Management Using Web Services: Architecture

Working Draft 01, 5 November 2003

Document identifier:
WSDM-MUWS-Architecture-Draft

Location:

Editors:
John DeCarlo, The MITRE Corp. (jdecarlo@mitre.org)
Zulah Eckert, Hewlett-Packard, (zulah_eckert@hp.com)

Contributors:
Igor Sedukhin, Computer Associates (Igor.Sedukhin@ca.com)

Abstract:

Status:

Table of Contents

Deleted: 5
Deleted: 5
Deleted: 5
Deleted: 5
Deleted: 5
Deleted: 6
Deleted: 7
Deleted: 7
Deleted: 8
Deleted: 9
Deleted: 10

1 Introduction

[Provide an introductory chapter, indicating if any parts of it are non-normative.]

1.1 Terminology

The key words *must*, *must not*, *required*, *shall*, *shall not*, *should*, *should not*, *recommended*, *may*, and *optional* in this document are to be interpreted as described in **Error! Reference source not found.**

Deleted: [RFC2119]

2 Context

3 Concepts

3.1 MUWS Architecture Introduction

The MUWS Architecture being addressed in this document consists of the pieces needed for management using Web Services of generic Information Technology resources. This requires that manageability of the manageable resource be presented via Web Services, whether or not the resource is a Web Service itself. The Introduction/Context section (Section 1) placed this work in the larger context of Web Services Architecture and following sections will provide more detail about the components of the MUWS Architecture.

3.2 MUWS Architecture Scope

The MUWS Architecture being defined consists of the Provider of Manageability via Web Services (which consists of the Web Services endpoint(s), service(s), and interface(s) that expose the manageability capabilities for the manageable resource), the Consumer of Manageability, and other required infrastructure.

In addition to providing detailed information on the components that make up the Provider of Manageability, this document will address other items. The following items require specific notes on which parts are in and out of scope for the MUWS Architecture:

- The Consumer of Manageability (each manager which needs to manage some aspect of a manageable resource using MUWS is a consumer of Manageability). The Consumer must be able to make use of the manageability interface(s) provided by or on behalf of manageable resources. Conventional management applications that do not support MUWS will not be addressed at all in the MUWS Architecture. The Consumer of Manageability, like any Web Service consumer, must be able to send messages to, receive responses from, and possibly receive notifications from the manageability interface. There are no requirements imposed on the use of information received.

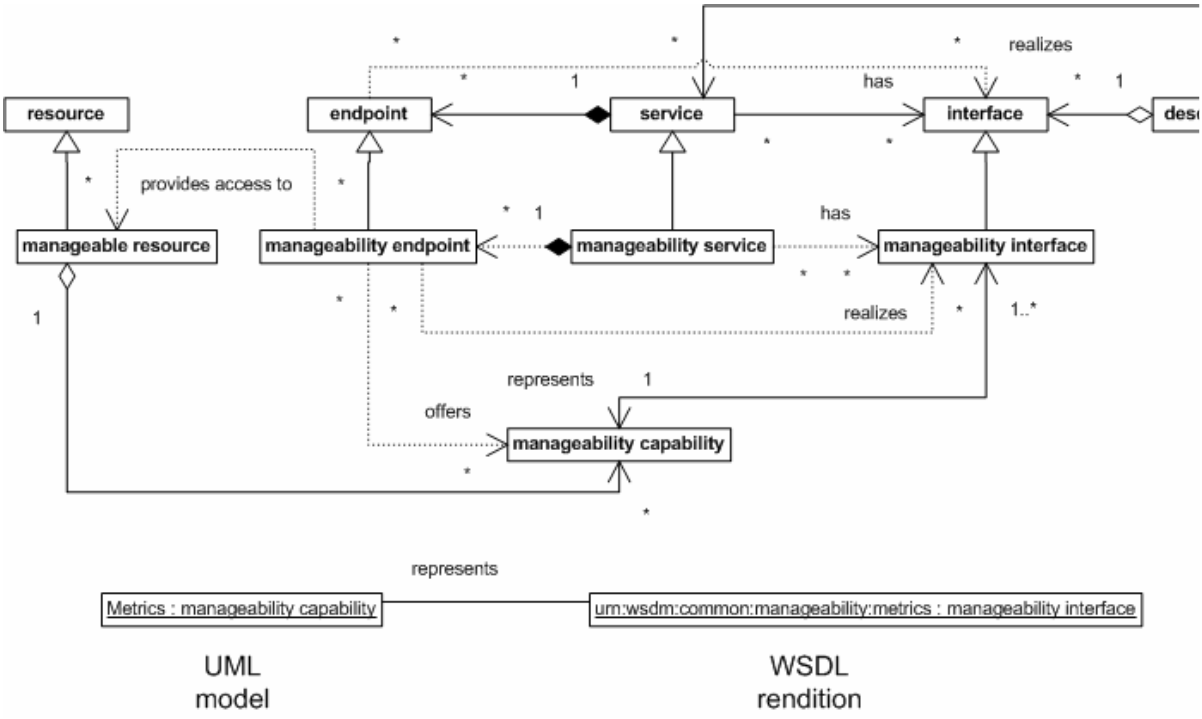
  NOTE: It is important to note that not every Consumer will have the same capabilities. For example, some may be able to process WSDL dynamically, others may not. Some may only be able to do monitoring, others may be able to do monitoring and configuring. This MUWS Architecture will refer to the Consumer in a generic sense, not requiring any particular implementation to provide any particular capability.

