink loads are computed over different intervals, the maximum interval is reported.

4.5 Connection Capability element

The Connection Capability element provides information on the connection status in the hotspot of the most commonly used communications protocols and ports. For example, a firewall upstream to the AN may allow communication on certain IP protocols and ports, while blocking communication on others. Figure 15 shows the format of the Connection Capability element payload.

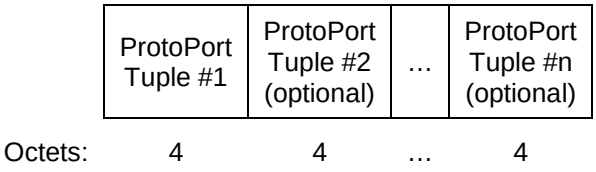


Figure 15: Connection Capability element payload format

Figure 16 shows the format of the ProtoPort Tuple field.

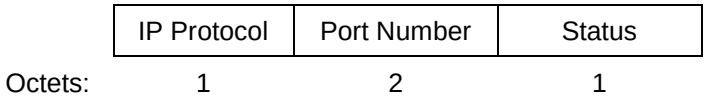


Figure 16: ProtoPort Tuple format

The IP Protocol is a 1-octet subfield whose value identifies the IP protocol. The IP Protocol subfield refers to the IP protocol field in IPv4 packets [3]; the IP protocol subfield refers to the next header field in IPv6 packets [6].

The Port Number is a 2-octet long subfield and contains the destination port number used in conjunction with the protocol defined by the IP Protocol subfield. When port numbers are not used in conjunction with an IP Protocol, the Port Number subfield is reserved.

The Status is a 1-octet long subfield. Table 8 lists its possible values.

Table 8: Status subfield values

Status value	Meaning
0	Closed
1	Open
2	Unknown
3-255	Reserved

A Status subfield value of 0 means that either the IP protocol or the associated port number is not open for communication (for instance, communication is blocked by a firewall or a network address translation (NAT) function) in the AN.

A Status subfield value of 1 means that the IP protocol or the associated port number is open for communication (i.e., communication is not blocked by a firewall or NAT function) in the AN.

A Status subfield value of 2 means that the IP protocol or the associated port number may or may not be open for communication in the AN − i.e., the exact status is not known.

The ProtoPort tuples listed in Table 9 are included by the AP in its response to a corresponding ANQP query.

Table 9: ProtoPort tuples

IP Protocol Value	Port Number Value	Description
1	0	ICMP, used for diagnostics
6	20	FTP
6	22	SSH
6	80	HTTP
6	443	Used by HTTPS and TLS VPNs
6	1723	Used by Point to Point Tunneling Protocol VPNs
6	5060	VoIP
17	500	Used by IKEv2 (IPsec VPN)
17	5060	VoIP
17	4500	May be used by IKEv2 (IPsec VPN)
50	0	ESP, used by IPsec VPNs

Additional ProtoPort tuples may be included by the AP.

Note: A Status value of 1 for the ProtoPort tuple (6, 80) indicates HTTP access to the network.

Example: To set up IPsec based VPNs (with or without UDP encapsulation), the Status values of the tuples (17, 500), (17, 4500) and (50, 0) are all set to 1.

4.6 NAI Home Realm Query element

The NAI Home Realm Query element is used by a requesting mobile device to determine whether the network access identifier (NAI) realms for which it holds security credentials are realms corresponding to SPs or other entities whose networks or services are accessible via this BSS. The requesting mobile device includes in each NAI Home Realm Query element that it transmits only the NAI Home Realm Name(s) for which it holds credentials.

In response to a received NAI Home Realm Query element, a responding AP returns a NAI Realm ANQP-element formatted as specified in subclause 8.4.4.10 of [2]. The AP includes in its transmitted NAI Realm ANQP-element only information on the realms that exactly match the realms contained in the received NAI Home Realm Query element.

The transmitted NAI Realm ANQP-element (see Figure 8-417 of [2]) may contain one or more NAI Realm Data fields each containing information about one or more matching realms (see Figure 8-418 of [2]).

If the responding AP has no matching NAI Realm subfields, then the NAI Realm ANQP-element is returned with the NAI Realm Count set to zero.

A mobile device may include the NAI Home Realm Query element as the sole ANQP-element in a GAS Initial Request frame that it transmits.

The format of the NAI Home Realm Query element payload is shown in Figure 17.

NAI Home Realm Count	NAI Home Realm Name Data #1	NAI Home Realm Name Data #2 (optional)	…	NAI Home Realm Name Data #n (optional)
Octets: 1	variable	variable		variable

Figure 17: NAI Home Realm Query element payload format

The NAI Home Realm Count is a 1-octet field that specifies the number of NAI Home Realm Name Data fields included in the NAI Home Realm Query element.

The format of the NAI Home Realm Name Data field is shown in Figure 18.

NAI Realm Encoding	NAI Home Realm Name Length	NAI Home Realm Name
Octets: 1	1	Variable

Figure 18: NAI Home Realm Name Data format

The NAI Realm Encoding is a 1-octet subfield whose format is shown in Figure 8-419 of [2] and specified in subclause 8.4.4.10 of [2].

The NAI Home Realm Name Length is a 1-octet subfield whose value is the length of the contents of the NAI Realm Name subfield.

The NAI Home Realm Name subfield contains information on one or more NAI home realms, formatted as defined for the NAI Realm subfield of the NAI Realm Data field specified in subclause 8.4.4.10 of [2]. The maximum length of this subfield is 255 octets.

The mobile device implementation should be designed to accommodate the fact that home realm information is transmitted in this element without confidentiality protection. The mobile device is not required to use the NAI Home Realm Query element.

4.7 Operating Class Indication element

The Operating Class Indication element provides information about the groups of channels in the frequency band(s) that the Wi-Fi AN is using. This element reports about the operating classes of APs that are in the same ESS as the AP transmitting this element. A mobile device that supports more than one frequency band (e.g., both 2.4GHz and 5GHz) may use this element for BSS selection purposes. The format of the Operating Class Indication element is shown in Figure 19.

Operating Class Indication #1 (optional)	…	Operating Class Indication #n (optional)
Octets: 1	…	1

Figure 19: Operating Class Indication element payload format

The Operating Class Indication field indicates the operating class (as defined in Annex E of [2]), for the mobile device to operate in after association. The value of the Operating Class Indication field is set to one of the operating class number values defined in Table E-4 of [2].

4.8 OSU Providers List element

The OSU Providers List element provides information about one or more entities that offer online sign up service. Figure 20 shows the format of the OSU Providers List element. For each OSU provider this element includes the following information: the friendly name (in one or more human languages), the NAI used to authenticate to the OSU ESS (if configured for OSEN), the Icon(s), and the URI of the OSU server.

At least one OSU Provider subfield is available if online sign up is supported. This is indicated by the Network Authentication Type Indicator holding the value of “On-line enrollment supported” in the Network Authentication Type ANQP-element (see Table 8-185 in [2]).

	OSU SSID Length	OSU SSID	Number of OSU Providers
Octets:	1	variable	1

	OSU Provider #1	…	OSU Provider #n (optional)
Octets:	variable	…	variable

Figure 20: OSU Providers list element payload format

The OSU SSID Length is a 1-octet subfield whose value is the length of the contents of the OSU SSID subfield.

The OSU SSID is a variable length subfield that contains the SSID that the mobile device uses for online sign up with all the OSU providers listed in this element. This subfield shall be formatted in accordance with subclause 8.4.2.2 of [2].

The Number of OSU Providers is a 1-octet subfield whose value is set to the number of OSU provider subfields present.

4.8.1 OSU Provider subfield

The format of the OSU Provider subfield is shown in Figure 21.

	OSU Provider Length	OSU Friendly Name Length	OSU Friendly Name Duple #1	…	OSU Friendly Name Duple #n	OSU Server URI Length	OSU Server URI	OSU Method list Length	OSU Method list
Octets:	2	2	variable		variable	1	variable		1 variable

	Icons Available Length	Icons Available	OSU_NAI Length	OSU_NAI (optional)	OSU Service Description Length	OSU Service Description Duple #1 (optional)	…	OSU Service Description Duple #n (optional)
Octets:	2	variable	1	variable	2	variable		variable

Figure 21: OSU Provider subfield format

The OSU Provider Length is a 2-octet subfield whose value is set to sum of 9 plus the sum of the lengths of the following subfields: OSU Friendly Name Duple(s), OSU Server URI, OSU Method List, Icons Available, OSU_NAI and OSU Service Description Duple(s).

4.8.1.1 OSU Friendly Name

The OSU Friendly Name Length is a 2-octet subfield whose value is set to the sum of the lengths of the OSU Friendly Name Duple(s).

Figure 22 shows the format of the OSU Friendly Name Duple field:

	Length	Language Code	OSU Friendly Name
Octets:	1	3	variable

Figure 22: OSU Friendly Name Duple field format

The Length is a 1-octet field whose value is equal to sum of 3 plus the number of octets in the contents of the OSU Friendly Name field.

The Language Code is a 3-octet ISO-14962-1997 encoded string field that defines the language used in the OSU Friendly Name field. The Language Code field value is a two or three character language code selected from ISO-639 [55]. A two character language code value has 0 (“null” in ISO-14962-1997) appended to make the value 3 octets long.

The OSU Friendly Name is a variable length UTF-8 formatted field that contains the friendly name of the OSU provider in the human language identified by the value of the Language Code field. Each friendly name exactly matches the same human-language friendly name drawn from the corresponding OSU server certificate. The mobile device selects the human language to display to the user. The maximum length of this field is 252 octets. The UTF-8 format is defined in IETF RFC 3629 [14].

4.8.1.2 OSU Server URI

The OSU Server URI Length is a 1-octet subfield whose value is set to the length of the contents of the OSU Server URI subfield.

The OSU Server URI is the URI of the OSU server that is used for OSU with the SP indicated in the Friendly Name subfield. The contents of the OSU Server URI subfield are formatted in accordance with [17].

4.8.1.3 OSU Method list

The OSU Method List Length is a 1-octet subfield whose value is set to the length of the contents of the OSU Method List subfield.

The OSU Method List subfield is a list of 1-octet integers identifying encoded OSU methods (see Table 10). These methods are listed in SP preferred order with the most-preferred first. The list contains the protocols supported by the OSU provider. The mobile device shall choose the highest preference protocol that it can support.

Table 10: OSU Method values

Meaning	Value
OMA DM (see section 8.3)	0
SOAP XML SPP (see section 8.4)	1
Reserved	2 - 255

4.8.1.4 Icons Available

The Icons Available Length is a 2-octet subfield whose value is set to the sum of the lengths of the contents of the Icon Metadata subfields.

The Icons Available subfield provides metadata about the OSU provider icon file(s) available for download. The Icons Available subfield provides metadata for zero or more OSU Provider icons; if more than one Icon Metadata subfield is present, each icon so represented provides the same image in a different image size. A mobile device uses the Icon Request HS2.0 ANQP-element to retrieve the icon of the desired image size. The Icons Available subfield is formatted in accordance with Figure 23. It is not a requirement on an OSU provider to have icons of different image size; if only one image size is provided, it is the responsibility of the mobile device to scale the icon as needed for display purposes.

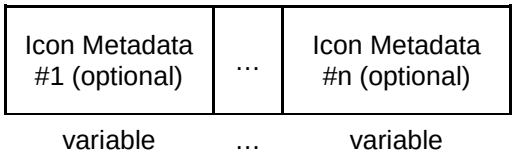


Figure 23: Icons Available subfield format

The Icon Metadata subfield is formatted in accordance with Figure 24.

	Icon Width	Icon Height	Language code	Icon Type Length	Icon Type	Icon Filename Length	Icon Filename
Octets:	2	2	3	1	variable	1	variable

Figure 24: Icon Metadata subfield format

The Icon Width is a 2-octet subfield whose value is the width in pixels of the OSU Provider icon named by the Icon Filename subfield.

The Icon Height is a 2-octet subfield whose value is the height in pixels of the OSU Provider icon named by the Icon Filename subfield.

The Language Code is a 3-octet ISO-14962-1997 encoded string field that defines the language used in the Icon Metadata field. The value of the Language Code field is a two or three character language code selected from ISO-639 [55]. A two character language code value has 0 (“null” in ISO-14962-1997) appended to make it 3 octets long. Note: Per ISO-639, if there is no linguistic content to the logo, the Language Code is set to “zxx”.

The Icon Type Length is a 1-octet subfield whose value is the length of the contents of the Icon Type subfield.

The Icon Type is a variable length field that contains the MIME media type of the binary icon file named by the Icon Filename subfield. The Icon Type subfield is formatted in accordance with RFC 4288 [23] and its value is selected from the IANA MIME media types registered at http://www.iana.org/assignments/media-types/index.html.

The mobile device shall support an Icon Type of image/png.

The Icon Filename Length is a 1-octet subfield whose value is the length of the contents of the Icon Filename subfield.

The Icon Filename subfield is a UTF-8 encoded subfield whose value contains the filename of the Icon that has the metadata provided in the matching Icon Metadata subfield.

4.8.1.5 OSU_NAI

The OSU_NAI Length is a 1-octet subfield whose value is set to the length of the contents of the OSU_NAI subfield. If the OSU_NAI subfield is not present, the value of the OSU_NAI Length subfield shall be 0.

The value of the OSU_NAI subfield is formatted in accordance with [22] and contains the NAI that is used for OSU with the SP indicated in the Friendly Name subfield. When the OSU_NAI subfield is present, the OSU ESS employs link-layer encryption (see sections 5.4.2 and 7.7). When the OSU_NAI subfield is not present, the OSU ESS is open (see section 5.4.1).

4.8.1.6 OSU Service Description

The OSU Service Description Length is a 2-octet subfield whose value is set to the sum of the OSU Service Description Duple(s).

Figure 25 shows the format of the OSU Service Description Duple field.

	Length	Language Code	OSU Service Description
Octets:	1	3	variable

Figure 25: OSU Service Description Duple field format

The Length is a 1-octet field whose value is equal to sum of 3 plus the number of octets in the OSU Service Description field.

The Language Code is a 3-octet ISO-14962-1997 encoded string field that defines the language used in the OSU Service Description field. The Language Code field value is a two or three character language code selected from ISO-639 [55]. Each two character language code value has 0 ("null" in ISO-14962-1997) appended to make it 3 octets long.

The OSU Service Description subfield is a variable length UTF-8 formatted field that contains the SP's description of the service offering. The maximum length of this field is 252 octets. UTF-8 format is defined in IETF RFC 3629 [14].

4.9 Icon Request element

The Icon Request element provides a filename for which a mobile device is requesting download. The Icon Filename is one of the filenames included in the OSU Providers List element. The format of the Icon Request element payload is provided in Figure 26 shows the format of the Icon Request element payload.

	Icon Filename
Octets:	variable

Figure 26: Icon Request element payload format

The Icon Filename subfield contains a value taken from an Icon Filename subfield present in the OSU Providers list element.

4.10 Icon Binary File element

The Icon Binary File element contains the binary contents of an OSU provider icon. The Icon Binary File HS2.0 ANQP-element is provided in response to an Icon Request element. The format of the Icon Binary File element payload is provided in Figure 27.

	Download Status Code	Icon Type Length	Icon Type	Icon Binary Data Length	Icon Binary Data
Octets:	1	1	variable	2	variable

Figure 27: Icon Binary File element payload format

The Download Status Code is a 1-octet field whose value is defined in Table 11. The Icon Type field and the Icon Binary Data field are only provided if the Download Status Code is equal to 0 (i.e., Success).

The Icon Type Length is a 1-octet subfield whose value is the length of the contents of the Icon Type subfield.

The Icon Type subfield is a variable length field that contains the MIME media type of the binary icon file named by the Icon Filename subfield included in the Icon Request HS2.0 ANQP-element. The Icon Type subfield is formatted in accordance with RFC 4288 [23] and its value is selected from the IANA MIME media types registered at http://www.iana.org/assignments/media-types/index.html.

The Icon Binary Data Length is a 2-octet subfield whose value is equal to the length of the contents of the Icon Binary Data subfield.

The Icon Binary Data field contains the binary data for the icon encoded per the Icon Type field. The SHA-256 hash of the Binary Data field exactly matches the SHA-256 hash for the same-named icon drawn from the corresponding OSU server certificate.

Table 11: Download Status Code field definition

Meaning	Value
Success	0
File not found	1
Unspecified file error	2
Reserved	3-255

5. Hotspot procedures and protocols

5.1 Layer 2 traffic inspection and filtering

Layer 2 inspection and filtering prevents frames exchanged between two mobile devices from being delivered by the Wi-Fi AN without first being inspected and filtered in either the hotspot operator network or the SP core network. Such processing provides the mobile device some protection against attack.

To support layer 2 traffic inspection and filtering:

- An HS2.0 AP shall not directly deliver frames received from a mobile device associated to its BSS to another mobile device associated to its BSS.
- An HS2.0 AP that has an embedded inspection and filtering function shall deliver frames received from mobile devices associated with its BSS destined to other mobile devices in the BSS or ESS after passing those frames through its inspection and filtering function.
- An HS2.0 AP that does not have an embedded inspection and filtering function shall deliver all frames received from mobile devices associated with its BSS to its embedded portal (e.g., Ethernet interface; see clause 5.2.5 of [2]).
- The Hotspot Operator's or service provider's network should provide a function that inspects and filters traffic delivered by the AP, which is exchanged between two mobile devices on the hotspot network. In many cases the inspection and filtering function will operate on traffic exchanged between two mobile devices on the same IP subnet (sometimes referred to as a transparent firewall) and on different IP subnets (the classical routed firewall function). The inspection and filtering function limits specific traffic types, such as ARP and DHCP messages, between peers to prevent malicious behaviors.
- An HS2.0 AP shall prohibit the initiation of DLS link establishment using the procedures defined in subclause 10.7 of [2].
- An HS2.0 AP shall prevent TDLS link establishment using the procedures defined in subclause 10.22 of [2].
  Note: HS2.0 APs do not support P2P operation and WPS because WPA2-Enterprise is required; thus operation using WPA2-Personal is not possible.

5.2 Downstream forwarding of group-addressed frames by the AP

The HS2.0 AP shall support the DGAF Disable feature to mitigate attacks exploiting the GTK (see clause 8 in [2]).

When the DGAF Disable bit subfield in the HS2.0 Indication element is set to 1 in frames transmitted by an HS2.0 AP, the AP:

- Shall set to a unique random value the value of the GTK employed in the 4-Way Handshake with each mobile device that becomes associated with the AP. The GTK used for each mobile device shall be different from every GTK used for the other mobile devices associated to the BSS.
- Shall not forward any group-addressed frames (i.e., multicast or broadcast frames) to any mobile device associated to the BSS.
- Shall convert all group-addressed DHCP (IP) packets received from its embedded portal (see subclause 5.2.5 of [2]) to individually-addressed IEEE 802.11 frames transmitted to the requesting mobile device. Note: this is a layer-2 multicast to unicast conversion and during this process the MSDU is not altered.
- Shall enable Proxy ARP service.
- Shall convert ICMPv6 Router Advertisement packets received from its embedded portal (see subclause 5.2.5 of [2]) to individually-addressed IEEE 802.11 frames transmitted to

associated mobile devices. This is a layer-2 multicast to unicast conversion and during this process the MSDU is not altered.

When the DGAF Disable bit subfield in the HS2.0 Indication element is set to 0 in frames transmitted by an HS2.0 AP, the AP:

- Shall forward all group addressed frames using a single GTK for all associated devices in the BSS.
- Shall not forward gratuitous ARP messages into the BSS.

  Note that the AP may convert a received gratuitous ARP request message to an individually-addressed IEEE 802.11 frame and transmit it to associated mobile devices (see the specification of gratuitous ARP in section 5.3).

- Shall not forward unsolicited IPv6 Neighbor Advertisement packets into the BSS.

  Note that the AP may convert a received unsolicited Neighbor Advertisement to an individually-addressed IEEE 802.11 frame and transmit it to associated mobile devices.

- May enable Proxy ARP service

Note: the purpose of the DGAF Disable feature is to mitigate the so called “hole-196” attack. By IEEE 802.11 design all mobile devices in a BSS use the same GTK, so forgery of group-addressed frames is always possible. However, in some hotspots multicast service using group-addressed frames is needed. In these cases, the DGAF Disable bit would be set to 0.

5.3 Proxy ARP service

An HS2.0 AP shall support the Proxy ARP service defined in subclause 10.23.13 of [2]. The specification for Proxy ARP service is in [2]. This specification defines additional requirements for duplicate address detection (IPv6), address conflict resolution (IPv4), and forwarding of gratuitous ARP messages and unsolicited neighbor advertisement packets.

The following requirements apply to the HS2.0 AP:

- When an IPv4 address in an ARP request packet, which is being checked for duplicate usage (i.e., address conflict detection, see [26]), is being used by a mobile device currently associated to the BSS, the HS2.0 AP's Proxy ARP service shall respond to the request on behalf of the mobile device that is already using that IP address. The HS2.0 AP shall insert the associated mobile device's MAC address as the sender’s MAC address in its response the ARP Response packet. In this case the AP shall not update its Hardware Address to Internet Address mapping due to the ARP request packet.
- The HS2.0 AP may forward a gratuitous ARP message into the BSS by converting its destination MAC address to an individual MAC address. The method used by an AP to choose whether to forward the gratuitous ARP message is outside the scope of this specification.
  Note 1: this is layer-2 multicast to unicast conversion and during this process the MSDU is not altered.
  Note 2: gratuitous ARP messages can be transmitted by an attacker for the purposes of ARP cache poisoning.
- When an IPv6 address in the Target Address field of a Neighbor Solicitation packet, sent from the unspecified source IP address, is being used by another mobile device currently associated to the BSS, the HS2.0 AP's Proxy ARP service shall respond to the request on behalf of the mobile device that is already using that IP address, as described in [2]. The HS2.0 AP shall insert the associated mobile device's MAC address as the sender’s MAC address in the Neighbor Advertisement packet. In this case, the AP shall not update its Hardware Address to Internet Address mapping due to the Neighbor Solicitation packet.
- The HS2.0 AP may forward an unsolicited Neighbor Advertisement message into the BSS by converting its destination MAC address to an individual MAC address. The

method used by an AP to choose whether to forward an unsolicited Neighbor Advertisement message is outside the scope of this specification.
Note 1: this is layer-2 multicast to unicast conversion and during this process the MSDU is not altered.
Note 2: unsolicited Neighbor Advertisement messages can be transmitted by an attacker for the purposes of ARP cache poisoning.

5.4 SSID configuration procedures for hotspots offering online sign up

Hotspots offering online sign up shall employ the SSID configuration procedures specified in this subsection.

Hotspots offering online sign up shall provision two ESSs:

- An OSU ESS that supports online sign up.
- A production ESS that provides network access to the authenticated mobile device.

The mobile device joins the production ESS using the procedures defined in section 6.4.

A production BSS (i.e., a member of the production ESS) and its corresponding OSU BSS (i.e., a member of the OSU ESS) may be members of a common Multiple BSSID Set; if so, the Multiple BSSID Set shall be compliant with the Multiple BSSID Set procedures defined in subclause 10.11.14 of [2].

The production ESS and OSU ESS SSID values are configured by the Hotspot Operator; the Hotspot Operator shall configure names compliant with SSID naming rules in subclause 8.4.2.2 of [2].

5.4.1 Open OSU ESS

Many hotspot operators have existing hotspot deployments that employ an open SSID and captive portal for authentication. These hotspot operators will typically add a second SSID to their hotspots for the Hotspot 2.0 network. For the purposes of online sign up, hotspots having an open SSID may use that SSID as the OSU SSID. The OSU Provider field in the OSU Providers list element contains the open OSU SSID (see section 4.8).

5.4.2 OSEN OSU ESS

Hotspot 2.0 defines an OSU Server-only authenticated layer 2 Encryption Network (OSEN). An OSEN is used only for OSU access. In an OSEN the network authentication service is authenticated and any STA is allowed to join. (In contrast, an RSN requires that both the client STA and network authentication service are mutually authenticated and only enrolled STAs are allowed to join.) An OSEN uses the WFA anonymous client 802.1X AKM defined in section 3.1.3.

An OSEN does not assume that clients are enrolled before establishing an OSEN Association (OSENA). It does assume that the server possesses credentials that can be used to authenticate it to the client. Shared symmetric credentials such as passwords or PSKs shall not be used with an OSENA. The OSEN uses only AKMs that provide the ability for the client to authenticate the server without previous enrollment and that prevent clients that can authenticate the server from spoofing the server's identity (no shared symmetric credentials). Currently, the anonymous client 802.1X AKM defined in section 3.1.3 is the only AKM defined for an OSEN.

In the 4-way handshake used for OSEN the value of the Key Descriptor Version shall be set to 0 (see clause 11.6.2 and Figure 11-29 in IEEE 802.11-2012).

Hotspot operators deploying an OSEN use the procedures defined in this subsection. The HS2.0 Indication element shall not be included in the Beacon and Probe Response frames transmitted by APs that are members of the OSEN.

Hotspot operators that deploy an OSEN shall configure those APs according to the following procedures:

- DGAF shall be disabled (see Figure 2). The use of the GTK is disallowed in an OSENA. In an OSENA, the handling of the broadcast frames and GTK is as defined in section 5.2 and section 6.5 with the DGAF bit set to 1.
- Interworking capability shall be disabled (i.e., dot11InterworkingServiceActivated shall be set to false, see subclause 10.24.2 of [2]).
- OSEN and RSN shall not be used in the same BSS.

The mobile device should re not cache OSU profiles, as this could adversely affect future network selection.

5.5 Hotspot procedures for free public hotspots

Hotspot operators may provide Hotspot 2.0 based free public hotspot service using OSU, subscription and AAA server functions. The requirements of this use case are met using the mechanisms and protocols defined in this specification.

1. The user in a Free Public Hotspot initiates the online-signup registration process with the Free Public Hotspot's OSU server.
2. During the registration protocol (REP) exchange, the OSU server presents the terms and conditions to the user.
3. If the user accepts the terms and conditions, the OSU server issues a credential; if the user refuses, no credential is provisioned.
4. When the user/mobile device returns to the same Free Public Hotspot, the previously provisioned credentials are used to provide secure, automatic access.

If the terms and conditions change, then the user is taken through a subscription remediation process during which the new terms and conditions are presented. If the user accepts the changed terms and conditions, then a new credential is provisioned.

6. Mobile device procedures

Section 2.2 specifies the required capabilities of HS2.0-compliant mobile devices. This section describes additional procedures that are applicable when that mobile device is joining or is associated to an HS2.0-compliant network. Subsections 6.1 through 6.4 specify procedures for each of the mobile device states (Discovery, Registration, Provisioning and Access). Subsection 6.5 specifies mobile device procedures for filtering frames encrypted using the GTK.

6.1 Discovery state procedures

During the Discovery process the mobile device scans for HS2.0 capable networks and performs a discovery ANQP-element exchange to determine the capabilities of these networks prior to IEEE 802.11 association. HS2.0 capable networks are identified by the presence of the HS2.0 indication in the AP's Beacon and Probe Response frames.

Note: The mobile device may use the ANQP Domain ID in the Hotspot 2.0 Indication element, as well as implementation-specific methods, to reduce the number of GAS requests in order to more efficiently use the medium.

Next, the mobile device determines if it has one or more credentials (stored in or referenced by the PerProviderSubscription MO) that it can use to access the available HS2.0 networks. If multiple networks are available, including non-HS2.0 networks, the mobile device may select a network based on overall priority and possibly other heuristics such as signal strength), and automatically proceed to the Access state.

If HS2.0 networks are available but the user does not have credential access to any of those networks, the mobile device may allow the user to manually select an HS2.0 network from a list of available options presented on a user interface (UI) for OSU. Once the user selects a network from the OSU list, the mobile device proceeds to the Registration state (see section 6.2).

Annex C contains an example connection manager flowchart.

6.1.1 Home SP identification and connecting to Home SP hotspot

When selecting a hotspot, a Hotspot 2.0 release 1 capable[1] mobile device shall give preference to the connection to a hotspot operated by its Home SP over a hotspot not operated by its Home SP, except when overridden by user preferences.

When a Hotspot 2.0 release 2 capable mobile device selects a hotspot, it shall prefer a connection to a hotspot operated by its Home SP over a hotspot not operated by its Home SP, except when the choice is overridden by user preferences or by policy (see section 8 and PerProviderSubscription/Policy in section 9.1). An FQDN identifying the SP associated with a credential is provisioned in the mobile device.

Note: To identify its Home SP a mobile device can use the domain name generated from the Home PLMN ID drawn from the IMSI in its (U)SIM (e.g., wlan.mnc<MNC>.mcc<MCC>.3gppnetwork.org, see [37]).

An FQDN in the ANQP Domain Name list indicates the entity operating the hotspot (see [2]).

In order to determine whether it is connecting to a Wi-Fi AN that is operated by its Home SP, the mobile device matches the FQDN associated with its credential to the Domain Name list retrieved from the AP. Such a match is defined as a suffix match of the Domain Name with the FQDN. A suffix match is computed as an exact match of each label (as defined by section 2.3.1 in [5]) in the provisioned FQDN of the Home SP (starting with the top-level domain) with the corresponding label from the Domain Name ANQP-element.

[1] The mobile device is not Release 2 capable.

Notes:

1. The FQDN(s) drawn from the Domain Name ANQP-element can have more labels (see [5]) than the provisioned FQDN of the Home SP.
2. For Release 1, the Home SP's FQDN is provisioned to the mobile device by a means outside the scope of this specification. For Release 2, the Home SP's FQDN is provisioned using the PerProviderSubscription MO (see section 8).

6.1.2 Mobile device support for user preferences

The mobile device shall support the following configuration capabilities and behaviors with respect to user preferences:

- The mobile device shall provide the capability for a user to add or configure a prioritized list of preferred networks (SSIDs) on the mobile device.
- The mobile device's client manager application may automatically configure or manipulate, without user intervention, a prioritized list of preferred networks. But the client manager shall not override the configured user preferences.

  When the mobile device detects that it is within range of several networks (whether those networks are HS2.0-capable or not), the mobile device shall be capable of associating to the network that has the highest configured preference level, assuming environmental conditions are adequate − e.g., receive signal strength indicator (RSSI) levels.
- The mobile device should be capable of presenting a prioritized list of preferred networks to the user for manual selection of the network with which to connect.
- The mobile device shall provide the capability for a user to add or configure a prioritized list of subscription realms (e.g. NAI Realm and/or 3GPP Cellular Network).

  Note that these subscription realms may be used to authenticate to a network regardless of whether that network is an HS2.0 hotspot or not. Upon selection of the preferred realm, the mobile device selects the corresponding security credential. The mobile device shall follow the policy, if any, associated with the selected realm and the security credential.

  When a mobile device associates to a network that supports authentication with more than one of the realms in the mobile device's credential store, the mobile device shall be capable of selecting the highest priority credential to use for network authentication.

  When the mobile device detects it is within range of several networks, then, assuming environmental conditions are adequate (e.g., RSSI levels), it shall select the network that can authenticate the mobile device using the highest configured realm preference.

6.2 Registration state procedures

The mobile device enters the Registration state after it associates to an OSU ESS to perform online sign up for an account with a service provider. If the mobile device already has credentials for the current HS2.0 network, it does not enter the Registration state but proceeds directly to the Access state.

During the OSU procedure the mobile device provides information, such as contact information and payment method, as required by the SP to obtain an account. The mobile device could provide this information in an automated manner (e.g., as described in Annex E) or the user could manually enter the information during the OSU process. Credentials and related metadata (provisioned as described in section 6.3) are bound to this account. Section 7 describes the OSU procedures, including OSU server certificate validation.

6.3 Provisioning state procedures

The mobile device enters the Provisioning state after the mobile device and the OSU server exit the Registration state.

The mobile device installs the following while it is in the Provisioning state:

- The trust anchor Certificate Authority (CA) certificate(s) (used to validate the SP's AAA server certificate, the subscription remediation server certificate and the policy server certificate), as described in Section 7.3.1.
- An EAP-TLS (x.509v3) client certificate that is used for access to HS2.0 networks, as specified in Section 8 (if required).
- The PerProviderSubscription MO with credentials or credential metadata, WLAN security settings and other metadata for access to HS2.0 networks, as specified in Section 8. Note: the PerProviderSubscription MO can contain a username/password credential.

Some operators may have other methods of provisioning policy (e.g., re-distribution of SIM cards) which are out of scope of this specification. In this case, the policy node is not present in the PerProviderSubscription MO.

Once the provisioning process is successfully completed, the mobile device disassociates from the OSU ESS, exits the Provisioning state and proceeds directly to the Access state.

6.4 Access state procedures

The mobile device enters the Access state after it has associated to a network for which it has login credentials and WLAN security settings and has successfully authenticated to that network. For HS2.0 networks these settings were previously configured on the mobile device, either in the Provisioning state or via other means.

In the Access state, the mobile device mutually authenticates with the SP's AAA server using one of the EAP methods described in Table 1. Section 7.3.3 specifies the steps to validate the AAA server certificate as part of the EAP authentication process.

If authentication with the AAA server is successful, the mobile device receives full access to the Wi-Fi hotspot network.

6.4.1 Subscription expiry

The PerProviderSubscription MO provides metadata for the subscription, such as the credential's expiration date/time and usage limits. In the PerProviderSubscription MO, when the DataLimit (SubscriptionParameters/UsageLimits/DataLimit) or TimeLimit (SubscriptionParameters/UsageLimits/TimeLimit) is reached or passed, the subscription expires (SubscriptionParameters/ExpirationDate) or the credential expires (Credential/ExpirationDate). However, the mobile device should not immediately disassociate itself from the Wi-Fi AN. Rather, it should wait for the network to take this action and then select an alternative network. The alternative network may be the same as the original Wi-Fi AN.

6.4.2 Expiry of the subscription update timer

When the SubscriptionUpdate/UpdateInterval timer expires, the mobile device should contact the subscription server indicated by the SubscriptionUpdate/URI leaf node in the PerProviderSubscription MO (per the contents of the SubscriptionUpdate/UpdateMethod and SubscriptionUpdate/Restriction leaf node parameters).

If the subscription server is not available, the mobile device should operate using existing subscription metadata until a new server can be contacted. The mobile device should not attempt authentication to a subscription server more than once per session.

6.4.3 Expiry of the policy update timer

When the Policy/UpdateInterval timer expires, the mobile device should contact the policy server indicated by the Policy/URI leaf node in the PerProviderSubscription MO per the Policy/UpdateMethod and Policy/Restriction leaf node parameters.

If the policy server is not available, the mobile device should operate using existing policies until new policies are available for download. The mobile device should not attempt authentication to a policy server more than once per session.

6.4.4 EAP authentication failure

The mobile device may fail to successfully complete EAP authentication to the hotspot using a particular credential. Failure may be due to a variety of reasons, including invalid credentials, network problems, misconfigured APs, etc. However, authentication failure does not necessarily mean there is a problem with a credential or subscription. The credential may still be valid with other APs. Therefore, in the case of an EAP authentication failure, the mobile device:

- Shall not attempt, in the same ESS using a given credential within a ten minute interval, more than 10 consecutive EAP authentications that result in EAP authentication failures. The authentication process may restart after the expiry of the ten minute interval.
- Should not disable this credential from being used with other BSSs.

6.4.5 Association failure

The mobile device may fail to successfully associate to a Hotspot 2.0 AP, as indicated by a non-zero status code in the Association Response frame transmitted by the AP. The failure may be due to a variety of reasons, including the inability of the AP to handle additional associated STAs, the insufficient bandwidth at the QoS AP to handle another QoS STA, network problems, etc.

However, association failure does not necessarily mean there is a problem with the mobile device's credential or subscription. The credential may still be valid with other APs. Therefore, in the case of an association failure, the mobile device should not disable a credential from being used with other BSSs.

6.5 Filtering frames encrypted using the GTK

Once the mobile device has associated to an HS2.0 network, the mobile device:

- Shall drop all received gratuitous ARP messages when the Proxy ARP field is set to 1 in the Extended Capabilities element of the serving AP.
- Shall drop all received unsolicited Neighbor Advertisement frames when the Proxy ARP field is set to 1 in the Extended Capabilities element of the serving AP.

When the serving AP transmits frames containing an HS2.0 Indication element in which the value of the DGAF Disable bit subfield is set to 0, the mobile device should discard all received unicast IP packets that were decrypted using the GTK.

7. Online sign up and certificate management

7.1 Overview and goals

Online sign up (OSU) is the process by which a mobile device registers with an SP, enabling a user to select a plan for obtaining network access, and is then provisioned with the credentials necessary to securely connect to an AN.

Figure 28 shows example network architecture for OSU. Each SP network has an OSU server, an AAA server, and (access to) a CA. These devices may be co-located or separate. If they are separate, the communication between them is outside the scope of this document and is assumed to be secure (the entities are authenticated and communication between them is confidentiality and integrity protected). The hotspot has its own AAA server, and optionally an OSU server. The hotspot is configured to allow only HTTPS traffic to OSU servers.

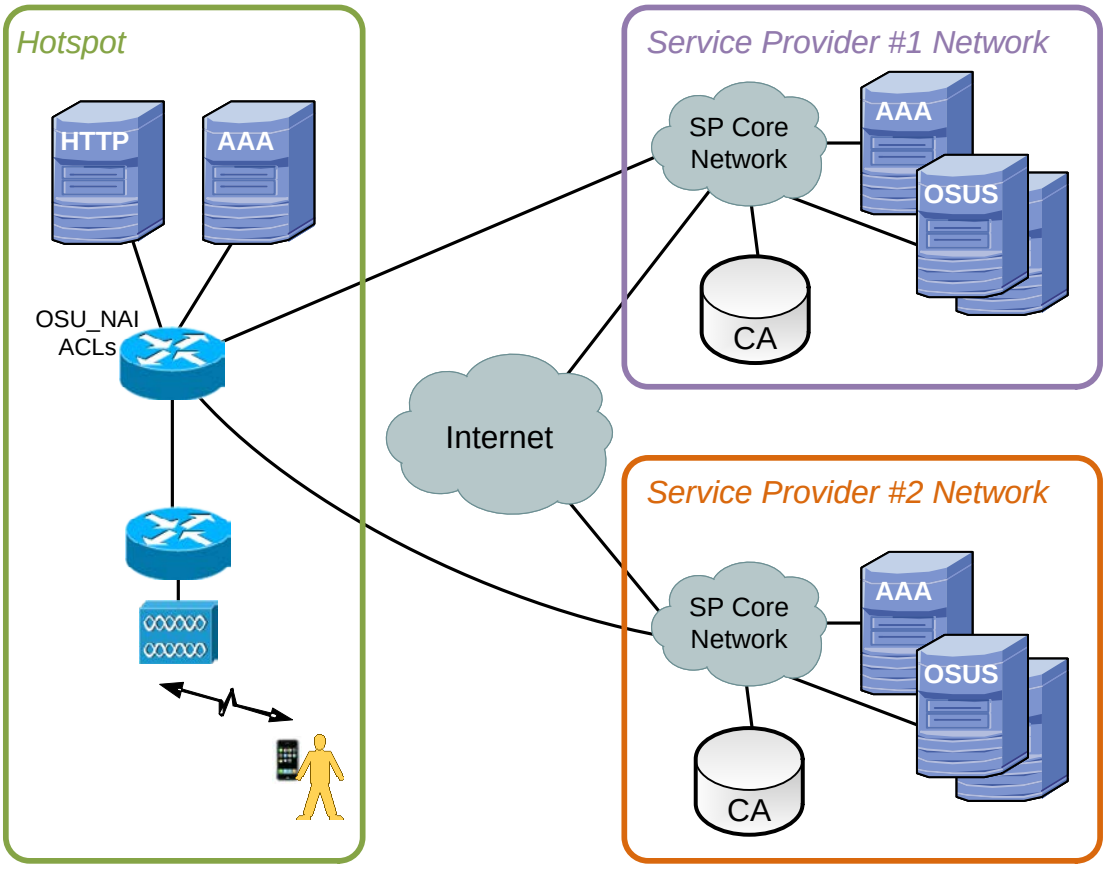


Figure 28: Example network architecture for online sign up

The OSU server registers new subscribers and provisions each subscriber's mobile device with a security credential. The OSU server may also initially provision the devices of existing subscribers. The OSU server does not provision SIM credentials, but might provision related metadata and policy. The SP's AAA server authenticates subscribers using the credentials obtained from the OSU server.

The security architecture for HS2.0 online sign up and initial authentication has the following goals:

1. Verify that the user is communicating with the intended SP network and OSU server.
2. Protect the communication between the mobile device and the OSU server from eavesdropping and modification.
3. Reduce the risk of an SP that has poor security practices from compromising other SPs.

The user's intent to register with a selected SP is indicated by the user's selection of an SP friendly name and/or icon, from a list of available options that are displayed on the mobile device UI.

Section 7.4 provides an overview of the message sequences used by a mobile device to securely connect with an OSU server.

7.2 Trust model

For the mobile device and the hotspot to trust each other:

1. Each OSU server shall hold a certificate signed by a Certificate Authority whose root certificate is issued by one of the CAs listed in [45].
2. The AAA server of the hotspot may act as an AAA proxy to relay messages to the AAA server of supported SPs. This communication is outside the scope of this document but is assumed to be secure (the entities are authenticated and sensitive communication between them is confidentiality and integrity protected).
3. An SP's AAA (EAP) server has a certificate signed by a Certificate Authority whose CA certificate is trusted by the connection manager of the mobile device.
4. Authentication credentials shall be securely stored on the mobile device and may only be accessed or modified by the user or corresponding SP.
5. The OSU server and the mobile device shall use HTTPS to exchange all registration, remediation, terms and conditions data.

Sections 7.3.2.2 and 7.3.3.2 contain the OSU and AAA certificate validation requirements.

As described in section 8, policy provisioning is a part of online signup. An SP that has a policy server shall identify that server using a policy server certificate (per section 7.3.6.1) and each mobile device shall validate the policy server certificates (per section 7.3.6.2).

After online signup, from time to time subscription remediation may be required (as described in section 8.1.3). Subscription remediation servers shall be identified by subscription remediation server certificates (per section 7.3.5.1), and each mobile device shall validate the subscription remediation server certificates (per section 7.3.5.2).

7.3 Public key certificate types

HS2.0 uses five types of public key certificates:

- Certificate Authority (CA) trust anchor certificates
- OSU server certificates
- AAA server certificates
- Subscription remediation server certificates
- Policy server certificates

All HS2.0 certificates shall be X.509v3 public key certificates based on RSA key pairs. The following subsections define the contents of these certificates and the required verification rules associated with them.

7.3.1 Certificate Authority trust root certificates

CA trust root certificates are installed on the mobile device and are trusted to validate Hotspot 2.0 server certificates received during the authentication process. OSU server CA trust root certificates are installed by a secure out-of-band method (e.g., during manufacturing) that is not defined by this specification. All OSU server CA trust root certificates authorized by the WFA

shall be installed in the mobile device. The mobile device shall use only WFA authorized OSU server CA trust root certificates for validating the OSU certificate chain containing an intermediate CA certificate and the server certificate (see[45]).

A mobile device installs trusted CA certificates downloaded during the credential and provisioning policy process into the Hotspot 2.0 trusted CA database. These CA certificates shall be used only for authenticating the following Hotspot 2.0 servers: AAA, subscription remediation, subscription update and policy servers.

The method for installing AAA server CA trust root certificates into the Hotspot 2.0 trusted CA database depends on the type of credential that the mobile device is using for authentication. If it is a username password credential, the mobile device shall download the AAA server trust root certificate(s) using the AAAServerTrustRoot node in the PerProviderSubscription Management Object (PPS MO) (see section 9.1). There may be multiple trust root certificates and they may include intermediate CA certificates.

When a certificate credential is used, it is possible that the CA employed to issue this credential is the same CA that issued the AAA server certificate. Therefore, if the credential is a certificate, the mobile device shall download CA certificate(s) using EST during the client certificate provisioning process (see section 7.6.1). If the PPS MO does not contain a AAAServerTrustRoot node, then these CA certificates are to be used as the AAA server CA trust anchor certificate(s). If the PPS MO contains a AAAServerTrustRoot node, this indicates that the AAA CA trust root certificate(s) are different than those used for issuing client certificates and the mobile device shall download the CA certificate(s) using information in the AAAServerTrustRoot node and use them as the AAA server CA trust anchor certificate(s).

The mobile device shall download and install CA trust root certificates for the subscription remediation server and policy server using the respective trust root node information in the PPS MO.

When the PPS MO is used to install CA trust root certificates, the mobile device shall use an HTTPS GET to download the certificates it does not already hold. The certificate fingerprint information in the PPS MO shall be used by the mobile device to determine whether CA trust root certificates are already installed. CA trust root certificates are expected to be formatted according to X.509v3 and encoded using the Distinguished Encoding Rules (DER) [56]. The mobile device shall support the download, installation, and use of X.509v3 CA trust root certificates that are DER encoded.

The mobile device shall support the following two types of CA certificates used for validating server certificates:

- Root CA certificate. The server certificate and any intermediate CA certificates shall properly chain to the root CA certificate.
- Intermediate CA certificate. The server certificate and any sub-CA certificates shall properly chain to the intermediate CA certificate.

7.3.2 OSU server certificate

OSU servers are web servers that support the HTTP protocol operating over TLS (i.e., HTTPS). HS2.0 OSU server certificates are used by OSU servers to facilitate the TLS connection. There are a small number of whitelisted CAs that are authorized to issue HS2.0 OSU server certificates (see [45]).

7.3.2.1 Composition of OSU server certificate

OSU server certificates are formatted according to X.509v3 and LogoType (RFC 3709, [15]). They shall contain an RSA public key and they should have the TLS Server Certificate attribute. The OSU server's public key should be at least 2048 bits and the OSU server certificate should be signed using sha256WithRSAEncryption employing the RSASSA-PKCS1-v1.5 signature

method defined in [13]. This assures maximum interoperability. The certificate should allow both signature operations and encryption operations for key transport.

An OSU server certificate:

a. Shall contain the OSU server FQDN as a DNSName type in the subjectAltName field.

b. Shall contain an Operator Friendly Name field and an Icon field.

c. The Operator Friendly Name shall be an otherName sequence to the subjectAltName. If multiple Operator Friendly name values are required (same operator, multiple human languages) then multiple otherName fields are present in the certificate. The type-id of the otherName shall be an id-wfa-hotspot-friendlyName:
   id-wfa OBJECT IDENTIFIER ::= { 1.3.6.1.4.1.40808 }
   id-wfa-hotspot OBJECT IDENTIFIER ::= { 1.3.6.1.4.1.40808.1.1 }
   id-wfa-hotspot-friendlyName OBJECT IDENTIFIER ::= { 1.3.6.1.4.1.40808.1.1.1}

   The id-wfa-hotspot-friendlyName contains exactly one language code and friendly name for an operator and shall be encoded as an ASN.1 type UTF8String. In the case where an operator friendly name is to be included in more than one human language, there shall be as many id-wfa-hotspot-friendlyName objects as there are human languages. The payload of the id-wfa-hotspot-friendlyName is the concatenation of the language code and the operator name, as provided in Figure 12. For example, for two human languages the following friendly names would be included as two separate otherName objects in the SubjectAltName: "engWi-Fi Alliance" and "fraWi-Fi Alliance".

d. The Icon field shall be a communityLogos type extension of LogotypeExtn [15]. This extension contains the hash of the logo that is displayed in the UI of a device. Multiple icons may be present, carried in the communityLogos extension. The operator shall provide a logotypeURI value indicating where the logo can be obtained. The filename provided in the Icon Metadata subfield of the OSU Providers list shall match the filename portion of the URL.

e. In the OSU server certificate, at least one LogotypeExtn that has a language code of “zxx” shall be present. Note: the mobile device uses the icon whose language code is set to “zxx” as the default icon when an exact match of the language displayed on the mobile device UI to the icon’s language is not available.

Note: RFC 3709 [15] states that LogotypeExtn shall be non-critical.

7.3.2.2 Processing of OSU server certificate

When the mobile device receives an OSU server certificate, it shall fully validate the certificate before it accepts the certificate and uses it to authenticate the OSU. The mobile device validation steps are:

1. Verify the validity of the certificate according to RFC 5280 [27]. If the certificate is not valid, verification fails and the certificate shall be rejected.

2. Perform both certificate and certificate path validation according to section 6 of RFC 5280 [27]. The OSU server certificate shall verify up the chain to a trusted root in the mobile device’s connection manager. The mobile device shall verify that the OSU server certificate trusted root certificate has been signed by one of the whitelisted OSU server trust roots. If the OSU server certificate has not been signed by any of the whitelisted trust roots, then verification fails and the mobile device shall abort the HS2.0 OSU process.

3. Verify that the identity of the OSU server certificate matches the identity of the SP chosen by the user of the mobile device. If one or more SP friendly names for the selected OSU provider were displayed on the user interface, this matching is accomplished by matching the chosen SP's friendly names with the id-wfa-hotspot-friendlyName tag fields. If one or more icons for the selected OSU provider were displayed on the user interface, this

matching is accomplished by matching the digest/tag(s) of the icon to the logotypeHash field of the LogotypeExtn(s) (see [15]). If both friendly name(s) and icon(s) were displayed on the user interface, then both matches shall be performed.

4. If the OSU server certificate does not contain an extended key usage attribute id-kp-serverAuth (TLS Server Certificate attribute), the mobile device should continue with certificate validation.

5. Certificate revocation services through OCSP [35] shall be provided for OSU server certificates. The mobile device shall verify that the OSU server's certificate has not been revoked using OCSP within the TLS connection, according to the procedures in [33], and should verify the revocation status of any intermediate CA certificates using the procedures in [35]. If the certificate has been revoked or the OCSP server returns any other error, or if the OCSP server cannot be contacted, then verification fails and the mobile device shall abort the HS2.0 OSU process. The OCSP responses shall be obtained through the TLS protocol by using the OCSP TLS Extension. The mobile device shall be capable of using the CA certificates received during the TLS exchange to build the certificate chain for OCSP response validation of the complete chain to the trust root.

6. Verify that the host name in the URL of the OSU server matches the DNSName in the OSU server certificate. If not, verification fails and the mobile device shall terminate the OSU procedure.

7.3.3 AAA server certificate

The AAA server in HS2.0 provides authentication, authorization, and accounting services to facilitate mobile device connections to an SP's network. The AAA server certificates are used for authentication through the EAP protocol.

7.3.3.1 Composition of AAA server certificate

The AAA server certificate for HS2.0 is formatted according to X.509v3. It shall contain an RSA public key. The AAA server's public key should be at least 2048 bits and the AAA server certificate should be signed using the sha256WithRSAEncryption using RSASSA-PKCS1-v1.5 signature method defined in [13]. This assures maximum interoperability. The certificate should allow both signature operations and encryption operations for key transport.

Either the AAA server certificate shall contain a SubjectAltName extension of type DNSName or the CommonName portion of the AAA server certificate's SubjectName shall be set to the AAA server's FQDN or domain name.

7.3.3.2 Processing of AAA server certificate

Upon receipt of an AAA server certificate, during the EAP protocol exchange, the mobile device shall process it for validity.

The validation steps made by the mobile device are:

1. Verify the validity of the certificate according to RFC 5280 [27]. The trust anchor CA certificates used to validate AAA server certificates are obtained during the online sign up process. If the certificate is not valid, verification fails and the certificate shall be rejected.

2. Verify in the AAA server certificate that the domain name from the FQDN drawn from the HomeSP/FQDN leaf node in the PerProviderSubscription MO is a suffix match of the domain name in at least one of the DNSName SubjectAltName extensions. If a SubjectAltName of type DNSName is not present, then the domain name from the FQDN shall be a suffix match to the CommonName portion of the SubjectName. If neither of these conditions holds, then verification fails.

3. Perform both certificate and certificate path validation according to section 6 of RFC 5280. The AAA server certificate chain shall contain the SP's certificate AAA trust root

identified in the PerProviderSubscription MO. If the AAA server chain does not contain the correct trust root, certificate validation fails.

4. If the AAA Server certificate contains one or more extended key usage attributes, unless one of those attributes is id-kp-serverAuth or anyExtendedKeyUsage, the certificate shall be deemed to fail validation and shall not be used for authentication.
5. If the AAA server certificate does not contain an extended Key usage attribute id-kp-serverAuth, the mobile device should continue with certificate validation; it may report an error to the user.
6. Certificate revocation services through OCSP should be provided for AAA server certificates. If OCSP is supported by the AAA server the Credential/CheckAAAServerCertStatus leaf node in the PerProviderSubscription MO shall be set to TRUE. When Credential/CheckAAAServerCertStatus is set to TRUE, OCSP responses shall be obtained through the TLS protocol by using the OCSP TLS Extension according to the procedures in [35]; the mobile device should verify the revocation status of any intermediate CA certificates using the procedures in [35]. In addition, if the certificate has been revoked or the OCSP server returns any other error, then verification fails and the mobile device shall abort the EAP authentication process. When the Credential/CheckAAAServerCertStatus leaf node is not present in the PerProviderSubscription MO or when the Credential/CheckAAAServerCertStatus leaf node value is set to FALSE, then the mobile device shall not require the AAA server certificate's revocation status to be available at the time of authentication.

7.3.4 AAA server certificate used with WFA Anonymous EAP-TLS

The AAA server in HS2.0 provides authentication, authorization, and accounting services to facilitate mobile device connections to a Hotspot network. The AAA server certificates are used for authentication through the EAP protocol.

7.3.4.1 Composition of AAA server certificate used with WFA Anonymous EAP-TLS

The AAA server certificate used with Anonymous EAP-TLS shall be formatted the same as the OSU server certificate, except that the DNSName type in the subjectAltName field is not required.

7.3.4.2 Processing of AAA server certificate used with WFA Anonymous EAP-TLS

Upon receipt of an AAA server certificate used with WFA Anonymous EAP-TLS, the mobile device shall fully validate the certificate before it accepts and uses the certificate to authenticate the AAA server. The validation steps made by the mobile device are:

1. Verify the validity of the certificate according RFC 5280 [27]. If the certificate is not valid, verification fails and the certificate shall be rejected.
2. Perform both certificate and certificate path validation according to section 6 of RFC 5280 [27]. The AAA server certificate shall verify up a chain to a trusted root in the mobile device’s connection manager. The mobile device shall verify that the AAA server certificate trusted root certificate has been signed by one of the whitelisted AAA server trust roots, as specified in [45]. If the AAA server certificate has not been signed by one of the whitelisted trust roots, then verification fails and the mobile device shall disassociate from the AP.
3. Verify that the identity of the AAA certificate matches the identity of the Hotspot operator chosen by the user of the mobile device. This matching is accomplished by matching the chosen hotspot operator's friendly name with the contents of the id-wfa-hotspot-friendlyName field if the Hotspot Operator was chosen by name, or by matching the

digest/tag of the icon to the logotypeHash field of the LogotypeExtn (see [15]), if the Hotspot Operator was chosen by icon.

4. If the AAA server certificate contains one or more extended key usage attributes without any of those attributes being id-kp-serverAuth or anyExtendedKeyUsage, the certificate shall be deemed to fail validation and shall not be used for authentication.

5. Certificate revocation services through OCSP shall be provided for AAA server certificates used for WFA Anonymous EAP-TLS. The mobile device shall verify the AAA server's certificate has not been revoked using OCSP within the TLS connection according to the procedures in [33] and should verify the revocation status of any intermediate CA certificates using the procedures in [35]. If the certificate has been revoked, then verification fails and the mobile device shall disassociate from the AP. The OCSP responses may be obtained through the TLS protocol by using the OCSP TLS Extension.

7.3.5 Subscription remediation server certificate

The subscription remediation server in HS2.0 provides subscription remediation services to facilitate the mobile device's ongoing connectivity to hotspots.

7.3.5.1 Composition of subscription remediation server certificate

Subscription remediation server certificates for HS2.0 are formatted according to X.509v3. They shall contain an RSA public key. The subscription remediation server's public key should be at least 2048 bits long and the subscription remediation server certificate should be signed using sha256WithRSAEncryption that employs the RSASSA-PKCS1-v1.5 signature method defined in [13]. This assures maximum interoperability. The certificate should allow both signature operations and encryption operations for key transport.

Subscription remediation server certificates shall contain a SubjectAltName extension of type DNSName.

7.3.5.2 Processing of subscription remediation server certificate

Upon receipt of a subscription remediation server certificate, the mobile device shall process it for validity before it accepts and uses the certificate with the TLS protocol to authenticate the subscription remediation server. The mobile device shall use the SubscriptionUpdate/URI leaf node in the PerProviderSubscription MO when it validates the subscription remediation server identity. The validation steps made by the mobile device are:

1. Verify the validity of the certificate according to RFC 5280 [27]. The trust root CA certificate used to validate the subscription remediation server certificate is obtained during the online signup process. If the certificate is not valid, verification fails and the certificate shall be rejected.

2. Verify that the domain name drawn from the SubscriptionUpdate/URI leaf node in the PerProviderSubscription MO is a suffix match of the domain name in the DNSName SubjectAltName. If the DNSName does not match, then verification fails. If the mobile device has not been provisioned with a PerProviderSubscription MO, then it shall use the URL provided in the WNM-Notification Request frame.

3. Perform both certificate and certificate path validation according to section 6 of RFC 5280 [27]. If the PPS MO has been provisioned, the subscription remediation server certificate chain shall contain the SP's subscription remediation trust root CA certificate. If a PPS MO is being initially provisioned to a mobile device (for example to one having a SIM credential; see section 8.5), the subscription remediation server's trust root shall be signed by one of the whitelisted CAs (see section 7.3.2) or by another CA whose trust root is installed, by a means outside the scope of this specification, for subscription

remediation server validation on the mobile device. If neither of these is true, then verification fails.

4. If the subscription remediation server certificate does not contain an extended key usage attribute id-kp-serverAuth, the mobile device should continue with certificate validation; it may report an error to the user.
5. Certificate revocation services through OCSP shall be provided for subscription remediation server certificates. The OCSP responses shall be obtained through the TLS protocol by using the OCSP TLS Extension according to the procedures in [33] and should verify the revocation status of any intermediate CA certificates using the procedures in [35].

7.3.6 Policy server certificates

Policy servers in HS2.0 provide network selection policy provisioning and update services to mobile devices according to the needs of Home SPs.

7.3.6.1 Composition of policy server certificate

The HS2.0 policy server certificate is formatted according to X.509v3. The certificate shall contain an RSA public key. The policy server's public key should be at least 2048-bits and the policy server certificate should be signed using sha256WithRSAEncryption using RSASSA-PKCS1-v1.5 signature method defined in [13]. This assures maximum interoperability. The certificate should allow both signature operations and encryption operations for key transport.

Policy server certificates shall contain a SubjectAltName extension of type DNSName.

7.3.6.2 Processing of policy server certificate

Upon receipt of a policy server certificate, the mobile device shall process it for validity before it accepts the certificate and uses the certificate with the TLS protocol to authenticate the policy server. The validation steps made by the mobile device are:

1. Verify the validity of the certificate according to RFC 5280 [27]. The trust root CA certificate used to validate the policy server certificates is obtained during the online signup process. If the certificate is not valid, verification fails and the certificate shall be rejected.
2. Verify that the Domain name drawn from the PolicyUpdate/URI leaf node in the PerProviderSubscription MO is a suffix match of the domain name in the DNSName SubjectAltName. If the suffix does not match, verification fails.
3. Perform certificate and certificate path validation, according to section 6 of RFC 5280 [27]. The policy server certificate chain shall contain the SP's policy trust root CA certificate. If not, verification fails.
4. If the policy server certificate does not contain an extended key usage attribute id-kp-serverAuth, the mobile device should continue with certificate validation, it may report an error to the user.
5. Certificate revocation services through OCSP shall be provided for policy server certificates. The OCSP responses shall be obtained through the TLS protocol by using the OCSP TLS Extension according to the procedures in [33] and should verify the revocation status of any intermediate CA certificates using the procedures in [35] .

