WEB SERVICES - RELIABILITY NOW AN OASIS STANDARD

Alan J. Weissberger

NEC Corp voting member of OASIS WS-RM TC

aweissberger@sbcglobal.net

Summary

At its Nov. 10-11 meeting, the OASIS WS-RM (Reliable Messaging) TC voted to send the WS Reliability specification version 1.1 to OASIS for publication as a standard. An OASIS Standard signifies the highest level of ratification for a specification developed by an OASIS TC. Developed through an open process, WS-Reliability enables companies to conduct reliable business-to-business trading or collaboration using Web services. Three protocol capabilities are provided by this standard: guaranteed delivery, ordered delivery, and duplicate elimination. These are described along with message reply patterns later in this article. Figures illustrating the reliable messaging model and the reply patterns are illustrated at the end of this article.

The milestones reached by the WS-RM TC are noted in Table 1.

Additionally, four companies (NEC, Fujitsu, Hitachi and Oracle) participated in a successful interoperability demo of the WS-Reliability specification at the November 19 XML 2004 Conference in Washington, DC. This was the third such validation of multi-vendor interoperability by the WS-RM TC. Other OASIS specs were also demo'd at that conference. Please refer to:

http://www.oasis-open.org/news/oasis_news_11_19_04.php

WS-Reliability will be used in the Japanese Business Grid project to ensure reliable delivery of SOAP formatted notification messages, which are sent based on fixed, predefined conditions, e.g. CPU/Server throughput exceeds a pre-set threshold level or drops below a "low water mark." These reliable notification messages may be sent between different companies at different geographical locations. This may facilitate disaster recovery or remote data base synchronization (or back-up) between multiple Grid sites. The standards that will be used in the Japanese Business Grid project are listed in the Table 2.

As other Web services standards are approved by OASIS, they will likely be encapsulated in WS Reliability envelopes to ensure reliable delivery between end points.

A presentation on the Japanese Business Grid Project was made at GGF12 Enterprise Grid Workshop session, and is available for free download from:

- forge.Gridforum.org/docman2/ViewCategory.php?group_id=136&category_id=836.

WS-Reliability Open Source Code Now Available

In late November, 2004, Fujitsu Limited, Hitachi Ltd, and NEC Corp announced that they are making available their jointly developed open source messaging software that implements the Web Services Reliability (WS-Reliability) standard. The software will be available for download free of charge as of Nov. 26 at the Web site of the Information-Technology Promotion Agency, Japan (http://businessGrid.i pa.go.jp/rm4gs/index-en.html).

Releasing this software as open source will help speed the widespread adoption of software products that incorporate this reliable messaging function, thereby making it possible for customers to develop highly reliable Web services systems in shorter time frames and at lower cost.

Since July of 2003, the three firms have jointly pursued the development of middleware software for the realization of Grid computing as part of the Business Grid Computing Project promoted by the Japanese Ministry of Economy, Trade and Industry (METI). The software was developed as part of this project.

For more information:

The Information Technology Promotion Agency's RM4GS (Reliable Messaging for Grid Services) page: http://businessGrid.i pa.go.jp/rm4gs/index-en.html.

I. Why Is Reliability (i.e. Reliable Message Delivery) Needed For Web Services?

As Web services (WS) start to be deployed across enterprise boundaries and for collaborative e-business and e-transaction scenarios, reliable delivery of messages becomes a critical issue. This is because communications over the Internet and Intranets is inherently unreliable, as the underlying "transport protocols" (HTTP, SMTP and JMS) do not offer any form of guaranteed or ordered delivery for SOAP messages. Yet, those messages must be delivered to the ultimate receiver, even in the presence of component, system or network failures! If a message cannot be reliably delivered, then the user must be so informed.