- The Manageable Resource. Trying not to change the resource, just specify manageability. No constraints or requirements will be placed on the actual resource itself. In particular, the constraints and requirements will be put on the manageability endpoint and manageability interface to properly provide what manageability capabilities are available for that manageable resource via Web Services. It is entirely possible for there to be manageability capabilities that are not directly supplied by the manageable resource, but are inferred or calculated by another entity and offered by the manageability endpoint.

  Comment: Need a definition that resource is manageable.

- Required infrastructure components. Examples include, but are not limited to, a Registry, a Policy Repository, or a Security service. They will be mentioned in the document where appropriate, and MUWS has requirements on these services, but they will not be defined here. Also, much of this work will be addressed via the MUWS Platform requirements.

3.3 MUWS Architecture Concept Diagram



This Management Using Web Services specification defines how manageability of an arbitrary IT resource can be accessed via Web services. Thus, manageability is one possible quality of a resource. '' Manageability 'is composed of a number of capabilities. Each capability has its own distinct semantics (e.g., could be expressed in a UML model). Therefore, a manageable resource composes a set of manageability capabilities. Figure ?, relates the concepts necessary for management using Web services.

According to the concepts in the WSDL specification, a Web service is an aggregate of endpoints each offering the service at an address and accessible according to a binding. A service has a number of interfaces that are realized by all of its endpoints. Each interface describes a set of named messages that could be exchanged and their format. Properly formatted messages could be sent to an endpoint's address in a way prescribed by the binding. A description (document, artifact) is composed of definitions of interfaces and services. A description may contain both or either of the definitions.

In accordance with the Web Services concepts expressed above, access to the manageability for a resource must be provided by an endpoint. We call such an endpoint a manageability endpoint. Implicitly, a manageability endpoint belongs to a manageability service, which has a

Deleted: A resource is a manageable resource if it has manageability, management policy, and management related semantics [cite WSA]. In this document, we are

Deleted: concerned with management

Deleted: using

Deleted: services

Deleted: which

Deleted: A

Deleted: Web services

number of manageability interfaces that are realized by manageability endpoints. Thus, a single manageability interface represents all or part of a manageability capability. Similarly, a single manageability capability may be represented in one or more interfaces. The semantics of a particular capability is represented in a set of possible message exchanges and rendered in message formats grouped into one or more interfaces.

For example, ability to offer metrics could be captured in a 'Metrics' UML model which is, therefore, an instance of the manageability capability concept. The semantics of offering metrics could be rendered from the UML model into a WSDL interface description defined in a "urn:wsdm:common:manageability:metrics" namespace. That would be an instance of the manageability interface concept.

This specification defines the base set of manageability capabilities that could be composed into a manageable resource or combined into aggregate capabilities. For example, a TotallyManagableResource uber-capability could be defined that includes all of the base manageability capabilities. Such aggregate capability could also be composed into a manageable resource, and in that sense, an aggregate capability is conceptually the same as any other capability. However, this specification does not currently attempt to define (identify) the aggregate capabilities and focuses on the definition of the base set.

Comment: This example needs to match our agreement on the "meta-model" for manageability capabilities. I would suggest that we put a place holder for an example and insert one once we have agreement on the meta model.

Comment: Although it may define some common aggregations.

Deleted: d into uber-

Deleted: uber

Deleted: -

Deleted: uber

Deleted: -

Deleted: □

Deleted: d uber

Deleted: -

4 Logical Architecture

4.1 Information Model

4.2 Roles

This section documents the roles various components of the MUWS Architecture, as well as related components, will have during management using Web Services. It is not intended to constrain the locus of implementation, but instead is intended to document the required components and which actions each is required to take.