7.4 Message overview for online sign up

An overview of the message exchange sequence for a mobile device to perform online sign up, including connection to an OSU server is shown in Figure 29.

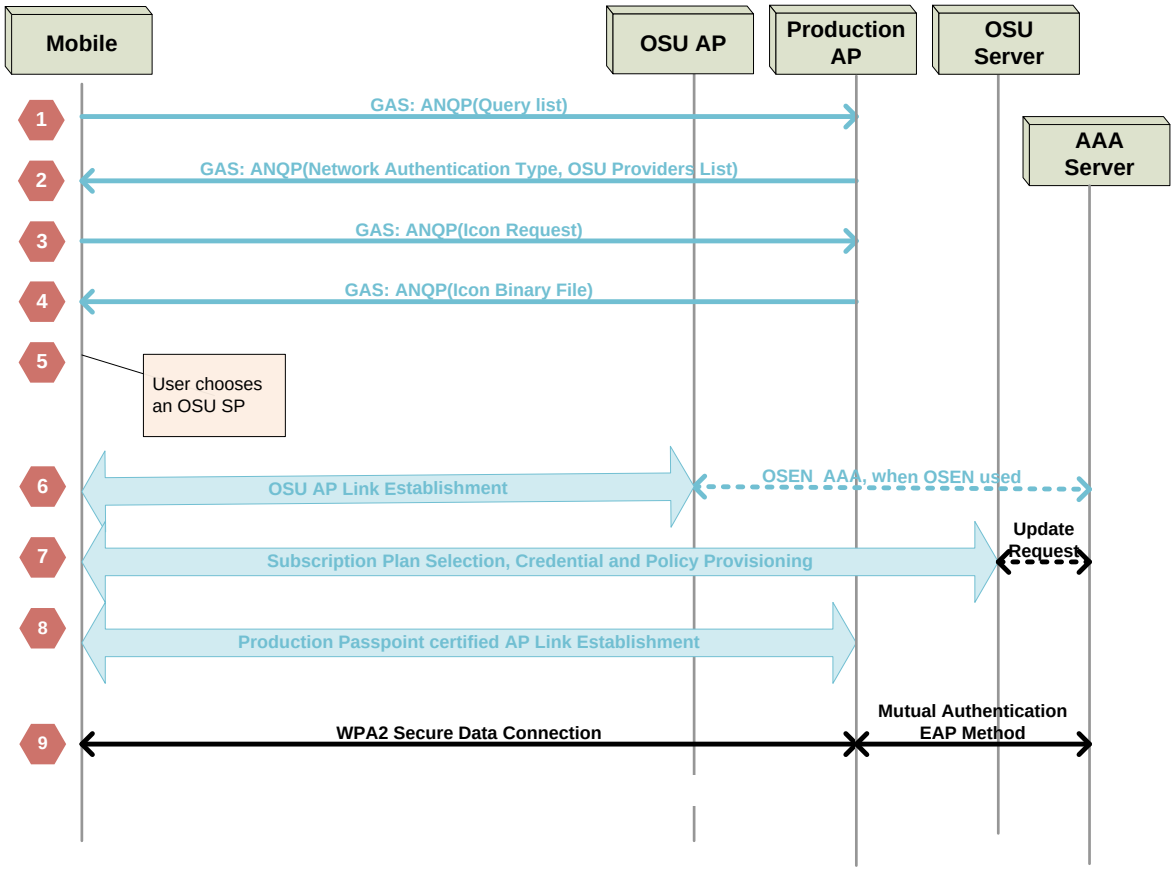


Figure 29: Message exchange diagram for connection to an OSU server

Steps 1 through 5 describe online sign up SP discovery and selection.

Step 1: The mobile device issues an ANQP Query frame for the network authentication type and OSU provider information.

Step 2: The AP returns the Network Authentication Type element and OSU Providers List element to the mobile device. If the Network Authentication Type message indicates that OSU is available, then the OSU Providers List element contains the OSU SSID and at least one OSU provider.

Step 3: The mobile device may request an OSU provider icon of the desired size in pixels using the Icon Request HS2.0 ANQP-element. In this example, an exchange of an icon is requested by the client device. A request of the icon is optional.

Step 4: If an icon was requested, the AP returns the icon binary file for the requested icon. If the OSU Providers List element contains the names of more than one OSU provider, steps 3 and 4 may be repeated for each provider.

Step 5: The mobile device displays on its UI a list of available OSU provider icon(s) and/or friendly name(s). If the user selects an icon and/or friendly name, indicating that online sign up for a subscription is desired, the mobile device continues with Step 6. The icon and/or friendly name of the OSU provider shall be validated per the requirements of section 7.3.2.2.

Step 6: The client device connects to the user-selected OSU ESS. The OSU ESS may be open or OSEN, see section 5.4.

Step 7: The user provides the information needed by the SP to sign up for a subscription. Credentials (certificate or username/password) and optionally network-selection policy are

provisioned on the mobile device. Details for this step are provided in section 8. When credentials are provisioned, the OSU server sends an update request to the AAA server with the mobile device's provisioned credential using a mechanism that is outside the scope of this specification.

Step 8: Using the newly provisioned credentials the mobile device disassociates from the OSU ESS and associates to an AP in the production ESS.

Step 9: The mobile device and AP establish a WPA2-Enterprise security association and the user is granted access privileges according to the user's subscription.

7.5 OSU operational requirements

An HS2.0 compliant mobile device that supports OSU meets the following operational requirements:

1. The mobile device shall not attempt OSU using HTTP (i.e., instead of HTTPS). If the mobile device is unable to initiate a TLS connection to the OSU server, it shall abort the OSU process.
2. If the mobile device UI displays the OSU provider's icon to the user, as an available option, either by itself or with the friendly name, and the user selects that icon thereby indicating his/her choice of registering with that OSU provider, the mobile device shall compute the SHA-256 hash of the icon retrieved using the Icon Binary Data ANQP-element. If the SHA-256 hash does not exactly match one of the SHA-256 hashes drawn from the OSU server's logotypeHash fields [15], the OSU process shall be aborted.
3. When selecting an icon, the mobile device shall select an icon whose language code exactly matches the mobile device's UI language; if an exact match is not available, the mobile device shall select an icon having the language code "zxx". Note: the above requirement does not preclude a mobile device implementation from displaying more than one icon for a given SP.
4. If the mobile device UI displays the OSU provider's friendly name to the user (in the human language selected by the mobile device from the available languages in the OSU Providers list element), as an available option, either by itself or with the icon, and the user selects that friendly name thereby indicating his/her choice to register with that OSU provider, the mobile device shall verify the friendly name exactly matches the same human-language friendly name drawn from the id-wfa-hotspot-friendlyName field of the OSU server certificate. If there is not an exact match, the OSU process shall be aborted. The mobile device shall only compare friendly names having the same human language code; if friendly names having same human language are not available from both the OSU Providers list element and the OSU server certificate, the OSU process shall be aborted.
5. All SOAP XML messages, excluding any OMA DM compliant MOs (if present), exchanged per the framework of Figure 44 shall be checked by the receiver of the message to ensure they are well formed and valid per the declared XML schema. The mobile device shall discard SOAP XML messages that are not well formed or that are not valid per the XML Schema. The subscription server may discard SOAP XML messages that are not well formed or that are not valid per the XML Schema.
6. All XML instance documents (i.e., XML not contained within a SOAP message), excluding any OMA DM compliant MOs (if present), that are exchanged per the framework of Figure 44 shall be checked by the receiver of the message to ensure they are well formed and valid per the declared XML schema. The mobile device shall discard XML instance documents that are not well formed or that are not valid per the XML Schema. The subscription server may discard XML instance documents that are not well formed or that are not valid per the XML Schema.

7. OMA DM compliant MOs should be validated according to their device description framework (DDF, see [39]). The mobile device should discard OMA DM compliant MOs that are not well formed or that are not valid per their DDF. The subscription server may discard OMA DM compliant MOs that are not well formed or that are not valid per their DDF.

7.6 Certificate enrollment and provisioning

7.6.1 Simple PKI enrollment using EST

When the mobile device requires a certificate for authentication purposes, it shall use the “Simple Enroll” request/response mechanism defined in EST [36], with the mobile device acting as the EST client and the OSU server acting as the EST server. EST operations and their specific URI operation paths shall be as described in [36]. The OSU server acts as a registration authority (RA) for the purposes of certificate enrollment.

The EST URI is constructed by appending the EST registered name and particular operational path to the enrollmentServerURI (see section A.3.5).

The mobile device obtains the CA certificate used to authenticate the AAA server, and any certificates necessary to chain up to it, by using the /cacerts operational path from EST. See the message sequence diagrams in section 7.6.4 and section 8 for additional details.

The mobile device shall generate a public/private key pair whose key length is suitable to provide at least 112 bits of security per SP 800-57 [46]. The mobile device should request Certificate Signing Request (CSR) attributes from the OSU server using the /csrattrs operational path mechanism defined in EST [36] and include them if applicable in its CSR.

If the OSU server requires use of a particular hash function or generation of a public key from a particular cryptosystem (e.g. a specific elliptic curve) it shall indicate that by including the relevant object identifiers in its CSR attributes response.

When forwarding the PKCS#10 CSR to the CA, the OSU server may request the CA to generate the certificate with modifications to the CSR. If necessary, CA servers may add, modify or remove extensions, or modify the subject name and/or subject alt name.

EST clients shall support the following identity attributes and include such information, when applicable, in a CSR if the attributes are returned by the OSU server in a CSR Attributes response:

- macAddress (OID 1.3.6.1.1.1.1.22), encoded as an IA5STRING type
- imei (OID 1.3.6.1.4.1.40808.1.1.3), encoded as an IA5STRING type
- meid (OID 1.3.6.1.4.1.40808.1.1.4), encoded as a BITSTRING type
- DevId (OID 1.3.6.1.4.1.40808.1.1.5), encoded as a PRINTABLESTRING type

The /csrattrs response, per EST [36], is an ASN.1 SEQUENCE OF objects and attributes, and each attribute is a SEQUENCE consisting of an object and a SET that contains 1 or more objects.

The listed identity objects shall be represented in the /csrattrs response as an attribute of Extension Request (OID 1.2.840.113549.1.9.14) and the specific identity objects shall be contained in the attribute's SET. Any identity object included in the resulting CSR shall be added as a PKCS#9 Extension Request [12].

When /csrattrs are used to indicate to the client to generate a particular kind of public key, an attribute indicating the type of public key shall be used with the attribute's SET indicating any particular characteristics of the public key. For example, a 4096-bit RSA key would be indicated with an attribute of rsaEncryption (OID 1.2.840.113549.1.1.1) and the attribute's SET containing a single ASN.1 INTEGER indicating the value 4096. An elliptic curve public key using the secp384r1 would be indicated as an attribute of id-ecPublic key (OID 1.2.840.10045.2.1) with the attribute's SET containing a single ASN.1 OBJECT IDENTIFIER indicating secp384r1 (OID 1.3.132.0.34). The subjectPublicKeyInfo of the resulting CSR shall contain the public key

algorithm of the indicated type and a public key that corresponds to the characteristics from the attributes SET.

When /csrattrs are used to indicate a preference for how to sign the subsequent CSR, an OBJECT IDENTIFIER in the SEQUENCE OF shall be used to indicate the particular preference. For example, eccdaWithSHA384 (OID 1.2.840.10045.4.3.3) or sha256WithRSAEncryption (OID 1.2.840.113549.1.1.11). The resulting CSR shall indicate a Signature Algorithm of the requested OBJECT IDENTIFIER.

An indication to the mobile device to provide proof-of-possession and binding of its identity to its public key shall be done by including the OBJECT IDENTIFIER challengePassword (OID 1.2.840.113549.1.9.7) in the SEQUENCE OF that constitutes the /csrattrs response.

The OSU server should validate that the identity attribute(s) included in the CSR is/are the same as the identity received during the mobile device's initial connection in the Wi-Fi DevDetail Extensions MO. If this information was not provided during initial connection, the OSU server should use a remote device management procedure to query the identity from the device and validate that it matches the identity included in the CSR.

If the mobile device identity does not match, the OSU server shall refuse to forward the CSR to the CA for certificate issuance and shall fail the certificate enrollment process for the mobile device. If the identity matches or the service provider does not enforce identity validation, the OSU server shall forward the CSR to the CA and let the CA policy decide whether to issue the certificate or not.

EST [36] defines a way for the client to link its identity with proof-of-possession. The OSU server shall indicate that the mobile device shall perform such a linkage by including the challengePassword object (1.2.840.11354.1.9.7) in its CSR attributes response. The mobile device shall perform proof-of-possession of its private key per EST [36].

7.6.2 Restricted use of HS2.0 client certificate

Since client certificate enrollment can be requested by a mobile device that is not registered as a trusted entity with the OSU service provider, it should be possible for a service provider to restrict the usage of the issued certificate for HS2.0 access authentication only. This provides a mechanism for the service provider to prevent unauthorized certificate enrollment and unauthorized access to other supported access technologies or services in their network, e.g., email server access authentication, VPN access authentication, IMS, etc.

The Key Usage and Extended Key Usage (EKU) values are used to restrict the purpose of a certificate. Currently, there is no value that is specific to HS2.0 or Wi-Fi access. Hence this specification is defining a new EKU for the purpose of HS2.0 access authentication. This HS2.0 new value for 'keyPurposeId' is 'id-kp-HS2.0Auth' and its OID is {1.3.6.1.4.1.40808.1.1.2}.

If the HS2.0 restriction is enforced by the service provider, when forwarding the PKCS#10 CSR to the CA, the OSU server shall indicate a requirement to include the HS2.0 EKU in the issued certificate. The CA should include the HS2.0 EKU in the issued certificate and flag that as critical for certificate processing.

If the HS2.0 restriction is enforced by the service provider, the AAA server processing the authentication request for HS2.0 access shall fail the authentication request if the above EKU is not included in the EKU list.

7.6.3 Processing of mobile device credentials

Upon receipt of a mobile device certificate, during the EAP (AAA) or TLS (HTTPS) protocol exchange, the AAA or HTTPS server shall process it for validity. The AAA or HTTPS server shall perform the following steps:

1. Check the integrity and validity of the certificate according to RFC 5280 [27]. If the certificate is not valid, authentication fails.

2. Perform certificate and certificate path validation according to section 6 of RFC 5280 [27]. If the client certificate chain does not contain the correct trust root, authentication fails.
3. Validate the mobile device's identity associated with the received mobile certificate.
4. Validate the HS2.0 EKU purpose 'id-kp-HS2.0Auth', if present.

7.6.4 Certificate enrollment message flow

Figure 30 shows the message exchange for the HS2.0 certificate enrollment process. The message exchange described in Figure 30 provides the details of step 9 in sections 8.3.3.2 and 8.4.2.2.

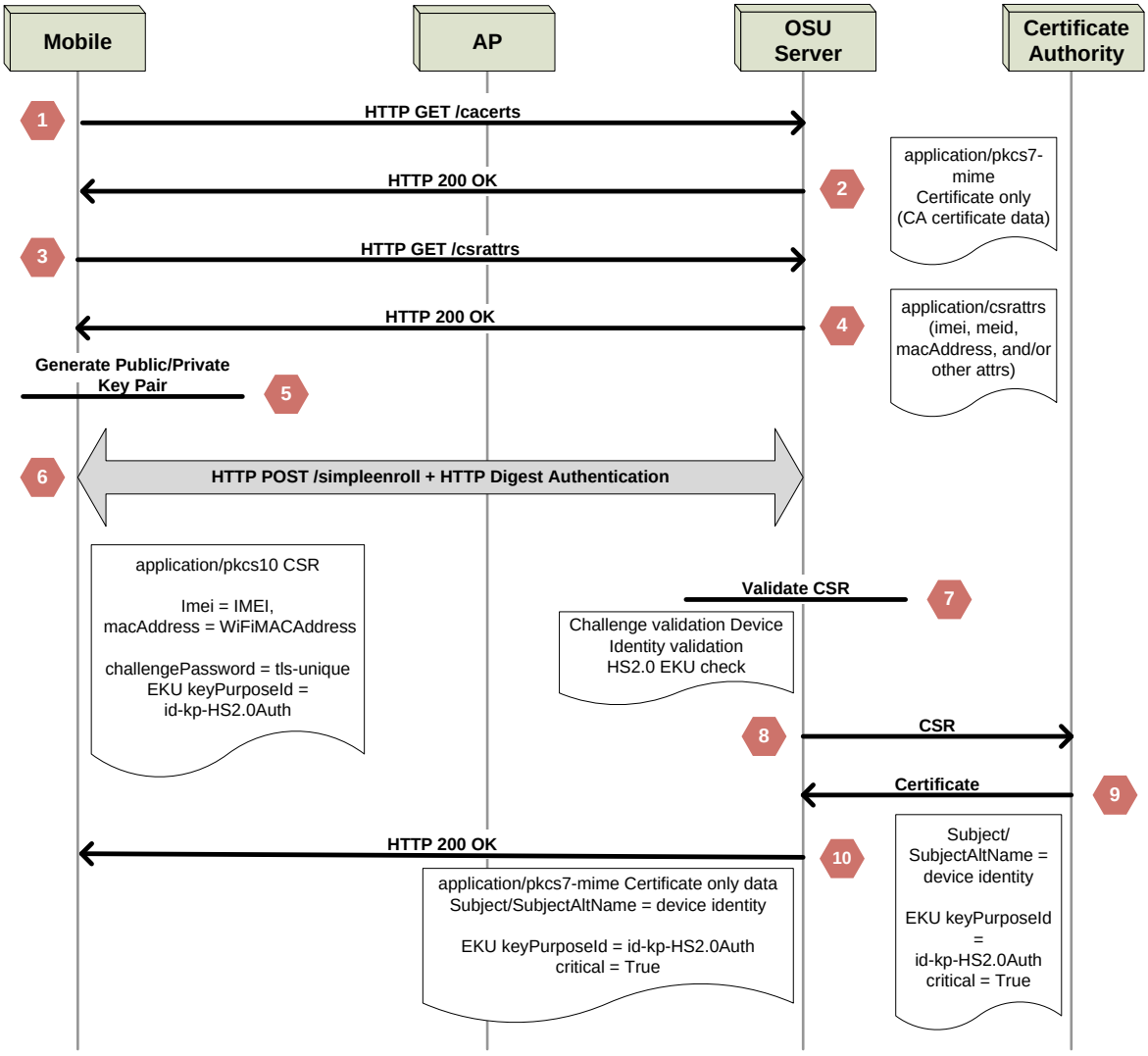


Figure 30: Certificate enrollment message exchange sequence

Immediately prior to step 1, the OSU server executes: getCertificate (step 8 in 8.3.3.2 and 8.4.2.2) it includes the getCertificate XML instance document defined in A.3.5. The received XML instance document contains the enrollment protocol, the enrollment server URI, the estUserId and the estPassword.

Step 1: The mobile device starts the certificate enrollment exchange by issuing HTTPS GET to the certificate enrollment URI (/cacerts). The client requests the CA certificate. See section 7.3.1.

Step 2: The OSU server responds to the client with a “degenerate” certs-only PKCS#7 message in the format of “application/pkcs7-mime” containing the CA certificate(s).

Step 3: By issuing HTTPS GET to the certificate enrollment URI (/csrattrs); the client requests the CSR attributes from the OSU server.

Step 4: The OSU server responds to the client with the CSR Attributes Response message in the format of “application/csrattrs” containing the required CSR attributes (including the required identity attributes, if any).

Step 5: The device generates the public/private key pair.

Step 6: The client POSTs a PKCS10-formatted message with the requested CSR attributes (including any device identity attributes and EKU indicating request for HS2.0 certificate enrollment). In this step, the OSU server shall request HTTP authentication using the Digest method in accordance with the procedures described in [9]. The mobile shall provide its username (estUserID, see section A.3.5) and password (estPassword, see section A.3.5) digest to the OSU server in accordance with [9].

Step 7: Upon successful HTTP authentication, the OSU server validates the challenge, device identity, and EKU.

Step 8: The certificate enrollment request (CSR) is delivered to the CA. Note: the mobile’s device identity is provided to the OSU server in the DevInfo and DevDetail MOs. The OSU server may use these identities in addition to those provided in the enrollment request. The OSU server may also request the CA to include in the certificate the EKU indicating HS 2.0 use by setting keyPurposeId to the value id-kp-HS2.0Auth including the Critical indicator.

Step 9: The CA issues a certificate and if required by the SP, includes the EKU indicating HS2.0 and Critical indicator turned on. The device identity is included in Subject field or SubjectAltName extension in the certificate, as per service provider’s CA policy. The CA delivers the certificate to the OSU server.

Step 10: The certificate is provisioned to the mobile device.

The exchange following certificate enrollment is described in sections 8.3.3.2 and 8.4.2.2 (i.e. steps 10 through 16).

7.7 Anonymous EAP-TLS

This specification defines a WFA Vendor specific EAP method, WFA Anonymous EAP-TLS, which can be used for OSU access. WFA Anonymous EAP-TLS is a profile of EAP-TLS (specified in [25]), in which the supplicant authenticates the AS, but the AS does not authenticate the client. WFA Anonymous EAP-TLS shall only be used for the OSU ESS; it shall not be used for the production ESS (see section 5.4.2).

WFA Anonymous EAP-TLS is indicated in EAP [16] using the expanded EAP type (254) having the Vendor-Id set to the WFA SMI code 0x009F68 and the Vendor-Type set to 0x0000000D.

WFA Anonymous EAP-TLS shall be compliant with [25], with the following constraints:

- The EAP server shall not transmit a certificate_request message to its EAP peer (i.e., the mobile device) and the peer does not provide a certificate (i.e., client authentication is not performed).
- The client's EAP identity is drawn from the OSU_NAI field in the OSU Provider List ANQP-element (see section 4.8.1.5) when WFA Anonymous EAP-TLS is used.
- Certificate validation shall be performed according to the procedures defined in section 7.3.4.2.

- Notes:
  - Section 2.1.4 of [25] is not supported.
  - The peer identity (see section 5.2 in [25]) is anonymous.

8. Subscription provisioning

Subscription provisioning comprises credential and related metadata provisioning, policy provisioning and Home SP information provisioning for ND&S (network discovery and selection).

See Annex A for message definitions and examples of OMA DM Packages and SOAP XML methods.

8.1 Overview

Subscription and policy provisioning for ND&S is carried out using either OMA DM or SOAP XML protocols. Provisioning using the OMA DM protocol is described in section 8.3. Provisioning using the SOAP XML protocol, SPP, is described in section 8.4.

Subscription and ND&S policy data are transported in the PerProviderSubscription MO, defined in section 9. The PerProviderSubscription MO is used regardless of whether the transfer protocol is OMA DM or SOAP XML.

Provisioning policy is optional for SPs. The policy provisioned by any SP applies only to the use of the credentials in the same PerProviderSubscription MO instance as the policy itself. If a mobile device has more than 1 subscription, user preferences determine the priority of each subscription and thus the order in which network-selection policies are applied in selecting a Wi-Fi network to join.

An example of a SP's core network is shown in Figure 31. In the figure, the three servers not labeled "AAA" are collectively referred to as "subscription servers":

- OSU: the OSU server is used to register new subscribers and provision them with credentials. The OSU Providers List ANQP-element (see section 4.8) informs the mobile device how to contact this server. The OSU server communicates with the Certificate Authority (CA) during certificate enrollment.
- Policy: the policy server is used to provision ND&S policy to the mobile device (see the policy node in sections 9.1.2, 8.3.4 and 8.4.4). The policy node of the PerProviderSubscription MO informs the mobile device how to contact this server.
- Subscription remediation (Sub Rem): the subscription remediation server is used to correct problems known to the SP with the subscriber's credentials, provisioned data (e.g., PerProviderSubscription MO) or the subscription itself (e.g., delinquent payment). The SubscriptionUpdate node of the PerProviderSubscription MO informs the mobile device how to contact this server.

Note: the subscription servers defined above are logical servers. Figure 31 does not imply a preferred implementation.

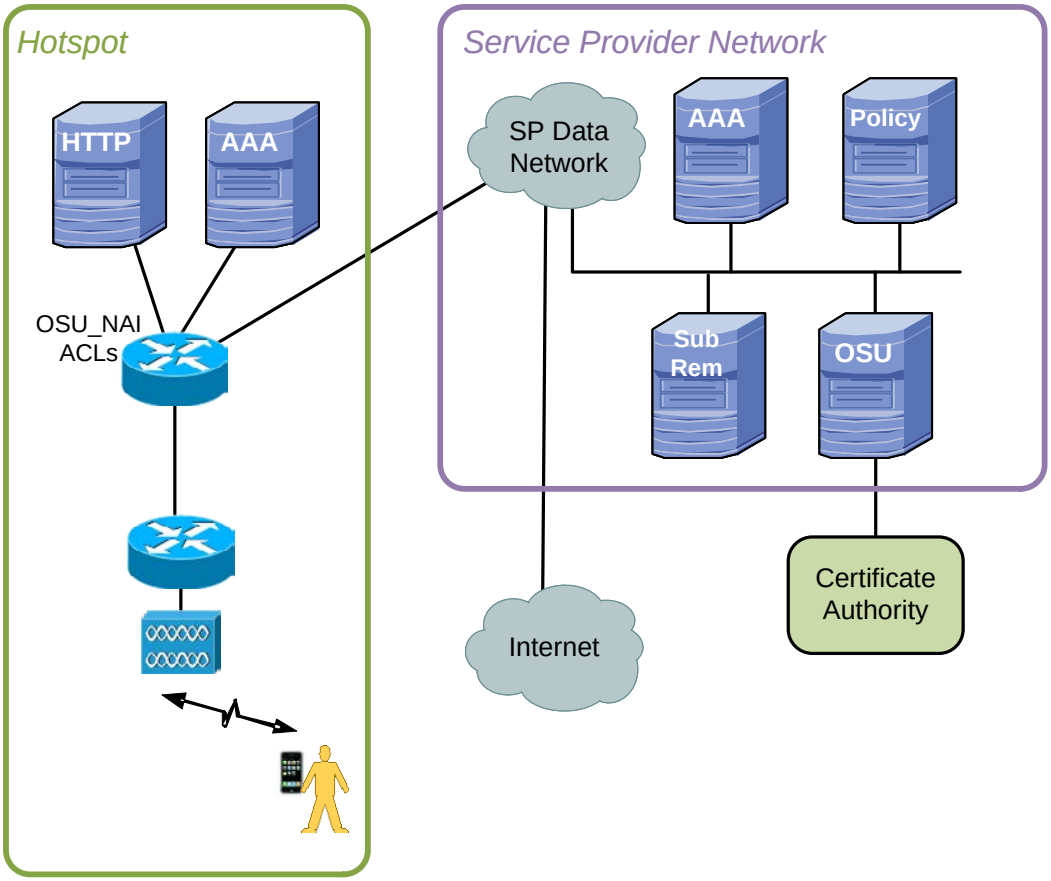


Figure 31: Example Service Provider Network with Subscription Servers

Networks typically impose access restrictions on the AP, which restricts mobile device connectivity to the subscription servers during the credential provisioning process. This is done to protect the network and to prevent a user from obtaining unauthorized Internet access. During the subscription management process this may be done to prevent the user from obtaining Internet access, for example, because the user has not paid his or her bill. However, these access restrictions do not prevent the mobile device from contacting a subscription server.

An SP decides whether to use OMA DM or SOAP XML procedures to perform credential provisioning and subscription remediation; the mobile device is informed of the protocol in the OSU Providers List ANQP-element (see section 4.8).

The message sequence diagrams are different for the mobile device holding (or being provisioned with) username/password credentials and the mobile device holding (or being provisioned with) certificate credentials. An example of this is shown in Figure 33 and Figure 34.

8.1.1 Subscription access restrictions

The mobile device implementation shall not provide read, write, modify or delete privileges for an instance of the PerProviderSubscription MO to any SP other than the SP that provisioned it.

Note: this is addressed by the use of the OMA DM access control list (ACL; see section 7.1.1 in [38]).

8.1.2 Subscription credential provisioning options

When a mobile device uses the OSU procedure to register for a subscription and be provisioned with credentials, the procedures defined in the following subsections shall be followed.

The SP decides the type of credentials to provision. The following credential types are supported:

- User provided username and password.
- SP provided username and password. In this case, password updates are machine-managed.
- Certificates that are supplied by the SP. These certificates may either be supplied using online enrollment (see section 7) or be provisioned onto the mobile device using an out-of-band method that is outside the scope of this specification.
- Certificates which are supplied by the mobile device manufacturer (e.g., an IEEE 802.1ar-compliant manufacturing certificate).

Note: Provisioning a SIM credential is out of scope of this specification.

Subscription data in the PPS MO is initially provisioned by the OSU server at the same time a mobile device's credentials are provisioned. Subscription data, for example, provides Home SP identifiers; see the PerProviderSubscription/HomeSP node and its child nodes. After initial provisioning, a mobile device may be requested to contact a subscription server periodically in order to obtain updated subscription data; this is accomplished by provisioning a PerProviderSubscription MO which includes a SubscriptionUpdate node containing the subscription server's URI and UpdateInterval leaf node set to the desired update interval. The UpdateInterval determines how often the mobile device should check for subscription data update. The mobile device shall keep a timer which keeps track of the time since the last subscription update. When the time since the last subscription update approaches, reaches or passes the UpdateInterval value and the mobile device is associated to a Wi-Fi network, subject to the restrictions in the PerProviderSubscription/SubscriptionUpdate/Restriction leaf node, the mobile device shall attempt to contact the subscription server for an update. If the mobile device is not associated to a Wi-Fi network meeting the restrictions, it shall attempt to contact that subscription server the next time it associates to a Wi-Fi network meeting those restrictions. The 'PerProviderSubscription/SubscriptionUpdate/URI' node identifies the subscription server. For subscription update, the mobile device uses the OMA DM or SOAP XML (SPP) protocol as enumerated in the SubscriptionUpdate/UpdateMethod leaf node in the PerProviderSubscription MO.

8.1.3 Subscription remediation

From time-to-time, the user's subscription may be in need of remediation. The SP determines when this is necessary; examples include: password expiration, delinquent payment of the account or acceptance of changes to terms and conditions.

For the purposes of this document, the term "remediation" refers broadly to the process of fixing a problem in the subscriber's subscription; this definition includes provisioning updated credentials to a mobile device, updating the PerProviderSubscription MO on a mobile device or performing some online function to update the subscription (i.e., no new credentials/data are provisioned to the mobile device).

During the subscription remediation process, the Wi-Fi infrastructure will typically restrict access of the mobile device to only the subscription remediation server.

The message exchange sequence for subscription remediation for OMA DM is provided in section 8.3.3 and for SOAP XML in section 8.4.3.

The need for subscription remediation is stored in a subscriber's database entry in the Home SP's infrastructure. If the mobile device is HS2.0 Release-2 capable or higher (see section 3.1.1) and the AP is HS2.0 Release-2 capable or higher, subscription remediation is possible (see

Annex D for information on RADIUS attributes). Then subsequent to an authentication request from that subscriber on a Wi-Fi AN, the need for remediation is signaled to the mobile device as shown in Figure 36. If subscription remediation is not possible then the Home SP may choose to send a RADIUS access reject.

Subscription remediation message sequences may be used to update data in the PerProviderSubscription MO when a network-initiated update method is needed. Otherwise, the mobile device initiated update method for the PerProviderSubscription MO (see PerProviderSubscription/<X+>/SubscriptionUpdate in section 9.1.2) may be used.

Note: the PerProviderSubscription/UpdateIdentifier present in the HS2.0 Indication element is provided to the AP during the association process and relayed to the AAA server using the HS2.0 mobile device version RADIUS attribute (see D.1.3). AAA servers can use the UpdateIdentifier to help determine if subscription remediation is needed. For subscription remediation the mobile device uses the OMA DM or SOAP XML (SPP) protocol as enumerated in the SubscriptionUpdate/UpdateMethod leaf node in the PerProviderSubscription MO.

The mobile device then engages in a subscription remediation message exchange with the subscription remediation server. Note that the subscription remediation server's FQDN is part of the PerProviderSubscription MO (see section 9.1).

There are two classes of remediation: machine and user remediation:

- Machine remediation means the problem(s) with the subscription can be remediated without any user intervention.
- User remediation means the problem(s) with the subscription require user intervention.

Mobile device authentication to the subscription remediation server uses different authentication methods, depending on the credential the mobile device possesses:

- HTTP digest (see [9]) using the username and password used for Wi-Fi network access.
- TLS (see [29]) using the mobile device's certificate used for Wi-Fi network access.
- HTTP digest (see [9]) using the username and password contained in the PerProviderSubscription/<X+>/SubscriptionUpdate/UsernamePassword node; if this node is present in the mobile device's PPS MO, it is used instead of the credential for Wi-Fi network access.
  Note: this method is recommended for subscriptions using a SIM credential (see section 8.5).

The message sequence diagrams for certificate re-enrollment (e.g., to remediate certificate expiry) are shown in Figure 40, Figure 41, Figure 52 and Figure 53.

8.1.4 Subscription management web content

During OSU or subscription remediation the user may be presented with a web page to provide subscription information. To ensure proper rendering of the OSU web pages by the mobile device OSU client (the application that renders OSU webpages) and in order to accommodate any operation requiring user intervention by the service provider, the web content should comply with the W3C standards [ref: http://www.w3.org/TR].

Note: for maximum OSU-client support, web pages should consider using HTML4.01/XHTML1.0 [ref: http://www.w3.org/TR/REC-html40 , ref: http://www.w3.org/TR/xhtml1] and CSS 2.1 [ref: http://www.w3.org/TR/CSS2] web standards. It is advised to run code compliancy validation against web site validation services. Proper scaling of images based on the OSU-client's supported resolution is highly desired for better user experience and successful operation.

Note: graceful degradation strategies should be considered for content that may not be visible or supported by OSU-clients or OSU-clients incapable of supporting advanced web scripting languages or new web standards if those are incorporated in the web page. It is recommended that OSU-clients enable support for javascript.

8.1.5 Policy provisioning and update

Policy may be initially provisioned by the OSU server at the same time a mobile device's credentials are provisioned. Alternatively, a mobile device may be requested to contact a policy server after initial credential provisioning has been completed; this is accomplished by provisioning a PerProviderSubscription MO which includes a PolicyUpdate node containing the policy server's URI and UpdateInterval leaf node set to a small value (e.g., 1 minute).

The policy includes the specification of how often the mobile device should check for policy update as specified in the PerProviderSubscription MO 'PerProviderSubscription/Policy/PolicyUpdate/UpdateInterval' node. The mobile device shall keep a timer which keeps track of the time since the last policy update. When the time since the last policy update approaches, reaches or passes the UpdateInterval value and the mobile device is associated to a Wi-Fi network, then, subject to the restrictions in the PerProviderSubscription/Policy/PolicyUpdate/Restriction leaf node, the mobile device shall attempt to contact the policy server for an update.

If the mobile device is not associated to a Wi-Fi network that meets the restrictions, it shall attempt to contact that policy server the next time it associates to a Wi-Fi network that meets the restrictions.

The 'PerProviderSubscription/Policy/PolicyUpdate/URI' node identifies the policy server. For policy update, the mobile device uses the OMA DM or SOAP XML (SPP) protocol as enumerated in the PolicyUpdate/UpdateMethod leaf node in the PerProviderSubscription MO.

If a Home SP needs to "push" an updated policy to a mobile device, it may use the subscription remediation process. During subscription remediation the subscription remediation server may provision the updated policy directly (assuming that the PerProviderSubscription's ACL server identity is set appropriately) or it may set the PolicyUpdate/UpdateInterval node to a small value to cause the mobile device to contact a policy server after a short time period.

When the mobile device possesses a SIM credential, policy provisioning using the PerProviderSubscription MO is still applicable for use with Wi-Fi ANs. The 3GPP mobile operator may use the policy provisioning procedures covered by this specification for provisioning. However, because there currently is no standardized method to use SIM credentials for TLS or HTTP authentication with a policy server, the update of SP Policy shall be accomplished after the mobile device has completed IEEE 802.1X authentication using SIM credentials (i.e., EAP-SIM, EAP-AKA or EAP-AKA').

To facilitate authentication for initial policy provisioning at a policy server, the AAA server passes the mobile device's IMSI, MSISDN and a 128-bit hash to the policy server (see the message sequence diagrams in section 8.5).

The protocols and procedures used between the authenticator (e.g., the AP) or AAA server and OSU server or policy server to communicate the mobile device's identity are outside the scope of this specification.

Mobile device authentication to the policy server uses different authentication methods, depending on the credential the mobile device possesses:

- HTTP digest (see [9]) using the username and password used for Wi-Fi network access.
- TLS (see [29]) using the mobile device's certificate used for Wi-Fi network access.
- HTTP digest (see [9]) using the username and password contained in the PerProviderSubscription/<X+>/Policy/PolicyUpdate/UsernamePassword node; if this node is present in the mobile device's PPS MO, it is used instead of the credential for Wi-Fi network access
  Note: this method is recommended when the mobile device possesses a SIM credential (see section 8.5).

8.2 Mobile device management tree

The OMA DM framework specifies a management tree that is used to organize all MOs in a mobile device into a hierarchical tree structure (see [38]). Each node in each MO in the tree may be uniquely addressed with a URI. OMA DM specifications do not impose any structure on the mobile device's management tree except for mandating the support of three MOs (DevInfo, DevDetail and DMAcc).

HS2.0 supports subscription and policy provisioning using both OMA DM and SOAP XML protocols. SOAP XML does not provide the capability to discover the mobile device's management tree structure. And since this specification assumes that any given mobile device may have multiple subscriptions for Wi-Fi ANs, it is imperative that two different SPs do not attempt to use the same position in the mobile device's management tree for their respective PerProviderSubscription MOs. To prevent this possibility, the mobile device shall make available the management tree structure shown in Figure 32.

The SP is not required to use this structure if it provisions subscriptions using the OMA DM protocol.

An MO ACL (see section 7.7.1 in [38]) may prevent operations on a mobile device's management tree; therefore, if a server decides to add its PerProviderSubscription MO at another location in the mobile device's management tree, it should first check that this location is vacant so that an Add command will not return a failure status. The SP provisioning a PerProviderSubscription MO using the SOAP XML protocol shall use the management tree structure shown in Figure 32.

The SP shall not provision subscriptions in the mobile device's management tree at any child node of ./Wi-Fi other than ./Wi-Fi/SP FQDN where SP FQDN is their (i.e., the provisioning SP's) domain name. The mobile device shall not permit a subscription server to add a PPS MO at the child node of ./Wi-Fi/SP FQDN unless the value of SP FQDN is a suffix match of the domain name in the DNSName of the SubjectAltName in that server's certificate.

In order to protect the user's privacy, the mobile device may return a "(401) Unauthorized" failure (see section 6.6.1 in [49]) or a "Permission denied" error (see Table 15) whenever a subscription server, whose <server-identifier> does not match the <server-identifier> in an ACL of any child node of ./Wi-Fi, attempts to access that child node, whether or not that child node exists. Note: in other words, these errors codes are used instead of "(404) Not found" (OMA DM) or "Not found" (SPP), so that an unauthorized subscription server cannot determine by trial and error whether a subscription from a particular SP exists on the mobile device or not.

In order to preserve the capability for multiple SPs to provision credentials, the root node in a HS2.0-compliant mobile device shall have its ACL set to "Add=*&Get=*" (cf. section 7.7.1.2 in [38]); furthermore, the mobile device shall not permit any server to change the value of the ACL on the Root node. For subscription remediation, the mobile device uses the OMA DM or SOAP XML (SPP) protocol as enumerated in the SubscriptionUpdate/UpdateMethod leaf node in the PerProviderSubscription MO.

The mobile device shall automatically add the "Wi-Fi" interior node when it is required as a container for the PerProviderSubscription MO. The PerProviderSubscription MO is a required child node of the SP FQDN interior node.

The mobile device's OMA DM tree shall contain the following mandatory MOs:

- DevInfo
- DevDetail
- Wi-Fi

The DevInfo and DevDetail are generic OMA DM MOs defined by the OMA DM specification [41], which describes how these objects are used in a mobile device.

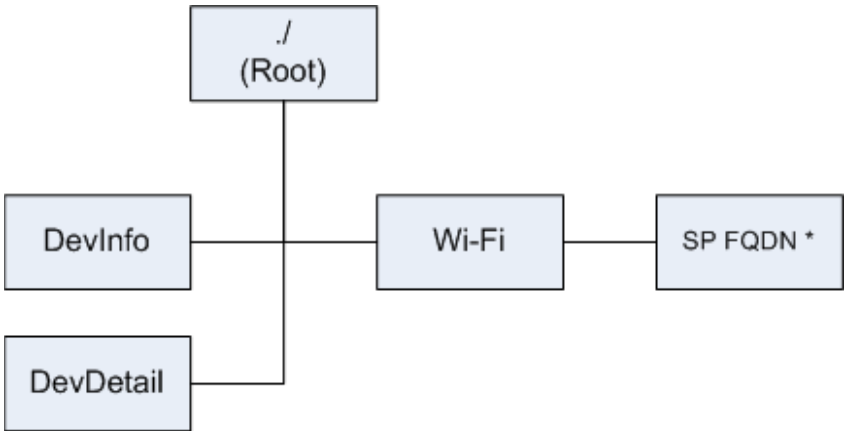


Figure 32: Required Mobile Device Management Tree Structure

./DevInfo

Status	Occurrence	Format	Min. Access Types
Required	One	Node	Get

This node contains mobile device information for the OMA DM or SOAP XML server, which is sent from the client to the server. “urn:oma:mo:oma-dm-devinfo:1.0”. See [41].

./DevDetail

Status	Occurrence	Format	Min. Access Types
Required	One	Node	Get

This node contains general mobile device information. The URN for this MO is “urn:oma:mo:oma-dm-devdetail:1.0”. See [41].

./Wi-Fi

Status	Occurrence	Format	Min. Access Types
Required	One	Node	Get

This interior node acts as a container for the Wi-Fi Management Object, extending the original OMA DM MO specification.

The Wi-Fi node shall have its scope set to Permanent (see section 6.2.3 in [38]) and its ACL set to "Add=*&Get=*" (see section 7.7.1.2 in [38]); furthermore, the mobile shall not permit any server to change the value of the Wi-Fi node's ACL.

./Wi-Fi/SP FQDN

Status	Occurrence	Format	Min. Access Types
Required	ZeroOrMore	Node	Add, Get, Replace and Delete

This interior node acts as a unique identifier for each SP PerProviderSubscription MO defined in section 9.1.

When the mobile device is initially provisioned with a PerProviderSubscription MO, it shall automatically create this node in its management tree (see section 6.6.1 in [49]).

Note: when the mobile device automatically creates this interior node, according to section 7.7.1.3 in [38], it shall set the node's ACL to "Add=<server-identifier>&Replace=<server-

identifier>&Delete=<server-identifier>&Get=*" or "Add=<server-identifier>&Replace=<server-identifier>&Delete=<server-identifier>&Get=<server-identifier>" where <server-identifier> means the name of the identity of the server creating the node.

The <server-identifier> shall be set to the DNSName drawn from the server certificate of the server provisioning the PerProviderSubscription MO.

8.3 Provisioning using OMA DM

Provisioning using OMA DM is defined for three cases:

1. Provisioning Username and Password credentials (see 8.3.2.1)
2. Provisioning Certificate credentials (see 8.3.2.2)
3. Provisioning using existing mobile device provided certificates (see 8.3.2.3)

The DM commands and messages described in this section are compliant with the DM protocol. [39].

8.3.1 Overview

The mobile device uses an OMA DM client for the provisioning process. The mobile device's OMA DM client receives Home SP network profile information (Subscription Information and Policies) from the (OMA DM) management server. After successful credential and policy provisioning, the mobile device accesses the secure Wi-Fi AN using the provisioned credentials.

An overview of the OMA DM provisioning framework is provided in [39]. Per [53], the mobile device may include a <MaxMsgSize> element in OMA DM messages. The mobile device or server that includes this element shall have this element's value set to 5000 bytes or greater. In addition, when the <MaxMsgSize> element is used by a mobile device, the value of LrgObj in the DevDetail MO shall be set to true.

OMA DM provisioning in Hotspot 2.0 shall use XML, which has the HTTP content type of application/vnd.syncml.dm+xml.

8.3.2 Subscription provisioning

The procedures used by the mobile device to connect with the OSU server are provided in section 8.1.

The OSU server should use the registration protocol defined in Annex E to embed standard tags for registration.

8.3.2.1 Provisioning username and password credentials

Figure 33 shows the message exchange sequence for provisioning a username and password to a mobile device using OMA DM. The figure shows the message exchange beginning with the connection to the OSU server.

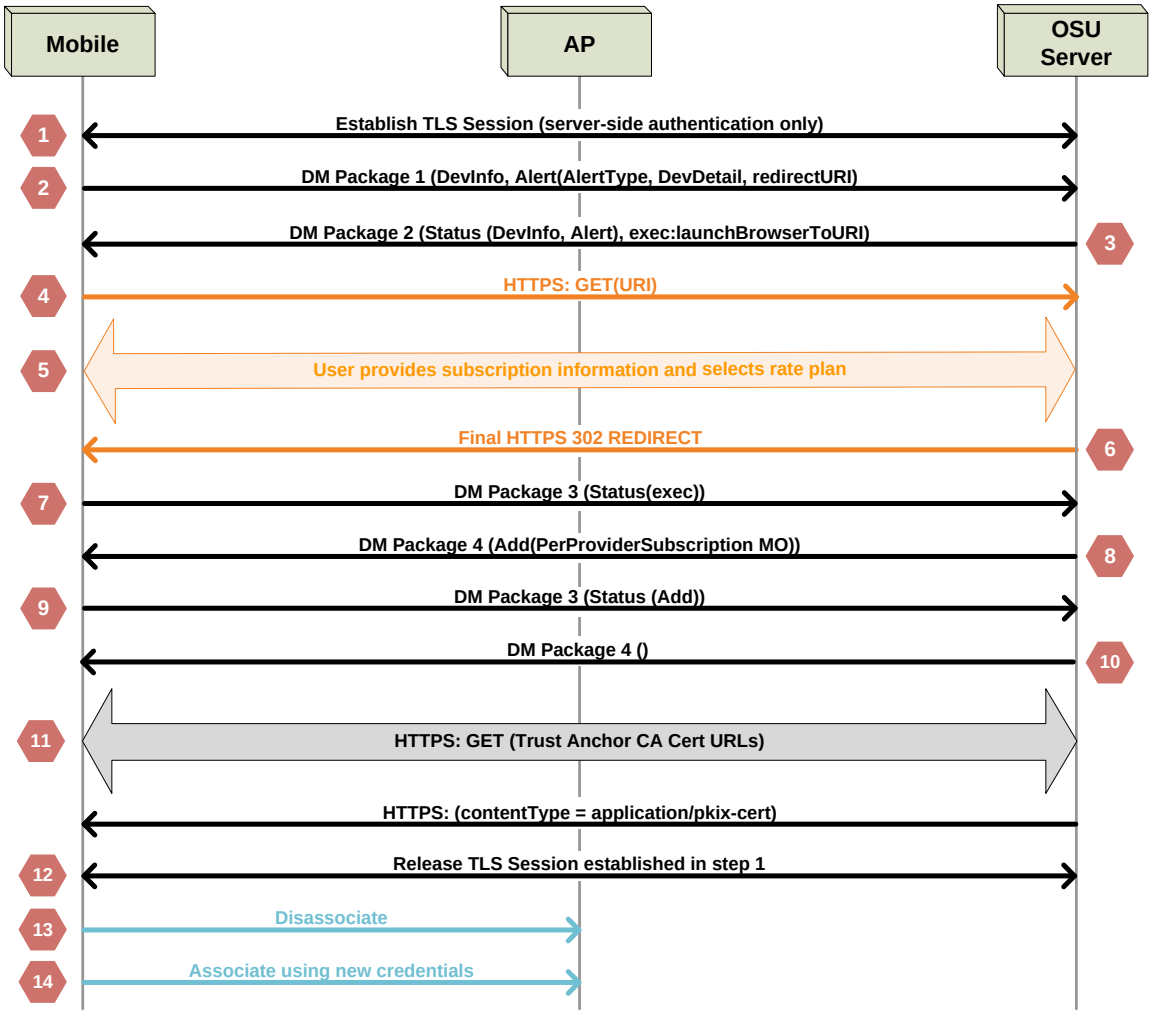


Figure 33: Provisioning username/password credentials and policy using OMA DM

The following are the steps in the online sign up and provisioning process for username/password type credentials.

Step 1: The mobile device initiates a TLS connection to the OSU server in accordance with the procedures in [11], [28] and section 7.5, performing server-side authentication using HTTPS. Note that this step is the one which initiates step 7 in Figure 29.

Separate TCP connections (and TLS sessions) might be used for the OMA DM exchanges that happen prior to and after the user interaction through a web browser.

Step 2: The mobile device sends a DM package 1 (see Figure 66) to the OSU server, containing the DevInfo and DevDetail MOs and a Generic Alert indicating the reason for contacting the OSU server. The Alert Type element in the Generic Alert is set to 'org.wi-fi.hotspot2dot0.SubscriptionCreation', indicating that the user of the mobile device wants to establish a new subscription with this SP. The DevDetail MO Wi-Fi extension Wi-Fi/ClientTriggerRedirectURI includes a redirectURI (see section 9.2). After the OSU server has received all the data needed to complete the registration process, it redirects the mobile device's web browser to the redirectURI (see step 6). The internal event caused by retrieval of the redirectURI resource may be used to signal to the mobile device's connection manager that the registration process is complete and to proceed to the next step in the message sequence.

Step 3: The OSU server may use the information provided in the DevInfo and DevDetail MOs to determine the type of credential to provision username/password or certificate. For this message sequence the OSU server has selected to provision username/password credentials. The OSU server returns a DM package 2 (see Figure 67) to the mobile device with the command Exec on the launchBrowsertoURI node (see Figure 60) with the URI to be used by the browser in the data field. The URI shall contain a 128-bit key value composed of hexadecimal digits generated by the OSU server, which will allow the OSU server to link the browser interaction with the DM session initiated in step 1 (e.g., https://osuserver.example.com/sign-up/smartphone/index.html?key=abcdef0123456789zz9876543210fedcba). This key is only required during the browser interaction and may be discarded after step 7. In addition, the OSU server shall include in DM package 2 a <respURI> element (see section 6.1.17 in [49]); the payload of the <respURI> element contains an absolute URI having a session key which may be used by the OSU server to bind the DM package transmitted by the mobile device in step 7 with the one transmitted in step 2. (The binding is needed if the mobile device uses different TLS/TCP sessions in steps 2 and 7).

After receipt of the command to launch the browser, the mobile device should initiate a listening port of the correct protocol for the redirectURI provided to the OSU server in step 2.

Other message flows in section 8 will also use the redirectURI method to signal completion of user input. However, not all message flows require user input (e.g., see section 8.3.3.1 on machine remediation). Since the mobile device might not know at the beginning of a message sequence whether the OSU server needs to use this method (i.e. the OSU server decides whether user input is required), the mobile device always supplies the redirectURI in the DevDetail MO Wi-Fi extension part, but does not need to open the listening port until receipt of a command to launch a browser.

According to the DM Representation Protocol (see [49]) the DM client (mobile device) shall transmit its next DM package to the URI specified in <respURI>.

The length of the session key parameter in the <respURI> payload depends on the OSU server implementation, but should be at least 128 bits to minimize the possibility of collision of concurrent DM sessions with other mobile devices and the possibility an attacker could successfully guess the session key value and disrupt or break the provisioning process of another mobile device, including the possibility to steal the device's credential.

Step 4: Upon receiving the Exec:LaunchBrowsertoURI command, the mobile device launches the browser[2], establishes a secure HTTPS connection to the URI returned in step 3 and sends an HTTPS GET request to the OSU server URI returned in step 3.

Step 5: The mobile device and OSU server exchange registration data as required by the SP.

In certain circumstances the user might have a subscription that was provisioned using a different mobile device than the one currently in use, or the original mobile device might have been re-imaged, or the user has previously set up an account with the SP (e.g., through a website). In these situations, the subscription has been established, but the mobile device is not in possession of the username/password credential. The message flows of this step shall accommodate this scenario and

[2] For the purposes of this specification, the term "browser" is used to refer to a browser function which is capable of rendering webpages on a mobile device's user interface. It is not meant to imply a standalone application. For example, an implementation could include a browser function in a connection manager.

permit the user to have the mobile device provisioned with the credential bound to the previously established subscription. The data exchanged and webpages used for this purpose are out of scope of this specification.

Step 6: At the end of the subscription provisioning process, the OSU server transmits a final HTTP 302 REDIRECT message to the mobile device which triggers the device to proceed with the next step.

If the TLS connection for the OMA DM session with the OSU server is dropped (intentionally or accidently) during step 4 to 5, the device shall wait for step 6 to trigger a new TLS connection to the OSU server.

Step 7: The mobile device sends a DM package 3 (see Figure 68) to the OSU server containing the status of the previous command execution. If there is an error in the establishment of a new subscription then the mobile device will receive a DM Package 4 (see Figure 71), which indicates the end of the OMA DM transaction.

If the TLS connection for the OMA DM session with the OSU server is dropped (intentionally or accidently) during step 4 to 5, the device shall open a TLS connection with the OSU server and continue with DM package 3.

Step 8: The OSU server sends a DM package 4 to the mobile device, containing the PerProviderSubscription MO and the command ADD, specifying the location of the PerProviderSubscription MO to be added to the OMA DM management tree on the mobile device. The OSU server shall include in DM package 4 a <respURI> element for the same purpose as described in step 3.

Step 9: The mobile device sends a DM Package 3 indicating the status of the previous operation.

When the mobile device transmits an error status, the mobile device shall deem credential provisioning to have failed and shall not attempt to use any credential which may have been received during the failed message sequence.

Step 10: The OSU server sends a DM Package 4 without commands; this indicates the end of the OMA DM session.

Step 11: The mobile device uses HTTPS to retrieve the trust anchor CA certificates for the AAA server, subscription remediation server, and policy server (if needed), using the CertURLs in the PPS MO according to section 7.3.1.

Step 12: The mobile device releases the TLS session that was established in step 1.

Step 13: The mobile device disassociates from the Wi-Fi AN.

Step 14: If the subscription was established successfully in step 7, the mobile device may associate with the new credentials.

8.3.2.2 Provisioning certificate credentials

Figure 34 shows the message exchange sequence for provisioning certificate based credentials using OMA DM.

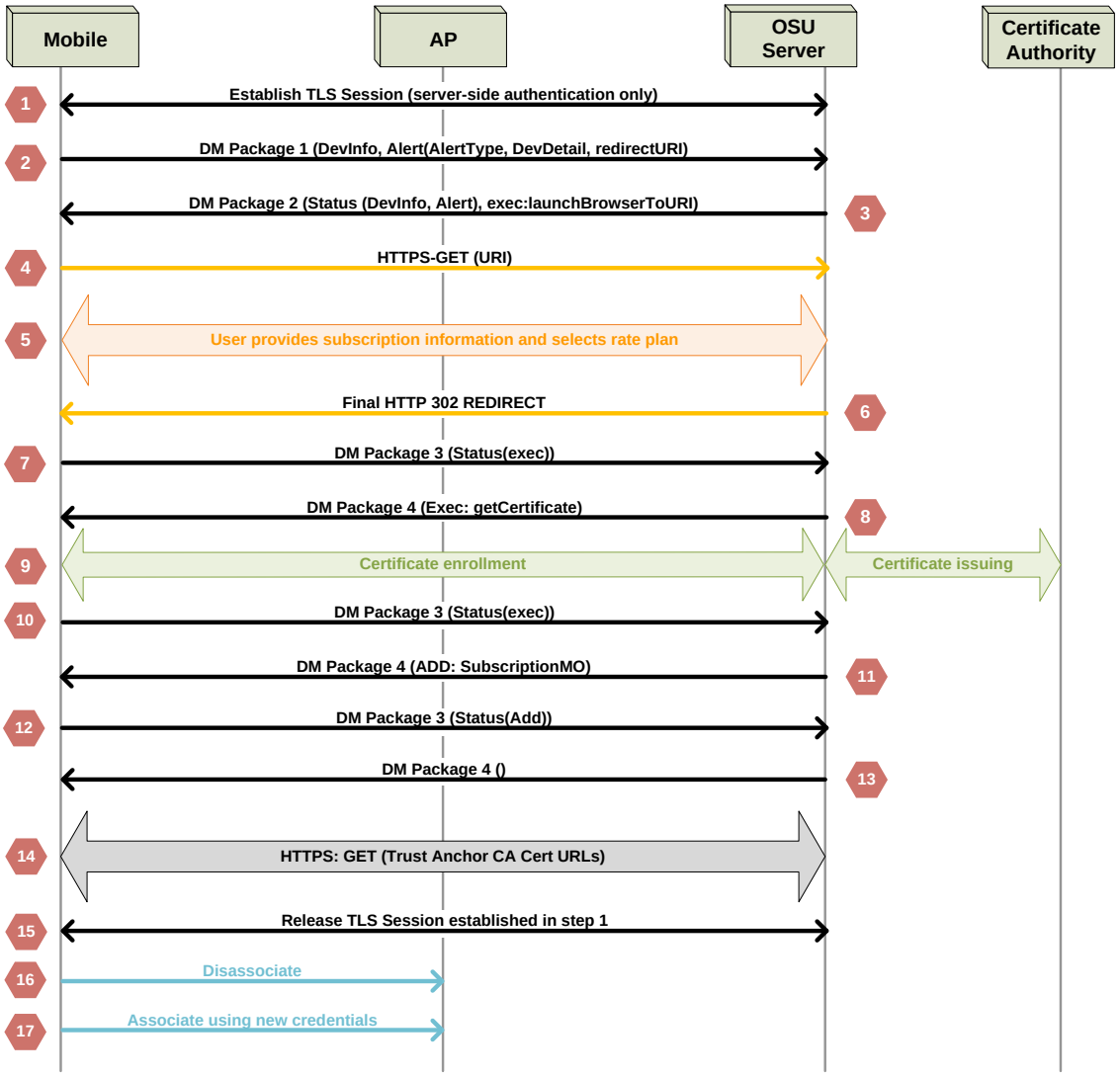


Figure 34: Provisioning certificate credentials and policy using OMA DM

Steps 1 – 7 are identical to those of provisioning username and password credentials above. Note: in step 2, the mobile device provides its unique identity (e.g. IMEI/MEID, Wi-Fi MAC address or the DevInfo MO's DevId) and the OSU server records it for a later use.

Step 8: The OSU server sends a DM package 4 (see Figure 71) to the client with the command Exec:getCertificate. This DM package includes the getCertificate XML instance document (as defined in A.3.5) in the data field. The OSU server shall include in DM package 4 a <respURI> element for the same purpose as described in section 8.3.2.1 step 3.

