TTC技術レポート
Technical Report

# TR-M2M-0002v0.2.0

# Architecture Analysis - Part 1: Analysis of architectures proposed for consideration by oneM2M

2014 年 1 月 17 日制定

一般社団法人
情報通信技術委員会

THE TELECOMMUNICATION TECHNOLOGY COMMITTEE

TR-M2M-0002v0.2.0

Architecture Analysis - Part 1: Analysis of architectures proposed for consideration by oneM2M

＜参考＞ [Remarks]

## １．国際勧告等の関連 [Relationship with international recommendations and standards]

本技術レポートは、oneM2M で作成された Technical Report 0002v0.2.0 に準拠している。

[This Technical Report is transposed based on the Technical Report 0002v0.2.0 developed by oneM2M.]

## ２．作成専門委員会 [Working Group]

oneM2M 専門委員会 [oneM2M Working Group]

# ONEM2M
# TECHNICAL REPORT

| | |
|---|---|
| Document Number | oneM2M-TR-0002-Architecture_Analysis_Part_1-V-0.2.0 |
| Document Name: | Architecture Analysis - Part 1:<br>Analysis of architectures proposed for consideration by oneM2M |
| Date: | 2013-Jul-28 |
| Abstract: | Technical analysis of existing M2M-related Architecture work undertaken by the oneM2M founding partners, including: ARIB, ATIS, CCSA, ETSI, TIA, TTA, and TTC, as well as other relevant organizations. |

This Specification is provided for future development work within oneM2M only. The Partners accept no liability for any use of this Specification.

The present document has not been subject to any approval process by the oneM2M Partners Type 1. Published oneM2M specifications and reports for implementation should be obtained via the oneM2M Partners' Publications Offices.

About oneM2M

The purpose and goal of oneM2M is to develop technical specifications which address the need for a common M2M Service Layer that can be readily embedded within various hardware and software, and relied upon to connect the myriad of devices in the field with M2M application servers worldwide.

More information about oneM2M may be found at:  http//www.oneM2M.org

Copyright Notification

Notice of Disclaimer & Limitation of Liability

The information provided in this document is directed solely to professionals who have the appropriate degree of experience to understand and interpret its contents in accordance with generally accepted engineering or other professional standards and applicable regulations. No recommendation as to products or vendors is made or should be implied.

NO REPRESENTATION OR WARRANTY IS MADE THAT THE INFORMATION IS TECHNICALLY ACCURATE OR SUFFICIENT OR CONFORMS TO ANY STATUTE, GOVERNMENTAL RULE OR REGULATION, AND FURTHER, NO REPRESENTATION OR WARRANTY IS MADE OF MERCHANTABILITY OR FITNESS FOR ANY PARTICULAR PURPOSE OR AGAINST INFRINGEMENT OF INTELLECTUAL PROPERTY RIGHTS. NO oneM2M PARTNER TYPE 1 SHALL BE LIABLE, BEYOND THE AMOUNT OF ANY SUM RECEIVED IN PAYMENT BY THAT PARTNER FOR THIS DOCUMENT, WITH RESPECT TO ANY CLAIM, AND IN NO EVENT SHALL oneM2M BE LIABLE FOR LOST PROFITS OR OTHER INCIDENTAL OR CONSEQUENTIAL DAMAGES. oneM2M EXPRESSLY ADVISES ANY AND ALL USE OF OR RELIANCE UPON THIS INFORMATION PROVIDED IN THIS DOCUMENT IS AT THE RISK OF THE USER.

# Contents

# 1 Scope

This informative Technical Recommendation (TR) provides an analysis and comparison of existing M2M-related Architecture work undertaken by the founding partners of oneM2M, including: the Association of Radio Industries and Businesses (ARIB) and the Telecommunication Technology Committee (TTC) of Japan; the Alliance for Telecommunications Industry Solutions (ATIS) and the Telecommunications Industry Association (TIA) of the USA; the China Communications Standards Association (CCSA); the European Telecommunications Standards Institute (ETSI); and the Telecommunications Technology Association (TTA) of Korea.

In addition, architectural work by other non-oneM2M Partner Type 1 organizations is provided for consideration.

This document is intended to ensure a common understanding of existing M2M Architectural approaches, in order to facilitate future normative work resulting in oneM2M Technical Specifications (TS).

This document has been prepared under the auspices of the oneM2M Technical Plenary, by the oneM2M Architecture Working Group.

# 2 References

References are either specific (identified by date of publication and/or edition number or version number) or non-specific. For specific references, only the cited version applies. For non-specific references, the latest version of the referenced document (including any amendments) applies.

## 2.1 Partner Type 1 Organization References

Documents from the oneM2M Partner Type 1 organizations provided as the basis for this analysis are listed in the Bibliography (Annex B). In order to avoid duplication, all documents listed in the Bibliography (Annex B) of this document shall also be considered as references.

## 2.2 Other References

The following referenced documents from other than oneM2M Partners Type 1 provide additional information for the analysis provided within the present document.

[i.1] 3GPP TS 23.682 Digital cellular telecommunications system (Phase 2+); Universal Mobile Telecommunications System (UMTS); LTE; Architecture enhancements to facilitate communications with packet data networks and applications".

[i.2] OMA-DM (OMA) "OMA Device Management".

[i.3] Broadband Forum TR-069: "CPE WAN Management Protocol".

[i.4] 3GPP2 X.R0067: "Machine to Machine (M2M); Architecture and Enhancements Study for cdma2000 Networks".

[i.5] 3GPP2 X.P0068: "Network Enhancements for Machine to Machine (M2M) that relate to the architectural enhancements and deployment models for supporting Machine to Machine services in 3GPP2 networks".

# 3 Abbreviations and Acronyms

## 3.1 Abbreviations

For this document, the following abbreviations apply:

| | |
|---|---|
| Arc | Architecture (Work Area) |
| Pro | Protocol (Work Area) |
| Req | Requirements (Work Area) |

## 3.2 Acronyms

For this document, the following acronyms apply:

| | |
|---|---|
| 3GPP | 3rd Generation Partnership Project |
| 3GPP2 | 3rd Generation Partnership Project 2 |
| A | Application (TTA) |
| AAA | Authentication, Authorization and Accounting |
| AAA-SD | Authentication, Authorization and Accounting - Smart Device (TIA) |
| ACS | Auto Configuration Server (BBF) |
| API | Application Programming Interface |
| AR | Application Repository (TTA) |
| ARIB | Association of Radio Industries and Businesses (JP) |
| AS | Application Server |
| ASP | Application Service Provider |
| ATIS | Alliance for Telecommunications Industry Solutions (US) |
| BBF | BroadBand Forum |
| CCSA | China Communications Standards Association (CN) |
| CoAP | Constrained Application Protocol (IETF) |
| CPE | Customer Premises Equipment |
| CRUD | Create, Retrieve, Update, Delete |
| CWMP | CPE WAN Management Protocol (BBF) |
| DA | Device Application (ETSI M2M) |
| DB | DataBase |
| DHCP | Dynamic Host Configuration Protocol (IETF) |
| DIP | Device xIP (ETSI M2M) |
| DM | Directory Manager (TTA) |
| DREM | Device Remote Entity Management |
| DSCL | Device Service Capabilities Layer (ETSI M2M) |
| EAP | Extensible Authentication Protocol (IETF) |
| ETSI | European Telecommunications Standards Institute (EU) |
| GA | Gateway Application (ETSI M2M) |
| GBA | Generic Bootstrapping Architecture (3GPP) |
| GIP | Gateway xIP (ETSI M2M) |
| GREM | Gateway Remote Entity Management |
| GSCL | Gateway Service Capabilities Layer (ETSI M2M) |
| HPLMN | Home Public Land Mobile Network (3GPP) |
| HTTP | HyperText Transfer Protocol (W3C/IETF) |
| HTTPS | HyperText Transfer Protocol - Secure (IETF) |
| IMS | IP Multimedia Subsystem (3GPP) |
| IP | Internet Protocol |
| IWF | Inter-Working Function |
| M2M | Machine to Machine (communications) |
| MPLS | Multi-Protocol Label Switching |
| MSISDN | Mobile Subscriber Integrated Services Digital Network-Number (3GPP) |
| MSP | Machine to Machine Service Provider (ATIS) |
| MTC | Machine Type Communications (3GPP) |
| MT-SMS | Mobile Terminated SMS (3GPP) |

| | |
|---|---|
| NA | Network Application (ETSI M2M) |
| NIP | Network xIP (ETSI M2M) |
| NREM | Network Remote Entity Management |
| NSCL | Network Service Capabilities Layer (ETSI M2M) |
| NTOE | Network Telco Operator Exposure (3GPP) |
| NW | Network |
| OMA | Open Mobile Alliance |
| OMA-DM | Open Mobile Alliance - Device Management |
| P2P | Peer-to-Peer |
| PANA | Protocol for carrying Authentication for Network Access (IETF) |
| PoA | Point of Attachment |
| PSK | Pre-shared key (Security) |
| QoS | Quality of Service |
| REM | Remote Entity Management |
| REST | Representational State Transfer (IETF) |
| RO | Read-Only |
| RPC | Remote Procedure Call |
| RSE | Resource Service Entity (TTA) |
| RW | Read / Write |
| SCL | Service Capabilities Layer (ETSI M2M) |
| SCS | Services Capability Server (3GPP) |
| SIM | Subscriber Identity Module (3GPP) |
| SLA | Service-Level Agreement |
| SME | Short Message Entity (3GPP) |
| SMS | Short Message Service (3GPP) |
| SMS-SC | Short Message Service-Service Centre (3GPP) |
| TCP | Transmission Control Protocol (IETF) |
| TIA | Telecommunications Industry Association (US) |
| TISPAN | Telecommunications and Internet Services and Protocols for Advanced Networking (ETSI) |
| TLS | Transport Layer Security (IETF) |
| TLV | Type-Length-Value |
| TSB | Technical System Bulletin (TIA) |
| TTA | Telecommunications Technology Association (KR) |
| TTC | Telecommunication Technology Committee (JP) |
| UDP | User Datagram Protocol (IETF) |
| UE | User Equipment |
| URI | Uniform Resource Indicator (IETF) |
| USN | Ubiquitous Sensor Network |
| USSD | Unstructured Supplementary Service Data |
| WAN | Wide Area Network |
| WIP | Work In Progress |
| WO | Write-Only |
| XMPP | Extensible Messaging and Presence Protocol (IETF) |

# 4 Conventions

The key words "Shall", "Shall not", "May", "Need not", "Should", "Should not" in this document are to be interpreted as described according to each contributing organization for their applicable documents.

# 5 M2M Architecture Description - ARIB

No descriptive information available at time of publication.

# 6 M2M Architecture Description - ATIS

This clause describes an Architecture for a Common Service Layer and/or ecosystem enabling M2M functionality, as documented by the ATIS M2M Focus Group.

## 6.1 Introduction

The ATIS M2M Focus Group output document articulates a list of common functionalities across different verticals and evaluates this list relative to the specific importance for each vertical. This process revealed what should be key functionalities of Common M2M Service Layer. Integrating as much commonality as possible across a Common M2M Service Layer will make it possible to more efficiently deliver services throughout the network, as well as enable the hybridization of services.

## 6.2 Functional Building Blocks

This Common M2M Service Layer should be agnostic to underlying network technology (yet leveraging the unique features of these underlying networks), and it will use network layer services (such as security (encryption and authentication), QoS, policy, provisioning, etc.) through an adaptation layer/APIs. These Common M2M Service Layer functions include:

- **Provisioning:** On board new devices by creating new subscription in the common M2M service layer and network layer; activate/deactivate/suspend/resume network and service subscriptions.

- **Device Management:** Manage all aspects of the devices including configuration, firmware upgrades, application lifecycle management, device lock and wipe.

- **Security:** generate relevant key material for secure communications; Authenticate devices before they can register and modify resources. Prevent unauthorized entities from sending IP packets to the devices. Provide a secure connection to the device

- **Application and Device Registrations:** Application and devices will be able to register with the service layer entity for various services. Registration will involve authentication or verification of credentials and creation or allocation of resources within the server and the database. A profile with the capabilities of the device and the type of services allowed for the applications are created.

- **Resource Management:** Applications and devices will be able to create, update, and delete resource objects containing various attributes in the service layer. Entities will be able to discover resources.

- **Content push/pull Services:** Provide API for applications to perform unicast and multicast data push to specified devices within the specified time window. Push may be result of a notification that is triggered as a result of modification of a resource. Provide API for applications to pull data from one or more devices within the specified time window or specified periodicity or other policies that have been established.

- **Store and Forward Messaging:** Applications may request messages to be sent to one or more devices that may not be registered with the network at that time. In this case the communications management entity shall store and aggregate the messages and forward them to the devices at a later time when the devices wake up.

- **Protocol Translation:** Translate protocols between application and device as needed. For example, applications may use HTTP while devices may use Constrained Application Protocol (CoAP) or Zigbee Smart Energy 2.0 protocol.

- **Subscribe/Notification:** Application and devices should be able to subscribe to receive notifications upon certain events or when certain resources are updated. Events may be specified as rules on certain resource data.

- **Policy Framework:** Framework to establish and incorporate in the session orchestration, data aggregation and storage, the network provider and application provider policies. Examples include incorporating a location tag or time stamp on all data, policy restricting sessions only to certain hours of the day.

- **Location and Geo Fencing:** Provide device and network based location and location related services such as creating a geo-fence or identifying a group of devices within a region or adding a location tag to the device data.

- **Groups Management:** Framework for creation of groups by specifying the members of the group through one of the identifiers of the device, adding additional members or removing members; setting group attributes.

- **Device Triggering:** Provide the capability to trigger the device to register with the network and an application through a secondary means such as an SMS. Provide information about the status of the device in the network.

- **Access Control:** Control the access to the data collected from the devices based on access restrictions specified by applications in terms which users or devices can access what resources.

- **Data Processing and Storage:** Provide temporary and permanent storage for data collected from devices. Process queries on data collected. Provide threshold and expression rules setting and execution on the various data collected from the devices. Notifications could be triggered based on the outcome of the rules testing.

- **Consumption Statistics/records:** Process queries regarding the usage of network resources by a device or a group of devices for billing reconciliation.

- **API Management:** Manage API usage, such as authentication and authorization of calls to APIs provided.

The functions above have been identified as essential by the following verticals: Connected Vehicle; Smart Grid; eHealth; and Connected Home. The goal of this diligence was to determine which Common M2M service layer functions were necessary for their respective vertical segment and to then converge on those common functions as the components for a "Common M2M Service Layer" capability. Future work may need to determine whether all of these functions remain in the service layer or are provided at the application layer.

## 6.3 Functions proposed for Common M2M Service Layer

The functions proposed for a Common M2M Service Layer are:

**Device Management**

- Provision/Activate (Individual and bulk) and Bootstrap
- Suspend/Resume
- Configuration Management
- Firmware/Software Management
- Inventory Management
- Diagnostics (resource information, status)

**Policy & Resource Management**

- Authentication and Registration (Identity Management)
- Establish communications session (Add/Delete/Modify)
- QoS/SLA for communication session
- Billing, Charging, and Rating rules
- Group Management
- Security Management (Data confidentiality, integrity, abuse prevention, privacy)

**API Services**

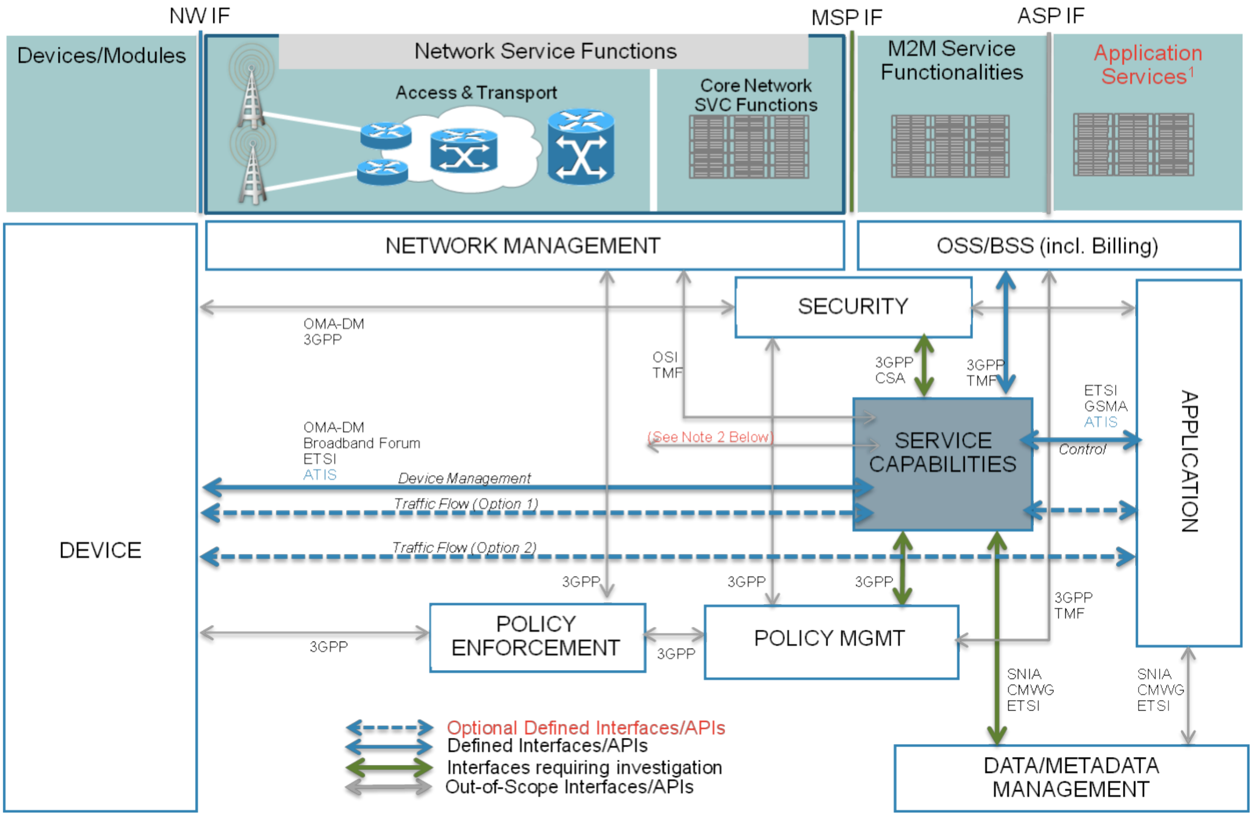
- Definition, Authentication/Authorization and Security
- Service to Device (Management, Establish/Teardown Communication Flows)

- Service to Policy/Resource Management (Rx Extensions for Group Management)
- Service to Data/Metadata Management (Storage/Retrieval)
- Service to Applications (Management, Communications Flows)

**Data/Metadata Management**

- Data processing and append (location, timestamp)
- Data storage/retrieval

Figure 6.1 shows the Common M2M Service Layer as described by ATIS M2M Focus Group. It should be noted that the location of a specific function in the figure is not reflecting assignment of that function to a certain part of the network.



***Out of scope interfaces/APIs included for completeness and informative reference***

***Note 1: "Application Services" is a generic term for the set of applications within a vertical.***

***Note 2: For simplicity, not all possible interfaces/APIs to the Service Capability entity are shown.***

**Figure 6.1: ATIS Common M2M Service Layer**

While a number of interfaces are shown that are out-of-scope of the ATIS standards and M2M environment, M2M Service Management, i.e. Policy Enforcement, Policy Management, and Data/Metadata Management, have broad reach into numerous existing systems.

## 6.4 Functional Entities and Interfaces

Mapping the blocks in the figure 6.1 to the entities defined below:

- Devices/Modules are the M2M Devices, Access & Transport
- Core Network Service functions; belong to the Network Provider
- M2M Service Functionalities belong to the M2M Service Provider

- Application Services would belong to the M2M User

From this mapping the interfaces in the diagram can be broadly grouped as the interfaces between the M2M Device, Network Provider, M2M Service Provider and M2M User as follows:

NOTE: These groups are not meant to be all-inclusive.

**Network Interface Group (NW IF):**

- This is the interface between the M2M Device and the Network Provider to provision and manage the connection of the device to the network. As indicated in the diagram it may also enforce the network policies.

**Machine to Machine Service Provider Interface Group (MSP IF):**

- This is the interface between the Network Provider and the M2M Service Provider to provision, manage service delivery, and define service capability policies on the M2M Devices.

**Application Service Provider Interface Group (ASP IF):**

- This is the interface between the M2M Service Provider and the Application Services to Bootstrap, Activate, Provision, Secure, Meter, Manage Application Services delivered on the M2M Devices.

NOTE: The Service Capabilities (Common M2M Service Layer) in figure 6.1 fall under the M2M Service Provider and may be provided by the Network Provider or the Application Services. In this case the Application Services would directly interface to the Network Provider. These interfaces present a way to evaluate the different functions, the level of integration of into the E2E entities, and their accessibility across other interfaces. For e.g. the Network Interface may not be directly visible at the MSP, but only through an abstraction to permit the MSP functions.

Given the breadth of organizations defining standards in the M2M space, figure 6.1 is not meant to include an exhaustive list of all standards organizations and interfaces defined that apply to the ATIS Common M2M Service Layer. There are other scenarios that may not be represented.

# 7 M2M Architecture Description - CCSA

See annex A for a summary of applicable documents.

No additional descriptive information available at time of publication.

# 8 M2M Architecture Description - ETSI

This clause describes the M2M Architecture as documented by ETSI Technical Committee M2M.

