Open Connection Manager API Requirements

Approved Version 1.0 – 26 Jan 2016

Open Mobile Alliance

OMA-RD-OpenCMAPI-V1_0-20160126-A

Contents

Figures

Tables

1. Scope (Informative)

This document defines the requirements for the OMA Open Connection Manager API (OpenCMAPI) V1.0.

The aim of the OMA OpenCMAPI V1.0 RD is to address requirements for:

- all connectivity and connection management aspects relevant such as:
  - Connection/Disconnection
  - All relevant elements related to the connection or the device and more specifically all elements necessary and useful to be provided to any UI and user experience or to any application needing information status on the connection.
  - Additional services such as SMS, USSD, GNSS… when associated to connection management and relevant for the device considered
  - Data service especially PUSH service configuration on the device side.

The intention is that this API is

- OS independent
- Supporting Multi-Instance (several applications/services can use it in parallel if necessary)
- Applicable to different types of devices requiring access to mobile internet such as Mobile Broadband devices, Wireless routers, M2M, Smartphones, Tablets, Cloud Devices…
- User Interface independent

2. References

2.1 Normative References

[3GPP TR 21.905] "TR 21.905 Technical Specification Group Services and System Aspects; Vocabulary for 3GPP Specifications", 3rd Generation Partnership Project (3GPP), URL:http://www.3gpp.org/ftp/Specs/archive/21_series/21.905/

[3GPP TS 22.011] "TS 22.011 Technical Specification Group Services and System Aspects; Service accessibility", 3rd Generation Partnership Project (3GPP), URL:http://www.3gpp.org/ftp/Specs/archive/22_series/22.011/

[3GPP TS 22.022] "TS 22.022 Technical Specification Group Services and System Aspects; Personalisation of Mobile Equipment (ME), Mobile functionality specification", 3rd Generation Partnership Project (3GPP), URL:http://www.3gpp.org/ftp/Specs/archive/22_series/22.022/

[3GPP TS 22.030] "TS 22.030 Technical Specification Group Services and System Aspects; Man-Machine Interface (MMI) of the User Equipment (UE)", 3rd Generation Partnership Project (3GPP), URL:http://www.3gpp.org/ftp/Specs/archive/22_series/22.030/

[3GPP TS 24.090] "TS 24.090 Technical Specification Group Core Network and Terminals; Unstructured Supplementary Service Data (USSD)", 3rd Generation Partnership Project (3GPP), URL:http://www.3gpp.org/ftp/Specs/archive/24_series/24.090/

[3GPP TS 31.101] "TS 31.101 Technical Specification Group Core Network and Terminals; UICC-terminal interface; Physical and logical characteristics", 3rd Generation Partnership Project (3GPP), URL:http://www.3gpp.org/ftp/Specs/archive/31_series/31.101/

[3GPP TS 31.102] "TS 31.102 Technical Specification Smart Cards; Characteristics of the Universal Subscriber Identity Module (USIM) application", 3rd Generation Partnership Project (3GPP), URL:http://www.3gpp.org/ftp/Specs/archive/31_series/31.102/

[3GPP TS 31.103] "TS 31.103 Technical Specification Group Core Network and Terminals; Characteristics of the IP Multimedia Services Identity Module (ISIM) application", 3rd Generation Partnership Project (3GPP), URL:http://www.3gpp.org/ftp/Specs/archive/31_series/31.103/

[3GPP TS 31.111] "TS 31.111 Technical Specification Group Core Network and Terminals; Universal Subscriber Identity Module (USIM), Application Toolkit (USAT)", 3rd Generation Partnership Project (3GPP), URL:http://www.3gpp.org/ftp/Specs/archive/31_series/31.111/

[3GPP TS 31.111] "TS 31.111 Technical Specification Group Core Network and Terminals; Universal Subscriber Identity Module (USIM), Application Toolkit (USAT)", 3rd Generation Partnership Project (3GPP), URL:http://www.3gpp.org/ftp/Specs/archive/31_series/31.111/

[3GPP TS 51.011] "TS 51.011 Technical Specification Group Terminals; Specification of the Subscriber Identity Module-Mobile Equipment (SIM - ME) interface", 3rd Generation Partnership Project (3GPP), URL:http://www.3gpp.org/ftp/Specs/archive/51_series/51.011/

[3GPP TS 51.014] "TS 51.014 Technical Specification Group Terminals; Specification of the SIM Application Toolkit for the Subscriber Identity Module - Mobile Equipment (SIM - ME) interface (Release 4)", 3rd Generation Partnership Project (3GPP), URL:http://www.3gpp.org/ftp/Specs/archive/51_series/51.014/

[3GPP2 C.S0023] "Removable User Identity Module for Spread Spectrum Systems", 3rd Generation Partnership Project 2 (3GPP2), Technical Specification 3GPP2 C.S0023, URL:http://www.3gpp2.org/

[3GPP2 C.S0035] "CDMA Card Application Toolkit (CCAT)", 3rd Generation Partnership Project 2 (3GPP2), Technical Specification 3GPP2 C.S0035, URL:http://www.3gpp2.org/

[3GPP2 C.S0065] "Cdma2000 Application on UICC for Spread Spectrum Systems", 3rd Generation Partnership Project 2 (3GPP2), Technical Specification 3GPP2 C.S0065, URL:http://www.3gpp2.org/

[3GPP2 C.S0068] "ME Personalization for cdma2000 Spread Spectrum Systems", 3rd Generation Partnership Project 2 (3GPP2), Technical Specification 3GPP2 C.S0068, URL:http://www.3gpp2.org/

[DMClientAPIFw v1.0] "Enabler Release for OMA Device Management Client API framework", OMA-ER-DMClientAPIfw-V1_0, Open Mobile Alliance™, URL:http://www.openmobilealliance.org/

[ETSI TR 102 216] "TR 102 216 Technical Report Smart Cards; Vocabulary for Smart Card Platform specifications", v3.0.0, European Telecommunications Standards Institute (ETSI), URL:http://www.etsi.org

[ETSI TS 102 221] "TS 102 221 Technical Specification, Smart Cards; UICC-Terminal interface; Physical and logical characteristics", European Telecommunications Standards Institute (ETSI), URL:http://www.etsi.org

[ETSI TS 102 223] "TS 102 223 Technical Specification, Smart Cards; Card Application Toolkit (CAT)", European Telecommunications Standards Institute (ETSI), URL:http://www.etsi.org

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

2.2 Informative References

[OMADICT] "Dictionary for OMA Specifications", Version 2.8, Open Mobile Alliance™, OMA-ORG-Dictionary-V2_8, URL:http://www.openmobilealliance.org/

3. Terminology and Conventions

3.1 Conventions

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

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

3.2 Definitions

Cloud Device	Device that needs to be connected and using online services to be fully functional.
Connection Manager	An entity or application that manages different network connections based on user profiles associated with these connections.
CSIM	A CDMA2000 Subscriber Identity Module is an application defined in [3GPP2 C.S0065] residing on the UICC to register services provided by 3GPP2 mobile networks with the appropriate security.
Device	A device is composed of one or several modems dealing with connectivity aspects.
ISIM	An IP Multimedia Services Identity Module is an application defined in [3GPP TS 31.103] residing in the memory of the UICC, providing IP service identification, authentication and ability to set up Multimedia IP Services.
M2M	Any other device with an embedded modem module using wireless network(s) to communicate with other devices or networks. It could be for example a module for an automotive system or an alarm system or even a consumer device such as a camera or a portable game device with embedded module.
Mobile Broadband Device	A datacard or USB modem or dongle that can be plugged in a laptop to assume data connectivity to cellular networks
NAA	Network Access Application as defined in [ETSI TR 102 216]. Examples of NAA on UICC: CSIM, ISIM, USIM.
Profile/User Profile/Connection Profile	The term Profile or User Profile or Connection Profile will be used to identify the information needed to establish a connection. There are two types of Connection Profiles: cellular profiles for connection to cellular and WLAN profiles for connection to WLAN.
Push Service	A service utilizing PUSH delivery mechanism that enables the mobile device to receive data traffic initiated by a dedicated server.
R-UIM	A Removable User Identity Module is a standalone module defined in [3GPP2 C.S0023] to register services provided by 3GPP2 mobile networks with the appropriate security.
SIM	A Subscriber Identity Module is a standalone module defined in [3GPP TS 51.011] to register services provided by 2G mobile networks with the appropriate security.
UICC	As defined in [OMA-DICT] and whose interface is specified in [3GPP TS 31.101].
UIM	A User Identity Module is a module defined in [3GPP2 C.S0023] to register services provided by 3GPP2 mobile networks with the appropriate security. The UIM can either be a removable UIM (R-UIM) or a non-removable UIM.
USIM	A Universal Subscriber Identity Module is an application defined in [3GPP TS 31.102] residing in the memory of the UICC to register services provided by 3GPP mobile networks with the appropriate security.
Wireless Router	A cellular network device that combines a router, switch and Wi-Fi access point (Wi-Fi base station) in one box. In the case of OpenCMAPI, the network to provide connectivity will be a cellular network. There could be two sorts of Wireless router: portable for nomadic usage or fixed for home usage in the case of Digital Dividend for example however in the document they will be considered as the same.

