RFC 20 Remote management architecture

Confidential, Draft

18 Pages

Abstract

This document describes a top-level remote management architecture for OSGi and is a request for comments (RFC) on this architecture. The architecture provides a framework for further work in different areas of remote management. Finding more detailed requirements and design for each area is deferred to studies of these different areas.

0 Document Information

0.1 Table of Contents

0.2 Status

This document specifies a remote management architecture for the Open Services Gateway Initiative, and requests discussion and suggestions for improvements. Distribution of this document is unlimited within OSGi.

All Page Within This Box

0.3 Acknowledgement

Any acknowledgments the authors feel are necessary.

0.4 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 [1].

`Source code is shown in this typeface.`

0.5 Revision History

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

Revision	Date	Comments
Draft A	2001-02-21	Initial draft. Tomas Bornefall, Remote management Expert Group Chair, OSGi. rmeg-chair@mail.osgi.org.
Draft B	2001-07-02	First published draft.
Draft C	2001-09-03	Moved some overview technical discussion from chapter 2 “Motivation” and Rationale” to chapter 3.1 Technical discussion – overview”. Added a picture on the proposed parts of a remote management specification to chapter 3.4. Removed some detail on initial provisioning from chapter 3.5 – 3.7, deferring such discussions to a specific RFC, e.g. RFC 27. Some other minor changes.

1 Introduction

The OSGi standard version 1.0 lacks a architecture for remote management and large-scale deployment of service gateways. There are APIs and services defined or under definition that provides means for life-cycle management and configuration of services and devices as well as management of users, their permissions and preferences. But there is no standardized way to access them remotely.

This is a request for comments on an OSGi remote management architecture providing the means for remote access to the locally available manageability. The architecture considers large-scale deployment issues, the security of management operations and allows additional manageability to be added later.

This architecture defines a framework for OSGi remote management but requires more detailed requirements analysis and design work during further work with different parts of the architecture.

<I use these where I e.g. have several wording options. >

All Page Within This Box

2 Motivation and Rationale

The concept of a service gateway includes a care-free usage by the end-user, allowing him or her to take decisions on a service subscription level, rather than worry about operating systems versions, which revisions of drivers are needed, conflicting dependencies between different software packages etc. The service gateway is also expected to have a high availability and deliver services with a specified level of determinism. For gateways operated by a gateway operator, these requirements lead to a need for remote management and monitoring of the gateway with services and devices.

3 Technical Discussion

3.1 Overview

The OSGi management architecture shall meet a number of requirements, defined in the RFP, [2]. These requirements have been copied to section 3.2 in this RFC, Assumptions and requirements. The management architecture itself will allow more detailed requirements to be defined for further work with different parts of the architecture.

Being a standard and not a design for a specific system, the remote management architecture of OSGi has to consider and support a number of different scenarios.

Starting from the production line, the gateway might or might not be pre-configured for a certain operator, operating a certain kind of management system. A not configured gateway might be configured on its way to the operational site where it is installed. And still other gateways have to be adapted to a specific operator and management system when they become activated at their operational site.

The gateway, by definition, is connected to a network, which often is an always on or dial up IP based WAN (Wide Area Network). The network might or might not have special provisions for service gateways. Support for the gateway to find it's management system could be one examples of such service gateway related functionality in the WAN. Another example is that external request and messages can be routed to the gateway. On the other hand, gateways might be introduced in currently available WANs that sees them as any clients, like PCs. Therefore the remote management architecture has to consider router and firewall configuration for different protocols etc. in widespread use today.

Other kinds of gateways might have intermittent IP connectivity and sometimes has to rely on non-IP networks, intermittently with no connection at all. This is e.g. the situation for vehicle gateways.

3.2 Assumptions and requirements

3.2.1 Assumptions

3.2.1.1 Core framework functionality

The OSGi remote management architecture can assume that, independent on programming language, the service gateway have a framework with the same basic service life cycle management as the current Java based OSGi framework. The rationale for this assumption is that OSGi is about deployment and execution of software based services. OSGi service gateways will therefore always have some mechanism to load software from a remote location and to execute that software.

3.2.1.2 Security architecture

The OSGi remote management architecture can assume that the Java security mechanisms used by OSGi is available on the gateway. If there will be support in OSGi for other languages than Java, it is assumed that similar security architecture will be available or possible to implement. The rationale for this assumption is that such security mechanism is e.g. needed to allow a framework to check the origin and correctness of downloaded software.

3.2.1.3 Communication

The OSGi remote management architecture can assume that communication services needed for interaction with a management system as well as for bundle transport are either:

- Factory pre-installed on a gateway.
- Available for download and installation with some specified, enumerated list of protocols, e.g. HTTP during some phase of the deployment cycle. This list of protocols shall be supplied in more detailed RFPs/RFCs.

Communication services are not defined as a part of the remote management architecture.