## 8.1 ETSI M2M architecture

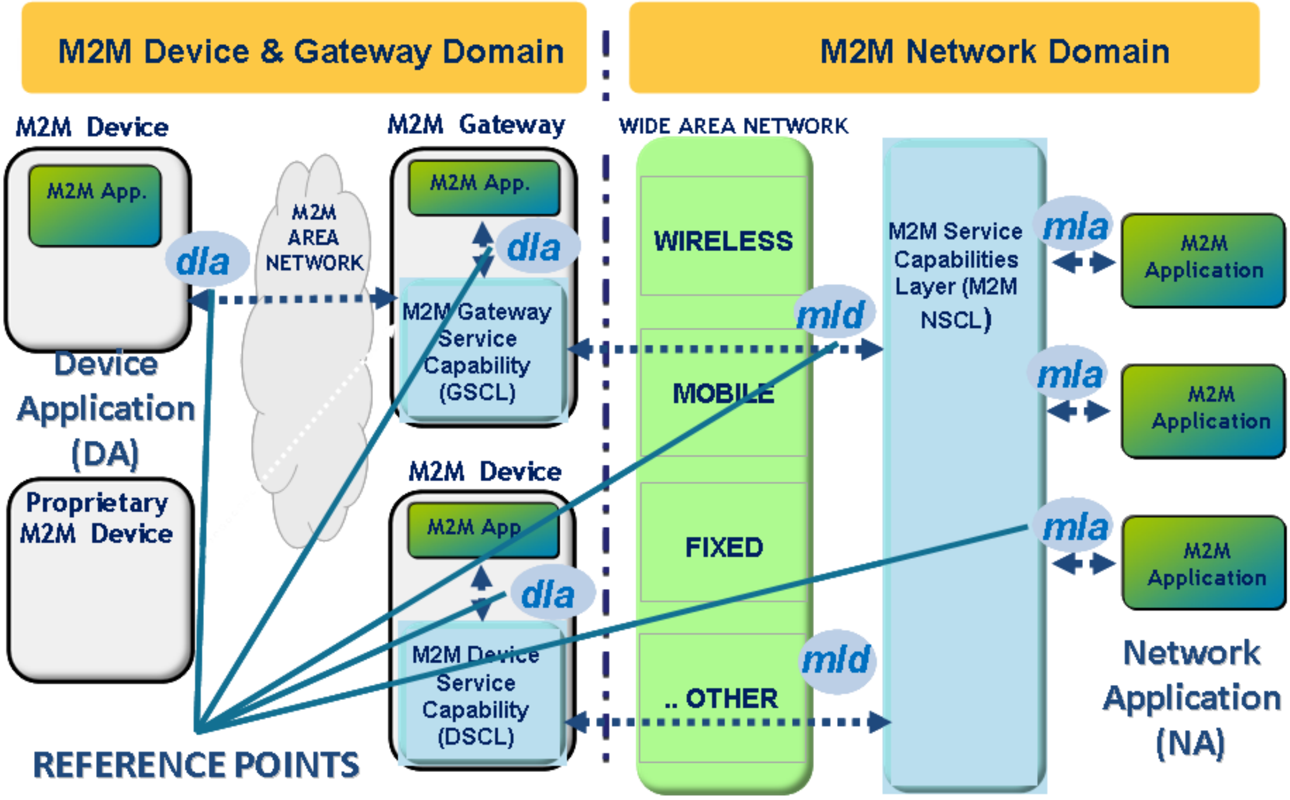


**Figure 8.1: M2M Functional Architecture Overview**

### 8.1.1 Functions and reference points

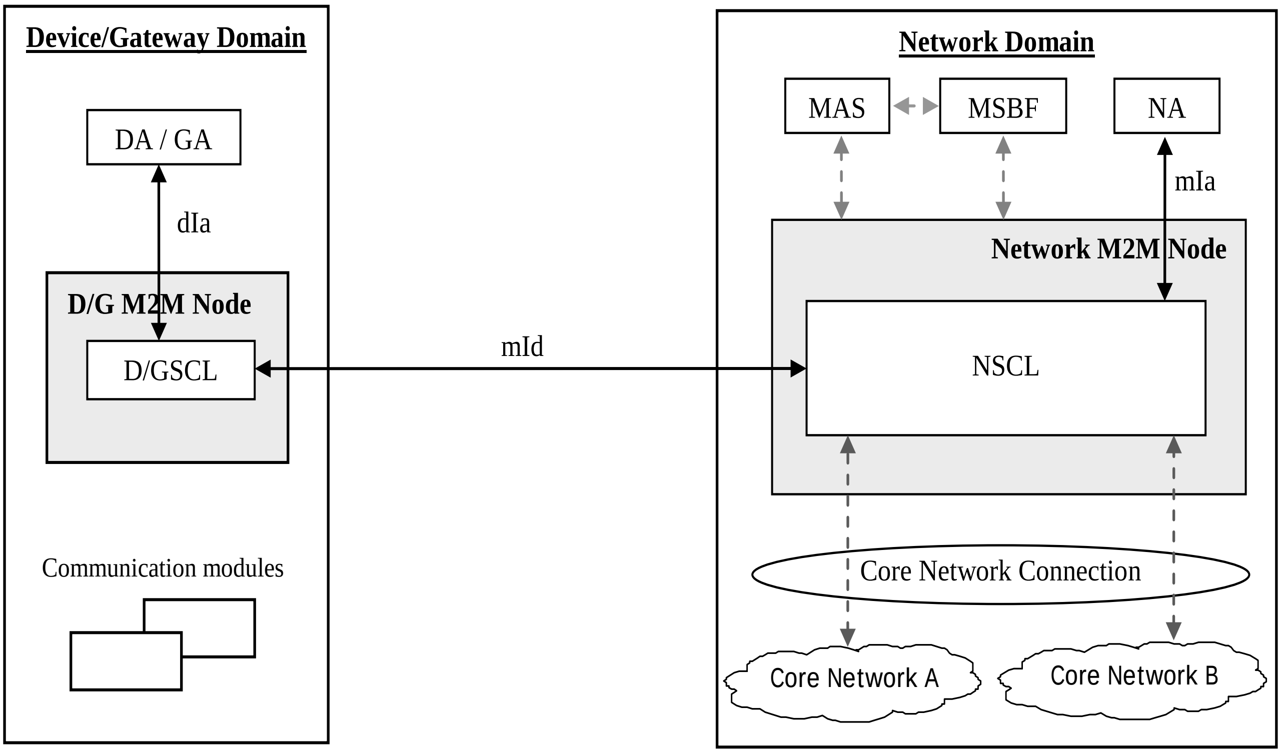


**Figure 8.2: M2M Functional Architecture Framework**

**M2M Service Capabilities** Layer provides functions that are to be exposed on the reference points. M2M SCs can use Core Network functionalities through a set of exposed interfaces (e.g. existing interfaces specified by 3GPP, 3GPP2, ETSI TISPAN, etc.). Additionally, M2M SCs can interface to one or several Core Networks.

In the remaining of the present document the following terms will be used:

- **NSCL:** Network Service Capabilities Layer refers to M2M Service Capabilities in the Network Domain.
- **GSCL:** Gateway Service Capabilities Layer refers to M2M Service Capabilities in the M2M Gateway.
- **DSCL:** Device Service Capabilities Layer refers to M2M Service Capabilities in the M2M Device.
- **SCL:** Service Capabilities Layer refers to any of the following: NSCL, GSCL, or DSCL.
- **D/G SCL:** refers to any of the following: DSCL, GSCL.

The external reference points (mIa, mId, dIa) are mandated and are required for ETSI M2M compliance.

**M2M Node:** is a logical representation of the M2M components in the M2M Device, M2M Gateway, or the M2M Core. An M2M Node shall include one SCL, and optionally an M2M Service Bootstrap function and an M2M Service Connection function. An M2M Node relies on a Secured Environment Domain, controlled by the M2M Service Provider associated with the SCL, to protect Sensitive Functions and Sensitive Data.

A Device/Gateway M2M Node shall be instantiated upon pre-provisioning or executing an M2M Service Bootstrap procedure on the M2M Device/Gateway with an M2M Service Provider. Each Device/Gateway M2M Node may be instantiated with only one M2M Service Provider.

**M2M Applications:** are respectively Device Application (DA), Gateway Application (GA) and Network Application (NA). DA could reside in an M2M Device which implements M2M Service Capabilities (referred to in clause 6.1 as D device) or alternatively reside in an M2M Device which does not implement M2M Service Capabilities referred to in clause 6.1 as D' device).

### 8.1.2 Reference points

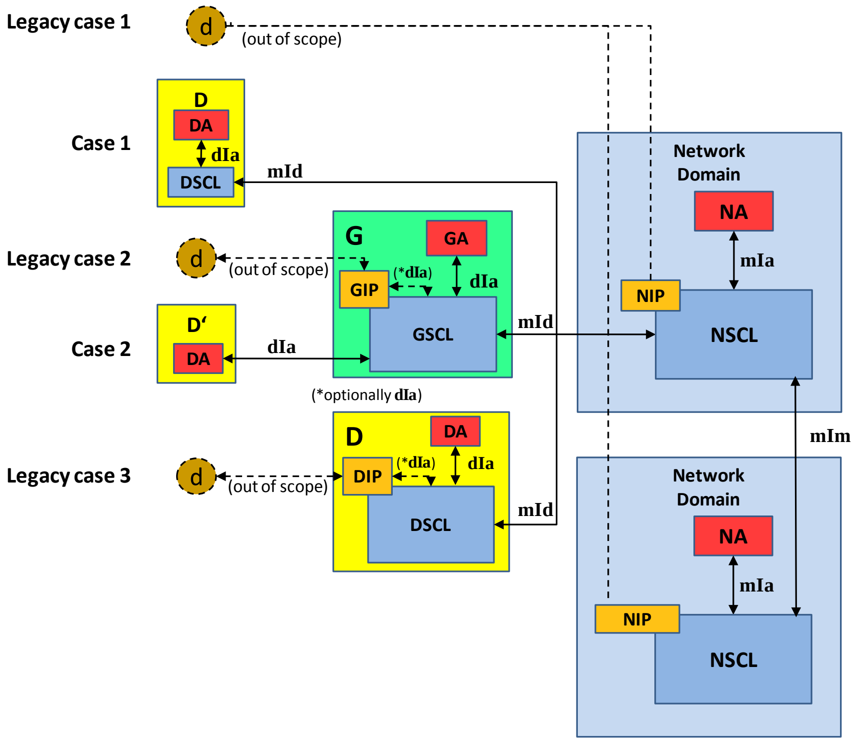


**Figure 8.3: Mapping of reference points to different deployment scenarios**

**Gateway (G):** shall provide M2M Service Capabilities (GSCL) that communicates to the NSCL using the mId reference point and to DA or GA using the dIa reference point.

**Device (D):** shall provide M2M Service Capability (DSCL) that communicates to an NSCL using the mId reference point and to DA using the dIa reference point.

**Device' (D'):** shall host DA that communicates to a GSCL using the dIa reference point. D' does not implement ETSI M2M Service Capabilities.

Additionally there is a non-ETSI M2M compliant device ('d') that connects to SCL using the xIP Capability (NIP, GIP, or DIP). 'd' devices do not use ETSI M2M defined reference points, however:

- GIP may either be an internal capability of GSCL or an application communicating via reference point dIa with GSCL.
- DIP may either be an internal capability of DSCL or an application communicating via reference point dIa with DSCL.

**mIa**

The mIa reference point offers generic and extendable mechanism for Network Applications interactions with the NSCL.

The mIa reference point, functions include:

- Registration of NA to the NSCL.
- Request to Read/Write, subject to proper authorization, information in the NSCL, GSCL or DSCL.
- Request device management actions (e.g. software upgrade, configuration management).
- Subscription and notification to specific events.
- Request the creation, deletion and listing of group(s).

**dIa**

- The dIa reference point offers generic and extendable mechanism for Device Application (DA)/Gateway Application (GA) interactions with the DSCL/GSCL.
- The dIa reference point functions include:
    - Registration of D/GA to GSCL.
    - Registration of DA to DSCL.
- Request to Read/Write, subject to proper authorization, information in the NSCL, GSCL, or DSCL.
- Subscription and notification to specific events.
- Request the creation, deletion and listing of group(s).

**mId**

The mId reference point offers generic and extendable mechanism for SCL interactions.

The mId reference point, functions include:

- Registration of a DSCL/GSCL to NSCL.
- Request to Read/Write, subject to proper authorization, information in the NSCL, GSCL, or DSCL.
- Request device management actions (e.g. software upgrade, configuration management).
- Subscription and notification to specific events.
- Request the creation, deletion and listing of group(s).
- Provides security related features as defined in clause 8.

**mIm**

The mIm reference point offers generic and extendable mechanisms for NSCL-to-NSCL interactions and communications.

## 8.1.3 M2M Resource Management and Procedures

### 8.1.3.1 Usage of resources in a RESTful architecture

ETSI M2M adopts a RESTful architecture style. This style governs how M2M Applications (DA, GA, NA) and/or M2M SCL are exchanging information with each other. A RESTful architecture is about the transfer of representations of uniquely addressable resources. ETSI M2M standardized the resource structure that resides on a SCL.

In a very simplistic view, imagine that certain resources are buckets that can hold some application specific data. These buckets - as far as the scope of an M2M service layer is concerned - reside in the respective SCL. The buckets have certain properties and are structured as suggested in more detail in the subsequent clauses.

To illustrate a very basic use of this mediator function of the M2M SCL, a simple example shall be described here. An application (DA) on an M2M Device that is not always connected wants to send some data to another application (NA) on the network by means of the M2M SCL. DA would write data to a resource in the NSCL and NA would read that resource. If configured accordingly, the NA could also be notified by the NSCL upon writing (update) of the resource by the DA, in order to facilitate synchronization between DA and NA. Figure 8.4 is meant to illustrate that process of the data flow and not the operation flow.

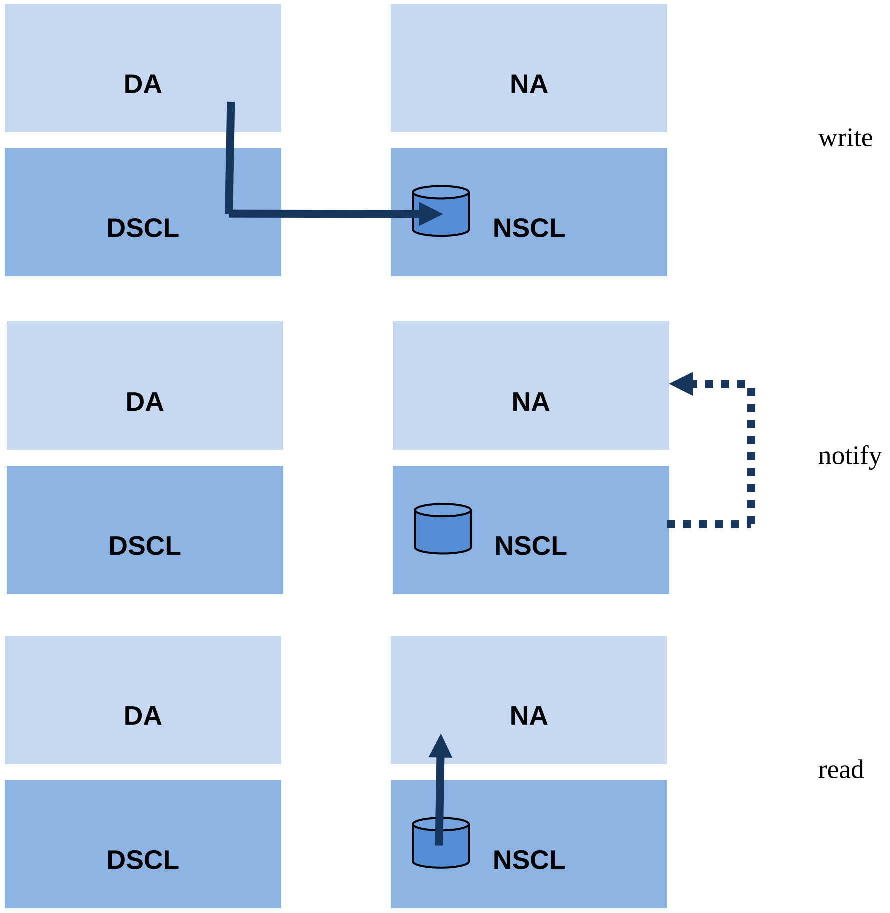


**Figure 8.4: Simple example for use of SCL resources to exchange data**

When handling resources in a RESTful architecture, there are four basic methods - so called "verbs" - that could be applied to resources:

- CREATE: Create child resources.
- RETRIEVE: Read the content of the resource.
- UPDATE: Write the content of the resource.
- DELETE: Delete the resource.

These methods are referred to as the CRUD methods below. In addition to these basic methods in a RESTful architecture, it is often also useful to define verbs actions that might not directly map to one of the specific method. Moreover a definition of these particular verbs helps the readability of the present document if the verb is chosen appropriately. The additional verbs introduced are:

- NOTIFY: used to indicate the operation for reporting a notification about a change of a resource as a consequence of a subscription. This verb would either map to a response of a RETRIEVE method in case that the long polling mechanism is used, or to an UPDATE method in case that the asynchronous mechanism is used.
- EXECUTE: for executing a management command/task which is represented by a resource. This verb corresponds to an UPDATE method without any payload data.

### 8.1.3.2 Definitions

This clause provides definitions that are used for describing the resource procedures:

**Resource:** is a uniquely addressable entity in the RESTful architecture. A resource has a representation that shall be transferred and manipulated with the verbs. A resource shall be addressed using a Universal Resource Identifier (URI).

**Sub-Resource:** also called child resource. It is a resource that has a containment relationship with the addressed (parent) resource. The parent resource representation contains references to the children. The lifetime of the sub-resource is linked to the parent's resource lifetime.

**Attribute:** is meta-data that provides properties associated with a resource representation.

**Attribute-Type:** attributes are distinguished by the following types:

**Table 8.1**

| | |
|---|---|
| RW | Read/Write by client |
| RO | Read-Only by client, set by the server |
| WO | Write-Once, can be provided at creation, but cannot be changed anymore |

Note that if an attribute is RW, it does not mean that the Issuer can set it to any value. Some values may be restricted by the Hosting SCL, e.g. due to policies. For example, an expiration time may be written by the Issuer, but it is treated as a suggestion by the Hosting SCL. The Hosting SCL is free to change (i.e. lower) the expiration time. If the Hosting SCL changes anything it shall send back a full representation of the resource as it is created, i.e. a success response with a body.

**Issuer:** is the actor performing a request. An issuer shall either be an Application or a SCL.

**Announced Resource:** the content of this resource refers to a resource hosted by the Hosting SCL (Master/original Resource). The purpose of this resource is to facilitate a discovery of the original resource, so that the issuer of the discovery does not have to contact all SCLs in order to find the resource.

**Local SCL:** The Local SCL is the SCL where an Application or a SCL shall register to. It is the first SCL that receives the request from the original issuer of the request (either an Application or a SCL):

- if the NA is the original issuer, the Local SCL is the NSCL;
- if the GA is the original issuer, the Local SCL is the GSCL;
- if the DA in a D device is the original issuer, the Local SCL is the DSCL;
- if the DA in a D' device is the original issuer, the Local SCL is the GSCL;
- if the DSCL in a D Device is the original issuer then the local SCL is the NSCL or the GSCL.

**Hosting SCL:** The SCL where the addressed (Master/original Resource) resource resides.

NOTE: In some cases the hosting SCL also act as Local SCL, this is valid in case of a registration.

**Announced-to SCL:** a SCL that contains the announced resource (a resource could be announced to multiple SCLs).

**Receiver:** it represents the actor that receives a request from an issuer. A receiver shall be a SCL or an Application.

### 8.1.3.3 Resource structure

This clause introduces the main types of resource used in a SCL. Since all of these resources will have to be addressed in some way and since there are relationships between them (like a parent-child containment relationship), a hierarchical tree structure for modelling their structure and relationships is included.

**SclBase Resource**

The sclBase resource shall contain all other resources of the hosting SCL. An sclBase resource is the root of all other resources it contains. The sclBase resource shall represented by an absolute URI. All other resources hosted in the SCL shall also be identified by a URI.

For example, a specific sclBase resource identifying a Network SCL could be http://m2m.myoperator.org/some/arbtrary/base/".

An example of a URI identifier of a container resource hosted by this network SCL could be "http://m2m.myoperator.org/some/arbtrary/base/containers/myExampleContainer".

**SCL Resource**

SCL resource shall represent an associated (remote) SCL that is authorized to interact with the hosting SCL. In order to be authorized to interact with the hosting SCL, the remote SCL has to go through a M2M service registration procedure. An SCL resource is created as a result of a successful registration of the remote SCL with its local SCL or vice-versa. SCL resource shall store context information about the registered SCLs.

**Application Resource**

Application resource shall store information about the Application. Application resource is created as a result of successful registration of an Application with the local SCL. Applications shall only register to their local SCL.

**AccessRight Resource**

AccessRight resource shall store a representation of permissions. An AccessRight resource is associated with resources that shall be accessible to entities external to the hosting SCL. Basically, these control "who" (permissionHolder) is allowed to do "what" (permissionFlag). Moreover, they can be used in the privacy protection.

**Container Resource**

Container resource is a generic resource that shall be used to exchange data between applications and/or SCLs by using the container as a mediator that takes care of buffering the data. Exchange of data between applications (e.g. on device and network side) is abstracted from the need to set up direct connections and allows for scenarios where both parties in the exchange are not online at the same time.

**LocationContainer Resource**

A LocationContainer resource shall represent a container for the location information of a M2M entity (e.g. M2M Device/Gateway). The location information may be generated by a Device/Gateway application or provided by location server in the network domain.

**Group Resource**

Group resource shall be used to define and access groups of other resources.

For example, a group resource could be used to write the same content to a group of M2M container resources (this allows a DA to write the same data to many container resources in a NSCL. The data is only sent once to the NSCL, while letting the NSCL replicate it to different container resources. This is a more optimal use of the dIa/mId reference point).

**Subscription Resource**

Subscription resource shall be used to keep track of status of active subscription to its parent resource. A subscription represents a request from the Issuer to be notified about modifications on the parent resource.

**M2MPoC Resource**

The M2MPoC resource shall represent information maintained in the NSCL on how to reach a DSCL or GSCL via a specific access network. The device or gateway maintains this information in the NSCL by creating, updating, or deleting this resource when their point of attachment changes.

**MgmtObj Resource**

A MgmtObj resource holds the management data which represents a certain type of M2M remote entity management function. For a remote entity (i.e. M2M Device/Gateway) multiple MgmtObj resources may be created on NREM for different management purposes.

**MgmtCmd Resource**

A MgmtCmd resource shall be only used to model non-RESTful management commands, i.e. BBF TR-069 [i.3] Remote Procedure Call (RPC) methods, as listed in table 9.1. This resource represents the RPC methods in a RESTful manner. With such RESTful modelling, a MgmtCmd (i.e. a RPC) shall be triggered by an NA using the RESTful verb UPDATE. If supported by the specific BBF TR-069 [i.3] procedure, a triggered MgmtCmd may be cancelled before it finishes by an NA using UPDATE verb or a DELETE verb.

**AttachedDevices Resource**

An AttachedDevices resource shall be used to collect the management information of all M2M D' devices that are attached to a M2M Gateway. It shall reside under a G <scl> resource created in the remote NSCL.

**AttachedDevice Resource**

An AttachedDevice resource shall be used to represent each M2M D' device that is attached to a M2M Gateway. The resource lives only in the NSCL and it shall reside under the AttachedDevices resource of the corresponding M2M Gateway.