Step 9: The mobile device performs certificate enrollment according to the procedures in section 7.6. This includes downloading CA certificate(s) that may be used as the AAA server trust anchor (see section 7.3.1).

Step 10: If the certificate enrollment execution was successful, the mobile device sends a DM Package 3 to the OSU server. If there is an error in the certificate enrollment execution, then the mobile device reports the error in DM Package 3 and then in step 11 the OSU server will send a DM Package 4 without further commands causing the mobile to release the connection (and skip Step14).

Step 11: The OSU server sends a DM package 4 to the mobile device, containing the PerProviderSubscription MO and the command ADD, specifying the location of the PerProviderSubscription MO to be added to the OMA DM management tree on the mobile device. The OSU server shall include in DM package 4 a <respURI> element for the same purpose as described in section 8.3.2.1step 3.

Step 12: The mobile device sends a DM Package 3 with the status of the executed command.

Step 13: The OSU server sends a DM Package 4, which indicates the end of the OMA DM transaction.

Step 14: The mobile device uses HTTPS to retrieve the trust anchor CA certificates for the subscription remediation server, policy server, and the AAA server (if needed) using the CertURLs in the PPS MO according to section 7.3.1.

Step 15: The mobile device releases the TLS session that was established in step 1.

Step 16: The mobile device disassociates from the network.

Step 17: If the subscription was established successfully in Step 7, the mobile device re-associates using the credentials obtained in step 9.

8.3.2.3 Provisioning using mobile device provided certificates

Some SPs may elect to use pre-provisioned client certificates. Pre-provisioned client certificates include certificates which have been provisioned by an SP or the manufacturer. The methods and protocols by which these certificates are provisioned to a mobile device are outside the scope of this specification. Figure 35 shows the message exchange sequence for the negotiation of client certificates.

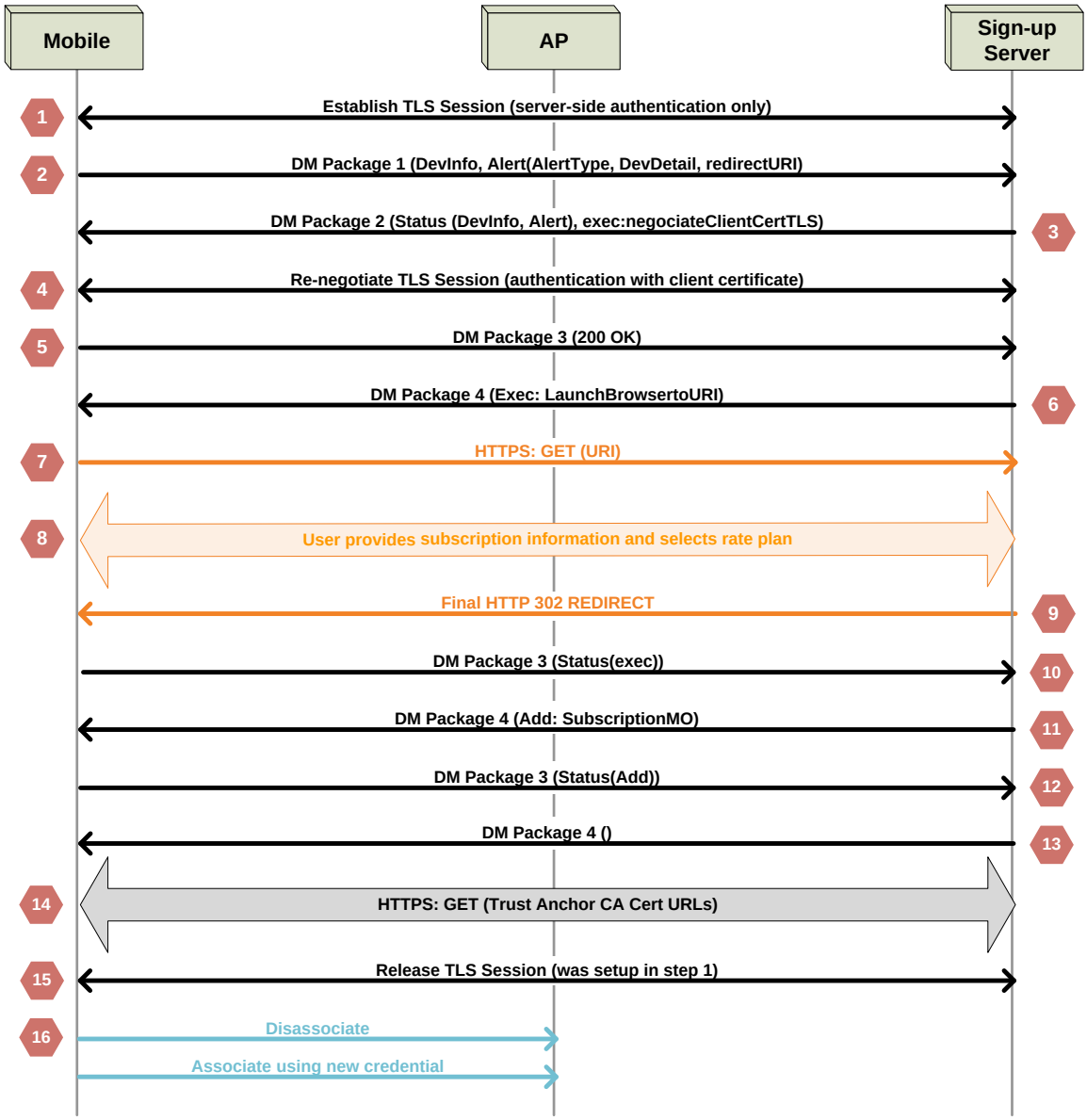


Figure 35: Message exchange diagram for negotiating client certificates using OMA DM

The description for each numbered message in Figure 35 is:

Step 1: Same as step 1 in section 8.3.2.2.

Step 2: Same as step 2 in section 8.3.2.2.

Note: in the DevDetail MO, there is a vendor-specific extension (see section 9.2.1) indicating whether the mobile device is in possession of an IEEE 802.1ar manufacturing certificate or other Service Provider issued certificate(s). [58]

Step 3: Same as step 3 in section 8.3.2.2, except the exec command is negotiateClientCertTLS.

The negotiateClientCertTLS command requests the mobile device to re-negotiate the TLS session using a client certificate. The data element of the Exec includes the useClientTLS element specified in section A.3.2 which contains information on the certificate types and issuers that are regarded as acceptable.

Step 4: If the mobile device possesses certificates acceptable to the OSU server, it renegotiates the TLS connection to the OSU server in accordance with the procedures in [31] using a client certificate. If the mobile device has more than one certificate meeting the criteria in the negotiateClientCertTLS message, then TLS negotiation is used to determine a mutually acceptable certificate. In case the mobile device and server are unable to negotiate to a mutually acceptable client certificate[3], the mobile device shall continue using the TLS connection with server-side only authentication.

Step 5: The mobile device sends a DM Package 3 message indicating the status of the previous operation.

Steps 6 to 16: Same as steps 3 to 13 respectively in section 8.3.2.2.

8.3.3 Subscription management

Subscription management refers to remediation of credential problems (e.g., password expiration), remediation of account problems (e.g., user did not pay their bill) or the update of subscription provisioning information (e.g., update of a roaming consortium OI).

This section describes subscription management using OMA DM for the following cases:

1. Machine remediation when the mobile device has a Username and Password (see section 8.3.3.1)
2. User remediation when the mobile device has Username and Password credentials (see section 8.3.3.2)
3. Machine remediation when the mobile device has certificate credentials (see section 0)
4. User remediation when a mobile device has certificate credentials (see section 8.3.3.3)
5. Updating a subscription using certificate credentials (see sections 8.3.3.4 and 8.3.3.5)

The subscription remediation server shall request authentication for the realm provided to the user in /PerProviderSubscription/<X+>/Credential/Realm. The subscription remediation server knows the realm value as the server is required to maintain a 1 to 1 relationship with that realm and the URI provided in /PerProviderSubscription/<X+>/SubscriptionUpdate/URI.

The OSU server should use the registration protocol defined in Annex E to embed standard tags, if applicable, for user remediation.

8.3.3.1 Machine remediation when a mobile device has username and password credentials

Figure 36 shows the message exchange sequence for machine remediation (renewal) of a subscription when the mobile device is in possession of a username and password credential.

[3] This should be a rare event since the mobile device should only be attempting to use client certificates which the OSU server has indicated are acceptable.

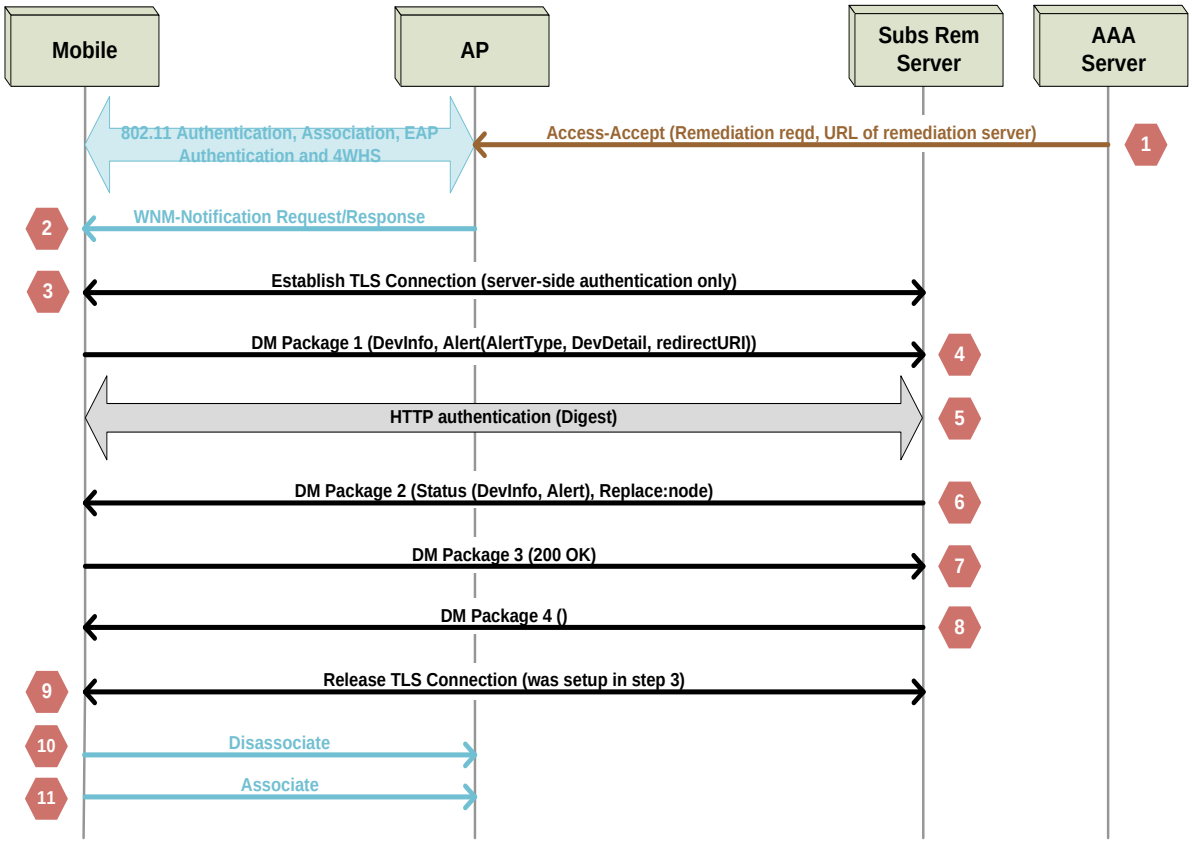


Figure 36: Message exchange diagram for machine remediation of username and password credentials

The description for each numbered message in Figure 36 is:

Step 1: During the authentication process, when a SP determines that subscription remediation (see section 3.2.1) is needed for a user, the AAA server sends an “Access-Accept” with remediation indication (see Annex D) to the authenticator (assumed to be co-located with the AP in Figure 36) at the end of the mobile device’s EAP authentication sequence.

If the AAA server signals that remediation is required, the AP shall not cache the PMKSA. (This forces the mobile device to perform a full EAP authentication in step 11, thereby removing any access restrictions enforced in step 1.)

Note: The URL is present in the Access-Accept message to support, for example, the upgrade of the mobile device from Release 1 to Release 2.

Step 2: After the completion of the four way handshake, the authenticator shall cause the AP (with which the mobile device authenticated) to transmit a WNM-Notification request action frame, indicating the need for subscription remediation, to the mobile device.

The WNM-Notification Request frame is a robust action frame because the use of PMF is required. The mobile device shall only use the URL contained in the WNM-Notification Request frame when it is attached to a hotspot operated by the Home SP.

The mobile device in possession of a PerProviderSubscription MO shall ignore the provisioning protocol method and URL contained in the WNM-Notification Request frame. The mobile device having a SIM credential but not in possession of a PerProviderSubscription MO can use this URL and provisioning protocol method to enable provisioning of a PerProviderSubscription MO, as defined in section 8.5.

Step 3: The mobile device initiates a TLS connection to the subscription remediation server in accordance with the procedures in [29], with server-side authentication only between the mobile device and server. The mobile device shall validate the subscription remediation server certificate according to the procedures in section 7.3.4.2. If the mobile device is unable to initiate a TLS connection to the subscription remediation server, it shall abort the remediation process. The mobile device shall not attempt subscription remediation using HTTP (i.e., instead of HTTPS).

Note: separate TCP connections (and TLS sessions) can be used for the OMA DM exchanges.

Step 4: The mobile device sends a DM Package 1 (see Figure 66) to the server, with the Alert Type element in the Generic Alert set to “org.wi-fi.hotspot2dot0.SubscriptionRemediation” and including the DevInfo and DevDetail MOs.

Step 5: The subscription remediation server shall request HTTP authentication using the Digest method in accordance with the procedures in [9]. The mobile device shall provide its username and password digest to the server in accordance with [9].

If the UsernamePassword node located at PerProviderSubscription/<X+>/SubscriptionUpdate/UsernamePassword is present, the credential contained in the UsernamePassword node shall be used to authenticate to the server. If HTTP authentication is not successful, remediation is not possible and the mobile device shall abort the process and should inform the user[4].

If the subscription remediation server and mobile device both support HTTP cookies, then cookies can be used to avoid the need for another HTTP Digest Authentication. Otherwise, the subscription remediation server shall cause another HTTP Digest Authentication with the same username and password used in step 5 to be carried out prior to step 7.

Step 6: The server responds with DM Package 2 (see Figure 67), including a Replace command for one or more interior nodes of the PerProviderSubscription MO (see section 9.1).

In addition, the OSU server shall include in DM package 2 a <respURI> element (see section 6.1.17 in [53]); the payload of the <respURI> element contains an absolute URI having a session key which can be used by the OSU server to bind the DM package transmitted by the mobile device in step 7 with the one transmitted in step 6, regardless of whether a single or multiple TLS/TCP sessions are used.

According to the DM Representation Protocol (see [49]) the DM client (mobile device) shall transmit its next DM package to the URI specified in <respURI>.

The length of the session key parameter in the <respURI> payload depends on the OSU server implementation, but should be at least 128 bits to minimize the possibility of collision of concurrent DM sessions with other mobile devices and the possibility an attacker could successfully guess the session key value and disrupt or break the provisioning process of another mobile device, including the possibility to steal the device’s credential.

Step 7: The mobile device performs the instructions received and responds with DM Package 3 to the server with the result of the operation (e.g., “200 OK”, indicating successful operation). If a Trust Root node is replaced, the mobile device shall download the new trust anchor CA certificate(s) and replace the old certificates with new certificates.

Step 8: The server closes the DM session by sending a DM Package 4 (see Figure 71) with Final to the mobile device and releasing the TLS connection.

Step 9: The mobile device releases the TLS connection established in step 3.

Step 10: The mobile device disassociates from the AP.

[4] The user should be informed because the AN may be restricting access such that the subscription may not be useable for their purpose.

Step 11: The mobile device associates again to gain network access.

8.3.3.2 User remediation when a mobile device has username and password

Figure 37 shows the message exchange sequence for user remediation of a subscription when the mobile device is in possession of a username and password credential and needs user intervention for remediation.

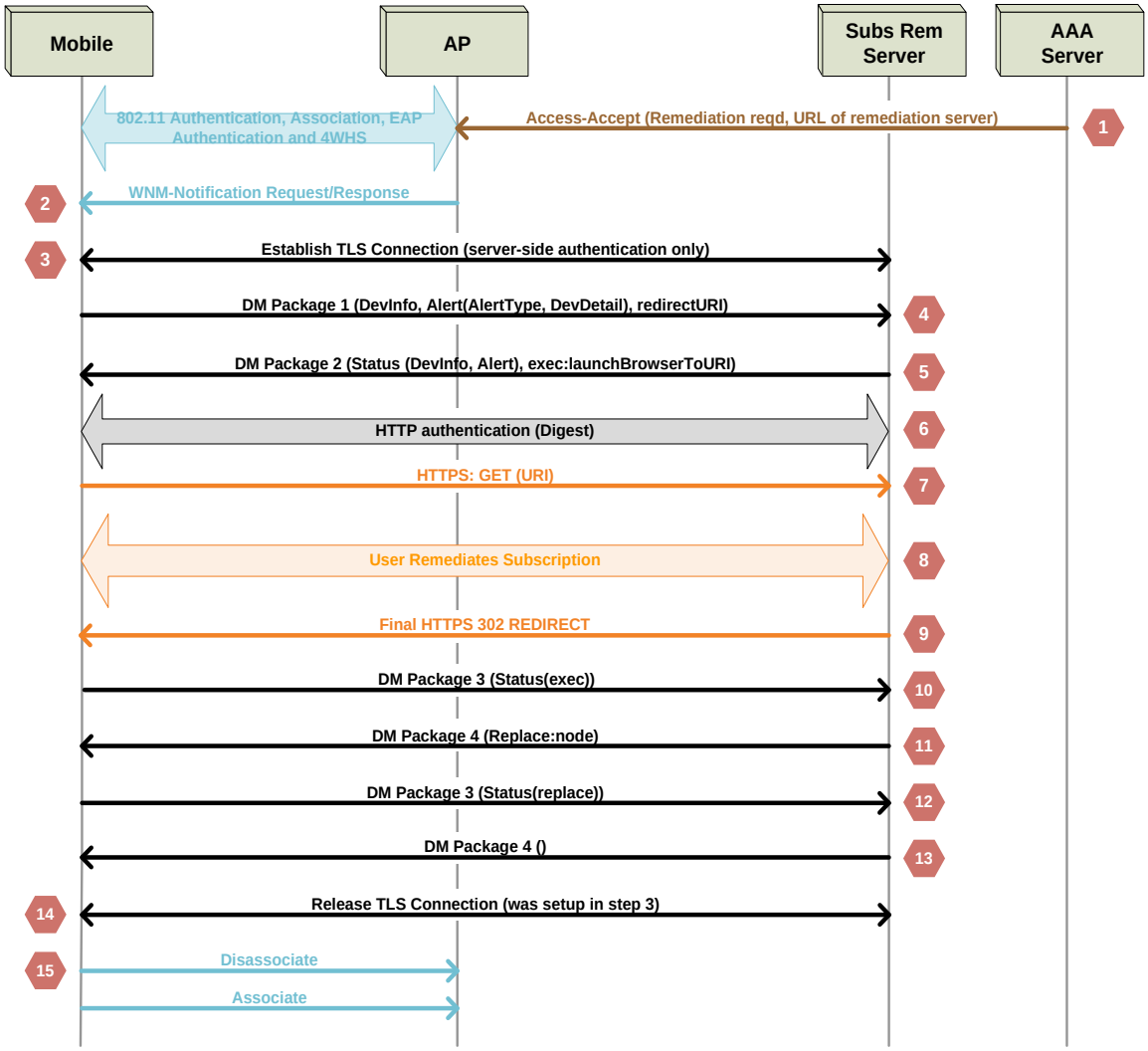


Figure 37: Message exchange diagram for user remediation of username and password credentials

The description for each numbered message in Figure 37:
Steps 1 – 5 are identical to those in section 8.3.3.1.
Step 6 is identical to step 3 of 8.3.2.1 with the URI of the subscription remediation server.
Steps 7 to 10 are identical to steps 4 to 7 of section 8.3.2.1.
Steps 11 – 15 are identical to steps 6 – 10 in section 8.3.3.1.

Machine remediation when a mobile device has certificate credentialsFigure 38 shows the message exchange sequence for machine remediation of a certificate credential (i.e., certificate re-enrollment). If the PerProviderSubscription/<X+>/SubscriptionUpdate/UsernamePassword is present, the credential contained in that node shall be used to authenticate to the server; in this situation, the procedures in section 8.3.3.1 are used.

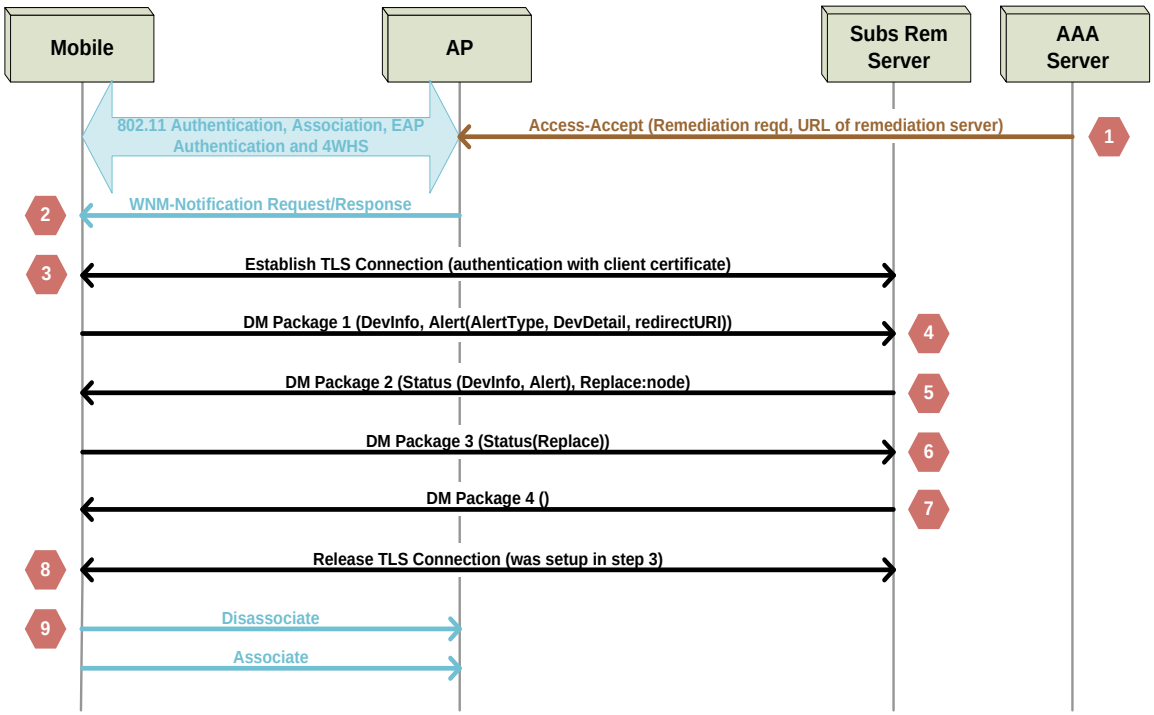


Figure 38: Message exchange diagram for machine remediation of certificate credentials

Steps 1 – 2 are identical to those in section 8.3.3.1.

Step 3: The mobile device initiates a TLS connection to the subscription remediation server in accordance with the procedures in [11] and [28]. As the mobile device has a certificate credential, it uses that certificate for TLS authentication. The mobile device shall validate the subscription remediation server certificate according to the procedures in section 7.3.5.2.

If the mobile device is unable to initiate a TLS connection to the subscription remediation server, it shall abort the remediation process. The mobile device shall not attempt subscription remediation using HTTP (i.e., instead of HTTPS).

Note: separate TCP connections (and TLS sessions) can be used for the OMA DM exchanges.

Step 4 is identical to that in section 8.3.3.1.

Steps 5 – 9 are the same as steps 6 – 10 in section 8.3.3.1.

8.3.3.3 User remediation when a mobile device has certificate credentials

Figure 39 shows the message exchange sequence for user remediation of a subscription when the mobile device is in possession of a certificate credential. If the UsernamePassword node located at PerProviderSubscription/<X+>/SubscriptionUpdate/UsernamePassword is present, the username/password credential contained in the UsernamePassword node shall be used to authenticate to the server; the procedures specified in section 8.3.3.2 are used for this case.

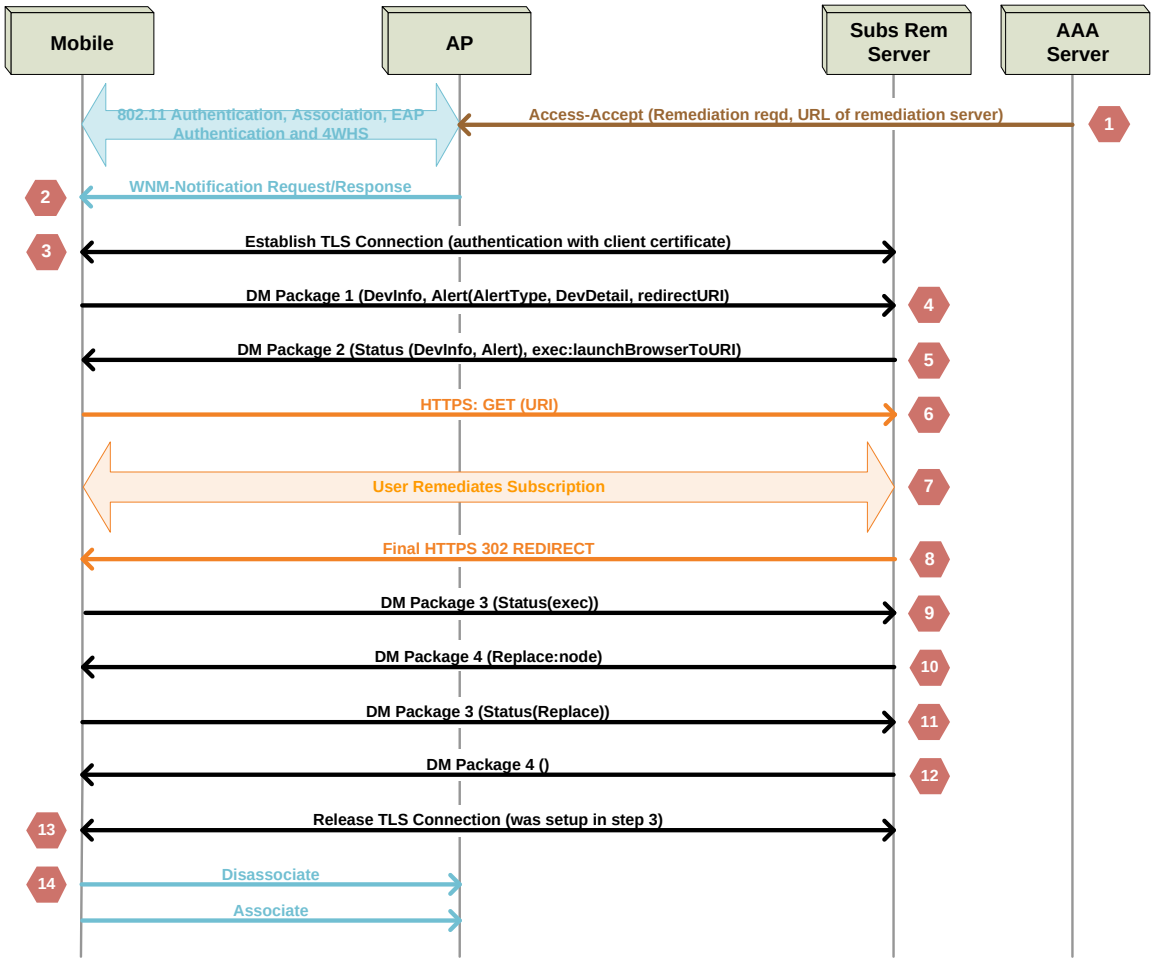


Figure 39: Message exchange diagram for user remediation of certificate credentials

Steps 1 - 4 are identical to those in section 0.

Steps 5 - 14 are the same as steps 6 - 15 respectively in section 8.3.3.2.

8.3.3.4 Updating a certificate credential

Figure 40 shows the message exchange sequence for machine remediation of a certificate credential (i.e., certificate re-enrollment). If the UsernamePassword node located at PerProviderSubscription/<X+>/SubscriptionUpdate/UsernamePassword is present, the credential contained in the UsernamePassword node shall be used to authenticate to the server; the procedures specified in section 8.3.3.5 are used for this case.

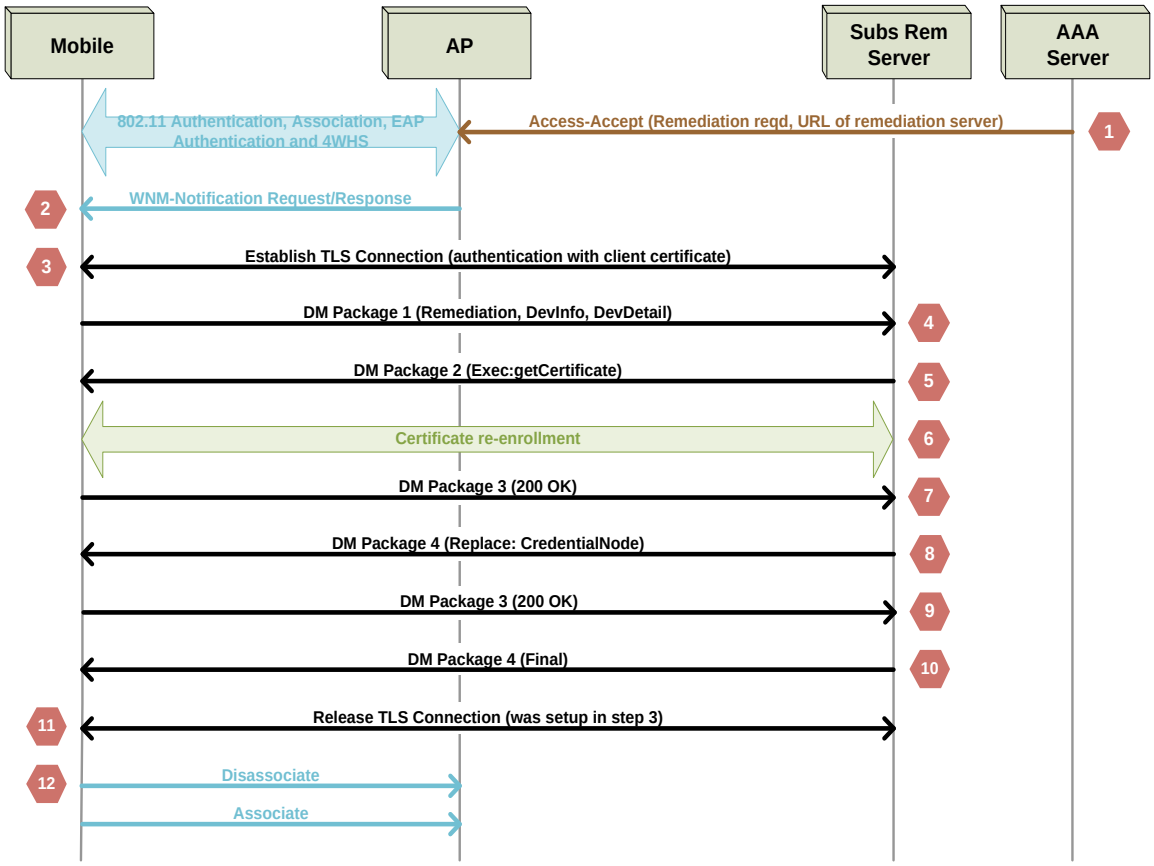


Figure 40: Message exchange diagram for updating certificate credentials

Steps 1 - 4 are identical to those in section 0.

Steps 5 - 6 are the same as steps 8 - 9 in section 8.3.2.2, except that in step 6, the HTTPS POST in EST shall request re-enrollment (see section 4.2.2 in [36]).

Steps 7 - 12 are the same as steps 9 - 14 in section 8.3.3.3.

8.3.3.5 Updating a certificate credential when the UsernamePassword node is present

Figure 41 shows the message exchange sequence for machine remediation of a certificate credential (i.e., certificate re-enrollment), when the UsernamePassword node is present in the PerProviderSubscription/<X+>/SubscriptionUpdate node in the PerProviderSubscription MO.

This message sequence differs from that of section 8.3.3.4, because the credential used to authenticate to the subscription remediation server is drawn from the information contained in that UsernamePassword node.

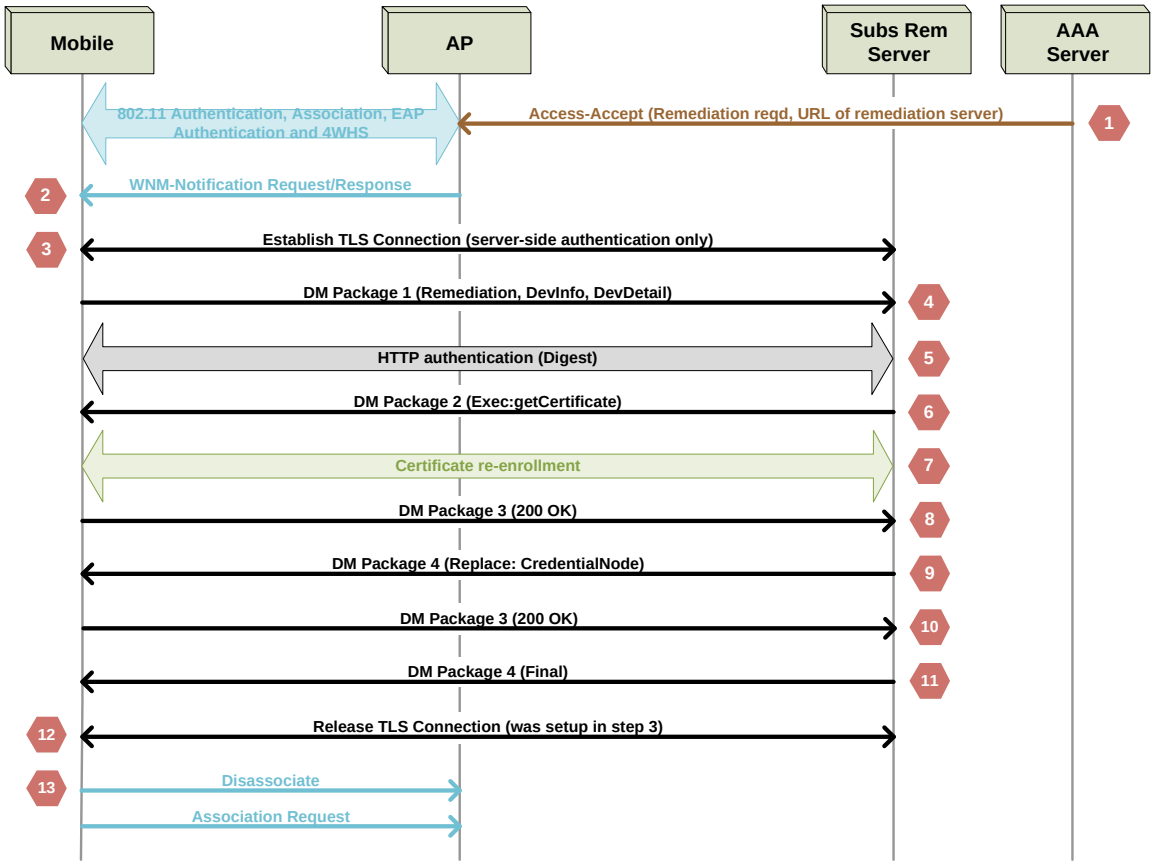


Figure 41: Message exchange diagram for updating certificate credentials

Steps 1 – 5 are identical to those in section 8.3.3.1.

Steps 6 – 13 are the same as steps 5 - 12 in section 8.3.3.4.

8.3.4 Policy provisioning

This section describes the procedures for the mobile device to obtain the Home SP policy from an authorized server over the OMA DM protocol. HTTPS transport is used in this specification. The mobile device shall obtain the Home SP policy from a policy server via a Wi-Fi network connection.

The Generic Alert mechanism specified in the OMA DM protocol specification [39] enables the mobile device to check with the server for policy updates. The mobile device may send a standard Generic Alert message to the server determined by the 'PerProviderSubscription/Policy/PolicyUpdate/URI'. The server may request further information in the same OMA DM session and decide if an update is needed. The server may send commands to the mobile device in the same session.

If the server finds an update to the 'PerProviderSubscription/Policy' node for the mobile device, the server shall send replace commands to the mobile device to update interior nodes from the 'PerProviderSubscription/Policy' subtree. The server shall include the status code "'200', OK" for the Generic Alert. If the server cannot find an update to the 'PerProviderSubscription/Policy' node, the server shall not send further commands, but shall send the response "'202', Accepted for Processing".

The elements in the Generic Alerts shall be as follows (see section 8.7.1 in [39]):

- Alert Type: 'org.wi-fi.hotspot2dot0.PolicyUpdate'.

- LocURI: The URI of the policy node subtree ('./Wi-Fi/<SP FQDN>/PerProviderSubscription/Policy'). The server can find out if an update is necessary for the MO identified by the LocURI.
- Data: This element shall be included. The mobile device vendor may include implementation specific data. Policy servers may ignore the contents of the data element.

8.3.4.1 Policy provisioning and update with username and password credentials

Figure 42 shows the message exchange sequence for SP policy provisioning and update when the mobile device is in possession of a username and password credential. Section A.2 specifies the OMA DM messages listed in the figure.

If the UsernamePassword node located at PerProviderSubscription/<X+>/PolicyUpdate/UsernamePassword is present, the username/password credential contained in that node shall be used to authenticate to the server.

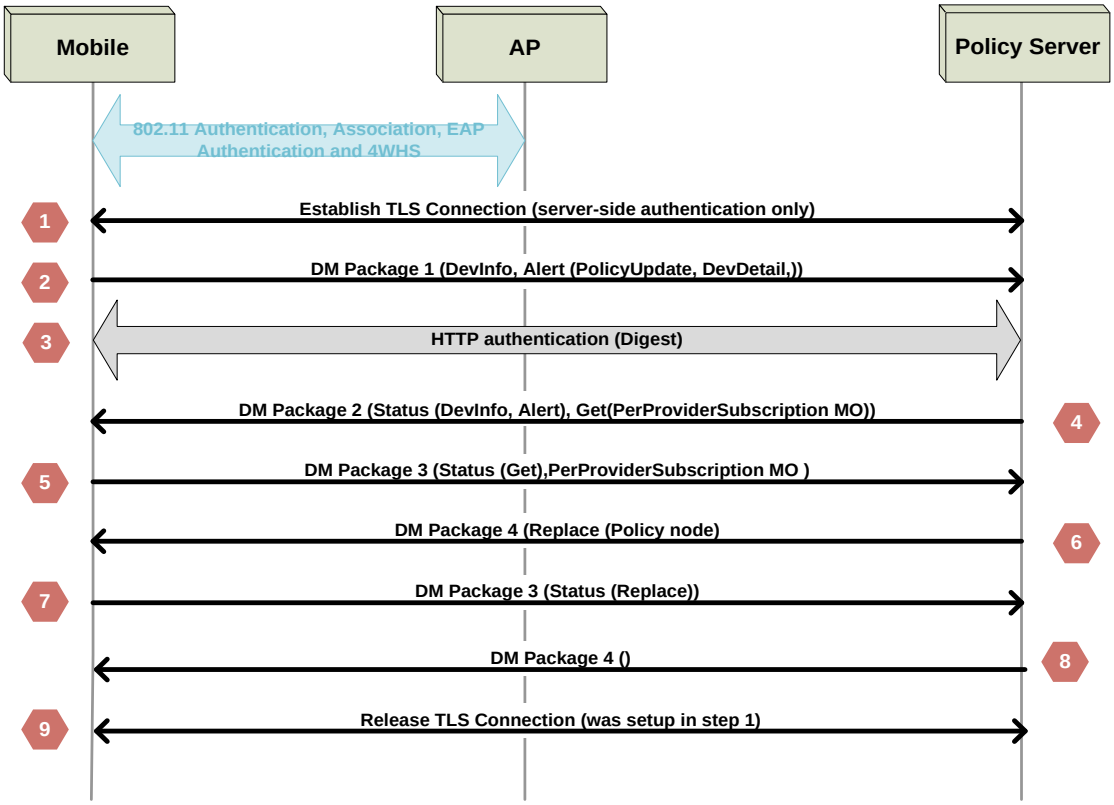


Figure 42: Message sequence diagram for SP policy provisioning and update when the mobile device has username and password credentials

The description for each numbered message in Figure 42 is:

Step 1: The mobile device initiates a TLS connection to the policy server in accordance with the procedures in [28]. The URI of the policy server is provided in the PerProviderSubscription MO. The mobile device shall validate the policy server certificate according to the procedures in section 7.3.6.2. If the mobile device is unable to initiate a TLS connection to the policy server, it shall abort the policy update process. The mobile device shall not attempt connection to the policy server using HTTP (i.e., instead of HTTPS).

Step 2: The mobile device sends a DM package 1 to the OSU server, containing the DevInfo and DevDetail MOs and a Generic Alert indicating the reason for contacting the OSU server. The Alert Type element in the Generic Alert shall be set to 'org.wi-fi.hotspot2dot0.PolicyUpdate',

Step 3: The policy server shall request HTTP authentication using the digest method in accordance with [9]. The mobile device shall provide its username and password digest to the server in accordance with [9]. If HTTP authentication is not successful, policy download is not possible and the mobile device shall abort the process.

Step 4 (optional for the policy server): The policy server may send a get command in package 2 to upload the PerProviderSubscription MO (see section 9.1).

In addition, the policy server shall include in DM package 2 a <respURI> element (see section 6.1.17 in [53]); the payload of the <respURI> element contains an absolute URI having a session key which can be used by the policy server to bind the DM package transmitted by the mobile device in step 5 with the one transmitted in step 2, regardless of whether a single or multiple TLS/TCP sessions are used.

Note 1: the DM Representation Protocol (see [49]) requires the DM client (mobile device) to transmit its next DM package to the URI specified in <respURI>.

Note 2: the length of the session key parameter in the <respURI> payload depends on the policy server implementation, but should be at least 128 bits to minimize the possibility of collision of concurrent DM sessions with other mobile devices.

Step 5 (not performed by mobile device unless step 4 is used by the policy server to upload the PPS MO): The mobile device transmits the PerProviderSubscription MO in package 3.

Step 6: If the policy server finds a need to update the 'PerProviderSubscription/Policy' node, it shall respond with DM package 4, including a Replace command for one or more interior nodes of the policy subtree of the PerProviderSubscription MO (see section 9.1) and including a <respURI> element as described in step 4. If the server cannot find an update to the 'PerProviderSubscription/Policy' node, the server will not send further commands as in step 8 and step 9.

Step 7: The mobile device performs the instructions received and responds back to the server with the result of the replace operation (e.g., "200 OK"), indicating successful operation. If the TrustRoot node is replaced the mobile device shall download the new trust anchor CA certificate and replace the old one with the new one.

Step 8: If no error occurred in step 5, the server closes the DM session by sending a DM Package 4 with no command to the mobile device.

Step 9: The mobile device releases the TLS session it established in step 1.

8.3.4.2 Policy provisioning and update with certificate credentials

Figure 43 shows the message exchange sequence for SP policy provisioning and update when the mobile device is in possession of a certificate credential. Section A.2 specifies the OMA DM messages listed in the figure.

If the UsernamePassword node located at PerProviderSubscription/<X+>/PolicyUpdate/UsernamePassword is present, the username/password credential contained in the UsernamePassword node shall be used to authenticate to the server using the message sequence shown in Figure 42.

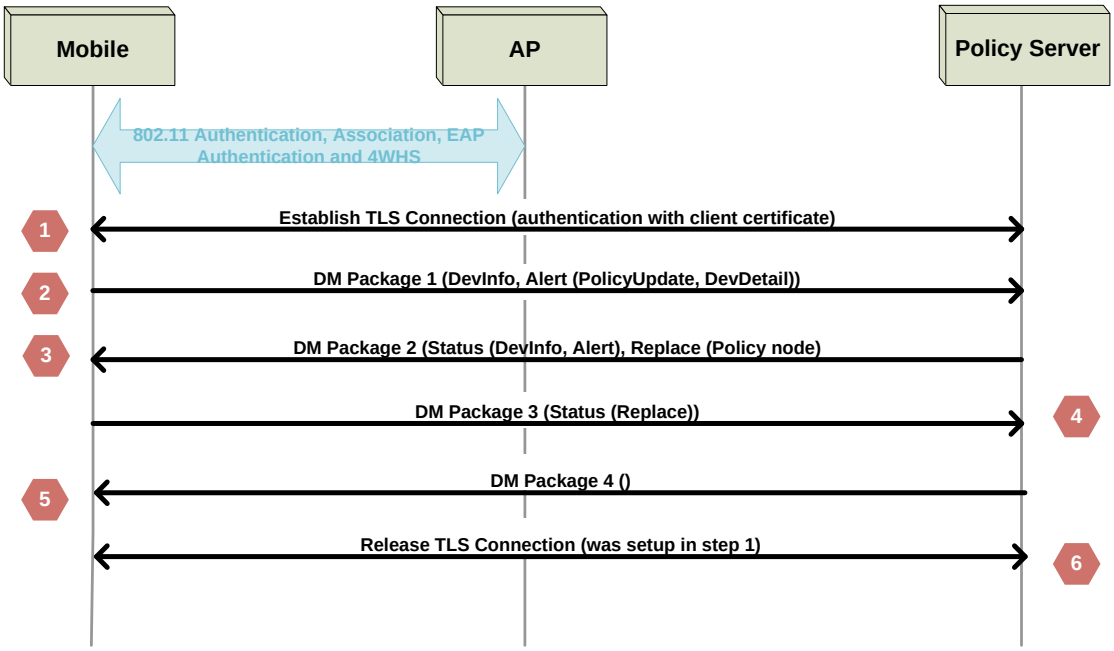


Figure 43: Message sequence diagram for SP policy provisioning and update when the mobile device has certificate credentials

The description for each numbered message in Figure 43 is:

Step 1: The mobile device initiates a TLS connection to the policy server in accordance with the procedures in [28]. The URI of the policy server is provided in the PerProviderSubscription MO. As the mobile device has a certificate credential, it uses that certificate for TLS authentication.

Step 2 is identical to step 2 in section 8.3.4.1.

Step 3: The policy server responds with DM package 2, including a Replace command for one or more interior nodes of the policy interior node of the PerProviderSubscription MO (see section 9.1). In addition, the policy server shall include in DM package 2 a <respURI> element (see section 6.1.17 in [53]); the payload of the <respURI> element contains an absolute URI having a session key which can be used by the policy server to bind the DM package transmitted by the mobile device in step 5 with the one transmitted in step 2, regardless of whether a single or multiple TLS/TCP sessions are used.

Note 1: the DM Representation Protocol (see [49]) requires the DM client (mobile device) to transmit its next DM package to the URI specified in <respURI>.

Note 2: the length of the session key parameter in the <respURI> payload depends on the policy server implementation, but should be at least 128 bits to minimize the possibility of collision of concurrent DM sessions with other mobile devices.

Steps 4 to 6 are identical to those in steps 7 to 9 in section 8.3.4.1.

8.4 Provisioning using SOAP XML

8.4.1 Overview

This section describes procedures that enable SPP (subscription provisioning protocol), a protocol based upon SOAP XML methods (see section A.3), to implement a subset of OMA DM functions in order to provision and manage the OMA DM compliant PerProviderSubscription MO in a mobile device's management tree [51].

The sections are as follows:

1. Procedures for SOAP XML provisioning are given in section 8.4.2.
2. Procedures for SOAP XML subscription management are given in section 8.4.3.
3. Procedures for SP policy provisioning and update are given in section 8.4.4.
4. SOAP XML message definitions are given in Annex A.

The OSU server should use the registration protocol defined in Annex E to embed standard tags for registration.

Figure 44 shows an overview of the SOAP XML credential provisioning and subscription management message exchange framework.

Note: the SOAP XML / OMA DM protocol negotiation is not included in Figure 44.

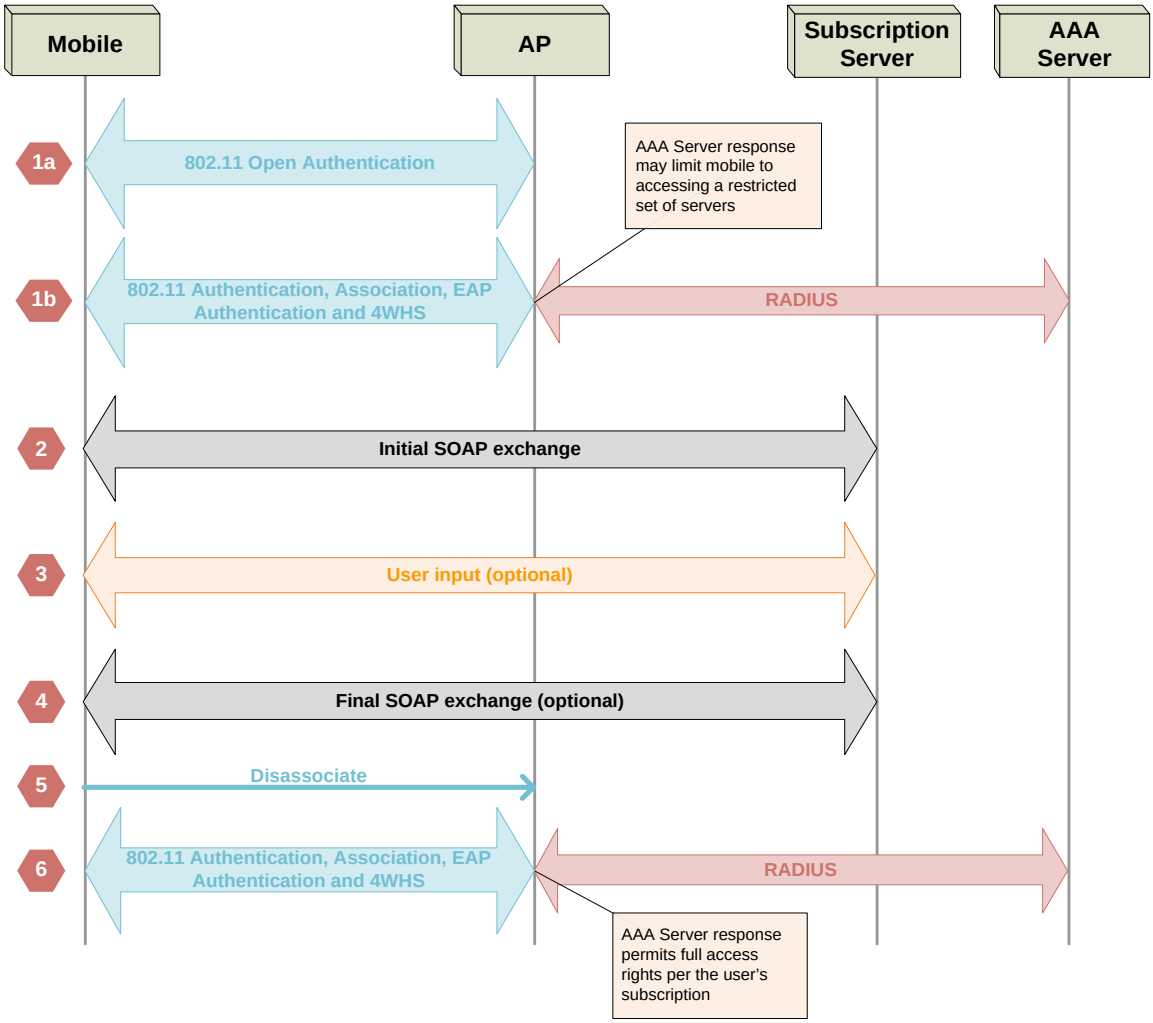


Figure 44: Message exchange framework for credential provisioning and subscription management using SOAP XML

The description for each numbered message flow[5] in Figure 44 is:

[5] The term "message flow" is used to mean there can be one or more messages sent in either direction to carry out a particular function.

Step 1a: If the AP uses open authentication, the mobile device initiates the credential provisioning process by associating to the Wi-Fi AN, using open authentication as described in section 8.4.2.

Step 1b: If the AP uses secure authentication, the mobile device initiates the credential provisioning process by associating to the Wi-Fi AN as described in section 7.4. The AP relays EAP messages to and from the AAA server. At the end of a successful EAP exchange culminating in an Access-Accept RADIUS message, the AAA server delivers access restrictions to the AP, which restricts the mobile device's connectivity to the subscription servers (see section 8.1 and Figure 31). If the AAA server signals remediation is required, the AP shall not cache the PMKSA. Note: this forces the mobile device to perform a full EAP authentication in step 6, thereby ensuring that any access restrictions enforced in step 1b are removed.

Step 2: The initial SOAP exchange is used by the mobile device to authenticate the Subscription server, to provide some mobile device capability information to the server and to inform the server of its request reason (e.g., credential provisioning, subscription remediation). When the mobile device already has credentials, as is the case for subscription remediation, the server will authenticate the mobile device.

Upon receipt of this information, the subscription server decides what to do next. If user input is required, then the subscription server informs the mobile device to launch a browser. If machine-only messages are needed, the server's response contains an indication that the message flow is complete or that additional machine exchanges are needed. Messages in this flow carry an HTTP content type of "application/soap+xml". Thus, they are delivered to the SOAP processing application in both the mobile device and subscription server. If the mobile device has lost its credentials (e.g., the mobile device was re-imaged) then the mobile device shall use the credential provisioning message sequences to re-acquire its credentials.

Step 3: In cases where user input is required (e.g., to sign up for service or to pay a bill), the mobile device launches a browser and the user is prompted for information via webpages. The content and number of these webpages is outside the scope of this specification. At the end of the user exchange, the message flows may be completed or further messages may be required; in either case, the Subscription server informs the mobile device the message exchange is complete or alternatively identifies the next step in the process.

Note that messages in this flow carry HTTP content types used for webpages. This includes, but is not limited to "text/html", "application/xml" and "application/xhtml+xml". Thus, they are delivered to the webserver processing application in both the mobile device and subscription server.

Step 4: The final SOAP exchange completes the credential provisioning process. This is a machine to machine communication (no user input); a final SOAP exchange is used when provisioning certificates (see section 8.4.2.2). Messages in this flow carry an HTTP content type of "application/soap+xml". Thus, they should be delivered to the SOAP processing application in both the mobile device and subscription server.

Step 5: The mobile device disassociates from the Wi-Fi AN.

Step 6: The mobile device associates and authenticates to the Wi-Fi AN. The AP relays EAP messages to and from the AAA server. At the end of a successful EAP exchange culminating in an Access-Accept RADIUS message, the AAA server deliver to the AP access restrictions, if applicable, according to the user's subscription. This removes access restrictions, if any, that were instantiated for credential provisioning or subscription remediation.

It should be noted that an SP's subscription servers may exchange network-to-network infrastructure messages as part of implementing the functionality required by the above framework and subsequent message flows. These network-to-network interfaces and messages are outside the scope of this specification.

8.4.2 Subscription provisioning

8.4.2.1 Provisioning username and password credentials

Figure 45 shows the message exchange sequence for provisioning a username and password to a mobile device using SPP. The figure shows the message exchange beginning with connection to the OSU server. Section A.3 specifies the SOAP XML methods listed in the figure.

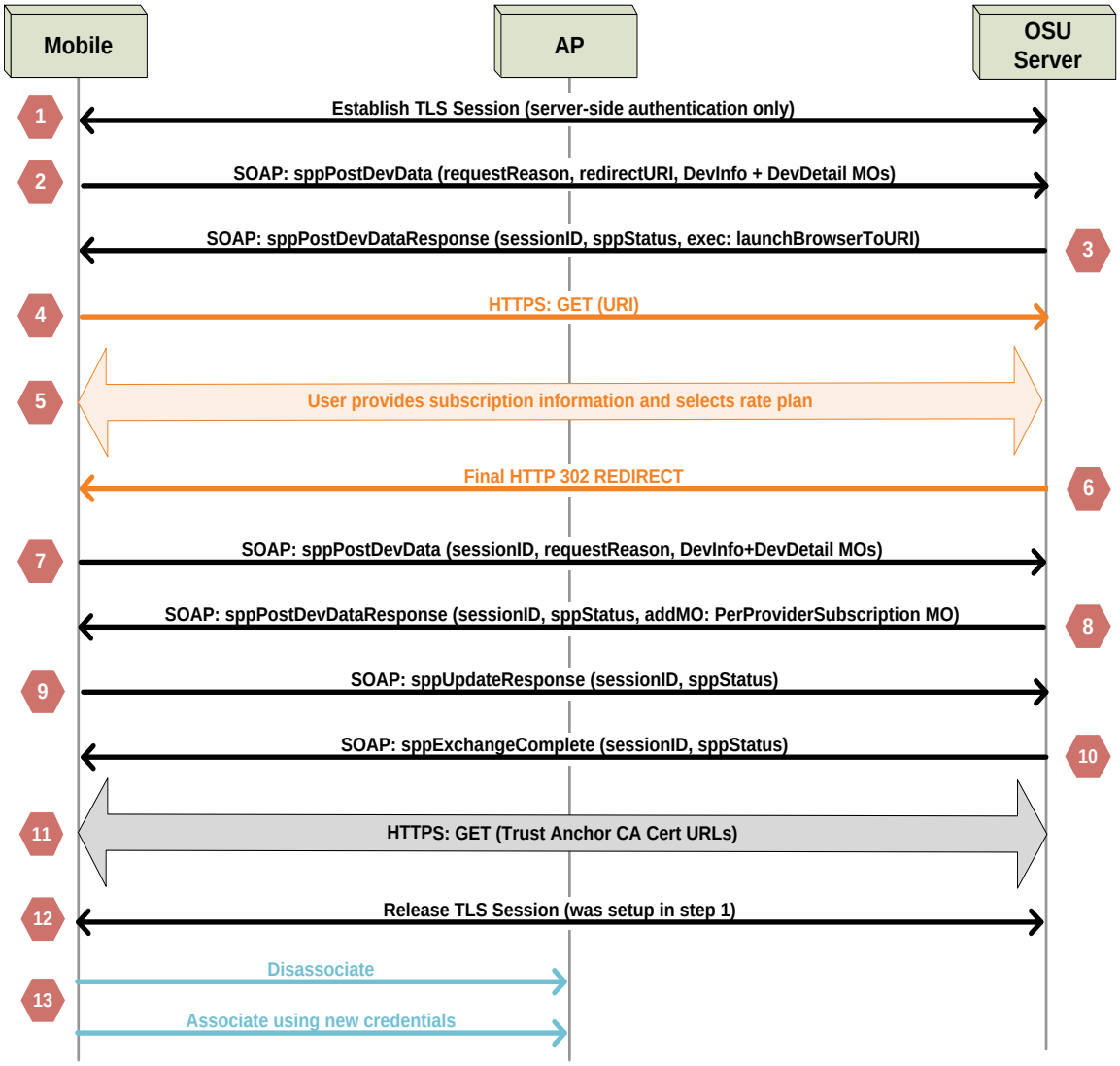


Figure 45: Message exchange diagram for username and password credential provisioning using SOAP XML

The description for each numbered message in Figure 45 is:

Step 1: The mobile device initiates a TLS connection to the OSU server in accordance with the procedures in [11], [28] and section 7.5, performing server-side authentication using HTTPS. Note that this step is the one which initiates step 7 in Figure 29.