3.2.2 Compatibility requirements

3.2.2.1 Management protocols

The remote management architecture shall be adaptable to different management protocols. Which protocols and what level of compatibility will be handled in the OSGi RFP – RFC process for each protocol mapping specification.

3.2.2.2 Transport protocols

Deleted, see 3.2.3.

3.2.2.3 Support for existing WANs

The remote management architecture shall not require any special WAN features others than those available for a PC connecting to a WAN using e.g. cable TV or XDSL modems. For non-IP networks a similar requirement applies, that is existing network functionality shall be enough for remote management operations.

3.2.2.4 Support for service gateway aware WANs

A WAN might have additional support for e.g. service discovery protocols enabling plug and play activation of a service gateway. The remote management architecture shall allow benefit from such support.

3.2.3 Deployment requirements

These requirements are based on the need to support different deployment scenarios:

- The service gateway is pre-configured for a certain operator and management system before installation.
- The service gateway is configured for an operator and management system at installation by user action.
- The service gateway is configured for an operator and management system at installation without user action.

There are also some other requirements in this chapter, related to deployment.

3.2.3.1 Pre-configuration for a certain operator and management system

The remote management architecture shall support that the service gateway is pre-configured for a certain operator, operating a certain kind of management system before installation at the operational site

3.2.3.2 On-site adaptation to a certain operator and management system, user initialized

The remote management architecture shall support that the service gateway is adapted to a specific operator and management system during activation at their operational site, initialized by user or installation personnel.

3.2.3.3 On-site adaptation to a certain operator and management system, Gateway PnP

The remote management architecture shall support that the service gateway is adapted to a specific operator and management system during activation at their operational site, initialized by the gateway without other user interaction than physical installation and power on. Of course the business relation between the user and the operator has to be set up in some way.

3.2.3.4 On-site adaptation to a certain operator and management system, Mgmt PnP

The remote management architecture shall support that the service gateway is adapted to a specific operator and management system during activation at their operational site, initialized by a management system.

3.2.3.5 Fallback to redundant management system

The architecture shall allow implementations with a capability for a gateway to fall back to a redundant management system.

3.2.3.6 Hand-over to another operator or another kind of management system

The architecture shall allow implementations with a capability for a gateway to be handed over to another operator or another kind/brand of management system.

3.2.3.7 Name space

The remote management architecture shall define name space(s) with unique names for service gateways. This shall be consistent with naming used for communication. <Should this be in a ”communications” chapter? >

3.2.4 Manageability requirements

The remote management architecture shall ensure manageability of gateway platforms, services, users and devices using existing OSGi APIs and services as well as finding requirements on new APIs and services. If needed, the remote management architecture can specify additional “convenience APIs”, easing the task to implement a management agent. This manageability shall include at least:

3.2.4.1 Inventory asset information

Provide asset information about the gateway, services and devices e.g. serial numbers, version etc.

3.2.4.2 Life cycle management

Life cycle management (install, update, start, stop and uninstall) for bundles and services.

3.2.4.3 Monitoring

The remote management architecture shall enable monitoring of usage statistics and resource allocations.

3.2.4.4 Events

The remote management architecture shall enable event listeners and event generation.

3.2.4.5 Control and configuration

The remote management architecture shall enable control and configuration of the gateway and it’s users, permissions and services. It shall allow a standardized way to bundle meta data as well as default configuration data with a bundle.

<What about devices? >

3.2.4.6 Gateway maintenance

The architecture shall consider maintenance of the gateway platform software, such as operating system, interpreters and service execution framework.

3.2.4.7 Extendable architecture

The architecture shall be future proof by easily allowing extensions and control of new framework and service features.

3.2.5 Implementation constraints

3.2.5.1 Footprint and cost

The remote management architecture must allow for low footprint implementation for cost optimized gateways.

3.2.5.2 Execution environment

Language specific remote management APIs shall follow the execution environment profiles specified for OSGi service gateways. E.g. a remote management Java API executed on a Java framework shall not use any classes other than those specified in the minimal Java profile for service gateways.

3.2.5.3 Scalability and large scale deployment

To allow large-scale deployment, management systems must be able to scale. The remote management architecture shall make such scaling feasible.

3.3 Overview of the remote management architecture

This architecture is build on the following basic ideas to fulfill the requirements stated in chapter 3.2:

1. The architecture tries to use the fact that OSG is not about execution of managed services on any legacy execution framework. In whatever language the services and framework is written, assumption 3.2.1.1 says that the main life cycle management characteristics will be similar to the current Java based framework.

2. The execution framework provides mechanisms for downloading software services from remote locations in a secure way (assumptions 3.2.1.1 and 3.2.1.2). The architecture sees a remote management agent as such a potentially downloadable service.