**Announced Resource**

An announced resource shall point to the original resource hosted in another SCL. The announced resource is an actual resource which consists of only a limited set of attributes, which are the searchStrings, the link to the original resource and the access right. The purpose of the announced resource is to facilitate a discovery of the original resource when querying the announced-to SCL, so that the issuer of the discovery does not have to contact all SCLs in order to find the resources. An announced resource itself shall only be visible when it is directly accessed via its full URI. During discovery a direct reference to the original resource shall be returned. Only locally created resources can be announced.

Removing an announced resource, for example due to deregistration of an SCL (which correspond to a removal of the parent SCL resource), does not remove the original resource, but does remove all the children of announced resource Reversely, when the original resource is removed, it is the responsibility of the original SCL, where the original resource is hosted, to remove the announced resource. If this is not done (e.g. because the original SCL is offline), the announcement resource shall be removed when it expires.

NOTE: There are collections with:

1) only real resources;

2) a mix of real resources and announced resources;

3) only announced resources.

A resource of the same type (e.g. <*application*> resources) might have different content depending on where in the tree it is located. There are different ways of representing this, i.e. either by defining different resources or by defining one collection resource that contain both types of children. The latter solution is chosen, but it is indicated in each case whether which child resources are allowed.

**NotificationChannel Resource**

A NotificationChannel resource shall be used by non-server capable client to receive asynchronous notifications. The notification channel is prepared to handle several mechanisms on how to receive these asynchronous notifications. However, currently only one mechanism is fully specified, which is the so-called "long polling" mechanism. This method is based on the server not responding to requests until a notification needs to be sent (or until a timeout occurs).

**Discovery Resource**

A discovery resource shall be used to allow discovery. It is used to retrieve the list of URI of resources matching a discovery filter criteria. It does not represent a real resource in the sense that it does not have a representation and it shall never be cached.

**Collection Resource**

This resource represents an abstract concept that is applicable to various resources in the resource structure. For details on the collection resources, see clause 9.3.

A collection resource normally has its own associated accessRightID and allows subscriptions on modification in the collection resource.

EXAMPLE: When resources contain a collection of similar sub-resources, this is modelled as a collection resource. There are several collection resources identified, e.g. the SCL resource mentioned above contains collection resources for <*group*> resources, for <*container*> and <*locationContainer*> resources, for <*application*> resources, for <*accessRight*> resources and for <mgmtObj> and <mgmtCmd> resources. A collection can contain local resources and/or the corresponding announced resources. A collection resource representation contains the sub-resources by reference and it may also contain attributes.

**Common attributes**

Many of the attributes of the resources described in the present document are common. Those attributes are described here once in order to avoid duplicating the description for every resource that contains it.

Attributes that are only used in one or two resource types are described only in the section for that resource.

**Table 8.2**

| Name | Description |
|---|---|
| accessRightID | URI of an access rights resource. The permissions defined in the accessRight resource that is referenced determine who is allowed to access the resource containing this attribute for a specific purpose (retrieve, update, delete, etc.).<br>If a resource type does not have an accessRightID attribute definition, then the accessRights for resources of that type are governed in a different way, for example, the accessRight associated with the parent may apply to a child resource that does not have an accessRightID attribute definition, or the permissions for access are fixed. Refer to the corresponding procedures to see how permissions are handled in these cases.<br><br>If a resource type does have an accessRightID attribute definition, but the (optional) accessRightID attribute is not set, or it is set to a value that does not correspond to an valid, existing, accessRight resource, or it refers to an accessRight resource that is not reachable (e.g. because it is located on a remote SCL that is offline or not reachable), then the system default access permissions shall apply.<br><br>The system default access permissions grant all permissions (i.e. the full set of permissionsFlags) to the following permission holders depending on the prefix of URI of the resource.<br><br>The permissionHolders for prefixes from most specific to least specific are as follows:<br>*<sclBase>/scls/<scl>/applications/<applicationAnnc>*: the hosting SCL, the SCL corresponding to the <*scl*> resource and the Application corresponding to the <*applicationAnnc*> resource shall be the permissionHolders.<br>*<sclBase>/scls/<scl>*: the hosting SCL and the SCL corresponding to the <*scl*> resource shall be the permissionHolders.<br>*<sclBase>/applications/<application>*: the hosting SCL and the Application corresponding to the <*application*> resource shall be the permissionHolders.<br><*sclBase*>: the hosting SCL shall be the permissionHolder. |
| announceTo | In a request on mIa or dIa, this is interpreted as the list of the SCLs that the SCL will try to announce to on behalf of the requestor. In responses, the list indicates the actual list of resources to which the resource is announced at the moment.<br>If this attribute is not provided in requests on the mIa or dIa, the local SCL will decide where the resource will be announced. |
| creationTime | Time of creation of the resource. |
| expirationTime | Absolute time after which the resource will be deleted by the hosting SCL. This attribute can be provided by the issuer, and in such a case it will be regarded as a hint to the hosting SCL on the lifetime of the resource. The hosting SCL can however decide on the real expirationTime. If the hosting SCL decides to change the expirationTime attribute value, this is communicated back to the issuer.<br>The lifetime of the resource can be extended by providing a new value for this attribute in an UPDATE verb. Or by deleting the attribute value, e.g. by not providing the attribute when doing a full UPDATE, in which case the hosting SCL can decide on a new value. |
| filterCriteria | These are criteria that filter the results. They can either be used in a GET (as query parameters) or in a subscribe. |

| Name | Description |
|---|---|
| lastModifiedTime | Last modification time of a resource. |
| link | URI of the related remote resource. In an announced resource, this is the URI of the announcing resource. In an *<scl>* resource, this is the URI of the *<sclBase>* resource of the registered SCL. |
| searchStrings | Tokens used as keys for discovering resources. |

**Notation**

A tree representation is used for describing how the different types of resources relate to each other. This is essential for deriving a meaningful way to navigate to the different resources and understand their use. The same resources structure applies to resources in the NSCL, the GSCL and the DSCL.

The following notations have been used in the present document:

- The notation <resourceName> means a placeholder for an identifier of a resource of a certain type. The actual name of the resource is not predetermined.

- The notation "attribute" denotes a placeholder for one or more fixed names. Attribute names and types are described in a table for each resource showing the resource structure.

- Without the delimiters < and > or "and", names appearing in boxes are literals for fixed resource names or attributes.

- Square boxes are used for resources and sub-resources. In order to be able to access sub-resources in a RESTful way, an Issuer shall access these resources directly, only by their references (URIs) which are part of the parent resource representation. The parent resource does not include the representation of its sub-resources, Any deviations from this rules are described in details through the present document.

- Rounded boxes are used for attributes of resources. In order to be able to address and access individual attributes, or parts of an attribute, in a RESTful way they shall be accessible in a way similar to as sub-resources. This is called partial addressing. The main difference is that the attributes are served as part of the containing resource (e.g. in case of http binding they do not have separate e-tag handling and modification times are not kept per attribute).

- Parent-Child relationship and multiplicity: The parent-child relationships are indicated by solid lines. At each end of a line an indicator for the number of valid elements of a parent/child is depicted. The symbol:

  - "*" indicates any number from 0 to infinity.

  - "k", "n", "m", etc. indicate a fixed but so far undefined number of elements. If a parent resource is deleted, the containment relation implies that all child-resource shall be deleted as well (recursively).

The following conventions are used throughout the resource description

- All resource type names shall be in lower case, in case of composed name the subsequent words shall start with a capitol letter, for example "accessRight".

- All resources shall indicate an object and not an action or verb.

- If a resource identifies a collection, then the resource name shall be in a plural form, for example if we need to indicate a collection of flowers the correspondent resource shall be called "flowers".

- All attributes shall be in lower case. In case of composed name the separation between word is indicated by the using a capitol letter for the following word. For example "searchStrings".

### 8.1.3.4 Examples of ETSI M2M Resources

***<sclBase>***

The <*sclBase*> resource in this tree shall be the root for all resources that are residing on the hosting SCL.

This SCL can be reached for registering remote SCLs and/or local Applications. The <*sclBase*> hosts and manages sub-resources. An SCL shall perform M2M procedures upon requests by other entities (other SCLs or Applications). Not all defined features may be supported by specific SCLs. For example; a DSCL or GSCL may not be server capable and/or publicly addressable. This implies that some resources may only be addressed locally and not from a remote SCL. A <*sclBase*> resource shall be addressed by an URI.

The <*sclBase*> resource may contain attributes that describe the hosting SCL.

Regardless of the accessRight, the <*sclBase*> resource shall not be created or deleted via the RESTful API. The <*sclBase*> is managed outside the scope of the API. However, it may be modified and read via the API by entities that have the correct authorization, as defined by the accessRight identified by the accessRightID attribute in the <*sclBase*> resource.

The <*sclBase*> resource does contain collection resources representing collections of <*scl*> resources, <*application*> resources, <*container*> resources, <*group*> resources, <*accessRight*> resources and <*subscription*> resources. In general, where in the resource tree an entity (Application or SCL) creates a <*container*>, a <*group*> or an <*accessRight*> resource depends mainly on the lifecycle requirements on that resource.

Resources created directly in a child collection of <*sclBase*> resource live as long as the <*sclBase*> lives, and this gives the resource the possibility to outlive its creator. The creator can either be a local or remote to the <*sclBase*>.

Resources created in the child collection of a <*scl*> resource on a remote <*sclBase*> resource live as long as the <*scl*> resource is available. They will be removed at deregistration of the SCL.

Resources created in the child collection of the <*application*> resource or a local application will live as long as the local <*application*> resource is available. They will be removed at deregistration of the application.

Resources created in the child collection of the <*applicationAnnc*> resource will live as long as that announcement resource lives. They will be removed when de-announcing of the application.

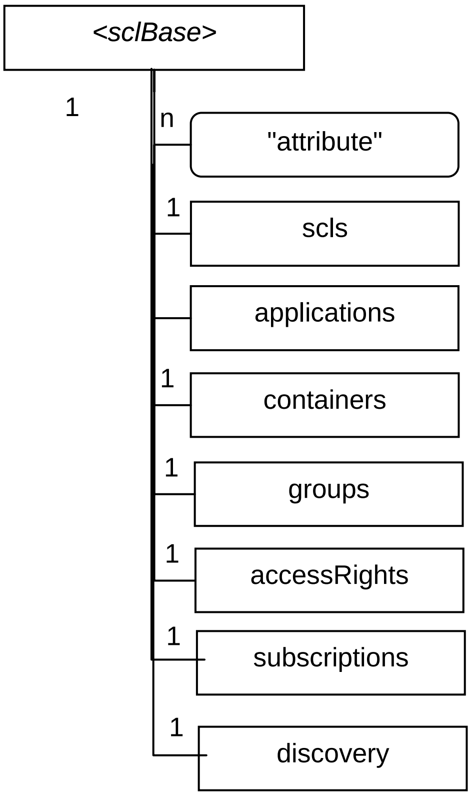


**Figure 8.5: Structure of <*sclBase*>-resources**

The <*sclBase*> sub resources are described in the following.

**Table 8.3**

| subResource | Description |
|---|---|
| scls | Collection of <*scl*> resources each representing remote SCLs with which the hosting SCL is registered, or that is registered with the hosting SCL. The collection only contains <*scl*> resources, representing remote SCLs. |
| applications | Collection of <*application*> resources which are registered the hosting SCL represented by the <*sclBase*> resource.<br>This collection contains only <*application*> resources, representing local Applications. |
| containers | Collection of <*container*> resources that do not have a containment relation with a specific remote entity (Application or SCL). This means that if the entity that created a <*container*> in this collection is deleted, the <*container*> shall not be deleted.<br>This collection contains local <*container*> resources (representing local containers created by local or remote entities). |
| groups | Collection of <*group*> resources that do not have a containment relation with a specific remote entity (Application or SCL). This means that if the entity that created a <*group*> in this collection is deleted, the <*group*> resource shall not be deleted.<br>This collection contains local <*group*> resources (representing local groups created by local or remote entities). |
| accessRights | Collection of <*accessRight*> resources that do not have a containment relation with a specific remote entity (Application or SCL). This means that if the entity that created an <*accessRight*> in this collection is deleted, the <*accessRight*> shall not be deleted.<br>This collection contains local <*accessRight*> resources created by local or remote entities. |
| subscriptions | Collection containing all active subscriptions for the <*sclBase*> resource. |
| discovery | Resource used for resource discovery. |

As an example the *<sclBase>* contains the following attributes.

**Table 8.4**

| AttributeName | Description |
|---|---|
| accessRightID | See Common attributes. The default may be set by the system at creation. |
| searchStrings | See Common attributes. This attribute is only applicable on the registered-to SCL's *<sclBase>* resource.<br>These searchStrings can be used by the registering SCL when creating a *<scl>* resource representing the registered-to SCL.<br>This allows the registered-to SCL to be discovered using search strings when the discovery procedure is executed on the discovery resource in the registering SCL. |
| creationTime | See Common attributes. |
| lastModifiedTime | See Common attributes. |
| aPocHandling | The *aPocHandling* attribute controls how SCL retargeting shall be performed. It can have two value; SHALLOW or DEEP.<br>SHALLOW means that only exact or shallow prefix matches (1 level deep) to elements in the *aPoCPaths* attribute are retargeted.<br>DEEP means that any prefix match will result in retargeting.<br><br>If the *aPocHandling* attribute is not present, the SCL shall act the same as if the value was SHALLOW.<br><br>This attribute shall only be modified as part of the scl registration procedure (see [create <*scl*>) or, in case sclBase resource represents a GSCL or DSCL, it can be modified by an NSCL with which the GSCL or DSCL is registered. |

### 8.1.3.4.1 Resource scls

The scls resource is a collection resource that shall represent a collection of 0 or more *<scl>* resources.

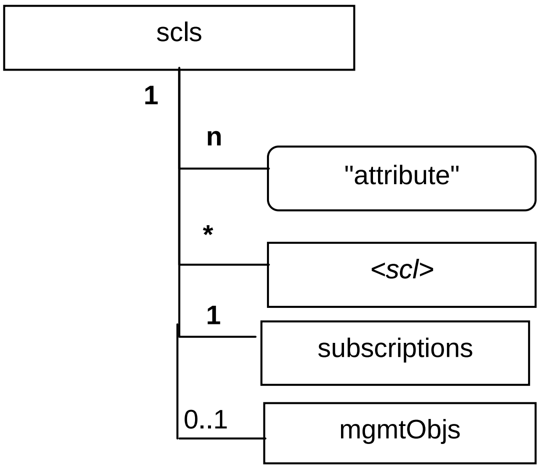


**Figure 8.6: Structure of scls resource**

### 8.1.3.4.2 Resource *<scl>*

An *<scl>* resource shall represent a remote SCL that is registered to the containing *<sclBase>*. This means that each remote SCL that is registered with the *<sclBase>* shall be represented by an *<scl>* resource in that *<sclBase>* (the registered remote SCL).

Conversely, each registered to SCL shall also be represented as a sub-set *<scl>* resource in the registering SCL's *<sclBase>*.

For example, when SCL1 registers with SCL2, there will be two *<scl>* resources created, one in SCL1, *<sclBase1>*/scls/*<scl2>* and one in SCL2: *<sclBase2>*/scls/*<scl1>*.

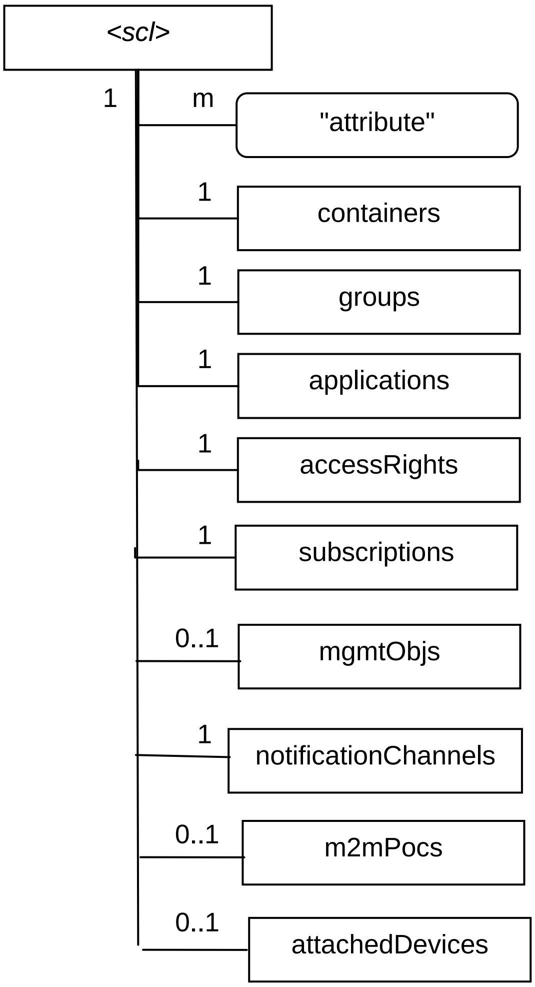


**Figure 8.7: Structure of <*scl*> resource**

As an example, the <*scl*> resource contains the following attributes.

**Table 8.5**

| AttributeName | Description |
|---|---|
| pocs | List of zero or more points of contact that can be used for out-of-band communication with an SCL.<br>This is only applicable for <*scl*> resources hosted on the NSCL, i.e. for <*scl*> resources that represent a registered DSCL or GSCL.<br>For <*scl*> resource hosted on the DSCL or GSCL, and therefore representing the NSCL, the value of the pocs is always the empty collection. |
| remTriggerAddr | Contains the "triggering address" of the remote entity represented by the M2M Device's or Gateway's <*scl*> registered resource with a hosting network <*sclBase*> for management purpose. It shall not be present in any registered network <*scl*> hosted in a DSCL or GSCL.<br>This attribute may be provided during the <*scl*> registration procedure or by other provisioning means including pre-configuration or bootstrapping.<br>The format of this attribute shall be a URI as specified by existing management protocols (e.g. a WAP/SIP/HTTP Push URI in OMA-DM [i.2], a HTTP URI in BBF TR-069 [i.3]).<br>For network-initiated management procedures, the NSCL shall send a triggering message (e.g. OMA-DM Notification, BBF TR-069 [i.3] Connection Request) to this address to request the remote entity to setup a management session with the NSCL. |

| AttributeName | Description |
|---|---|
| onlineStatus | Indicates if the SCL is reachable for M2M REST traffic.<br>For <*scl*> resource hosted on the NSCL, the value is set as described below.<br>The status is set by the hosting SCL based on the provided m2mPoc information and/or long polling activity.<br>If the <*scl*> resource contains at least one active (online) m2mPoc, then the onlineStatus of that SCL resource shall be set to ONLINE.<br>If the <*scl*> resource is currently involved in long-polling, the online status of that SCL resource shall be set to be ONLINE.<br>If there are no m2mPocs defined or if all m2mPocs are marked as OFFLINE and no long-polling is ongoing, then the onlineStatus of that SCL shall be set to OFFLINE.<br>If there are m2mPocs, and all of them are marked as NOT_REACHABLE, the onlineStatus of the SCL shall be set to NOT_REACHABLE to indicate that the SCL cannot be reached using any of the m2mPocs. NOT_REACHABLE can be regarded as a sub-state of ONLINE.<br>For an <scl> resource hosted on a G/DSCL the value shall be ONLINE |
| serverCapability | When set to TRUE it means that this SCL could try to issue connections towards the registered SCL.<br>This attribute is always set to TRUE for <*scl*> resources hosted on the DSCL or GSCL, i.e. for <*scl*> resources representing an NSCL (i.e. the registered-to SCL).<br>For <*scl*> resource hosted on the NSCL, the value of serverCapability is set to TRUE only if there are m2mPocs available. |
| Link | The URI of the <*sclBase*> of the remote SCL. |
| schedule | Represents the connection schedule of the remote / registered SCL. This is provided for information purposes only. The present document does not mandate any specific handling associated with the schedule.<br>Only applicable for registered SCLs, i.e. only in the NSCL's <*sclBase*> resource tree. |
| expirationTime | Common attributes. This represents the expiration time of the registration. If the SCL does not refresh its registration before that time the resource is deleted. |
| accessRightID | Common attributes. |
| searchStrings | Common attributes. |
| creationTime | Common attributes. |
| lastModifiedTime | Common attributes. |
| locTargetDevice | The device address to be used for retrieving the location information of the M2M Device or Gateway which is represented by this <*scl*>.<br>Only present for <*scl*> resource hosted on the NSCL.<br>This attribute is only used in the case that the location information is provided by a network-based location server (e.g. a 3GPP location server). It will be provided to the location server by the hosting SCL (i.e. NTOE) for the location information retrieval.<br>The format of this attributed shall conform to the interface provided by the location server (e.g. MSISDN for a 3GPP location server). |
| mgmtProtocolType | It is defined and used to store the management protocol that this <*scl*> supports.<br>Only applicable for <*scl*> resource hosted on the NSCL.<br>An M2M Device (or a M2M Gateway) shall indicate the mgmtProtocolType they support during the procedures such as SCL Registration, Update SCL Registration, and normal RESTful Update of this attribute. |
| integrityValResults | Indicates the signed Integrity Validation results for the registering SCL.<br>This attribute is optional and relates only to D/GSCL since Integrity Validation is not performed on the NSCL. |
| aPocHandling | The *aPocHandling* attribute as received during scl registration is the basis for *aPocHandling* attribute that is set on the <*sclBase*> resource, which in turn controls the SCL retargeting behaviour. |
| sclType | This attribute indicates the SCL type: NSCL, GSCL, or DSCL. |

### Resource accessRights

The accessRights resource represents a collection of <*accessRight*> resources and/or <*accessRightAnnc*> resources. The following combinations are possible:

- <*sclBase*>/accessRights - contains accessRight resources only (created by local or remote entities).
- <*sclBase*>/scls/<*scl*>/accessRights - contains accessRightAnnc resources announced by <*scl*> and/or accessRight resources.
- <*sclBase*>/applications/<*app*>/accessRights - contains local accessRight resource only, typically created by the Application corresponding to <*app*>.

- *<sclBase>*/scls/*<scl>*/applications/*<applicationAnnc>*/accessRights - contains accessRightAnnc resource announced by the SCL corresponding to *<scl>* and/or accessRight resource typically created by the SCL corresponding to *<scl>* or the Application on whose behalf *<applicationAnnc>* is created.