Note: separate TCP connections (and TLS sessions) can be used for the SOAP XML exchanges that happen prior to and after the user interaction via a web browser.

Step 2: The mobile device transmits sppPostDevData SOAP method to the server which includes the DevInfo and DevDetail MOs. The value for requestReason is set to “Subscription registration” indicating that it wishes to register for credentials. The sppPostDevData SOAP method also includes a redirectURI formatted in accordance with [19], generated by the mobile device, which resolves to a resource internal in the mobile device (e.g., http://127.0.0.1:12345/index.htm). After the OSU server has received all the data needed to complete the registration process, it redirects the mobile device’s web browser to the redirectURI (see step 6). The internal event caused by retrieval of the redirectURI resource can be used to signal to the mobile device’s connection manager that the registration process is complete and to proceed to the next step in the message sequence. The mobile device implementation to accomplish this is out-of-scope of this specification.

Step 3: The OSU server transmits the sppPostDevDataResponse SOAP method, including a server generated sessionID value, to the mobile device. The sessionID is a 128-bit random number used to bind together messages from a specific mobile device to the OSU server; the sessionID value generated in this step shall be used in all subsequent SOAP messages exchanged in this flow. The content of the response method informs the mobile device of the next step in the OSU process. Since the user has signaled for subscription registration, the OSU server returns a command for the mobile device to launch a browser to the URI supplied in the message. The URI shall contain the sessionID value copied from the sppPostDevDataResponse SOAP method’s sessionID attribute (see section A.3.2), e.g., “https://osuserver.example.com/sign-up/smartphone/index.html?sessionID=abcdef0123456789zz9876543210fedcba”.

Note: the length of the sessionID is intended to minimize the possibility an attacker could successfully guess the sessionID value and disrupt or break the provisioning process of another mobile device, including the possibility to steal their credential.

If no errors occur, sppStatus is set to “OK”; an sppStatus value of “OK” also indicates that the credential provisioning process is not complete.

An OSU server may use the information provided in the DevInfo and DevDetail MOs to determine the type of credential to provision: username/password or certificate.

Note 1: After receipt of the command launch the browser, the mobile device should initiate a listening port of the correct protocol for the redirectURI provided to the OSU server in step 2.

Note 2: Other message flows in section 8 will also use the redirectURI method to signal completion of user input. However, not all message flows require user input (e.g., see section 8.3.3.1 on machine remediation). Since the mobile device may not know at the beginning of a message sequence whether the OSU server needs to use this method (the OSU server decides whether user input is required), the mobile device shall always supply the redirectURI in the sppPostDevData message, but does not need to open the listening port until receipt of a command to launch a browser.

Step 4: The mobile device transmits an HTTPS GET request to the URI provided in the sppPostDevDataResponse in accordance with the procedures in [11]

Step 5 is identical to step 8 in section 8.3.2.3.

Step 6: At the end of the exchange of registration data, the OSU server transmits a final HTTP 302 REDIRECT message to the mobile device.

Step 7: The mobile device transmits an sppPostDevData message to the OSU server containing the sessionID and requestReason. The requestReason shall be set to “User input completed”.

Note: the sessionID facilitates the use of multiple TLS tunnels and/or TCP connections between a given mobile device and the OSU server. For implementation reasons, a mobile device may use more than one TLS tunnel and/or more than one TCP connection throughout this message sequence.

Step 8: The OSU server transmits an sppPostDevDataResponse message to the mobile device. The sppPostDevDataResponse includes a sessionID, an addMO command and the PerProviderSubscription MO

Step 9: If the mobile device was able to successfully add the PerProviderSubscription MO (and other MOs, if present) to its management tree, the mobile device shall transmit to the OSU server an sppUpdateResponse SOAP method that contains the sessionID and the sppStatus set to "OK". If the mobile device was not able to add the PerProviderSubscription MO (and other MOs, if any are present), the mobile device shall transmit to the OSU server an sppUpdateResponse message with sppStatus set to "Error occurred" and include an Error Code.

When the mobile device transmits an sppUpdateResponse message with the sppStatus field value set to "Error occurred", the mobile device shall deem credential provisioning to have failed and shall not attempt to use any credential which may have been received during the message sequence shown in Figure 45.

Step 10: If no error occurred in step 9, the OSU server transmits an sppExchangeComplete SOAP method, containing the sessionID, to the mobile device with sppStatus set to "Exchange complete, release TLS connection". If in step 9 the OSU server received an sppUpdateResponse message with sppStatus set to "Error occurred", it shall take one of the following actions:

- Transmit an sppExchangeComplete message with sppStatus set to "Exchange complete, release TLS connection" and an errorCode set to "Credentials cannot be provisioned at this time". In this case, the OSU server is declaring the credential provisioning process failed. A mobile device receiving this message shall abort the OSU process.
- Transmit an sppPostDevDataResponse message with sppStatus set to "Provisioning complete, request sppUpdateResponse" and include the exec command to add the PerProviderSubscription MO, re-trying mobile device provisioning.

Step 11: The mobile device uses HTTPS to retrieve the trust anchor CA certificates for the AAA server, subscription remediation server, and policy server (if needed) using the CertURLs in the PPS MO according to section 7.3.1.

Step 12: The mobile device releases the TLS session it established in step 1.

Step 13: The mobile device shall disassociate and then associate to the Wi-Fi AN using the newly established credentials, if available.

8.4.2.2 Provisioning certificate credentials

Figure 46 shows the message exchange sequence for certificate provisioning to a mobile device using enrollment. Section A.3 specifies the SOAP XML methods listed in the figure.

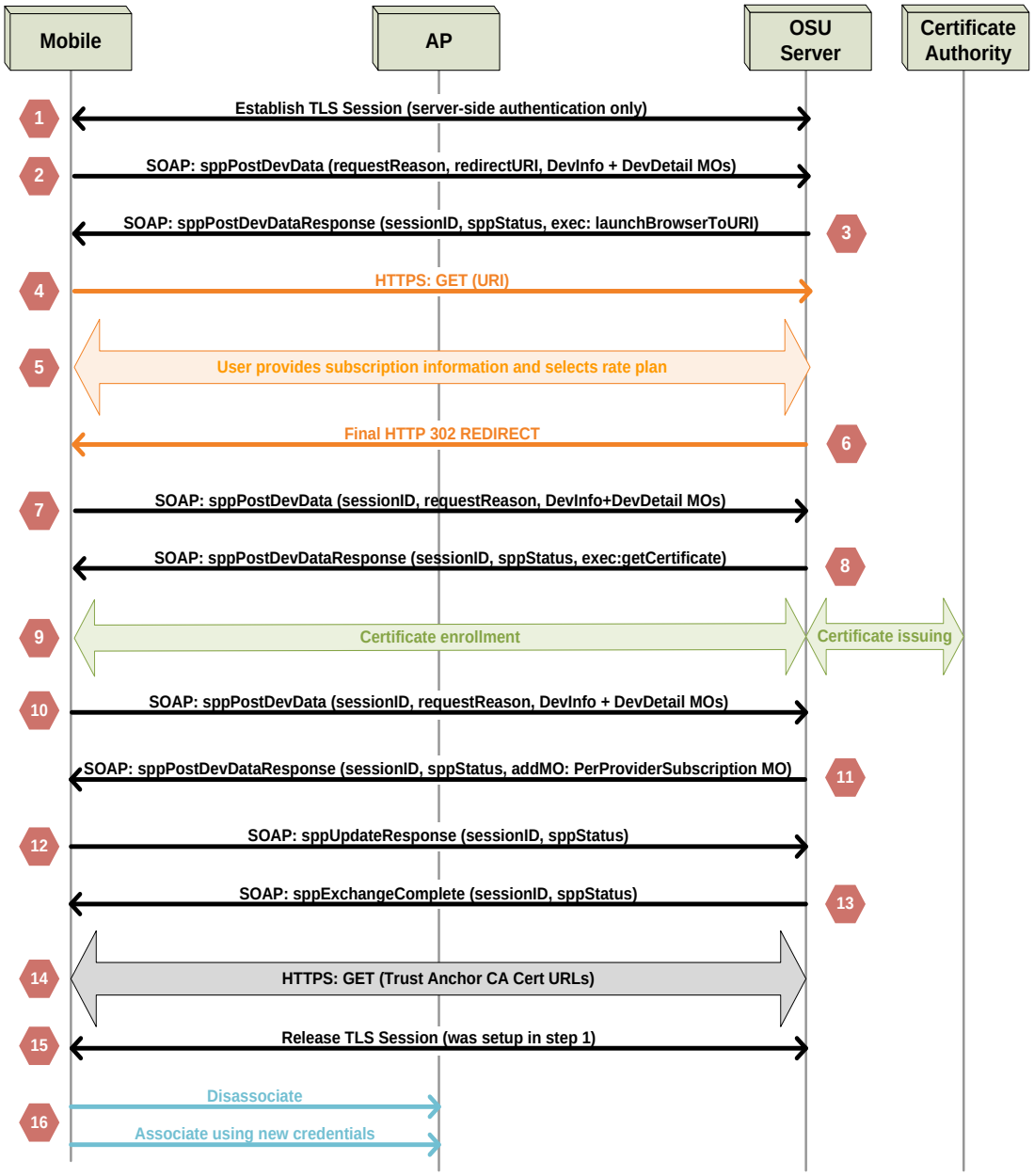


Figure 46: Message exchange diagram for certificate credential provisioning using SOAP XML

The description for each numbered message in Figure 46 is:

Steps 1 – 5 are identical to those above in section 8.4.2.1. Note: in step 2, the mobile device provides its unique identity (e.g. IMEI/MEID, Wi-Fi MAC address or the DevInfo MO's DevId) and the OSU server records it for a later use.

Step 6: At the end of the exchange of registration data, the OSU server transmits a final HTTP 302 REDIRECT message to the mobile device.

Step 7: The mobile device transmits an sppPostDevData SOAP method to the OSU server containing the sessionID copied from the sppPostDevDataResponse in step 3 and requestReason. The requestReason shall be set to "User input completed".

Note: the sessionID facilitates the use of multiple TLS tunnels and/or TCP connections between a given mobile device and the OSU server. For implementation reasons, a mobile device may use more than one TLS tunnel and/or more than one TCP connection throughout this message sequence.

Step 8: The OSU server transmits an sppPostDevDataResponse SOAP method to the mobile device. The sppPostDevDataResponse message includes a sessionID and a getCertificate Exec command.

Step 9: The mobile device performs certificate enrollment according to the procedures in section 7.6. This includes downloading CA certificate(s) that may be used as the AAA server trust anchor (see section 7.3.1).

Step 10: The mobile device transmits the sppPostDevData SOAP method to the server, which includes a sessionID, requestReason and the OMA DM DevInfo and DevDetail MOs. The mobile device shall set the value for requestReason to “Certificate enrollment completed” if certificate enrollment succeeded or “Certificate enrollment failed” if certificate enrollment failed.

Step 11: If certificate enrollment succeeded, the OSU server transmits the sppPostDevDataResponse SOAP method to the mobile device that includes the addMO command and the PerProviderSubscription MO. The PerProviderSubscription MO is specified in section 9.1 and contains certificate identifiers binding the provisioned certificate to the subscription as well as ancillary provisioning data. In the sppPostDevDataResponse SOAP method, the sessionID is included, the sppStatus is set to “Provisioning complete” request sppUpdateResponse” to indicate that the subscription and certificate provisioning process steps has have been completed and requesting the mobile device to confirm installation of the PerProviderSubscription MO (and other MOs, if present).

If certificate enrollment failed, the OSU server shall respond with one of the following:

1. It switches to provisioning a machine-managed username/password credential. In this case, the sppPostDevDataResponse SOAP method returned in this step shall contain an addMO command and PerProviderSubscription MO; the PerProviderSubscription MO will contain the provisioned, machine-managed username/password credential. The sppStatus shall be set to “Provisioning complete, request sppUpdateResponse”.
2. It switches to provisioning a user-selected username/password credential. In this case, the sppPostDevDataResponse SOAP method returned in this step shall contain an exec command to launch a browser to the provided URI. The sppStatus is set to “OK”. The message flow continues as described in section 8.4.2.1, step 4.
3. It aborts the credential provisioning. In this case, the sppExchangeComplete sppPostDevDataResponse SOAP method returned in this step will include an sppStatus set to “Exchange complete, release TLS connection”, and the sppError element including an errorCode set to “Credentials cannot be provisioned at this time” shall be returned.

Step 12: If the sppStatus value in step 11 was set to "Provisioning complete, request sppUpdateResponse" and the mobile device was able to successfully add the PerProviderSubscription MO (and other MOs, if present) to its management tree, the mobile device shall transmit to the OSU server an sppUpdateResponse SOAP method with a sessionID and sppStatus set to "OK"; if the mobile device was not able to add the PerProviderSubscription MO (and other MOs, if present), the mobile device shall transmit to the OSU server an sppUpdateResponse message with sppStatus set to "Error occurred" and include an Error Code.

Step 13: If no error occurred in step 12, the OSU server transmits an sppExchangeComplete SOAP method to the mobile device with sppStatus set to "Exchange complete, release TLS connection".

If in step 12 the OSU server received an sppUpdateResponse message with sppStatus set to "Error occurred", it shall take one of the following actions:

- Transmit an sppExchangeComplete message with sppStatus set to "Exchange complete, release TLS connection" and an errorCode set to "Credentials cannot be provisioned at this time". In this case, the OSU server is declaring the credential provisioning process failed. A mobile device receiving this message shall abort the OSU process and shall not attempt to use any credential which may have been received during the message sequence of Figure 46.
- Transmit an sppPostDevDataResponse message with sppStatus set to "Provisioning complete, request sppUpdateResponse" and include the exec command to add the PerProviderSubscription MO, re-trying mobile device provisioning.

Step 14: The mobile device uses HTTPS to retrieve the trust anchor CA certificates (if needed) for the subscription remediation server, policy server, and maybe the AAA server using the CertURLs in the PPS MO, according to section 7.3.1.

Step 15: The TLS session established in step 1 is released.

Step 16: The mobile device shall dissociate and then associate to the Wi-Fi AN using the newly established credentials, if available.

8.4.2.3 Provisioning using mobile device provided certificates

Some SPs may elect to use pre-provisioned client certificates. The decision whether to use these certificates for Wi-Fi network access is up to the discretion of the SP. The message sequences in this section assume that the SP's credential provisioning policy is to use these certificates, if possible (per the certification negotiation procedures described).

Pre-provisioned client certificates include certificates which have been provisioned by an SP or the manufacturer. The methods and protocols by which these certificates are provisioned to a mobile device are outside the scope of this specification. Figure 47 shows the message exchange sequence for the negotiation of client certificates. Section A.3 specifies the SOAP XML methods listed in the figure.

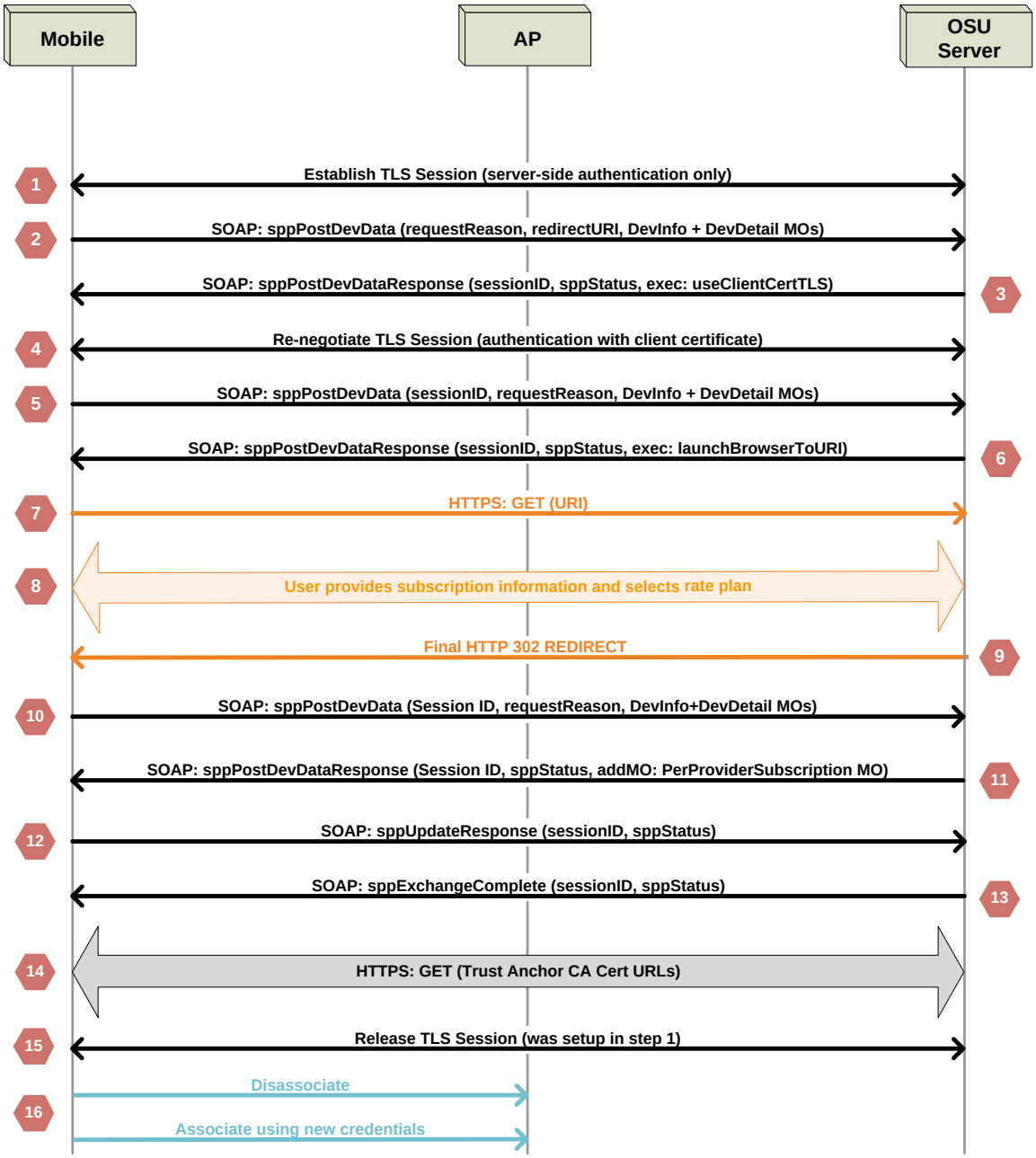


Figure 47: Message exchange diagram for negotiating client certificate using SOAP XML

The description for each numbered message in Figure 47 is:

Steps 1 – 2 are identical to those in section 8.4.2.1. Note: in the DevDetail MO, there is a vendor-specific extension indicating whether the mobile device is in possession of an IEEE 802.1ar manufacturing certificate or other Service Provider issued certificate(s). [58]

Step 3: The OSU server transmits the sppPostDevDataResponse SOAP method, including a server-generated sessionID, to the mobile device. The sessionID is a 128-bit random number used to bind together messages from a specific mobile device to the OSU server; the sessionID value generated in this step shall be used in all messages exchanged in this flow. Since the user has signaled for subscription registration and the DevDetail MO indicates the mobile device is in possession of either a manufacturing certificate or other SP issued certificates, the

sppPostDevDataResponse contains the exec command, useClientCertTLS. If no errors occur, sppStatus shall be set to “OK”; an sppStatus value of “OK” indicates that the credential provisioning process is not complete.

Note: the length of the sessionID is intended to minimize the possibility an attacker could successfully guess the sessionID value and disrupt or break the provisioning process of another mobile device.

The useClientCertTLS command requests the mobile device to re-negotiate the TLS session using a client certificate. The sppPostDevDataResponse SOAP method shall have its moreCommands attribute set to true. This indicates that the mobile device is requested to send another sppPostDevData SOAP message to the server after TLS re-negotiation is completed. The useClientCertTLS element, which is specified in section A.3, contains information on the certificate types and issuers that are regarded as acceptable.

Step 4: If the mobile device possesses certificates acceptable to the OSU server (i.e., it possesses a certificate identified by the type or issuer in the useClientCertTLS XML instance document), it re-negotiates the TLS connection to the OSU server in accordance with the procedures in [31] using a client certificate. If the mobile device has more than one certificate meeting the criteria in the useClientCertTLS XML instance document, then TLS negotiation shall be used to determine a mutually acceptable certificate. In case the mobile device and server are unable to negotiate to a mutually acceptable client certificate[6], the mobile device shall continue using the TLS connection with server-side only authentication.

Step 5: The mobile device transmits the sppPostDevData SOAP method, containing the sessionID, the OMA DM DevInfo and DevDetail MOs to the OSU server. The mobile device successfully re-negotiated the TLS connection, authenticating with a client certificate, it shall set the value for requestReason to “Subscription registration”.

If the mobile device is not in possession of an acceptable client certificate or was unable to successfully perform TLS re-negotiation using a client certificate, it shall set the value for requestReason to “No acceptable client certificate” and continue using the TLS connection setup in Step 1.

Note: the sessionID facilitates the use of multiple TLS tunnels and/or TCP connections between a given mobile device and the OSU server. For implementation reasons, a mobile device may use more than one TLS tunnel and/or more than one TCP connection throughout this message sequence.

Step 6: The OSU server transmits the sppPostDevDataResponse SOAP method, including the sessionID, to the mobile device. Since the user has signaled for subscription registration, the OSU server returns a command for the mobile device to launch a browser to the URI supplied in the message. The URI shall contain the sessionID value copied from the sppPostDevDataResponse SOAP method’s sessionID attribute (see section A.3.2) (e.g., “https://osuserver.example.com/sign-up/smartphone/index.html&sessionID=abcdef0123456789zz9876543210fedcba")

Step 7: The mobile device transmits an HTTPS GET request to the URI provided in the sppPostDevDataResponse in accordance with the procedures in [11].

Steps 8 – 16 are identical to steps 5 – 13 in section 8.4.2.1.

[6] This should be a rare event since the mobile device should only be attempting to use client certificates which the OSU server has indicated are acceptable.

8.4.3 Subscription management

Subscription management refers to remediation of credential problems (e.g., password expiration), remediation of account problems (e.g., user did not pay their bill) or the update of subscription provisioning information (e.g., update of a roaming consortium OI).

This section describes subscription management using SOAP XML for the following cases:

1. Machine remediation when the mobile device has a Username and Password (see section 8.4.3.1)
2. User remediation when the mobile device has Username and Password credentials (see section 8.4.3.2)
3. Machine remediation when the mobile device has certificate credentials (see section 8.4.3.3)
4. User remediation when a mobile device has certificate credentials (see section 8.4.3.4)
5. Updating a subscription using certificate credentials (see sections 8.4.3.5 and 8.4.3.6)

The subscription remediation server shall request authentication for the realm provided to the user in /PerProviderSubscription/<X+>/Credential/Realm. The subscription remediation server knows the realm value as the server is required to maintain a one to one relationship with that realm and the URI provided in /PerProviderSubscription/<X+>/SubscriptionUpdate/URI.

The OSU server should use the registration protocol defined in Annex E to embed standard tags, if applicable, for user remediation.

8.4.3.1 Machine remediation when a mobile device has username and password credentials

Figure 48 shows the message exchange sequence for machine remediation of a subscription when the mobile device is in possession of a username and password credential is provided. Section A.3 specifies the SOAP XML methods listed in the figure.

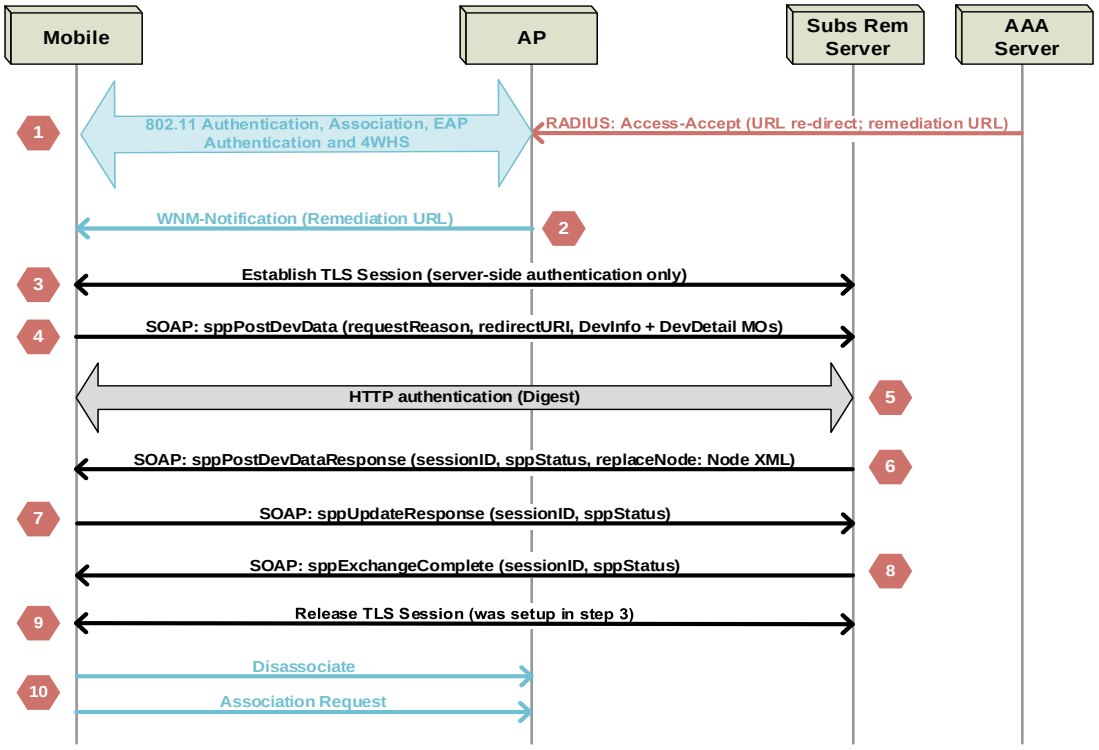


Figure 48: Message exchange diagram for machine remediation of a subscription using username and password credentials

The description for each numbered message in Figure 48 is:

Step 1: During the authentication process, when a SP determines that subscription remediation (see section 3.2.1) is needed for a user, the AAA server sends an Access-Accept message with remediation indication (see D.1) to the authenticator (assumed to be co-located with the AP in Figure 48) at the end of the mobile device's EAP authentication sequence.

If the AAA server signals remediation is required, the AP shall not cache the PMKSA. Note: this forces the mobile device to perform a full EAP authentication in step 11, thereby ensuring that any access restrictions enforced in step 1 are removed.

Note: The URL is present in the Access-Accept message to support, for example, upgrade of mobile device from Release 1 to Release 2.

Step 2: The AP shall transmit a WNM-Notification Request frame containing the remediation URL (see section 3.2.1).

The WNM-Notification request frame is a robust action frame because the use of PMF is required. The mobile device shall only use the URL contained in the WNM-Notification Request frame when it is attached to a hotspot operated by the Home SP.

The mobile device that is in possession of a PerProviderSubscription MO shall ignore the provisioning protocol method and the URL contained in the WNM-Notification Request frame. The mobile device having a SIM credential but not in possession of a PerProviderSubscription MO can use this URL and provisioning protocol method to enable provisioning of a PerProviderSubscription MO, as defined in section 8.5.

Step 3: The mobile device initiates a TLS connection to the subscription remediation server in accordance with the procedures in [29]. The mobile device shall validate the subscription remediation server certificate according to the procedures in section 7.3.5.2. If the mobile device is unable to initiate a TLS connection to the subscription remediation server, it shall abort the remediation process. The mobile device shall not attempt subscription remediation using HTTP (i.e., instead of HTTPS).

Note: separate TCP connections (and TLS sessions) can be used for the SOAP XML exchanges.

Step 4: The mobile device transmits the sppPostDevData SOAP method to the server that includes the OMA DM DevInfo, DevDetail and PerProviderSubscription MOs. The mobile device shall set the value for requestReason to "Subscription remediation". The sppPostDevData SOAP method also includes a redirectURI formatted in accordance with [17], generated by the mobile device, which resolves to a resource internal to the mobile device (e.g., http://127.0.0.1:1234/index.htm).

Note: Since the mobile device does not know at the beginning of a remediation sequence whether the OSU server needs user input, the mobile device shall supply the redirectURI in the sppPostDevData message. In this case (machine remediation), the redirectURI is not used.

Step 5: The subscription remediation server shall request HTTP authentication using the digest method in accordance with the procedures in [9]. The mobile device shall provide its username and password digest to the server in accordance with [9]. If HTTP authentication is not successful, remediation is not possible and the mobile device shall abort the process and should inform the user[7].

Step 6: The subscription remediation server transmits the sppPostDevDataResponse SOAP method, including a server generated sessionID value, to the mobile device that includes XML data for one or more interior nodes of the PerProviderSubscription MO (see section 9.1).

[7] The user should be informed because the AN may be restricting access such that the subscription may not be useable to the subscriber for their purpose.

The 128-bit sessionID is used throughout this message sequence to identify messages that belong to the same OSU message exchange. The sessionID value generated in this step shall be used in all subsequent SOAP messages exchanged in this flow.

Note: this facilitates the use of multiple TLS tunnels and/or TCP connections between a given mobile device and the subscription remediation server. For implementation reasons, a mobile device may use more than one TLS tunnel and/or more than one TCP connection throughout this message sequence.

The sppStatus in the sppPostDevDataResponse is set to “Remediation complete request sppUpdateResponse” to indicate the subscription remediation process has been completed and requesting the mobile device to confirm installation of the updated node(s).

Step 7: If the sppStatus value in step 6 was set to "Remediation complete, request sppUpdateResponse" and the mobile device was able to successfully update the requested nodes in its management tree, the mobile device shall transmit to the OSU server an sppUpdateResponse SOAP method, containing the sessionID, having the sppStatus set to "OK"; if the mobile device was not able to update the requested nodes, the mobile device shall transmit to the OSU server an sppUpdateResponse message with sppStatus set to "Error occurred" and include an Error Code. If a TrustRoot node is updated the mobile device shall download the new trust anchor CA certificate(s) and replace the old ones with the new ones.

Step 8: If no error occurred in step 7, the OSU server transmits an sppExchangeComplete SOAP method to the mobile device with sppStatus set to "Exchange complete, release TLS connection".

If in step 7 the OSU server received an sppUpdateResponse message with sppStatus set to "Error occurred", it shall take one of the following actions:

- Transmit an sppExchangeComplete message with sppStatus set to "Exchange complete, release TLS connection" and an errorCode set to "Remediation cannot be completed at this time". In this case, the OSU server is declaring the subscription remediation process failed.

  Note: during subscription remediation, the mobile device's connectivity to the Internet may be restricted. Since the remediation process failed, the problem with the user's subscription remains. It is up to the Home SP what subsequent action to take, if any (e.g., whether to continue restricting user privileges the next time the mobile device uses those credentials to access a Wi-Fi network).

- Transmit an sppPostDevDataResponse message with sppStatus set to "Remediation complete, request sppUpdateResponse" and include one or more updateNode element(s), re-trying mobile device remediation.

Step 9: The mobile device releases the TLS session it established in step 3.

Step 10: The mobile device shall then disassociate and associate to the Wi-Fi AN. If the credentials were updated during the remediation process, then these shall be used.

8.4.3.2 User remediation when a mobile device has username and password credentials

Figure 49 shows the message exchange sequence for user remediation of a subscription when the mobile device is in possession of a username and password credential. Section A.3 specifies the SOAP XML methods listed in the figure.

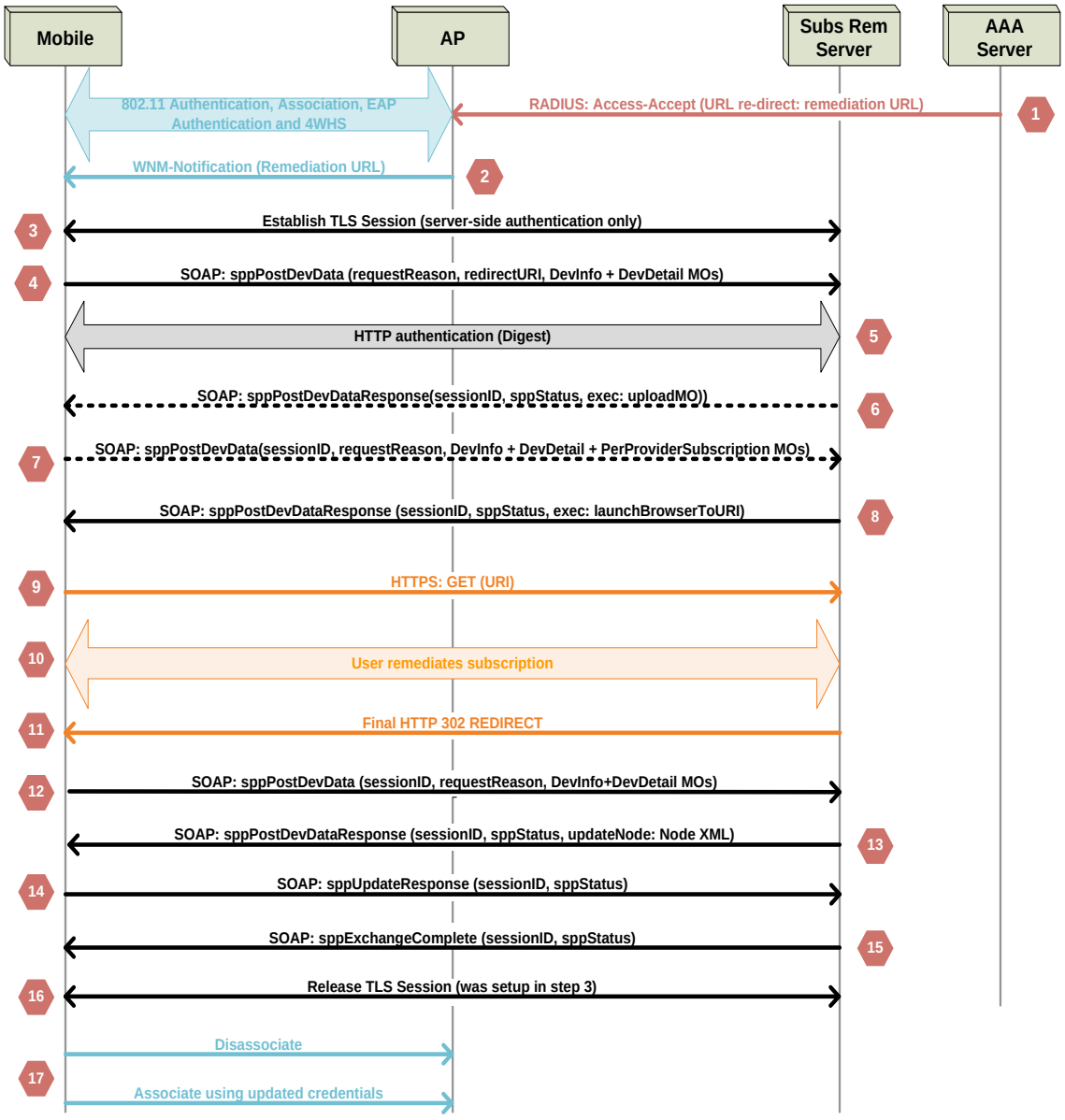


Figure 49: Message exchange diagram for user remediation of a subscription using username and password credentials

The description for each numbered message in Figure 49 is:

Steps 1 – 5 are identical to those above in section 8.4.3.1.

Step 6 (optional for subscription remediation servers): The subscription remediation server transmits the sppPostDevDataResponse SOAP method, including a server-generated sessionID, to the mobile device with an exec command to upload the PerProviderSubscription MO (see section 9.1). The sessionID is a 128-bit random number used to bind together messages from a specific mobile device to the OSU server; the sessionID value generated in this step shall be used in all subsequent SOAP messages exchanged in this flow.

Step 7 (not performed by mobile device unless step 6 is used by the subscription remediation server): The mobile device transmits the sppPostDevData SOAP method to the server, which includes the sessionID copied from the sppPostDevDataResponse in step 6 and the OMA DM

DevInfo, DevDetail and PerProviderSubscription MOs. The mobile device shall set the value for requestReason to “MO upload”. If the subscription remediation server and mobile device both support HTTP cookies, then cookies may be used to avoid the need for another HTTP digest authentication. Otherwise, the subscription remediation server shall cause another HTTP digest authentication with the same username and password used in step 5 to be carried out prior to step 7.

Step 8: The subscription remediation server transmits the sppPostDevDataResponse SOAP method, including the sessionID, to the mobile device. The contents of the response method inform the mobile device to launch a browser to the subscription remediation server’s URI supplied in the message. The URI shall contain the sessionID value copied from the sppPostDevDataResponse SOAP method’s sessionID attribute (see section A.3.2), (e.g. “https://remediation-server.example.com/sign-up/smartphone/index.html&sessionID=abcdef0123456789zz9876543210fedcba”)

Step 9: The mobile device transmits an HTTPS GET request to the URI provided in the sppPostDevDataResponse in accordance with the procedures in [8].

Step 10: The mobile device and subscription remediation server exchange remediation data as required by the SP. The contents of this exchange are outside the scope of this specification.

Step 11: At the end of the exchange of registration data, the OSU server transmits a final HTTP 302 REDIRECT message to the mobile device.

Step 12: The mobile device transmits an sppPostDevData message to the OSU server containing the sessionID and requestReason. The requestReason shall be set to “User input completed”.

Step 13: The subscription remediation server transmits the sppPostDevDataResponse SOAP method, including the sessionID, to the mobile device that includes XML data for one or more interior nodes of the PerProviderSubscription MO (see section 9.1). The sppStatus in the sppPostDevDataResponse is set to “Remediation complete, request sppUpdateResponse” to indicate the subscription remediation process has been completed and requesting the mobile device to confirm installation of the updated node(s).

Step 14: If the sppStatus value in step 6 was set to "Remediation complete, request sppUpdateResponse" and the mobile device was able to successfully update the requested nodes in its management tree, the mobile device shall transmit to the OSU server an sppUpdateResponse message having the sppStatus set to "OK"; if the mobile device was not able to update the requested nodes, the mobile device shall transmit to the OSU server an sppUpdateResponse message with sppStatus set to "Error occurred" and include an Error Code.

After the subscription remediation server successfully receives the sppUpdateResponse message, it knows the new node XML has been installed on the mobile device (which may include an updated password).

Steps 15 - 17 are identical to steps 8 – 10, respectively, in section 8.4.3.1.

8.4.3.3 Machine remediation when a mobile device has certificate credentials

Figure 50 shows the message exchange sequence for machine remediation of a subscription when the mobile device is in possession of a certificate credential. Note that this message sequence cannot be used to update a certificate credential; see section 8.4.3.5 for that message sequence. Section A.3 specifies the SOAP XML methods listed in the figure.

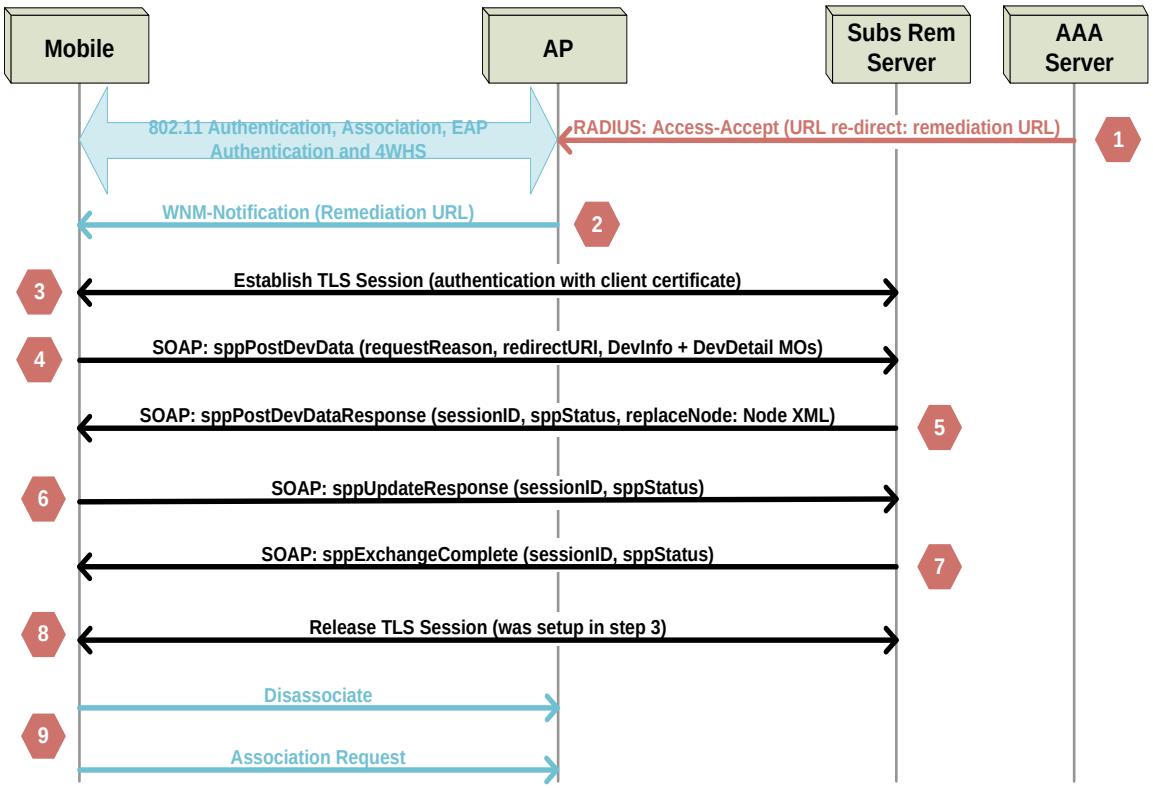


Figure 50: Message exchange diagram for machine remediation of a subscription using certificate credentials

The description for each numbered message in Figure 50 is:

Steps 1 – 2 are identical to those above in machine remediation when a mobile device has username and password credentials

Step 3: The mobile device initiates a TLS connection to the subscription remediation server in accordance with the procedures in [29]. The mobile device shall validate the subscription remediation server certificate according to the procedures in section 7.3.5.2. If the mobile device is unable to initiate a TLS connection to the subscription remediation server, it shall abort the remediation process. The mobile device shall not attempt subscription remediation using HTTP (i.e., instead of HTTPS).

Steps 4 – 9 are identical to steps 5 – 10 in section 8.3.3.1.

8.4.3.4 User remediation when a mobile device has certificate credentials

Figure 51 shows the message exchange sequence for user remediation of a subscription when the mobile device is in possession of a certificate credential. Section A.3 specifies the SOAP XML methods listed in the figure.

If the UsernamePassword node located at PerProviderSubscription/<X+>/SubscriptionUpdate/UsernamePassword is present, the username/password credential contained in the UsernamePassword node shall be used to authenticate to the server.

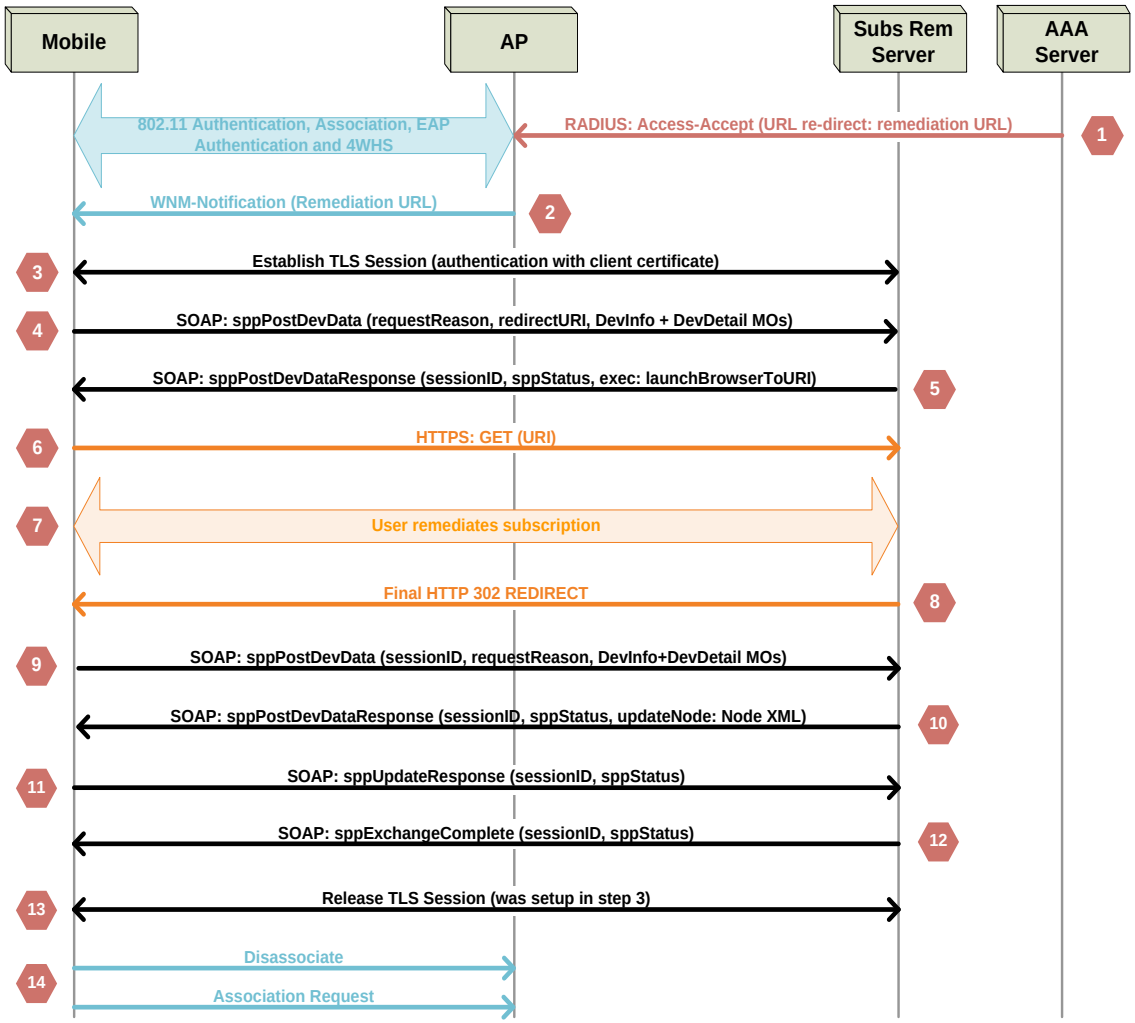


Figure 51: Message exchange diagram for user remediation of a subscription using certificate credentials

The description for each numbered message in Figure 51 is:

Steps 1 – 3 are identical to those above in section 8.4.3.3.

Step 4 is identical to that in section 8.4.3.2.

Steps 5 – 14 are the same as steps 7 – 16 in section 8.4.3.2.

8.4.3.5 Updating a certificate credential

Figure 52 shows the message exchange sequence for machine remediation of a certificate credential (i.e., certificate re-enrollment). Section A.3 specifies the SOAP XML methods listed in the figure.

If the UsernamePassword node located at PerProviderSubscription/<X+>/SubscriptionUpdate/UsernamePassword is present, the username/password credential contained in the UsernamePassword node shall be used to authenticate to the server.

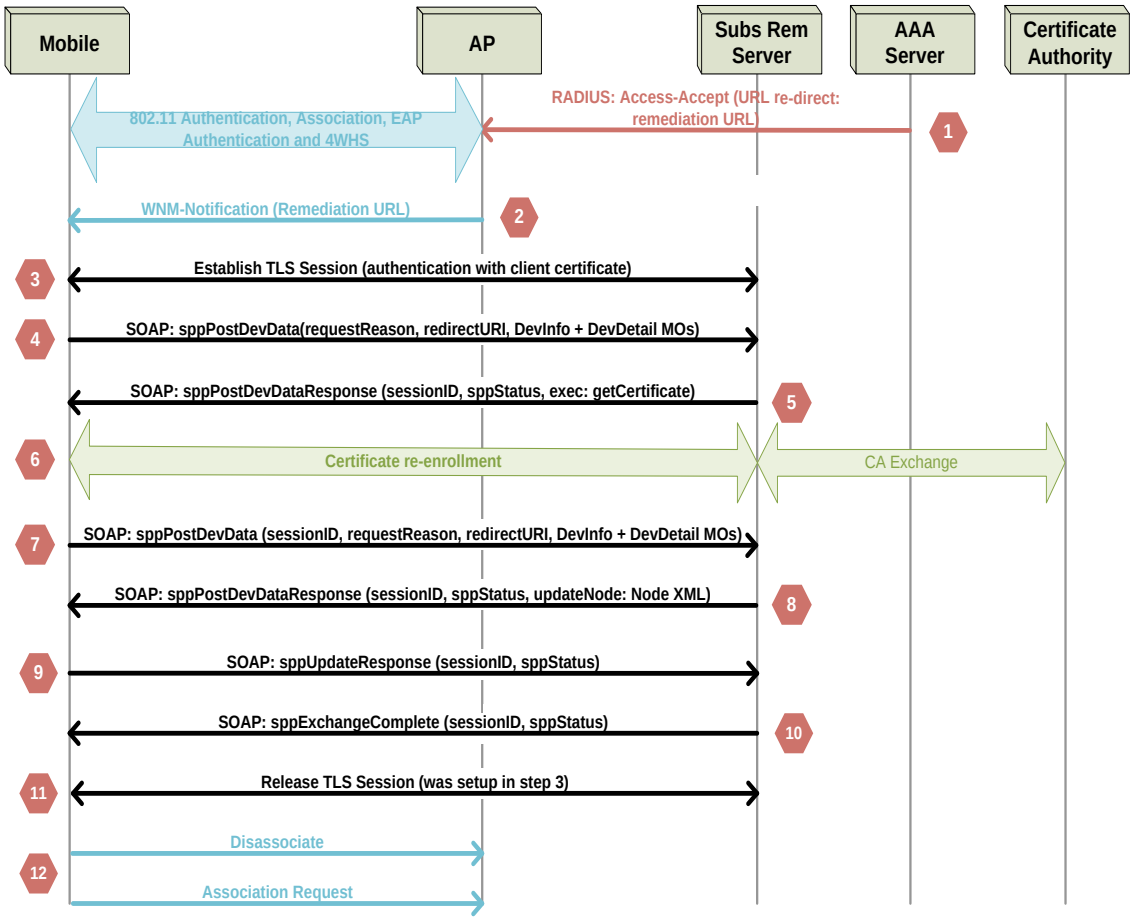


Figure 52: Message exchange diagram for certificate re-enrollment

The description for each numbered message in Figure 52 is:

Steps 1 – 4 are identical to those above in section 8.4.3.3.

Step 5: The subscription remediation server transmits the sppPostDevDataResponse SOAP method to the mobile device which includes a server-generated sessionID and the exec command, getCertificate. The sessionID is a 128-bit random number used to bind together messages from a specific mobile device to the OSU server; the sessionID value generated in this step shall be used in all subsequent SOAP messages exchanged in this flow.

Step 6: is the same as step 9 in section 8.4.2.2, except that in step 6, the HTTPS POST in EST shall request re-enrollment (see section 4.2.2 in [36]).

Step 7 is the same as step 4 in 8.4.3.1 except for the requestReason is set to Certificate enrollment completed.

Step 8: If certificate re-enrollment succeeded, the subscription remediation server transmits the sppPostDevDataResponse SOAP method to the mobile device that includes a sessionID and an updated node XML for the PerProviderSubscription MO. In the sppPostDevDataResponse SOAP method, the sppStatus is set to “Remediation complete, request sppUpdateResponse” to indicate that the certificate re-enrollment process has been completed and to request the mobile device to confirm installation of the updated node(s).

If certificate re-enrollment failed, the subscription remediation server shall respond with one of the following:

1. It instructs the mobile device to continue using its existing certificate. In this case, the sppPostDevDataResponse SOAP method returned in this step will include an sppStatus set to ““Exchange complete, release TLS connection”, and the sppError element including an errorCode set to “Continue to use existing certificate” shall be returned. This completes the subscription remediation process.
2. It informs the mobile device to use a machine-managed username and password credential. In this case, the sppPostDevDataResponse SOAP method returned in this step will include updated node XML for the PerProviderSubscription MO; the PerProviderSubscription MO will contain the provisioned, machine-managed username/password credential. The sppStatus is set to “Provisioning complete, request sppUpdateResponse”.
3. It switches to provisioning a user-selected username/password credential. In this case, the sppPostDevDataResponse SOAP method returned in this step will contain an Exec command to launch a browser to the provided URI. The sppStatus is set to “OK”. The message flow continues as described in section 8.4.3.2, step 7.
4. It aborts certificate re-enrollment. In this case, the sppPostDevDataResponse SOAP method returned in this step will include an sppStatus set to “Exchange complete, release TLS connection”, and the sppError element including an errorCode set to “Credentials cannot be provisioned at this time” shall be returned. This response effectively means the user shall be unable to authenticate to the network until the certificate credential problems are resolved at some future point in time.

Steps 9 - 12 are identical to steps 6 - 9 in section 8.4.3.3.

8.4.3.6 Updating a certificate credential when UsernamePassword node is Present

Figure 53 shows the message exchange sequence for machine remediation of a certificate credential (i.e., certificate re-enrollment), when the UsernamePassword node is present in the PerProviderSubscription/<X+>/SubscriptionUpdate node in the PerProviderSubscription MO.

This message sequence differs from that of section 8.4.3.5 because the credentials used to authenticate to the subscription remediation server are drawn from the information contained in that UsernamePassword node.

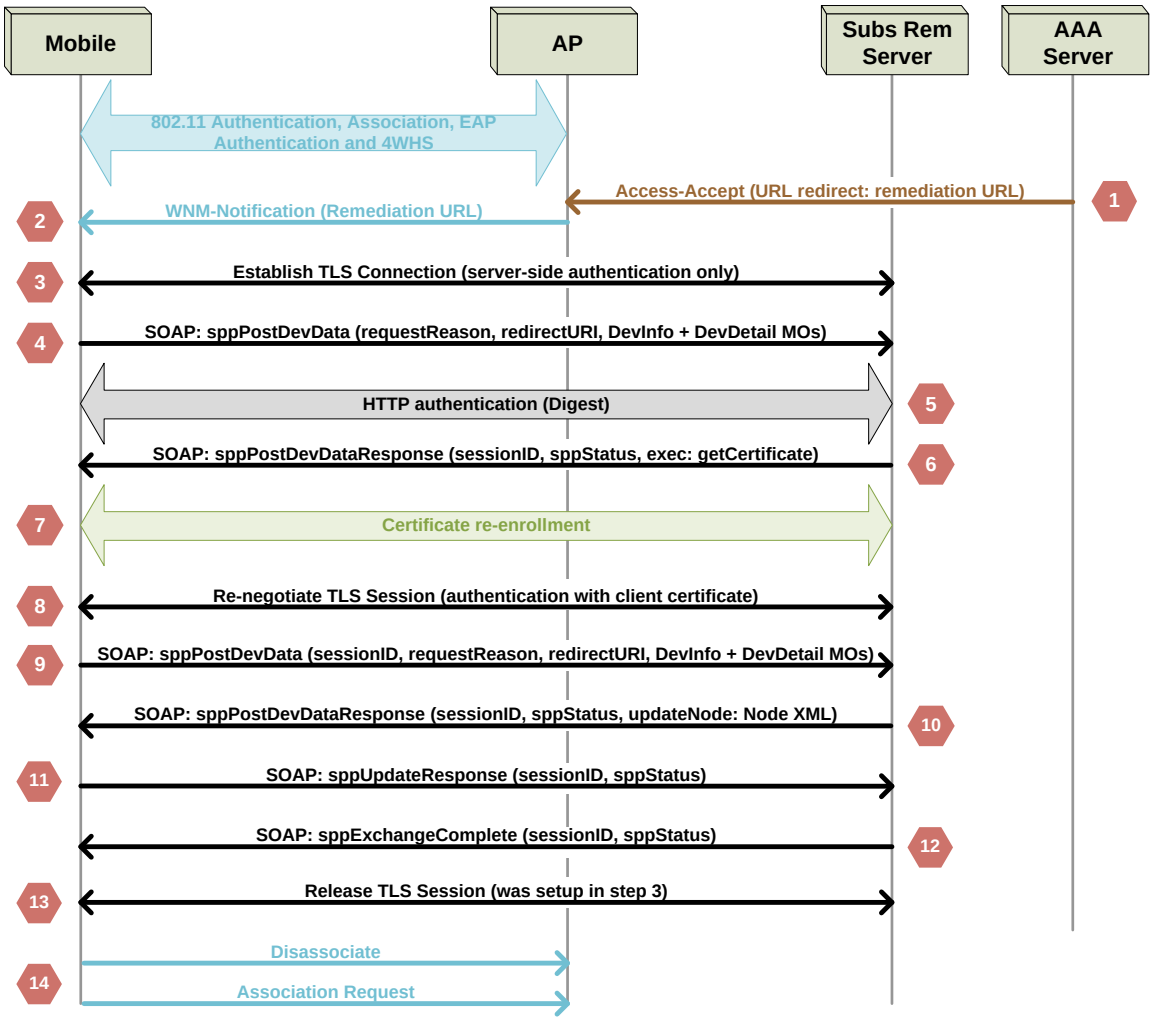


Figure 53: Message exchange diagram for updating certificate credentials

Steps 1 – 5 are identical to those in section 8.4.3.5.

Step 6 is the same as step 5 in section 8.4.3.2.

Steps 7 - 14 are the same as steps 6 - 13 in section 8.4.3.5.

8.4.4 Policy provisioning

This section provides the procedures for a mobile device to retrieve SP policy for a subscription. The mobile device shall not attempt to use these procedures until it has been provisioned with credentials for that subscription.

8.4.4.1 Policy provisioning and update with username and password credentials

Figure 54 shows the message exchange sequence for SP policy provisioning and update when the mobile device is in possession of a username and password credential. Section A.3 specifies the SOAP XML messages listed in the figure.

If the UsernamePassword node located at PerProviderSubscription/<X+>/Policy/PolicyUpdate/UsernamePassword is present, the username/password credential contained in the UsernamePassword node shall be used to authenticate to the server.