3. Existing and coming “local” life-cycle management and configuration services will be used for remote management.

4. Support for different management protocols can be added step by step, even on already deployed gateways by using 1 and 2 above.

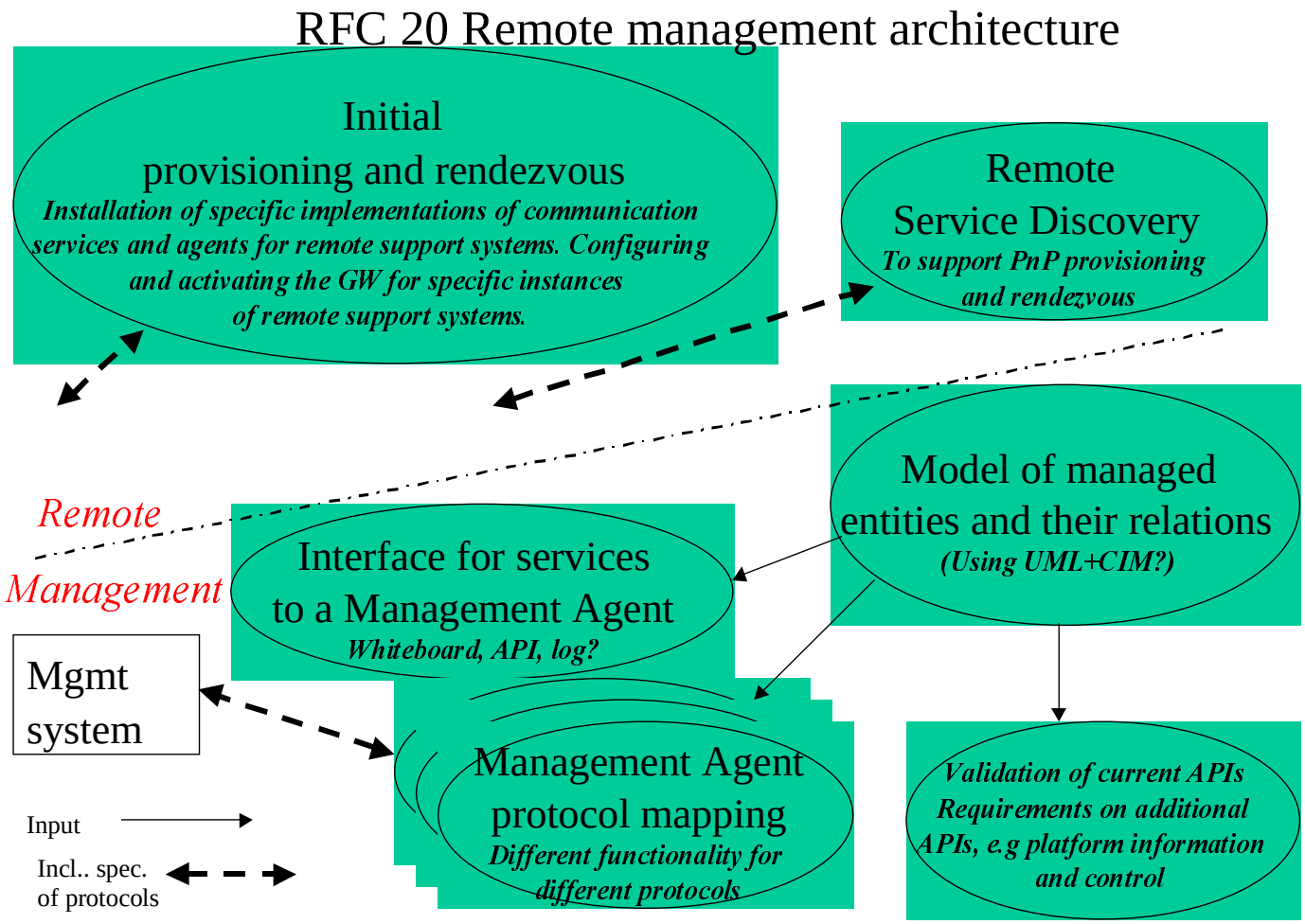
5. The architecture tries to partition the architecture into different services, each useful in itself. The architecture also proposes some new services in the vicinity of the remote management core area.

3.4 The remote management architecture

The remote management specification can include the following parts::

- A standard for staging a gateway, binding to an operator and downloading an initial management bundle.
- A way to provide default configuration for a downloaded bundle.
- Standards for how to map different management protocols to manage OSG service platforms.

Possible extensions include: External Service Discovery used to discover external systems providing e.g. management services, Means for other services to trig management actions, monitoring and control of the underlyinh H/W and O/S platform.



All Page Within This Box

4 Security Considerations

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

5 Document Support

5.1 References

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

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.***

5.2 Author's Address

Name	
Company	
Address	
Voice	
e-mail	

5.3 Acronyms and Abbreviations

5.4 End of Document

All Page Within This Box