RFP 177 - IoT Protocols

Draft

8 Pages

Abstract

In the IoT domain there is a widespread of communication protocols available for letting devices interact with eachother. Protocols for energy efficient wireless communication, protocols for remote device management, protocols for application development. This RFP focuses on the latter, describing ways to integrate protocols such as MQTT and CoAP with OSGi. The goal is twofold. First, allow low-end devices that are unable to run a complete OSGi stack communicate with an OSGi framework using these protocols. Second, allow OSGi frameworks to leverage these IoT protocols to be used in Distribution Providers and for distributed eventing.

0 Document Information

0.1 License

DISTRIBUTION AND FEEDBACK LICENSE, Version 2.0

The OSGi Alliance is willing to receive input, suggestions and other feedback ("Feedback") on the Distribution. By providing such Feedback to the OSGi Alliance, you grant to the OSGi Alliance and all its Members a non-exclusive, non-transferable, 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 6.1.

`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	*October 9 2015*	*Initial contribution* *Tim Verbelen, iMinds – Ghent University, tim.verbelen@intec.ugent.be*

1 Introduction

Internet of Things (IoT) is becoming an important application domain of OSGi. The ability to run an OSGi framework on a gateway device as well as a Cloud server, together with the ability of transparently calling remote services using distributed OSGi makes it perfect base for an IoT platform. However, the proliferation of IoT protocols makes it difficult to integrate the many technologies available. This RFP seeks a proposal to hide away a lot of the IoT protocols, leveraging the clean and simple OSGi APIs while still being able to interface with external systems.

2 Application Domain

In the current IoT domain we see a proliferation of different protocols for device access, remote management and IoT applications.

Various device access protocols are defined to interface with small (wireless) sensor devices, such as enOcean, ZigBee, Z-Wave, etc. Also for device management different protocols exist such as OMA DM, TR-69, etc. While these protocols are already being handled in the OSGi specification (103 – Device Access, 117 – Dmt Admin, 141- Device Abstraction layer, and specific protocol adapters for enOcean, TR069, …), there are also many IoT protocols that are widely used on an application level.

Two of the more popular application-level IoT protocols nowadays are the service-based CoAP and the event based MQTT.

2.1 CoAP

Constrained Application Protocol (CoAP) is standardized by the IETF and is intended to provide a lightweight protocol for machine to machine communication. It is a RESTful protocol that was designed to provide transparent mapping to HTTP. Devices (also called “Resources” in CoAP) are made available through URLs, and clients access these using GET, PUT, POST and DELETE methods. It uses UDP as transport layer in order to limit bandwidth and overhead.

2.2 MQTT

MQTT is a lightweight publish/subscribe messaging protocol on top of TCP/IP standardized by OASIS. MQTT uses a central broker where all clients send their messages to and where clients can subscribe on certain topics.

3 Problem Description

Although there are already solutions for using MQTT and/or CoAP within an OSGi application, these still leave it up to the developer to use the specific APIs of the MQTT/CoAP library used, which is currently not standardized in OSGi. Also, this could leverage the existing OSGi specifications regarding RemoteServiceAdmin and (distributed) EventAdmin.

Concretely, this RFP seeks a solution to:

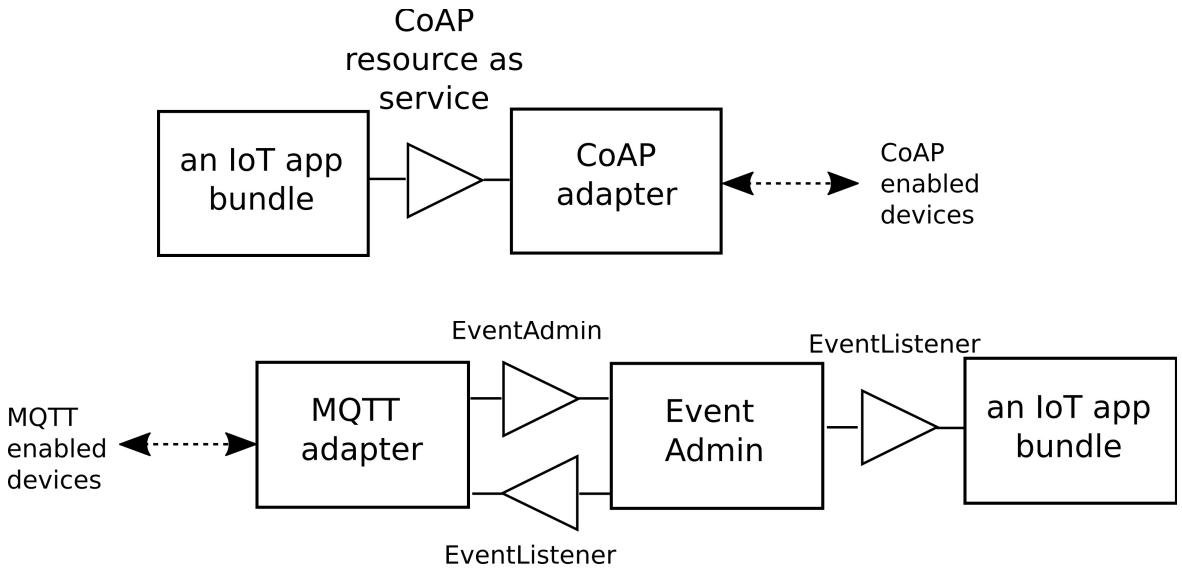
- allow to import CoAP resources as OSGi services in an OSGi framework and to export OSGi services in an OSGi framework as CoAP resources

- use the (distributed) EventAdmin API to communicate via MQTT events with external devices

4 Use Cases

4.1 Integrate with small, embedded devices unable to run OSGi

Many devices are powered with a small microcontroller (e.g. Arduino-like) that is unable to run a full OSGi stack, but often does support communicating over IoT protocols such as CoAP or MQTT. By providing a nice integration with these protocols, we can seamlessly integrate such small devices in an OSGi application. For example, CoAP resources could be exposed as OSGi services, and MQTT messages could be made available as OSGi Events.



4.2 MQTT as distributed eventing mechanism between OSGi frameworks

An adapter that translates MQTT events to OSGi events through EventAdmin and vice versa could provide a mechanism for distributed eventing in OSGi.

TODO: relation with RFP 158

4.3 Use CoAP as Distribution Provider for OSGi

The RESTful design of CoAP could also allow it to be used as Distribution Provider for a RemoteServiceAdmin allowing to execute remote service calls via the CoAP protocol.

5 Requirements

5.1 CoAP

- C0010 – The solution MUST provide a Distributed OSGi configuration type name
- C0020 – The solution MUST define a mapping from a Java interface to a CoAP resource and vice versa.
- C0030 – The solution MUST provide a type safe way for parameters on the CoAP POST/GET methods to be made available to implementation methods.
- C0040 – The solution MAY provide a mechanism to discover CoAP endpoints and their metadata

5.2 MQTT

- M0010 – The solution MUST define a mapping from an MQTT topic to an EventAdmin topic
- M0020 – The solution MUST define a mapping from an MQTT message to an OSGi Event
- M0030 – The solution SHOULD provide a mechanism to define which OSGi Events should be sent to the MQTT broker

6 Document Support

6.1 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

6.2 Author's Address

Name	Tim Verbelen
Company	iMinds, Ghent University
Address	Gaston Crommenlaan 8 box 201, 9050 Ghent, Belgium
Voice	
e-mail	tim.verbelen@intec.ugent.be

6.3 End of Document