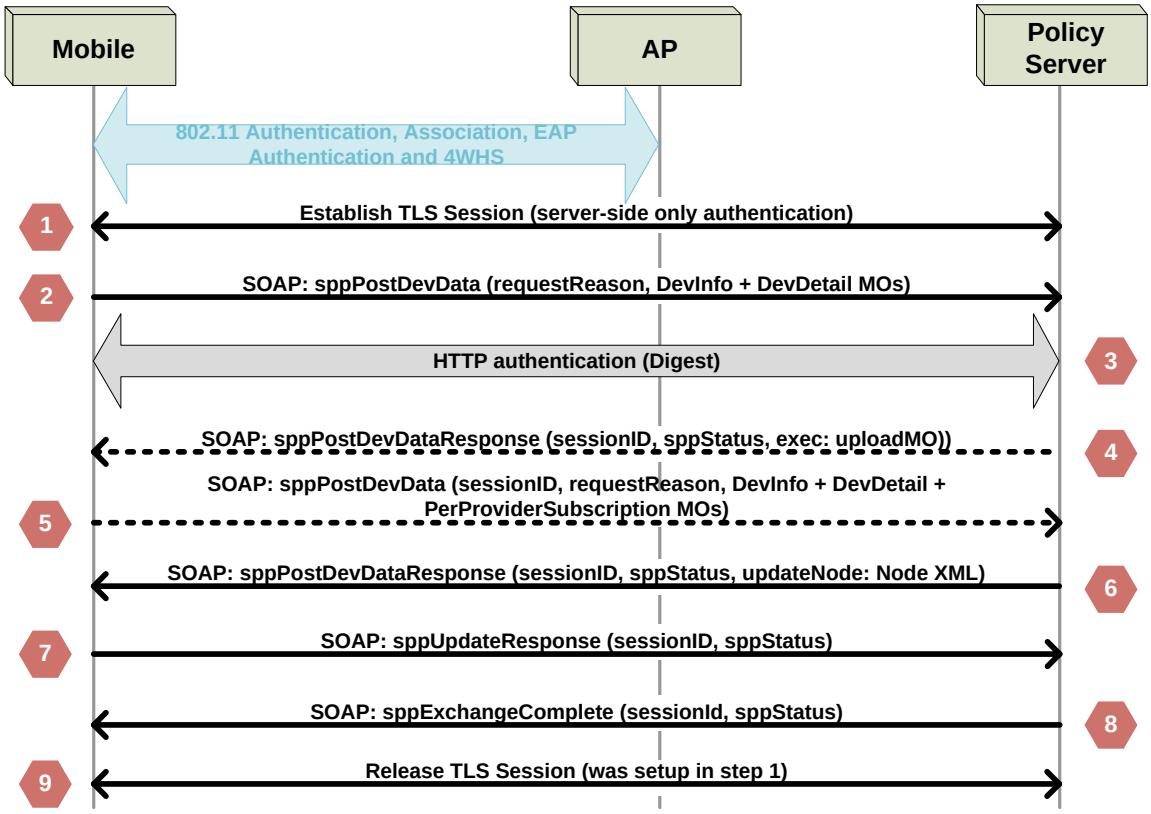


Figure 54: Message sequence diagram for SP policy provisioning and update when the mobile device has username and password credentials

The description for each numbered message in Figure 54 is:

Step 1: The mobile device initiates a TLS connection to the policy server in accordance with the procedures in [29]. The URI of the policy server is provided in the PerProviderSubscription MO. The mobile device shall validate the policy server certificate according to the procedures in section 7.3.6.2. If the mobile device is unable to initiate a TLS connection to the policy server, it shall abort the policy update process. The mobile device shall not attempt connection to the policy server using HTTP (instead of HTTPS).

Note: separate TCP connections (and TLS sessions) can be used for the SOAP XML exchanges.

Step 2: The mobile device transmits the sppPostDevData SOAP method to the server. This method includes the OMA DM DevInfo and DevDetail MOs. The mobile device shall set the value for requestReason to “Policy update”.

Step 3: The policy server shall request HTTP authentication using the digest method, in accordance with [9]. The mobile device shall provide its username and password digest to the server in accordance with [9]. If HTTP authentication is not successful, policy download is not possible and the mobile device shall abort the process.

Step 4 (optional for policy server): The policy server transmits the sppPostDevDataResponse SOAP method, including a sessionID that was generated by the server, to the mobile device with an exec command to upload the PerProviderSubscription MO (see section 9.1). The sessionID is a 128-bit random number used to bind together messages from a specific mobile device to the OSU server; the sessionID value generated in this step shall be used in all subsequent SOAP messages exchanged in this flow.

Note: this facilitates the use of multiple TLS tunnels and/or TCP connections between a given mobile device and the policy server. For implementation reasons, a mobile device may use more than one TLS tunnel and/or more than one TCP connection throughout this message sequence.

Step 5 (not performed by the mobile device unless step 4 is used by the policy server): The mobile device transmits the sppPostDevData SOAP method including a sessionID value copied from the sppPostDevDataResponse in step 4 to the server, which includes the OMA DM DevInfo, DevDetail and PerProviderSubscription MOs. The mobile device shall set the value for requestReason to “MO upload”.

Step 6: The policy server transmits the sppPostDevDataResponse SOAP method to the mobile device which includes the sessionID, sppStatus and XML data for the policy interior node of the PerProviderSubscription MO (see section 9.1). The sppStatus in the sppPostDevDataResponse is set to “Update complete, request sppUpdateResponse” to indicate the policy provisioning or update process steps has have been completed and to request the mobile device to confirm installation of the updated node(s).

Step 7: If the mobile device was able to successfully update the requested nodes in its management tree, the mobile device shall transmit to the OSU server an sppUpdateResponse message having the sessionID and having the sppStatus set to "OK"; if the mobile device was not able to update the requested nodes, the mobile device shall transmit to the OSU server an sppUpdateResponse message with sppStatus set to "Error occurred" and include an Error Code. If the TrustRoot node is updated the mobile device shall download the new trust anchor CA certificate and replace the old one with the new one.

Step 8: If no error occurred in step 7, the OSU server transmits an sppExchangeComplete message to the mobile device with the sessionID and having the sppStatus set to "Exchange complete, release TLS connection".

If in step 7 the OSU server received an sppUpdateResponse message with sppStatus set to "Error occurred", it shall take one of the following actions:

- Transmit an sppExchangeComplete message that contains the sessionID, has its sppStatus set to "Exchange complete, release TLS connection" and its errorCode set to "Provisioning cannot be completed at this time". In this case, the OSU server is declaring the policy update process failed.
- Transmit an sppPostDevDataResponse message that contains the sessionID, has its sppStatus set to "Update complete, request sppUpdateResponse", and includes one or more updateNode element(s), re-trying the mobile device policy provisioning/update.

Step 9: The mobile device releases the TLS session it established in step 1.

8.4.4.2 Policy provisioning and update with certificate credentials

Figure 55 shows the message exchange sequence for SP policy provisioning and update when the mobile device is in possession of a certificate credential. Section A.3 specifies the SOAP XML methods listed in the figure.

If the UsernamePassword node located at PerProviderSubscription/<X+>/Policy/PolicyUpdate/UsernamePassword is present, the username/password credential contained in the UsernamePassword node shall be used to authenticate to the server.

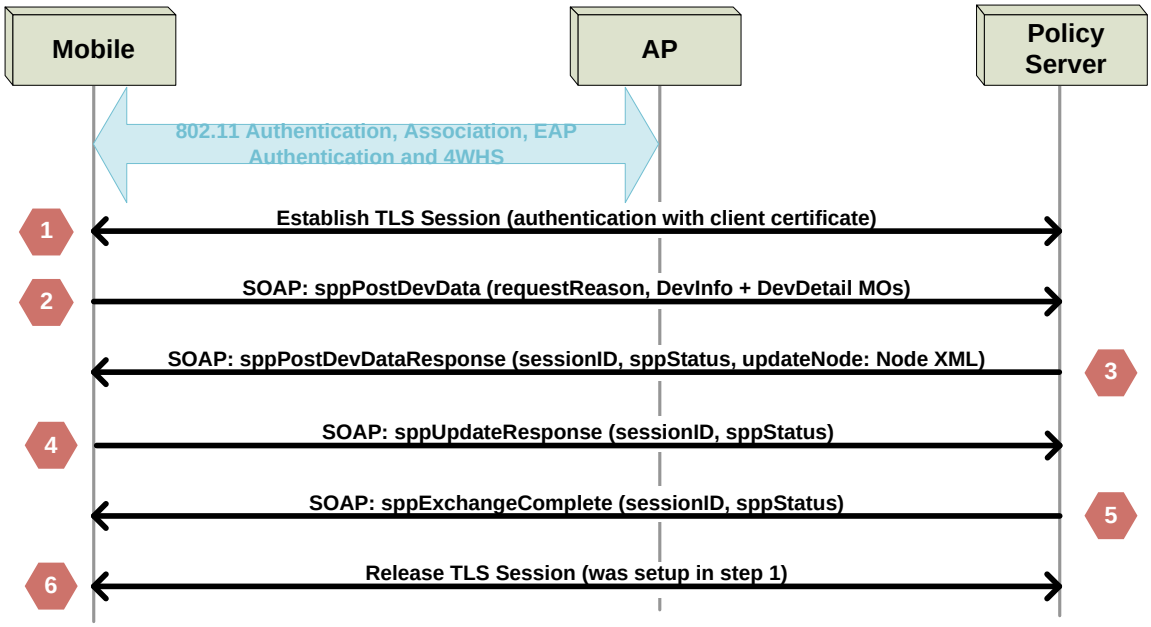


Figure 55: Message sequence diagram for SP policy provisioning and update when the mobile device has certificate credentials

The description for each numbered message in Figure 55 is:

Step 1: The mobile device initiates a TLS connection to the policy server in accordance with the procedures in [29]. The URI of the policy server is provided in the PerProviderSubscription MO. As the mobile device has a certificate credential, it uses that certificate for TLS authentication.

Steps 2 – 6 are identical to those in steps 4 - 8 in section 8.4.4.1.

8.5 Provisioning of a mobile device that has a SIM card

This section describes provisioning procedures that are needed in the case where a mobile device possesses a SIM, noting that credentials are pre-provisioned in the SIM card by the 3GPP mobile operator. These SIM card provisioning procedures are outside the scope of this specification.

The 3GPP mobile operator may use the procedures in this section to deliver the PerProviderSubscription MO used for Wi-Fi AN selection. The choice of using policy included in the PerProviderSubscription MO or a different network selection policy for Wi-Fi access depends on the 3GPP mobile operator.

If the mobile device is HS2.0 Release 2 capable or higher (see section 3.1.1) and the AP is HS2.0 Release 2 capable or higher, then the initial policy provisioning of a mobile device that has a SIM/USIM credential may be accomplished using the subscription remediation process (see the message sequence charts in the following subsections).

8.5.1 Initial subscription metadata and policy provisioning using OMA DM

Figure 56 shows the message exchange sequence for initial subscription metadata and policy provisioning. Section A.1 specifies the OMA DM messages listed in the figure.

Notes:

1.
2. Figure 56 shows the sequence when no PerProviderSubscription MO is available in the mobile device. The message flow supports only machine provisioning; any provisioning requiring user input is not supported.
3. This figure does not show the messages used by the mobile device to obtain an IP address and for server name resolution (e.g., DNS).

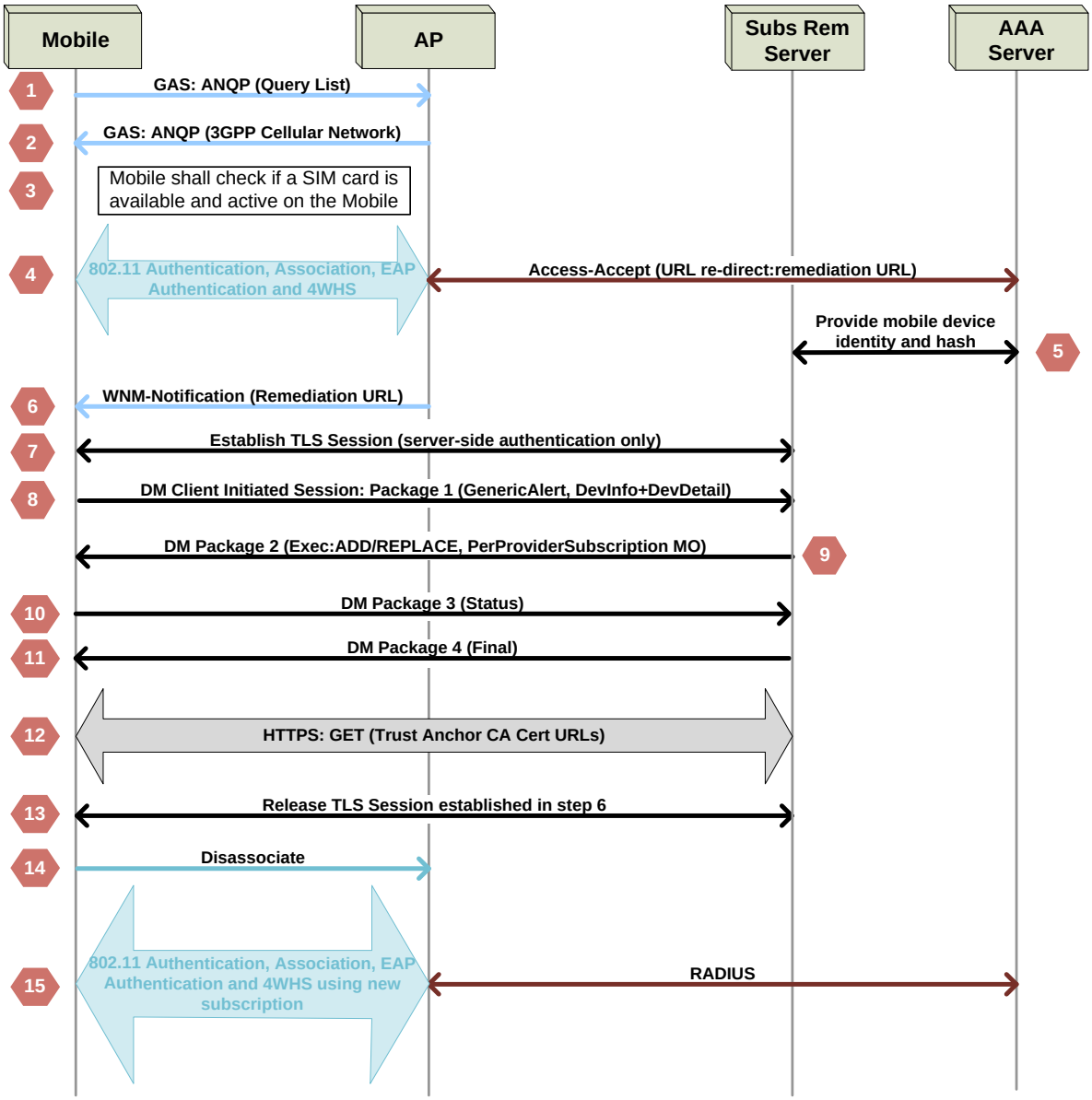


Figure 56: Provision/remediation subscription and policy MO using OMA-DM for the SIM case

The description for each numbered message in

Figure 56 is:

Step 1: The mobile device performs an ANQP query for the 3GPP Cellular Network ANQP-element.

Step 2: The AP returns the 3GPP Cellular Network ANQP-element to the mobile device.

Step 3: If that ANQP query response indicates that the network supports SIM based authentication, the mobile device checks whether it has an active and available SIM card.

Step 4: If a SIM card is available and active in the mobile device, the hotspot operator is the mobile device's Home SP (see section 6.1.1), and the network supports SIM based authentication, then the mobile device initiates association to the Wi-Fi AN using authentication based on the SIM credentials.

When an SP determines that subscription provisioning is needed, then at the end of the EAP authentication sequence its AAA server sends an Access-Accept message with URL re-direct to the Authenticator (i.e., to the AP). The URL supplied to the Authenticator shall include as a parameter an 128-bit hotspot2dot0-mobile-identifier-hash, encoded as 32 hexadecimal digits. The hash is a 128-bit random number generated by the AAA server. The following is an example URL: https://subscription-remediation.example.com/omadm?hotspot2dot0-mobile-identifier-hash=abcdef0123456789zz9876543210fedcba.

Step 5: The AAA server shall provide the tuple {hotspot2dot0-mobile-identifier-hash, mobile device IMSI, mobile device MSISDN} to the subscription remediation server. The means by which this tuple is provided to the server is outside the scope of this specification.

Step 6: The Authenticator causes the AP, with which the mobile device authenticated, to transmit a WNM-Notification Request frame containing a Subscription Remediation subelement, indicating the need for subscription provisioning along with the provisioning protocol, to the mobile device. The Subscription Remediation subelement shall include the URL received from the AAA server.

Step 7: The mobile device initiates a TLS connection to the subscription remediation server in accordance with the procedures in [29]. The mobile device shall validate the subscription remediation server certificate according to the procedures in section 7.3.5.2. If the mobile device is unable to initiate a TLS connection to the subscription remediation server, it shall abort the remediation process. The mobile device shall not attempt subscription remediation using HTTP (i.e., instead of HTTPS). The subscription remediation server shall verify that the hotspot2dot0-mobile-identifier-hash received from the mobile device exactly matches the hotspot2dot0-mobile-identifier-hash received from the AAA server; if the hashes do not exactly match, the server shall terminate the provisioning session.

Step 8: The mobile device sends a DM package 1 message to the subscription remediation server, containing the DevInfo and DevDetail MOs and a Generic Alert indicating the reason for contacting the subscription remediation server. The Reason element in the Generic Alert shall be set to 'org.wi-fi.hotspot2dot0.SubscriptionProvisioning', indicating that the user of the mobile device is attempting to initially provision subscription metadata or policy.

Step 9: The subscription remediation server returns a DM package 2 (see Figure 67) to the mobile device with the command Add PerProviderSubscription MO. In addition, the OSU server shall include in DM package 2 a <respURI> element (see section 6.1.17 in [53]); the payload of the <respURI> element contains an absolute URI having a session key which may be used by the OSU server to bind the DM package transmitted by the mobile device in step 10 with the one transmitted in step 8, regardless of whether a single or multiple TLS/TCP sessions are used.

Step 10: The mobile device sends a DM Package 3 (see Figure 68) indicating the status of the previous operation.

Step 11: The subscription remediation server sends a DM Package 4 (see Figure 71), which does not include any new commands. This indicates the end of the OMA DM session.

Step 12: The mobile device uses HTTPS to retrieve the trust anchor CA certificates for the subscription remediation server and policy server using the CertURLs in the PPS MO according to section 7.3.1

Step 13: The mobile device releases the TLS session that was established in step 6.

Step 14: The mobile device disassociates from the Wi-Fi AN.

Step 15: If the subscription was established successfully in step 7, the mobile device associates again with the same Wi-Fi AN.

8.5.2 Initial subscription metadata and policy provisioning using SOAP XML

Figure 57 shows the message exchange sequence for subscription provisioning. Section A.3 shows the SOAP XML messages used in the figure. The same specification as described in section 8.5.1 applies here.

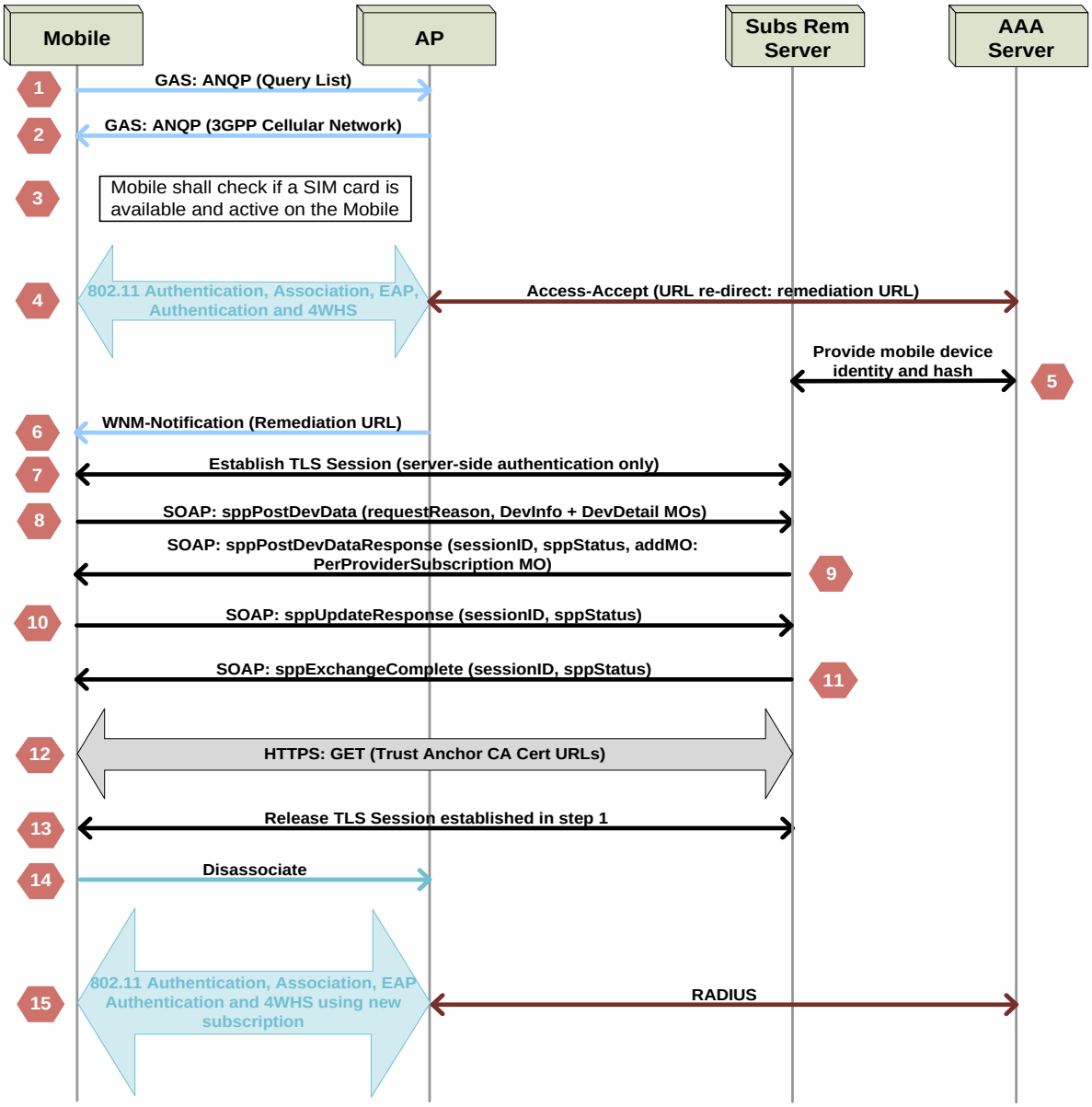


Figure 57: Provision/remediation subscription and policy MO using SOAP XML for the SIM case

Steps 1 – 5 are identical to those in section 8.5.1.

Step 7: The mobile device transmits the sppPostDevData SOAP method to the server which includes the OMA DM DevInfo and DevDetail MOs. The mobile device shall set the value for requestReason to “Subscription provisioning” indicating that the mobile device wants to be provisioned with subscription metadata or policy from its Home service provider.

Step 8: The subscription remediation server transmits the sppPostDevDataResponse SOAP method to the mobile device that includes a server generated sessionID value, an addMO command containing the PerProviderSubscription MO (and zero or more other MOs) and sppStatus status set to "Provisioning complete, request sppUpdateResponse" indicating the provisioning steps have been completed and requesting the mobile device to confirm installation of the updated MO(s) or node(s). The sessionID is a 128-bit random number used to bind together messages from a specific mobile device to the OSU server; the sessionID value generated in this step shall be used in all subsequent SOAP messages exchanged in this flow.

Note: this facilitates the use of multiple TLS tunnels and/or TCP connections between a given mobile device and the OSU server. For implementation reasons, a mobile device may use more than one TLS tunnel and/or more than one TCP connection throughout this message sequence.

Step 9: If the sppStatus value in step 8 was set to "Provisioning complete, request sppUpdateResponse" and the mobile device was able to successfully update the requested MO(s) or node(s) in its management tree, the mobile device shall transmit to the OSU server an sppUpdateResponse message that contains the sessionID and has the sppStatus set to "OK". If the mobile device was not able to update the requested MO(s) or node(s), the mobile device shall transmit to the OSU server an sppUpdateResponse message that contains the sessionID and with the sppStatus set to "Error occurred" and include an Error Code.

Step 10: If no error occurred in step 9, the OSU server transmits an sppExchangeComplete SOAP method to the mobile device that contains the sessionID and has the sppStatus set to "Exchange complete, release TLS connection".

If in step 9 the OSU server received an sppUpdateResponse message with sppStatus set to "Error occurred", it shall take one of the following actions:

- Transmit an sppExchangeComplete message that contains the sessionID and has the sppStatus set to "Exchange complete, release TLS connection" and an errorCode set to "Provisioning cannot be completed at this time". In this case, the OSU server is declaring that the policy provisioning/update process failed.
- Transmit an sppPostDevDataResponse message that contains the sessionID, has sppStatus set to "Provisioning complete, request sppUpdateResponse" and includes one or more addMO or updateNode element(s), retrying mobile device policy provisioning/update.

Step 12: The mobile device uses HTTPS to retrieve the trust anchor CA certificates for the subscription remediation server and policy server using the CertURLs in the PPS MO according to section 7.3.1.

Step 13: The mobile device releases the TLS session that was established in step 6.

Step 14: The mobile device disassociates from the Wi-Fi AN.

Step 15: If the subscription was established or updated successfully in step 8, the mobile device associates again with the same Wi-Fi AN, this time using the newly obtained subscription.

9. Management objects

The management objects (MOs) specified in sections 9.1 and 9.2 are defined in accordance with [38]. All MOs transferred between the mobile device and the subscription server shall be encoded according to [53].

9.1 PerProviderSubscription MO

This section defines the HS2.0 PerProviderSubscription Management Object (MO) that contains subscription and policy specific parameters supporting SP subscriptions. This MO is defined according to the principles specified in the OMA Mobile Device Management Tree and Descriptions Specification [38], and is therefore an OMA DM compliant MO, although it may be used with both the OMA DM and SOAP XML transfer protocols.

It shall be possible for the network to create and update the PerProviderSubscription MO in the mobile device using either the OMA DM or the SOAP XML protocol. The HS2.0 capable mobile device shall use HTTPS as the transport mechanism while connecting to the SP's subscription servers.

The mobile device shall use the provisioned PerProviderSubscription MO to select and authenticate a network in accordance with the identifiers, policies, credentials and related metadata contained in the MO.

The user shall not be provided a user interface on a mobile device that allows modification of the PerProviderSubscription MO.

Subscription and policy servers shall generate valid XML names for all the dynamic nodes in the PerProviderSubscription MO provisioned to the mobile device. These XML names shall be unique within the MO. Note: in Figure 58 and Figure 59 dynamic nodes are indicated by the notation "<X+>".

The implementation shall use case insensitive comparison when processing PerProviderSubscription MO node names. The case of the names shall be maintained, e.g., when uploading the MO to the server from which it was received.

The use of term "up to" in this section indicates the expectation that a leaf node should not be any longer than a specified limit and that all lengths up to that limit should/shall (depending on the context) be supported, e.g., "Mobile device implementations shall support Username strings up to 63 octets long".

9.1.1 Graphical representation

Figure 58 and Figure 59 shows the structure of the HS2.0 PerProviderSubscription MO.

The MO Identifier for this MO shall be: "urn:wfa:mo:hotspot2dot0-perprovidersubscription:1.0".

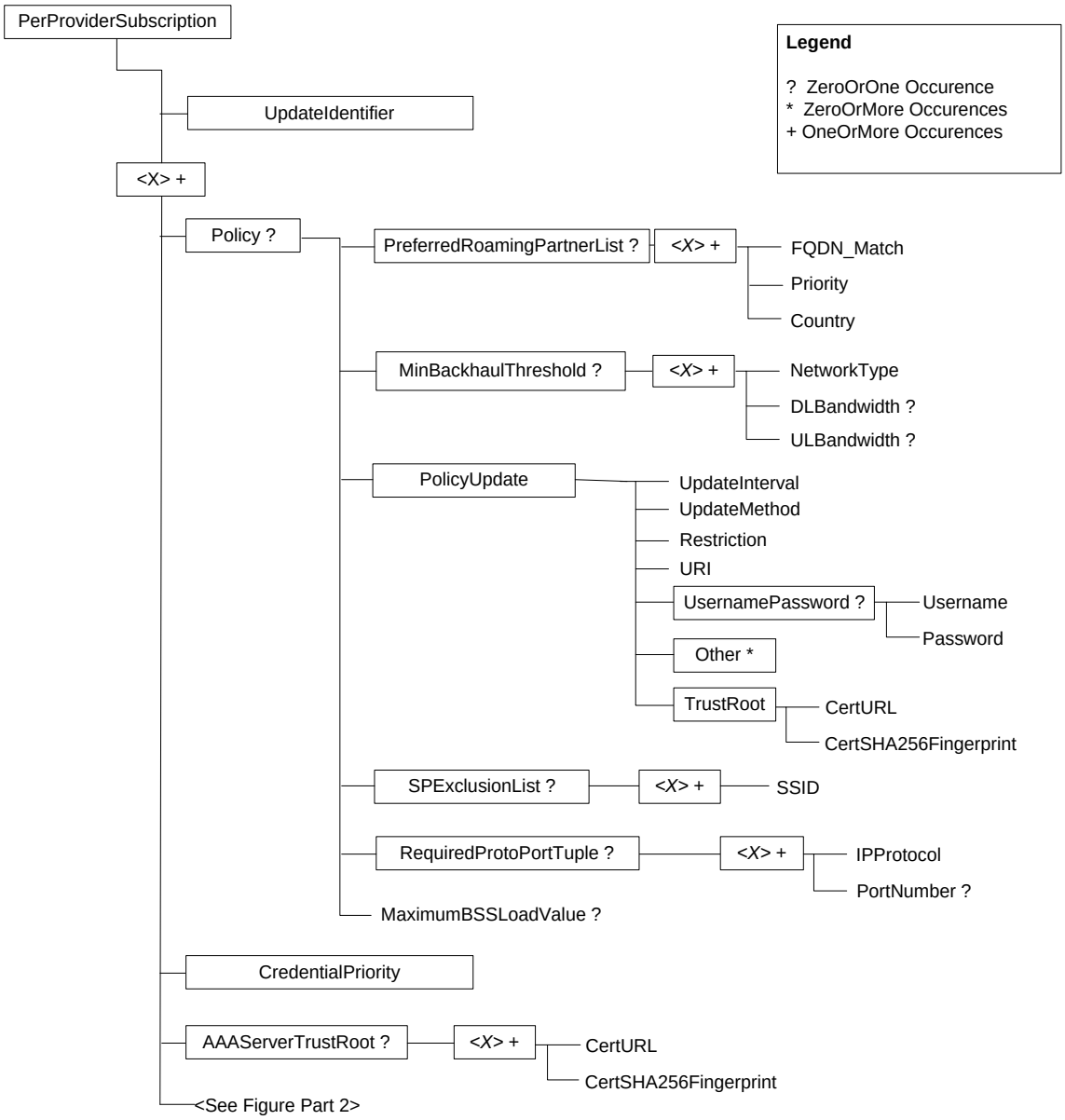


Figure 58: Graphical representation of PerProviderSubscription MO part 1

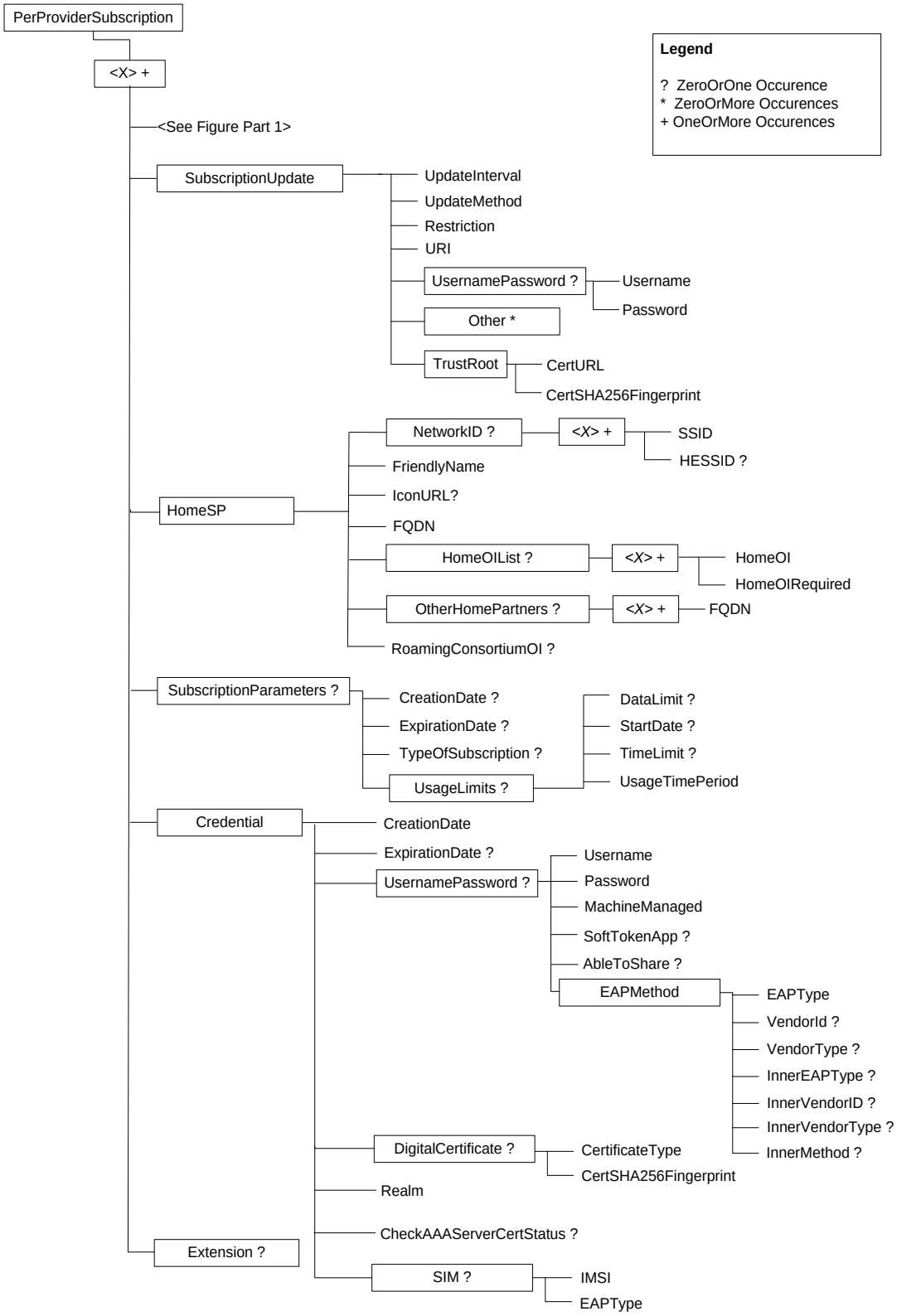


Figure 59: Graphical representation of PerProviderSubscription MO part 2

9.1.2 Node descriptions

All nodes in this MO are encoded using UTF-8 (see [14]), unless stated otherwise. In the following descriptions, the node Status indicates whether or not the mobile device needs to support the node. If the Status is “Required”, then the mobile device shall support that node, provided the parent node of this node is supported. If the Status is “Optional”, the mobile device is not required to support the node.

PerProviderSubscription

Status	Occurrence	Format	Access Types
Required	One	Node	Add, Delete, Get, Replace

This interior node acts as a container for one or more credentials and associated metadata and policy.

PerProviderSubscription/UpdateIdentifier

Status	Occurrence	Format	Access Types
Required	One	Int	Get, Replace

This leaf node specifies the Update Identifier for the PerProviderSubscription MO. The UpdateIdentifier is an un-signed, 16-bit integer set by a subscription server. The subscription servers should change the value of the UpdateIdentifier every time any node in the PerProviderSubscription MO is added, deleted or modified. The default value of this leaf node is zero, signifying un-provisioned values in the MO.

PerProviderSubscription/<X+>

Status	Occurrence	Format	Access Types
Required	OneOrMore	Node	Add, Delete, Get, Replace

This interior node contains the Home SP information, subscription policy, management and credential information.

PerProviderSubscription/<X+>/Policy

Status	Occurrence	Format	Access Types
Required	ZeroOrOne	Node	Add, Delete, Get, Replace

This interior node identifies the Home SP policy.

PerProviderSubscription/<X+>/Policy/PreferredRoamingPartnerList

Status	Occurrence	Format	Access Types
Required	ZeroOrOne	Node	Add, Delete, Get, Replace

This interior node identifies the roaming partner list. Any roaming partner not in this list has a default priority value of 128 (midrange). Mobile device implementations should support up to 256 roaming partner tuples, where each tuple is comprised of the following child leaf nodes: {FQDN_match, priority, Country}.

PerProviderSubscription/<X+>/Policy/PreferredRoamingPartnerList/<X+>

Status	Occurrence	Format	Access Types
Required	OneOrMore	Node	Add, Delete, Get, Replace

This interior node identifies one roaming partner SP in the roaming partner list.

PerProviderSubscription/<X+>/Policy/PreferredRoamingPartnerList/<X+>/FQDN_Match

Status	Occurrence	Format	Access Types
Required	One	Chr	Get, Replace

This leaf node specifies the FQDN of an SP in the roaming partner list. The value of the leaf node is the concatenated string "FQDN" || "," || {"includeSubdomains" | "exactMatch"} where ||= concatenation and | = logical OR.

For example, if the concatenated string is "example.com,includeSubdomains", a name “host.example.com” would match. But if the concatenated string is "example.com,exactMatch", then “host.example.com” would not be a match, as “example.com” is only a partial string.

Mobile device implementations shall be capable of supporting FQDNs up to 255 octets long for each occurrence of this leaf node. Note: the maximum length of the FQDN_Match leaf node is 273 octets long when the string “,includeSubdomains” is appended to the FQDN.

PerProviderSubscription/<X+>/Policy/PreferredRoamingPartnerList/<X+>/Priority

Status	Occurrence	Format	Access Types
Required	One	Int	Get, Replace

This leaf node specifies the priority of a roaming partner SP in the roaming partner list. The lower the value, the higher is the priority. The format of the priority is an 8 bit unsigned integer. A roaming partner not in this list has a default priority value of 128 (midrange).

PerProviderSubscription/<X+>/Policy/PreferredRoamingPartnerList/<X+>/Country

Status	Occurrence	Format	Access Types
Required	One	Chr	Add, Delete, Get, Replace

This leaf node allows SPs to state different preferences when the mobile is roamed internationally. The encoding shall be one or more, comma delimited (i.e., ",") ISO/IEC 3166-1 two character country strings or the country-independent value, "*". If there is a country-specific priority in one or more of the {FQDN_Match, Priority, Country} tuples, it shall override the default, country-independent priority. If the Country element is not present in the serving AP's Beacon or Probe Response frames, then only default behaviors (indicated by "*") are permitted.

Mobile device implementations shall support Country strings up to 600 octets long.

PerProviderSubscription/<X+>/Policy/MinBackhaulThreshold

Status	Occurrence	Format	Access Types
Optional	ZeroOrOne	Node	Add, Delete, Get, Replace

This optional interior node specifies the policy for the minimum threshold of available backhaul (WAN) parameters at a hotspot network. Policies may be set for uplink bandwidth, downlink bandwidth or both. If uplink or downlink load measurements are not available at a particular hotspot (see the WAN Metrics element in section 4.4), then the respective uplink or downlink policy is not evaluated at that hotspot.

When present, this policy should be evaluated by the mobile device for network selection, unless this policy prevents the mobile device from selecting any AN (e.g., cellular, Wi-Fi, etc.) or no hotspot has a backhaul parameter greater than or equal to the defined thresholds.
When enforced, this policy is only applicable for the initial association to an ESS.

PerProviderSubscription/<X+>/Policy/MinBackhaulThreshold/<X+>

Status	Occurrence	Format	Access Types
Required	OneOrMore	Node	Add, Delete, Get, Replace

This interior node identifies a container for the Minimum Backhaul Threshold policy for either the home or roaming network. A maximum of two instances of this interior node may be created.

PerProviderSubscription/<X+>/Policy/MinBackhaulThreshold/<X+>/NetworkType

Status	Occurrence	Format	Access Types
Required	One	Chr	Get, Replace

This leaf node specifies the type of network, "home" or "roaming", for which the policy applies.

PerProviderSubscription/<X+>/Policy/MinBackhaulThreshold/<X+>/DLBandwidth

Status	Occurrence	Format	Access Types
Optional	ZeroOrOne	Int	Add, Delete, Get, Replace

This leaf node specifies the minimum available downlink bandwidth (in kilobits per second) calculated based on the downlink speed and backhaul load. A mobile device shall attempt to join any hotspot having a backhaul available downlink bandwidth greater than or equal to the DLBandwidth leaf value. If no hotspot meets the defined threshold, the mobile device shall ignore the DLBandwidth leaf value when it selects a network.

The backhaul available downlink bandwidth is calculated as the Downlink Speed * (1 – Downlink Load/255), where the downlink speed and load parameters are drawn from the WAN Metrics element at that hotspot. The DLBandwidth leaf is a 32-bit unsigned integer representing units of kilobits per second.

PerProviderSubscription/<X+>/Policy/MinBackhaulThreshold/<X+>/ULBandwidth

Status	Occurrence	Format	Access Types
Optional	ZeroOrOne	Int	Add, Delete, Get, Replace

This leaf node specifies the minimum available uplink bandwidth (in kilobits per second) calculated according to the uplink speed and backhaul load. A mobile device is permitted to join

any hotspot that has a backhaul available uplink bandwidth greater than or equal to the ULBandwidth leaf value. If no hotspot complies with the defined threshold, the mobile device shall ignore the ULBandwidth leaf value when it selects a network.

The backhaul available uplink bandwidth is calculated as the Uplink Speed * (1 – Uplink Load / 255), where the uplink speed and load parameters are drawn from the value of the WAN Metrics element at that hotspot. The ULBandwidth leaf is a 32-bit unsigned integer that represents units of kilobits per second.

PerProviderSubscription/<X+>/Policy/PolicyUpdate

Status	Occurrence	Format	Access Types
Required	One	Node	Get, Replace

The parameters of this interior node identify the policy server along with metadata related to SP policy updates.

PerProviderSubscription/<X+>/Policy/PolicyUpdate/UpdateInterval

Status	Occurrence	Format	Access Types
Required	One	Int	Get, Replace

This leaf node specifies how often the mobile device shall check with the policy server for updates. The format of the UpdateInterval is a non-zero 32-bit unsigned integer and its value is specified in minutes. A value of 0xFFFFFFFF indicates that policy update is not applicable to this subscription. If the UpdateInterval is set to its maximum value, then the policy interior node is never updated.

PerProviderSubscription/<X+>/Policy/PolicyUpdate/UpdateMethod

Status	Occurrence	Format	Access Types
Required	One	Chr	Get, Replace

This leaf node specifies the method used to update the policy. Permitted values are "OMA-DM-ClientInitiated" and "SPP-ClientInitiated".

PerProviderSubscription/<X+>/Policy/PolicyUpdate/Restriction

Status	Occurrence	Format	Access Types
Required	One	Chr	Get, Replace

This leaf node specifies the hotspots at which the subscription is permitted to be updated. Permitted values are “HomeSP”, “RoamingPartner”, or “Unrestricted”.

If the value is set to:

- “RoamingPartner” then the mobile device may update its PerProviderSubscription MO when it is associated to a roaming partner’s HS2.0 compliant hotspot or its Home SP’s HS2.0 compliant hotspot.
- “Unrestricted” then the mobile device may update its PerProviderSubscription MO when connected to any WLAN connected to the public Internet.
- “HomeSP” then the mobile device shall update its policy only when it is connected to a hotspot operated by its Home SP.

PerProviderSubscription/<X+>/Policy/PolicyUpdate/URI

Status	Occurrence	Format	Access Types
Required	One	Chr	Get, Replace

This leaf node specifies the URI, formatted in accordance with [17], of the policy server. The mobile device shall use the procedures defined in [29] to connect with the policy server, as identified by this URI, to update the policy interior node data in this PerProviderSubscription MO.

The mobile device shall support URI strings at least 1023 octets long.

PerProviderSubscription/<X+>/Policy/PolicyUpdate/UsernamePassword

Status	Occurrence	Format	Access Types
Required	ZeroOrOne	Node	Add, Delete, Get, Replace

This interior node is a container for the username and password values to use when authenticating to the policy server identified by PerProviderSubscription/<X+>/Policy/PolicyUpdate/URI. If this node is not present, then the mobile device uses the credential from PerProviderSubscription/<X+>/Credential to authenticate to the policy server.

PerProviderSubscription/<X+>/ Policy/PolicyUpdate /UsernamePassword/Username

Status	Occurrence	Format	Access Types
Required	One	Chr	Get, Replace

This leaf node specifies the username formatted in accordance with [22]. Note that the realm is not included in this parameter. Mobile device implementations shall support Username strings up to 63 octets long.

PerProviderSubscription/<X+>/ Policy/PolicyUpdate /UsernamePassword/Password

Status	Occurrence	Format	Access Types
Required	One	b64	NoGet, Replace

This leaf node specifies the password. Mobile device implementations shall support a b64-encoded Password up to 255 octets long. Note that b64 decoding reduces the password's length by approximately 25%.

PerProviderSubscription/<X+>/Policy/PolicyUpdate/Other

Status	Occurrence	Format	Access Types
Optional	ZeroOrMore	Node	Add, Delete, Get, Replace

This leaf node specifies optional vendor specific methods that the Home SP can use to update policy.

PerProviderSubscription/<X+>/Policy/PolicyUpdate/TrustRoot

Status	Occurrence	Format	Access Types
Required	One	Node	Get, Replace

This interior node is a container for the certificate URL and certificate fingerprint (SHA-256).

PerProviderSubscription/<X+>/Policy/PolicyUpdate/TrustRoot/CertURL

Status	Occurrence	Format	Access Types
Required	One	Chr	Get, Replace

This leaf node provides the HTTPS URL at which the mobile device may retrieve a trust root. This trust root is used by the mobile device to validate the policy server's identity. The URL is formatted in accordance with [17]. Mobile device implementations shall support CertURL strings at least 1023 octets long.

PerProviderSubscription/<X+>/Policy/PolicyUpdate/TrustRoot/CertSHA256 Fingerprint

Status	Occurrence	Format	Access Types
Required	One	Chr	Get, Replace

This leaf node provides the SHA-256 fingerprint of the certificate located at CertURL. This field shall be formatted as follows (defined in XML regular expression syntax): [a-f0-9]{64}. In other words, 64 lowercase hexadecimal characters. The SHA-256 fingerprint is calculated over the X.509 ASN.1 DER encoded certificate.

PerProviderSubscription/<X+>/Policy/SPExclusionList

Status	Occurrence	Format	Access Types
Required	ZeroOrOne	Node	Add, Delete, Get, Replace

This interior node contains the SP exclusion list, which is a list of SSIDs that are not preferred by the Home SP. The mobile device shall not autonomously select a hotspot operated by a SP listed in the exclusion list. However the user may manually select such a network. Mobile device implementations should be capable of supporting an SPExclusionList having up to 128 SSID child leaf nodes.

PerProviderSubscription/<X+>/Policy/SPExclusionList/<X+>

Status	Occurrence	Format	Access Types
Required	OneOrMore	Chr	Get, Replace

This interior node identifies a container for the SPExclusionList.

PerProviderSubscription/<X+>/Policy/SPExclusionList/<X+>SSID

Status	Occurrence	Format	Access Types
Required	One	Chr	Get, Replace

This leaf node specifies the SSID of an SP in the Exclusion list and is formatted in accordance with the SSID information element as specified in [2].

PerProviderSubscription/<X+>/Policy/RequiredProtoPortTuple

Status	Occurrence	Format	Access Types
Optional	ZeroOrOne	Node	Add, Delete, Get, Replace

This internal node specifies the IP protocol and port number required by one or more operator supported application on the mobile device. That operator-supported application(s) on the mobile device is required to function properly.

The listed IP protocol and port number values are required to be present in the Connection Capability element with Status=1 (Open). The advertised information does not meet this policy criteria if it does not include all the protocol/port values listed here or if any of those values has Status value other than 1.

When present, this policy should be evaluated by the mobile device for network selection when connecting to a visited network, unless this policy prevents the mobile device from selecting any AN (e.g., cellular, Wi-Fi, etc.), or if the Connection Capability element (see section 4.5) is not advertised or has zero-length payload. When enforced, this policy is only applicable for the initial association to an ESS.

This policy is set by an operator to ensure that its own applications are functioning when attached to a visited HS 2.0 network.

Evaluation of this policy on home networks is unnecessary because they are expected to provide needed connectivity.

PerProviderSubscription/<X+>/Policy/RequiredProtoPortTuple/<X+>

Status	Occurrence	Format	Access Types
Required	OneOrMore	Node	Add, Delete, Get, Replace

This interior node identifies a container for the RequiredProtoPortTuple.
The operator may define one or more occurrence of this internal node, depending on the application(s).

PerProviderSubscription/<X+>/Policy/RequiredProtoPortTuple/<X+>/IPProtocol

Status	Occurrence	Format	Access Types
Required	One	Int	Get, Replace

This leaf specifies the IP Protocol type required by one or more operator-supported application(s) on the mobile device. This leaf node is an 8-bit unsigned integer. This leaf refers to the IP protocol field in IPv4 packets or the next header field in IPv6 packets.

PerProviderSubscription/<X+>/Policy/RequiredProtoPortTuple/<X+>/PortNumber

Status	Occurrence	Format	Access Types
Required	ZeroOrOne	Chr	Get, Replace

This leaf node, if present, specifies one or more port numbers used in conjunction with the IP Protocol leaf, required by one or more operator supported applications on the mobile device. Each port number in this leaf node is a 16-bit unsigned integer. Multiple, comma delimited (i.e.,

","") port numbers may be defined if required by the application(s); for example: "21, 22". Mobile device implementations shall support PortNumber strings up to 64 octets long.

PerProviderSubscription/<X+>/Policy/MaximumBSSLoadValue

Status	Occurrence	Format	Access Types
Optional	ZeroOrOne	Int	Add, Delete, Get, Replace

This optional node specifies the maximum acceptable BSS Load policy. The purpose of this node is to prevent a mobile device from joining an AP whose channel is overly congested with traffic and/or interference. This interior node shall only be evaluated when the mobile device is in the presence of a home network.

When present, this policy should be evaluated by the mobile device for network selection, unless this policy prevents the mobile device from selecting any AN (e.g., cellular, Wi-Fi, etc.), or no AP in the ESS have its BSSLoad less than the defined MaximumBSSLoadValue leaf threshold

A mobile device is permitted to join any AP in an ESS that has a channel utilization value (see subclause 8.4.2.30 of [2]) less than the MaximumBSSLoad value.

This policy is advisory: If the mobile device cannot find an AP with channel utilization less than the defined MaximumBSSLoadValue leaf, or if BSSLoad is not available, it may ignore this policy.

PerProviderSubscription/<X+>/CredentialPriority

Status	Occurrence	Format	Access Types
Required	One	Int	Get, Replace

This parameter specifies the credentialpriority for the subscription node. This policy node indicates the priority of the credential, when multiple credentials are included in a single PerProviderSubscription MO instance. The lower the value of priority, the higher is the credentialpriority. This leaf node is an 8-bit, unsigned integer.

PerProviderSubscription/<X+>/AAAServerTrustRoot

Status	Occurrence	Format	Access Types
Required	ZeroOrOne	Node	Add, Delete, Get, Replace

This interior node identifies the AAA server trust root(s).

PerProviderSubscription/<X+>/AAAServerTrustRoot/<X+>

Status	Occurrence	Format	Access Types
Required	OneOrMore	Node	Add, Delete, Get, Replace

This interior node is a container for the certificate URL and certificate fingerprint (SHA-256).

PerProviderSubscription/<X+>/AAAServerTrustRoot/<X+>/CertURL

Status	Occurrence	Format	Access Types
Required	One	Chr	Get, Replace

This leaf node provides the HTTTPS URL at which the mobile device may retrieve an AAA server trust root. This trust root is used by the mobile device to validate the AAA server's identity when

performing EAP authentication. The URL is formatted in accordance with [17]. The certificate identified by CertURL is an ASN.1 DER encoded X.509 certificate. Mobile device implementations shall support CertURL strings at least 1023 octets long.

PerProviderSubscription/<X+>/AAAServerTrustRoot/<X+>/CertSHA256Fingerprint

Status	Occurrence	Format	Access Types
Required	One	Chr	Get, Replace

This leaf node provides the SHA-256 fingerprint of the certificate located at CertURL. This field shall be formatted as follows (defined in XML regular expression syntax): [a-f0-9]{64}. In other words, 64 lowercase hexadecimal characters. The SHA-256 fingerprint is calculated over the X.509 ASN.1 DER encoded certificate.

PerProviderSubscription/<X+>/SubscriptionUpdate

Status	Occurrence	Format	Access Types
Required	One	Node	Get, Replace

The parameters of this interior node identify the subscription server along with metadata related to SP subscription updates and subscription remediation.

PerProviderSubscription/<X+>/SubscriptionUpdate/UpdateInterval

Status	Occurrence	Format	Access Types
Required	One	Int	Get, Replace

This leaf node specifies how often the mobile device shall check with the subscription server for updates. The format of the UpdateInterval is a non-zero 32-bit unsigned integer and its value is specified in minutes. A value of 0xFFFFFFFF indicates that subscription update is not applicable to this subscription; i.e., the subscription related parameters are never updated.

Note: a mobile device may check with a subscription server for an update of the PerProviderSubscription MO whenever its list of supported EAP methods changes. The purpose of the update is to inform a subscription server of additional supported EAP methods; in response, a subscription server may choose to modify information in the PerProviderSubscription MO and subsequently update that MO in the mobile device. Further note that the mobile device provides its DevInfo and DevDetail MOs to the subscription server as part of the PerProviderSubscription MO update process; the DevDetail MO contains the mobile device's supported EAPMethodList (see section 2**).**

PerProviderSubscription/<X+>/SubscriptionUpdate/UpdateMethod

Status	Occurrence	Format	Access Types
Required	One	Chr	Get, Replace

This leaf node specifies the method (e.g., OMA DM or SOAP XML) used to update the subscription. Permitted values are "OMA-DM-ClientInitiated" and "SPP-ClientInitiated".

PerProviderSubscription/<X+>/SubscriptionUpdate/Restriction

Status	Occurrence	Format	Access Types
Required	One	Chr	Get, Replace

This leaf node specifies the hotspots at which the subscription is permitted to be updated. Permitted values are “HomeSP”, “RoamingPartner”, or “Unrestricted”.

If the value is set to:

- “RoamingPartner” then the mobile device may update its PerProviderSubscription MO when associated to a roaming partner’s HS2.0 compliant hotspot or its Home SP’s HS2.0 compliant hotspot.
- “Unrestricted” then the mobile device may update its PerProviderSubscription MO when connected to any WLAN connected to the public Internet.
- “HomeSP” then the mobile device shall update its policy only when it is connected to a hotspot operated by its Home SP.

PerProviderSubscription/<X+>/SubscriptionUpdate/URI

Status	Occurrence	Format	Access Types
Required	One	Chr	Get, Replace

This leaf node specifies the URI, formatted in accordance with [17], of the subscription server. The mobile device shall use the procedures defined in [29], to connect with the subscription server, as identified by this URI, to update the subscription related parameters in this PerProviderSubscription MO or to perform subscription remediation. Mobile device implementations shall support URI strings at least 1023 octets long.

PerProviderSubscription/<X+>/SubscriptionUpdate/UsernamePassword

Status	Occurrence	Format	Access Types
Required	ZeroOrOne	Node	Add, Delete, Get, Replace

This interior node is a container for the username and password values to use when authenticating to the subscription server identified by PerProviderSubscription/<X+>/SubscriptionUpdate/URI. If this node is not present, then the mobile device uses the credential from PerProviderSubscription/<X+>/Credential to authenticate to the subscription server.

PerProviderSubscription/<X+>/SubscriptionUpdate/UsernamePassword/Username

Status	Occurrence	Format	Access Types
Required	One	Chr	Get, Replace

This leaf node specifies the username formatted in accordance with [22]. Note that the realm is not included in this parameter. Mobile device implementations shall support Username strings up to 63 octets long.

PerProviderSubscription/<X+>/SubscriptionUpdate/UsernamePassword/Password

Status	Occurrence	Format	Access Types
Required	One	b64	NoGet, Replace

This leaf node specifies the password. Mobile device implementations shall support a b64-encoded Password up to 255 octets long. Note that b64 decoding reduces the password's length by approximately 25%.

PerProviderSubscription/<X+>/SubscriptionUpdate/Other

Status	Occurrence	Format	Access Types
Optional	ZeroOrMore	Chr	Add, Delete, Get, Replace

This leaf node specifies optional vendor specific methods that the Home SP can use to update the subscription.

PerProviderSubscription/<X+>/SubscriptionUpdate/TrustRoot

Status	Occurrence	Format	Access Types
Required	One	Node	Get, Replace

This interior node is a container for the certificate URL and certificate fingerprint (SHA-256).

PerProviderSubscription/<X+>/SubscriptionUpdate/TrustRoot/CertURL

Status	Occurrence	Format	Access Types
Required	One	Chr	Get, Replace

This leaf node provides the HTTPS URL at which the mobile device may retrieve a trust root. This trust root is used by the mobile device to validate the subscription server's identity. The URL is formatted in accordance with [17]. Mobile device implementations shall support CertURL strings at least 1023 octets long.

PerProviderSubscription/<X+>/SubscriptionUpdate /TrustRoot/CertSHA256Fingerprint

Status	Occurrence	Format	Access Types
Required	One	Chr	Get, Replace

This leaf node provides the SHA-256 fingerprint of the certificate located at CertURL. This field shall be formatted as follows (defined in XML regular expression syntax): [a-f0-9]{64}. In other words, 64 lowercase hexadecimal characters. The SHA-256 fingerprint is calculated over the X.509 ASN.1 DER encoded certificate.

PerProviderSubscription/<X+>/HomeSP

Status	Occurrence	Format	Access Types
Required	One	Node	Get, Replace

This node is a container for the Home SP information for this subscription.

The HomeSP node provides three categories of information related to the Home SP (Note: Home SP is a defined term, see section 6.1.1):

- Three identifiers for the Home SP: the FriendlyName, IconURL and HomeOI leaf nodes.
- Three identifiers which can be used to determine whether a hotspot is a home or visited network: NetworkID, FQDN and OtherHomePartners leaf nodes.
- One identifier which indicates whether the Home SP is a member of a roaming consortium: the RoamingConsortiumOI leaf node.

Note: OIs advertised by a hotspot inform the mobile device whether authentication with that hotspot is possible (see subclause 10.24.8 of [2]). OIs advertised by a hotspot do not inform the mobile device whether that hotspot is a home or visited network.

PerProviderSubscription/<X+>/HomeSP/NetworkID

Status	Occurrence	Format	Access Types
Required	ZeroOrOne	Node	Add, Delete, Get, Replace

This node is a container for network identifier related information.

The following matching criteria apply when a {SSID, HESSID} duple drawn from {HomeSP/NetworkID/<X+>/SSID, HomeSP/NetworkID/<X+>/HESSID} and that has the same <X+> node value, also has a null value for HESSID. A hotspot network that advertises an SSID that matches an SSID value drawn from HomeSP/NetworkID/<X+>/SSID is deemed to be a home network.

The following matching criteria apply when a {SSID, HESSID} duple drawn from {HomeSP/NetworkID/<X+>/SSID, HomeSP/NetworkID/<X+>/HESSID}, has the same value of the <X+> node and a non-null value for HESSID. A hotspot network that advertises an SSID that matches an SSID value drawn from HomeSP/NetworkID/<X+>/SSID and advertises a HESSID that matches the HESSID value drawn from HomeSP/NetworkID/<X+>/HESSID is deemed to be a home network. If either the SSID or the HESSID advertised by the hotspot does not match, that Wi-Fi network is not deemed to be a home network.