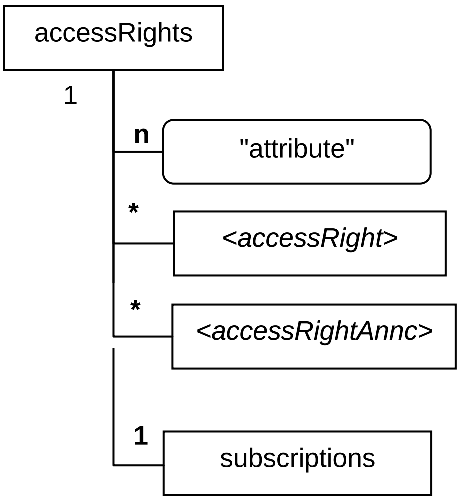


**Figure 8.8: Structure of accessRights-resource**

**Resource *<accessRight>***

Access rights are defined as "white lists" or permissions, i.e. each permission defines "allowed" entities (defined in the permissionHolders) for certain access modes (permissionFlags). Sets of permissions are handled such that the resulting permissions for a group of permissions are the sum of the individual permissions. I.e. an action is permitted if it is permitted by some / any permission in the set.

By setting an accessRightID attribute on a resource, the permissions for accessing that resource are then defined by the permissions defined in the accessRight resource.

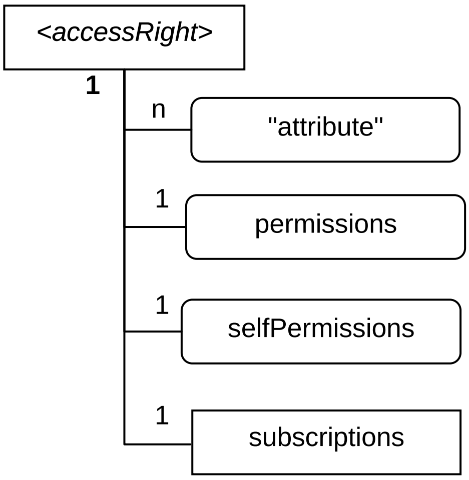


**Figure 8.9: Structure of *<accessRight>* resource**

The <*accessRight*> resource contains attributes such as:

**Table 8.6**

| AttributeName | Description |
|---|---|
| permissions | The collection of permissions defined by this <*accessRight*>. These permissions are applied to resources referencing this accessRight resource using the *accessRightID* attribute. |
| selfPermissions | Defines the collection of permissions for the <*accessRight*> resource itself. |

The permissionFlags and the permissionHolders could then be generalised to actions (which might be granting access, but might also be more specific, like granting access to a subset, i.e. filtering part of the data). The permissionHolders could be generalised to conditions, which may include things like the identity of the requestor, everybody except specified identities, but it might also include time based conditions, etc.

**Resource containers**

The containers resource represents a collection of container resources and containerAnnc resources. The following combinations are possible:

<*sclBase*>/containers - can contain the following type of resources

- container resources only (either created by local or remote entities).

<*sclBase*>/scls/<*scl*>/containers - can contain a mix of the following resources

- containerAnnc resources announced by the SCL corresponding to <*scl*>.
- container resources.

<*sclBase*>/applications/<*app*>/containers - can contain a mix of the following type of resources

- container resources, typically created by the Application corresponding to <*app*>.
- locationContainer, typically created by the Application corresponding to <*app*>.

<*sclBase*>/scls/<*scl*>/applications/<*applicationAnnc*>/containers - can contain a mix of the following type of resources

- <*containerAnnc*> resources announced by the SCL corresponding to <*scl*>.
- <*locationContainerAnnc*> resources announced by the SCL corresponding to <*scl*>.
- <*container*> resources typically created by the SCL corresponding to <*scl*> or corresponding to the Application on whose behalf <*applicationAnnc*> is created.
- <*locationContainer*> resources typically created by the SCL corresponding to <*scl*> or corresponding to the Application on whose behalf <*applicationAnnc*> is created.

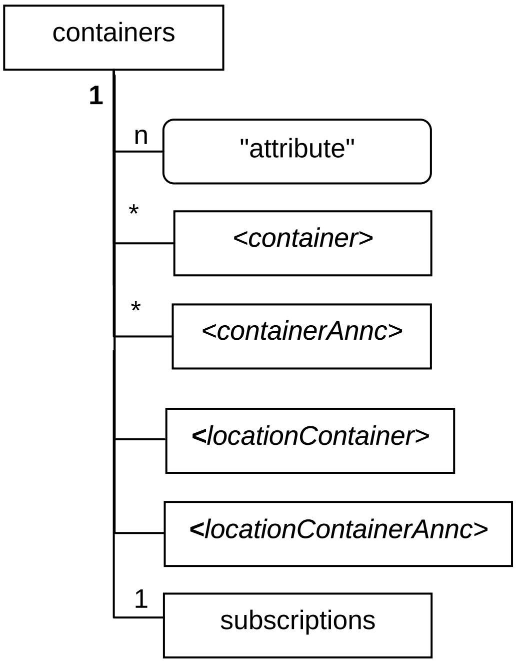


**Figure 8.10: Structure of containers-resource**

**Resource <*container*>**

The <*container*> resource represents a container for instances.

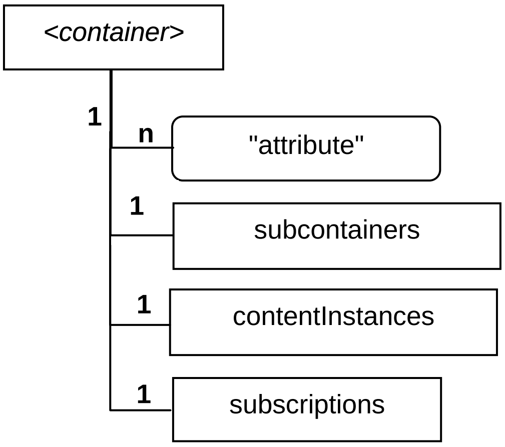


**Figure 8.11: Structure of <*container*> resource**

**Resource contentInstances**

The *contentInstances* resource represents the collection of content instances in a container. It shall also keep track of the latest (newest) and the oldest instance. These shall reference the newest instance and the oldest instance in the collection, respectively. If there are no instances in the collection, the latest and oldest resources shall not be present.

The *contentInstances* resource is special in that a retrieve on the *contentInstances* shall give in the returned resource representation the content of the <*contentInstance*> resources in the collection (subject to the filter criteria) and not just the references to these child resources. When a retrieve is performed on the *contentInstances* resource it shall be possible to indicate that either only the meta-data of the *contentInstance* resources in the collection matching the filter criteria shall be returned or whether both the meta-data together with the actual content of each *contentInstance* resource matching the filtercriteria shall be returned.

The *contentInstances* resource does not have its own AccessRights. Instead the accessRights of the parent <*container*> resource shall apply.

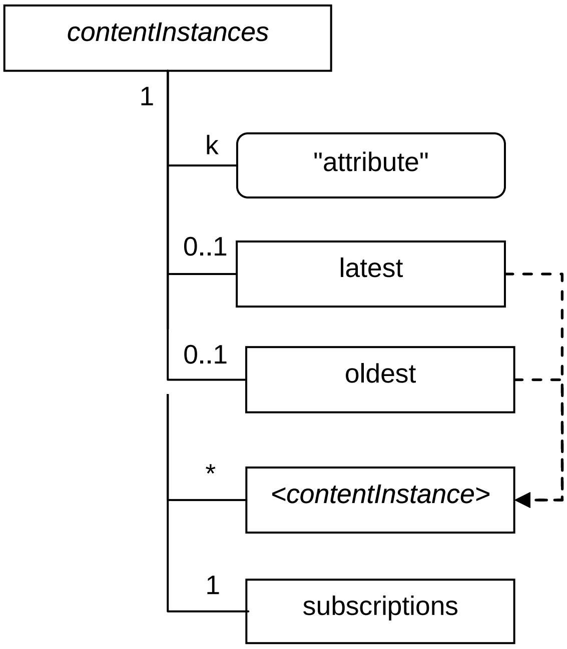


**Figure 8.12 Structure of *contentInstances* resource**

**Resource <*contentInstance*>**

The <*contentInstance*> resource represents a data instance in the container. The content of the instance is opaque to the M2M platform and it might even be encrypted. However, there is meta-data associated with an instance which shall be accessible.

Contrary to other resources, the <*contentInstance*> resource cannot be modified once created, regardless of the accessRightID associated with the parent resource. An instance may be deleted explicitly or it may be deleted by the platform based on policies. If the platform has policies for the instance retention these shall be represented by the attributes maxByteSize, maxNrOfInstances and/or maxInstanceAge on the <*container*> resource. If multiple policies are in effect, the strictest policy shall apply.

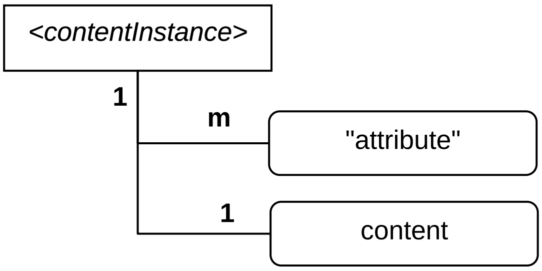


**Figure 8.13: Structure of <*contentInstance*> resource**

## 8.2 Security

Data origin authentication, integrity and replay protection, confidentiality and privacy must be supported on the mId Reference Point (between the Device/Gateway Domain and the Network Domain). However in some cases it is possible to rely on the Access Network to provide some of these services. Therefore, security at the M2M service layer consists of a set of optional functionalities. It comprises the following functionality:

- M2M Service Bootstrap procedures for secure provisioning of (secret) M2M Root Key in M2M Device/Gateway and M2M Authentication Server.

- M2M Service Connection procedures, based on a provisioned M2M Root Key, for mutual authentication, authorization and secure session establishment.

- Integrity Validation reporting.

- Secured Environment in M2M Device/Gateway (for secure storage of Sensitive Data and secure execution of Sensitive Functions used by the M2M service layer).

M2M Service Bootstrap procedures may be access network assisted (using 3GPP GBA-based or EAP-based methods) or access network independent (using EAP-over-PANA or TLS-over-TCP methods). M2M Service Connection procedures may be based on EAP/PANA, TLS-PSK or 3GPP GBA.

## 8.3 Management

Remote Entity Management (REM) service capability shall be supported by all M2M SCLs. Both OMA-DM [i.2] and BBF TR-069 [i.3] are considered as the supporting enablers to be reused in ETSI M2M functional architecture to implement the REM service capability. An M2M system may choose to implement either or both of them. Other device management enablers may also be reused in a similar manner, but the details are out of scope.

The mIa reference point shall support the RESTful interface procedures to allow the NREM to handle the request from M2M Network Applications for the purpose of the remote entity management.

The mId reference point shall support the RESTful interface procedures to allow the D/GREM to create *<mgmtObj>* and/or *<mgmtCmd>* resources in the NREM for the purpose of the remote entity management thereafter. It shall also support corresponding OMA-DM [i.2]/TR-069 [i.3] protocol interfaces and procedures for managing M2M Devices/Gateways enabled by OMA-DM [i.2]/TR-069 [i.3].

A *<mgmtObj>* or *<mgmtCmd>* resource in the NSCL represents either:

- high-level management functionalities (e.g. ETSI M2M specific data model) which shall be supported by the underlying Management Object(s) on the remote entity; or

- low-level functionalities on the remote entity mapped from the data model as specified by existing device management technologies (e.g. OMA-DM, BBF TR-069 [i.3]).

Through the manipulation of *<mgmtObj>* or *<mgmtCmd>* resources, ETSI M2M REM supports the following management functions at different layers of remote entities:

- M2M application lifecycle management: installing, removing and upgrading applications in an M2M Device/M2M Gateway.

- M2M service management: configuration management for the M2M Service Capabilities in the M2M Device/M2M Gateway.

- M2M Area Network management: configuration management for the M2M Area Networks (namely Capillary Network).

- M2M device management: configuration management of the M2M Device/M2M Gateway.

### 8.3.1 Managing M2M Device and M2M Gateway

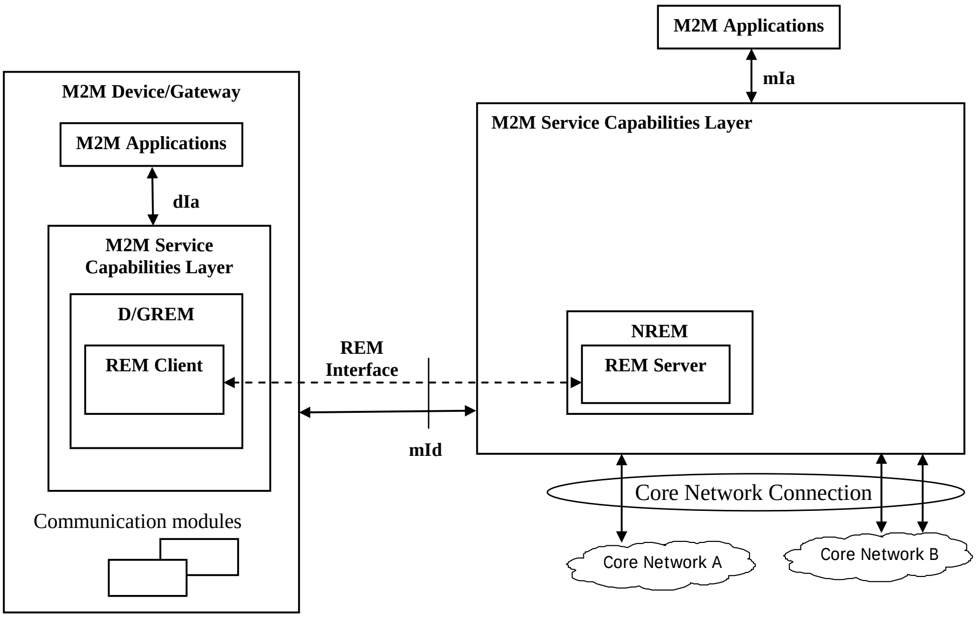


**Figure 8.14: Reference Architecture for managing M2M Device/Gateway**

As shown in figure 8.14 the Device Management Client is integrated as a part of the D/GREM Service Capability, while the Device Management Server is integrated as a part of the NREM Service Capability.

NOTE 1: Alternatively to REM Server being integrated as a part of the NREM a REM Server may be external to the NREM but interface with the NREM via an implementation-specific interface exposed by the REM Server.

NOTE 2: When ETSI M2M architecture is deployed over 3GPP network, the OMA-DM Server in the 3GPP network layer may be integrated for the purpose of remote management.

Table 8.7 shows the mapping between ETSI M2M entities/interface to OMA-DM/TR-069 [i.3] entities/interface.

**Table 8.7: Mapping between ETSI M2M and OMA-DM/TR-069 entities**

| ETSI M2M | OMA DM | BBF TR069 |
|---|---|---|
| REM Server | DM Server | ACS |
| REM Client | DM Client | CPE |
| REM Interface (over mld) | DM-1, DM2 | TR069-CWMP |

On the M2M Device and M2M Gateway side:

- The D/GREM may collect the Management Object data from the REM Client in the local M2M Device/Gateway and create/update the corresponding <*mgmtObj*> and/or <*mgmtCmd*> resource(s) in the NSCL via the mId reference point.

- Any device management activity (e.g. firmware update, fault management) in the Device/Gateway is carried out by the REM Client, communicating with a REM Server in NREM via existing Device Management interfaces.

On the NSCL side:

- The NREM triggers OMA-DM or BBF TR-069 Device Management procedures over mId resulting from actions on the <*mgmtObj*> or <*mgmtCmd*> resources by M2M Network Applications via mIa or by M2M Management Functions.

### 8.3.2 Managing M2M Devices behind an M2M Gateway

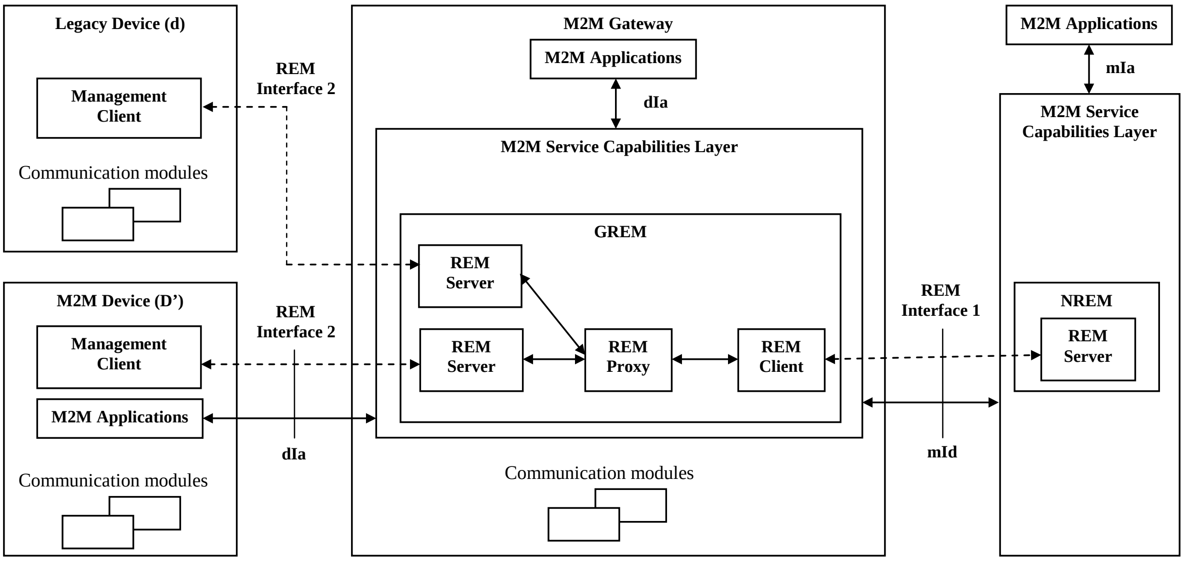


**Figure 8.15: Reference Architecture for Managing devices behind M2M Gateway**

Figure 8.15 illustrates the integrated reference architecture for managing devices behind M2M Gateway, which may be legacy devices (d-type) or limited M2M Devices (D'-type) without M2M Service Capabilities and may not support OMA-DM or BBF TR069. In order to manage devices behind M2M Gateway, GREM needs to have Server, Client and Proxy function and NREM may need additional feature in order to support the proxy function at GREM.

Table 8.8 shows the mapping between ETSI M2M entities/interface to OMA-DM/TR-069 entities/interface.

**Table 8.8: Mapping between ETSI M2M and OMA-DM/TR-069 entities/interface**

| ETSI M2M | OMA DM | BBF TR069 |
|---|---|---|
| REM Server N) | DM Server | ACS |
| REM Server (G) | Legacy Server | Control Point |
| REM Proxy | GwMO | Proxy Module |
| REMP Client | DM Client | CPE |
| Management Client | Legacy Client | Legacy Client |
| REM Interface 1 (over mld) | DM-1, DM-2 | TR069-CWMP |
| REM Interface 2 (over dla) | Legacy Interface (out of scope) | Legacy Interface (out of scope) |

In addition to support the functionalities described in clause 8.5.1 for managing the M2M Gateway itself, the GREM shall also be responsible for the management of the D'(or d)-type devices associated with the M2M Gateway. The interactions between the GREM and the D'(or d)-type device for device management purpose is out of scope.

The NREM acts as the same role and follows the same procedure as described in clause 8.3.1 to interact with GREM for the purpose of managing devices in the M2M Area Network, i.e. behind an M2M Gateway.

# 9 M2M Architecture Description - TIA

This clause describes the M2M Architecture as documented by the TIA TR-50 Committee.

## 9.1 TIA TR-50 Functional architecture

This clause provides a snap-shot of the TIA TR-50 M2M Smart Device Communication System Architecture, and a description of the reference points. The detailed Architecture descriptions can be found in TIA PN-4940.005 [i.6].

**Terms and Definitions:**

- **AAA-SD:** provide authentication, authorization and accounting services to other entities in the network to establish and enforce security policies. The services may include generation of keys, generation and validation of certificates, validation of signatures, etc.
- NOTE: This term was intended to be used as "AAA for Smart Devices". The scope was, indeed, may be broader than the AAA, which has been commonly used in the wireless network. The term was left unchanged as TIA TR-50 has not officially defined this entity although a report on "Threat Analysis" has been developed (TIA PN-4940.005) as the first step for this work.
- **Home Application:** The home application is a logical entity that is responsible for the business logic, either directly or via supervision and interaction with node applications and PoA applications and with PoA devices.
- **Node Application:** The node application is a logical entity that acts as an intermediary between the home application and the PoA application and between the home application and the PoA device. The node application interacts with home application, other node applications, PoA application or PoA device, and may perform functions such as a data aggregation, storage, load balancing, etc.
- **PoA Application:** PoA application is a logical entity that provides resources to node or home applications or to other PoA applications. The PoA application interacts with home, node, other PoA applications or with PoA devices. The PoA application may perform functions such as autonomous reporting of values reported by devices, monitoring for values reported by devices that exceed specified limits, trend analysis of values reported by devices, etc.
- **Container:** The container is a logical entity that provides services to the applications that operate within it, and enforce security policies.
-

### 9.1.1 TIA TR-50 M2M Smart Device Communications System Architecture

#### 9.1.1.1 Functions and reference points

The TR-50 System Architecture pertains to the access agnostic monitoring and bi-directional communication of events and information between smart devices and other devices, applications or networks. Layers in the protocol stack at and below the transport layer are assumed to exist (including but not limited to TCP/IP, UDP/IP, HTTP, HTTPS, DHCP, Diff-Serv, MPLS, XMPP) and their descriptions are beyond the scope of this document.

To maintain a consistent interface to the transport layers (over fixed-point wireless, over wireless local area network, over digital subscriber line, etc.) a convergence layer is introduced into the protocol stack, as illustrated in figure 9.1. (The dotted lines in the node indicate optional capability.)

PoA, node and server are considered above the access networks.