3.3 Abbreviations

3GPP	3rd Generation Partnership Project
3GPP2	3rd Generation Partnership Project 2
AKA	Authentication and Key Agreement
API	Application Programming Interface
APN	Access Point Name
CDMA	Code Division Multiple Access
CHAP	Challenge Handshake Authentication Protocol
CM	Connection Manager
CSIM	CDMA2000 Subscriber Identity Module
DM	Device Management
DNS	Domain Name System
EAP	Extensible Authentication Protocol
EDGE	Enhanced Data rates for GSM Evolution
ETSI	European Telecommunications Standards Institute
e-UTRAN	evolved Universal Terrestrial Radio Access Network
GAN	Generic Access Network
GERAN	GSM EDGE Radio Access Network
GNSS	Global Navigation Satellite System
GPRS	General Packet Radio Service
GPS	Global Positioning System
GSM	Global System for Mobile communications
HSPA	High Speed Packet Access
ISIM	IP Multimedia Services Identity Module
LTE	Long Term Evolution
MAC	Media Access Control
MMS	Multimedia Messaging Service
NAA	Network Access Application
NDIS	Network Driver Interface Specification
NMEA	National Marine Electronics Association
ODM	Original Device Manufacturer
OEM	Original Equipment Manufacturer
OMA	Open Mobile Alliance
OpenCMAPI	Open Connection Manager (CM) Application Programming Interface (API)
PAP	Password Authentication Protocol
PDN	Public Data Network
PIN	Personal Identification Number
PLMN	Public Land Mobile Network
PRL	Preferred Roaming List
PSK	PreShared Key

PUK	Pin Unlocking Key
QoS	Quality of Service
RAS	Remote Access Service
RAT	Radio Access Technologies
RFC	Request For Comments
RSSI	Received Signal Strength Indicator
R-UIM	Removable User Identity Module
SIM	Subscriber Identity Module
SMS	Short Message Service
SMS-C	Short Message Service Center
SSID	Service Set Identifier
UI	User Interface
UICC	Universal Integrated Circuit card
UIM	User Identity Module
UMA	Unlicensed Mobile Access
UMTS	Universal Mobile Telecommunications System
USIM	Universal Subscriber Identity Module
USSD	Unstructured Supplementary Service Data
UTRAN	Universal Terrestrial Radio Access Network
VPN	Virtual Private Network
WEP	Wired Equivalent Privacy
Wi-Fi	Wireless Fidelity
WiMAX	Worldwide Interoperability for Microwave Access
WISPr	Wireless Internet Service Provider roaming
WLAN	Wireless Local Area Network
WPA2	Wi-Fi Protected Access Version 2
WPS	Wireless Protected Setup
WWAN	Wireless Wide Area Network

4. Introduction (Informative)

With the introduction of faster and faster mobile networks and the introduction of new devices such as Mobile Broadband devices, smartphones, tablets benefiting of the capabilities of these networks and needing more and more bandwidth and ubiquity to meet the customers' expectations, the need for proper management of connectivity becomes critical.

In addition, the multiplicity of possible mobile networks available (2G, 3G, 4G) as well as the congestion of some networks leading a lot of operators to use Wi-Fi Hotspots to offload the traffic is putting even more emphasize on the connection management aspects.

Furthermore, new or future types of devices such as cloud devices and new types of applications will be relying even more on the networks and the need for always on connection or for more information related to the connection established or the ones available.

A Connection Manager application, using the OpenCMAPI, is the main point of contact to manage the connections as the name indicates but also to provide information status on the connection including networks and device or modem component used and other services relevant/associated to.

Up to now, there is no existing standard or de facto standard for Connection Managers. Operators and OEM/ODM have to develop and use different and dedicated solutions, thus increasing the effort and time to market.

For Mobile Broadband devices, this situation is critical and leading to a strong effort for service providers to develop Connection Manager applications as there are already several networks to support and any new mobile broadband device such as USB modem requires to redevelop existing Connection Managers to be implemented and supported by these applications.

For smartphones or tablets, the importance of management of Wi-Fi offloads for example and/or the need to expose information status on the connection to applications is requiring a solution through the Connection Manager application.

Furthermore, new fast growing businesses such as Connected Devices & M2M are facing the same hurtles and will need as well a solution to reduce the impacts and efforts to deal with the connection management aspects.

The Open Connection Management (CM) API – Enabler addresses these aspects by providing a specification relevant for the whole industry.

A Connection Manager is basically composed of 2 parts:

1. The hardware & connectivity engine part to manage the device with the necessary functions relevant for the user/customer of the Connection Manager or for the application requiring information status on the connection
2. The user experience presented to the customer and composed mainly of the UI, the profiles and the services offered to the user based on actions and answers from the hardware engine part.

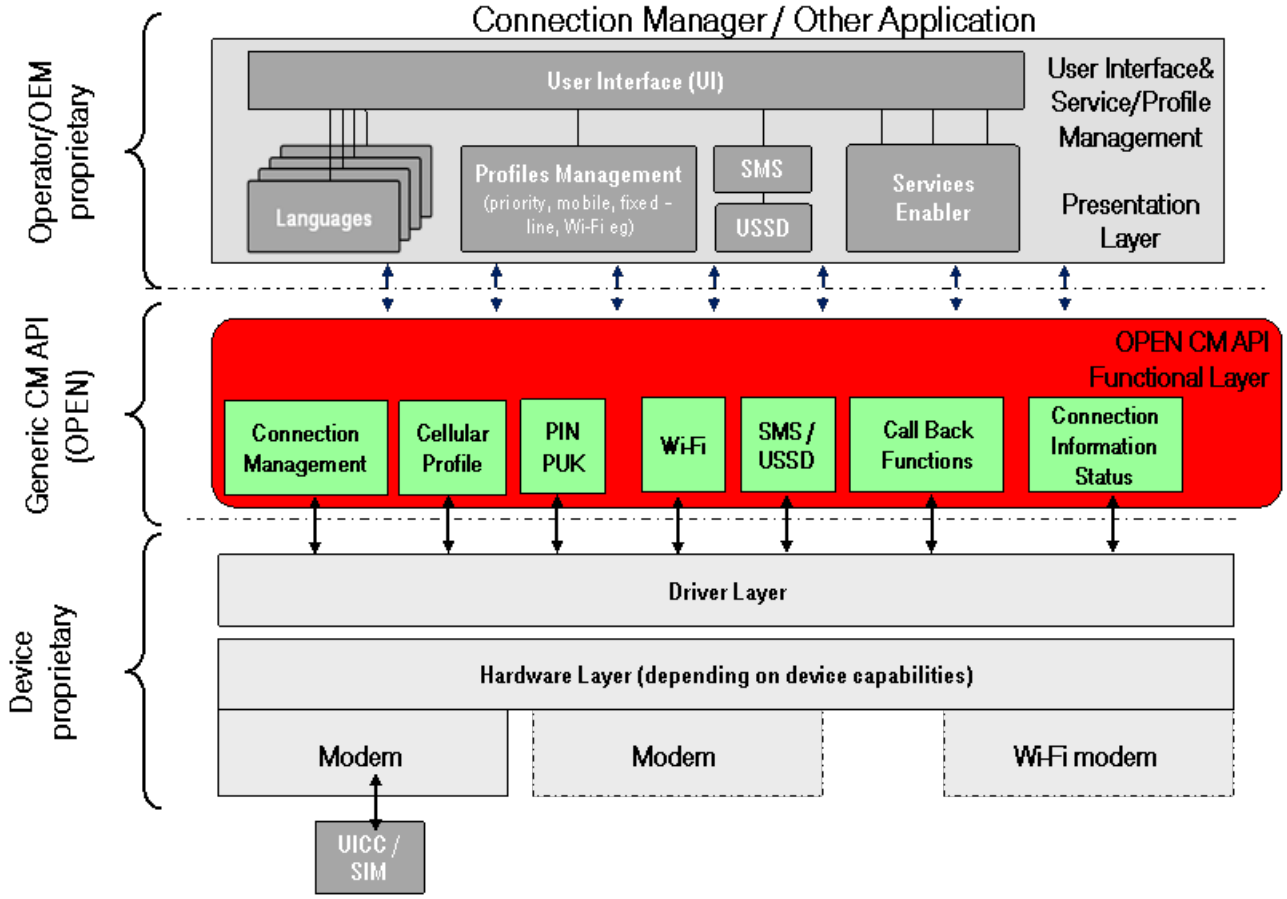

Figure 1: High Level diagram for Open CM API Enabler

The purpose of the proposed work is to define an Open Connection Manager API to assume the hardware and connectivity engine part in order to facilitate development of the top/presentations layer representing the user experience as well as application that can rely on information status about the connection and to avoid issues for integration of any new device within already defined user experience.

Operators and OEM/ODM or any application developers could develop by using such API the user experience in line with their business perspectives (UI, differentiation, services...). Furthermore, without any additional effort, it will be possible to integrate or to exchange any device/hardware to support this user experience.

4.1 Version 1.0

This document defines the requirements for OpenCMAPI Enabler Version 1.0 extending the functionalities of service APIs to 3rd party applications:

- Network Types
- Cellular Network Management
- Device Service Handling
- PIN/PUK Management
- Connection Management
- WLAN handling & WLAN authentication

- CallBack
- Status information handling
- Statistics Management
- SMS service handling
- USSD service handling
- GNSS service handling
- Power Management
- Tethering handling
- UICC interface
- PUSH Services

These functions are described in detailed functional requirements in section “Requirements”.

5. OpenCMAPI release description (Informative)

The focus of the OpenCMAPI enabler is the standardization of new functional APIs essential for applications to develop connection manager user interface and to extend applications and services with information related to the connection.

