RFC 236 - OSGi Interworking Service for oneM2M Networks

Draft

10 Pages

Abstract

Defines the OSGi interworking service for oneM2M networks that exposing the exists OSGi-based devices to the oneM2M network.

0   Document Information

0.1   License

DISTRIBUTION AND FEEDBACK LICENSE, Version 2.0

worldwide, perpetual, irrevocable, royalty-free copyright license to copy, publish, license, modify, sublicense or otherwise distribute and exploit your Feedback for any purpose. Likewise, if incorporation of your Feedback would cause an implementation of the Distribution, including as it may be modified, amended, or published at any point in the future ("Future Specification"), to necessarily infringe a patent or patent application that you own or control, you hereby commit to grant to all implementers of such Distribution or Future Specification an irrevocable, worldwide, sublicenseable, royalty free license under such patent or patent application to make, have made, use, sell, offer for sale, import and export products or services that implement such Distribution or Future Specification. You warrant that (a) to the best of your knowledge you have the right to provide this Feedback, and if you are providing Feedback on behalf of a company, you have the rights to provide Feedback on behalf of your company; (b) the Feedback is not confidential to you and does not violate the copyright or trade secret interests of another; and (c) to the best of your knowledge, use of the Feedback would not cause an implementation of the Distribution or a Future Specification to necessarily infringe any third-party patent or patent application known to you. You also acknowledge that the OSGi Alliance is not required to incorporate your Feedback into any version of the Distribution or a Future Specification.

I HEREBY ACKNOWLEDGE AND AGREE TO THE TERMS AND CONDITIONS DELINEATED ABOVE.

0.2   Trademarks

OSGi™ is a trademark, registered trademark, or service mark of the OSGi Alliance in the US and other countries. Java is a trademark, registered trademark, or service mark of Oracle Corporation in the US and other countries. All other trademarks, registered trademarks, or service marks used in this document are the property of their respective owners and are hereby recognized.

0.3   Feedback

This document can be downloaded from the OSGi Alliance design repository at https://github.com/osgi/design The public can provide feedback about this document by opening a bug at https://www.osgi.org/bugzilla/.

0.4   Table of Contents

0.5   Terminology and Document Conventions

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY" and "OPTIONAL" in this document are to be interpreted as described in 10.1.

```
Source code is shown in this typeface.
```

0.6   Revision History

The last named individual in this history is currently responsible for this document.

Revision	Date	Comments
Initial	*June 21 2017*	*Initial Document* *Seven Gan, Huawei, seven.ganlu@huawei.com*

1   Introduction

oneM2M is a global organization that creates requirements, architecture, API specifications, security solutions and interoperability for Machine-to-Machine and IoT technologies. Its specifications provide a framework to support a wide range of applications and services such as smart cities, smart grid, connected car, home automation, public safety, and health. As an increasing number of big players participating in oneM2M standard, it is becoming more and more popular and is likely to become one of the most mainstream IoT standards in the world. Integrating OSGi-based devices into oneM2M network can promote the application of OSGi technology in IoT Scenario.

This RFC provides a solution to expose the existing OSGi-based devices to the oneM2M network.