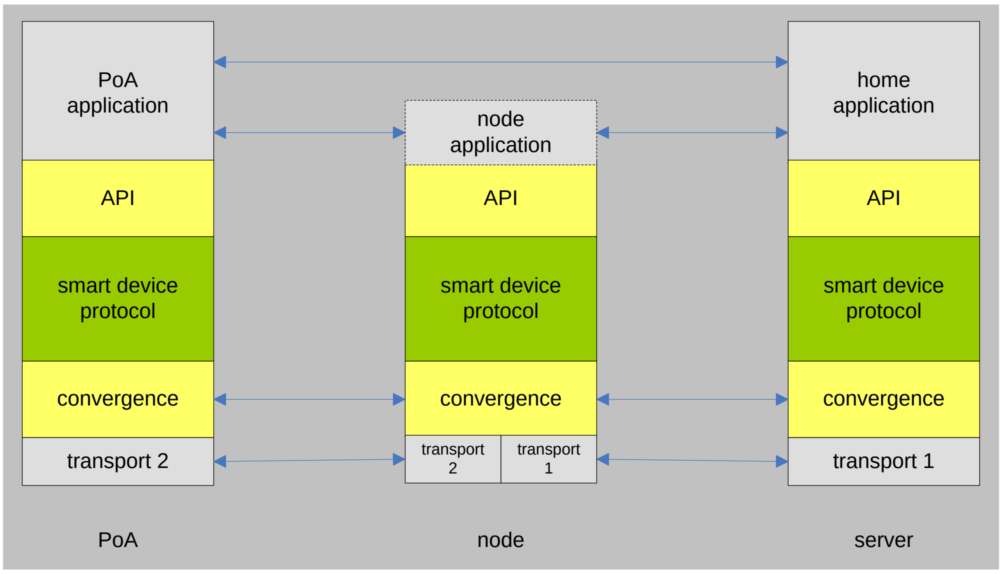


**Figure 9.1: TR-50 Conceptual Protocol Stack**

Figure 9.2 depicts the high level system architecture for the TIA TR-50 M2M Smart Device Communication.

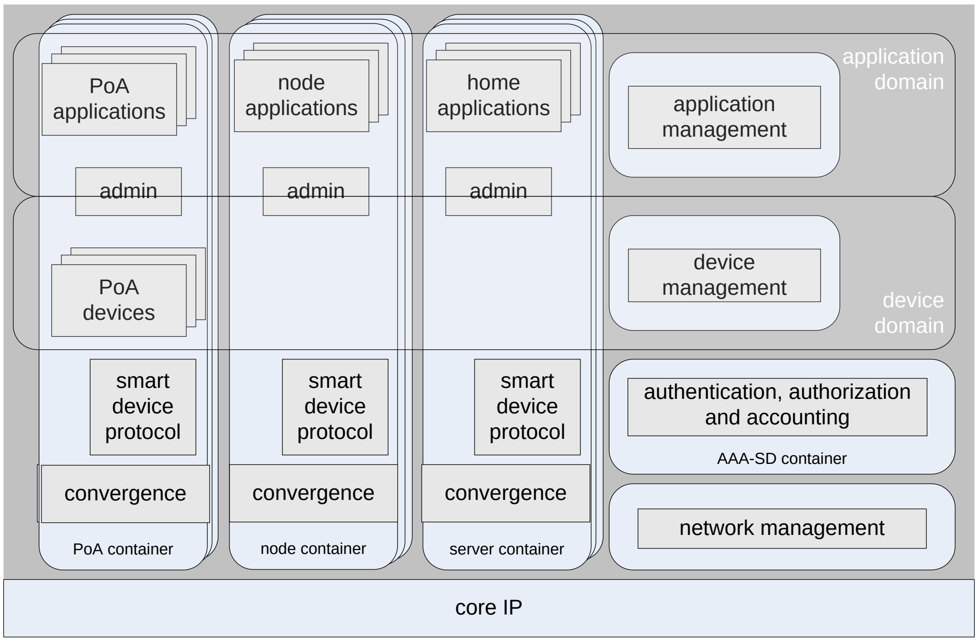


**Figure 9.2: TR-50 High Level System Architecture**

The high level system architecture shown above may be described as a distributed cooperative computing system. The container provides services to the application(s) that operate within it, and enforce security policies.

In figure 9.2, some containers are labelled, **PoA container**, **node container** and **server container**. The labels are for ease of reference and imply some level of logical grouping.

Some containers are not labelled implying that the entities within them can operate in any convenient appropriate container.

## 9.2 Overview

### 9.2.1 Reference Architecture Diagram

Figure 9.3 depicts the TIA TR-50 reference architecture diagram, showing functional elements, and the interconnection reference points. Light blue boxes represent containers while light yellow boxes represent applications and/or devices.

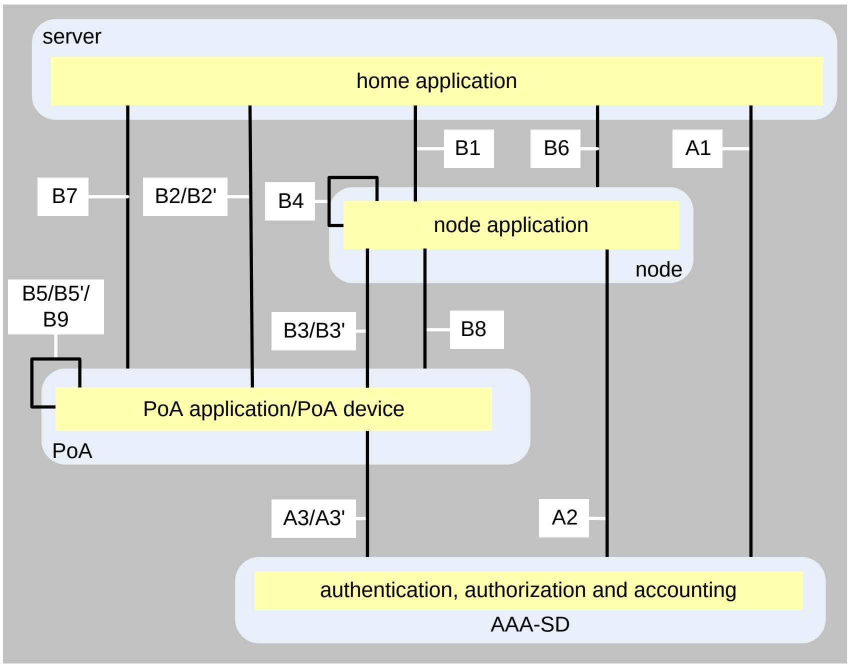


**Figure 9.3: Reference architecture**

### 9.2.2 Functional Elements

**AAA-SD**

- A container that hosts an entity or entities that provide authentication, authorization and accounting services and interact with the admin entities in other containers in establishing and enforcing security policies.

**home application**

- The *home application* is responsible for the business logic, either directly or via supervision and interaction with node and *PoA applications* and with *PoA devices*.

**node application**

- The *node application* acts as an intermediary between the *home application* and the *PoA application* and between the *home application* and the *PoA device*. It interacts with *home application*, *PoA application*, and *PoA device*, and may perform functions such as data aggregation, and load balancing.

**PoA application**

- PoA application provides resources to node application, home applications or to other PoA applications. PoA applications interact with home application, node applications, PoA applications and with PoA devices, and may perform functions such as autonomous reporting of values reported by devices, monitoring for values reported by devices that exceed specified limits, trend analysis of values reported by devices, etc.

- *PoA applications* may be instances of a standardized class to facilitate their invocation by, for example, specifying parameters for their operation. For the purposes of illustration, consider the following:

  - stream: an instance to this class autonomously reads data from a specified source and streams it to a specified target according to some specified criteria.

- average: an instance to this class autonomously reads data from a specified source, computes the average according to some specified criteria, and reports the result to a specified target according to some specified criteria.
- limit: an instance to this class autonomously reads data from a specified source, compares the data with some specified limits, and reports to a specified target if the limits are exceeded.
- trend: an instance to this class autonomously reads data from a specified source, computes the trend according to some specified criteria, and reports to a specified target if the computed trend exceeded some specified criteria.

**PoA device**

A *PoA device* is a resource that represents a physical device. The means by which physical devices are interfaced are outside the scope of this document. The combination of services provided by the PoA container and the software that implements a *PoA device* are responsible for whatever conversion is necessary to represent the physical device as a standardized resource.

### 9.2.3 Reference Points

**A1:** provides for interaction between the AAA-SD container and the *home application*.

**A2:** provides for interaction between the AAA-SD container and the *node application*.

**A3:** provides for interaction between the AAA-SD container and the *PoA application*.

**A3':** provides for interaction between the AAA-SD container and the *PoA device*.

- The realization of A3 and A3' may be identical.
- The realization of A1, A2, A3 and A3' may be identical.

**B1:** provides for interaction between the *home application* and *a node application*, including bi-directional communication of control information, events and data.

**B2:** provides for interaction between a *PoA application* and the *home application*, including bi-directional communication of control information, events and data.

**B2':** provides for interaction between a *PoA device* and the *home application*, including bi-directional communication of control information, events and data.

- The realization of B2 and B2' may be identical.

**B3:** provides for interaction between a *PoA application* and a *node application*, including bi-directional communication of control information, events and data.

**B3':** provides for interaction between a *PoA device* and a *node application*, including bi-directional communication of control information, events and data.

- The realization of B3 and B3' may be identical.

**B4:** provides for interaction between the different *node applications*, possibly in different containers, including bi-directional communication of control information, events and data.

- The realization of B1 and B4 may be identical.

**B5:** provides for interaction between the different *PoA applications*, possibly in different containers, including bi-directional communication of control information, events and data.

**B5':** provides for interaction between the different *PoA devices*, possibly in different containers, including bi-directional communication of control information, events and data.

- The realization of B5 and B5' may be identical.
- The realization of B2, B2', B3, and B3' may be identical.

**B6:** provides for interaction between the *home applications* and a *node container*, including bi-directional communication of control information, events and data.

**B7:** provides for interaction between the *home application* and a *PoA container*, including bi-directional communication of control information, events and data.

**B8:** provides for interaction between *node applications* and a *PoA container*, including bi-directional communication of control information, events and data.

**B9:** provides for interaction between a *PoA application* and a *PoA device*, including bi-directional communication of control information, events and data.

## 9.3 Security Considerations

TIA TR-50 has started exploiting possible threats and the need for further standardization for AAA-SD. TIA plans to publish a Technical System Bulletin (TSB), PN-4940 "Security Aspects", to provide threat analysis.

TIA TR-50 architecture currently relies on Transport Layer Security (TLS) to provide the communication security over internet.

# 10 Open M2M Architecture Description - TTA

This clause describes the Open M2M Architecture as documented by TTA Project Group 708 (PG708).

## 10.1 Overview of Open M2M architecture

This clause provides an overview of Open M2M architecture. The architecture supports Resource registration and discovery process in global environments. High-level model of the Open M2M architecture is shown in figure 10.1.

### 10.1.1 Terms and Definitions

General terms and definitions are as follows:

- **Resource:** In Open M2M architecture, Resource is any physical or virtual component that provides useful data to users. Resource can include a device, an application, a topic, content, a context, a service, etc.
- **Topic:** Topic is a group related to a common subject of interest. Resources registered to a topic can be managed by using the topic.
- **Discovery domain:** Discovery domain comprises device directory, topic directory, and application directory. Each directory keeps a profile about resources located in resource domain and provides device-, topic-, application-based discovery functions.
- **Device directory:** Device directory is a component of discovery domain. It provides device registration, device management, device authentication, and device discovery.
- **Topic directory:** Topic directory is a component of discovery domain. It provides topic registration, topic-based device discovery, and topic-based device/user management.
- **Application directory:** Application directory is a component of discovery domain, which provides application registration and discovery functions. It also offers mapping information between application and device's model.
- **Resource domain:** Resource domain is composed of resource server, resource gateway, and resource endpoint. This domain represents a group of physical entities in which resource data is located.
- **Resource DB:** Resource DB is an organized collection of resources. Resources are typically generated by a resource server, resource gateway, or resource endpoint. Resource DB is located in physical entities such as server, gateway, and endpoint.

### 10.1.2 Open M2M architecture Components

In Open M2M architecture, Discovery domain provides a process of resource (topic, device, or application) registration and discovery in global environments. Discovery domain keeps metadata profile related to real resources located in Resource domain. Discovery domain is composed of Device directory, Topic directory, and Application directory.

Device directory supports device registration and discovery process. Topic Directory enables each device to be managed and accessed by using topic (for example, topic-based device registration & discovery and topic-based device access control). Application Directory provides a function of application registration and discovery.

Resource Domain is composed of Resource Server, Resource Gateway, Resource End-Point, and Application Repository. Each Resource End-Point is connected to either Resource Gateway or Resource Server according to its communication capability. Resource End-Point can also be connected to a user directly. Application Repository contains Service Application and Service Web. Resource Domain interacts with Discovery Domain through Interaction Domain.

Resource Server, Resource Gateway, and Resource End-Point have a Resource DB containing an organized collection of resource data or linked pointer to other Resource DBs. Resource Server can manage resource data of Resource Gateway or Resource End-Point registered to the Resource Server. Resource Gateway can manage resource data of Resource End-Points registered to the Resource Gateway. Resource End-point can manage resource data of sensors or actuators registered to itself.

Each user can access resource data through Interaction Domain. To access resource data, user searches metadata profiles in Device directory, Topic Directory, or Application Directory. Through Interaction Domain, the user can access the resource data regardless of the physical or virtual location the resource exists.

In Open M2M architecture, there are two types of communication methods to access resource data. In the indirect method, user can access the resource in Resource End-Point via Resource Server or Resource Gateway. In contrast, in the direct method, user can access the resource located in Resource End-Point through a direct peer-to-peer connection between user and Resource End-Point.

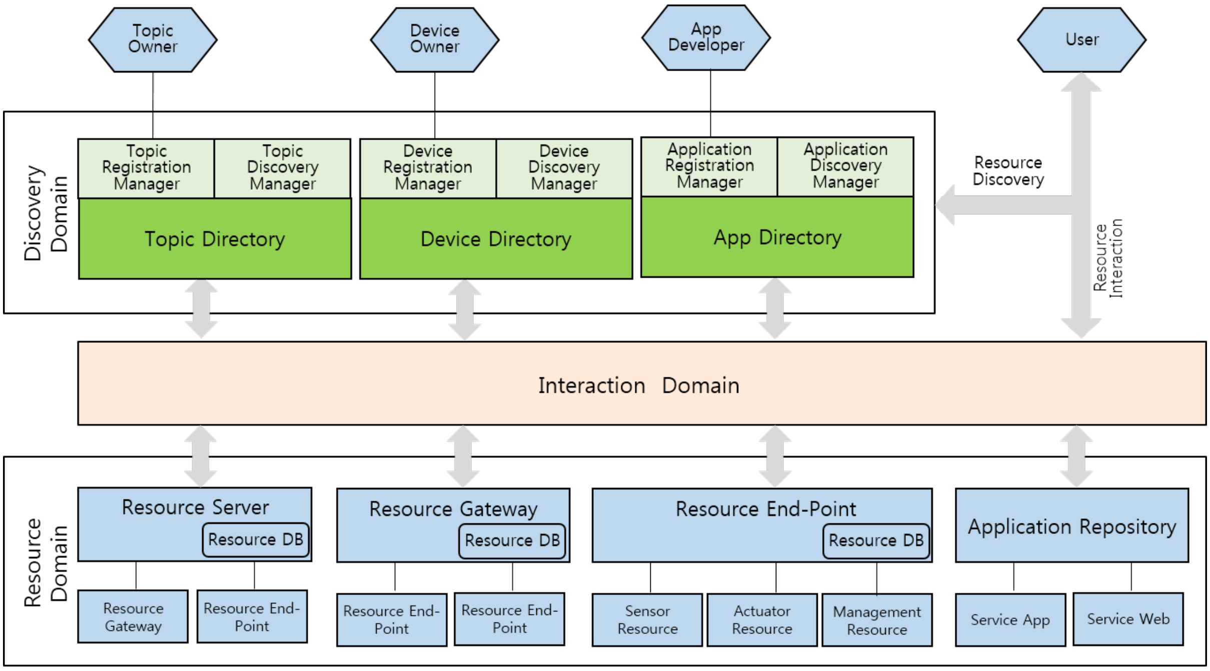


**Figure 10.1: TTA Open M2M Architecture**

## 10.2 Open M2M Architecture

### 10.2.1 Discovery Domain

In Open M2M architecture, Discovery domain provides a process of resource (topic, device, or application) registration and discovery in global-scale. Discovery domain keeps metadata profiles related to real resources located in Resource domain. The Discovery Domain is shown in figure 10.2.

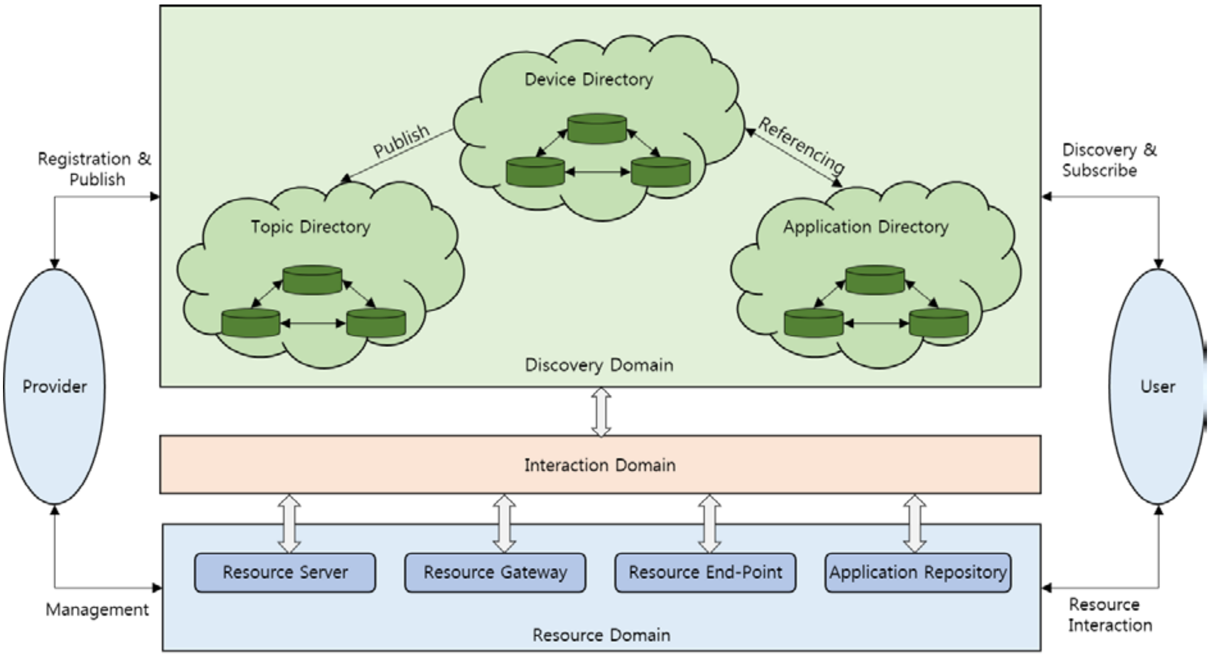


**Figure 10.2: Discovery Domain**

- Discovery domain is composed of Device directory, Topic directory, and Application directory.
- Discovery domain maintains metadata profile about resources located in resource domain. Metadata profile is an abstracted description about actual resource in Resource domain, and it provides information about resources.
- Providers can register resources (topic, device, or application) to Discovery domain.
- Users can globally search and discover resource data related to topic, device, or application in Discovery domain.
- Device metadata profile (abstracted description about devices located in Resource domain) in Device directory can be published to Topic directory.
- Device metadata profile in Device directory and Application metadata profile (abstracted description about applications located in application repository in Resource domain) in Application directory make reference to each other:
  - When a user discovers an application in Application directory, he or she can find devices which are referenced to the application.
  - When a user discovers a device in Device directory, he or she can find applications which are referenced to the device.
- Resource domain is composed of resource server, resource gateway, resource end-point, and application repository.
- Users can access resources in Resource domain through Interaction domain after discovering real resource-related metadata profile in Discovery domain.

### 10.2.2 Topic Directory

This clause describes the concept of a resource publication and subscription to a topic and a topic-based resource access control in Open M2M architecture. The Topic Directory is shown in figure 10.3.

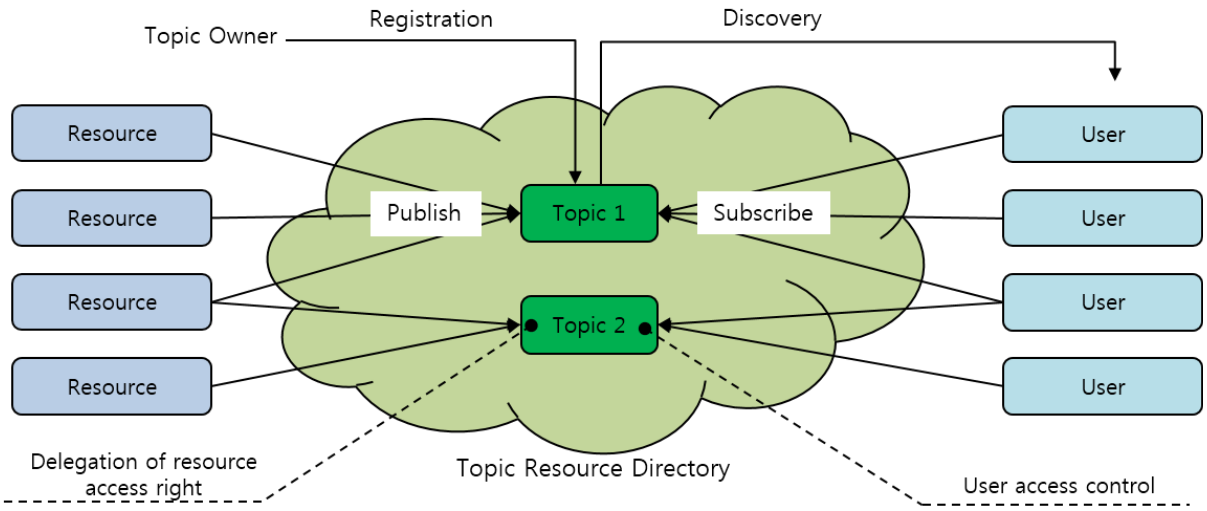


**Figure 10.3: Topic Directory**

- Topic owner can create a topic and register a topic located in Topic directory.
- Resources can be published to a topic located in Topic directory.
- Each user can search and discover a topic located in Topic directory.
- Each user can subscribe to a specific topic located in Topic directory and get resources associated with the topic.
- When resources are published to a topic, the access right of the published resources are delegated to the topic.
- When a user subscribes to a specific topic, the user can acquire the access right of the resources associated with the specific topic.

### 10.2.3 The Structure of M2M Resource DB

This clause describes the skeleton of Resource Domain and Resource DB in Open M2M Architecture. The Resource Domain is shown in figure 10.4.