In order to allow for advanced service creation based on multiple services/enablers, interface functionalities for SMS, USSD as well as GNSS are included.

The intention is to be supported by different types of devices such as Mobile Broadband devices, Wireless routers, M2M, Smartphones, Tablets, and Cloud Devices requiring access to mobile internet.

The OpenCMAPI v1.0 functionalities are designed independently of a specific framework architecture or application domain.

5.1 End-to-end Service Description

The API functionalities as proposed in the OpenCMAPI v1.0 aims at creating a new set of OMA service interfaces to enhance value of the connectivity and access to multiple networks by allowing the industry to easily develop services, differentiation and their own User experience on top of the connection management API.

This enabler will allow service providers to develop easily connection manager application and dedicated user interface to work across all their devices in their portfolio without additional effort to integrate or support a new device. Moreover, it will help to improve new types of applications relying almost solely on having a good always on connection such as virtual reality applications to be always informed about the status of the connection established or the ones available.

From device manufacturer point of view, OpenCMAPI will allow reducing effort and costs to be compliant with the requirements of different service providers and OEM/ODM (laptop manufacturers) and will provide immediate support of the services and user experience developed by these Service providers.

From the OEM/ODM such as laptop’s manufacturers’ point of view, OpenCMAPI will allow to develop connection managers applications that can easily interwork with any modems embedded and will decrease the complexity for customization and support for multiple Business models with service providers.

Further more, the OpenCMAPI will allow Corporate or Enterprise customers to develop their own connection managers, their own UI and services easily across numerous devices and without having to redevelop any time they have a new device to be supported.

6. Requirements (Normative)

6.1 High-Level Functional Requirements

This section identifies the high level functional requirements for the OpenCMAPI Enabler. The detail requirements are further identified in the following sections according to detailed functions identified. The relationship between high level requirements and detail requirements are described in the 'Informational Note' of the individual requirements in this section when necessary.

Label	Description	Release
CMAPI-HLF-001	The Open CM API enabler SHALL be able to support different types of devices such as Mobile Broadband devices, Laptops, Wireless routers, M2M, Smartphones, Tablets, Cloud Devices requiring access to mobile internet and other technologies. Some of the requirements/functionalities described below and in the whole RD are more relevant for certain types of devices rather than others. The Appendix B identifies the relevance of these requirements and therefore if the requirements are mandatory or optional according to the type of devices.	1.0
CMAPI-HLF-002	The OpenCMAPI Enabler SHALL support the management of different types of network (e.g. GERAN, UTRAN, CDMA2000, E-UTRAN, WLAN). **Informational Note:** The required functionality of this requirement is as specified in requirements listed in section 6.2.	1.0
CMAPI-HLF-003	The OpenCMAPI Enabler SHALL support Basic Connectivity (connect/disconnect to Networks) and basic detection/selection functionalities. **Informational Note:** The required functionality of this requirement is as specified in requirements CMAPI-CON-001 to CMAPI-CON-019 listed in section 6.6.	1.0
CMAPI-HLF-004	The OpenCMAPI Enabler SHALL support the retrieval of Network information (e.g. Radio Interface, Band, Attach, Registration, PLMN type, Roaming State, Signal strength…). **Informational Note:** The required functionality of this requirement is as specified in requirements CMAPI-NETM-001 to CMAPI-NETM-006 listed in section 6.3.	1.0
CMAPI-HLF-005	The OpenCMAPI Enabler SHALL support different network selection modes (Automatic, Manual…). **Informational Note:** The required functionality of this requirement is as specified in requirements CMAPI-SEL-001 to CMAPI-SEL-007 listed in section 6.6.	1.0
CMAPI-HLF-006	The OpenCMAPI Enabler SHALL support access to Device information (e.g. IMSI, IMEI, operator name, FW version…). **Informational Note:** The required functionality of this requirement is as specified in requirements listed in section 6.4.	1.0
CMAPI-HLF-007	The OpenCMAPI Enabler SHALL allow the management of the Connection Profile/user settings including mobile network parameters (e.g. APN, DNS, IP…/ parameters could be defined for each profile/user) by a connection manager application. **Informational Note:** The required functionality of this requirement is as specified in requirements CMAPI-PRO-001 to CMAPI-PRO-005 listed in section 6.6.	1.0
CMAPI-HLF-008	The OpenCMAPI Enabler SHALL be able to support multi APNs simultaneously. **Informational Note:** The required functionality of this requirement is as specified in requirements CMAPI-PRO-006 to CMAPI-PRO-009 listed in section 6.6.	1.0
CMAPI-HLF-009	The OpenCMAPI Enabler SHALL retrieve statistics information (e.g number of kB sent or received, upload/download speed…). **Informational Note:** The required functionality of this requirement is as specified in requirement listed in section 6.9.	1.0

CMAPI-HLF-010	The OpenCMAPI Enabler SHALL be able to provide status information of the cellular interface. **Informational Note:** The required functionality of this requirement is as specified in requirements listed in section 6.8.	1.0
CMAPI-HLF-011	The OpenCMAPI Enabler SHALL be able to support some power management aspects (e.g. hibernation, switching off radio for Flight Mode). **Informational Note:** The required functionality of this requirement is as specified in requirements listed in section 6.12.	1.0
CMAPI-HLF-012	The OpenCMAPI Enabler SHALL support the management of SMS functions (e.g. send, receive, delete…). **Informational Note:** The required functionality of this requirement is as specified in requirements listed in section 6.10.1	1.0
CMAPI-HLF-013	The OpenCMAPI Enabler SHALL support the management of USSD features. **Informational Note:** The required functionality of this requirement is as specified in requirements listed in section 6.10.2	1.0
CMAPI-HLF-014	The OpenCMAPI Enabler SHALL allow multiple applications to use in parallel the Open CM API.	1.0
CMAPI-HLF-015	The OpenCMAPI Enabler SHALL support access to the OMA Device Management (DM) functionality on the device (it is recommended to use the [DMClientAPIFw v1.0] functionality if it is available on the device). **Informational Note:** OpenCMAPI Release 1.0 will support access to the minimum OMA DM functionality needed to configure certain cdma2000 network information, as specified in requirement CMAPI-C2K-005 in section 6.3.3. Requirements for access to additional OMA DM functionality, including related enablers such as FUMO, SCOMO and DiagMon, will be added in future OpenCMAPI releases.	1.0
CMAPI-HLF-016	The OpenCMAPI Enabler SHALL allow a Connection Manager application to interact with different mechanism (e.g. OMA DM, FUMO) to perform a device firmware upgrade	1.0
CMAPI-HLF-017	The OpenCMAPI Enabler SHALL be able to allow an update process (e.g. OMA DM, SCOMO) to update to a newer version of an implementation.	1.0
CMAPI-HLF-018	The OpenCMAPI Enabler SHALL be able to support tethering functionalities subject to device capability. **Informational Note:** The required functionality of this requirement is as specified in requirements listed in section 6.15.	1.0
CMAPI-HLF-019	The OpenCMAPI Enabler SHALL be able to support GNSS features subject to device capability. **Informational Note:** The required functionality of this requirement is as specified in requirements listed in section 6.11.	1.0
CMAPI-HLF-020	The OpenCMAPI Enabler SHALL be able to support in parallel WLAN data connection and CS domain (e.g. SMS) via GERAN/UTRAN/CDMA2000/EVDO network, subject to device capability. The OpenCMAPI Enabler SHALL not interfere with the voice call and video call that are managed by another entity.	1.0
CMAPI-HLF-021	The OpenCMAPI Enabler SHALL be able to support in parallel WLAN data connection and PS domain service (e.g. MMS) via GERAN/UTRAN/E-UTRAN/CDMA2000/EVDO network, subject to device capability.	1.0
CMAPI-HLF-022	The OpenCMAPI Enabler SHALL be able to support the preferred RAT settings for different service types, subject to operator policy (e.g. WLAN is configured as the user preferred connection for data service).	1.0
CMAPI-HLF-023	The OpenCMAPI Enabler SHALL support RAT type management function. **Informational Note:** The required functionality of this requirement is as specified in requirements listed in section 6.3.2	1.0

CMAPI-HLF-024	The OpenCMAPI Enabler SHALL be able to support callbacks to enable CM applications to be notified when specified events of interest occur (e.g. a data session becomes disconnected, signal strength falls below a threshold). **Informational Note:** The required functionality of this requirement is as specified in requirements in section 6.13.	1.0
CMAPI-HLF-025	The OpenCMAPI Enabler SHALL be able to support error handling mechanism including returning error codes to Connection Manager applications (e.g., the invalid parameters are input when call from Connection Manager applications).	1.0
CMAPI-HLF-026	The OpenCMAPI Enabler MAY provide support for ensuring the continuity of an ongoing service when the terminal device switches its connection among different networks (e.g., GERAN, UTRAN, CDMA2000, E-UTRAN, WLAN) depending on device capability.	1.0
CMAPI-HLF-027	The OpenCMAPI enabler SHALL support the handling of the WLAN component. **Informational Note:** The required functionality of this requirement is as specified in requirements listed in section 6.7.	1.0
CMAPI-HLF-028	The OpenCMAPI enabler SHALL support interfaces related to UICC. **Informational Note:** The required functionality of this requirement is as specified in requirements listed in section 6.14.	1.0
CMAPI-HLF-029	The OpenCMAPI enabler SHALL support data PUSH service option. **Informational Note:** The required functionality of this requirement is as specified in requirements listed in section 6.16.	1.0
CMAPI-HLF-030	The OpenCMAPI Enabler SHALL be able to support Phone Book /Contacts management	Future Release
CMAPI-HLF-031	The OpenCMAPI Enabler SHALL be able to support VPN hooks	Future Release

Table 1: High-Level Functional Requirements

6.1.1 Security

This section identifies the high-level security requirements for the OpenCMAPI Enabler.