2 Application Domain

When adding a OSGi-based device (such as IoT gateways, enhanced capability devices) into oneM2M network, it requires the oneM2M protocol communication support of these devices. So that these OSGi devices and their connected sensors can be remotely discovered and operated on the oneM2M IoT platform as shown in Fig 1.
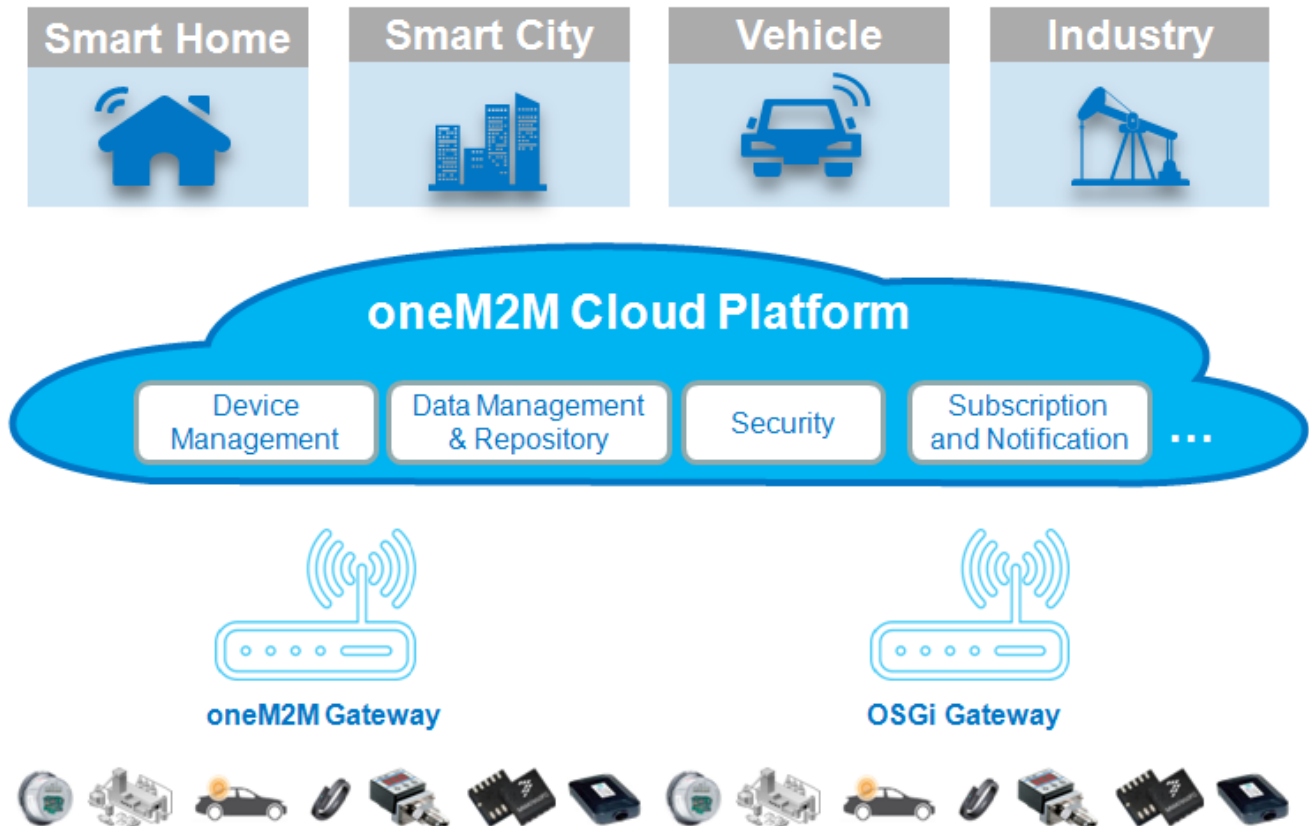
Fig 1

3 Problem Description

OSGi defines Device Abstraction Layer which provides a unified interface for OSGi application developers to interact with sensor, devices, etc. And also defines Configuration Admin service which allows operators to configure deployed bundles. Other OSGi standard device services (e.g. UPnP Device Service, EnOcean Device Service, Zigbee Device Service, SDT Device Service) and OSGi-based open source projects (e.g. Eclipse

SmartHome, Kura) have the similar mechanisms. It is impossible to integrate these devices into oneM2M network by re-developing them as fully oneM2M compliance devices. The better way is to develop an oneM2M protocol interworking service which can interact with these device services and configuration admin service, expose these devices and configurations as oneM2M resources to oneM2M network without affecting the exist implementation of OSGi devices.

There are two interworking scenarios. One is to consider OSGi gateways as an oneM2M AE who communicates with CSE through Mca reference point, as shown in Fig 2. The other one is to consider OSGi gateways as an oneM2M CSE who communicates with other CSE through Mcc reference point and communicates with AE through Mca reference point, as shown in Fig 3.

The interworking service requested in this RFC would give developers a common way of developing such service.

Fig 2

Fig 3

4 Requirements

R1: The solution MUST provide a standard service for exposing Device service and Configuration Admin service as oneM2M resources to oneM2M network without any modification of Device Service and Configuration admin service.