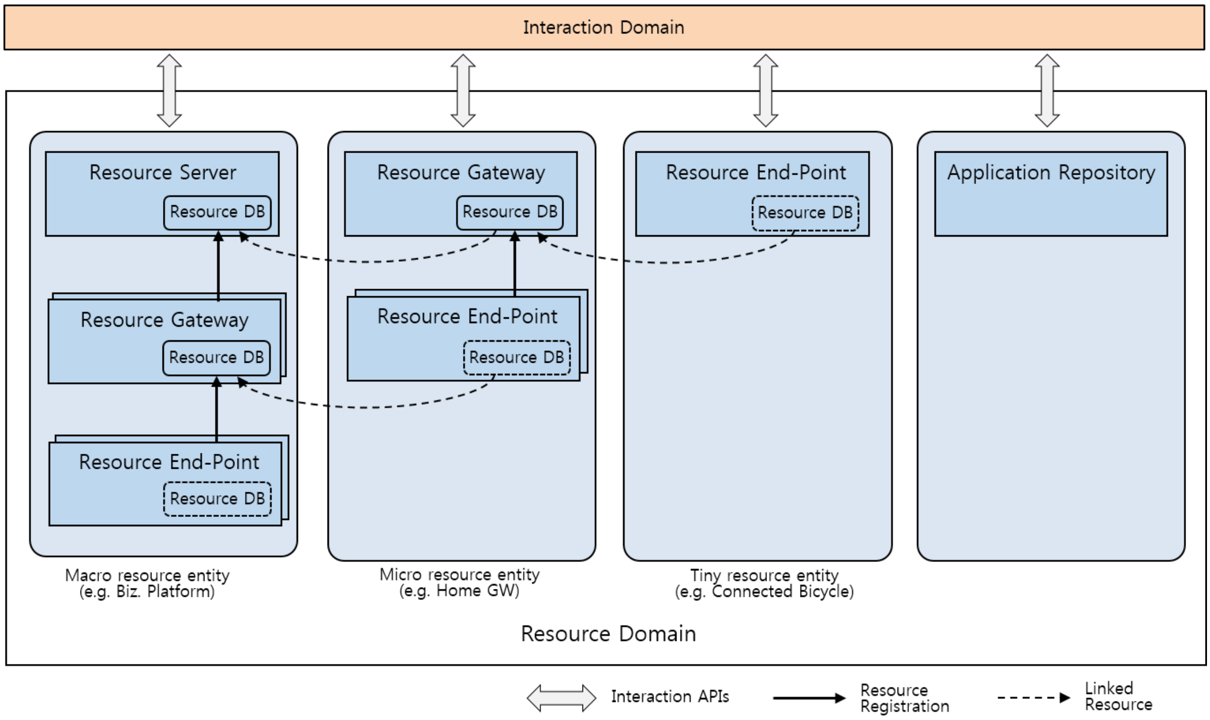


**Figure 10.4: Resource Domain**

- In Resource domain, there are three types of resource entity models: Macro resource entity, Micro resource entity, and Tiny resource entity.
- Macro resource entity contains Resource Server, Resource Gateway, and Resource End-point. SCLs-based model in ETSI standard can be considered as a type of this Macro resource entity. Through Resource Server, user can access resources in Resource Gateway or Resource End-point.
- Micro resource entity comprises Resource Gateway and Resource End-point. Resource Gateway can provide a resource access point without Resource Server in Macro resource entity.
- Tiny resource entity has Resource End-point. Resource End-Point can provide a resource access point without Resource Server and Resource Gateway.
- Resource Server, Resource Gateway, and Resource End-Point have Resource DB, but legacy devices might not have Resource DB.
- Each Resource DB can contain a collection of resources and linked pointers to other Resource DBs.
- Interaction Domain provides users with a way to access resources in Resource End-Point, Resource Gateway, and Resource Server.
- Resources in Resource Domain can be accessed through Interaction APIs provided by the resource entity.
- Open M2M architecture supports direct machine-to-machine P2P communication.

## 10.3 Open M2M Architecture Diagram

### 10.3.1 Functional Elements

This clause describes functional elements of Open M2M architecture. Open M2M Functional Elements are shown in figure 10.5.

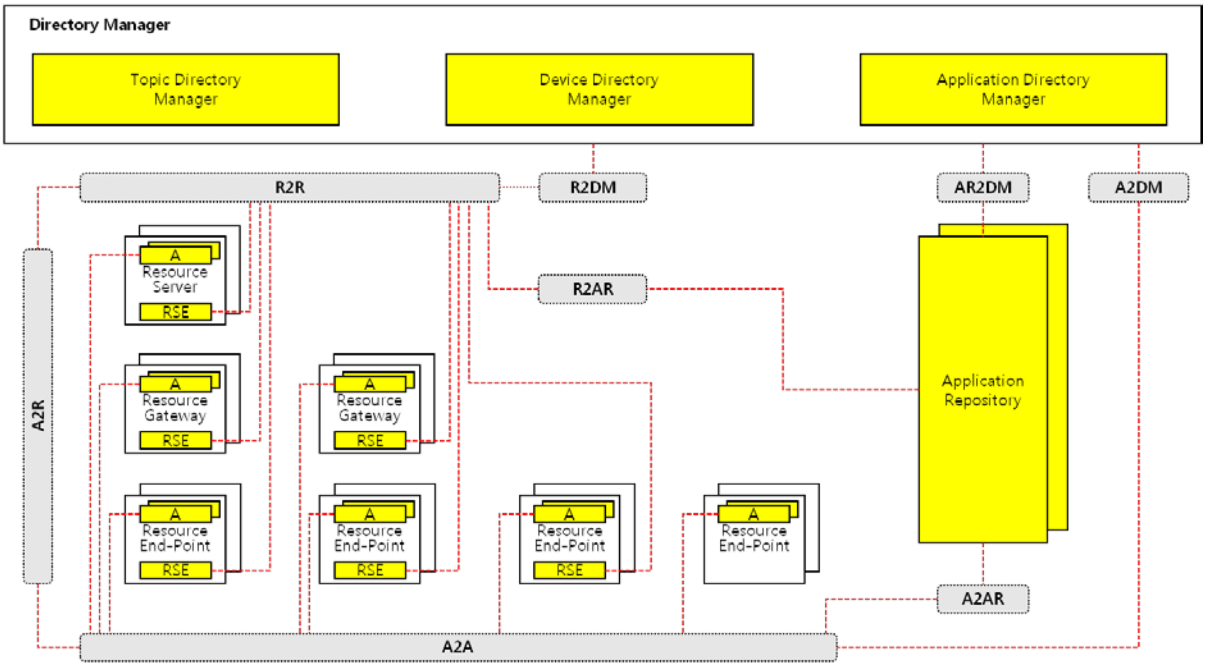


**Figure 10.5: Functional Elements of Open M2M Architecture**

**Application (A)**

- Application is a software program which provides users with useful data based on devices resources and enables them to control devices. Application runs on end-point, gateway, or server.

**Resource Service Entity (RSE)**

- Resource Service Entity is a functional module located in a device. It is responsible for device resource management and offers service functions such as network, security, registration/discovery, subscription, billing, etc. for an application, another resource entity, a directory manager, or an application repository.

**Application Repository (AR)**

- Application Repository is a storage designed to be able to upload and download applications. It manages application descriptions and registers its application descriptions to Application Directory Manager.

**Directory Manager (DM)**

- Directory Manager includes Device Directory Manager, Topic Directory Manager, and Application Directory Manager. Each directory manager provides device-, topic-, application-based registration/discovery and topic-based publish/subscribe functions.

### 10.3.2 Reference Points

This clause describes reference points of Open M2M architecture. Open M2M Reference Points are shown in figure 10.6.

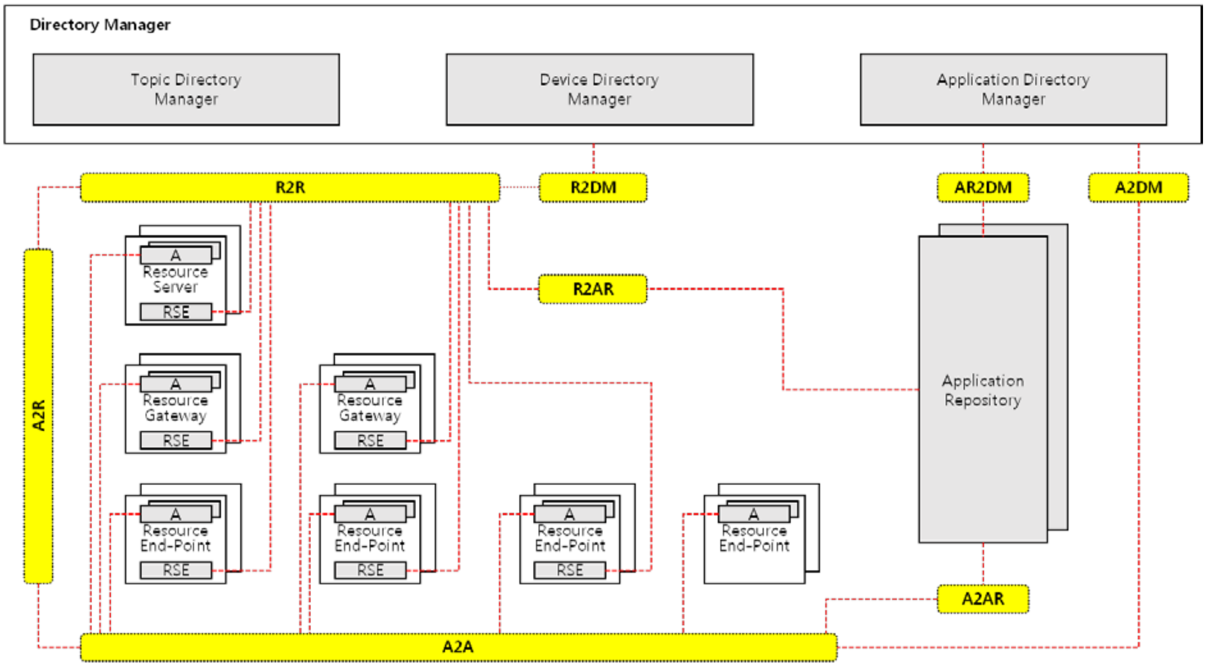


**Figure 10.6: Reference Points of Open M2M Architecture**

**R2R**

- R2R reference point offers generic functions for interaction between a resource service entity and another resource service entity.
- The R2R reference point, functions include:
  - Registration between Resource Service Entities.
  - To get/put/post/delete data between Resource Service Entities.
  - Request/response for Resource Service Entity functions such as data-, network-, security-, group-management, etc. between Resource Service Entities.

**A2A**

- A2A reference point offers generic functions for interaction between an application and another application.
- The A2A reference point, functions include:
  - Common functions between Applications.
  - Remote method invocation between Applications.

**A2R**

- A2R reference point offers generic functions for interaction between an application and a resource service entity.
- The A2R reference point, functions include:
  - Registration of Application to Resource Service Entity.
  - To get/put/post/delete data between Application and Resource Service Entity.

- Request/response for Resource Service Entity functions such as data-, network-, security-, group-management, etc. between Application and Resource Service Entity.

**R2DM**

- R2DM reference point offers generic functions for interaction between a resource service entity and a directory manager.
- The R2DM reference point, functions include:
    - Registration of device descriptions (ID, name, location, keyword, network address, etc.) to Device Directory Manager.
    - Discovery of devices/topics/apps in Directory Manager from Resource Service Entity.
    - Subscription to a topic in Topic Directory Manager from Resource Service Entity.

**R2AR**

- R2AR reference point offers generic functions for interaction between a resource service entity and an application repository.
- The R2AR reference point, functions include:
    - Download an application from Application Repository.

**AR2DM**

- AR2DM reference point offers generic functions for interaction between an application repository and a directory manager.
- The AR2DM reference point, functions include:
    - Registration of Application description (ID, name, URI, category, keyword, etc.) of Application Repository to Application Directory Manager.

**A2AR**

- A2AR reference point offers generic functions for interaction between an application and an application repository.
- The A2AR reference point, functions include:
    - Upload an application to Application Repository.

**A2DM**

- A2DM reference point offers generic functions for interaction between an application and a directory manager.
- The A2DM reference point, functions include:
    - Discovery of devices, topics, and apps in Directory Manager.
    - Creation of a topic in Topic Directory Manager.
    - Subscription to a topic in Topic Directory Manager.

# 11 M2M Architecture Description - TTC

No descriptive information available at time of publication.

# 12 Other M2M Architectural Approaches for Consideration

The following architectural approaches, developed by organizations which are not oneM2M Partner Type 1s, were not part of the formal transfer process into oneM2M. This information is provided for information in order to facilitate consideration of potential oneM2M work.

## 12.1 M2M Architecture Description - 3GPP

The 3rd Generation Partnership Program (3GPP) has been developing specifications for providing enhancements and optimizations for their networks for supporting M2M services. While taking a Service Layer independent approach, the 3GPP have developed a set of M2M deployment models and associated architectural enhancements.

There have been requirements from the 3GPP network operators for the consideration of such 3GPP deployment models while oneM2M develops its Service Layer specifications. This contribution presents an extract from 3GPP TS 23.682 [i.1] (Release 11) "Architecture enhancements to facilitate communications with packet data networks and applications", that relate to such architectural enhancements and deployment models for supporting Machine Type Communications (MTC)/M2M services.

### 12.1.1 General Concepts for M2M Communications as foreseen by 3GPP

The end-to-end communications, between the MTC Application in the User Equipment (UE) and the MTC Application in the external network, uses services provided by the 3GPP system, and optionally the services provided by a Services Capability Server (SCS).

NOTE: MTC (Machine Type Communications) is the term used by 3GPP for describing M2M communications.

The MTC Application in the external network is typically hosted by an Application Server (AS) and may make use of an SCS for additional value added services. The 3GPP system provides transport, subscriber management and other communication services including various architectural enhancements motivated by MTC (e.g. device triggering).

Different models are foreseen for machine type of traffic for communication between the AS and the 3GPP system and based on the provider of the SCS (clause 4). Such architectural models include the following:

- Direct Model - The AS connects directly to the 3GPP operator network in order to perform direct user plane communications with the UE without the use of any SCS. The Application in the external network may make use of services offered by the 3GPP system.

- Indirect Model - The AS connects indirectly to the 3GPP operator network through the services of a SCS. The SCS is an entity that may include value added services for MTC (e.g. control plane device triggering) and performs user plane and/or control plane communication with the UE. 3GPP has defined an interface (Tsp) as an interface between the SCS and the 3GPP operator network for control plane communication.
  The SCS may be:

  - MTC Service Provider controlled, in which case the Tsp is an inter-domain interface between the SCS and the 3GPP operator network; or

  - 3GPP network operator controlled, in which case the Tsp is an interface internal to the 3GPP operator network.

- Hybrid Model: The AS uses the direct model and indirect models simultaneously in order to connect directly to the 3GPP operator network to perform direct user plane communications with the UE while also using a SCS. From the 3GPP system perspective, the direct user plane communication from the AS and any value added control plane related communications from the SCS are independent and have no correlation to each other even though they may be servicing the same MTC Application.

  When using the hybrid model, the MTC Service provider controlled SCS, and the 3GPP network operator controlled SCS may offer different capabilities to the MTC Applications.

NOTE: 3GPP has specified MTC-IWF (Machine Type Communications Inter Working Function) as a functional entity located at the edge of the 3GPP operator network that communicates with the SCS for Indirect and Hybrid models.

Such different models are not mutually exclusive, but just complementary. It is, therefore, possible for a 3GPP network operator to combine them for different applications. This may include a combination of both MTC Service Provider and 3GPP network operator controlled SCSs communicating with the same 3GPP operator network.

### 12.1.2 MTC Deployment Scenarios as foreseen by 3GPP

In the indirect and hybrid models, the deployment of an SCS may be inside or outside the 3GPP network operator domain as illustrated in figures 12.1 and 12.2. When the SCS is part of the 3GPP network operator domain (figures 12.1 "C" and 12.2), the SCS is considered a 3GPP network operator internal network function, is operator controlled, and may provide operator value-added services. In this case, security and privacy protection for communication between the MTC-IWF and SCS is optional. When the SCS is deployed outside the 3GPP network operator domain (figures 12.1 "B" and 12.2), the SCS is MTC Service Provider controlled. In this case, security and privacy protection for communication between the MTC-IWF and SCS is needed. In the direct model (figure 12.1 "A"), there is no external or internal SCS in the communication path.

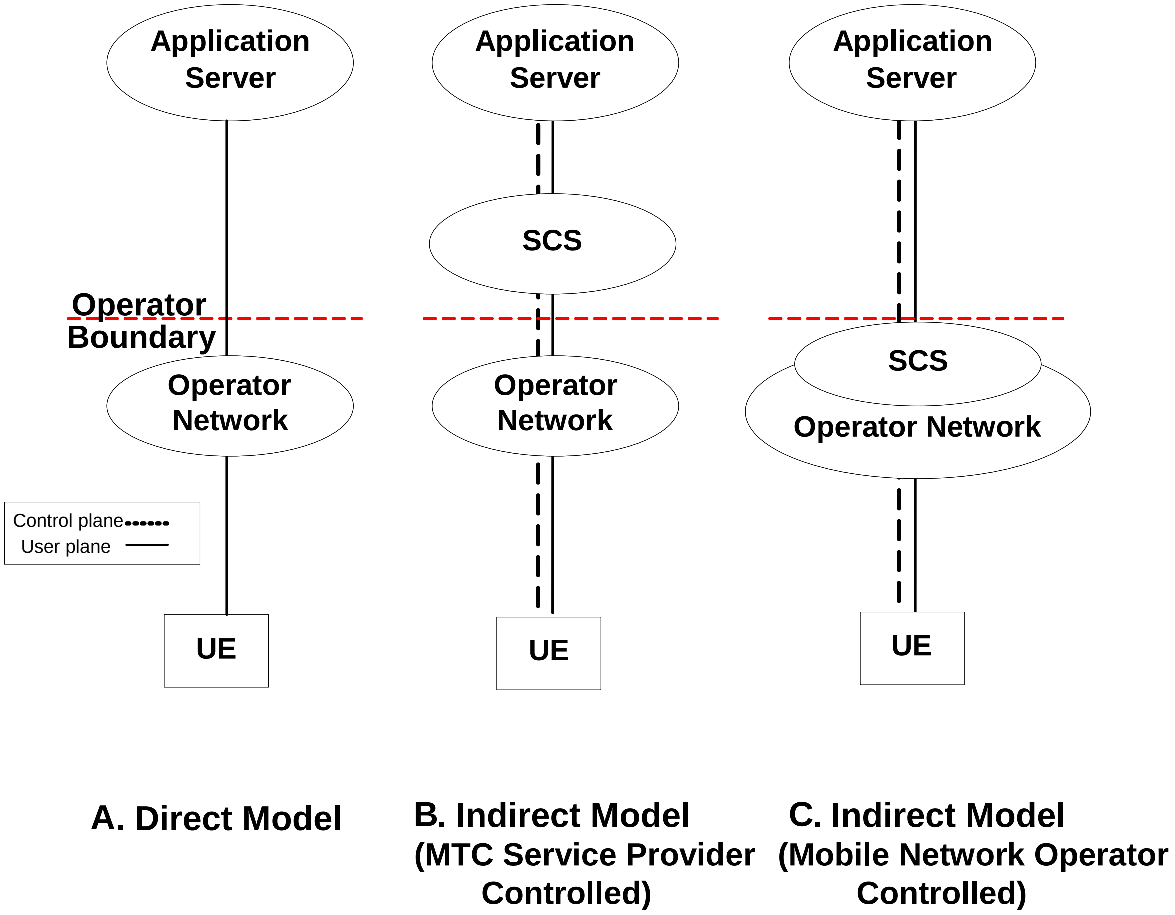


**Figure 12.1: 3GPP Deployment scenarios for direct and indirect model**

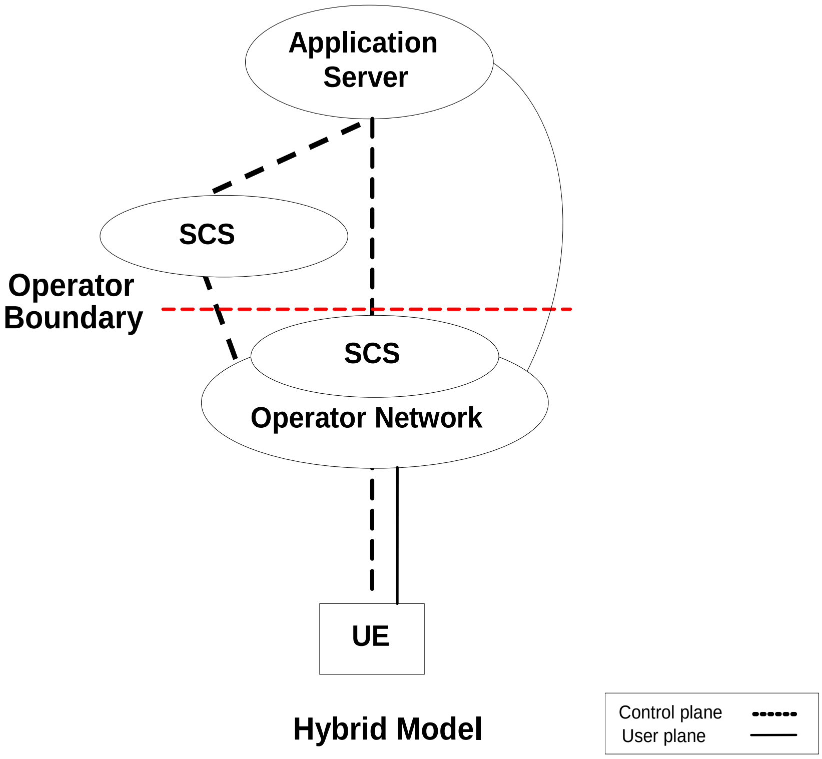


**Figure 12.2: 3GPP Deployment scenarios for hybrid model**

A 3GPP network operator may deploy the hybrid model with a combination of no internal and external SCS (as in the Direct Model) and internal and/or external SCS (as in the Indirect Model). As shown in figure 12.2, a UE may be in communications with multiple SCSs in an HPLMN (home 3GPP operator network) which can be made up of a combination of 3GPP network operator controlled and MTC service provider controlled SCSs. In that scenario, the MTC Service provider controlled SCS, and the 3GPP network operator controlled SCS may offer different capabilities to the MTC Applications.

Though not illustrated, it is also possible that the deployment of an AS may be inside the 3GPP network operator domain and under operator control.

### 12.1.3 Communication between oneM2M Service Layer and 3GPP Network

Communication between the Service Layer platform (to be specified by oneM2M) and the 3GPP network will make use of the User Plane and the Control Plane communication paths, as needed for different 3GPP deployment models. The definition of such User Plane and Control Plane communication path can include, but not limited to the interfaces and/or the APIs specified by the 3GPP. Clause 12.1.4 illustrates the Architecture Reference Model for M2M services, as specified by the 3GPP in their Release 11 specifications. This Architecture Reference Model illustrates reference points, such as the Tsp, Gi/SGi, Tsms etc., to be used for communications between the SCS/AS and the entities in the 3GPP network.