Notes:

1. A mobile device selects home networks in preference to visited networks, as described in section 6.1.1.
2. A hotspot may be deemed to be a home network by several methods, including matching the {SSID, HESSID} duple, as described above.
3. Care should be taken when using a non-null SSID value in HomeSP/NetworkID/<X+>/SSID, since there is no assigned authority for issuing globally unique SSIDs. In other words, hotspot operators with different Wi-Fi networks might use the same SSID value. For this reason, and because provisioning {SSID, HESSID} duples can be more difficult to manage on a mobile device (especially if the home operator has a large number of SSIDs in their network), it is recommended that Home SP identify its hotspots using FQDN (i.e., HomeSP/FQDN).

PerProviderSubscription/<X+>/HomeSP/NetworkID/<X+>

Status	Occurrence	Format	Access Types
Required	OneOrMore	Node	Add, Delete, Get, Replace

This interior node is a container for the SSID, HESSID duple.

PerProviderSubscription/<X+>/HomeSP/NetworkID/<X+>/SSID

Status	Occurrence	Format	Access Types
Required	One	Chr	Get, Replace

This leaf node specifies the Wi-Fi AN name formatted in accordance with the SSID information element as specified in [2].

PerProviderSubscription/<X+>/HomeSP/NetworkID/<X+>/HESSID

Status	Occurrence	Format	Access Types
Required	ZeroOrOne	Chr	Add, Delete, Get, Replace

This leaf node specifies the homogeneous ESS identifier of the Wi-Fi AN as defined in subclause 8.4.2.94 of [2]. This leaf node, which is a MAC address, shall be formatted as follows (defined in XML regular expression syntax): [a-f0-9]{12}; in other words, using 12 concatenated lowercase hexadecimal characters.

PerProviderSubscription/<X+>/HomeSP/FriendlyName

Status	Occurrence	Format	Access Types
Required	One	Chr	Get, Replace

This leaf node specifies the friendly name of the Home SP. The FriendlyName is intended to be used by the mobile device for display purposes only.

Note: the Home SP friendly name is in the human language chosen by the SP.

PerProviderSubscription/<X+>/HomeSP/IconURL

Status	Occurrence	Format	Access Types
Optional	ZeroOrOne	Chr	Add, Delete, Get, Replace

This leaf node specifies the location of icon related to the Home SP. The URL is the network location from which the icon may be retrieved and may be used by the mobile device in an implementation dependent manner or used in a manner agreed between the SP and the mobile device manufacturer. The icon is intended to be used by the mobile device for display purposes only.

PerProviderSubscription/<X+>/HomeSP/FQDN

Status	Occurrence	Format	Access Types
Required	One	Chr	Get, Replace

This leaf node specifies the FQDN of the Home SP formatted in accordance with [5]. A mobile device uses the FQDN for AAA server certificate validation as described in section 7.3.3.2 and for policy server certificate validation as described in section 7.3.6.2. It also uses the FQDN to determine if the hotspot is operated by the mobile device's Home SP or a visited SP according to the procedures in section 6.1.1.

PerProviderSubscription/<X+>/HomeSP/HomeOIList

Status	Occurrence	Format	Access Types
Required	ZeroOrOne	Node	Add, Delete, Get, Replace

This optional interior node is a container for a list of organizational identifiers identifying the Home SP of which this provider is a member (see subclause 8.4.2.98 of [2]).

PerProviderSubscription/<X+>/HomeSP/HomeOIList/<X+>

Status	Occurrence	Format	Access Types
Required	OneOrMore	Node	Add, Delete, Get, Replace

This optional interior node is a container for a list of organizational identifiers identifying the Home SP of which this provider is a member (see subclause 8.4.2.98 of [2]).

This is to facilitate, for example, the following two use cases:

1. Situations in which an SP has different levels of subscription, e.g., gold, silver, bronze. When this happens, the realm/PLMN ID is identical for all customers (whether they have gold, silver or bronze subscriptions), but not all hotspots might be accessible by each subscription level. In that situation a hotspot may advertise support for a specific HomeOI (e.g., one in which the customer l has purchased a gold subscription to access) and the Home SP can provision the mobile device with a PerProviderSubscription MO that has the "gold" HomeOI and the HomeOIRequired value set to true.

2. Situations in which a SP has multiple home OIs. This supports the use case of one SP acquiring a 2nd SP. It also supports the use case of SPs partnering with one another, each wanting to be identified by the mobile as a home OI.

The OI_duple is defined as {HomeOI, HomeOIRequired} drawn from {HomeSP/HomeOIList/<X+>/HomeOI, HomeSP/HomeOIList/<X+>/HomeOIRequired}, having the same value of the <X+> node.

If there is exactly one OI_duple in the HomeOIList, matching rules are provided below in the descriptions for HomeSP/HomeOIList/<X+>/HomeOI and HomeSP/HomeOIList/<X+>/HomeOIRequired.

The following matching criteria apply when there are multiple OI_duples:

- All OI_duple having HomeOIRequired set to false be logically ORed. In other words, any HomeOI whose corresponding HomeOIRequired value is false matches an OI in the Roaming Consortium advertised by a hotspot operator. Consequently, successful authentication with that hotspot is possible.
- All OI_duple having HomeOIRequired set to true be logically ANDed. In other words, every HomeOI whose corresponding HomeOIRequired value is true shall match an OI in the Roaming Consortium advertised by a hotspot operator. Consequently, successful authentication with each such hotspot is possible.

Note: if the HomeOIList contains an OI_duple having HomeOIRequired set to TRUE, it causes the mobile device's connection manager to disregard all other OI_duples having HomeOIRequired set to FALSE.

PerProviderSubscription/<X+>/HomeSP/HomeOIList/<X+>/HomeOI

Status	Occurrence	Format	Access Types
Required	One	Chr	Get, Replace

This leaf node is the organizational identifier of the Home SP (see subclause 8.4.2.98 of [2]). The Chr encoding shall be lowercase ASCII hexadecimal characters only with no white space and no preceding “0x”, e.g., “506f9a”.

Per subclause 10.24.8 of [2], if the value of HomeOI matches an OI in the Roaming Consortium advertised by a hotspot operator, successful authentication with that hotspot is possible.

PerProviderSubscription/<X+>/HomeSP/HomeOIList/<X+>/HomeOIRequired

Status	Occurrence	Format	Access Types
Required	One	Bool	Get, Replace

This leaf node determines whether the HomeOI leaf node is required. If the value of HomeOIRequired is true, then the mobile device shall not attempt authentication unless the HomeOI value is included in the hotspot's Roaming Consortium (i.e., advertised in the hotspot's Roaming Consortium element or Roaming Consortium ANQP-element, see subclauses 8.4.2.98 and 8.4.4.7 of [2]).

Note: If the value of HomeOIRequired is false, then the mobile device should attempt authentication when the mobile's realm, PLMN ID, HomeOI(s) or RoamingConsortiumOI(s) is/are advertised by the AP.

PerProviderSubscription/<X+>/HomeSP/OtherHomePartners

Status	Occurrence	Format	Access Types
Required	ZeroOrOne	Node	Add, Delete, Get, Replace

This optional interior node is a container for a list of the HomeSP's partners that the mobile device shall regard as home hotspot operators. This container node and its child nodes facilitate the use case in which operators agree to be co-home providers via a commercial agreement or the use case in which one operator merges or acquires another. In these situations a mobile device needs to recognize more than one FQDN as belonging to its Home SP.

Note: a mobile device recognizes and gives Wi-Fi network selection preference to home operators, as described in section 6.1.1.

PerProviderSubscription/<X+>/HomeSP/OtherHomePartners/<X+>

Status	Occurrence	Format	Access Types
Required	OneOrMore	Node	Add, Delete, Get, Replace

This optional interior node is a container for a list of FQDNs.

PerProviderSubscription/<X+>/HomeSP/OtherHomePartners/<X+>/FQDN

Status	Occurrence	Format	Access Types
Required	One	Chr	Get, Replace

This leaf node specifies the FQDN of a partner operator of the Home SP that the mobile device shall regard as a home operator. It is formatted in accordance with [5]. Mobile device implementations shall support FQDNs up to 255 octets long for each occurrence of this leaf node.

PerProviderSubscription/<X+>/HomeSP/RoamingConsortiumOI

Status	Occurrence	Format	Access Types
Required	ZeroOrOne	Chr	Get, Replace

This leaf node, if present, contains one or more, comma delimited (i.e., ",") organizational identifiers identifying a roaming consortium of which this provider is a member (see subclauses 8.4.2.98 and 8.4.4.7 of [2]). Per clause 10.24.8 in [2], if an OI contained in the RoamingConsortiumOI leaf node matches an OI in the Roaming Consortium advertised by a hotspot operator (i.e., advertised in the hotspot's Roaming Consortium element or Roaming Consortium ANQP-element), successful authentication with that hotspot is possible. The Chr encoding for each OI shall be lowercase ASCII hexadecimal characters only with no white space and no preceding “0x”, e.g., “506f9a”. Mobile device implementations shall support RoamingConsortiumOI strings up to 255 octets long.

PerProviderSubscription/<X+>/SubscriptionParameters

Status	Occurrence	Format	Access Types
Optional	ZeroOrOne	Node	Add, Delete, Get, Replace

The parameters of this interior node identify the subscription parameters.

PerProviderSubscription/<X+>/SubscriptionParameters/CreationDate

Status	Occurrence	Format	Access Types
Optional	ZeroOrOne	Chr	Add, Delete, Get, Replace

This parameter specifies the date and time (UTC) that the PerProviderSubscription MO was initially provisioned to the mobile device. The date and time is formatted as YYYY-MM-DDTHH:MM:SSZ as per ISO 8601 [42] (combined date and time in UTC) where:

- YYYY is four digits for the year
- MM is two digits for the month, ranging from 01 to 12
- DD is two digits for the day of the month, ranging from 01 to 31
- HH is two digits for the hour of the day (24-hour clock), ranging from 00 to 23
- MM is two digits for the minute of the hour, ranging from 00 to 59
- SS is two digits for the second of the minute, ranging from 00 to 59

An example CreationDate is “2011-01-30T08:31:14Z”

PerProviderSubscription/<X+>/SubscriptionParameters/ExpirationDate

Status	Occurrence	Format	Access Types
Optional	ZeroOrOne	Chr	Add, Delete, Get, Replace

This parameter specifies the date and time (UTC) that the subscription will expire. After the expiration date, the mobile device should not expect to be able to successfully authenticate with the corresponding credentials. If this attribute is not present, there is no pre-determined expiration time and date. The format is the same as SubscriptionParameters/CreationDate.

PerProviderSubscription/<X+>/SubscriptionParameters/ TypeOfSubscription

Status	Occurrence	Format	Access Types
Optional	ZeroOrOne	Chr	Add, Delete, Get, Replace

This optional parameter specifies the type of subscription associated with the account. Subscription types are defined by the Home SP and are out of scope of this specification; example values are "Gold", "Silver" and "Bronze".

PerProviderSubscription/<X+>/SubscriptionParameters/UsageLimits

Status	Occurrence	Format	Access Types
Optional	ZeroOrOne	Node	Add, Delete, Get, Replace

This node specifies, if present, the accumulated usage limits for this subscription.

PerProviderSubscription/<X+>/SubscriptionParameters/UsageLimits/DataLimit

Status	Occurrence	Format	Access Types
Optional	ZeroOrOne	Int	Add, Delete, Get, Replace

This leaf node (in unsigned 32 bit integer format) specifies the cumulative data limit in megabytes for the UsageTimePeriod. If the value of this parameter is zero, there is unlimited data usage for this account. When the measured amount of data, has been transmitted between the mobile device and the network, approaches, reaches or passes this limit, the consequences are per the user's subscription, as described in section 6.1.

PerProviderSubscription/<X+>/SubscriptionParameters/UsageLimits/StartDate

Status	Occurrence	Format	Access Types
Optional	ZeroOrOne	Chr	Add, Delete, Get, Replace

This parameter specifies the date and time (UTC) at which usage statistics accumulation begins. The format is the same as SubscriptionParameters/CreationDate.

PerProviderSubscription/<X+>/SubscriptionParameters/UsageLimits/TimeLimit

Status	Occurrence	Format	Access Types
Optional	ZeroOrOne	Int	Add, Delete, Get, Replace

This leaf node (in unsigned 32 bit integer format) specifies the cumulative time limit in minutes for the UsageTimePeriod. If the value of this parameter is zero, there is unlimited time usage for this subscription. When the measured amount of time, used by the mobile device, approaches, reaches or passes this limit, the consequences are per the user's subscription, as described in section 6.4.

PerProviderSubscription/<X+>/SubscriptionParameters/UsageLimits/UsageTimePeriod

Status	Occurrence	Format	Access Types
Optional	One	Int	Add, Delete, Get, Replace

This parameter specifies a time period for usage statistics accumulation (in unsigned 32 bit integer format). A value of zero means that usage statistics are not accumulated on a periodic basis (e.g., a one-time limit for “pay as you go” - PAYG service). A non-zero value specifies the usage interval in minutes. After the expiry of this time period, the usage statistics are reset to zero. (e.g., the expiry of a billing period interval would reset the DataLimit).

The values of 1 to 31 are reserved to indicate that usage statistics (e.g., DataLimit) are monthly and reset on the day of the month indicated by the value (e.g., if the value is 10, the usages statistics are kept on a monthly basis and reset on the 10th of each month); the statistics should be reset at midnight in the mobile device's local timezone.

PerProviderSubscription/<X+>/Credential

Status	Occurrence	Format	Access Types
Required	One	Node	Get, Replace

This interior node is a container for the credentials of the subscription. The subscription server shall ensure that exactly one of the "UsernamePassword", "DigitalCertificate" or "SIM" interior nodes is present.

PerProviderSubscription/<X+>/Credential/CreationDate

Status	Occurrence	Format	Access Types
Optional	One	Chr	Get, Replace

This parameter specifies the date and time (UTC) that the credential was either created or last updated. The date and time is formatted as the "/SubscriptionParameters/CreationDate" above.

PerProviderSubscription/<X+>/Credential/ExpirationDate

Status	Occurrence	Format	Access Types
Required	ZeroOrOne	Chr	Add, Delete, Get, Replace

This parameter specifies the date and time (UTC) that the credentials will expire. This is an optional attribute; if not present, there is no pre-determined expiration time and date. The formatting of the ExpirationDate is the same as Subscription/CreationDate. Once the credentials expire the consequences are per the user's subscription, as described in section 6.4.

PerProviderSubscription/<X+>/Credential/UsernamePassword

Status	Occurrence	Format	Access Types
Required	ZeroOrOne	Node	Add, Delete, Get, Replace

This interior node is a container for the username and password values of the credential.

PerProviderSubscription/<X+>/Credential/UsernamePassword/Username

Status	Occurrence	Format	Access Types
Required	One	Chr	Get, Replace

This leaf node specifies the username, formatted in accordance with [22]. The realm is not included in this parameter; the realm is provided in PerProviderSubscription/Credential/Realm. Mobile device implementations shall support Username strings up to 63 octets long.

PerProviderSubscription/<X+>/Credential/UsernamePassword/Password

Status	Occurrence	Format	Access Types

Required	One	b64	NoGet, Replace

This leaf node specifies the password. Mobile device implementations shall support a b64-encoded Password up to 255 octets long. Note that b64 decoding reduces the password's length by approximately 25%.

PerProviderSubscription/<X+>/Credential/UsernamePassword/MachineManaged

Status	Occurrence	Format	Access Types
Required	One	Bool	Get, Replace

This parameter specifies whether the password is machine managed. If the SP has provided the username and password, the value of the parameter shall be true.

PerProviderSubscription/<X+>/Credential/UsernamePassword/SoftTokenApp

Status	Occurrence	Format	Access Types
Optional	ZeroOrOne	Chr	Get, Replace

This parameter specifies to the mobile the application that should be used to generate the password. When this leaf node is present, the password leaf node should have a null value.

PerProviderSubscription/<X+>/Credential/UsernamePassword/AbleToShare

Status	Occurrence	Format	Access Types
Optional	ZeroOrOne	Bool	Add, Delete, Get, Replace

This parameter indicates whether the credential is usable only on the mobile device which subscribed (i.e., the value of AbleToShare is false), or usable by other mobile devices of the user as well (i.e., the value of AbleToShare is true). When the AbleToShare leaf node is not present, credential sharing is not allowed by this subscription. When the value of AbleToShare is true, the entire PPS MO shall be provisioned to other mobile devices, not just the credential. The means by which the PPS MO is shared between multiple mobile devices on a common subscription is outside the scope of this specification.

PerProviderSubscription/<X+>/Credential/UsernamePassword/EAPMethod

Status	Occurrence	Format	Access Types
Required	One	Node	Get, Replace

This interior node is a container for information related to the EAPMethod.

PerProviderSubscription/<X+>/Credential/UsernamePassword/EAPMethod/EAPType

Status	Occurrence	Format	Access Types
Required	One	Int	Get, Replace

This leaf node contains the EAP Type value. The possible values are listed in the IANA EAP Registry in the Method Types section in [47].

PerProviderSubscription/<X+>/Credential/UsernamePassword/EAPMethod/VendorID

Status	Occurrence	Format	Access Types
Optional	ZeroOrOne	Int	Add, Delete, Get, Replace

Vendor-Id for an expanded EAP method, if used. The possible values are listed in [48].

PerProviderSubscription/<X+>/Credential/UsernamePassword/EAPMethod/VendorType

Status	Occurrence	Format	Access Types
Optional	ZeroOrOne	Int	Add, Delete, Get, Replace

Vendor-Type of the expanded EAP method, if used. These values are defined by the vendor identified by VendorId.

PerProviderSubscription/<X+>/Credential/UsernamePassword/EAPMethod/InnerEAPType

Status	Occurrence	Format	Access Types
Required	ZeroOrOne	Chr	Add, Delete, Get, Replace

This leaf node contains the EAP Type value for the inner EAP method, if used with this EAP method. The possible values are listed in the IANA EAP Registry in the Method Types section in [47].

PerProviderSubscription/<X+>/Credential/UsernamePassword/EAPMethod/InnerVendorID

Status	Occurrence	Format	Access Types
Optional	ZeroOrOne	Int	Add, Delete, Get, Replace

This leaf node identifies the Vendor-Id for an inner expanded EAP method, if used. The possible values are listed in [48].

PerProviderSubscription/<X+>/Credential/UsernamePassword/EAPMethod/InnerVendorType

Status	Occurrence	Format	Access Types
Optional	ZeroOrOne	Int	Add, Delete, Get, Replace

This leaf node identifies the Vendor-Type of the inner expanded EAP method, if used. These values are defined by the vendor identified by InnerVendorId.

PerProviderSubscription/<X+>/Credential/UsernamePassword/EAPMethod/InnerMethod

Status	Occurrence	Format	Access Types
Required	ZeroOrOne	Chr	Add, Delete, Get, Replace

This leaf node identifies an inner non-EAP method, if used with this EAP method. The permitted values are: PAP, CHAP, MS-CHAP and MS-CHAP-V2.

PerProviderSubscription/<X+>/Credential/DigitalCertificate

Status	Occurrence	Format	Access Types
Required	ZeroOrOne	Node	Add, Delete, Get, Replace

This interior node is a container for information related to the certificate credential.

PerProviderSubscription/<X+>/Credential/DigitalCertificate/CertificateType

Status	Occurrence	Format	Access Types
Required	One	Chr	Get, Replace

This parameter specifies the certificate type. The value is selected from the following enumerations: “802.1ar” or “x509v3”.

PerProviderSubscription/<X+>/Credential/DigitalCertificate/ CertSHA256Fingerprint

Status	Occurrence	Format	Access Types
Required	One	Chr	Get, Replace

This leaf node provides the SHA-256 fingerprint of the certificate credential for this subscription. This parameter specifies the Issuer Distinguished Name in the certificate credential. In conjunction with the certificate serial number, it uniquely identifies a certificate.

This field shall be formatted as follows (defined in XML regular expression syntax): [a-f0-9]{64}. In other words, 64 lowercase hexadecimal characters. The SHA-256 fingerprint is calculated over the X.509 ASN.1 DER encoded certificate.

PerProviderSubscription/<X+>/Credential/Realm

Status	Occurrence	Format	Access Types
Required	One	Chr	Get, Replace

This leaf node specifies the realm associated with the credential. A mobile device determines if it should be able to successfully authenticate to a hotspot by comparing the realms returned in the NAI Realm ANQP-element (see subclause 8.4.4.10 of [2]) with this realm. Mobile device implementations shall support Realms up to 253 octets long.

PerProviderSubscription/<X+>/Credential/CheckAAAServerCertStatus

Status	Occurrence	Format	Access Types
Required	ZeroOrOne	Bool	Add, Delete, Get, Replace

This leaf node specifies whether a mobile device shall check the AAA server certificate’s revocation status during EAP authentication (i.e., for EAP methods which employ an AAA server certificate). If this leaf node is present and set to a value of true, then the mobile device shall use OCSP, as defined in section 7.3.3.2, to check the AAA server certificate’s revocation status during EAP authentication. If this leaf node is not present or is present and its value is false, then

the mobile device shall not require the AAA server certificate's revocation status to be available at the time of authentication.

PerProviderSubscription/<X+>/Credential/SIM

Status	Occurrence	Format	Access Types
Optional	ZeroOrOne	Node	Add, Delete, Get, Replace

This interior node is a container for information related to the SIM credential. This node is optional for mobile devices not having a SIM card; this node is mandatory for mobile devices possessing a SIM card.

PerProviderSubscription/<X+>/Credential/SIM/IMSI

Status	Occurrence	Format	Access Types
Required	One	Chr	Get, Replace

This parameter specifies the IMSI (International Mobile device Subscriber Identity) [44].

A mobile device having SIM/USIM credentials determines if it should be able to successfully authenticate to a hotspot by comparing the MCC/MNC from its IMSI with the PLMN ID returned in the 3GPP Cellular Network ANQP-element (see subclause 8.4.4.11 of [2]). (A PLMN ID contains an MCC and MNC.)

Note: the IMSI is included so that the PerProviderSubscription MO can be bound to the correct SIM card when there is more than one SIM card in a mobile device.

PerProviderSubscription/<X+>/Credential/SIM/EAPType

Status	Occurrence	Format	Access Types
Required	One	Int	Get, Replace

This leaf node contains the EAP Type value. The possible values are listed in the IANA EAP Registry in the Method Types section in [47]. Only EAP-SIM, EAP-AKA, and EAP-AKA' methods are permitted.

PerProviderSubscription/<X+>/Extension

Status	Occurrence	Format	Access Types
Optional	ZeroOrOne	Node	Add, Delete, Get, Replace

The Extension is an interior node where vendor specific information about the PerProviderSubscription MO is placed. Usually the vendor extension is identified by vendor specific name under the extension node. The tree structure under the vendor identifier is not defined and may therefore include one or more non-standardized sub-trees.

9.2 DevDetail MO vendor specific extensions

This section defines the HS2.0 vendor specific extensions to the DevDetail standard OMA DM MO that contains data needed for the credential provisioning. The DevDetail MO is defined in [41].

The MO Identifier for this vendor-specific extension shall be: “urn:wfa:mo-ext:hotspot2dot0-devdetail-ext:1.0”.

9.2.1 Graphical representation

Figure 60 provides the structure of the WFA vendor specific extensions to the OMA DM DevDetail standard MO (see [41]). In Figure 60 the “DevDetail”, “Ext?”, “Bearer?” and “LrgObj” interior nodes are part of the defined OMA DM MO and shall not be duplicated; they are only shown in the figure to provide context. The ellipsis in Figure 60 indicates that other nodes, which are not relevant to these vendor-specific extensions, are not shown.

The mobile device shall generate a valid XML names for each dynamic node in the Wi-Fi extensions to the DevDetail MO sent to a subscription or policy server. Each XML name shall be unique within the MO. Note: dynamic nodes are indicated in Figure 60 by the notation "<X+>".

The Wi-Fi MO extension is defined in this section. This extension shall be uniquely identified by using following path in the DevDetail subtree: “./DevDetail/Ext/org.wi-fi/Wi-Fi”.

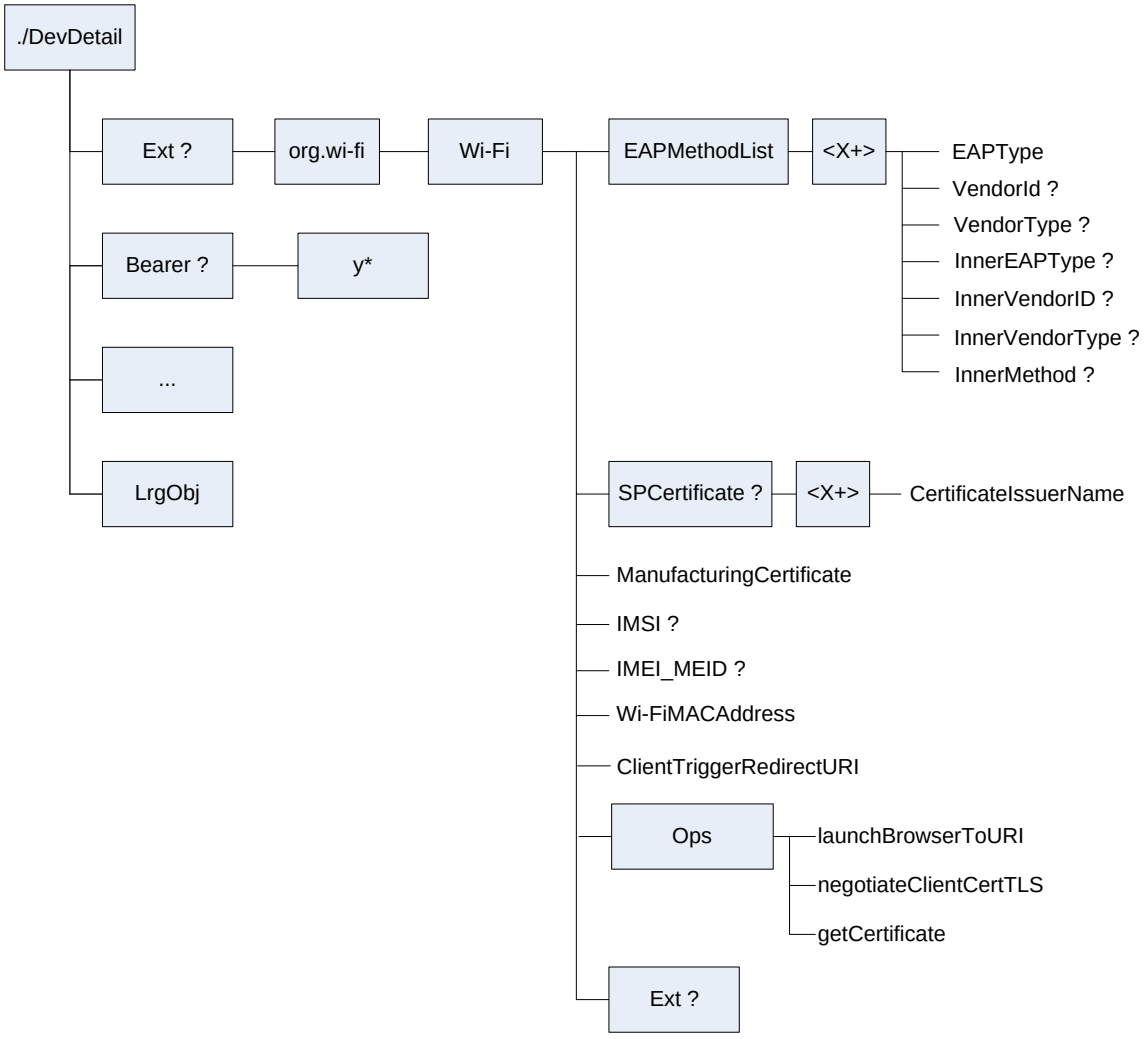


Figure 60: Graphical representation of the Vendor specific extension to the DevDetail Standard MO

9.2.2 Node descriptions

In order to preserve the capability for the mobile device to add the new nodes shown in Figure 60, the DevDetail node and its child x* node[8] in a HS2.0-compliant mobile device shall have its ACL set to "Add=*&Get=*" (cf. section 7.7.1.2 in [38]); furthermore, the mobile device shall not permit any server to change the value of the ACL on either of these nodes.

Only the Wi-Fi node is defined in this section and all nodes in this MO are encoded using UTF-8 (see [14]), unless stated otherwise. In the following descriptions, the node Status indicates whether or not the mobile device (OMA DM Client) needs to support the node. If the Status is “Required”, then the mobile device shall support that node, provided the parent node of this node is supported. If the Status is “Optional”, the mobile device is not required to support the node.

Wi-Fi

Status	Occurrence	Format	Access Types
Required	One	Node	Get

This interior node acts as a container for mobile device information related to Wi-Fi credential provisioning. In order to preserve the capability for the mobile device to provide this information in this MO to any SP with which the user has a subscription, the scope ((see section 6.2.3 in [38]) of the Wi-Fi node shall be permanent and it is default ACL shall be set to "Add=*&Get=*" (see section 7.7.1.2 in [38]); the mobile device shall not permit any server to change this ACL. The MO Identifier for this node shall be: “urn:wfa:mo-ext:hotspot2dot0-devdetail-ext:1.0”.

Wi-Fi/EAPMethodList

Status	Occurrence	Format	Access Types
Required	One	Node	Get

This interior node acts as a container for the list of EAPMethods supported by the mobile device and authorized for use on the Wi-Fi interface.

Wi-Fi/EAPMethodList/<X+>

Status	Occurrence	Format	Access Types
Required	OneOrMore	Node	Get

This interior node contains the EAP methods implemented by the mobile device.

Wi-Fi/EAPMethodList/<X+>/EAPType

Status	Occurrence	Format	Access Types
Required	One	Int	Get

This leaf node contains the EAP Type value. The possible values are listed in the IANA EAP Registry in the Method Types section in [47].

[8] The node named “x*” (see Figure 5 in [41]) and the node named “org.wi-fi” in Figure 60 are one and the same node.

Wi-Fi/EAPMethodList/<X+>/VendorID

Status	Occurrence	Format	Access Types
Optional	ZeroOrOne	Int	Get

This leaf node contains the Vendor-Id for an expanded EAP method, if used. The possible values are listed in [48].

Wi-Fi/EAPMethodList/<X+>/VendorType

Status	Occurrence	Format	Access Types
Optional	ZeroOrOne	Int	Get

This leaf node contains the Vendor-Type of the expanded EAP method, if used. These values are defined by the vendor identified by VendorId.

Wi-Fi/EAPMethodList/<X+>/InnerEAPType

Status	Occurrence	Format	Access Types
Required	ZeroOrOne	Chr	Get

This leaf node contains the EAP Type value for the inner EAP method, if used with this EAP method. The possible values are listed in the IANA EAP Registry in the Method Types section in [47].

Wi-Fi/EAPMethodList/<X+>/InnerVendorID

Status	Occurrence	Format	Access Types
Optional	ZeroOrOne	Int	Get

This leaf node identifies the Vendor-Id for an inner expanded EAP method, if used. The possible values are listed in [48].

Wi-Fi/EAPMethodList/<X+>/InnerVendorType

Status	Occurrence	Format	Access Types
Optional	ZeroOrOne	Int	Get

This leaf node identifies the Vendor-Type of the inner expanded EAP method, if used. These values are defined by the vendor identified by InnerVendorId.

Wi-Fi/EAPMethodList/<X+>/InnerMethod

Status	Occurrence	Format	Access Types
Required	ZeroOrOne	Chr	Get

This leaf node identifies an inner non-EAP method, if used with this EAP method (parent node). The permitted values are: PAP, CHAP, MS-CHAP and MS-CHAP-V2.

Wi-Fi/SPCertificate

Status	Occurrence	Format	Access Types
Optional	ZeroOrOne	Node	Get

This interior node acts as a container for mobile device information related to certificates provisioned by SPs.

Wi-Fi/SPCertificate/<x+>

Status	Occurrence	Format	Access Types
Required	OneOrMore	Node	Get

This interior node acts as a container for mobile device information related to certificates provisioned by SPs.

Wi-Fi/SPCertificate /<x+>/CertificateIssuerName

Status	Occurrence	Format	Access Types
Required	One	Chr	Get

This leaf node contains the issuer Distinguished Name (DN) Common Name (CN) attribute character string of the SP certificate.

Wi-Fi/ManufacturingCertificate

Status	Occurrence	Format	Access Types
Required	One	Bool	Get

This leaf node contains a boolean value that is set to true if the mobile device is in possession of an IEEE 802.1ar-compliant manufacturing certificate and is authorized to use that certificate for mobile device AAA authentication.

Wi-Fi/IMSI

Status	Occurrence	Format	Access Types
Optional	ZeroOrOne	Chr	Get

This leaf node contains the mobile device's IMSI(s); if the mobile device is not in possession of an IMSI, this leaf node is not present. The mobile device shall not provide the IMSI to an SP that did not issue the IMSI. This node is optional for mobile devices not having a SIM card; this node is mandatory for mobile devices possessing a SIM card.

Wi-Fi/IMEI_MEID

Status	Occurrence	Format	Access Types
Optional	ZeroOrOne	Chr	Get

If the device is a dual mode Wi-Fi/cellular device, this leaf node contains the mobile device's IMEI or MEID. If the device is Wi-Fi only, this leaf node is not present. The format of the value of this node shall be the following:

- imei:<IMEI>, where <IMEI> is equipment identifier for a 3GPP mobile device or dual mode 3GPP2/3GPP mobile device.
- meid:<MEID>, where <MEID> is equipment identifier for a 3GPP2 mobile device.

This node is optional for mobile devices not having a SIM card; this node is mandatory for mobile devices possessing a SIM card.

Wi-Fi/Wi-FiMACAddress

Status	Occurrence	Format	Access Types
Required	One	Chr	Get

This leaf node contains the MAC address associated with device's Wi-Fi interface. This leaf node shall be formatted as follows (defined in XML regular expression syntax): [a-f0-9]{12}; in other words, using 12 concatenated lowercase hexadecimal characters.

Wi-Fi/ClientTriggerRedirectURI

Status	Occurrence	Format	Access Types
Required	One	Chr	Get

This leaf node shall include a redirectURI formatted in accordance with [15], generated by the mobile device, which resolves to a resource internal to the mobile device. An OSU server involved in a browser interaction may redirect, at the end of the user exchange, the mobile device's web browser to the redirectURI. The device-internal event caused by retrieval of the redirectURI resource may be used to signal to the mobile device's connection manager to contact the OSU server by continuing the logical exchange (package 3) or starting a new session (package 1). The mobile device implementation to accomplish this is out-of-scope of this specification (see Annex E for an example flow showing messages internal to the mobile device). RedirectURI example http://127.0.0.1:12345/index.htm.

Note: If the mobile needs to support multiple concurrent sessions with multiple Servers, it shall send different redirectURI values of this leaf node to each server and contact the relevant server associated with the redirectURI it receives.

Wi-Fi/Ops

Status	Occurrence	Format	Access Types
Required	One	Node	Get

This node is a placeholder for operations that may be executed.

Wi-Fi/Ops/launchBrowserToURI

Status	Occurrence	Format	Access Types
Required	One	Chr	Exec

This leaf node is the target of an 'Exec' command to start the browser application or an integrated browser function to be used for online signup. The URI parameter is given by the data element of the Exec command. The mobile device shall launch the browser and shall open the listening port to receive the clientTriggerRedirectURI message.

Wi-Fi/Ops/negotiateClientCertTLS

Status	Occurrence	Format	Access Types
Required	One	Chr	Exec

This leaf node is the target of an ‘Exec’ command to requests the mobile device to re-negotiate the TLS session using a client certificate as defined in 8.3.2.3. The XML element <useClientCertTLS> as defined in A.3.2 is given by the data element of the Exec command.

Wi-Fi/Ops/getCertificate

Status	Occurrence	Format	Access Types
Required	One	Chr	Exec

This leaf node is the target of an ‘Exec’ command to start the certificate enrollment application to be used during online signup when a certificate is provisioned. The XML element <getCertificate> as defined in A.3.2 is given by the data element of the Exec command.

Wi-Fi/Ext

Status	Occurrence	Format	Access Types
Optional	ZeroOrOne	Node	Get

This interior node acts as a container for further extensions.

Annex A : Messages and definitions

A.1 OMA DM messages and definitions

The OMA DM protocol uses XML commands to perform management actions in the mobile device. This Annex describes the OMA DM commands and provides examples to further help developers. The examples are used for the development of subscription provisioning and policy management in an HS2.0 environment.

The mobile device shall be compliant with the OMA DM Protocol version 1.2 specification [39] and the protocol shall follow the managed objects specified in chapter 9. Subsection A.1.6 is normative and uses normative language to identify elements mandated by the current specifications. Other subsections are informative.

A.1.1 Generic Alert (informative)

The use of Generic Alert is described in the OMA DM Protocol specification [39]. Generic Alert is an OMA DM Alert with Alert code '1226' originating from the mobile device with or without reference to an MO. Figure 61 provides an example of a Generic Alert for subscription creation.

```
<Alert>
   <CmdID>2</CmdID>
   <Data>1226</Data><!-- Alert code for Generic Alert -->

   <Item>
      <Source><LocURI>./Subscription</LocURI></Source>
      <Meta>
         <Type xmlns="syncml:metinf">
            Reversed-Domain-Name:org.wi-fi.hotspot2dot0.SubscriptionCreation
            <!-- For Subscription Creation -->
         </Type>
      </Meta>
      <Data>
         <!-- For HS2.0 defined alerts the TNDS serialized DevDetail MO instance tree -->
      </Data>
   </Item>
</Alert>
```

Figure 61: Example OMA DM Generic Alert

In Figure 61 the text within the "<Type>" element is shown for example purposes only.

A.1.2 Exec command (informative)

The use of the Exec command is specified in the OMA DM Protocol specification [39]. The Exec command is used to execute the indicated process at the recipient. The Exec command also specifies the argument to be passed to the process. Figure 62 provides an example for Exec command for subscription creation.

```
<Exec>
   <CmdID>5</CmdID>
   <Item>
      <Target>
         <LocURI>./DevDetail/Ext/org.wi-fi/Wi-Fi/Ops/launchBrowsertoURI</LocURI> <!-- URI to the
         target node in the DM tree which describes the behavior of the Exec command -->
      </Target>
      <Data>https://onlinesignup.example.com?key=abcdef0123456789zz9876543210fedcba <!-- data is
      the node target of the Exec command (see section 9.2). The example in this case is the URL of
      the subscription server to be used by the browser.-->
      </Data>
   </Item>
</Exec>
```

Figure 62: Example OMA DM Exec command for subscription creation

A.1.3 Add command (informative)

The Add command is used to create nodes in the mobile device's management tree. The use of the Add command is specified in the OMA DM Representation Protocol specification [39]. An MO may be created with multiple Add commands, adding nodes one at a time or a single Add command with the MO serialized in XML form, following the OMA DM specification [53]. Figure 63 provides an example for Add Command for subscription creation.

```
<Add>
  <CmdID>4</CmdID>
  <Item>
    <Target>
       <LocURI>./Wi-Fi/SPFQDN1</LocURI>
    </Target>
    <Meta>
       <Format xmlns='syncml:metinf'>xml</Format>
       <Type xmlns='syncml:metinf'>application/vnd.syncml.dmtnds+xml</Type>
    </Meta>
    <Data>
       <!-- PerProviderSubscription MO serialized in XML form based on the OMA DM TNDS
Specification -->
    </Data>
  </Item>
</Add>
```

Figure 63: Example OMA DM Add command for subscription creation

A.1.4 Replace command (informative)

The use of the Replace command is specified in the OMA DM representation protocol [39]. The Replace command changes the values of nodes in the management tree, when the nodes already exist in the mobile device. If a node does not exist, it is not created. The OMA Management Standardized Objects document [41] specifies the operations of the Replace command.

Figure 64 provides an example for Replace command for subscription remediation.

```
<Replace>
   <CmdID>4</CmdID>
   <Item>
      <Target>
          <LocURI>./Wi-
          Fi/SPFQDN1/PerProviderSubscription/1/Credential/UsernamePassword/Password</LocURI>
      </Target>
      <Meta>
          <Format xmlns='syncml:metinf'>b64</Format>
      </Meta>
      <Data>
          bmV3cGFzc3dvcmQ= <!-- New password for this subscription -->
      </Data>
   </Item>
</Replace>
```

Figure 64: Example OMA DM Replace command for password replacement for an existing subscription

A.1.5 Status Management element (informative)

The Status Management element is used to return the status of protocol commands. The OMA Device Management Representation Protocol [49] specifies the status codes for DM commands. Figure 65 presents an example of returning the status for an Add command.

```
<Status>
   <MsgRef>2</MsgRef>
   <CmdID>1</CmdID>
   <CmdRef>3</CmdRef>
   <Cmd>Add</Cmd> <!-- Status is for the Add command -->
   <Data>200</Data> <!-- Command successfully completed -->
</Status>
```

Figure 65: Example Status for Add command

A.1.6 OMA DM elements (normative)

Table 12 specifies the usage of OMA DM XML elements used in the DM commands in case of subscription management. The normative language in this table mandates the procedures that operate with these elements.

Table 12: OMA DM elements

Command Elements	Common/Data Elements	Definition
Generic Alert	<Data>	Shall take the value of '1226' for Generic Alert as described in [39].
	<Item>	Item element, as specified in [39].
	<Item><Source><LocURI>	The Source element should identify the MO for which the Alert is originated. For example for subscription management, the Source element should identify the PerProviderSubscription MO instance URI (e.g., './Wi-Fi/SPFQDN1/PerProviderSubscription/001/SubscriptionUpdate'). For policy update the Source element should identify the URI of the policy node subtree (e.g.,'./Wi-Fi/SPFQDN1/PerProviderSubscription/001/Policy').
	<Item><Type>	Alert Type is used to specify the purpose of the alert. The server identifies the purpose and initiates appropriate management action based on the Alert Type. The following Alert Types shall be used for Wi-Fi HS2.0 subscription management. Subscription registration: org.wi-fi.hotspot2dot0.SubscriptionCreation Subscription provisioning (used when mobile device has SIM): org.wi-fi.hotspot2dot0.SubscriptionProvisioning Certificate enrollment: org.wi-fi.hotspot2dot0.CertificateEnrollment Subscription remediation: org.wi-fi.hotspot2dot0.SubscriptionRemediation Policy update for ND&S: org.wi-fi.hotspot2dot0.PolicyUpdate
	<Item><Data>	Generic alerts defined in this specification shall include the serialized DevDetail MO instance, according to [53].
Exec	<Target><LocURI>	The Target element identifies the URI of the node in the Device DM tree that is targeted by the command.
	<Data>	As defined in the operations nodes in section 9.2. For example, for launchBrowserToURI, the Data element identifies the URL of the web server that the browser will use.
Add	Meta	When Add command is used to create a new node in the management tree, the Meta information identifies the format and type of the node based on the definition of the node in the MO specification. When the Add command is used to create an entire subtree, the Meta information is as specified in [38].
	<Target><LocURI>	When the Add command is used to create a new node, the Target element identifies the URI of the new node to be created. When the Add command is used to create a subtree, the Target identifies the root of the subtree.
	Data	When the Add command is used to create a new node, the Data element carries the value of the Target node in the Add command. When the Add command is used to create a subtree, the Data element carries the entire subtree in serialized form, based on [38].

Command Elements	Common/Data Elements	Definition
Replace	<Target><LocURI>	When the Replace command is used to change the value of a single node, the Target element identifies the URI of the node. When the Replace command is used to replace an entire subtree, the Target identifies the root of the subtree.
	Meta	When the Replace command is performed on a single node in the management tree, the Meta information identifies the Format and Type of the node based on the definition of the node in the MO specification. When the Replace command is used to replace an entire subtree, the Meta information is as specified in [38].
	<Data>	When the Replace command is used to change the value of a single node, the Data element carries the new value. When the Replace command is used to change an entire subtree, the Data element carries the entire subtree in serialized form, based on [38].
Status	<Item><Source><LocURI>	When used to report the status of a command, the <Source><LocURI> element identifies the target URI used in the command.
	<Cmd>	Name of the DM Command for which status is indicated.
	<Data>	The Data element contains a valid result code, as specified in [38].

A.2 OMA DM messages – examples (informative)

A.2.1 DM package 1 (mobile device to server)

The OMA DM client initiates a session by sending package 1 to the server. A package is a logical grouping of DM messages (see [38]). In this example package 1 includes the DM Sync header specified in [38], an Alert indicating a client initiated session, a Replace command with DevInfo, and a Generic Alert with Alert Type indicating the reason for the session (e.g., subscription creation), and the DevDetail MO instance in the Generic Alert data element. A <Final/> element is included to complete the package.

```
<SyncML xmlns='SYNCML:SYNCML1.2'>
   <SyncHdr>
      <VerDTD>1.2</VerDTD>
      <VerProto>DM/1.2</VerProto>
      <SessionID>1</SessionID>
      <MsgID>1</MsgID>
      <Target><LocURI>http://www.serviceprovider.org/subscription-server</LocURI></Target>
      <Source><LocURI>IMEI:493005100592800</LocURI></Source>
      <Meta><MaxMsgSize xmlns="syncml:metinf">5000</MaxMsgSize></Meta>
       <!-- Maximum message size for the client, i.e., 5000 bytes -->
   </SyncHdr>
   <SyncBody>
   <Alert>
      <CmdID>1</CmdID>
      <Data>1201</Data> <!-- Alert Code for Client-initiated session -->
   </Alert>
   <Alert> <!-- Generic Alert -->
      <CmdID>2</CmdID>
      <Data>1226</Data><!-- Alert code for Generic Alert -->
      <Item>
         <Source><LocURI>./Wi-Fi/SPFQDN1/PerProviderSubscription</LocURI></Source>
         <Meta>
            <Type xmlns="syncml:metinf">
               Reversed-Domain-Name:org.wi-fi.hotspot2dot0.SubscriptionCreation
               <!-- For Subscription Creation -->
            </Type>
            <Format xmlns="syncml:metinf">xml</Format>
         </Meta>
         <Data><!-- TNDS serialized DevDetail MO instance tree -->
             <![CDATA[<MgmtTree xmlns='syncml:dmddf1.2'>
               <VerDTD>1.2</VerDTD>
               <Node><NodeName>DevDetail</NodeName>
                 <Node><NodeName>Ext</NodeName>
                   <Node><NodeName>org.wi-fi</NodeName>
                      <RTProperties><Type><DDFName>urn:wfa:mo-ext:hotspot2dot0-devdetail-
                           ext:1.0</DDFName></Type></RTProperties>
                      <Node><NodeName>Wi-Fi</NodeName><Node>
                         <NodeName>EAPMethodList</NodeName>
                            <Node><NodeName>Method001</NodeName> <!-- EAP-SIM -->
                               <Node><NodeName>EAPMethod</NodeName><Value>18</Value></Node>
                            </Node>
                            <!-- the rest of the EAP Methods not shown for the sake of brevity -
   ->
                         </Node> <!-- End of EAPMethodList -->
                         <!-- the rest of the MO's nodes not shown for the sake of brevity -->
                      </Node> <!-- End of Wi-Fi -->
                   </Node> <!-- End of org.wi-fi -->
                 </Node> <!-- End of Ext -->
                 <!-- the rest of DevDetail MO's nodes not shown for the sake of brevity -->
               </Node>
             </MgmtTree>]]>
        </Data>
      </Item>
   </Alert>
   <Replace>
      <CmdID>3</CmdID>
      <Item> <!-- Sending DevInfo -->
         <Source><LocURI>./DevInfo/DevId</LocURI></Source>
         <Meta><Format xmlns='syncml:metinf'>Chr</Format>
            <Type xmlns='syncml:metinf'>text/plain</Type></Meta>
         <Data>IMEI:493005100592800</Data>
      </Item>
      <Item>
         <Source><LocURI>./DevInfo/Man</LocURI></Source>
         <Meta>
            <Format xmlns='syncml:metinf'>Chr</Format>
            <Type xmlns='syncml:metinf'>text/plain</Type>
         </Meta>
         <Data>Mobile Device Factory, Inc.</Data>
      </Item>
      <Item>
         <!-- Repeated for ./DevInfo/Mod, ./DevInfo/DmV, ./DevInfo/Lang -->
      </Item>
   </Replace>
   <Final/>
</SyncBody>
</SyncML>
```

Figure 66: Example OMA DM package 1

A.2.2 DM package 2 (server to mobile device)

Package 2 is sent from the server to the mobile device in response to the messages in package 1. The server may include additional DM messages in package 2. In the example in Figure 67 package 2 includes the DM Sync header as specified in [39]. The DM Sync body includes the status of each parameter sent in package 1 (DM sync header, Alert, Generic Alert and / or DevInfo Replace command). The server also includes in this package an Exec command to launch the browser to a specified URI given by the Exec command. A <Final/> element completes the package.

```
<SyncML xmlns='SYNCML:SYNCML1.2'>
   <SyncHdr>
      <VerDTD>1.2</VerDTD>
      <VerProto>DM/1.2</VerProto>
      <SessionID>1</SessionID>
      <MsgID>1</MsgID>
      <Target>
         <LocURI>IMEI:493005100592800</LocURI>
      </Target>
      <Source>
         <LocURI>http://www.serviceprovider.org/subscription-server</LocURI>
      </Source>
   </SyncHdr>
   <SyncBody>
      <Status> <!-- Status to the Header -->
         <MsgRef>1</MsgRef><CmdRef>0</CmdRef>
         <Cmd>SyncHdr</Cmd>
         <CmdID>1</CmdID>
         <TargetRef> http://www.serviceprovider.org/subscription-server</TargetRef>
         <SourceRef>IMEI:493005100592800</SourceRef>
         <!-- Authenticated for the session -->
         <Data>212</Data>
      </Status>
      <Status>   <!-- Status to the Alert -->
         <MsgRef>1</MsgRef><CmdRef>1</CmdRef>
         <CmdID>2</CmdID>
         <Cmd>Alert</Cmd>
         <Data>200</Data> <!-- OK -->
      </Status>
      <Status>  <!—- Status to the Generic Alert -->
         <MsgRef>1</MsgRef><CmdRef>2</CmdRef>
         <CmdID>3</CmdID>
         <Cmd>Alert</Cmd>
         <Data>200</Data> <!-- OK -->
      </Status>
      <Status> <!—- Status to the DevInfo and DevDetail -->
         <MsgRef>1</MsgRef><CmdRef>3</CmdRef>
         <CmdID>4</CmdID>
         <Cmd>Replace</Cmd>
         <Data>200</Data> <!-- OK -->
      </Status>
      <Exec>
         <CmdID>5</CmdID>
         <Item>
            <Target>
               <LocURI>./DevDetail/Ext/org.wi-fi/Wi-Fi/Ops/launchBrowserToURI</LocURI>
               <!-- URI to the node in the DM tree to be the executed process -->
            </Target>
            <Data> https://onlinesignup.example.com?key=abcdef0123456789zz9876543210fedcba
               <!-- The URL to be used by the browser for subscription selection -->
            </Data>
         </Item>
      </Exec>
      <Final/>
   </SyncBody>
</SyncML>
```

Figure 67: Example OMA DM package 2

A.2.3 DM package 3 (mobile device to server)

Package 3 is sent from the mobile device to the server. In the example below, package 3 includes DM sync header as specified in [39]. The status of the DM sync header and Exec command which were sent in package 2 are included in the DM sync body. In addition, a Generic Alert is included with the Alert Type specified for PerProviderSubscription MO creation. A <Final/> element is included to complete the package.

```
<SyncML xmlns='SYNCML:SYNCML1.2'>
   <SyncHdr>
      <VerDTD>1.2</VerDTD>
      <VerProto>DM/1.2</VerProto>
      <SessionID>1</SessionID>
      <MsgID>2</MsgID>
      <Target>
         <LocURI>http://www.serviceprovider.org/subscription-server<LocURI>
      </Target>
      <Source>
         <LocURI>IMEI:493005100592800</LocURI>
      </Source>
   </SyncHdr>
   <SyncBody>
      <Status>
         <MsgRef>1</MsgRef><CmdRef>0</CmdRef>
         <CmdID>1</CmdID>
         <Cmd>SyncHdr</Cmd>
         <Data>212</Data>
      </Status>
      <Status>
         <MsgRef>1</MsgRef>
         <CmdRef>5</CmdRef>
         <CmdID>2</CmdID>
         <Cmd>Exec</Cmd>
         <TargetRef>./DevDetail/Ext/org.wi-fi/Wi-Fi/Ops/launchBrowserToURI</TargetRef>
         <Data>200</Data><!-- The Exec command are completed successfully. OK -->
      </Status>
      <Final/>
   </SyncBody>
</SyncML>
```

Figure 68: Example OMA DM package 3

A.2.4 DM package 4 – Exec:getCertificate (server to mobile device)

Package 4 is sent from the server to the mobile device to report the status of commands in package 3 and additional management commands. It is used for closing a session by including only the <Final> element in DM sync body. To close a session, the server does not include any new commands requiring response from the mobile device (see [39]). The status of the DM sync header and Alerts received in package 3 is returned. If any new commands are included in package 4, that require response from the mobile device, the protocol restarts from package 3 (see [39]).

In the example shown in Figure 69 the DM package 4 contains an Exec command to instruct the device to initiate Certificate enrollment.

```
<SyncML xmlns='SYNCML:SYNCML1.2'>
   <SyncHdr>
      <VerDTD>1.2</VerDTD>
      <VerProto>DM/1.2</VerProto>
      <SessionID>1</SessionID>
      <MsgID>3</MsgID>
      <Target>
         <LocURI>IMEI:493005100592800</LocURI>
      </Target>
      <Source>
         <LocURI>http://www.serviceprovider.org/subscription-server</LocURI>
      </Source>
   </SyncHdr>
   <SyncBody>
      <Status>
         <MsgRef>2</MsgRef><CmdRef>0</CmdRef>
         <CmdID>1</CmdID>
         <Cmd>SyncHdr</Cmd>
         <Data>212</Data>
      </Status>
      <Exec>
         <CmdID>5</CmdID>
         <Item>
            <Target>
               <LocURI>./DevDetail/Ext/org.wi-fi/Wi-Fi/Ops/getCertificate</LocURI>
               <!-- URI to the node in the DM tree to be executed -->
            </Target>
            <Data> <!-- The getCertificate XML element -->
              <![CDATA[ <spp:getCertificate
               xmlns:spp="http://www.wi-fi.org/specifications/hotspot2dot0/v1.0/spp"
               enrollmentProtocol="EST">
               <spp:enrollmentServerURI> https://www.example.com/.well-known/est
                  </spp:enrollmentServerURI>
               <spp:estUserID>tempest-user</spp:estUserID>
               <spp:estPassword>ZVN0d2l0aGhTMi4w</spp:estPassword>
               <!-- base64 encoded for "eStwithhS2.0" -->
              </spp:getCertificate> ]]>
            </Data>
         </Item>
      </Exec>
      <Final/>
   </SyncBody>
</SyncML>
```

Figure 69: Example OMA DM package 4 – Exec:getCertificate

A.2.5 DM package 3 (mobile device to server)

Figure 70 show an example of the DM Package 3 that includes the status response to the Exec command instructing the mobile device to initiate Certificate enrolment.

```
<SyncML xmlns='SYNCML:SYNCML1.2'>
   <SyncHdr>
      <VerDTD>1.2</VerDTD>
      <VerProto>DM/1.2</VerProto>
      <SessionID>1</SessionID>
      <MsgID>4</MsgID>
      <Target>
         <LocURI>http://www.serviceprovider.org/subscription-server<LocURI>
      </Target>
      <Source>
         <LocURI>IMEI:493005100592800</LocURI>
      </Source>
   </SyncHdr>
   <SyncBody>
      <Status>
         <MsgRef>3</MsgRef><CmdRef>0</CmdRef>
         <CmdID>1</CmdID>
         <Cmd>SyncHdr</Cmd>
         <Data>212</Data>
      </Status>
      <Status>
         <MsgRef>3</MsgRef>
         <CmdRef>5</CmdRef>
         <CmdID>3</CmdID>
         <Cmd>Exec</Cmd>
         <TargetRef>./DevDetail/Ext/org.wi-fi/Wi-Fi/Ops/getCertificate</TargetRef>
         <Data>200</Data><!-- The Exec command completed successfully. OK -->
      </Status>
   <Final/>
   </SyncBody>
</SyncML>
```

Figure 70: Example OMA DM Package 3

A.2.6 DM package 4 (server to mobile device)

Package 4 is sent from the server to the mobile device to return the status of commands in package 3 and additional management commands. It is used for closing a session by including only the <Final/> element in the DM sync body. To close a session, the server should not include any new commands requiring response from the mobile device (see [39]. Status of DM sync header and Alerts received in package 3 shall be returned. If any new commands are included in package 4, that require response from the mobile device, the protocol restarts from package 3 (see [39].

Figure 71 show an example of the DM package 4 that includes the Add command to create the PerProviderSubscription MO instance.

```
<SyncML xmlns='SYNCML:SYNCML1.2'>
   <SyncHdr>
      <VerDTD>1.2</VerDTD>
      <VerProto>DM/1.2</VerProto>
      <SessionID>1</SessionID>
      <MsgID>2</MsgID>
      <Target>
      <LocURI>IMEI:493005100592800</LocURI>
      </Target>
      <Source>
      <LocURI>http://www.serviceprovider.org/subscription-server</LocURI>
      </Source>
   </SyncHdr>
   <SyncBody>
      <Status>
         <MsgRef>4</MsgRef><CmdRef>0</CmdRef>
         <CmdID>1</CmdID>
         <Cmd>SyncHdr</Cmd>
         <Data>212</Data>
      </Status>
      <!-- Sending the PerProviderSubscription MO in Serialized form -->
      <Add>
      <CmdID>3</CmdID>
         <Item>
            <Target>
               <LocURI>./Wi-Fi/SPFQDN1</LocURI>
            </Target>
            <Meta>
               <Format xmlns='syncml:metinf'>xml</Format>
               <Type xmlns='syncml:metinf'>application/vnd.syncml.dmtnds+xml</Type>
            </Meta>
            <Data>
               <!-- Data element carrying the serialized PerProviderSubscription MO -->
            </Data>
         </Item>
      </Add>
      <Final/>
   </SyncBody>
</SyncML>
```

Figure 71: Example OMA DM package 4

A.3 SOAP XML messages and definitions

SOAP methods defined in this Annex shall be compliant with SOAPv1.2, as specified in [43].

Note that not all SPP messages are SOAP methods (e.g., the getCertificate XML element can also be an XML instance document in OMA DM message exchanges)

A.3.1 The sppPostDevData SOAP method

Figure 72 shows the components of the sppPostDevData SOAP method. Figure 73 provides an example sppPostDevData message. Table 13 contains definitions of the sppPostDevData message elements and attributes.