Label	Description	Release
CMAPI-SEC-001	The OpenCMAPI Enabler SHALL support PINs/PUKs management	1.0
CMAPI-SEC-002	The OpenCMAPI Enabler SHALL protect against potential security threats	1.0
CMAPI-SEC-003	The OpenCMAPI enabler SHALL NOT support functions accessing to the UICC equivalent to following AT commands (Connected SIM) +CSIM.	1.0
CMAPI-SEC-004	The OpenCMAPI Enabler SHALL support WPS with both PIN & Push-Button methods for 802.11b/g/n.	1.0

Table 2: High-Level Functional Requirements – Security Items

6.1.1.1 Authentication

This section identifies the high-level authentication needs for the OpenCMAPI Enabler.

Label	Description	Release
CMAPI-AUT-001	The OpenCMAPI Enabler SHALL offer selectable authentication mechanisms for Cellular at least: ▪ PAP ▪ CHAP ▪ Automatic	1.0
CMAPI-AUT-002	The OpenCMAPI Enabler SHALL support EAP SIM authentication in conjunction with WPA2-E key management using the SIM/RUIM or UICC application credentials.	1.0
CMAPI-AUT-003	The OpenCMAPI Enabler SHALL support EAP AKA authentication in conjunction with WPA2-E key management using the SIM/RUIM or UICC application credentials.	1.0
CMAPI-AUT-004	The OpenCMAPI Enabler SHALL support WPA-PSK and WPA2-PSK authentication of WLAN network.	1.0
CMAPI-AUT-005	The OpenCMAPI Enabler SHALL support EAP AKA' authentication (IETF RFC 5448) in conjunction with WPA2-E key management using the SIM/RUIM or UICC application credentials.	1.0

Table 3: High-Level Functional Requirements – Authentication Items

6.1.1.2 Authorization

This section identifies the high-level authorisation needs for the OpenCMAPI Enabler

Label	Description	Release
CMAPI-AUTH-001	The OpenCMAPI enabler SHALL support the authorization of the mobile users and/or an application when authenticated using authentication mechanisms, including EAP SIM, EAP AKA, WPA-PSK and WPA2-PSK.	1.0

Table 4: High-Level Functional Requirements – Authorization Items

6.1.2 Administration and Configuration

This section identifies the high-level Administration and Configuration High Level requirements for the OpenCMAPI Enabler.

Label	Description	Release
CMAPI-ADM-001	The OpenCMAPI Enabler SHALL be able to work without administrator rights. Administrator rights MAY be required for installation or other critical operations.	1.0

Table 5: High-Level Functional Requirements – Administration and Configuration Items

6.2 Network Types Functional Requirements

This section identifies the requirements for the types of networks supported by the OpenCMAPI Enabler subject to device capabilities.

Label	Description	Release
CMAPI-NET-001	The OpenCMAPI Enabler SHALL support GERAN Network type (GPRS, EDGE)	1.0
CMAPI-NET-002	The OpenCMAPI Enabler SHALL support UTRAN Network type (WCDMA FDD, TD-SCDMA)	1.0
CMAPI-NET-003	The OpenCMAPI Enabler SHALL support UTRAN Network type (WCDMA TDD, HCR FDD)	Future Release
CMAPI-NET-004	The OpenCMAPI Enabler SHALL support CDMA 2000 Network type (1XRTT, HRPD)	1.0
CMAPI-NET-005	The OpenCMAPI Eenabler SHALL support E-UTRAN Network Type (LTE FDD , LTE TDD)	1.0
CMAPI-NET-006	The OpenCMAPI enabler SHALL support WLAN (Wi-Fi) Network type	1.0
CMAPI-NET-007	The OpenCMAPI enabler SHALL be able to support Wimax Networks	Future Release
CMAPI-NET-008	The OpenCMAPI enabler SHALL be able to support Hotspot 2.0 **Informational Note:** Hotspot 2.0 is the name of a Wi-Fi Alliance certification program, using a common set of standards (IEEE 802.1X and IEEE 802.11u), for Wi-Fi networks.	Future Release

Table 6: Network Type Supported Functional Requirements

6.3 Cellular Network Management Functional Requirements

6.3.1 Network Management Functional Requirements

This section identifies the requirements for interfaces related to the Management of the network before connection.

Label	Description	Release
CMAPI-NETM-001	The OpenCMAPI Enabler SHALL support the retrieval of the RF info (radio access technology, band class, data rate supported and channel).	1.0
CMAPI-NETM-002	The OpenCMAPI Enabler SHALL support the retrieval of the status of the network access: ▪ Registered ▪ Attached ▪ Out of Service As defined in [3GPP TR 21.905] **Informational note**: In LTE there is no CS registration, only PS registration is possible and PDP activation is automatic with the attach.	1.0
CMAPI-NETM-003	The OpenCMAPI Enabler SHALL support the retrieval of the Signal strength information as well as the Signal quality	1.0
CMAPI-NETM-004	The OpenCMAPI Enabler SHALL support the retrieval of the Home network information	1.0
CMAPI-NETM-005	The OpenCMAPI Enabler SHALL support the retrieval of the Serving network identity and capabilities	1.0
CMAPI-NETM-006	The OpenCMAPI Enabler SHALL support the retrieval of the different RAT Types.	1.0

Table 7: Network Management Functional Requirements

6.3.2 RAT Type Functional Requirements

This section identifies the requirements related to RAT type supported by the Open CM API Enabler.

Label	Description	Release
CMAPI-RAT-001	The OpenCMAPI Enabler SHALL be able to interact with different RAT Types. (e.g. UTRAN, GERAN, WLAN, E-UTRAN)	1.0

Table 8: RAT Type Functional Requirements

6.3.3 CDMA Specific Functional Requirements

This section identifies the requirements dedicated to CDMA 2000 supported by the OpenCMAPI Enabler subject to device capabilities.

Label	Description	Release
CMAPI-C2K-001	The OpenCMAPI Enabler SHALL be able to support the specification of Mobile IP configuration (e.g. NAI, home IP address, primary and secondary Home Agent (HA) IP addresses, etc) for cdma2000 networks.	1.0
CMAPI-C2K-002	The OpenCMAPI Enabler SHALL be able to support the specification of Mobile IP parameters (e.g. re-registration periods and registration retry configuration) for cdma2000 networks.	1.0
CMAPI-C2K-003	The OpenCMAPI Enabler SHALL be able to support the setting and retrieval of cdma2000 network parameters (e.g. Access Overload Class (ACCOLC), roaming preference).	1.0
CMAPI-C2K-004	The OpenCMAPI Enabler SHALL be able to support cdma2000 automatic and manual service activation.	1.0
CMAPI-C2K-005	The OpenCMAPI Enabler SHALL be able to support the trigger of the available bootstrapped OMA Device Management (DM) client for the configuration of cdma2000 device provisioning data and 3GPP2 Preferred Roaming List (PRL) information.	1.0

Table 9: CDMA specific Functional Requirements

6.3.4 Mobile IP Specific Functional Requirements

The following requirements are applicable for cdma2000 devices supporting Mobile IP.

Label	Description	Release
CMAPI-MIP-001	The OpenCMAPI Enabler SHALL support the mechanism to enable and disable the device's Mobile IP functionality.	1.0
CMAPI-MIP-002	The OpenCMAPI Enabler SHALL be able to get and set the Mobile IP configuration, which includes ▪ Home IPv4/IPv6 address ▪ Primary and secondary Home Agent IP addresses ▪ Reverse tunneling status (enabled/disabled) ▪ NAI ▪ Security parameters and key strings	1.0
CMAPI-MIP-003	The OpenCMAPI Enabler SHALL be able to get and set the following Mobile IP parameters: ▪ Mobile IP mode (on, preferred or off) ▪ Registration retry attempt limit and interval ▪ Re-registration period ▪ Authentication method	1.0

Table 10: Mobile IP specific Functional Requirements

6.4 Device related Functional Requirements

6.4.1 Device Service Functional Requirements

This section identifies the requirements for interfaces related to the Device elements.