### 12.1.4 3GPP Architectural Reference Model

Figure 12.3 shows the architecture for a UE used for MTC connecting to the 3GPP network. The architecture supports various architectural models described in clause 12.1.2.

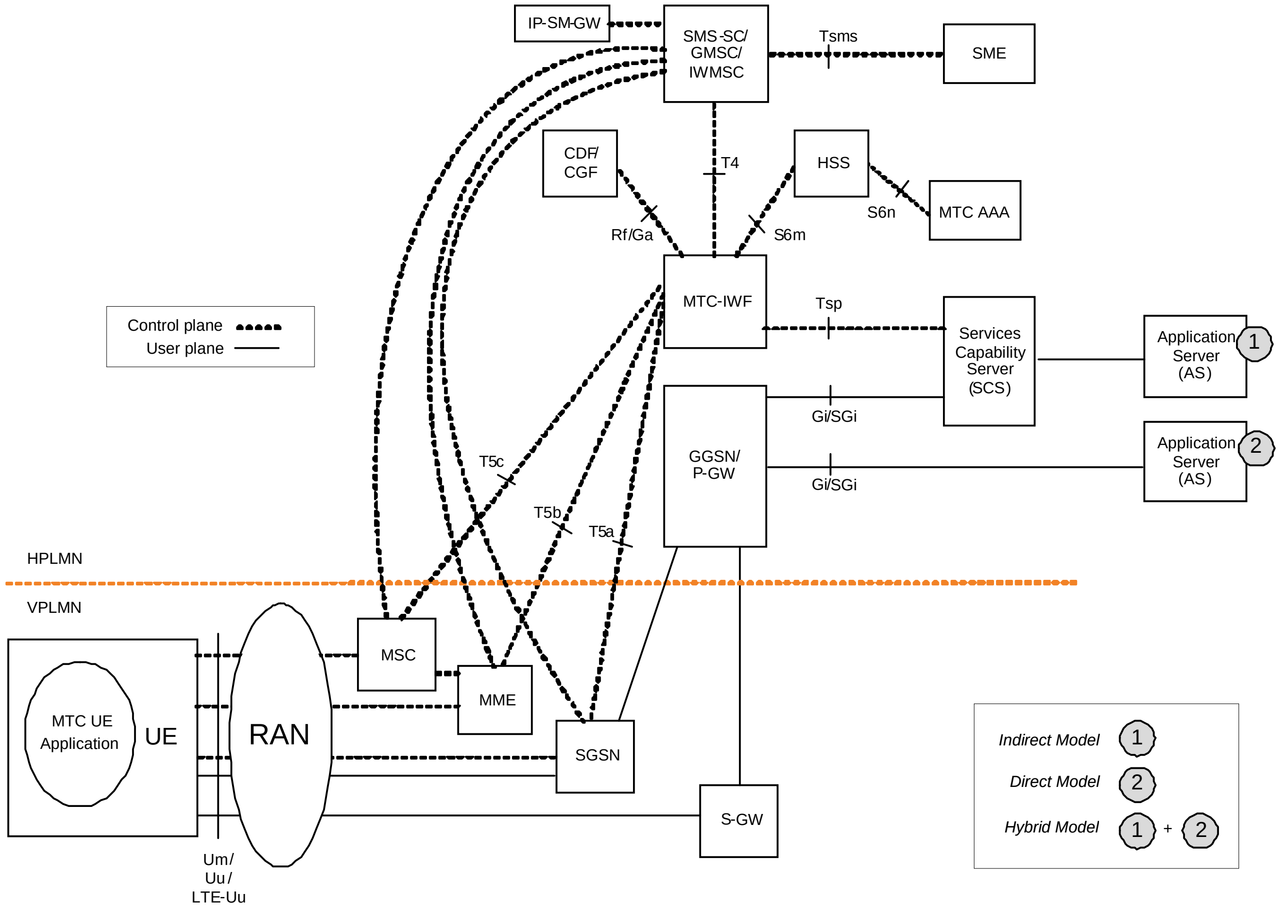


**Figure 12.3: 3GPP Architecture for Machine-Type Communication**

NOTE 1: SCS may use other reference points and/or APIs as specified by 3GPP, OMA, etc.

NOTE 2: Figure 12.3 provides a snapshot of the architecture foreseen by 3GPP for MTC communication based on 3GPP TS 23.682 (V11.3.0) [i.1]. Refer to the most recent version of this specification for a full description.

The SCS is an entity which connects to the 3GPP operator network to communicate with UEs used for MTC and the MTC-IWF in the HPLMN. The SCS offers capabilities for use by one or multiple MTC Applications. A UE can host one or multiple MTC Applications. The corresponding MTC Applications in the external network are hosted on one or multiple ASs.

Tsms is the interface that encompasses various SMS-SC (Short Message Service-Service Centre) to SME (Short Message Entity) interface standards and not specified by 3GPP. Tsms can be used to send a trigger to a UE encapsulated in a MT-SMS (Mobile Terminated SMS) as an over-the-top application by any network entity (e.g. SCS) acting as a SME. Tsp is a 3GPP standardized interface to facilitate value-added services motivated by MTC (e.g. control plane device triggering).

## 12.1.5 Security Aspects for communication between Service Capability Server and the 3GPP Network

In the indirect and hybrid models illustrated in clause 12.1.2, the deployment of an SCS may be inside or outside the 3GPP network operator domain. When the SCS is inside the 3GPP network operator domain, the SCS is considered a 3GPP network operator internal network function. In this case, security and privacy protection for communication between the MTC-IWF and SCS is optional. When the SCS is deployed outside the 3GPP network operator domain, the SCS is MTC Service Provider controlled. In this case, security and privacy protection for communication between the MTC-IWF and SCS is needed. 3GPP SA3 have specified requirements for security and privacy aspects for communication between the SCS and the MTC-IWF. Such security aspects for Tsp interface are detailed in clause 4.8 of 3GPP TS 23.682 [i.1], and provide for the following:

NOTE: Refer to the most recent version of 3GPP TS23.682 [i.1] specification for a full description of Security Aspects over the Tsp interface:

- mutual authentication between the MTC-IWF and the SCS, including the ability to initiate such mutual authentication by either entity;
- means to assure the integrity of the data transferred over the Tsp interface;
- means to protect against replay protection of the data transferred over the Tsp  interface;
- means to assure the confidentiality of the data transferred over the Tsp interface;
- means to assure the privacy of any sensitive data transferred over the Tsp interface and processed by either party.

## 12.2 M2M Architecture Description - 3GPP2

This clause introduces aspects that are specific to the enhancements and optimizations, and the associated deployment requirements being developed by the 3GPP2 for supporting M2M services.

### 12.2.1 Background - 3GPP2

It is understood that the 3GPP2 have been developing specifications for providing enhancements and optimizations for their networks for supporting M2M services. While taking a Service Layer independent approach, the 3GPP2 have developed a set of M2M deployment models and associated architectural enhancements.

There have been requirements from the 3GPP2 network operators for the consideration of such 3GPP2 deployment models while oneM2M develops its Service Layer specifications. This contribution presents an extract from 3GPP2 X.R0067 [i.4] (Machine to Machine Architecture and Enhancements Study for cdma2000 Networks) and from 3GPP2 X.P0068 [i.5] (Network Enhancements for Machine to Machine) that relate to the architectural enhancements and deployment models for supporting Machine to Machine services in 3GPP2 networks.

### 12.2.2 General Concepts for M2M Communications - 3GPP2

End-to-end services between the M2M Applications in the UEs and the M2M Applications in the external network use services provided by the 3GPP2 system, and optionally services provided by an M2M Server. The 3GPP2 system provides transport and communication services (including 3GPP2 bearer services, IMS and SMS) and various optimizations that can facilitate M2M type of services.

Different models are foreseen for M2M type of traffic for communications between the M2M Applications and the 3GPP2 network. Different architectural models that are supported by the 3GPP2 Architectural Reference Model specified in X.P0068 (see clause 12.2.3) include the following:

- Direct Model: Direct Communication provided by the 3GPP2 Network Operator:
  - The M2M Applications in the external network connect directly to the M2M Applications in the UEs used for M2M, via the 3GPP2 network without the use of any M2M Server.
- Indirect Model - M2M Service Provider controlled communication:
  - Uses an M2M Server that is an entity outside the 3GPP2 network operator domain for enabling communications between the M2M Applications in the external network and at the UEs used for M2M. In this model, interfaces M2Msp and SMS (see clause 12.2.5) are the interfaces that the third party M2M Server supports with the entities in the 3GPP2 network.
- Indirect Model - 3GPP2 Operator controlled communication:
  - Uses an M2M Server that is an entity inside the 3GPP2 network operator domain for enabling communications between the M2M Applications and at the UEs used for M2M. Interfaces M2Msp and SMS (see clause 12.2.5) are the interfaces that the 3GPP2 network operator controlled M2M Server supports with other entities in the 3GPP2 network.
- Hybrid Model:

- Direct and Indirect models are used simultaneously in the hybrid model i.e. performing Control Plane signalling using the Indirect Model and connecting the M2M Applications in the external network and at the UEs used for M2M over User Plane using the Direct Model.

NOTE: 3GPP2 has specified M2M-IWF (Machine to Machine Inter Working Function) as a functional entity located in the 3GPP2 network that communicates with the M2M Server for Indirect and Hybrid models.

Since the different models are not mutually exclusive, but just complementary, it is possible for a 3GPP2 network operator to combine them for different applications. 3GPP2 network operator can provide value added services to an M2M Application by using an operator controlled M2M Server. In addition, the 3GPP2 network operator can also offer M2M related services by using a third party provided M2M Server.

### 12.2.3 M2M Deployment Scenarios - 3GPP2

In the indirect and hybrid models, the deployment of an M2M Server may be inside or outside the 3GPP2 network operator domain as illustrated in figures 12.4 and 12.5. When the M2M Server is part of the 3GPP2 network operator domain (figures 12.4 (C) and 12.5), the M2M Server is considered a 3GPP2 network operator internal network function, is operator controlled, and may provide operator value-added services. In this case, security and privacy protection for communication between the M2M-IWF and the M2M Server is optional. When the M2M Server is deployed outside the 3GPP2 network operator domain (figures 12.4 (B) and 12.5), the M2M Server is M2M Service Provider controlled. In this case, security and privacy protection for communication between the M2M-IWF and the M2M Server is needed. In the direct model (figure 12.4 (A)), there is no external or internal M2M Server in the communication path.

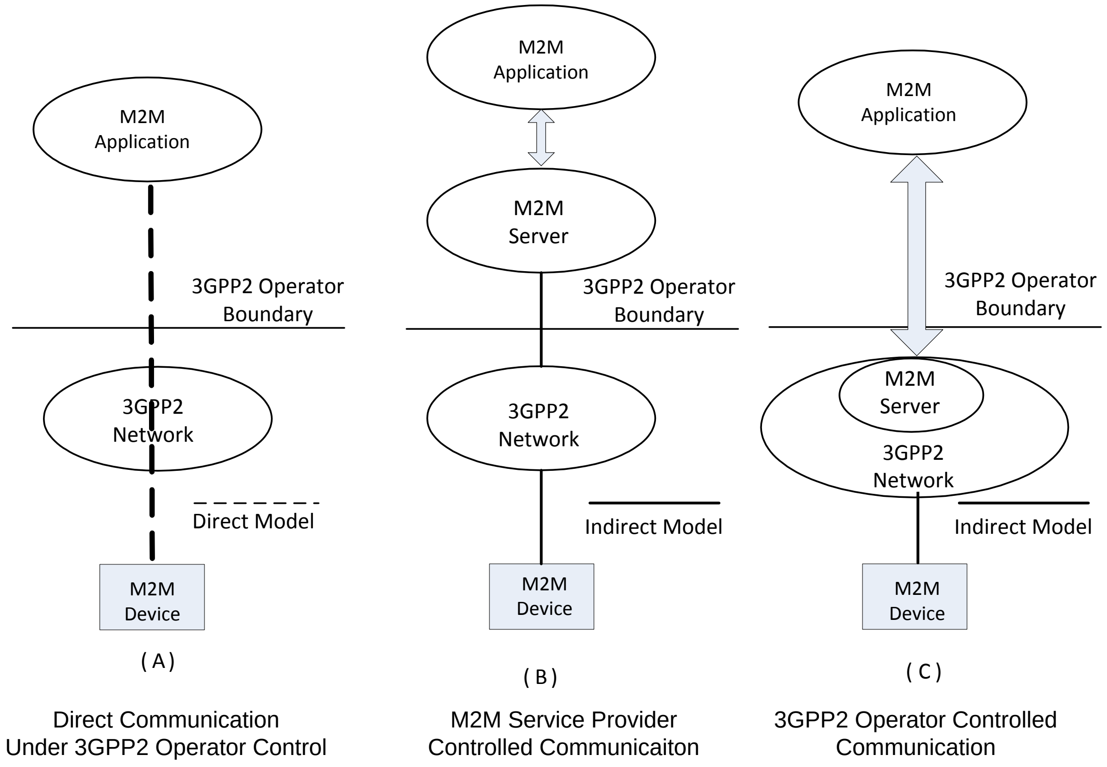


**Figure 12.4: M2M Application to M2M Device Communication Models**

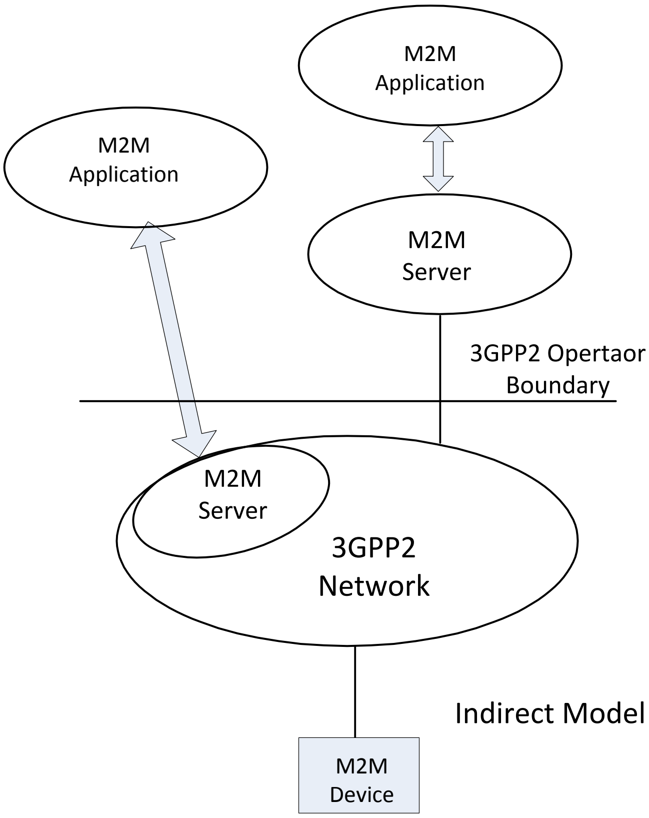


**Figure 12.5: Multiple M2M Applications using Diverse Communication Models**

A 3GPP2 network operator may deploy the hybrid model with a combination of no internal and external M2M Server (as in the Direct Model) and internal and/or external M2M Server (as in the Indirect Model). As shown in figure 12.5, a UE (M2M Device) may be in communication with multiple M2M Servers which can be made up of a combination of 3GPP2 network operator controlled and M2M Service Provider controlled M2M Servers. In that scenario, the M2M Service Provider controlled M2M Server, and the 3GPP2 network operator controlled M2M Server may offer different capabilities to the M2M Applications.

Though not illustrated, it is also possible that in the Indirect Service Model with 3GPP2 network operator controlled M2M Server; the M2M Application may be inside the 3GPP2 network operator domain and under 3GPP network operator control.

### 12.2.4 Communication between oneM2M Service Layer and the 3GPP2 Network

Communication between the Service Layer platform (to be specified by oneM2M) and the 3GPP2 network will make use of the User Plane and the Control Plane communication paths, as needed for different 3GPP2 deployment models. The definition of such User Plane and Control Plane communication path can include, but not limited to the interfaces and/or the APIs specified by the 3GPP2. Clause 12.2.5 illustrates the Architecture Reference Model for M2M services, as specified by the 3GPP2 in their X.P0068 [i.5] specifications. This Architecture Reference Model illustrates the reference points, such as the M2Msp, SMS, USSD, IP etc., to be used for communications between the M2M Server/M2M Application and the entities in the 3GPP2 network.

### 12.2.5 3GPP2 Architectural Reference Model

Figure 12.6 shows the architecture for a UE used for M2M connecting to the 3GPP2 network. The architecture supports various architectural models described in clause 12.2.3.

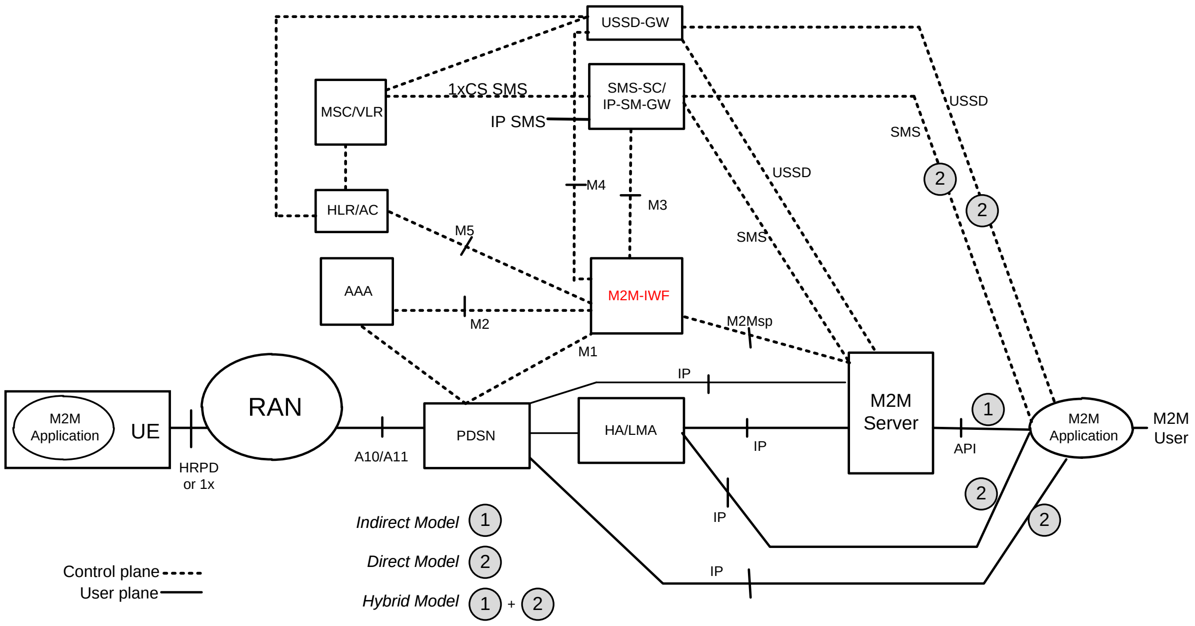


**Figure 12.6: 3GPP2 Enhanced Network Architecture to Support M2M**

The M2M Server is the entity which connects to the 3GPP2 network for providing communication with the UEs used for M2M. The M2M Server offers capabilities for use by one or multiple M2M Applications hosted by the UE. The corresponding M2M Applications in the external network are hosted on one or multiple M2M Application platform(s).

The SMS interface shown in figure 12.6 encompasses various SMS-SC (Short Message Service-Service Centre) to SME (Short Message Entity) interface standards and not specified by 3GPP2. The SMS interface can be used to send a trigger to a UE encapsulated in a MT-SMS (Mobile Terminated SMS) as an over-the-top application by any network entity (e.g. M2M Server) acting as a SME. M2Msp is a 3GPP2 standardized interface to facilitate value-added services for M2M (e.g. control plane device triggering). USSD interface is used for providing unstructured supplementary IMS services.

NOTE: Figure 12-6 provides a snapshot from 3GPP2 X.P0068 [i.5] (X50-20121108-002) of the architecture foreseen by 3GPP2 for supporting M2M services. Refer to the most recent version of 3GPP2 X.P0068 [i.5] for a full description.

# 13 Summary

For an overview analysis of architecture information which has been contributed by oneM2M Partners, see annex A.

# Annex A: Additional Architecture Information

The purpose of this Annex is to help navigate content related to architecture which has been contributed by oneM2M Partners and is available in the document pool.

The goal of this annex is to provide the following information:

- Help members assess the level of advancement of M2M architecture contributed by oneM2M Partners Type 1.
- Classify by logical functions the content submitted to the document pool.
- Indicate which Partners are still working on each topic.
- Give exposure to documents not written in English.

oneM2M has discussed the building blocks of an M2M architecture. This list is not exhaustive, but can be used to organize the information contributed by the partners.

Below is a table with the names for each building block in a row. The definitions for each building block are provided below the table.

## A.1 Nomenclature

| | |
|---|---|
| • Req | there is Requirements-related content in the pool |
| • Arc | there is Architecture-related content in the pool |
| • Pro | there is Protocol-related content in the pool |
| • No content | no related content has been contributed to the pool |
| • WIP | the oneM2M Partner has Work-In-Progress on the related topic |

## A.2 Architecture Contribution Areas by oneM2M Partners Type 1

| Building Block | CCSA | TIA | TTA | ETSI |
|---|---|---|---|---|
| Subscription Information Handling | Arc | No Content | Arc & WIP | Req, Arc, & Pro |
| Device Management | Arc | Arc | WIP | Req, Arc, & Pro |
| Security | Req & Arc | (tbd) | WIP | Req, Arc, & Pro |
| Application and Device Registrations | Arc & WIP | Arc | WIP | Req, Arc, & Pro |
| Resource Management | Arc & WIP | Arc | Arc & WIP | Req, Arc, & Pro |
| Content push/pull Services | Req | Arc | No content | Req, Arc, & Pro |
| Store and Forward Messaging | Arc & WIP | Arc | No content | Req, Arc, & Pro |
| Protocol Translation (binding) | No content<br>WIP: 2012-2517T-YD | No Content | (tbd) | Req, Arc, & Pro |
| Subscribe/Notification | Arc & WIP | Arc & WIP | No content | Req, Arc, & Pro |
| Location and Geo Fencing | No content | No Content | Arc &WIP | Req, Arc, & Pro |
| Groups Management | WIP | No Content | Arc & WIP | Req, Arc, & Pro |
| Device Triggering | No content | No Content | No content | No content |
| Access Control | No content | Arc | WIP | Req, Arc, & Pro |
| Data Processing and Storage | WIP | No Content | WIP | Req, Arc, & Pro |
| Consumption Statistics/Records | No content | No Content | No content | Arc |
| API Management | No content | Arc | Arc | Req, Arc, & Pro |

## A.3 Detailed annex references

References for architecture related content are provided below.

