|  |  |
| --- | --- |
| Input Contribution | |
| Meeting ID\* | SEC#32 |
| Title:\* | Input on “Architecture Proposal” for the App-ID Registry Function TR-0048 |
| Source:\* | Wolfgang Granzow, Qualcomm, [wgranzow@qti.qualcomm.com](mailto:wgranzow@qti.qualcomm.com)  Phil Hawkes, Qualcomm, [phawkes@qti.qualcomm.com](mailto:phawkes@qti.qualcomm.com) |
| Uploaded Date:\* | 2017-11-13 |
| Document(s)  Impacted\* | TR-0048v0\_0\_2 |
| Intended purpose of  document:\* | Decision  Discussion  Information  Other <specify> |
| Decision requested or recommendation:\* | Include the text proposed in this contribution into TR-0048v0.0.2. |
| Template Version:23 February 2015 (Do not modify) | |

**oneM2M Notice**

The document to which this cover statement is attached is submitted to oneM2M. Participation in, or attendance at, any activity of oneM2M, constitutes acceptance of and agreement to be bound by terms of the Working Procedures and the Partnership Agreement, including the Intellectual Property Rights (IPR) Principles Governing oneM2M Work found in Annex 1 of the Partnership Agreement.

**Introduction**

The App-ID is defined in clause 7.1.3 of TS-0003v3\_7\_0 as follows:

### *Application Identifier (App-ID)*

*An Application Identifier (App-ID) uniquely identifies an M2M Application in a given context. More precisely, there are two types of App-ID: registration authority defined App-ID (registered App-ID) and non-registered App-ID. The establishment of the registered App-ID is guaranteed to be globally unique; the non-registered App‑ID is not guaranteed to be globally unique. The detail format is described in clause 7.2.*

|  |  |  |  |
| --- | --- | --- | --- |
| APP-ID | App-ID | App-ID is either registered with the M2M App‑ID Registration Authority or non-registered.  Registered App-IDs shall be in the format:  R{authority‑ID}.{reverseDNS}.{applicationName}  The {reverseDNS} part shall be a string value following 'reverse DNS notation', which is constructed in the reverse order of domain name components (see IETF RFC 1035 [i.7])  Non-registered App-IDs shall be in the format:  N{non-registered-App-ID}  Examples:   * Ra01.com.company.smartcity * Nk836-t071-fc022 | AE Registration Procedure described in clause 10.2.2.2.  The first character of the App-ID shall be a capital letter of ‘R’ for registered and ‘N’ for non-registered. |

The present concept of App-ID Registry just enables manufacturers/application developers to issue App-IDs according to their own preferences on terminology, within the constraints imposed by the format definition given in the above table. The central App-Registry only ensures that registered entities are globally unique.

The objective of WI-0073 and TR-0048 is to investigate mechanisms to enhance the current App-ID Registry function with application profile data to enable the oneM2M system to automatically enrol the connecting application.

In this contribution following potential enhancements to the App-ID Registry function are addressed:

1) Providing proof that a registered application/App-ID has passed compliance testing and certification.

2) Defining and associating information about which specific application functions have been tested, which specific functions are supported and intended to be used by the application (for instance which resource types are created by the application), and constraints on operations on resources (read only, creation, read and update etc.)

3) Bundles of functions could be identified in terms of Application Class identifiers, which could be added as App-ID meta information.

4) Possible role of the App-ID Registry to support subscription management on the IN-CSE (i.e. creation of service profiles indicating functions permitted by a service subscription.

5) Possible role of the MEF in communication procedures between AEs and the App-ID Registry.

Any information which may be relevant to an M2M service provider with regard to e.g. network/resource management, service subscription management, charging and accounting procedures could be captured in application profiles supplied as meta data to an App-ID.

For instance, a simple temperature sensor which reports periodically temperature measurements from a smart home environment would have a different App-Id than an AE implemented as an app on a smart phone, which can read data reported from that temperature sensor and of other sensors of a given smart home service subscriber.

*======== start of proposed text 1 =============================*

6.x Architecture Proposal For Management of Application Profiles

6.x.1 Functionality Requirements

The following is a list of basic requirements to be considered when specifying management of Application Profiles, and a list of design assumptions used for the proposed architecture solution.