Label	Description	Release
CMAPI-DEV-001	The OpenCMAPI Enabler SHALL be able to provide the Name of the Manufacturer of the device	1.0
CMAPI-DEV-002	The OpenCMAPI Enabler SHALL be able to provide the product model ID of the device	1.0
CMAPI-DEV-003	The OpenCMAPI Enabler SHALL be able to provide the commercial name of the Device.	1.0
CMAPI-DEV-004	The OpenCMAPI Enabler SHALL be able to provide the Device capabilities (max transmit and receive speeds, etc)	1.0
CMAPI-DEV-005	The OpenCMAPI Enabler SHALL be able to provide the Hardware Version	1.0
CMAPI-DEV-006	The OpenCMAPI Enabler SHALL be able to provide the product type of the device one or several of the following: ▪ WCDMA ▪ CDMA ▪ LTE ▪ SCDMA ▪ WLAN ▪ Unknown	1.0
CMAPI-DEV-007	The OpenCMAPI Enabler SHALL be able to provide the IMSI	1.0
CMAPI-DEV-008	The OpenCMAPI Enabler SHALL be able to provide the MDN or MSISDN (if any present)	1.0
CMAPI-DEV-009	The OpenCMAPI Enabler SHALL be able to provide the serial numbers (IMEI, ESN, MEID)	1.0
CMAPI-DEV-010	The OpenCMAPI Enabler SHALL be able to provide the device Status: ▪ Available ▪ Unavailable ▪ Plugged ▪ Unplugged ▪ Unknown	1.0
CMAPI-DEV-011	The OpenCMAPI Enabler SHALL be able to provide the SIM/R-UIM/UICC State: ▪ Inserted ▪ Not inserted ▪ Available ▪ Invalid ▪ Changed SIM/R-UIM/UICC ▪ Unavailable ▪ Unknown	1.0
CMAPI-DEV-012	The OpenCMAPI Enabler SHALL be able to provide the Device radio switch status: ▪ On ▪ Off	1.0
CMAPI-DEV-013	The OpenCMAPI Enabler SHALL be able to provide the version number of the OpenCMAPI used	1.0

CMAPI-DEV-014	The OpenCMAPI Enabler SHALL be able to provide the Firmware version of the device	1.0
CMAPI-DEV-015	The OpenCMAPI Enabler SHALL be able to provide the PRL version (CDMA 2000)	1.0
CMAPI-DEV-016	The OpenCMAPI Enabler SHALL be able to provide the Network Time	1.0

Table 11: Device Service Functional Requirements

6.4.2 WWAN WLAN Module Functional Requirements

This section identifies the requirements related to the management of WWAN/WLAN module supported by the OpenCMAPI Enabler.

Label	Description	Release
CMAPI-MOD-001	The OpenCMAPI Enabler SHALL provide the following APIs to enable the CM application to manage the WWAN/WLAN modules connected to the host device (laptop, tablet, etc): ▪ List the available WWAN/WLAN modules ▪ Connect to a specific WWAN/WLAN module ▪ Disconnect from a specific WWAN/WLAN module	1.0
CMAPI-MOD-002	The OpenCMAPI Enabler SHALL provide an API that enables the execution of the upgrade of a WWAN/WLAN module's firmware.	1.0
CMAPI-MOD-003	The OpenCMAPI Enabler SHALL provide APIs that enables the CM application to obtain information about the WWAN/WLAN module's currently installed firmware (e.g. firmware version, applicable region).	1.0

Table 12: WWAN WLAN Functional Requirements

6.5 PINs / PUKs Functional Requirements

This section identifies the requirements related to the management of the PINs/PUKs functionality supported by the OpenCMAPI Enabler.

Label	Description	Release
CMAPI-PIN-001	The OpenCMAPI Enabler SHALL support Personal Identification Number (PINs) and Personal Unblocking Key (PUKs) management feature	1.0
CMAPI-PIN-002	The OpenCMAPI Enabler SHALL provide APIs for: ▪ Enabling and disabling PIN protection ▪ And the corresponding answers from the SIM/R-UIM/NAA on UICC	1.0
CMAPI-PIN-003	The OpenCMAPI Enabler SHALL provide APIs for: ▪ Verifying a PIN ▪ And the corresponding answers from the SIM/R-UIM/NAA on UICC	1.0
CMAPI-PIN-004	The OpenCMAPI Enabler SHALL provide APIs for: ▪ Unblocking a PIN ▪ And the corresponding answers from the SIM/R-UIM/NAA on UICC	1.0
CMAPI-PIN-005	The OpenCMAPI Enabler SHALL provide APIs for: ▪ Changing a PIN ▪ And the corresponding answers from the SIM/R-UIM/NAA on UICC	1.0
CMAPI-PIN-006	The OpenCMAPI Enabler SHALL provide APIs for: ▪ Checking current PIN status (ready, not initialised, enabled, disabled, blocked, permanently blocked , unblocked) ▪ And the corresponding answers from the SIM/R-UIM/NAA on UICC	1.0
CMAPI-PIN-007	The OpenCMAPI Enabler SHALL verify that the CM application provides only digits 0-9 for the PINs/PUKs entry. Any other values shall be prohibited.	1.0
CMAPI-PIN-008	The OpenCMAPI Enabler SHALL verify that the PIN provided by the CM application is any four to eight digits number before sending it to the SIM/R-UIM/NAA on UICC.	1.0

Table 13: PINs / PUKs Functional Requirements

6.6 Connection Management Functional Requirements

6.6.1 Profile Management for Cellular Network Functional Requirements

This section identifies the requirements for the profile Management aspects supported by the OpenCMAPI Enabler.

Label	Description	Release
CMAPI-PRO-001	The OpenCMAPI Enabler SHALL provide functions to manage the storing of several cellular profiles.	1.0
CMAPI-PRO-002	The OpenCMAPI Enabler SHALL be able to add/remove a profile in the cellular profile management list.	1.0
CMAPI-PRO-003	The OpenCMAPI Enabler SHALL be able to manage cellular profiles with the following fields: ▪ APN ▪ User/password (can be empty) ▪ DNS ▪ IP	1.0
CMAPI-PRO-004	The OpenCMAPI Enabler SHALL be able to support IPv4 / IPv6 addresses	1.0
CMAPI-PRO-005	The OpenCMAPI Enabler SHALL be able to configure the APNs and related parameters PDN type (IPv4, IPv6, IPv4v6) per APN.	1.0
CMAPI-PRO-006	The OpenCMAPI Enabler SHALL be able to handle multiple cellular profiles (e.g. Internet profile, MMS profile) working in parallel.	1.0
CMAPI-PRO-007	The OpenCMAPI enabler SHALL support the option to have a second APN bearing no data traffic in parallel to the default APN. The second APN is only configured but never activated by the OpenCMAPI and SHALL not be provided to any application. **Informational Note:** This is valid only for non IMS situation. This second APN could be used for example to simulate in LTE the equivalent of Attachment in 3G as in LTE, there is no similar behaviour - always connected.	1.0
CMAPI-PRO-008	The Open CM API enabler SHALL be able to support other/different Multiple APN options in the future with the introduction of IMS.	Future Release
CMAPI-PRO-009	The OpenCMAPI Enabler SHALL enable use of different DNS servers on different APNs.	1.0
CMAPI-PRO-010	The OpenCMAPI Enabler SHALL be able to configure the value of the timeout for inactive PDN connections.	1.0

Table 14: Profile Management for Cellular Network Functional Requirements

6.6.2 Network Selection Functional Requirements

This section identifies the requirements for interfaces related to the Network Selection supported by the OpenCMAPI Enabler.

Label	Description	Release
CMAPI-SEL-001	For devices with SIM/R-UIM/NAA on UICC the OpenCMAPI Enabler SHALL be configured to use the preferred PLMN with access technology present in the SIM/R-UIM/NAA on UICC, in accordance with 3GPP/3GPP2 specifications.[3GPP TS 22.011]	1.0
CMAPI-SEL-002	The OpenCMAPI Enabler SHALL provide the option to configure two modes: ▪ Automatic Network Selection (default mode) ▪ Manual Network Selection	1.0
CMAPI-SEL-003	In automatic mode, the network SHALL be selected by the OpenCMAPI Enabler according to the preferred network list of the SIM/R-UIM/NAA on UICC or non-removable UIM.	1.0
CMAPI-SEL-004	If automatic network selection is activated, the OpenCMAPI enabler SHALL always use the automatic selection irrespective of any manually chosen network.	1.0
CMAPI-SEL-005	When the manual network option is selected, the OpenCMAPI Enabler SHALL be able to provide the list of available networks organised as follows: ▪ Networks defined in the list of preferred PLMN ▪ Networks that are available but not defined in the list of preferred PLMN	1.0
CMAPI-SEL-006	When a network is selected in a list of available networks and there is no corresponding cellular profile, the OpenCMAPI Enabler SHALL prevent to connect to this network.	1.0
CMAPI-SEL-007	If manual network selection is activated and a connection is selected, the OpenCMAPI Enabler SHALL always use the selection until another networks is selected or automatic selection is activated	1.0

Table 15: Network Selection Functional Requirements

6.6.3 Network Connectivity Functional Requirements

This section identifies the requirements related to Network connectivity supported by the OpenCMAPI Enabler.