### A.3.1 Subscription Information Handling References

On-board new devices by creating new subscription in the common M2M service layer and network layer; activate/deactivate/suspend/resume network and service subscriptions.

- CCSA
  1) YDT 2399-2012 section 5.1.7.
  2) WIP: 2012-2549T-YD (eUICC subscription provision) sections 5, 6 & 8.2.4.
- ETSI
  1) Platform only.
- TTA
  1) TTAK.KO-06.0168 section 6 - USN meta-data models.
  2) WIP: Consortium draft1.1- e-health system requirement description on the initialization procedure.

NOTE 1: WIP completion scheduled for 3Q2013.

3) WIP: TTA draft2.1- high level model of M2M interworking.

NOTE 2: WIP completion scheduled for 3Q2013.

### A.3.2 Device Management References

Manage all aspects of the devices including configuration, firmware upgrades, application lifecycle management, device lock and wipe.

- CCSA
  1) YDT 2399-2012 sect 4 and 5.
  2) WIP: 2012B93 sect 5, 7 and 9.2.

NOTE: WIP completion scheduled for 4Q2013.

- TIA
  1) TIA-4940-020 M2M-Smart Device Communications; Protocol Aspects (Section 6 Security Commands).
  2) TIA TSB-4940 Smart Device Communications; Security Aspect.
- TTA
  1) Consortium draft1.2- smart home use case.
  2) Consortium draft1.1- e-health system requirement description on the remote management of the devices.
  3) TTA draft2.3-M2M device registration and discovery.
- ETSI
  1) TS 103 092 (OMA DM compliant model).
  2) TS 103 093 (BBF TR069 compliant model).

3) TS 102 690, clauses 5 and 9, annexes B and E.

4) TS 102 921, clause 10 and annex E.

### A.3.3 Security References

Generate relevant key material for secure communications; Authenticate devices before they can register and modify resources. Prevent unauthorized entities from sending IP packets to the devices. Provide a secure connection to the device

- CCSA

  1) YDT 2399-2012, sections 4.4 and 5.2.

- ETSI

  1) TS 102 690, clauses 5.2.6, 5.4.6 and 8.

- TTA

  1) Consortium draft1.2- smart home use case.

### A.3.4 Application and Device Registration References

Application and devices will be able to register with the service layer entity for various services. Registration will involve authentication or verification of credentials and creation or allocation of resources within the server and the database. A profile with the capabilities of the device and the type of services allowed for the applications are created.

- CCSA

  1) YDT 2399-2012, section 4.3.1, annexes B, G (device reg); and H (app reg).

  2) 2012-0336T-YD sections 6, 9 (req.).

  3) WIP: 2012B93 section 9.1 (device reg).

NOTE 1: WIP completion scheduled for 4Q2013.

  4) WIP: 2012B93 section 9.3 (app reg).

NOTE 2: WIP completion scheduled for 4Q2013.

- TIA

  1) TIA-4940-005 Smart Device Communications; Reference Architecture.

  2) TIA-4940-020 (Section 5 Basic Commands and Section 6 Security Commands).

- TTA

  1) WIP: TTA draft2.1- high level model of M2M interworking.

NOTE 3: WIP completion scheduled for 3Q2013.

  2) TTA draft2.4-requirements of P2P.

- ETSI

  1) TS 102 690, clause 7.3.

  2) Reg, clauses 9 and 9.3.2.8.2.

  3) Device Reg, clauses 8 and 9.

## A.3.5 Resource Management References

Applications and devices will be able to create, update, and delete resource objects containing various attributes in the service layer. Entities will be able to discover resources.

- CCSA

  1) 2012-0336T-YD sect 9, 11 (req.).
  2) YDB 064-2011 sect 6.1, 9.1.
  3) WIP: 2012B93 sect 9.1, 9.2, 9.8 (most related to the concept of resource management defined below).

NOTE 1: WIP completion scheduled for 4Q2013.

- TIA

  1) TIA-4940-005 (General).
  2) TIA-4940-020 (Section 5 Basic Commands).

- TTA

  1) TTAK.KO-06.0282 section 5.3.
  2) TTAK.KO-06.0167 section 4,5,6 - directory service requirement & interface specification for managing metadata of USN resources.
  3) TTAK.KO-06.0168 section 5 - USN resource identifiers requirements.
  4) TTA draft2.1- high level model of M2M interworking.

NOTE 2: WIP completion scheduled for 3Q2013.

- ETSI

  1) TS 102 690, clause 9.

## A.3.6 Content push/pull Services References

Provide API for applications to perform unicast and multicast data push to specified devices within the specified time window. Push may be result of a notification that is triggered as a result of modification of a resource. Provide API for applications to pull data from one or more devices within the specified time window or specified periodicity or other policies that have been established.

- CCSA

  1) No content.

NOTE 1: Requirement addressed in 2012-0336T-YD sect 5.4.4, YDB 064-2011 sect 5.2.3 and YDB 100-2012 sect 4.4.1.

- TIA

  1) TIA-4940-020 (new Section 5).
  2) WIP: TIA-4940-050 M2M-Smart Device Communications; Capabilities (will include "Push and Pull").

NOTE 2: WIP completion scheduled for 3Q2013.

- TTA
- No content ETSI

  1) TS 102 690, clause 9.

## A.3.7 Store and Forward Messaging References

Applications may request messages to be sent to one or more devices that may not be registered with the network at that time. In this case the communications management entity shall store and aggregate the messages and forward them to the devices at a later time when the devices wake up.

- CCSA

  1) YDT 2399-2012 sect 3.2, 3.3, 4.3, 4.4.3.1.3.3, 5.1, Annex E).

  2) WIP: 2012B93 sect 4, 7, 9.7.

NOTE: WIP completion scheduled for 4Q2013.

- TIA

  Only high-level description.

- TTA

  No content.

- ETSI

  1) TS 102 690, clause 9.1.1.

## A.3.8 Protocol Translation References

Translate protocols between application and device as needed. For example, applications may use HTTP while devices may use Constrained Application Protocol (CoAP) or Zigbee SE 2.0 protocol.

- CCSA

  1) No content.

NOTE 1: Requirement addressed in 2012-0336T-YD, YDB 064-2011 and YDB 100-2012).

  2) WIP: 2012-2517T-YD

NOTE 2: WIP completion scheduled for 4Q2013.

- TIA

  No Content (Assumed to be Protocol agnostic).

- TTA

  1) WIP: TTA draft2.5- UM3 protocols.

NOTE 3: WIP completion scheduled for 3Q2013.

- ETSI

  1) TS 102 921, section binding.

## A.3.9 Subscribe/Notification References

Application and devices should be able to subscribe to receive notifications upon certain events or when certain resources are updated. Events may be specified as rules on certain resource data.

- CCSA

  1) YDT 2399-2012 sect 5.1.5.

2) WIP: 2012B93 sect 4, 7, 9.7. ("subscribe / notification" is considered as part of "store and forward" in this document).

NOTE 1: WIP completion scheduled for 4Q2013.

- TIA

  1) TIA-4940-020 (Section 5).

  2) WIP: TIA-4940-050.

NOTE 2: WIP completion scheduled for 3Q2013.

- TTA

  No Content.

- ETSI

  1) TS 102 690, clause 9.

## A.3.10 Policy Framework References

Framework to establish and incorporate in the session orchestration, data aggregation and storage, the network provider and application provider policies. Examples include incorporating a location tag or time stamp on all data, policy restricting sessions only to certain hours of the day.

- No content

## A.3.11 Location and Geo Fencing References

Provide device and network based location and location related services such as creating a geo-fence or identifying a group of devices within a region or adding a location tag to the device data.

- ETSI

  1) TS 102 690, clauses 9 and 9.3.2.13.

- TTA

  1) TTAK.KO-06.0170 section 4.6.2 - USN service platform requirements for providing WIPe-area application service deployment.

  2) WIP: TTA draft2.2- M2M specific functional requirements.

NOTE: WIP completion scheduled for 3Q3013.

## A.3.12 Groups Management References

Framework for creation of groups by specifying the members of the group through one of the identifiers of the device, adding additional members or removing members; setting group attributes.

- CCSA

  1) WIP: 2012B93 sect 4, 7, 9.8

NOTE 1: WIP completion scheduled for 4Q2013.

- TIA

  No Content.

- TTA

  1) TTAK.KO 06.0283 section4, section5 - this document is for "USN resource community" which implies the groups-management of resources.

  2) Consortium draft1.2- smart home use case.

  3) WIP: TTA draft2.2- M2M specific functional requirements.

NOTE 2: WIP completion scheduled for 3Q2013.

- ETSI

  1) TS 102 690, clause 9.

## A.3.13 Device Triggering References

Provide the capability to trigger the device to register with the network and an application through a secondary means such as an SMS. Provide information about the status of the device in the network.

- No content

## A.3.14 Access Control References

Control the access to the data collected from the devices based on access restrictions specified by applications in terms which users or devices can access what resources.

- TIA

  1) TIA-4940-020 (Section 5).

- ETSI

  1) By association of proper access right resources (TS 102 690, clause 9).

- TTA

  1) Consortium draft1.2- smart home use case.

## A.3.15 Data Processing and Storage References

Provide temporary and permanent storage for data collected from devices. Process queries on data collected. Provide threshold and expression rules setting and execution on the various data collected from the devices. Notifications could be triggered based on the outcome of the rules testing.

- CCSA

  1) WIP: 2012B93 sect 4, 7, 9.7.

NOTE 1: WIP completion scheduled for 4Q2013.

- ETSI

  1) TS 102 690, clause 9 related to the container entity, its ability to historize container instances; then it uses notifications.

NOTE 2: ETSI M2M has also a basic search mechanism based on search strings and some basic expressions (Date/time, number of instances, etc.) but it is not so sophisticated. ETSI M2M has linked the design of a more sophisticated mechanism to the development of semantic aspects.

- TTA

  1) Consortium draft1.1- e-health system requirement description on the pre-definition of the alarm level.

2) Consortium draft1.3- Fleet management use case.

### A.3.16 Consumption Statistics/records References

Process queries regarding the usage of network resources by a device or a group of devices for billing reconciliation.

- ETSI

    1) TR 101 603 clause 4.4

### A.3.17 API Management References

Manage API usage, such as authentication and authorization of calls to APIs provided.

- CCSA

    No content.

- TIA

    1) TIA-4940-020 (Section 6).

- ETSI

    1) TS-102 921, mIa, dIa and mId interfaces.

NOTE: In ETSI M2M, On mId (D/G <->Network) is fully developed.
On dIa and mIa is a function that is explicitly requested, but that left for implementation in the initial phase under the following rationale:

- ▪ for dIa in mainly an internal device (or an area network aspect in case of D';
- ▪ on mIa is a service provider application issue.

- TTA

    1) TTAK.KO-06.0282 sections 6.3, 7.2.2, 7.2.4.

    - ▪ The section describes the requirements on Web-based open APIs, Semantic open APIs service, and Semantic query processing.

## A.4 Annex Documents - Detailed ToC

Translation of the Table of Contents (ToC) for referenced non-English language documents in the pool is provided herein to put referenced sections in context.

NOTE: These translations are not official.

### A.4.1 CCSA Reference Documents

#### A.4.1.1 2012-0336T-YD Requirement Document (CCSA Project - WIP)

NOTE: WIP completion scheduled for 4Q2013General Requirements on WAN Based Remote Measurement and Control Applications for Intelligent Agriculture:

1 Introduction

2 Normative references

3 Abbreviations

7.2 Technical requirements

7.2.1 Gateway requirements

7.2.2 Sensing node requirements

7.2.3 Sensor category

7.2.4 Actuator categories

8 Network layer requirements

8.1 Overall requirements

8.2 Technical requirements

9 Application layer requirements

9.1 Overall requirements

9.2 Technical requirements

9.2.1 Service adaptation module requires

9.2.2 The business logic module

9.2.3 Management logic module

9.2.4 Ability to open module

9.2.5 Business application layer requirements

9.2.6 Other requirements

10 Security requirements

10.1 Equipment safety requirements

10.2 Network security requirements

10.3 The business of security requirements

10.4 Information security requirements

10.5 User privacy security

#### A.4.1.2 YDT 2399-2012, Architecture Document (Published)

1 Scope

2 Terminology, definitions and abbreviations

2.1 Terms, definitions

2.2 Abbreviations

3 Overview

3.1 General Overview

3.2 Communication protocol between M2M device and M2M platform

3.3 Communication protocol between M2M platform and M2M applications

3.4 M2M application communication protocol stack

4 M2M terminal devices with M2M platform agreement

4.1 Protocol packets description

4.1.1 Protocol packet structure

4.1.2 Packet Classification

4.1.3 Encapsulation of TLV Messages, and mechanism for split of oversized messages

4.1.4 Packet byte order

4.2 Basic principles of the protocol interaction

4.2.1 Long connection and short connection

4.2.2 Synchronous interaction and asynchronous interaction

4.3 Protocol functions and their interactions

4.3.1 Registered terminal equipment

4.3.2 Terminal equipment Login

4.3.3 Terminal equipment Logout

4.3.4 Connection detection

4.3.5 Terminal information reported

4.3.6 Terminal information query

4.3.7 Remote control

4.3.8 Remote upgrade

4.3.9 Ordinary configuration parameters

4.3.10 Safety parameter settings

4.3.11 Forwarding business data

4.4 Protocol security mechanism

4.4.1 SIM card M2M terminal equipment certification

4.4.2 Terminal equipment and data platform to interact safety

4.4.3 Password and key distribution

4.4.4 Local manual removal interactive data terminal equipment platform security settings

4.4.5 Security Failure handling processes

4.4.6 Abnormalities in the communication process and retransmission

5 Agreement between the M2M platform and M2M applications

5.1 Protocol interaction mechanisms

5.1.1 M2M applications M2M platform landing

5.1.2 M2M applications M2M platform Logout

5.1.3 Terminal device information query

5.1.4 Terminal device status alarm

5.1.5 Terminal device notification

5.1.6 Remote control terminal equipment

# Annex B: Bibliography

The following documents have been provided by the oneM2M Partners as input to a common repository (pool); for the purpose of drawing material for contribution to work items.

Documents listed below have been made available via the oneM2M pool directory at: ftp://ftp.onem2m.org/Pool.

## B.1 ARIB Documents

Not available at time of Publication.

## B.2 ATIS Documents

Not available at time of Publication.

## B.3 CCSA Documents

[b.3.a] 2009H57 (CCSA) "General Framework and Technical Requirements of IoT (Internet of Things)".

[b.3.b] 2010T56 (CCSA) "Technical Requirements of Smart Home within Communication Network-supported Ubiquitous IoT Application".

[b.3.c] 2011T91 (CCSA) "General Technical Requirements of Vehicle Networking".

[b.3.d] 2012-0336T-YD (CCSA) "General Requirements on WAN Based Remote Measurement and Control Applications for Intelligent Agriculture".

[b.3.e] YDB 062-2011 (CCSA) "Terms of the Ubiquitous Network".

[b.3.f] YDB 064-2011 (CCSA) "Vehicle Telematics Service Requirement and General Framework".

[b.3.g] YDB 083-2012 (CCSA) "Identifiers, Resolution and Addressing System in Ubiquitous Network".

[b.3.h] YDB 100-2012 (CCSA) "Requirements of Internet of Things".

[b.3.i] YDB 101-2012 (CCSA) "Security Requirements of Internet of Things".

[b.3.j] YDB 102-2012 (CCSA) "General Architecture of Intelligent Transportation System Based on Telecommunication Networks".

[b.3.k] YDT 2398-2012 (CCSA) "General Technical Requirements of M2M Service".

[b.3.l] YDT 2399-2012 (CCSA) "Technical Requirements of M2M Service Communication Protocol".

## B.4 ETSI Documents

[b.4.a] ETSI TR 101 584: "Machine-to-Machine communications (M2M); Study on Semantic support for M2M Data".

[b.4.b] ETSI TS 102 689: "Machine-to-Machine communications (M2M); M2M service requirements".

[b.4.c] ETSI TS 102 690: "Machine-to-Machine communications (M2M); Functional architecture".

[b.4.d] ETSI TR 102 725: "Machine-to-Machine communications (M2M); Definitions".

[b.4.e] ETSI TR 102 732: "Machine-to-Machine communications (M2M); Use cases of M2M applications for eHealth".

[b.4.f] ETSI TR 102 857: "Machine-to-Machine communications (M2M); Use cases of M2M applications for Connected Consumer".

[b.4.g] ETSI TR 102 897: "Machine-to-Machine communications (M2M); Use cases of M2M applications for City Automation".

[b.4.h] ETSI TR 102 898: "Machine-to-Machine communications (M2M); Use cases of Automotive Applications in M2M capable networks".

[b.4.i] ETSI TS 102 921: "Machine-to-Machine communications (M2M); mIa, dIa and mId interfaces".

[b.4.j] ETSI TR 102 935: "Machine-to-Machine communications (M2M); Applicability of M2M architecture to Smart Grid Networks; Impact of Smart Grids on M2M platform".

[b.4.k] ETSI TR 102 966: "Machine-to-Machine communications (M2M); Interworking between the M2M Architecture and M2M Area Network technologies".

[b.4.l] ETSI TS 103 092: "Machine-to-Machine communications (M2M); OMA DM compatible Management Objects for ETSI M2M".

[b.4.m] ETSI TS 103 093: "Machine-to-Machine communications (M2M); BBF TR-069 compatible Management Objects for ETSI M2M".

[b.4.n] ETSI TR 103 104: "Machine-to-Machine communications (M2M); Interoperability Test Specification for CoAP Binding of ETSI M2M Primitives".

[b.4.o] ETSI TR 103 167: "Machine-to-Machine communications (M2M); Threat analysis and counter-measures to M2M service layer".

## B.5 TIA Documents

[b.5.a] TIA-4940.000 (TIA TR-50-1) "Smart Device Communications; List of Parts".

[b.5.b] TIA-4940.005 (TIA TR-50-1) "Smart Device Communications; Reference Architecture".

[b.5.c] TIA-4940.020 (TIA TR-50-1) "Smart Device Communications; Protocol Aspects; Introduction".

## B.6 TTA Documents

[b.6.a] (TTA) "Use case - Fleet management services".

[b.6.b] (TTA) "Requirements for M2M-based Remote Pulse Monitoring System".

[b.6.c] (TTA) "Use Case - Energy Management System for Public Buildings and Surveillance".

[b.6.d] (TTA)"Service Requirement - Energy Management System for Public Buildings and Surveillance".

[b.6.e] (TTA) "Use Case - Smart Home Service".

[b.6.f] (TTA) "Requirements of open architecture for M2M device registration and discovery".

[b.6.g] (TTA) "Scenarios and basic requirement of P2P communication for M2M devices".

[b.6.h] TTAK.KO-06.0282 (TTA) "Requirements and Reference Architecture for Open USN Service Framework".

[b.6.i] TTAK.KO-06.0283 (TTA) "Reference Model and Operational Requirements for USN Resource Community".

[b.6.j] TTAK_KO-06_0167R1 (TTA) "USN Metadata Directory Service Interface Specification".

[b.6.k] TTAK_KO-06_0168R1 (TTA) "USN Metadata Model".

[b.6.l] TTAK_KO-06_0169R1 (TTA) "Standard Interface for Heterogeneous Sensor Networks".

[b.6.m] TTAK_KO-06_0170R1 (TTA) "USN Service Middleware Platform Reference Model".

[b.6.n] (TTA) "UM3 Protocol Recommendation" 2012-05_v01_03_03.

## B.7 TTC Documents

Not available at time of Publication.

# History

| **Approval history** | | |
|---|---|---|
| *V.1.0.0* | *<dd Mmm 2013>* | *<Milestone: Scheduled for TP#5- tbc>* |
| | | |
| | | |
| | | |
| | | |

| **Draft history** (to be removed on publication) | | |
|---|---|---|
| V.0.0.1 | 26 Oct 2012 | Rapporteur Input - Skeleton Draft |
| V0.0.2 | 28 Nov 2012 | Rapporteur Input -Draft; Including Contributions to Annex B and Annex C<br>oneM2M-REQ-2012-0037R01-Initial_architecture_discussion<br>oneM2M-REQ-2012-0042R02-Initial_architecture_Introduction,_TIA_TR-50 |
| V0.0.3 | 02 Dec 2012 | Rapporteur Input Draft for TP#2;<br>Editorial corrections only: Bullets, italics, spelling… plus several comments |
| V0.0.4 | 12 Dec 2012 | Output version from TP#2 |
| V0.0.5 | 13 Dec 2012 | Cut-Paste Corrections |
| V0.0.6 | 18 Feb 2013 | Rapporteur Input Draft for TP#3;<br>Including oneM2M-ARC-2013-0008R05-Intro_to_3GPP |
| V0.0.7 | 26 Feb 2013 | Updated Input Draft for TP#3;<br>Including oneM2M-REQ-2013-0101R01- -Transfer_of_ATIS_ARCH |
| V0.0.8 | 26 Feb 2013 | Updated Input Draft per WG2 discussion at TP#3<br>Title, scope text, Sect 12 intro changed |
| V0.0.9 | 26 Feb 2013 | Updated Input Draft per WG2 discussion at TP#3<br>Sect 12 intro revised |
| V0.1.1 | 12 Mar 2013 | Output Draft for WG2 after ARC#2 / TP#3, including:<br>oneM2M-ARC-2013-0012R01 - 3GPP2 (Section 12.2)<br>oneM2M-ARC-2013-0207R01 - 3GPP Security (Section 12.1.5)<br>oneM2M-ARC-2013-0142R03 - Pool Mapping (Annex A) |
| V0.1.2 | 17 Apr 2013 | Updated in WG2 ARC drafting session at TP#4 |
| V0.1.3 | 16 Jun 2013 | Progress Draft with ARC input from TP#5<br>oneM2M-ARC-2013-0248R06-open_M2M_Architecture - TTA (Section 10) |
| V0.1.4 | 20 Jun 2013 | Interim Output draft - ARC WG at TP#5<br>Editorial Changes |
| V0.1.5 | 26 Jun 2013 | Final Output draft - ARC WG at TP#5 - for Approval<br>Editorial Changes, References, and Bibliography |
| V0.1.5a | July 2013 | Clean-up done by ***editHelp!*** e-mail: mailto:edithelp@etsi.org |
| V0.2.0 | 28 Jul 2013 | Editorial Changes and editHelp suggestions<br>Final version for TP Approval |