An Application Profile refers to a set of information which characterizes an application in a compact and meaningful way for any stakeholder of an M2M system.

Requirements:

* In the M2M infrastructure a function is required which enables manufacturers and application software developers to register identifiers which characterize their products in a way meaningful to M2M network operators, M2M Service Providers, M2M Application Services Providers and M2M service subscribers:
  + an M2M Network Operator (i.e. operator of the underlying Wide Area Network, e.g. a mobile network) needs to know the characteristics (volume, distribution, message size) of the traffic created by an application
  + an M2M Service Provider (i.e. operator of the IN-CSE and associated infrastructure) needs to know the resource types created by an application, the expected amount of data volume created, the frequency and distribution of transactions triggered. The IN-CSE must be enabled to authenticate any identifiers, including security credential identifiers, App-IDs, to authenticate and assign AE-IDs, and to associate identifiers with each other such as device, node, entity and application identifiers. It also must check if operations requested by an AE are permitted in terms of the application user’s service subscription.
  + M2M Application Services Providers (i.e. the provider of value added services beyond plain Service Layer functions, typically the stakeholder who maintains a service agreement with both the M2M service subscribers and the M2M Service Provider). Examples include e.g. service providers of smart home services, Intelligent Transport Services (ITS), or eHealth services.
  + M2M service subscribers (i.e. individual end users who own M2M field devices or an organization which operates the M2M field system on behalf of the actual owner) may need to be enabled to discover other applications in order to interoperate and create added value
* An M2M system cannot rely on just a single central App-ID registry. The architecture must allow that multiple App-ID registries can coexist in the system preferably without need for communication between them

Design assumptions:

* The App-ID Registry represents an infrastructure function which e.g. like the M2M Enrolment Function (MEF) may be operated by third party (i.e. a special M2M Application Services Provider) which configures information related to applications (Application Profiles) on the IN-CSE. This includes resources accessible via Mca and Mcc reference points such as <*m2mServiceSubscriptionProfile*>, <*serviceSubscribedNode*> <*serviceSubscribedAppRule*> and possibly in addition data which should not be exposed over Mca and Mcc. This implies that an App-ID Registry could include the functionality of an IN-AE for communication with the IN-CSE.
* The architecture solution considered aims to avoid direct communication between field entities and the App-ID registry, in order to not burden field devices to support another client and interface for communication with the App-ID registry. The solution described here would use the MEF as a proxy for any required communication between field entities and the App-ID Registry.
* For assignment and registration of App-IDs it is assumed that the procedures as defined today are continued to be used: the manufacturer / application developer proposes App-ID and the App-ID Registry validates uniqueness. The registered App-IDs is provisioned to the field device before it becomes operational in the M2M system.

### 6.x.1 Architecture

The proposed architecture solution is illustrated in Figure 6.x.1-1. The figure shows the functions and entities potentially having a role when considering the App-ID Registry. Solid lines define communication paths. These are referenced with a number in the figure and described as follows:

1) The manufacturer creates a digital certificate for his device, or an application software certificate, issuing a Certificate Signing Request (CSR) to a Certification Authority (CA). The certificate is initially installed on the field entity at manufacturing stage or software download/installation and used for Security Association Establishment with a MEF. The MEF-FQDN (i.e. the MEF ID) is also preconfigured on the device SW module.

2) Communication path for CSRs issued by the operational field node and communicated to the CA via the MEF (acting as PKI Registration Authority).

3) Communication path for CSRs issued by a Compliance Certification Body (CCB). In the proposed architecture, the CCB is assumed to issue a digital certificate upon having passed certification test successfully. This certificate is also installed on the field entity and used as electronic proof that the device or software product is oneM2M certified. It includes all relevant identifiers, App-ID (which is assumed to be specific manufacturer/developer and allowing its identification), identification of the device (if applicable), identification of the software instance (if applicable). Details of these identifiers are to be defined.

4) Communication path for CSRs issued by a Device Management (DM) Server. This function may be needed to ensure secure communication between a DM server and a DM client on the field node using OMA or BBF defined protocols (as defined in TS-0005, TS-0006 and TS-0022). This interface is out of scope of this document.