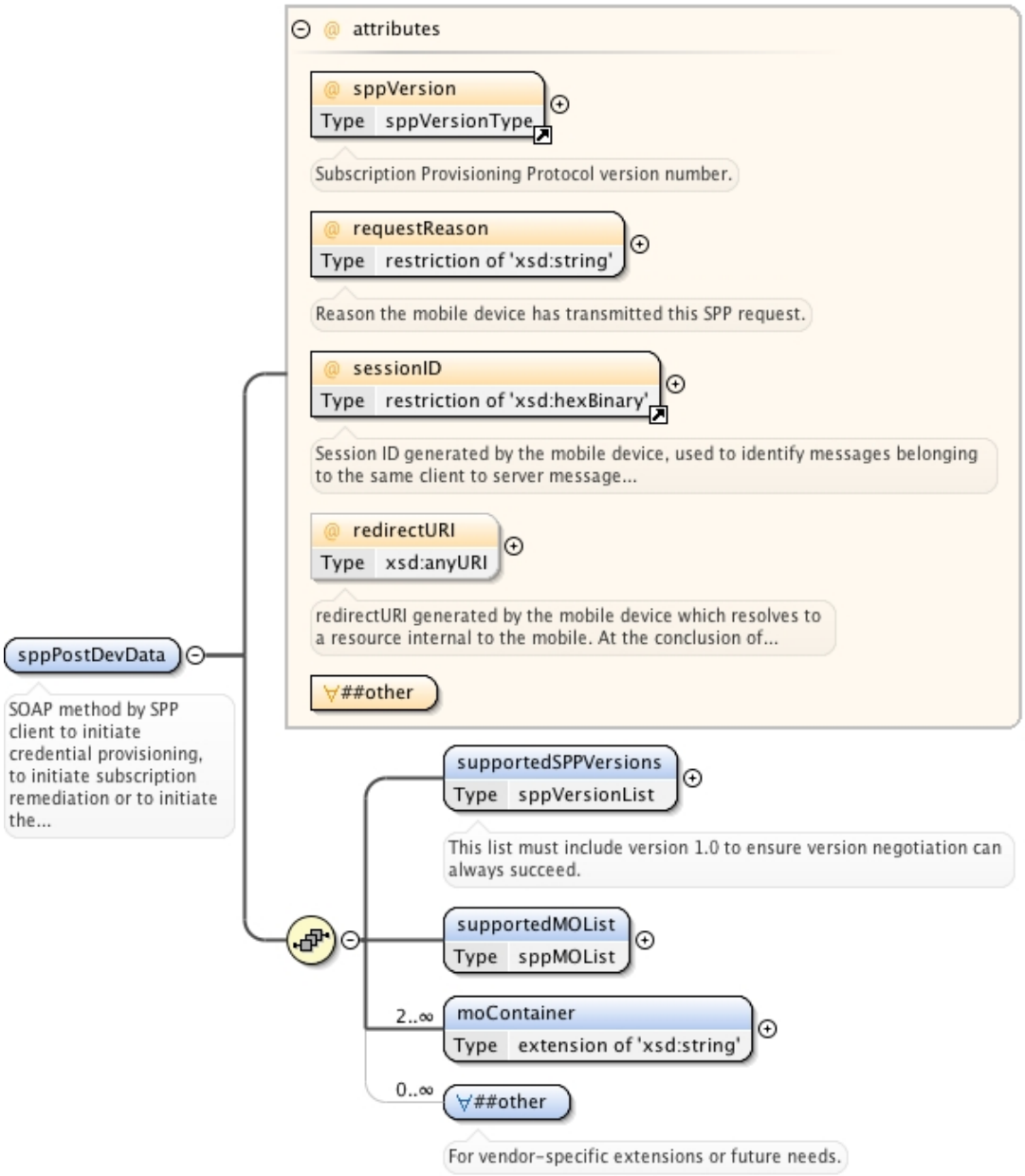


Figure 72: Diagram of the sppPostDevData SOAP method

```
<s12:Envelope   xmlns:s12="http://www.w3.org/2003/05/soap-envelope"
    xmlns:spp="http://www.wi-fi.org/specifications/hotspot2dot0/v1.0/spp">
  <s12:Body>
       <spp:sppPostDevData spp:sppVersion="1.0" requestReason="User input completed"
           spp:sessionID="abcdef0123456789987654321 0fedcba"
           redirectURI="http://127.0.0.1:12345/index.htm">
     <spp:supportedSPPVersions>1.0</spp:supportedSPPVersions>
     <spp:supportedMOList>urn:wfa:mo:hotspot2dot0-perprovidersubscription:1.0
          urn:oma:mo:oma-dm-devinfo:1.0 urn:oma:mo:oma-dm-devdetail:1.0
          urn:wfa:mo-ext:hotspot2dot0-devdetail-ext:1.0 </spp:supportedMOList>
     <spp:moContainer spp:moURN="urn:oma:mo:oma-dm-devinfo:1.0">
       &lt;MgmtTree&gt;
         &lt;VerDTD&gt;1.2&lt;/VerDTD&gt;
         &lt;Node&gt;&lt;NodeName&gt;DevInfo&lt;/NodeName&gt;
           &lt;RTProperties&gt;&lt;Type&gt;&lt;DDFName&gt;urn:oma:mo:oma-dm-
              devinfo:1.0&lt;/DDFName&gt;&lt;/Type&gt;&lt;/RTProperties&gt;
         &lt;/Node&gt;
         &lt;Node&gt;&lt;NodeName&gt;DevID&lt;/NodeName&gt;&lt;Path&gt;DevInfo&lt;/Path&gt;
             &lt;Value&gt;urn:acme:00-11-22-33-44-55&lt;/Value&gt;&lt;/Node&gt;
         &lt;Node&gt;&lt;NodeName&gt;Man&lt;/NodeName&gt;&lt;Path&gt;DevInfo&lt;/Path&gt;
             &lt;Value&gt;ACME&lt;/Value&gt;&lt;/Node&gt;
         &lt;Node&gt;&lt;NodeName&gt;Mod&lt;/NodeName&gt;&lt;Path&gt;DevInfo&lt;/Path&gt;
             &lt;Value&gt; HS2.0-01&lt;/Value&gt;&lt;/Node&gt;
         &lt;Node&gt;&lt;NodeName&gt;DmV&lt;/NodeName&gt;&lt;Path&gt;DevInfo&lt;/Path&gt;
             &lt;Value&gt;1.2&lt;/Value&gt;&lt;/Node&gt;
         &lt;Node&gt;&lt;NodeName&gt;Lang&lt;/NodeName&gt;&lt;Path&gt;DevInfo&lt;/Path&gt;
             &lt;Value&gt;en-US&lt;/Value&gt;&lt;/Node&gt;
       &lt;/MgmtTree&gt;
     </spp:moContainer>
     <spp:moContainer spp:moURN="urn:oma:mo:oma-dm-devdetail:1.0">
       <![CDATA[ <MgmtTree>
         <VerDTD>1.2</VerDTD>
         <Node><NodeName>DevDetail</NodeName>
           <RTProperties><Type><DDFName>urn:oma:mo:oma-dm-devdetail:1.0</DDFName></Type>
                </RTProperties>
           <Node><NodeName>Ext</NodeName>
             <Node><NodeName>org.wi-fi</NodeName>
               <RTProperties><Type><DDFName>urn:wfa:mo-ext:hotspot2dot0-devdetail-
                  ext:1.0</DDFName></Type></RTProperties>
               <Node><NodeName>Wi-Fi</NodeName>
                 <Node><NodeName>EAPMethodList</NodeName>
                   <Node><NodeName>Method01</NodeName><!-- EAP-TTLS/MS-CHAPv2 -->
                     <Node><NodeName>EAPType</NodeName><Value>21</Value></Node>
                     <Node><NodeName>InnerMethod</NodeName><Value>MS-CHAP-V2</Value></Node></Node>
                   <Node><NodeName>Method02</NodeName><!-- EAP-TLS -->
                     <Node><NodeName>EAPType</NodeName><Value>13</Value></Node></Node>
                   <Node><NodeName>Method03</NodeName><!-- EAP-SIM -->
                     <Node><NodeName>EAPType</NodeName><Value>18</Value></Node></Node>
                   <Node><NodeName>Method04</NodeName><!-- EAP-AKA -->
                     <Node><NodeName>EAPType</NodeName><Value>23</Value></Node></Node>
                   <Node><NodeName>Method05</NodeName><!-- EAP-AKA' -->
                     <Node><NodeName>EAPType</NodeName><Value>50</Value></Node></Node>
                   <Node><NodeName>Method06</NodeName><!-- Supported method (EAP-TTLS/PAP) not
                                                           mandated by Hotspot 2.0 -->
                     <Node><NodeName>EAPType</NodeName><Value>21</Value></Node>
                     <Node><NodeName>InnerMethod</NodeName><Value>PAP</Value></Node></Node>
                   <Node><NodeName>Method07</NodeName><!-- Supported method (PEAP/EAP-GTC) not
                                                           mandated by Hotspot 2.0 -->
                     <Node><NodeName>EAPType</NodeName><Value>25</Value></Node>
                     <Node><NodeName>InnerEAPType</NodeName><Value>6</Value></Node></Node>
                 </Node>
                 <Node><NodeName>ManufacturingCertificate</NodeName><Value>FALSE</Value></Node>
                 <Node><NodeName>Wi-FiMACAddress</NodeName><Value>001d2e112233</Value></Node>
                 <Node><NodeName>ClientTriggerRedirectURI</NodeName>
                   <Value>http://127.0.0.1:12345/index.htm</Value></Node>
                 <Node><NodeName>Ops</NodeName>
                   <Node><NodeName>launchBrowserToURI</NodeName><Value></Value></Node>
                   <Node><NodeName>negotiateClientCertTLS</NodeName><Value></Value></Node>
                   <Node><NodeName>getCertificate</NodeName><Value></Value></Node>
                 </Node>
               </Node><!-- End of Wi-Fi node -->
             </Node><!-- End of org.wi-fi node -->
           </Node><!-- End of Ext node -->
```

Figure 73: Example sppPostDevData SOAP message

```
            <Node><NodeName>URI</NodeName>
              <Node><NodeName>MaxDepth</NodeName><Value> 32 </Value></Node>
              <Node><NodeName>MaxTotLen</NodeName><Value> 2048 </Value></Node>
              <Node><NodeName>MaxSegLen</NodeName><Value> 64 </Value></Node>
            </Node>
            <Node><NodeName>DevType</NodeName><Value> Smartphone </Value></Node>
            <Node><NodeName>OEM</NodeName><Value> ACME </Value></Node>
            <Node><NodeName>FwV</NodeName><Value> 1.2.100.5 </Value></Node>
            <Node><NodeName>SwV</NodeName><Value> 9.11.130 </Value></Node>
            <Node><NodeName>HwV</NodeName><Value> 1.0 </Value></Node>
            <Node><NodeName>LrgObj</NodeName><Value> TRUE </Value></Node>
          </Node>
        </MgmtTree> ]]>
      </spp:moContainer>
    </spp:sppPostDevData>
  </s12:Body>
</s12:Envelope>
```

Figure 74: Figure 73 continued

Notes:

1. In Figure 73 the CDATA directive indicates to an XML parser that the encapsulated data is not to be parsed. Implementations may choose to escape all XML metacharacters (e.g., use "<" for "<") instead of using the CDATA directive.
2. In Figure 73 the DevDetail MO is encoded in a hierarchical style, using <Node> elements which are child elements of other <Node elements>. The DevInfo MO is encoded in an alternative style, in which all the <Node> elements are at the same level and the <Path> element provides the hierarchy. Both encoding methods are specified by [38].
3. The requestReason provided by the mobile device does not restrict server actions. For example, the Exec command could be used to launch the web browser during policy update.

Table 13: sppPostDevData Elements and Attributes Descriptions

Element Name / Attribute Name	Definition
sppPostDevData	SOAP Method to initiate an SPP message exchange with a Subscription server.
sppVersion	The version of the SPP protocol used by the mobile device in the sppPostDevData message. This shall be version 1.0 for the initial sppPostDevData message to a subscription server unless the mobile device has out-of-band information that a higher version number is supported by the subscription server. The subscription server selects the SPP version for subsequent messages in its initial sppPostDevDataResponse message. Once the server has chosen the SPP version to use, all subsequent SPP exchanges shall continue using that SPP version until the message sequence is completed.

Element Name / Attribute Name	Definition
requestReason	The reason the mobile device is sending the sppPostDevData message. Defined values of requestReason are: • Subscription registration • Subscription provisioning • Subscription remediation • User input completed • No acceptable client certificate • Certificate enrollment completed • Certificate enrollment failed • Subscription metadata update • Policy update • MO upload • Retrieve next command • Unspecified Notes: • The “Unspecified” value of requestReason is included for future proofing.
sessionID	A 128-bit random number generated by the SPP server used to identify messages belonging to the same SPP client to SPP server message exchange. The sessionID value is copied from the sppPostDevDataResponse SOAP method into this attribute. Note: this facilitates the use of multiple TLS tunnels and/or TCP connections between a given mobile device (SPP client) and a subscription server (SPP server).
supportedSPPVersions	List of SPP version numbers supported by the mobile device. This list shall contain the value of sppVersion included in the attribute of the sppPostDevData element. This list shall include version 1.0. Note that there may be more than one supported version in this list.
supportedMOList	List of URNs corresponding to OMA DM MOs defined for use with SPP which are supported by the mobile device. This is intended to be used, for example, if additional MOs are defined in the future or if there is a revision to the PerProviderSubscription MO. OMA DM standard MOs required by SPP are included in this list. All supported versions of each MO are listed individually. Support for the following MOs is required by SPP version 1.0: • urn:wfa:mo:hotspot2dot0-perprovidersubscription:1.0 • urn:wfa:mo-ext:hotspot2dot0-devdetail-ext:1.0 • urn:oma:mo:oma-dm-devinfo:1.0 • urn:oma:mo:oma-dm-devdetail:1.0 If more than one version of an MO is supported by a mobile device, the subscription server determines which version to use.
moContainer	This element is used as a container object for an OMA DM managed object.
moURN	The URN of the managed object in the moContainer element.
any	This element is used so that future versions of SPP can add new elements while retaining backward compatibility to version 1.0.

A.3.2 The sppPostDevDataResponse SOAP method

Figure 75 through Figure 81 show the components of the sppPostDevDataResponse SOAP method, the sppPostDevDataResponse SOAP exec methods and related XML methods. Figure 82 and Figure 83 provide an example of sppPostDevDataResponse messages. Table 14 contains definitions of the sppPostDevDataResponse message elements and attributes.

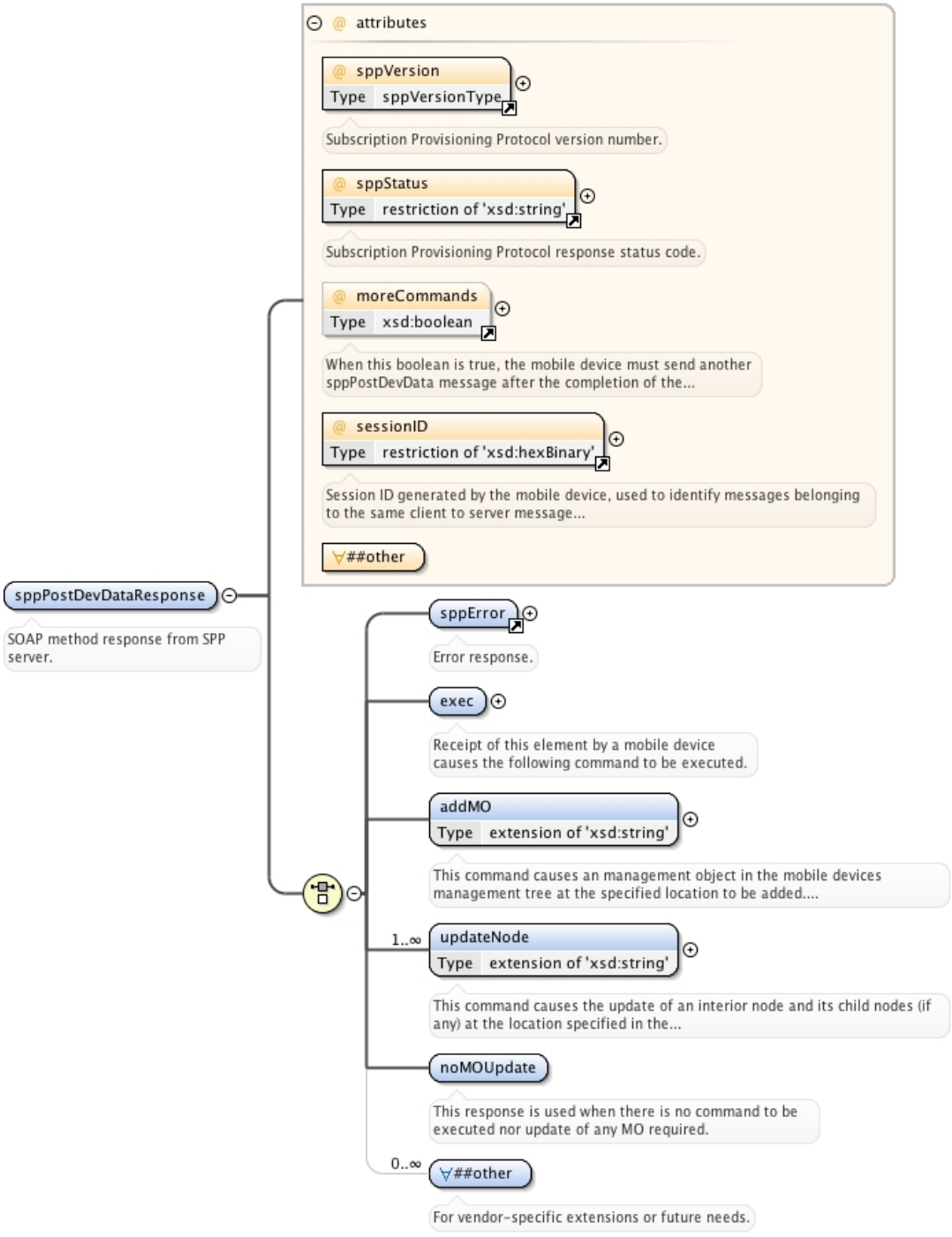


Figure 75: Graphical diagram of the sppPostDevDataResponse SOAP method

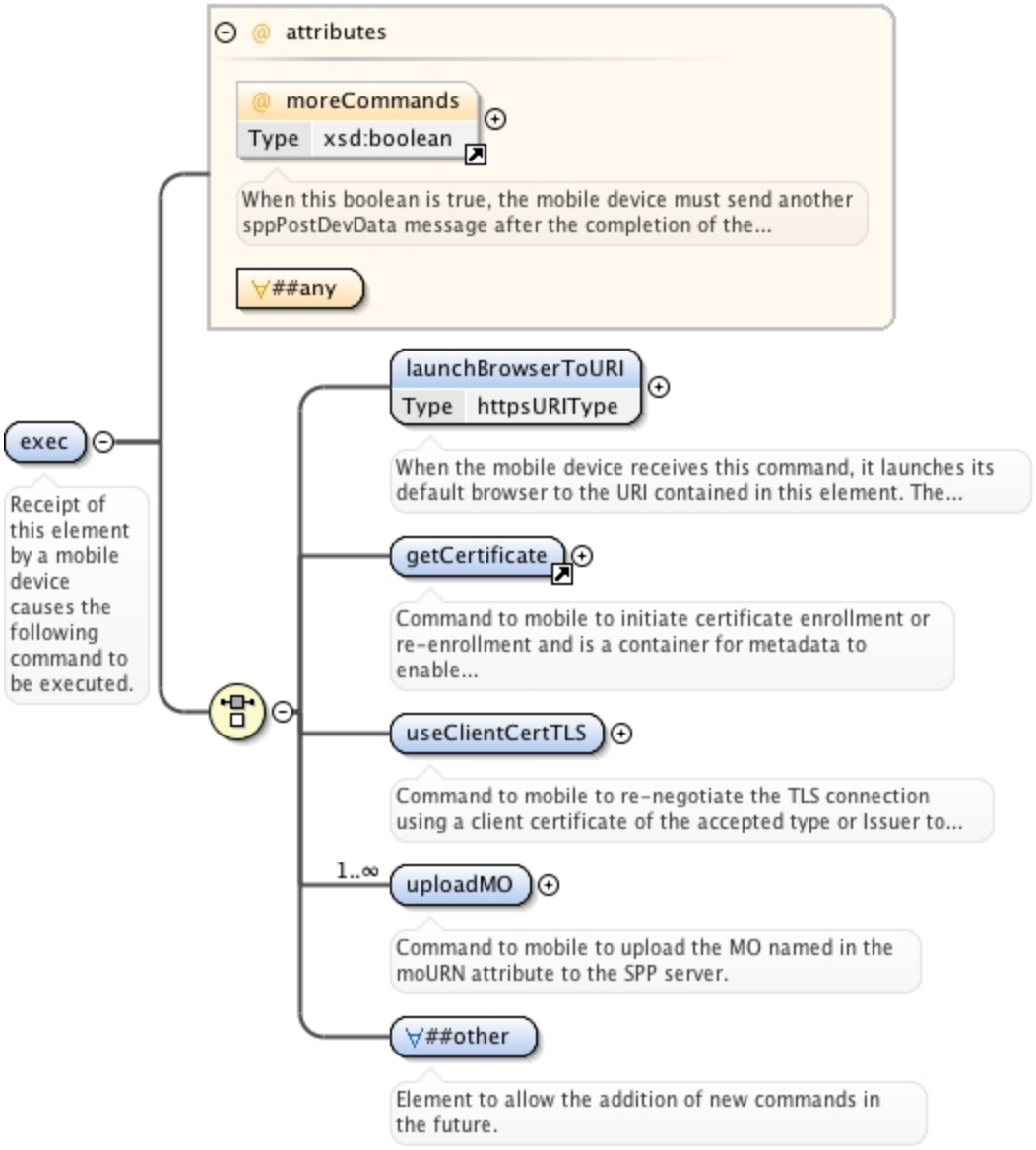


Figure 76: Graphical diagram of the sppPostDevDataResponse SOAP exec methods

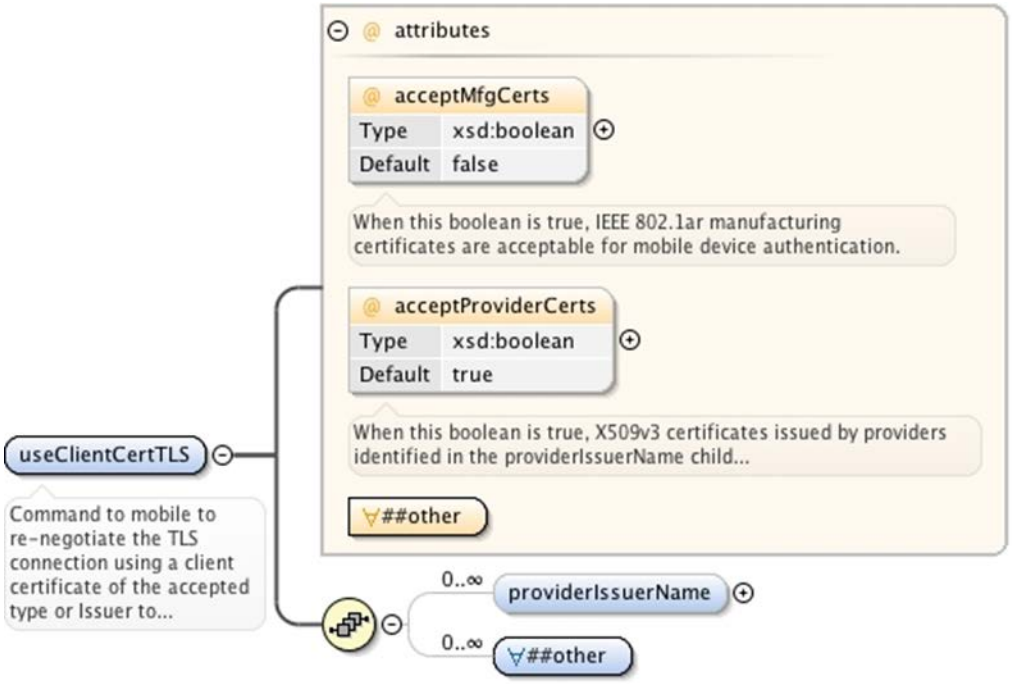


Figure 77: Graphical diagram of the useClientCertTLS XML element

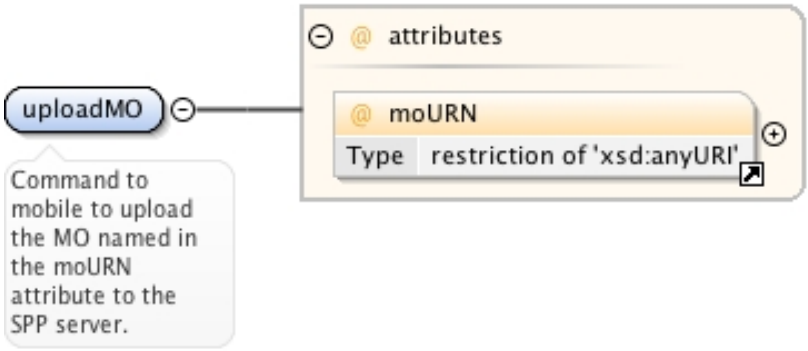


Figure 78: Graphical diagram of the uploadMO XML element

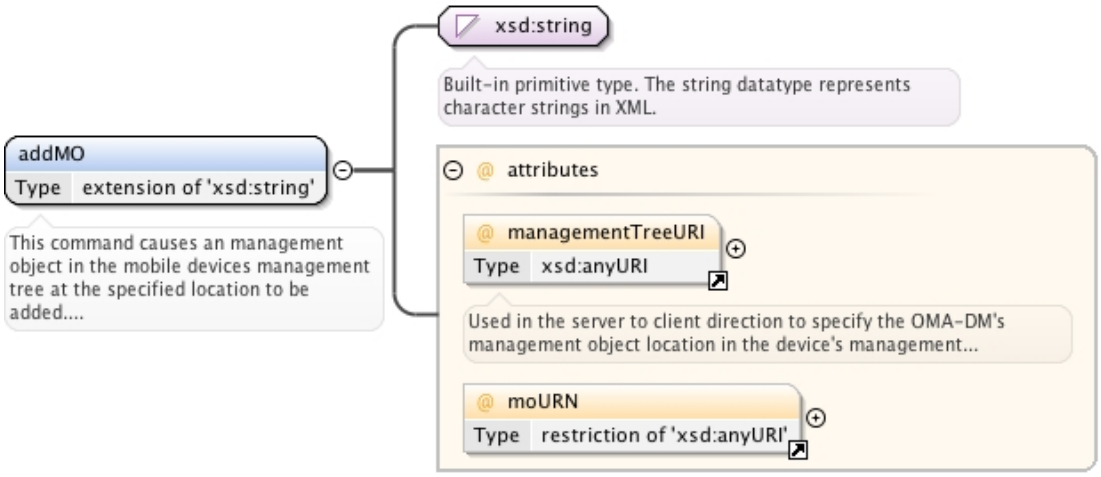


Figure 79: Graphical diagram of the addMO XML element

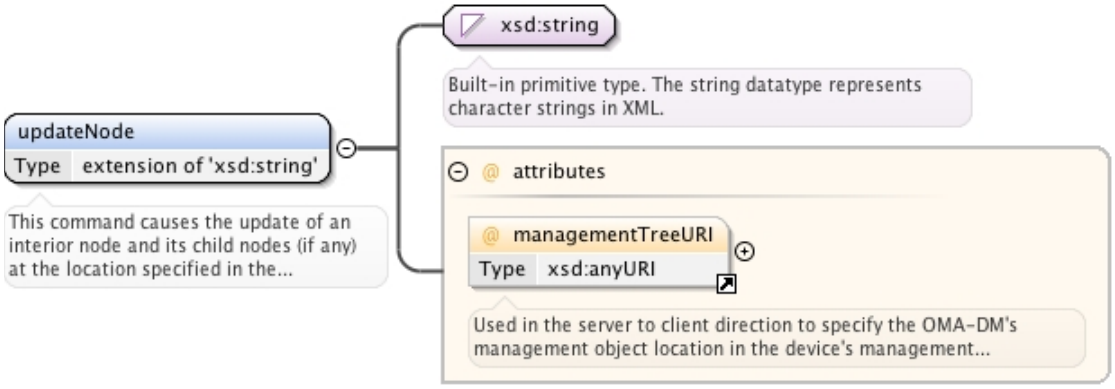


Figure 80: Graphical diagram of the updateNode XML element

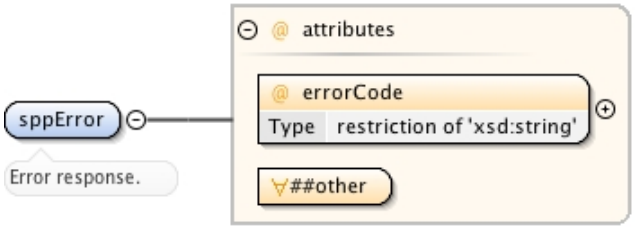


Figure 81: Graphical diagram of the sppError XML element

```
<?xml version="1.0" encoding="UTF-8"?>
<s12:Envelope    xmlns:s12="http://www.w3.org/2003/05/soap-envelope"
                 xmlns:spp="http://www.wi-fi.org/specifications/hotspot2dot0/v1.0/spp">
   <s12:Body>
      <spp:sppPostDevDataResponse spp:sppVersion=”1.0” spp:sppStatus=”OK”
          spp:sessionID="abcdef0123456789876543210fedcba">
          <spp:exec>
             <spp:launchBrowserToURI> https://sign-
up.example.com/hotspot2.0/signup.html?sessionID=abcdef0123456789876543210fedcba
             </spp:launchBrowserToURI>
          </spp:exec>
      </spp:sppPostDevDataResponse >
   </s12:Body>
</s12:Envelope>
```

Figure 82: Example sppPostDevDataResponse SOAP message #1

```
<?xml version="1.0" encoding="UTF-8"?>
<s12:Envelope    xmlns:s12="http://www.w3.org/2003/05/soap-envelope"
   xmlns:spp="http://www.wi-fi.org/specifications/hotspot2dot0/v1.0/spp">
   <s12:Body>
      <spp:sppPostDevDataResponse spp:sppVersion="1.0" spp:sppStatus="OK"
            spp:sessionID="abcdef0123456789987654321 0fedcba">
         spp:addMO spp:managementTreeURI="./Wi-Fi/wi-fi.org/PerProviderSubscription
               spp:moURN="urn:wfa:mo:hotspot2dot0-perprovidersubscription:1.0">
            <![CDATA[ <MgmtTree xmlns="syncml:dmddf1.2">
               <VerDTD>1.2</VerDTD>
               <Node>
                  <NodeName>PerProviderSubscription</NodeName>
                  <RTProperties>
                     <Type>
                        <DDFName>urn:wfa:mo:hotspot2dot0-perprovidersubscription:1.0</DDFName>
                     </Type>
                  </RTProperties>
                  <Node><NodeName>x1</NodeName>
                     <Node><NodeName>CredentialPriority</NodeName><Value> 1 </Value></Node>
                     <Node><NodeName>SubscriptionUpdate</NodeName>
   <Node><NodeName>UpdateInterval</NodeName><Value>4294967295</Value></Node>

   <Node><NodeName>UpdateMethod</NodeName><Value>ClientInitiated</Value></Node>
                           <Node><NodeName>Restriction</NodeName><Value>HomeSP</Value></Node>
                           <Node><NodeName>URI</NodeName>
                                 <Value>subscription-server.R2-testbed.wi-fi.org</Value></Node>
                        </Node>
                        <Node><NodeName>HomeSP</NodeName>
                           <Node><NodeName>FriendlyName</NodeName><Value>Wi-Fi Alliance</Value></Node>
                           <Node><NodeName>FQDN</NodeName><Value>wi-fi.org</Value></Node>
                           <Node><NodeName>HomeOIList</NodeName>
                              <Node><NodeName>x1</NodeName>
                                 <Node><NodeName>HomeOI</NodeName><Value>506f9a</Value></Node>
                                 <Node><NodeName>HomeOIRequired</NodeName><Value>FALSE</Value></Node>
                              </Node>
                              <Node><NodeName>x2</NodeName>
                                 <Node><NodeName>HomeOI</NodeName><Value>004096</Value></Node>
                                 <Node><NodeName>HomeOIRequired</NodeName><Value>FALSE</Value></Node>
                              </Node>
                           </Node>
                        </Node>
                        <Node><NodeName>SubscriptionParameters</NodeName></Node>
                        <Node><NodeName>Credential</NodeName>
                           <Node><NodeName>CreationDate</NodeName><Value>2012-12-
01T12:00:00Z</Value></Node>
                           <Node><NodeName>UsernamePassword</NodeName>
                              <Node><NodeName>Username</NodeName><Value>test01</Value></Node>
   <Node><NodeName>Password</NodeName><Value>Q2hhbmdlTWU=</Value></Node>
                              <Node><NodeName><MachineManaged</NodeName><Value>TRUE</Value></Node>
                              <Node><NodeName>EAPMethod</NodeName>
                                 <Node><NodeName>EAPType</NodeName><Value>21</Value></Node>
                                 <Node><NodeName>InnerMethod</NodeName><Value>MS-CHAP-
V2</Value></Node>
                              </Node>
                           </Node>
                           <Node><NodeName>Realm</NodeName><Value>wi-fi.org</Value></Node>
                        </Node>
                     </Node>
                  </Node>
               </Node> ]]>
            </spp:addMO>
      </spp:sppPostDevDataResponse >
   </s12:Body>
</s12:Envelope>
```

Figure 83: Example sppPostDevDataResponse SOAP Message #2

In the example shown in Figure 83, the XML element, <x1>, is an example of a valid dynamic node name (per the PerProviderSubscription MO DDF [40]. Other examples are possible.

In Figure 83 the CDATA directive indicates to an XML parser that the encapsulated data is not to be parsed. Implementations may choose to escape all of the XML metacharacters (e.g., use "<" for "<") instead of using the CDATA directive.

Table 14: sppPostDevDataResponse Elements and Attributes Descriptions

Element Name / Attribute Name	Definition
sppPostDevDataResponse	SOAP method response from the subscription server.
sppVersion	The SPP version number chosen by the subscription server from the supportedSPPVersions element of the sppPostDevData message. Once the subscription server has chosen a version in the initial message exchange with a particular mobile device, both the server and the mobile device shall continue to use that version until the message exchange is completed (see sppStatus below).
sppStatus	Status code of the sppPostDevDataResponse SOAP response. The permitted values are: • OK • Provisioning complete, request sppUpdateResponse • Remediation complete, request sppUpdateResponse • Update complete, request sppUpdateResponse • No update available at this time • Exchange complete, release TLS connection • Error occurred Note that if sppStatus is set to “Error occurred” additional information is provided in the errorCode attribute described below.
moreCommands	The value of this optional attribute is set to true when a subscription server is requesting the mobile device to send a follow up sppPostDevData message after the completion of the current command to retrieve an additional command. The default value of this attribute is false.
sessionID	See Table 13. A 128-bit random number generated by the SPP server which is used to identify a message exchange sequence.
sppError	This empty element contains the errorCode from a subscription server.
errorCode	Subscription server error code. The permitted values are: • SPP version not supported • One or more mandatory MOs not supported • Credentials cannot be provisioned at this time • Remediation cannot be completed at this time • Provisioning cannot be completed at this time • Continue to use existing certificate • Other Note: The first three codes are critical errors
any	This attribute is used so that future versions of SPP can add new attributes at this location while retaining backward compatibility to version 1.0.

Element Name / Attribute Name	Definition
exec	Container element for a command that the mobile device is being requested to execute. There is exactly one command in the container. This element provides a function similar to the OMA DM exec command.
launchBrowserToURI	When a mobile device receives this command, it shall launch its default browser to the URI contained in this element. The URI shall use HTTPS as the transfer protocol.
getCertificate	See Table 17.
any	This element is used so that future versions of SPP can add new elements while retaining backward compatibility to version 1.0. Note that the “any” element is used in several places in the figure.
useClientCertTLS	When a mobile device receives this command, it shall re-negotiate the TLS connection with the subscription server using a mobile device-provided certificate of the accepted type or Issuer.
acceptMfgCerts	This boolean indicates whether a subscription server will accept a mobile device provided manufacturing certificate as an acceptable certificate type for network access authentication.
acceptProviderCerts	This boolean indicates whether a subscription server will accept for network access authentication certificates pre-provisioned by a certificate provider. If so, one or more providerIssuerName elements are present.
providerIssuerName	This empty element gives the Issuer Name for an acceptable provider certificate. The name itself is provided in this element’s attribute.
name	The Issuer Name for an acceptable provider certificate.
uploadMO	This element is a command that causes the mobile device to upload the MO specified in the moURN attribute to the SPP server.
moURN	See Table 13.
addMO	This element is a command that causes an MO to be added in the mobile device's management tree at the location specified in the managementTreeURI attribute. If there is already an MO at that location, the object is replaced. The MO added is identified by the moURN attribute in the element. It shall be chosen from the supportedMOList provided by the mobile device in the sppPostDevData message.
managementTreeURI	This attribute provides the absolute URI in a mobile device’s MO tree at which to perform a specified operation. This object is equivalent in function to the LocURI OMA DM object and follows its syntax. Note: since the PerProviderSubscription MO uses dynamic nodes, the subscription servers provisioning this MO shall keep track of the [dynamic] node names; otherwise the server(s) may not be able to construct a valid managementTreeURI.
updateNode	This element is a command that causes an interior node and all its child nodes in an existing MO at the location specified in the managementTreeURI attribute to be updated (i.e., replaced). This element shall not be used to replace an entire MO; it shall only be used to add, update, replace or delete nodes in an MO which is already present in the mobile device.

Element Name / Attribute Name	Definition
noMOUpdate	This element is included in the response method when there is no command to be executed nor update of any MO required.

A.3.3 The sppUpdateResponse SOAP Method

Figure 84 shows the components of the sppUpdateResponse SOAP method. Figure 85provides an example sppUpdateResponse message. Table 15 contains definitions of the sppUpdateResponse message elements and attributes.

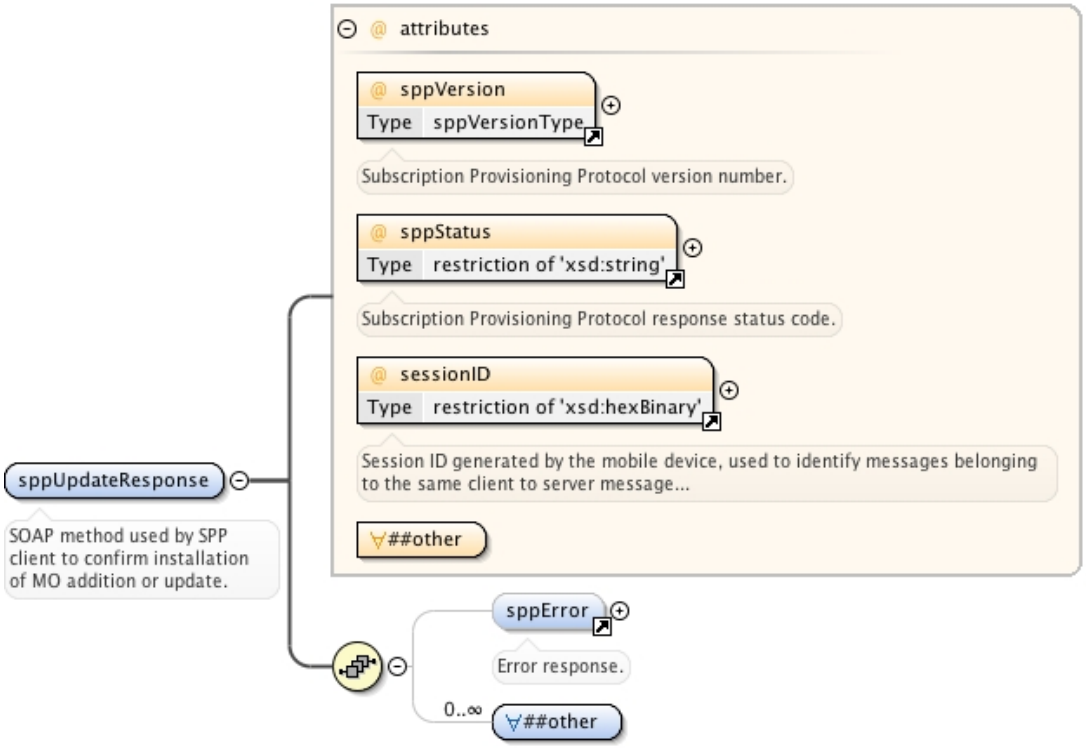


Figure 84: Graphical diagram of the sppUpdateResponse SOAP method

```
<?xml version="1.0" encoding="UTF-8"?>
<s12:Envelope    xmlns:s12="http://www.w3.org/2003/05/soap-envelope"
xmlns:spp="http://www.wi-fi.org/specifications/hotspot2dot0/v1.0/spp">
    <s12:Body>
        <spp:sppUpdateResponse spp:sppVersion="1.0" spp:sppStatus="OK"
            spp:sessionID="abcdef0123456789987654321 0fedcba">
        </spp:sppUpdateResponse>
    </s12:Body>
</s12:Envelope>
```

Figure 85: Example sppUpdateResponse SOAP message

Table 15: sppUpdateResponse Elements and Attributes Descriptions

Element Name / Attribute Name	Definition
sppUpdateResponse	SOAP Method to inform a subscription server that an MO was successfully added to the mobile device's management tree, that a node (and its child nodes) was successfully updated, or that an error occurred. In the case an error occurred, the mobile device is indicating the requested MO could not be added or the node update (including child nodes) could not be performed.
sppVersion, sppStatus, sessionID, any	See Table 13.
sppError	See Table 13.
errorCode	See Table 14 In addition, the following error codes are defined: • Permission denied • Command failed • MO addition or update failed (mobile status only) • Device full • Bad management tree URI • Requested entity too large • Command not allowed • Command not executed due to user • Not found
any	See Table 13.

A.3.4 The sppExchangeComplete SOAP Method

Figure 86 shows the components of the sppExchangeComplete SOAP method. Figure 87 provides an example sppExchangeComplete message. Table 16 contains definitions of the sppExchangeComplete message elements and attributes.

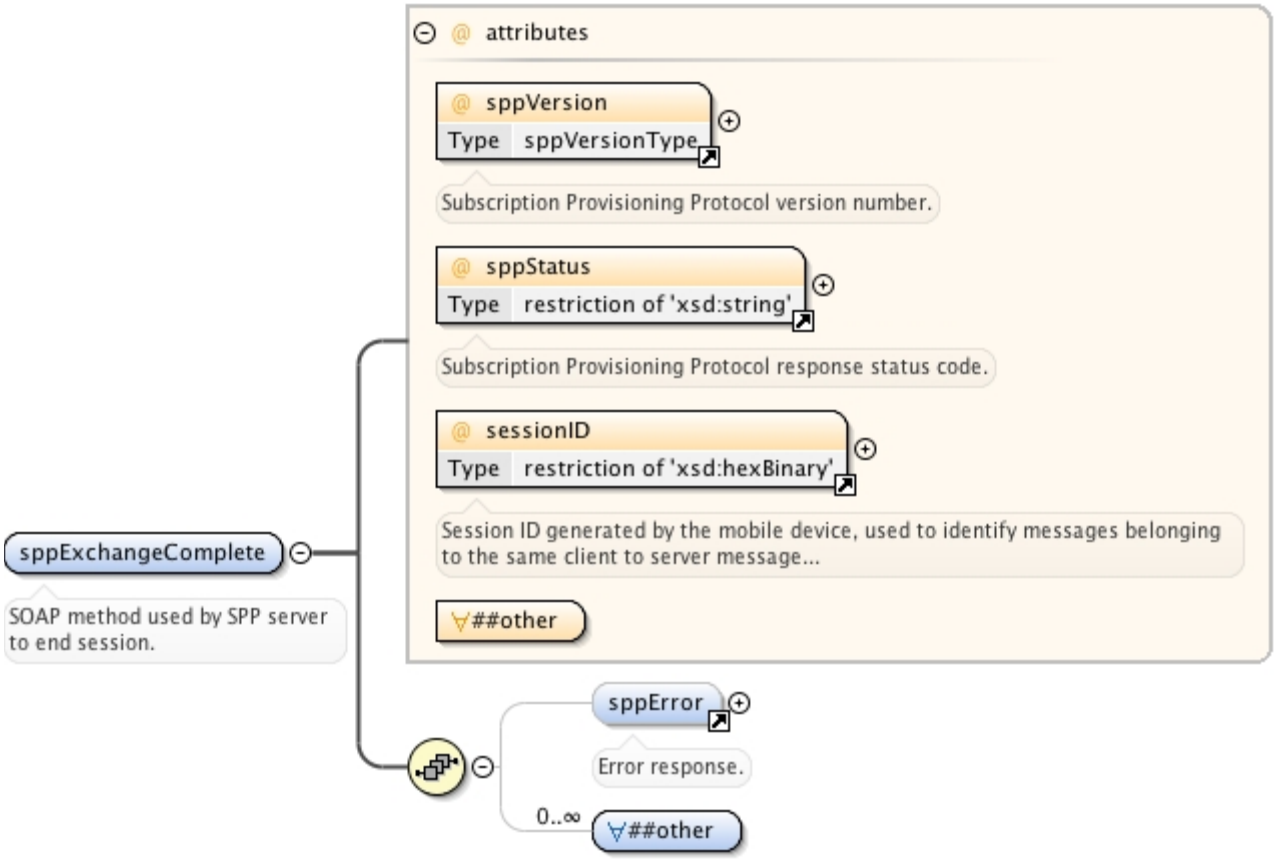


Figure 86: Graphical diagram of the sppExchangeComplete SOAP method

```
<?xml version="1.0" encoding="UTF-8"?>
<s12:Envelope    xmlns:s12="http://www.w3.org/2003/05/soap-envelope"
xmlns:spp="http://www.wi-fi.org/specifications/hotspot2dot0/v1.0/spp">
   <s12:Body>
      <spp:sppExchangeComplete spp:sppVersion="1.0"
                spp:sppStatus="Exchange complete, release TLS connection"
                spp:sessionID="abcdef0123456789zz9876543210fedcba">
      </spp:sppExchangeComplete>
   </s12:Body>
</s12:Envelope>
```

Figure 87: Example sppExchangeComplete SOAP message

Table 16: sppExchangeComplete Elements and Attributes Descriptions

Element Name/ Attribute Name	Definition
sppExchangeComplete	SOAP Method to inform the mobile device that the message exchange sequence has been completed and that the TLS connection should be released.
sppVersion, sppStatus, sessionID, any	See Table 14.
sppError	See Table 14.
errorCode	See Table 14.
any	See Table 13.

A.3.5 The getCertificate XML Instance Document

Figure 88 shows the getCertificate message. Figure 89 provides an example getCertificate message. Note: this XML instance document is used in the OMA DM message exchange to instruct the mobile device to initiate certificate enrollment. Table 17 contains definitions of the getCertificate elements and attributes.

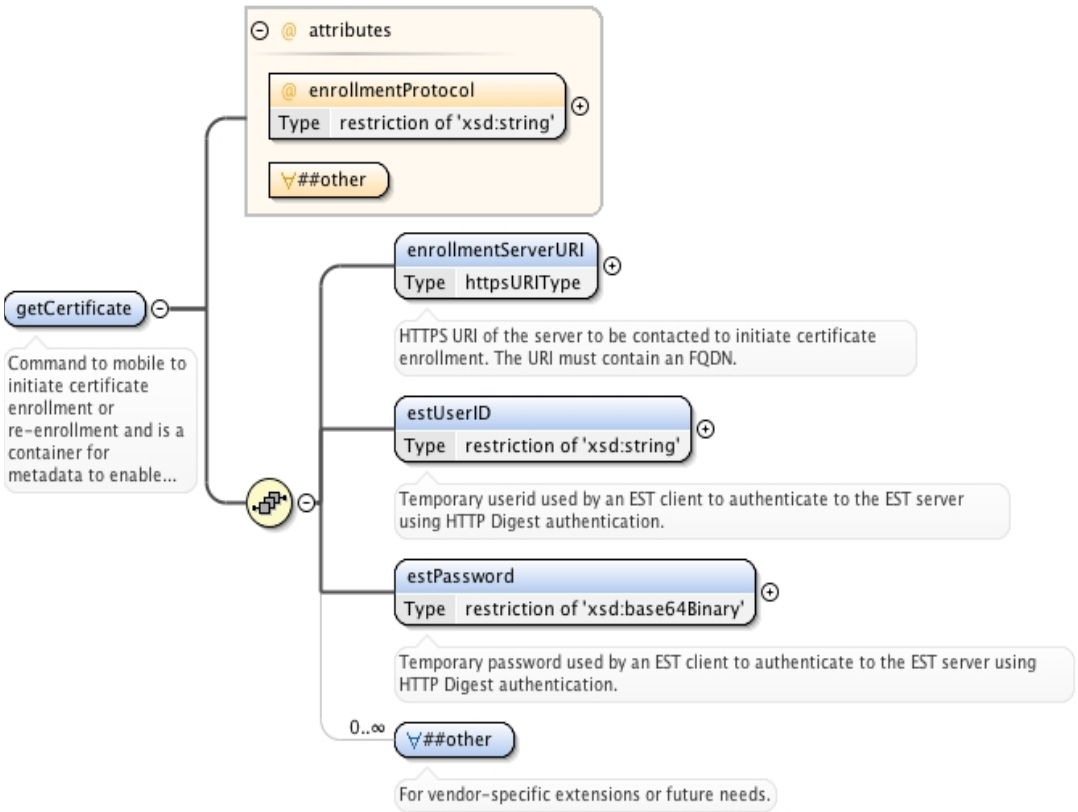


Figure 88: Graphical diagram of the getCertificate XML instance document

```
<?xml version="1.0" encoding="UTF-8"?>
<spp:getCertificate xmlns:spp="http://www.wi-fi.org/specifications/hotspot2dot0/v1.0/spp"
enrollmentProtocol="EST">
    <spp:enrollmentServerURI> https://www.example.com/.well-known/est </spp:enrollmentServerURI>
    <spp:estUserID>tempest-user</spp:estUserID>
    <spp:estPassword>ZVN0d2l0aGhTMi4w</spp:estPassword> <!-- base64 encoded for "eStwithhS2.0" -->
</spp:getCertificate>
```

Figure 89: Example getCertificate XML instance document

Table 17: getCertificate elements and attributes

Element Name / Attribute Name	Definition
getCertificate	XML response from the subscription server. This command instructs the device to initiate certificate enrollment or re-enrollment.
enrollmentProtocol	Name of the certificate enrollment protocol that the mobile device shall use to obtain a certificate. The only permitted value is “EST”.
enrollmentServerURI	The HTTPS URI of the certificate enrollment server that the mobile device contacts to obtain a certificate.
estUserID	A temporary userid used by an EST client to authenticate to the EST server when HTTP-based Digest authentication is required. This element shall be used for initial certificate enrollment; its use is optional for certificate re-enrollment
estPassword	A temporary password used by an EST client to authenticate to the EST server when HTTP-based Digest authentication is required. This element shall be used for initial certificate enrollment; its use is optional for certificate re-enrollment.
any	See Table 13.

A.3.6 Web Services Description Language (WSDL)

A WSDL file to accompany the SPP schema is provided by [50].

Annex B : Example GAS Query using ANQP Query List and HS Query List (informative)

This annex provides example GAS queries describing how the Advertisement Protocol ANQP is transported in GAS frames

B.1 Example 1: 3GPP Cellular Network and the Operator Friendly Name

Example 1 in Figure 90 illustrates the possibility of including an ANQP Query List and a HS Query List in a single GAS Initial Request frame including a request for the 3GPP Cellular Network element and the Operator Friendly Name element.

	Category	Public Action	Dialog Token	Advertisement Protocol element	Query Request length	Query Request
Octets:	1	1	1	variable	2	variable

Figure 90: GAS Initial Request frame (Action frame)

The value of the Category field is set to 4 (indicating the Public category), per Table 8-38 in [2].

The value of the Public Action field is set to 10 (for GAS Initial Request), per Table 8-210 in [2].

The Dialog Token field is used for matching action responses with action requests when there are multiple concurrent action requests, per Figure 8-439 in [2].

The value of the Advertisement Protocol element uses the Advertisement Protocol ID of 0 (for ANQP), as defined in subclause 8.4.2.95 of [2].

The value of the Query Request length field is set to the total number of octets in the Query Request field.

The Query Request field comprises both the ANQP Query list and the HS Query List, as shown in Figure 91 and Figure 92, in which the details for each ANQP-element are provided.

	Query List ANQP-element	HS Query List
Octets:	6	11

Figure 91: Example Query Request field

	ANQP Query List Info ID	Length	ANQP Query List Information	HS2.0 ANQP-element Header	HS2.0 ANQP-element Payload
Octets:	2	2	2	10	1

Figure 92: Example Query Request field details

The value of the ANQP Query List Info ID subfield is set to 256 as per Table 8-184 in [2].

The value of the Length sub-field is set to 2.

The value of the ANQP Query List Information subfield is set to 264 (for 3GPP Cellular Network), per Table 8-184 in [2].

The HS2.0 ANQP-element header subfield is defined in section 4, with the Subtype field value set to 1 (for HS Query list).

The value of the HS2.0 Payload field is set to 3 (indicating Operator Friendly Name), as defined in section 4.

B.2 Example 2: Icon Request

Example 2 in Figure 93 illustrates an ANQP Query payload carried in a GAS Initial Request frame.

In this example the Query List ANQP-element is requesting a single ANQP-element (e.g., Venue Name) and the Icon Request is for a filename that has 10 characters (encoded in UTF-8, taking 10 octets).

	Query List ANQP-element	Icon Request
Octets:	6	20

Figure 93: Example #2 Query Request field

Annex C : SP policy network connection (informative)

C.1 Example Network Selection Flowchart

This annex contains an example flowchart of possible steps for a mobile device's connection manager, based on the assumptions and procedures described in this specification.

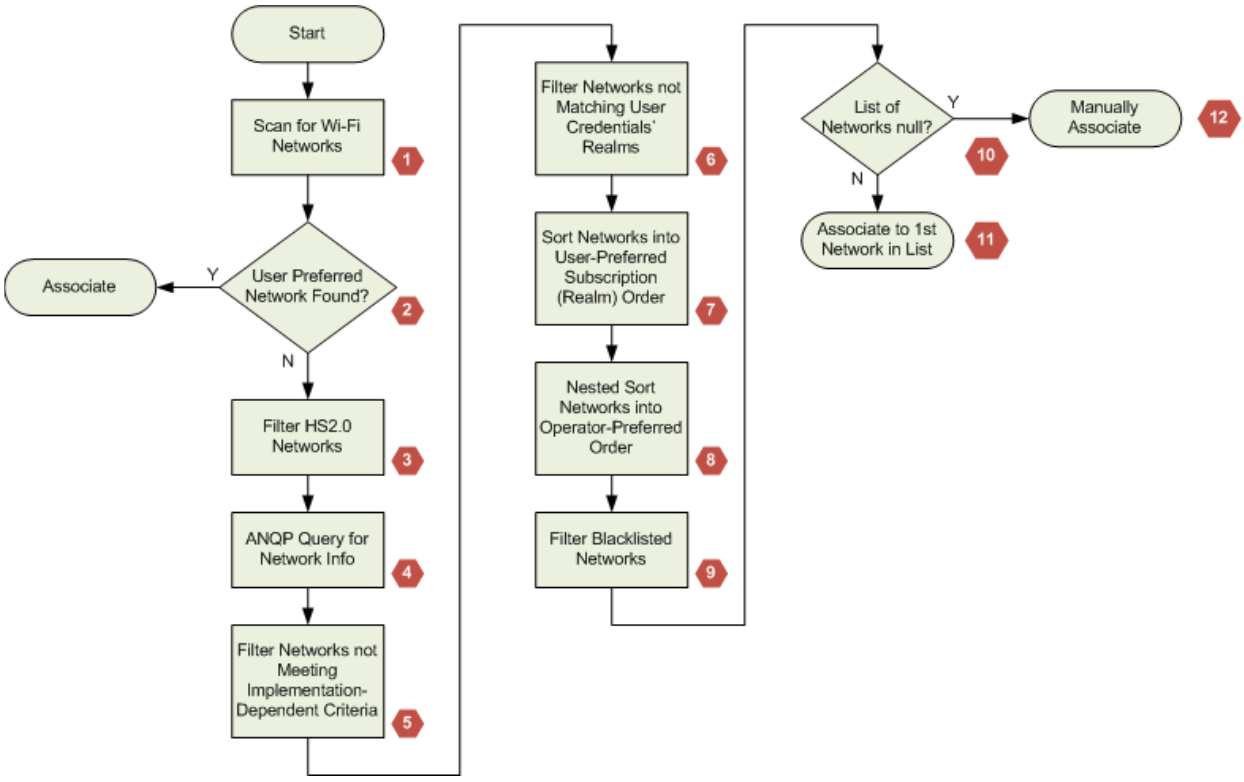


Figure 94: Example mobile device network selection flowchart

Step 1: The CM (connection manager on the mobile device) kicks off the network selection process with an active or passive scan for Wi-Fi ANs.

Step 2: The CM compares discovered networks with the list of preferred networks that the user has previously configured on the mobile device and, if a match is found, causes the mobile device to associate with the Wi-Fi AN that has the highest user preference.

Step 3: If no user-preferred network is found, the list of discovered networks from step 1 is filtered for HS2.0 networks (i.e., non-HS2.0 networks are deleted from the list). Note: network selection for legacy networks is outside the scope of this specification.

Step 4: When an HS2.0 network does not have a cached profile, ANQP queries (such as a query for the NAI Realm and/or 3GPP Cellular Network information) are performed as needed.

Step 5: The discovered network list is filtered based on implementation dependent criteria. Examples include minimum RSSI level, protocols/ports blocked by hotspot firewall, PLMNs, etc.

Step 6: The discovered network list is filtered for realms matching the user's credentials. Wi-Fi ANs whose NAI Realm, 3GPP Cellular Network and OIs do not match user credentials (optionally including credential types) are filtered.

Step 7: When the user has more than one Wi-Fi subscription, the remaining Wi-Fi ANs in the list are sorted by subscription preference which the user has previously configured

Step 8: Then the Wi-Fi ANs in the list go through a 2nd-level sort (i.e., nested sort) in which the networks are order by subscription preference (step 7) and by operator preference

Step 9: The sorted list is then filtered for blacklisted networks

Step 10: If the filtered and sorted list of networks has one or more remaining networks …

Step 11: The CM causes the mobile device to associate to the first entry (highest sorted network) in the list

Step 12: Otherwise, the CM may optionally request the user to decide whether to associate with a Wi-Fi AN.

C.2 Example Network Selection Scenarios

In order to illustrate the use of SP-policy driven network selection, a few sample scenarios are described below. The purpose of these scenarios is to aid developers with their implementations. Only PPS MO data which is necessary to explain each of the following network selection scenarios is provided in order to simplify the description and maintain focus on the relevant information. In all the scenarios below, the assumption is made that each hotspot has sufficiently high RSSI and sufficiently low load (i.e., as reported in the BSS Load element and WAN Metrics ANQP element) so that these parameters do not affect the network selection decision; in other words, the network selection decision is based on policy as described in section 6.1.

C.2.1 Network Selection Scenarios Connecting to a Home Network

In the following scenarios, the mobile device has been provisioned with a PPS MO having data in its HomeSP container node (see section 9.1) as shown in Table 18; furthermore, the PPS MO has its Credential/Realm node set to a value of "sp-blue.com".

