**Source: ATIS**

**Title: Proposed Edits to oneM2M Scope**

**Agenda item: TBD**

|  |  |
| --- | --- |
| Decision | **X** |
| Discussion |  |
| Information |  |

**Contribution for:**

**ABSTRACT**

The proposed change suggests revising the Scope to reflect that the southbound API is defined by the lower layers and not by the Service Layer.

**Introduction**

This proposal suggests changes to the drafted Scope in order to clarify that oneM2M will not standardize southbound APIs.

**Rationale of the changes:**

The last bullet in the current drafted scope reads:

* Common use cases, terminal/module aspects, including Service Layer interfaces/APIs between:
	+ Application and Service Layers
	+ Service Layer and communication functions

But the lower levels should provide API to the upper levels and not the opposite way around. The southbound APIs between the Service Layer and the communication functions should therefore not be defined as part of the M2M Service Layer but by the access network standards, based on guidance given from the Service Layer. It is therefore suggested to revise the Scope to reflect this.

**Proposal**

It is proposed to review and agree the below suggested change to the oneM2M Scope.

**Suggested modifications to the oneM2M Scope and CoU as it currently exists in the CoU:**

**Scope**

The SDOs have reached a common understanding on the scope of oneM2M to include:

* Use cases and requirements for a common set of Service Layer capabilities
* Service Layer aspects with high level and detailed service architecture, in light of an access independent view of end-to-end services
* Protocols/APIs/standard objects based on this architecture (open interfaces & protocols)
* Security and privacy aspects (authentication, encryption, integrity verification)
* Reachability and discovery of applications
* Interoperability, including test and conformance specifications
* Charging aspects (charging data, not billing)
* Identification and naming of devices and applications
* Information models and data management (including store and subscribe/notify functionality)
* Management aspects (including remote management of entities)
* Common use cases, terminal/module aspects, including Service Layer interfaces/APIs between Application and common Service Layers.
* Description on how the interfaces between common Service Layer and communication functions provided by the lower layers are used.

With changes accepted, the last two bullets would read:

* Common use cases, terminal/module aspects, including Service Layer interfaces/APIs between Application and common Service Layers.
* Description on how the interfaces between common Service Layer and communication functions provided by the lower layers are used.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_