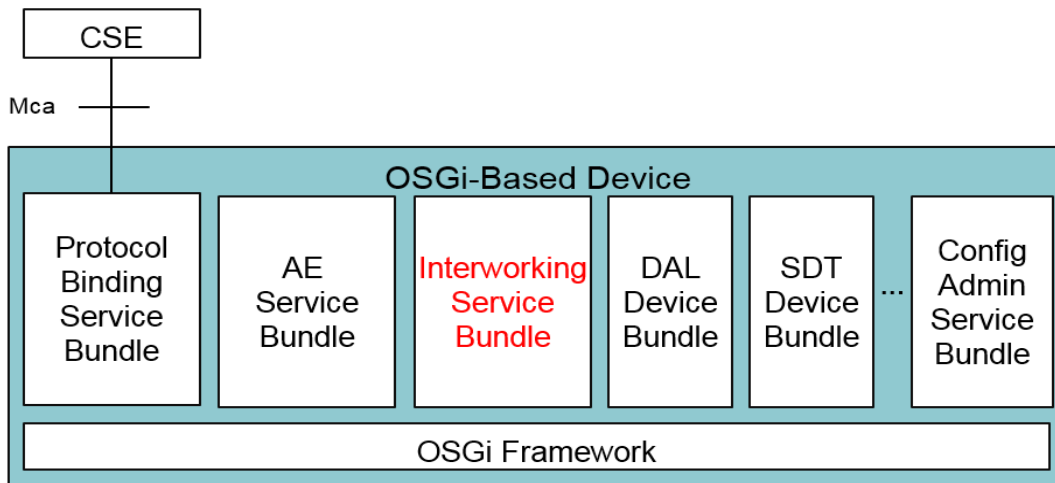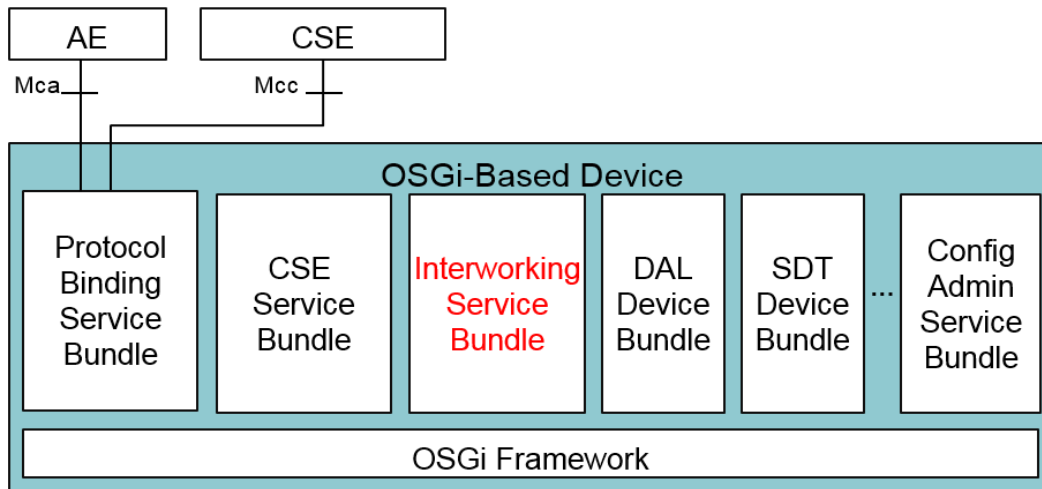R2: The solution MUST *provide a set of rules to map between Device service and oneM2M resource for Interworking service developers.*

R3: The solution MUST *provide a set of rules to map between Configuration Admin service and oneM2M resource for Interworking service developers.*

R4: The solution MUST *define API for configuring the Interworking service.*

R5: The solution MUST be *independent of specific transport protocol between OSGi devices and oneM2M devices.*

5 Technical Solution

5.1 Interworking Service

oneM2M CSE/AE function should be firstly supported on OSGi gateway if you want to integrate your devices into oneM2M network. For the oneM2M side, it interacts with external oneM2M entities through Mcc or Mca reference point. For the OSGi side, it provides the interface to execute the oneM2M resource operations for other bundles. The technical solution of OSGi based oneM2M CSE and AE function should be defined in Service Layer API specification. This RFC is to define the interworking service that converts OSGi device to oneM2M resource, so that supporting oneM2M protocol would not affect existing systems.