5) This refers to the communication path between manufacturer/developer and CCB to obtain the certificate described at no. 3) above. Details of this interface are out of scope of the present discussion.

6) The manufacturer/developer obtains the registered App-ID on this communication link from the App-Id Registry. After the manufacturer/developer has obtained a CCB-issued certificate which includes this registered App-ID, it returns this certificate to the App-Id Registry as proof of successful compliance registration. This information is stored as metadata in the Registry.

7) This refers to delivery of the device/software to the end customer (if the AE is implemented as Software module, it could be downloaded e.g. from a web store).

8) The manufacturer needs to agree with the operator of the MEF (provisioned to the device or SW module) on an applicable Security Association Establishment Framework (SAEF). Assumption in the following description is that certificate-based SAEF is used.

9) When the application is started for the first time, the MEF client contacts the MEF to receive security credentials and configuration data needed for registration to its registrar.

10) The MEF may trigger the field device to perform a DM procedure.

11) The DM server can configure the MEF to employ a specific authentication profile when communicating with a MEF client.

12) The DM server can be triggered by the IN-CSE to perform DM procedures with DM client on the field device, e.g. to configure CMDH policies.

13) If required, the MEF client on the field device could send information over communication path 9) which triggers a transaction between the MEF and the App-ID Registry. This feature would represent an extension to the currently given functionality on the MEF.

14) The App-ID Registry can send requests to the IN-CSE and receive responses. The App-ID would behave like a IN-AE on this communication path, i.e. Mca procedures would be applicable.

15) If the authentication profile applicable to the field device for registration indicates use of MAF-based SAEF, the field device establishes symmetric key credentials with a MAF. If certificate-based authentication between the field device and its registrar is configured to be used, the MAF is not involved.

16) When MAF-based SAEF is used, the registrar needs to retrieve the symmetric key credentials from the MAF, when the registree contacts the registrar.

17) When symmetric key credentials are assigned by the MEF for end-to-end security frameworks, the respective end-node needs to retrieve these credentials from the MEF (end-to-end security with symmetric key credentials could also be done with MAF procedures).

18) The field node may contact the IN-CSE either directly or via a Middle Node CSE. In the above descriptions it is assumed that a Middle Node is not present. The Middle node needs to register first to its own registrar (IN-CSE Figure 6.x.1-1, before the AE registers to the MN-CSE.

19) This communication path could be used to exchange information between App-ID Registry and CCB, for instance to define sets of testable functional features associated with Application Profiles for which compliance is certified by the CCB.



Figure 6.x.1-1: Nodes and communication interfaces of the proposed architecture

Summary of new features supported by the proposed solution:

* Validation if application has been certified by CCB by using digital certificates. This could be done by the App-ID Registry using communication via MEF, or by the IN-CSE (at registration, additional functions to be defined)
* CCB certificates could include information on tested functionality, which converts into service layer functions permitted to be used. These functions represent the *Application Profile*.
* App-ID Registry could act as IN-AE(s) towards IN-CSEs.
* For end.to-end security frameworks, App-IDs could be revoked by revoking App-ID certificates. New App-IDs could be assigned by installing new App-ID certificates on the AE, using MEF procedures triggered by the App-ID Registry function. App-ID certificates could use the same private/public key pair as the AE's certificate. Once the AE's Certificate establishes a security association with the Registrar, the AE could create "AppIDCert" resources under its <AE> resources on the Registrar in order to have AppIDs associated with the AE. (This could be made flexible by allowing the AE to create the certificates there and then allow activating/deactivating an individual AppID).

#### 6.1.1.1 Nodes

The nodes of the overall proposed architecture are shown in Figure 6.x.1-1. Except for the App-ID Registry and CCB, all nodes are already specified in oneM2M Release 2A. The functionality of MEF clients and the MEF would require extensions. Some new resource types (e.g. on IN-CSE for App-ID metadata) and/or extensions to existing resource types (e.g. <AE> resource type) would need to be defined.

#### 6.1.1.2 Reference Points

The new reference points to be considered for standardization are the communication links of the App-ID Registry, marked with numbers 6), 13), 14 and 19). 6), and 13) may not need to be standardized. The communication link 14) between App-ID Registry and IN-CSE could be implemented compliant with the Mca reference point.

*======== End of proposed text 1 =============================*