NOTE: One application implementation may have many roles or a full role may be implemented by a combination of many different applications.

The major roles are Consumer of the Manageability Service and Provider of the Manageability Service. Related roles are Manageable Resource and related infrastructure components, such as a Directory.

There are also actions only referred to here, because there is no direct relationship to the manageability service, and standardization is not required. Such as getting a new manageability service or component up and running for the first time.

4.2.1 Consumer of Manageability

The Consumer of Manageability plays a role in the management of manageable resources. Because the Manageability Service is a Web Service, the Consumer must follow the Web Services rules. Needs to do the following. Consume information, manage the resource (monitor, configure, etc). Needs to understand the resource. Using information provided by manageability. And to control and configure the resource using the manageability capabilities.

- The Consumer must send properly formatted messages (based on the WSDL describing the service) to the appropriate Provider of the manageability service. .
- The Consumer must be able to locate the appropriate Provider for the manageable resource being managed.
- The Consumer must be able to receive responses from the Provider.
- In order to receive Notifications, the Consumer must also provide a Web Service (making it a specialized Provider of a Notification Receipt Web Service) that supports receiving notifications from the Provider and responds appropriately.
- The Consumer may be capable of discovering manageable resources from a Provider which has a relationship with another Provider or manageable resource or through a Directory.
- The Consumer must follow the security requirements of the Provider and properly authenticate with the Provider as well as using interoperable confidentiality and integrity mechanisms.

Comment: We have a difference of opinion here on the purpose of MUWS. In my mind the specification defines the mapping from any model to an interface. It does not itself define a model. What we need to discuss is the issue of some canonical manageabiliy items. Ones that need to be cannonical in order to provide a uniform platform for manageability (e.g., identity), and the ones that either are canonical because we believe any model has them - which IMO are not where we should be focusing. So I think that this is a point of discussion.

Deleted: Service

Deleted: the

Deleted: Service

4.2.2 Provider of Manageability

The Provider of the Manageability Service plays the largest role in the management of manageable resources via MUWS. The Provider supplies Manageability for a manageable resource. It provides sufficient information for Consumer according to the manageability capabilities of the resource. And may assist with configuration.

Deleted: the

Deleted: Service

NOTE: The Provider may be implemented in the manageable resource or it may not. The Provider may supply the Manageability Service for more than one manageable resource. In other words, this is not intended to constrain the locus of implementation.

- The Provider must describe the Manageability Service provided for a manageable resource in WSDL.
- The Provider must be able to receive properly formatted messages as described in the WSDL.
- The Provider must be able to respond to properly formatted messages appropriately.
- The Provider may be able to generate Notifications and send them to a Consumer as indicated by the Consumer or via the Consumer's WSDL.
- The Provider must follow the security requirements of the environment.

4.2.3 Manageable Resource

The Manageable Resource must perform the business tasks it is normally required to do. Because there are no restrictions on the locus of implementation, the manageable resource may or may not implement the role of Provider of the Manageability Service.

4.2.4 Infrastructure Components

The Web Services Infrastructure Components are identified in this document as providing specific services that the Consumer or Provider requires in order to consume or provide the Manageability Service.

4.3 Processing Model and Interaction Patterns

Deleted: Diagram

4.4 Delegation Architecture

5 Implementation Architecture

6 References

6.1 Normative

.

Appendix A. Acknowledgments

The following individuals were members of the committee during the development of this specification:

Appendix B. Revision History

Rev	Date	By Whom	What
1	30 October 2003	Zulah Eckert	Set up the original template
1	5 November 2003	Zulah Eckert and John DeCarlo	Add material on scope, roles, concept diagram, and other text

Appendix C. Notices

OASIS takes no position regarding the validity or scope of any intellectual property or other rights that might be claimed to pertain to the implementation or use of the technology described in this document or the extent to which any license under such rights might or might not be available; neither does it represent that it has made any effort to identify any such rights. Information on OASIS's procedures with respect to rights in OASIS specifications can be found at the OASIS website. Copies of claims of rights made available for publication and any assurances of licenses to be made available, or the result of an attempt made to obtain a general license or permission for the use of such proprietary rights by implementors or users of this specification, can be obtained from the OASIS Executive Director.

OASIS invites any interested party to bring to its attention any copyrights, patents or patent applications, or other proprietary rights which may cover technology that may be required to implement this specification. Please address the information to the OASIS Executive Director.