OSGi device service mechanism is similar with oneM2M resource, each device operation can map to an oneM2M resource operation:

OSGi Device	oneM2M Resource
Register service	Create resource
Unregister service	Delete resource
Modify service property	Update resource
Generate device function event	Create resource

The interworking service listens to the service registry, obtains device service objects and converts them to oneM2M resource structure. Then call the oneM2M CSE/AE function interface to create/modify/delete resources on oneM2M side. It also provides the service interface for oneM2M CSE/AE function to handle the operation request from oneM2M side, convert resource operation to device operation, as shown in Fig 4.
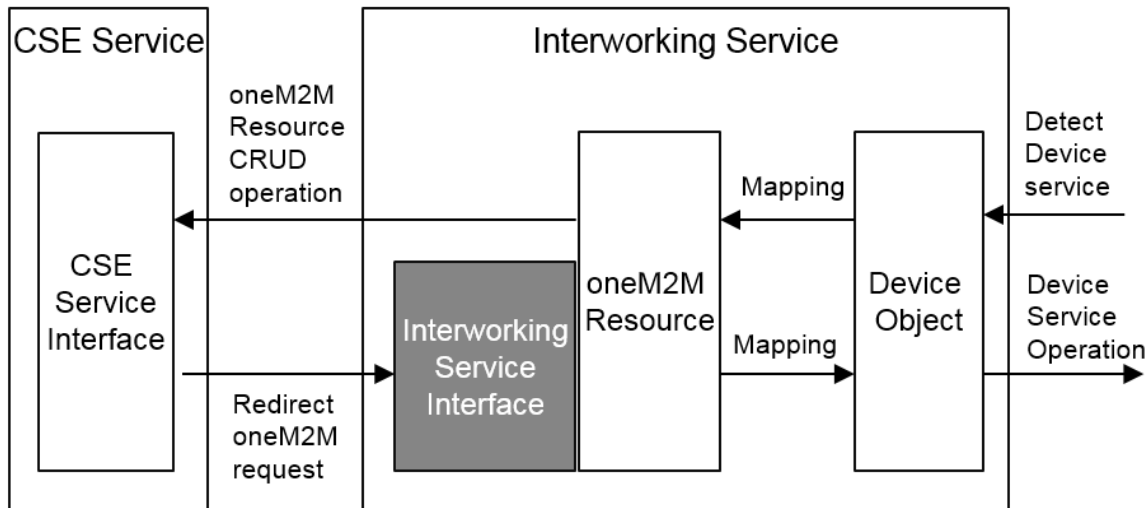

Fig 4

The attribute point of access is used for marking resources created by different interworking services, so that CSE/AE function can redirect the request to the resource creator.

```
public interface InterworkingService {

        public static final String POINT_OF_ACCESS = "onem2m.resource.poa";

        public ResponsePrimitiveDTO doExecute(RequestPrimitiveDTO request);
}
```

6   Data Transfer Objects

RFC 185 defines Data Transfer Objects as a generic means for management solutions to interact with runtime entities in an OSGi Framework. DTOs provides a common, easily serializable representation of the technology.

For all new functionality added to the OSGi Framework the question should be asked: would this feature benefit from a DTO? The expectation is that in most cases it would.

The DTOs for the design in this RFC should be described here and if there are no DTOs being defined an explanation should be given explaining why this is not applicable in this case.

This section is optional and could also be provided in a separate RFC.

7 Javadoc

Please include Javadoc of any new APIs here, once the design has matured. Instructions on how to export Javadoc for inclusion in the RFC can be found here: https://www.osgi.org/members/RFC/Javadoc

8 Considered Alternatives

For posterity, record the design alternatives that were considered but rejected along with the reason for rejection. This is especially important for external/earlier solutions that were deemed not applicable.

9 Security Considerations

Description of all known vulnerabilities this may either introduce or address as well as scenarios of how the weaknesses could be circumvented.

10 Document Support

10.1 References

[1].   Bradner, S., Key words for use in RFCs to Indicate Requirement Levels, RFC2119, March 1997.

[2].   Software Requirements & Specifications. Michael Jackson. ISBN 0-201-87712-0

[3].   oneM2M Release 2 specifications, TS 0001 Functional Architecture v2.10.0

*Add references simply by adding new items. You can then cross-refer to them by chosing <Insert><Cross Reference><Numbered Item> and then selecting the paragraph. **STATIC REFERENCES (I.E. BODGED) ARE NOT ACCEPTABLE, SOMEONE WILL HAVE TO UPDATE THEM LATER, SO DO IT PROPERLY NOW.***

10.2 Author's Address

Name	
Company	
Address	
Voice	
e-mail	

10.3 Acronyms and Abbreviations

10.4 End of Document