Table 18: Example HomeSP Provisioned Data Set #1

Home SP Child Node	Data
NetworkID/<x+>/SSID	Hotspot 2.0 Wi-Fi
NetworkID/<x+>/HESSID	001d2e0011a0
FriendlyName	Blue
IconURL	http://www.sp-blue.com/icons/blue_icon.png
FQDN	sp-blue.com
HomeOIList/x1/HomeOI	001d2e
HomeOIList/x1/HomeOIRequired	FALSE
OtherHomePartners/f1/FQDN	example.com
RoamingConsortiumOI	001bc50050, 001bc500b5

C.2.1.1 Scenario #1

A mobile device having the PPS MO data described in section C.2.1 detects several hotspots advertising data as described in Table 19. In this scenario, the mobile device selects, associates and authenticates to Hotspot #1. The following rationale is relevant to this selection:

- The mobile device recognized the SSID for Hotspot #1 as matching the HomeSP/<x+>/SSID leaf node and therefore determined Hotspot #1 was a home

network. According to section 6.1.1, the mobile device prefers/joins its home network over a visited network.

- Since the mobile device recognized the SSID, it determines Hotspot #1 was a home network without the need to perform any ANQP queries. Nevertheless, if the mobile performed an ANQP query for Domain Name List, it would have discovered the returned Domain Name matched the FQDN in HomeSP/FQDN and concluded the hotspot was a home network.#
- Note: because Hotspot #2 and Hotspot #3 advertised the NAI Realm "sp-blue-com" and the OI "001bc50050", the mobile device could successfully authenticate at any of the three hotspots.

Table 19: Hotspot Environment #1

Parameter	Hotspot #1	Hotspot #2	Hotspot #3
Hotspot 2.0 element	present	present	present
SSID	Hotspot 2.0 Wi-Fi	Fast Wi-Fi	Downtown Wi-Fi
Domain Name List	sp-blue.com	sp-green.com	sp-red.com
OIs	001d2e	001bc50050	001bc50050
NAI Realms	sp-blue.com,	sp-green.com, sp-blue.com	sp-red.com, sp-blue.com

C.2.1.2 Scenario #2

A mobile device having the PPS MO data described in section C.2.1 detects several hotspots advertising data as described in Table 20. In this scenario, the mobile device selects, associates and authenticates to Hotspot #1. The following rationale is relevant to this selection:

- The mobile device determined none of the SSIDs matches the HomeSP/<x+>/SSID leaf node and therefore determined that ANQP queries would be useful. Therefore, the mobile device queries an AP in each Hotspot for the Domain Name List. In the ANQP query response from Hotspot #1 AP, the mobile device discovers the returned Domain Name matches the FQDN in HomeSP/FQDN and concludes that hotspot is a home network. According to section 6.1.1, the mobile device prefers/joins its home network over a visited network.
- Note: if Hotspot #1 advertised the OI "001d2e" in the Roaming Consortium element (which is included in Beacon and Probe Response frames), the mobile device could have concluded successful authentication is possible without the need to perform any ANQP queries.
- Note that because Hotspot #2 and Hotspot #3 advertise the NAI Realm "sp-blue.com" and the OI "001bc50050", the mobile device could successfully authenticate at any of the three hotspots.

Table 20: Hotspot Environment #2

Parameter	Hotspot #1	Hotspot #2	Hotspot #3
Hotspot 2.0 element	present	present	present

Parameter	Hotspot #1	Hotspot #2	Hotspot #3
SSID	Blue Wi-Fi	Fast Wi-Fi	Downtown Wi-Fi
Domain Name List	sp-blue.com	sp-green.com	sp-red.com
OIs	001d2e	001bc50050	001bc50050
NAI Realms	sp-blue.com,	sp-green.com, sp-blue.com	sp-red.com, sp-blue.com

C.2.1.3 Scenario #3

- A mobile device having the PPS MO data described in section C.2.1 detects several hotspots advertising data as described in Table 21. In this scenario, the mobile device selects, associates and authenticates to either Hotspot #1 or Hotspot #3. The following rationale is relevant to this selection:
- The mobile device determined none of the SSIDs matches the HomeSP/<x+>/SSID leaf node and therefore determined that ANQP queries would be useful. Therefore, the mobile device queries an AP in each Hotspot for the Domain Name List. In the ANQP query response from Hotspot #1 AP, the mobile device discovers the returned Domain Name matches the FQDN in HomeSP/FQDN and concludes that hotspot is a home network; also, the mobile device discovers the returned Domain Name from Hotspot #3 matches the FQDN in OtherHomePartners/<x+>/FQDN and concludes that hotspot is a home network too. According to section 6.1.1, the mobile device prefers/joins its home network over a visited network. Since there are two choices, of home network, it is up to the mobile device to decide which one to join.

Table 21: Hotspot Environment #3

Parameter	Hotspot #1	Hotspot #2	Hotspot #3
Hotspot 2.0 element	present	present	present
SSID	Blue Wi-Fi	Fast Wi-Fi	Downtown Wi-Fi
Domain Name List	sp-blue.com	sp-green.com	example.com
OIs	001d2e	001bc50050	001bc50050
NAI Realms	sp-blue.com,	sp-green.com, sp-blue.com	example.com sp-blue.com

C.2.2 Network Selection Scenarios in which OI is required

In the following scenarios, the mobile device has been provisioned with a PPS MO having data in its HomeSP container node (see section 9.1) as shown in Table 22; furthermore, the PPS MO has its Credential/Realm node set to a value of "sp-blue.com".

Table 22: Example HomeSP Provisioned Data Set #2

Home SP Child Node	Data
NetworkID/<x+>/SSID	Hotspot 2.0 Wi-Fi

Home SP Child Node	Data
NetworkID/<x+>/HESSID	001d2e0011a0
FriendlyName	Blue
IconURL	http://www.sp-blue.com/icons/blue_icon.png
FQDN	sp-blue.com
HomeOIList/x2/HomeOI	001bc500bb
HomeOIList/x2/HomeOIRequired	TRUE
OtherHomePartners/f1/FQDN	example.com
RoamingConsortiumOI	001bc50050, 001bc500b5

C.2.2.1 Scenario #4

A mobile device having the PPS MO data described in section C.2.2 Table 22 detects several hotspots advertising data as described in Table 23. In this scenario, the mobile device selects, associates and authenticates to Hotspot #2. The following rationale is relevant to this selection:

- The mobile device mobile device queries an AP in each Hotspot for the Roaming Consortium ANQP-element. The mobile device shall do this because in the PPS MO there is a HomeOIList having a HomeOIRequired leaf node set to TRUE. In the ANQP query, only the response from Hotspot #2 AP contains the required OI, "001bc500bb". Therefore, the mobile device joins Hotspot #2.
- Note: this example illustrates that provisioning HomeOIRequired leaf node can have the effect of causing a mobile device to select a visited network when a home network is available (i.e., the Wi-Fi AN operated by "Green").

Table 23: Hotspot Environment #4

Parameter	Hotspot #1	Hotspot #2	Hotspot #3
Hotspot 2.0 element	present	present	present
SSID	Blue Wi-Fi	Fast Wi-Fi	Downtown Wi-Fi
Domain Name List	sp-blue.com	sp-green.com	example.com
OIs	001d2e	001bc500bb	001bc50050
NAI Realms	sp-blue.com,	sp-green.com, sp-blue.com	example.com, sp-blue.com

C.2.3 Network Selection Scenarios with Home SP Policy

In the following scenarios, the mobile device has been provisioned with a PPS MO having data in its HomeSP container node (see section 9.1) as shown in Table 18 and Policy container node as shown in Table 24; furthermore, the PPS MO has its Credential/Realm node set to a value of "sp-blue.com".

Table 24: Example Policy Provisioned Data Set #1

Policy Child Node	Data
PreferredRoamingPartnerList/x1/FQDN_Match	sp-blue.com,exactMatch
PreferredRoamingPartnerList/x1/Priority	10
PreferredRoamingPartnerList/x1/Country	*
PreferredRoamingPartnerList/x2/FQDN_Match	sp-green.com,includeSubdomains
PreferredRoamingPartnerList/x2/Priority	140
PreferredRoamingPartnerList/x2/Country	*
PreferredRoamingPartnerList/x3/FQDN_Match	sp-orange.com,exactMatch
PreferredRoamingPartnerList/x3/Priority	5
PreferredRoamingPartnerList/x3/Country	*

C.2.3.1 Scenario #5

A mobile device having the PPS MO data described in section C.2.1 detects several hotspots advertising data as described in Table 25. In this scenario, the mobile device selects, associates and authenticates to Hotspot #3. The following rationale is relevant to this selection:

- The mobile device queries an AP in each Hotspot for the Domain Name List. In the ANQP query response from Hotspot #1 AP, the mobile device discovers that none of returned Domain Name matches the FQDN in HomeSP/FQDN and concludes that all hotspots are a visited network. Hotspot #2 has a roaming priority of 140 (see Table 24), and Hotspots #1 and #3 have a priority of 128 (the default priority, see the description of PreferredRoamingPartnerList/<x+>/Priority leaf node in section 9.1.2). However, since SP Blue is not a roaming partner of SP Pink, the mobile device joins Hotspot #3.

Table 25: Hotspot Environment #5

Parameter	Hotspot #1	Hotspot #2	Hotspot #3
Hotspot 2.0 element	present	present	present
SSID	Pink Rocks	Fast Wi-Fi	Downtown Wi-Fi
Domain Name List	sp-pink.com	airports.sp-green.com	sp-red.com
NAI Realms	sp-pink.com	sp-green.com, sp-blue.com	sp-red.com, sp-blue.com

C.2.3.2 Scenario #6

A mobile device having the PPS MO data described in section C.2.1 detects several hotspots advertising data as described in Table 26. In this scenario, the mobile device selects, associates and authenticates to Hotspot #2. The following rationale is relevant to this selection:

- The mobile device queries an AP in each Hotspot for the Domain Name List. Hotspot #1 is a home network having a priority of 10, Hotspot #2 and #3 are visited networks and have roaming priorities of 5 (see Table 24), and 128 respectively. The mobile device joins Hotspot #2 since it has the highest priority.
- Note: this example illustrates that provisioning a PreferredRoamingPartnerList can have the effect of causing a mobile device to select a visited network when a home network is available (i.e., the Wi-Fi AN operated by SP Orange).

Table 26: Hotspot Environment #6

Parameter	Hotspot #1	Hotspot #2	Hotspot #3
Hotspot 2.0 element	present	present	present
SSID	Blue Wi-Fi	Ultra Wi-Fi	Downtown Wi-Fi
Domain Name List	sp-blue.com	sp-orange.com	sp-red.com
NAI Realms	present	sp-orange.com, sp-blue.com	sp-red.com, sp-blue.com

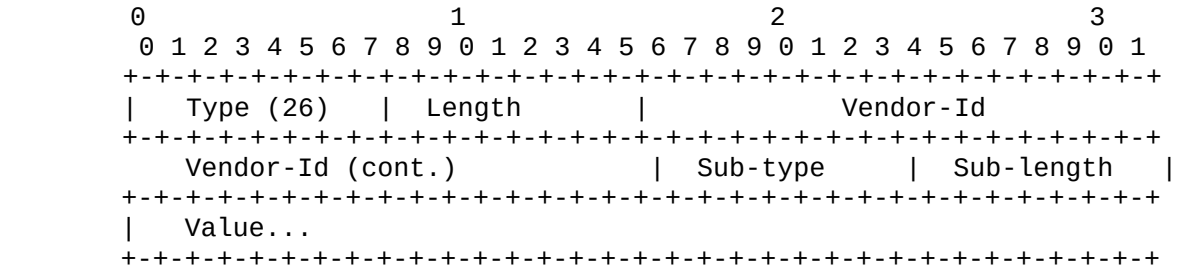
.

Annex D : Wi-Fi Alliance Vendor-Specific RADIUS attributes (informative)

This annex describes the set of Wi-Fi Alliance (WFA) vendor-specific RADIUS attributes that are used in WFA protocols between multiple vendors. The attribute follows the format described for Vendor-Specific attribute in RFC 2865. The Vendor-Id field is set to the Private Enterprise Number 40808 allocated for Wi-Fi Alliance. These RADIUS attributes are exchanged between an AAA server and AP (Authenticator).

RADIUS messages may include multiple WFA vendor-specific attributes.

WFA vendor-specific attributes use the following format:



Type: 26 for Vendor-Specific

Length: Length of the entire attribute including Type, Length, Vendor-Id, Sub-type, Sub-length and Value fields >= 8

Vendor-Id: 4 octets encoding the WFA Vendor-Id of 40808 in network byte order

(Note: the first octet is 0 since the Vendor-Id is 24 bits long)

Sub-type: Sub-type identifying WFA vendor-specific RADIUS attribute.

Sub-length: Length of the sub-attribute in octets including Sub-type, Sub-length and Value fields.

D.1 Wi-Fi Alliance Vendor-Specific RADIUS attribute sub-type formats

1 - HS2.0 subscription remediation needed

(may be included in Access-Accept messages)

2 - HS2.0 AP version

(may be included in Access-Request messages)

3 - HS2.0 mobile device version

(may be included in Access-Request messages)

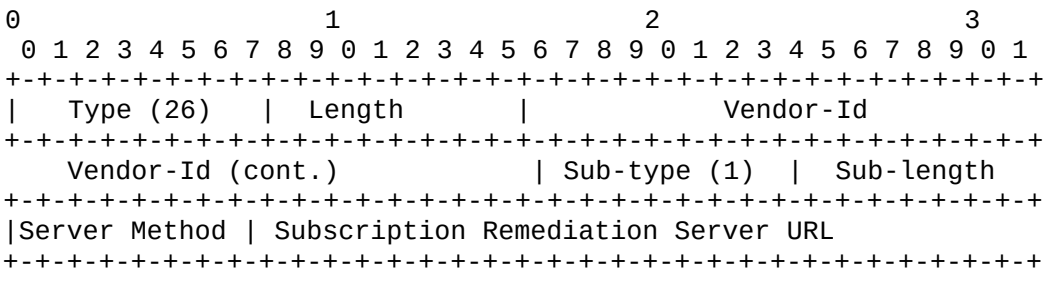
4 – HS2.0 deauthentication request

(may be included in Access-Accept or Disconnect-Request messages)

5 – HS2.0 session information URL

(may be included in Access-Accept messages)

D.1.1 HS2.0 subscription remediation needed



Field Definitions

Type field: 26 for Vendor-Specific

Length field: Length of the entire attribute including Type, Length, Vendor-Id, Sub-type, Sub-length and Subscription Remediation Server URL fields >= 8

Vendor-Id field: 4 octets encoding the WFA Vendor-Id of 40808 in network byte order

(Note: the first octet is 0 since the Vendor-Id is 24 bits long.)

Sub-type field: 1 - HS2.0 subscription remediation needed

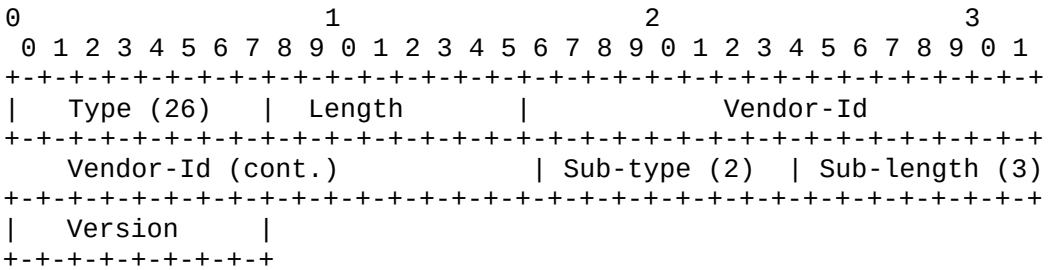
Sub-length field: Length of the sub-attribute in octets including Sub-type, Sub-length, Server Method and URL fields.

Server Method field: Provisioning protocol supported by the subscription remediation server whose value is chosen from Table 10. This is an optional field.

Subscription Remediation Server URL field: This is a UTF-8 encoded field formatted in accordance with [15]. This is an optional field. This field shall be present when the Server Method field is present and shall be absent when the Server Method field is absent.

This attribute may be included in Access-Accept message to indicate that the mobile device requires subscription remediation. This requests the HS2.0 AP to send a WNM-Notification Request frame containing a Subscription Remediation subelement to the mobile device after a successfully completed authentication. The value of the Subscription Remediation Server URL field is used as the server URL in that notification. The AAA server should only provide the values of the Server Method field and Subscription Remediation Server URL field to an AP at one of its home hotspots. The AP may also use this attribute as a request to limit network access for the station to allow connection only to the remediation server.

D.1.2 HS2.0 AP version



Field Definitions

Type field: 26 for Vendor-Specific

Length field: Length of the entire attribute including Type, Length, Vendor-Id, Sub-type, Sub-length and Version fields = 9.

Vendor-Id field: 4 octets encoding the WFA Vendor-Id of 40808 in network byte order

(Note: the first octet is 0 since the Vendor-Id is 24 bits long.)

Sub-type field: 2 - HS2.0 AP version

Sub-length field: Length of the sub-attribute in octets including Sub-type, Sub-length and Version fields, 3.

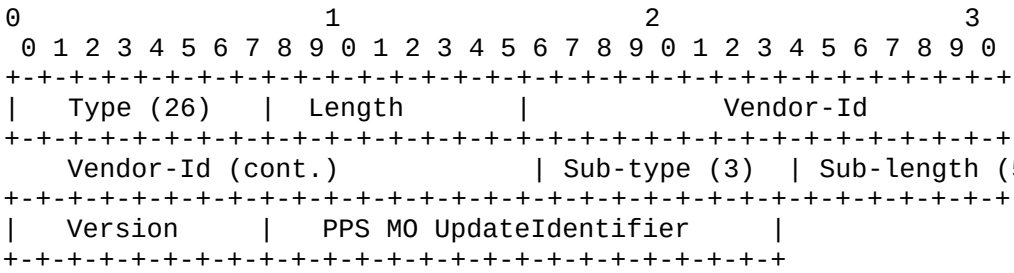
Version field: HS2.0 Release number supported by the AP

0 = Release 1

1 = Release 2

This attribute may be included in Access-Request message to indicate that the sending HS2.0 AP supports HS2.0 with the specified release.

D.1.3 HS2.0 mobile device version



Field Definitions

Type field: 26 for Vendor-Specific

Length field: Length of the entire attribute including Type, Length, Vendor-Id, Sub-type, Sub-length, Version fields = 11.

Vendor-Id field: 4 octets encoding the WFA Vendor-Id of 40808 in network byte order

(Note: the first octet is 0 since the Vendor-Id is 24 bits long.)

Sub-type field: 3 - HS2.0 mobile device version

Sub-length field: Length of the sub-attribute in octets including Sub-type, Sub-length, Version and PPS MO UpdateIdentifier fields, 5.

Version field: HS2.0 Release number supported by the mobile device

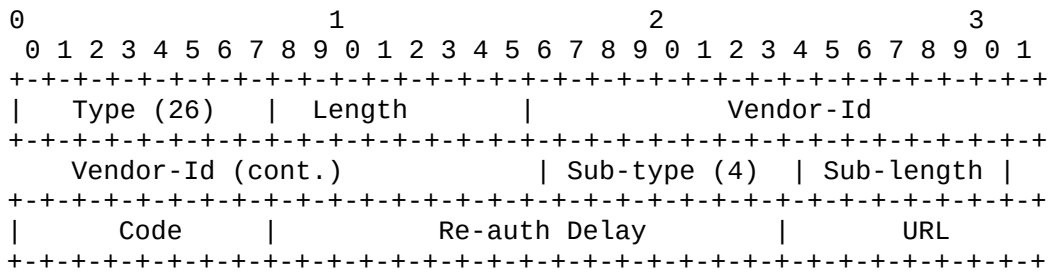
0 = Release 1

1 = Release 2

PPS MO UpdateIdentifier field: PerProviderSubscription Management Object Update Identifier provided by the mobile device indicating the specific version of the MO present in the device. This is a 16-bit unsigned integer in network byte order.

This attribute may be included in Access-Request message to indicate that the mobile device for which this Access-Request is transmitted supports HS2.0 with the specified release number. If the mobile device does not include HS2.0 indication element in the (Re)Association Request frame, this attribute is not included.

D.1.4 HS2.0 deauthentication request



Field Definitions

Type field: 26 for Vendor-Specific

Length field: Length of the entire attribute including Type, Length, Vendor-Id, Sub-type, Sub-length, Code, Re-auth Delay and UI message fields.

Vendor-Id field: 4 octets encoding the WFA Vendor-Id of 40808 in network byte order

(Note: the first octet is 0 since the Vendor-Id is 24 bits long.)

Sub-type field: 4 - HS2.0 Deauthentication Request

Sub-length field: Length of the sub-attribute in octets included Sub-type, Sub-length, Code, Re-auth Delay and URL message fields.

Code field: Reason the mobile device is being deauthenticated

0 = User's subscription does not allow or no longer allows access at this BSS

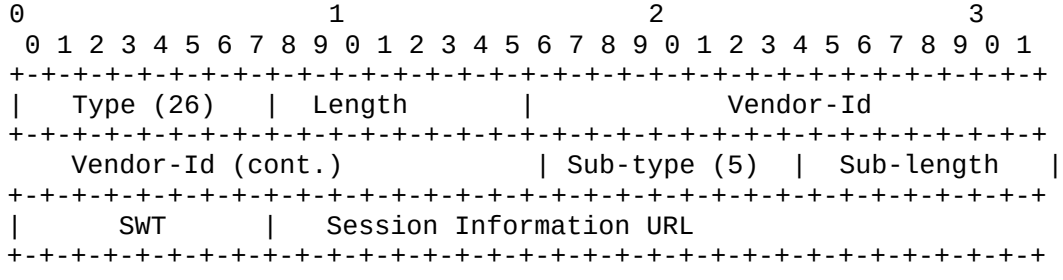
1 = User's subscription does not allow or no longer allows access at this ESS

Re-auth Delay field: delay in seconds that a mobile device waits before attempting reassociation to the same BSS, ESS (as indicated in the Code). A value of 0 indicates the Re-auth Delay may be decided by the mobile device.

URL field: UTF-8 encoded field which is a URL, formatted in accordance with [17],which provides a webpage explaining why the mobile device was not authorized (or is no longer authorized) and will be deauthenticated from the BSS or ESS. It is implementation dependent how the device will use this URL.

This attribute may be included in Access-Accept or Disconnect-Request message to cause the mobile device to be deauthenticated from the Wi-Fi AN. This message is only used when the mobile device is HS2.0 Release 2 capable or higher.

D.1.5 HS2.0 session information URL



Field Definitions

Type field: 26 for Vendor-Specific

Length field: Length of the entire attribute including Type, Length, Vendor-Id, Sub-type, Sub-length, SWT and Session Information URL fields.

Vendor-Id field: 4 octets encoding the WFA Vendor-Id of 40808 in network byte order

(Note: the first octet is 0 since the Vendor-Id is 24 bits long.)

Sub-type field: 5 - HS2.0 session information URL

Sub-length field: Length of the sub-attribute in octets including Sub-type, Sub-length, SWT and Session Information URL fields.

SWT field: Session Warning Time is the number of minutes of advance notice an AP shall provide to the mobile device before terminating its session. When SWT is set to the special value of 255, the AP (802.1X authenticator) chooses the session warning time value.

Session Information URL field: URL, formatted in accordance with [17], which is transmitted to a mobile device in a BSS Transition Management Request frame SWT minutes before the mobile device's session is terminated. The URL provides the location of a webpage with information for the user on how to extend the session.

This attribute may be optionally be included in Access-Accept message, requesting AP to send a BSS Transition Management Request frame SWT minutes before the mobile device's session is terminated, warning the user that the session is about to end. The URL provides the location of a web page that contains information on how to extend the session (perhaps for a fee).

Annex E : Standardized OSU registration flow (normative)

E.1 General

A mobile device may present the OSU subscription plan selection and registration forms to the user in a consistent manner. The mobile device may populate some or all of the fields before presentation to the user. To assist a mobile device in identifying relevant fields and presenting relevant information to the user, subscription plan selection and registration forms shall include standardized tags as defined in this section.

The web pages for a registration flow using the standardized UI include the following XML:

```
<!--
    <RegistrationProtocol>
    {
        Elements related to registration flow
    }
    </RegistrationProtocol>
-->
```

E.2 OSU Registration Flow

A standardized OSU registration flow proceeds as follows:

- The mobile device sends an HTTP GET to the OSU server. The HTTP GET includes an Accept-Language in the header field indicating one of the languages supported by OSU as determined from the DevInfo MO in the sppPostDevData & sppPostDevDataResponse exchange or the OMA DM Package 1 and Package 2 exchange.
- The OSU server responds with an HTML page with RegistrationProtocol XML with the ServerGroup group element present. The ServerGroup indicates information needed from the user to complete online signup. The ServerGroup indicates the supported versions for the RegistrationProtocol. The mobile device gathers the needed information from the user and returns it to the OSU server in an HTTP POST containing the RegistrationProtocol XML with the ClientGroup group element present. The mobile device shall select the highest version of the RegistrationProtocol for the XML data. If the mobile device does not support any of the version presented in the by the server, then it shall respond with the appropriate StatusCode (19 - Client and Server versions are incompatible). The OSU server responds with an HTML page with RegistrationProtocol XML with the StatusGroup group element present. The StatusGroup indicates success (online signup complete) or indicates that an error occurred together with information on how to correct the error. The mobile device may then return new information in another ClientGroup group element with the OSU server issuing another StatusGroup in turn. This continues until the mobile device disconnects or the OSU server issues a StatusGroup indicating success.

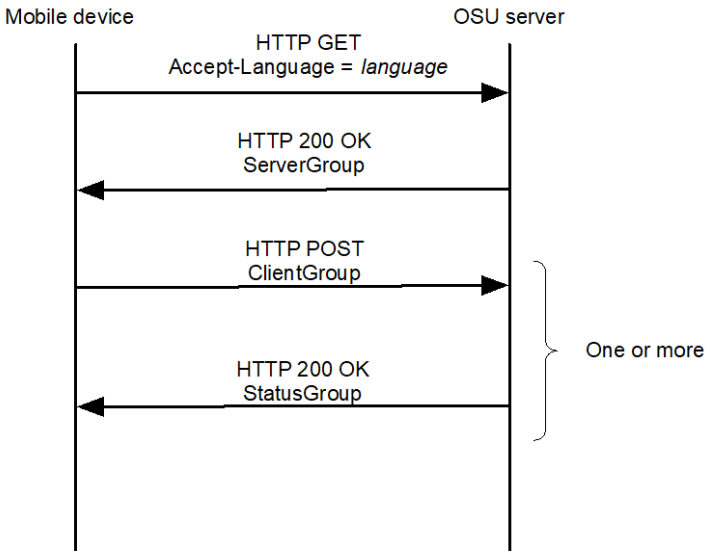


Figure 95: OSU Registration Flow

E.3 OSU Registration Schema

The namespace for Registration Protocol is http://www.wi-fi.org/specifications/hotspot2dot0/v1.0/rep, see [52].

E.3.1 The RegistrationProtocol element

The RegistrationProtocol element contains a ServerGroup in the initial message sent by the OSU server. The RegistrationProtocol element contains a ClientGroup in messages sent by the mobile device. The RegistrationProtocol element contains a StatusGroup in messages from the OSU server following the initial message.

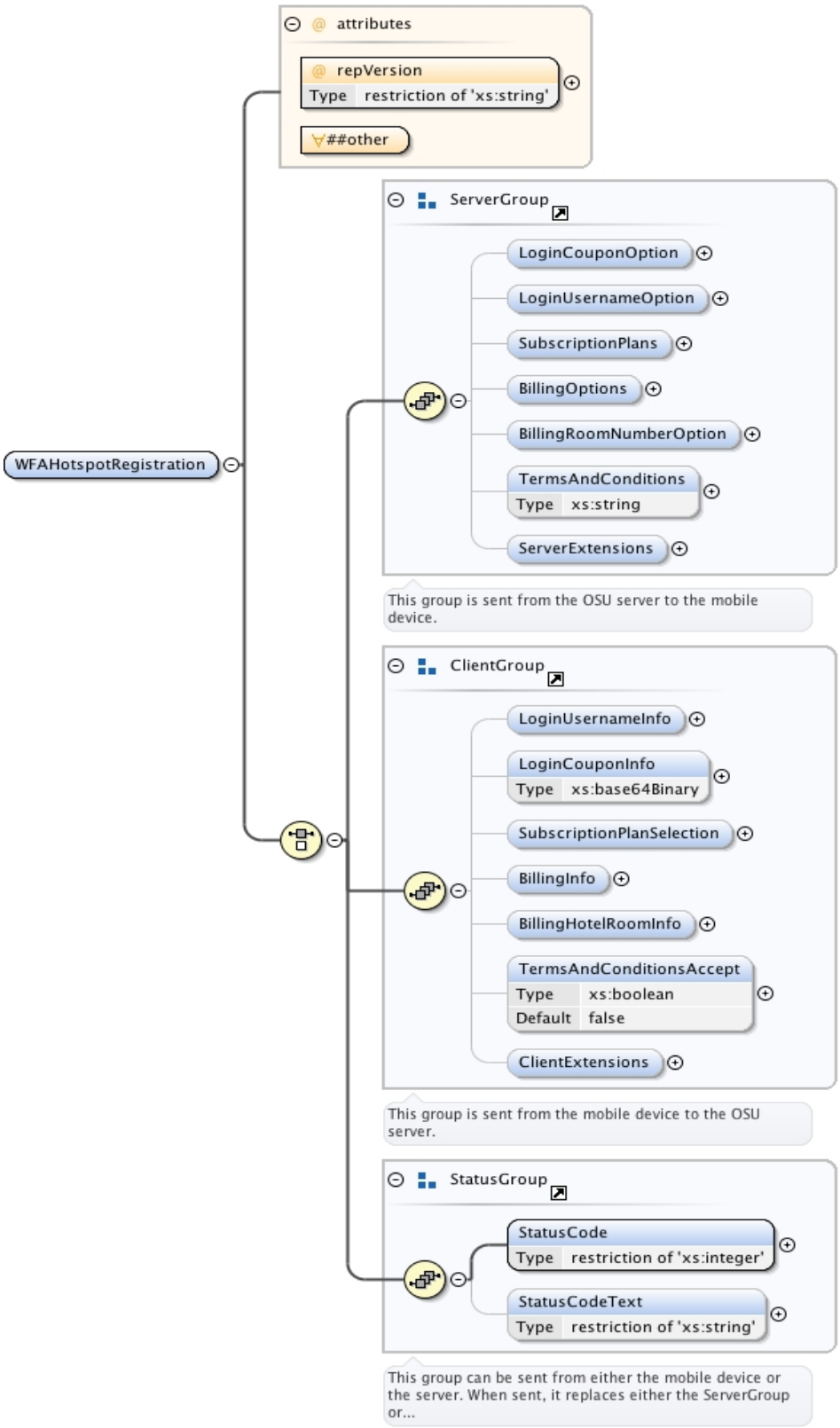


Figure 96: Registration Protocol Schema

E.3.2 ServerGroup element group

The ServerGroup element group is present in the initial message from the OSU server.

The ServerGroup element group may include one or more of the following elements:

- LoginCouponOption
- LoginUsernameOption
- SubscriptionPlans

If the SubscriptionPlans element is present, the ServerGroup shall include either or both of the following elements:

- BillingOptions
- BillingRoomNumberOption

If the SubscriptionPlans element is not present, then the ServerGroup shall not include either the BillingOptions element or the BillingHotelRoomOption element.

There are three billing options provided as follows:

- Credit Card – in this option, the mobile device provides credit card information and the OSUS bills to that credit card.
- Web Payment – in this option, the mobile device's browser is re-directed to the web payment service chosen by the user (e.g., PayPal); the user then follows directions provided by the web payment service.
- Cellular operator billing – in this option, the mobile device provides its cellular phone number (MSISDN) and its Home SP sends an SMS with a onetime password with which to obtain network access. The mobile device uses EAP-TTLS, its IMSI as the username and the one time password for EAP authentication to the hotspot.

The ServerGroup element group shall include the SupportedVersions element, which provides information of the supported Standardized XML tag versions.

The ServerGroup element group may include the TermsAndConditions element.

The ServerGroup element group may include the ServerExtensions element.

E.3.2.1 LoginCouponOption element

The LoginCouponOption is present in the initial response from the OSU server if a coupon field is to be presented to the user. The OSU server provides the text for the coupon field in the CouponFieldLabel element.

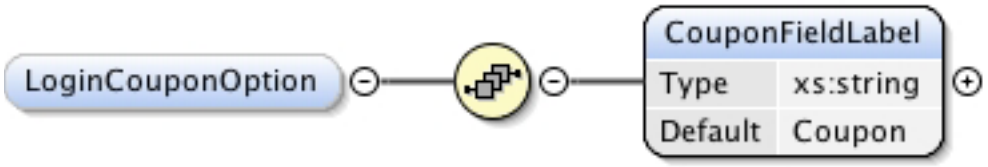


Figure 97: LoginCouponOption element

E.3.2.2 LoginUsernameOption element

The LoginUsernameOption element is present in the initial response from the OSU server if the username and password fields are to be presented to the user. The OSU server provides text labels for the fields in the UsernameFieldLabel and PasswordFieldLabel.

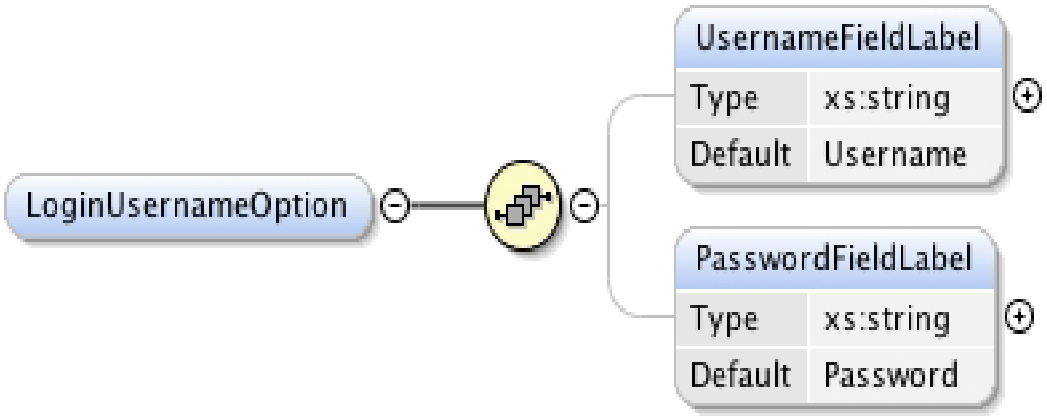


Figure 98: LoginUsernameOption element

E.3.2.3 SubscriptionPlans element

The SubscriptionPlans element is present in the initial response from the OSU server if subscriptions options are to be presented to the user. Introductory text for the subscription options is provided with the SubscriptionPlansIntro element. One or more SubscriptionPlanOption elements are also present, one for each subscription plan offered.

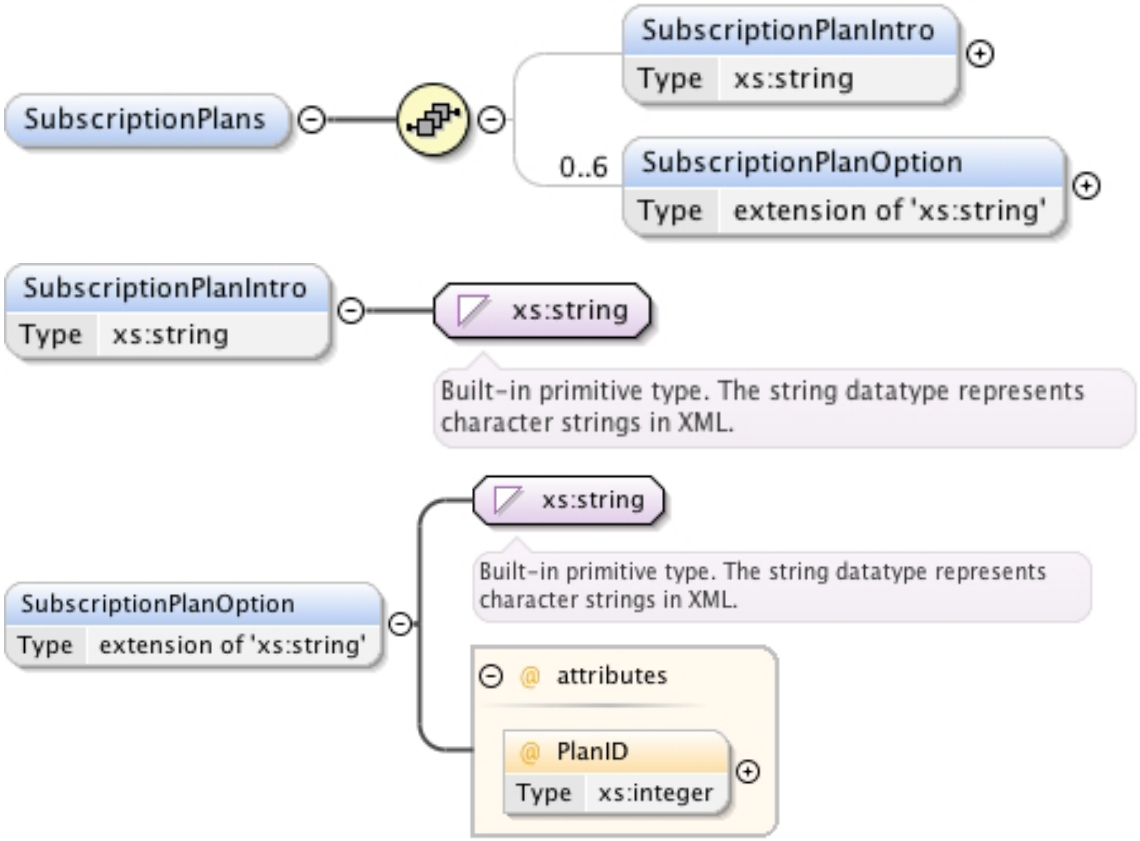


Figure 99: SubscriptionPlans element

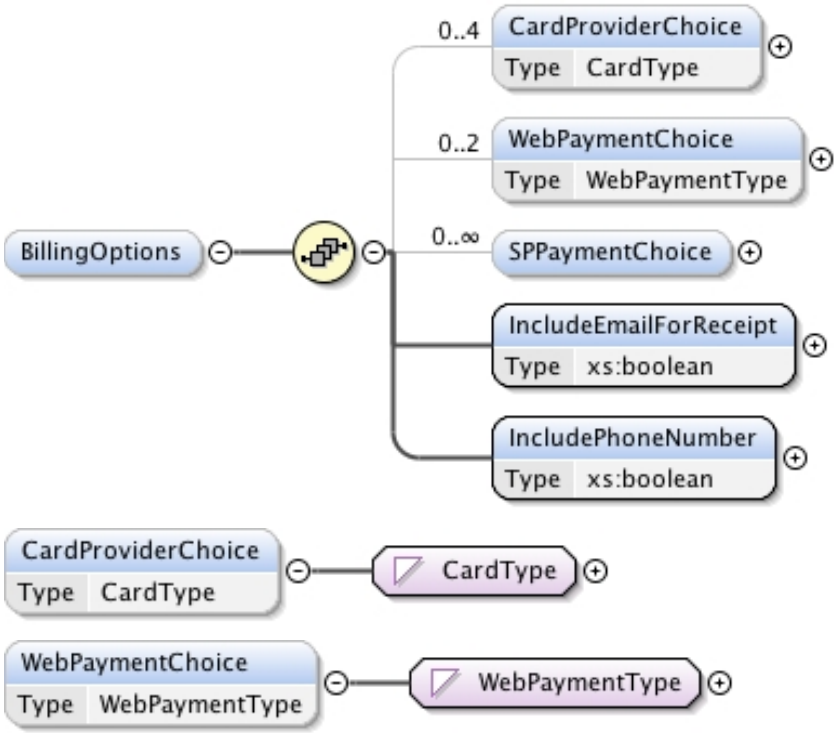
E.3.2.4 BillingOptions element

The BillingOptions is present if the SubscriptionPlans element is present and payment can be made using a credit card, a web-payment site (e.g., PayPal) or by billing to the user's home cellular operator. The choice of billing options is at the discretion of the hotspot operator. The hotspot operator may use one, two or all 3 methods of payment.

If payment can be made using a credit card, one or more CreditCardChoice elements are present; if payment cannot be made using a credit card, then zero CreditCardChoice elements are present. If payment can be made using a web-payment service, one or more WebPaymentChoice elements are present; if payment cannot be made using a web-payment service, then zero WebPaymentChoice elements are present. If payment can be made using a user's home cellular operator, one or more SPPaymentChoice elements are present; if payment cannot be made using a cellular operator, then zero SPPaymentChoice elements are present.

The IncludeEmailForReceipt element indicates whether or not the user should provide an email address to which a receipt will be sent. This element is only present if at least one CreditCardChoice element is present or at least one WebPaymentChoice is present.

The IncludePhoneNumber element indicates that a phone number is required as part of the credit card validation process. This element is only present if at least one CreditCardChoice element is present or at least one WebPaymentChoice is present.



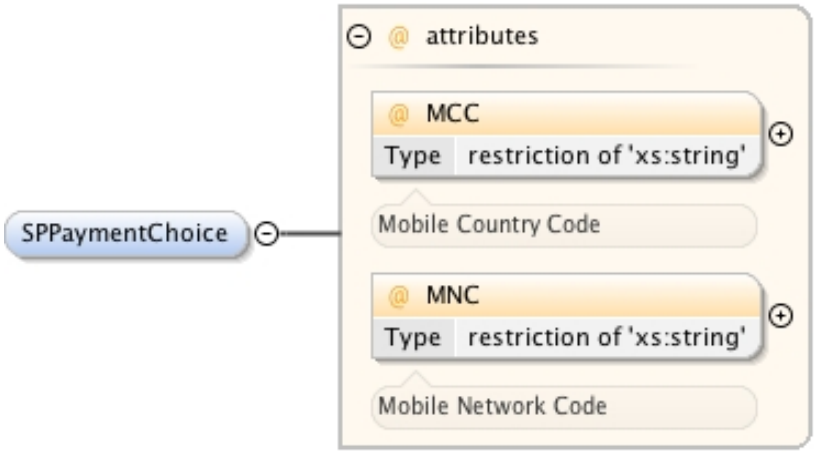


Figure 100: BillingOptions element

An SP (cellular operator) is identified by its PLMN ID, which is the {MCC, MNC} duple. As a hotspot operator may have billing agreements with many cellular operators, many SPPaymentChoice elements may be present in the BillingOptions element. A mobile device searches this list for a PLMN ID which matches the PLMN ID in the IMSI drawn from its (U)SIM; if there is an exact match, billing is possible for that user. Note that a given cellular operator may be identified by more than one PLMN ID.

E.3.2.5 BillingRoomNumberOption element

The BillingRoomNumberOption element is preset if the SubscriptionPlan element is present and payment can be made to a hotel room. The RequestLastName element indicates true if a last name is needed to validate the room number entered.

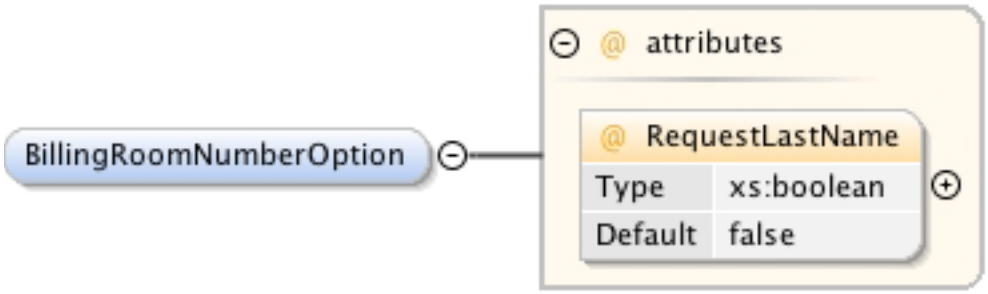


Figure 101: BillingRoomNumberOption element

E.3.2.6 TermsAndConditions element

The TermsAndConditions element is present in the initial OSU sever response if the user is required to accept written terms and conditions in order to gain access to the network. The element provides the text of the terms and conditions to be accepted.

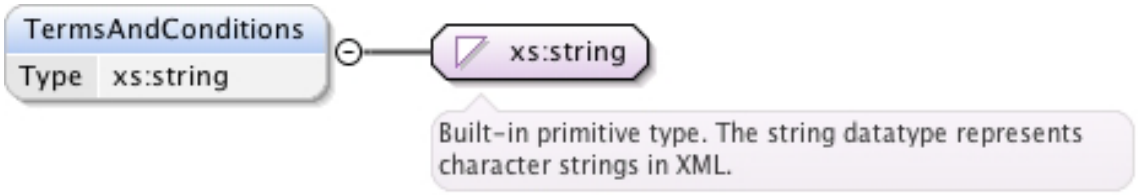


Figure 102: TermsAndConditions element

E.3.2.7 ServerExtensions element

The ServerExtensions element may be present in the initial OSU server response. This element is a container for zero or more any elements, which may be used for extensibility of this protocol in the future or for vendor specific extensions.

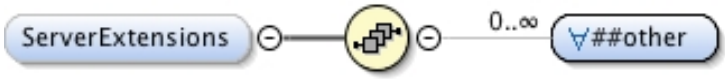


Figure 103: ServerExtensions element

E.3.3 ClientGroup element group

The ClientGroup element group is present in messages from the mobile device.

The ClientGroup element group shall include one and only one of the following elements: LoginUsernameInfo, LoginCouponInfo or SubscriptionPlanSelection.

If the SubscriptionPlanSelection element is present, then the ClientGroup element group shall include one and only one of the following elements: BillingInfo or BillingHotelRoomInfo.

The ClientGroup element group may include the ClientExtensions element.

E.3.3.1 LoginUsernameInfo element

The LoginUsernameInfo element provides the username and password that, if validated, enables network access.

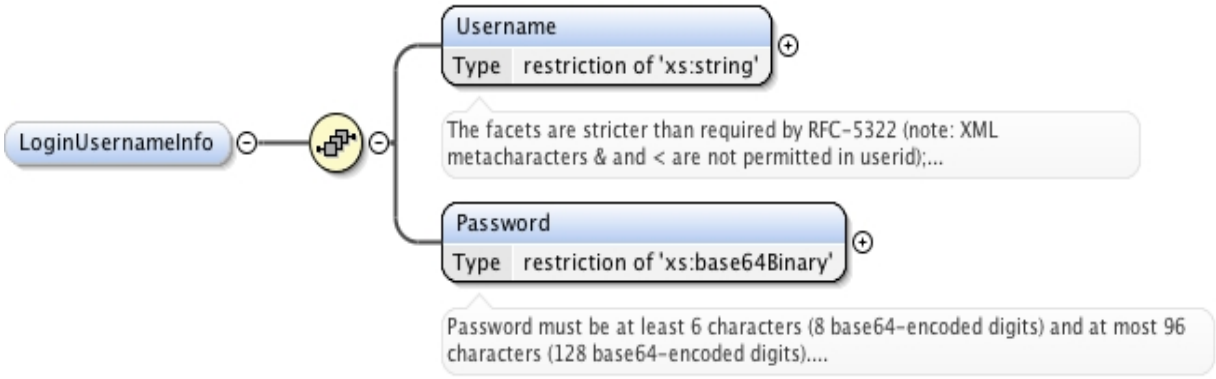


Figure 104: LoginUsernameInfo element

E.3.3.2 LoginCouponInfo element

The LoginCouponInfo element is indicates the user entered coupon that, if validated, enables network access.

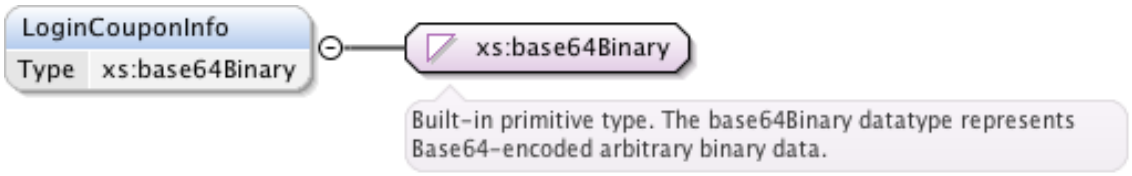


Figure 105: LoginCouponInfo element

E.3.3.3 SubscriptionPlanSelection element

The SubscriptionPlanSelection element indicates the selected subscription plan.

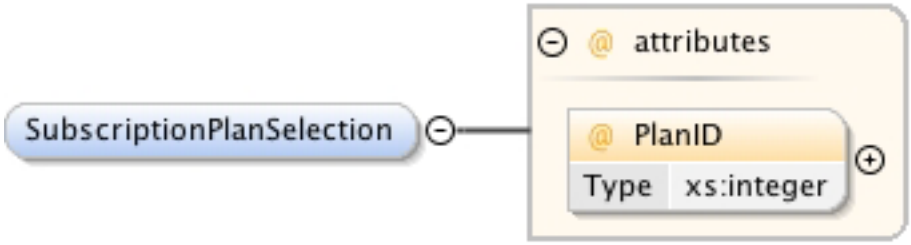


Figure 106: SubscriptionPlanSelection element

E.3.3.4 BillingInfo

The BillingInfo element provides billing details.

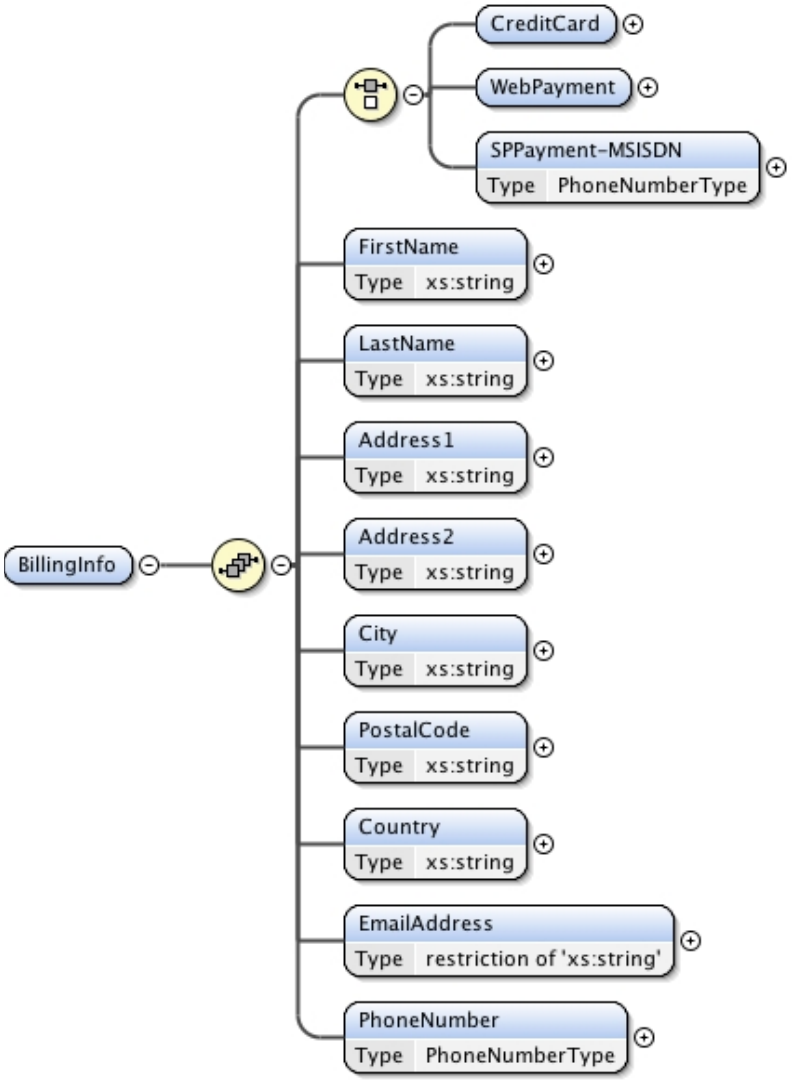


Figure 107: BillingInfo element

E.3.3.5 BillingHotelRoomInfo element

The BillingHotelRoomInfo element provides information for billing to a hotel room. The RoomNumber element shall always be present and indicates the room number to which billing should be applied. The LastName element is shall be present if the previous ServerGroup/BillingRoomNumberOption/LastNameRequired indicates true.

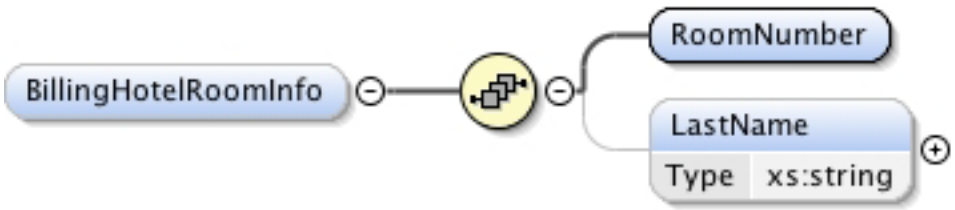


Figure 108: BillingHotelRoomInfo element

E.3.3.6 ClientExtensions element

The ClientExtensions element may be present in the initial OSU server response. This element is a container for zero or more any elements, which may be used for extensibility of this protocol in the future or for vendor specific extensions.

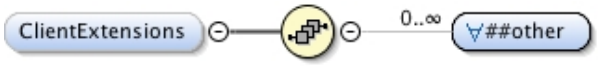


Figure 109: ClientExtensions element

E.3.4 The StatusGroup element

The StatusGroup is present in an OSU server message that follows a client issued ClientGroup and provides the status indication for the submitted information.

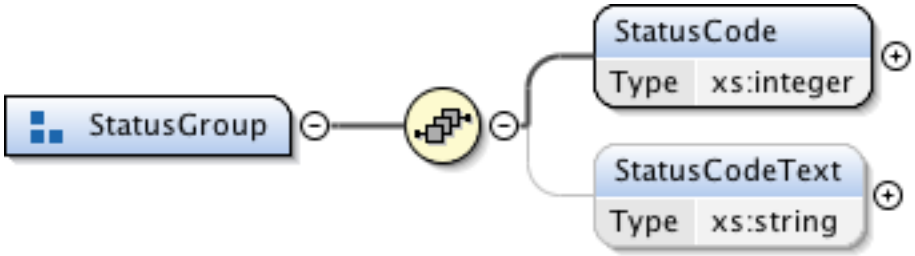


Figure 110: StatusGroup element

E.3.4.1 StatusCode element

The StatusCode element provides a status code for the information submitted in the immediately previous ClientGroup.

Possible error codes are listed in Table 27.

Table 27: Possible StatusCode values

StatusCode	Description
0	Success
1	Generic processing error; StatusCodeText provides details
2	Missing Required Information; StatusCodeText provides details
3	Invalid Login Information (username or password)
4	Invalid Coupon
5	Room Number and Name mismatch
6	Invalid Credit Card Type
7	Invalid Card Number
8	Invalid Expiration Date
9	Missing Security Code
10	Invalid Security Code
11	Credit Card Number and Name mismatch
12	Credit Card Number and Billing Address are invalid
13	Acceptance of Terms and Conditions is required
14	Credit Card Expired
15	Transaction Failed
16	Names does not match account
17	Session timeout
18	Service temporarily unavailable
19	Client and Server versions are incompatible
20	Account disabled
21	Password expired
22	Maximum Login attempts exceeded
23	Invalid Credit Card Name

E.3.4.2 StatusText element

The StatusText element provides text to help the user resolve the particular error indicated by the StatusCode.

E.4 XML schema

The XML schema for Registration Protocol is provided in [52].

E.5 Example transaction 1

The mobile device sends an HTTP GET request to the OSU server indicating its language preference using the Accept-Language header field. The language preference has previously been negotiated.

The OSU server returns an HTML page with embedded XML. The remainder of this example details only the included XML.

The OSU server indicates an option for login using username/password and an option for selecting a subscription plan. If a subscription plan is selected, billing is to either a credit card or hotel room number. For credit card billing, a list of accepted credit card vendors is provided. The OSU server requires that the provided terms and conditions are accepted.

```
<!--
<?xml version="1.0" encoding="UTF-8"?>
<RegistrationProtocol
 xmlns="http://www.wi-fi.org/specifications/hotspot2dot0/v1.0/rep"
 repVersion="1.0">
    <LoginUsernameOption>
        <UsernameFieldLabel>My Company ID</UsernameFieldLabel>
        <PasswordFieldLabel>Password</PasswordFieldLabel>
    </LoginUsernameOption>
    <SubscriptionPlans>
        <SubscriptionPlanIntro>Signup Today. Get a really good deal. </SubscriptionPlanIntro>
        <SubscriptionPlanOption PlanID="1">1 for one hour</SubscriptionPlanOption>
        <SubscriptionPlanOption PlanID="2">2 for one day</SubscriptionPlanOption>
        <SubscriptionPlanOption PlanID="3">3 for one month, billed monthly</SubscriptionPlanOption>
    </SubscriptionPlans>
    <BillingOptions>
        <CardProviderChoice>Visa</CardProviderChoice>
        <CardProviderChoice>Master Card</CardProviderChoice>
        <CardProviderChoice>American Express</CardProviderChoice>
        <CardProviderChoice>Discover</CardProviderChoice>
        <WebPaymentChoice>PayPal</WebPaymentChoice>
        <SPPaymentChoice MCC="310" MNC="02" />
        <IncludeEmailForReceipt>true</IncludeEmailForReceipt>
        <IncludePhoneNumber>false</IncludePhoneNumber>
    </BillingOptions>
    <BillingRoomNumberOption RequestLastName="true" />
    <TermsAndConditions>You agree not to sue us.</TermsAndConditions>
</RegistrationProtocol>
-->
```

Figure 111: Sample subscription plan options

The mobile device responds with user selected subscription plan and billing details. The user also indicates acceptance of the terms and conditions.

```
<!--
<?xml version="1.0" encoding="UTF-8"?>
<RegistrationProtocol
    xmlns="http://www.wi-fi.org/specifications/hotspot2dot0/v1.0/rep"
    repVersion="1.0">
    <SubscriptionPlanSelection PlanID="2"/>
    <BillingInfo>
        <CreditCard>
            <CardProvider>American Express</CardProvider>
            <CardHolderName>John W Smith</CardHolderName>
            <CardNumber>1234000056780000</CardNumber>
            <CardExpirationDate>2015-05</CardExpirationDate>
            <CardSecurityCode>667</CardSecurityCode>
        </CreditCard>
        <FirstName>John</FirstName>
        <LastName>Smith</LastName>
        <Address1>1 Infinite Loop</Address1>
        <City>Cupertino</City>
        <PostalCode>95014</PostalCode>
        <Country>USA</Country>
        <EmailAddress>john.smith@noemail.com</EmailAddress>
        <PhoneNumber>+1-408-555-1212</PhoneNumber>
    </BillingInfo>
    <TermsAndConditionsAccept>true</TermsAndConditionsAccept>
</RegistrationProtocol>

-->
```

Figure 112: User selects an option and provide details

The OSU server indicates success.

```
<!--
<?xml version="1.0" encoding="UTF-8"?>
<RegistrationProtocol
    xmlns="http://www.wi-fi.org/specifications/hotspot2dot0/v1.0/rep"
    repVersion="1.0">
    <StatusCode>0</StatusCode>
</RegistrationProtocol>

-->
```

Figure 113: OSU server success