Label	Description	Release
CMAPI-CON-001	The OpenCMAPI Enabler SHALL be able to support RAS mode	1.0
CMAPI-CON-002	The OpenCMAPI Enabler SHALL be able to support Network Interface Card (NIC) APIs including NDIS and others.	1.0
CMAPI-CON-003	The OpenCMAPI Enabler SHALL be able to request to connect to and disconnect from the network.	1.0
CMAPI-CON-004	The OpenCMAPI Enabler SHALL be able to manage the different states of connection: ▪ Connected ▪ Disconnected (it may be possible to distinguish between passive and active disconnection) ▪ Connecting ▪ Disconnecting ▪ Connection failed ▪ Disconnection failed ▪ Already connected from the bearer wanted by new connection ▪ Already connected from another bearer ▪ Connection cancelled ▪ Scanning ▪ Unknown state	1.0
CMAPI-CON-005	The OpenCMAPI Enabler SHALL be able to cancel an in progress connect or disconnect request, both in automatic and manual network connection.	1.0
CMAPI-CON-006	The OpenCMAPI Enabler SHALL include an Auto Connect Feature with the following options: ▪ Enabled in all cases ▪ Enabled only in non-roaming cases ▪ Disabled (default)	1.0
CMAPI-CON-007	If PIN protection is enabled, the OpenCMAPI enabler SHALL require the Connection Manager application to provide a valid PIN when enabling the auto connect function.	1.0
CMAPI-CON-008	The OpenCMAPI Enabler SHALL be able to enable and disable the On Demand feature for a particular application, if this feature is available on the device. **Informational Note:** The “Connect On demand” Feature is the capability to connect on demand of an application. For example, if the “Connect on Demand” feature is enabled and an Internet Browser is starting i.e. requesting an internet connection, the connection will be established.	1.0
CMAPI-CON-009	The OpenCMAPI Enabler SHALL provide an auto reconnect functionality for all supported bearers when a connection break down happened (minimum and maximum re-try parameters are configurable)	1.0
CMAPI-CON-010	The OpenCMAPI Enabler SHALL be able to support the retrieval of the following information about the device’s data connection for a given profile: ▪ Data connection state (e.g. connected, disconnected) ▪ Network type (e.g. GERAN, UTRAN. WLAN) ▪ IP address ▪ Data rate ▪ Packet statistics (e.g. numbers of packets and bytes sent and received) ▪ Data connection duration	1.0

CMAPI-CON-011	The OpenCMAPI Enabler SHALL provide APIs that enable a CM application to manually establish a data connection with the following parameters: ▪ DNS addresses ▪ APN name ▪ Preferred IP address (if any) ▪ Authentication information (e.g. user name and password)	1.0
CMAPI-CON-012	The OpenCMAPI Enabler SHALL provide an API that enables the setting of a default connection profile to be used when automatically establishing a data connection.	1.0
CMAPI-CON-013	The OpenCMAPI Enabler SHALL provide APIs that enable the scanning for available networks and their associated Radio Access Technologies (RATs).	1.0
CMAPI-CON-014	The OpenCMAPI Enabler SHALL provide APIs that enable the device to register with and attach to an available network.	1.0
CMAPI-CON-015	The OpenCMAPI Enabler SHALL be able to support the retrieval and setting of the device's network registration preferences (i.e. which network to register with when multiple networks are available).	1.0
CMAPI-CON-016	By default, the OpenCMAPI Enabler SHALL connect automatically to the best available mobile network technology accordingly to the priority operator list preconfigured in priority order: ▪ in the SIM/R-UIM/NAA on UICC, if any, inserted in the device ▪ In the device, if not available in the SIM/R-UIM/NAA on UICC **Informational Note**: In general, this priority bearer list is organized by the maximum data rate associated with the different technologies (Higher data rate technology such as 4G is more important than lower such as 3G, 2G)	1.0
CMAPI-CON-017	The OpenCMAPI Enabler SHALL allow to restrict the permitted mobile specific bearer (2G, 3G, 4G) to be used for the connection. If a mobile specific bearer has been selected, the device will connect only to this mobile specific bearer and the restriction SHALL remain active until a new selection is done. A new selection can be automatic (e.g. return to default mode of device), or a specified mobile specific bearer (2G, 3G, 4G).	1.0
CMAPI-CON-018	The OpenCMAPI Enabler SHALL provide the "current" state of mobile specific bearer network selection (automatically or manually, and in this case which technology is used).	1.0
CMAPI-CON-019	The OpenCMAPI SHALL provide support for devices that include multiple radio technologies (e.g. dual-mode cdma2000 HRPD and LTE devices), including handoffs between different radio technologies according to the 3GPP2 and 3GPP specifications.	1.0

Table 16: Network Connectivity Functional Requirements

6.7 WLAN Functional Requirements

This section identifies the requirements for interfaces related to the management of WLAN.

Label	Description	Release
CMAPI-WLAN-001	The OpenCMAPI Enabler SHALL provide a function call to query whether the device is supporting WLAN functionality or not.	1.0
CMAPI-WLAN-002	The OpenCMAPI Enabler SHALL be able to enable/disable WLAN radio on the device.	1.0
CMAPI-WLAN-003	The OpenCMAPI Enabler SHALL provide a function call to query if WLAN functionality is enabled or disabled.	1.0
CMAPI-WLAN-004	The OpenCMAPI Enabler SHALL be configured to use the operator defined list of preferred SSID preconfigured in the device and/or the WSID (WLAN Specific Identifier) list in accordance with [3GPP TS 24.234] if present in the SIM/RUIM/NAA on UICC	1.0
CMAPI-WLAN-005	The OpenCMAPI Enabler SHALL be able to manage WLAN profile with the following fields: ▪ SSID ▪ Secured Network or OPEN Network (Open is referring to a non secured network) ▪ Security or authentication mechanism used ▪ Associated security Key if relevant	1.0
CMAPI-WLAN-006	The OpenCMAPI Enabler SHALL be able to support and identify two types of WLAN Network: ▪ Known networks which are prelisted by the operator or that have already been used/predefined by the user ▪ Unknown networks	1.0
CMAPI-WLAN-007	The OpenCMAPI Enabler SHALL be able to access/scan to the list of available SSID and provide the type of WLAN network (ex: unknown detected networks…),	1.0
CMAPI-WLAN-008	The OpenCMAPI Enabler SHALL be able to force the association on a SSID, visible or not.	1.0
CMAPI-WLAN-009	The OpenCMAPI Enabler SHALL be capable of storing several WLAN profiles	1.0
CMAPI-WLAN-010	The OpenCMAPI Enabler SHALL be able to add a WLAN profile.	1.0
CMAPI-WLAN-011	The OpenCMAPI Enabler SHALL be able to modify or delete only WLAN profile that are not predefined by the operator.	1.0
CMAPI-WLAN-012	The OpenCMAPI Enabler SHALL provide a function call to connect/disconnect to/from an WLAN access point	1.0
CMAPI-WLAN-013	The OpenCMAPI Enabler SHALL be able to listen to the WLAN events: ▪ new available network ▪ loss of network ▪ association successful on a dedicated SSID ▪ failure such as authentication failure, failure getting an IP address	1.0
CMAPI-WLAN-014	The OpenCMAPI Enabler SHALL support automatic and manual connection modes.	1.0
CMAPI-WLAN-015	The OpenCMAPI Enabler SHALL allow the user or the application using the Open CMAPI to connect to Known network or Unknown network manually.	1.0
CMAPI-WLAN-016	The OpenCMAPI Enabler SHALL allow the user or the application using the Open CMAPI to connect automatically only to Known networks.	1.0

CMAPI-WLAN-017	The OpenCMAPI Enabler SHALL be able to read/modify settings of the WLAN profile: ▪ automatic or manual mode, ▪ association priorities ▪ list of the favourite networks which are associated	1.0
CMAPI-WLAN-018	The OpenCMAPI Enabler SHALL be able to access the detailed information of SSID, at least including: ▪ SSID ▪ Signal strength per SSID (active or inactive SSID) ▪ Security or authentication mechanism used ▪ Known network or Unknown network	1.0
CMAPI-WLAN-019	The OpenCMAPI Enabler SHALL provide a function to reset the WLAN device	1.0
CMAPI-WLAN-020	If the Flight Mode is enabled, the OpenCMAPI enabler SHALL disable WLAN activities.	1.0
CMAPI-WLAN-021	The OpenCMAPI Enabler SHALL be able to access the information of WLAN connection which is currently used. At least the following information shall be included: ▪ IP address ▪ MAC address ▪ Subnet address ▪ HTTP Proxy	1.0
CMAPI-WLAN-022	The OpenCMAPI Enabler SHOULD be able to support the modification of WLAN connection information, including at least: IP address, Subnet address and/or HTTP Proxy. An access control policy mechanism SHALL be put in place to protect the access to this feature.	1.0

Table 17: WLAN Functional Requirements

6.8 Information Status Functional Requirements

This section identifies the requirements for interfaces related to Information status.

Label	Description	Release
CMAPI-INF-001	The OpenCMAPI Enabler SHALL be able to provide the Status of the PIN code	1.0
CMAPI-INF-002	The OpenCMAPI Enabler SHALL be able to provide the PLMN Name and/or the Service Provider Name as defined in [3GPP TS 22.101]	1.0
CMAPI-INF-003	The OpenCMAPI Enabler SHALL be able to provide the network selection mode.	1.0
CMAPI-INF-004	The OpenCMAPI Enabler SHALL be able to provide the signal strength.	1.0
CMAPI-INF-005	The OpenCMAPI Enabler SHALL be able to provide the CS network registration.	1.0
CMAPI-INF-006	The OpenCMAPI Enabler SHALL be able to provide the PS network registration.	1.0
CMAPI-INF-007	The OpenCMAPI Enabler SHALL be able to provide the APN set in a given connection profile.	1.0
CMAPI-INF-008	The OpenCMAPI Enabler SHALL be able to provide the IP address of the connection.	1.0

CMAPI-INF-009	The OpenCMAPI Enabler SHALL be able to provide the Roaming Status ▪ Roaming ▪ Non Roaming	1.0
CMAPI-INF-010	The OpenCMAPI Enabler SHALL be able to provide Driver Version Number	1.0
CMAPI-INF-011	The OpenCMAPI Enabler SHALL be able to provide the status of the connection: ▪ Connected ▪ Disconnected ▪ Connecting ▪ Disconnecting ▪ Unknown	1.0
CMAPI-INF-012	The OpenCMAPI Enabler SHALL be able to provide the information of the RAT Type currently in use.	1.0
CMAPI-INF-013	The OpenCMAPI Enabler SHALL be able to provide the information about the type of IP address (v4/v6) provided by the PDP context.	1.0
CMAPI-INF-014	The OpenCMAPI Enabler SHALL be able to provide the information about the QoS if provided by the network.	1.0
CMAPI-INF-015	The OpenCMAPI Enabler SHALL be able to provide the information about the SSID in use in case of connection to WLAN.	1.0

Table 18: Information Status Functional Requirements

6.9 Statistic Functional Requirements

This section identifies the requirements for interfaces related to statistics.

Label	Description	Release
CMAPI-STAT-001	The OpenCMAPI Enabler SHALL be able to provide statistics information on: ▪ Number of kB sent ▪ Number of kB received ▪ Average upload speed ▪ Average Download speed ▪ Max upload speed ▪ Max Download speed	1.0

Table 19: Statistic Functional Requirements

6.10 SMS & USSD Functional Requirements

6.10.1 SMS Functional Requirements

This section identifies the requirements for interfaces related to SMS handling

Label	Description	Release
CMAPI-SMS-001	The OpenCMAPI Enabler SHALL be able to support to send SMS	1.0
CMAPI-SMS-002	The OpenCMAPI Enabler SHALL be able to support to receive/get SMS	1.0
CMAPI-SMS-003	The OpenCMAPI Enabler SHALL be able to delete SMS	1.0
CMAPI-SMS-004	The OpenCMAPI Enabler SHALL be able to manage the storage of SMS messages in the SIM/R-UIM/NAA on UICC or in device memory.	1.0
CMAPI-SMS-005	The OpenCMAPI Enabler SHALL be able to get the list of SMS stored on SIM/R-UIM/NAA on UICC or device.	1.0
CMAPI-SMS-006	The OpenCMAPI Enabler SHALL be able to modify the status of an SMS stored	1.0
CMAPI-SMS-007	The OpenCMAPI Enabler SHALL be able to get the SMSC address	1.0
CMAPI-SMS-008	The OpenCMAPI Enabler SHALL be able to change the SMSC address	1.0
CMAPI-SMS-009	The OpenCMAPI Enabler SHALL be able to indicate that an SMS stored has been read or not	1.0
CMAPI-SMS-010	The OpenCMAPI Enabler SHALL be able to indicate that SMS has been received or not (SMS notification)	1.0
CMAPI-SMS-011	The OpenCMAPI Enabler SHALL be able to set the period of validity of a SMS	1.0
CMAPI-SMS-012	The OpenCMAPI enabler SHALL be able to set and retrieve information about how different types of incoming SMS messages are routed (save to SIM/R-UIM/UICC or device memory, discard, notify recipients, etc.)	1.0
CMAPI-SMS-013	The OpenCMAPI Enabler SHALL allow to use SMS services (send, receive…) regardless of the status of the data connection (connected or not).	1.0

Table 20: SMS Functional Requirements

6.10.2 USSD Functional Requirements

This section identifies the requirements for interfaces related to USSD service handling.

Label	Description	Release
CMAPI-USSD-001	The OpenCMAPI Enabler SHALL support mobile initiated USSD operation and perform transmission of service codes of non-implemented supplementary services and short strings via USSD (refer to [3GPP TS 22.030], section 6.5.3 and [3GPP TS 24.090]).	1.0
CMAPI-USSD-002	The OpenCMAPI Enabler SHALL allow to use USSD services regardless of the status of the connection (connected or not).	1.0
CMAPI-USSD-003	The OpenCMAPI Enabler SHALL support to transmit to an application enabling the USSD service functionality through the OpenCMAPI, text strings transparently transported from the network.	1.0
CMAPI-USSD-004	The OpenCMAPI Enabler SHALL support the transmission of Activation service codes via USSD according to the format/settings defined by the service provider.	1.0
CMAPI-USSD-005	The OpenCMAPI Enabler SHALL support the transmission of Interrogation service codes via USSD according to the format/settings defined by the service provider.	1.0

Table 21: USSD Functional Requirements

6.11 GNSS Functional Requirements

This section identifies the requirements for interfaces related to GNSS service handling.

Label	Description	Release
CMAPI-GNSS-001	The OpenCMAPI Enabler SHALL be able to support the use of standalone GNSS functionality as well as Assisted GPS	1.0
CMAPI-GNSS-002	The OpenCMAPI Enabler SHALL enable a CM application to determine the types of GNSS support (assisted GPS, standalone, neither or both) that are available, and to select which type will be used for device positioning requests.	1.0
CMAPI-GNSS-003	The OpenCMAPI enabler SHALL provide an API that enables a CM application to request the device's current position.	1.0
CMAPI-GNSS-004	The OpenCMAPI Enabler SHALL provide APIs for retrieving and setting the GNSS state (enabled or disabled).	1.0
CMAPI-GNSS-005	The OpenCMAPI Enabler SHALL provide APIs for retrieving the GNSS tracking status.	1.0
CMAPI-GNSS-006	The OpenCMAPI Enabler SHALL provide APIs for configuring GNSS operational parameters, including position fix timeout, fix request interval and accuracy threshold (in meters).	1.0
CMAPI-GNSS-007	The OpenCMAPI Enabler SHALL provide an API for specifying a GNSS system time reference.	1.0
CMAPI-GNSS-008	The OpenCMAPI Enabler SHALL provide APIs for the downloading of positioning assistance data.	1.0
CMAPI-GNSS-009	The OpenCMAPI Enabler SHALL provide APIs for the retrieval and setting of the Assisted GNSS Server IP address and port number.	1.0
CMAPI-GNSS-010	The OpenCMAPI Enabler SHALL provide a callback function to notify a CM application when NMEA positioning data is available.	1.0
CMAPI-GNSS-011	The Open CM API Enabler SHALL provide APIs for the retrieval and setting the FQDN of the Assisted GPS Server.	1.0

Table 22: GNSS Functional Requirements

6.12 Power Management Functional Requirements

This section identifies the requirements related to power management supported by the OpenCMAPI Enabler.

Label	Description	Release
CMAPI-POW-001	The OpenCMAPI Enabler SHALL support entering to and returning from hibernation and standby mode.	1.0
CMAPI-POW-002	The OpenCMAPI Enabler SHALL provide a configuration option to enable/disable the flight mode (switch Off/On all radio activity)	1.0

Table 23: Power Management Functional Requirements

6.13 Callback Functional Requirements

This section identifies the requirements for callback interfaces supported by the OpenCMAPI Enabler.

Label	Description	Release
	The OpenCMAPI Enabler SHALL be able to support the registration of callbacks for	
CMAPI-CBK-001	- session state changes (connected, disconnected)	1.0
CMAPI-CBK-002	- for data bearer status change (GPRS, WCDMA, etc)	1.0
CMAPI-CBK-003	- traffic channel dormancy status change	1.0
CMAPI-CBK-004	- activation status change for cdma2000 networks	1.0
CMAPI-CBK-005	- power state (on, off, flight mode) change	1.0
CMAPI-CBK-006	- roaming indication	1.0
CMAPI-CBK-007	- signal strength crossing a threshold range	1.0
CMAPI-CBK-008	- RF information (radio technology, band class, channel) change	1.0
CMAPI-CBK-009	- GNSS enablement and tracking status change	1.0
CMAPI-CBK-010	- new SMS message received	1.0
CMAPI-CBK-011	- Rx/Tx byte counts (periodic notification)	1.0
CMAPI-CBK-012	- Card Application Toolkit (CAT) Proactive Command received	1.0
CMAPI-CBK-013	- RAT Type changes	1.0
CMAPI-CBK-014	- change in the QoS if provided by the network	1.0
CMAPI-CBK-015	- incoming call	1.0
CMAPI-CBK-016	- USSD reception	1.0

Table 24: Callback Functional Requirements

6.14 UICC Functional Requirements

This section identifies the requirements for interfaces related to UICC by the OpenCMAPI Enabler.

Label	Description	Release
CMAPI-UICC-001	The OpenCMAPI Enabler SHALL provide an API that enables the CM application to retrieve the SIM/R-UIM/UICC ICCID.	1.0
CMAPI-UICC-002	The OpenCMAPI Enabler SHALL provide APIs for managing the de personalisation cycle using the CCK (Corporate Control Key), NCK (Network Control Key), NSCK (Network Subset Control Key), PCK (Personalisation Control Key), SPCK (Service Provider Control Key) as defined in [3GPP TS 22.022] and the CCK (Corporate Control Key), HNCK (HRPD Network Control Key), NCK1 (Network Type 1 Control Key), NCK2 (Network Type 2 Control Key), PCK (Personalization Control Key), SPCK (Service Provider Control Key) as defined in [3GPP2 C.S0068]	1.0
CMAPI-UICC-003	The OpenCMAPI Enabler SHALL provide an Access Control mechanism to control the access of CM applications to the Card Application Toolkit (CAT) features (see [3GPP TS 31.111], [3GPP2 C.S0035], [ETSI TS 102 223]).	1.0
CMAPI-UICC-004	The OpenCMAPI enabler SHALL support an Access Control mechanism to control the access of Connection Manager applications sending any APDU to the Card	1.0
CMAPI-UICC-005	The OpenCMAPI Enabler SHALL provide an API that enables a CM application to have access to the Card Application Toolkit (CAT) functionality including, at least, the retrieval of the content of the following Proactive UICC Commands: ▪ Display Text ▪ Get In-Key ▪ Get Input ▪ Setup Menu ▪ Select Item ▪ Setup Event list ▪ Setup Idle Mode Text ▪ Language Notification ▪ Refresh (see [3GPP TS 31.111], [3GPP2 C.S0035], [ETSI TS 102 223]).	1.0
CMAPI-UICC-006	The OpenCMAPI enabler SHALL provide a mean for a Connection Manager application to enrich the Terminal Profile Command sent by the device/modem to the Card with the Connection Manager application supported Proactive UICC Commands when these Proactive UICC Commands are not already supported by the device/modem.	1.0
CMAPI-UICC-007	The OpenCMAPI enabler SHALL provide a mean for a Connection Manager application to receive from the card and implement any Proactive UICC Commands when these Proactive UICC Commands are not already supported by the device/modem" (examples of connection related Proactive UICC Commands: "Send SMS", "Send USSD", BIP = Bearer Independent Protocol ("Open Channel"(all different modes), "Close Channel", "Receive data", Send Data", "Get Channel Status", "Data available event", "Channel status event"), "Launch Browser", "Browser termination event").	1.0
CMAPI-UICC-008	The OpenCMAPI Enabler SHALL provide APIs to enable CM applications to send, at least, the following CAT ENVELOPE Commands to the UICC: ▪ Menu Selection ▪ Event Download – User Activity ▪ Event Download – Idle Screen Available ▪ Event Download – Language Selection (see [3GPP TS 31.111], [3GPP2 C.S0035], [ETSI TS 102 223], [ETSI TS 102 221]).	1.0

CMAPI-UICC-009	The OpenCMAPI Enabler SHALL support Card Application Toolkit (CAT) functionality including, at least, the previously mentioned Proactive UICC Commands and the following Commands to manage the Proactive session: ▪ Terminal Profile ▪ Fetch ▪ Terminal Response (see [3GPP TS 31.111], [3GPP2 C.S0035], [ETSI TS 102 223], [ETSI TS 102 221]).	1.0
CMAPI-UICC-010	The OpenCMAPI Enabler SHALL provide a callback that notifies a CM application when the UICC has sent a Card Application Toolkit (CAT) Proactive UICC Command.	1.0
CMAPI-UICC-011	The OpenCMAPI Enabler SHALL only enable one CM application at a time to register a callback for a specific Proactive UICC Command.	1.0
CMAPI-UICC-012	The OpenCMAPI Enabler SHALL provide an API that enables a CM application to send to the UICC a response to a CAT Proactive UICC Command if this CM application has a callback registered for the Command and is allowed to send this response.	1.0

Table 25: UICC Functional Requirements

6.15 Tethering Functional Requirements

This section identifies the requirements for the tethering supported by the OpenCMAPI Enabler.

Label	Description	Release
CMAPI-TETH-001	When mobile phone is connected by the user (e.g. via USB cable or via Bluetooth) and is acting as a modem, the OpenCMAPI Enabler SHALL be able to manage the mobile phone as a datacard or embedded module (e.g. com port, modem port...).	1.0
CMAPI-TETH-002	When mobile phone is connected by the user (e.g. via USB cable or via Bluetooth) and is acting as a modem, the OpenCMAPI Enabler SHALL rely on the security related aspects (e.g. PIN code, PUK code) which are addressed and handled by the mobile phone (e.g. PIN code set by the user on his/her mobile phone).	1.0
CMAPI-TETH-003	The OpenCMAPI Enabler SHALL support same functionalities (e.g. feature as connect/disconnect, capabilities of get strength signal, SMS features) regardless whether the Open CM API enabler is used in tethering situation or not.	1.0

Table 26: Tethering Functional Requirements

6.16 Data PUSH Service Functional Requirements

This section identifies the requirements related to PUSH Service by the OpenCMAPI Enabler. These Requirements will apply only for Smartphones and Tablets.

Label	Description	Release
CMAPI-PUSH-001	The OpenCMAPI Enabler SHALL provide an access control mechanism to manage the data service using PUSH service. When the PUSH option is turned ON by the user, application using the OpenCMAPI Enabler is able to receive a PUSH service; when the PUSH option is turned OFF by the user, application using the CMAPI is not able to receive PUSH service without affecting other data service manually performed by end user.	1.0
CMAPI-PUSH-002	The OpenCMAPI Enabler SHALL provide a callback that notifies an application when a new PUSH message has been received.	1.0
CMAPI-PUSH-003	The OpenCMAPI Enabler SHALL be able to provide the content type (e.g. content-type: application/vnd.wap.sia) from the PUSH message to an application.	1.0
CMAPI-PUSH-004	The OpenCMAPI Enabler SHALL be able to provide the application id (e.g. x-wap-application-id: x-wap-application:push.sia) from the PUSH message to an application.	1.0
CMAPI-PUSH-005	The OpenCMAPI Enabler SHALL be able to provide the current bearer type over which the PUSH session is established to an application.	1.0

Table 27: Data PUSH Services Functional Requirements

Appendix A. Change History (Informative)

A.1 Approved Version History

Reference	Date	Description
OMA-RD-OpenCMAPI-V1_0-20160126-A	26 Jan 2016	Status changed to Approved by TP TP Ref # OMA-TP-2016-0009-INP_OpenCMAPI_V1_0_ERP_for_final_Approval

Appendix B. Corresponding Table – Device Types and requirements groups (Normative)

The OpenCMAPI Enabler consists of the following functional groups of requirements:

- **Network Types**: This group of requirements defines the list of Networks/Bearers that OpenCMAPI Enabler will support
- **Cellular Network Management:** This group of requirements defines the capabilities of OpenCMAPI Enabler facilitating the management of the connection to cellular Networks.
- **Device Service Handling:** This group of requirements defines the capabilities of OpenCMAPI Enabler facilitating the handling of the different information related to the device.
- **Connection Management:** This group of requirements defines the capabilities of OpenCMAPI Enabler facilitating the management of the connection including user profile management and Network selection modes.
- **CallBack:** This group of requirements defines the capabilities of OpenCMAPI Enabler to present callback functionalities
- **WLAN handling:** This group of requirements defines the capabilities of OpenCMAPI Enabler facilitating the management of WLAN.
- **WLAN authentication:** This group of requirements defines the capabilities of OpenCMAPI in term of auto login to a WLAN network type
- **Status information handling:** This group of requirements defines the capabilities of OpenCMAPI Enabler facilitating the management of different status information.
- **Statistics Management:** This group of requirements defines the capabilities of OpenCMAPI Enabler facilitating the management of statistics information.
- **SMS service handling:** This group of requirements defines the capabilities of OpenCMAPI Enabler facilitating the management of SMS service.
- **USSD service handling:** This group of requirements defines the capabilities of OpenCMAPI Enabler facilitating the management of USSD service.
- **GNSS service handling:** This group of requirements defines the capabilities of OpenCMAPI Enabler facilitating the management of GNSS service.
- **Power Management:** This group of requirements defines the capabilities of OpenCMAPI Enabler in regards of power management aspects.
- **Tethering handling:** This group of requirements defines the capabilities of OpenCMAPI Enabler facilitating the management of tethering.
- **PIN/PUK Management:** This group of requirements defines the capabilities of OpenCMAPI Enabler facilitating the management of PIN/PUK.
- **UICC interface:** This group of requirements defines the capabilities of OpenCMAPI Enabler in regards of interfacing with the UICC

Some of these groups of requirements are more relevant for certain types of devices rather than others.

The table below identifies the relevance of these requirements and therefore if the requirements are mandatory or optional according to the type of devices between Mobile Broadband device, laptop, Wireless router (including portable router), M2M, Smartphone, Tablets or Cloud Device

	Mobile Broadband Device	Laptop	Wireless Router	M2M	Smartphone	Tablet	Cloud Devices
API Management	**M**	**M**	**M**	**M**	**M**	**M**	**M**
Device Discovery APIs	**M**	**M**	**M**	**M**	**M**	**M**	**M**
Cellular Network Management APIs	**M**	**M**	**M**	**M**	**M**	**M**	**M**
Connection Management APIs	**M**	**M**	**M**	**M**	**M**	**M**	**M**
Network Management APIs	**M**	**M**	**M**	**M**	**M**	**M**	**M**
CDMA2000 APIs	**O**	**O**	**O**	**O**	**O**	**O**	**O**
Device Service APIs	**M**	**M**	**M**	**M**	**M**	**M**	**M**
PINs/PUKs Management APIs	**M**	**M**	**M**	**M**	**M**	**M**	**M**
UICC Management APIs	**O**	**O**	**O**	**O**	**M**	**M**	**O**
WLAN APIs	**O**	**M**	**O**	**O**	**M**	**M**	**M**
Statistics APIs	**M**	**M**	**M**	**M**	**M**	**M**	**M**
Information Status APIs	**M**	**M**	**M**	**M**	**M**	**M**	**M**
SMS Management APIs	**M**	**M**	**M**	**M**	**M**	**M**	**M**
USSD Management APIs	**M**	**M**	**M**	**M**	**M**	**M**	**M**
GNSS APIs	**O**	**O**	**O**	**O**	**O**	**O**	**O**
Data Push Service Management APIs	**O**	**O**	**O**	**O**	**M**	**M**	**O**
Callback APIs	**M**	**M**	**M**	**M**	**M**	**M**	**M**

Table 28: OpenCMAPI Mandatory/Optional group of requirements relevance per device type

M – Mandatory

O – Optional