For Web services messaging to be robust within an enterprise, or to be used across firewalls, it is imperative that a large amount of control, management and security related protocol information be delivered over a reliable connection. It is also important to ensure that user data exchanges are similarly delivered in a reliable fashion to the Application entity. A Reliable Messaging sender and receiver must co-operate to achieve this WS Reliability. The "users" of reliable messaging are either other WS protocols (e.g. WS Security, WS Notification, WS Resource Properties, WS Distributed Management, etc) and/or Application layer/user information exchanges between the end points of the connection.

Accordingly, reliable messaging becomes one of the first problems that need to be addressed for Web services to become a truly viable software technology. (Would you

consider sending credit transactions to your bank or placing a stock purchase or sale order over an unreliable Web service connection?)

II. The OASIS WS Reliability Specification Explained

A. Overview:

WS Reliability is an open specification for ensuring reliable message delivery for Web services. Reliability, in this context, is defined as the ability to guarantee message delivery to "users" with a chosen level of protocol capability and Quality of Service (QOS). Again, the users are either other WS protocols (e.g. WS Security, WS Distributed Management, WS-Notifications, etc), or Application layer/user information messages which are exchanged between the end points of the connection.

To facilitate WS Reliability, there is a need for SOAP based Reliable Messaging Processors (RMPs) -- in the sender and in the receiver endpoints* -- that work together to ensure that messages are delivered in a reliable manner over a connection that may be inherently unreliable.

The sender and receiver RMPs operate on newly defined SOAP headers that are transmitted as either self contained messages, or they are attached to other WS protocol messages or user data messages (all of which are SOAP/XML encoded). Fault messages may extend to the SOAP message body.

*Intermediaries are considered to be transparent in the WS Reliability specification.

The "users" determine the level of WS Reliability. Reliability may include one or more reliable messaging protocol capability for the delivery of WS messages (see II C below for detailed description of these capabilities):

- Guaranteed delivery to the user or Application entity (the message MUST be persisted (i.e. stored in non-volatile memory) in the sender RMP until delivery to the ultimate receiver has been acknowledged. Either a message is delivered, or the sending application is notified of a delivery failure. A resending mechanism controlled by acknowledgements and handled by RMPs, will overcome occasional connection failures or message loss.

- Duplicate elimination -- Delivery at most once -- with duplicates detected and eliminated by the RMP receiver. Duplicate messages could be generated accidentally by some network component (e.g. a router), or intentionally by a resending mechanism. In both cases, it is critical for applications that require only a single instance of the message be delivered, independent of how much time elapsed between the reception of a message and its duplicate.

- Guaranteed message ordering -- when delivered by the RMP receiver to the user, the messages are properly sequenced, in the same order as they were sent. The problem arises when messages are received out of sequence or were resent when acknowledgements are lost. The RMP reorders the

messages before delivery to the application, waiting for delayed messages to arrive. (Solution: RMP transmitter retransmits unacknowledged messages -- after a time-out -- and the RMP receiver re-orders received out of sequence messages so that they are properly delivered to the user/Application entity)

The users of the WS Reliability protocol may agree upon any or all of the above message delivery capabilities. Different users or applications may choose different protocol capabilities, which are conveyed to the RMP sender and receiver prior to initiating communications. Alternatively, the receiver RMP can determine the protocol capability via explicit parameter values sent in each reliable message request.

For purposes of the WS RM TC, QOS is defined as the ability to determine the following aspects:

- Message persistence (ability to store a message until it is reliably delivered to the Application).
- Message acknowledgement (by the receiver and resending (by sender on No Ack time-out).
- Ordered delivery of messages (by use of Sequence numbers).
- Delivery status awareness for both sender and receiver (via state saving and status check- pointing).

The WS Reliability specification defines extensions to SOAP Headers. It is assumed that the payload (user information) is specified using a WSDL description (fault messages may also use the payload to convey fault code information). While WS Reliability is currently based on SOAP 1.1, it could be updated for use with SOAP 1.2, when it becomes a W3C Recommendation.

B. Reliable Messaging (RM) Model and RM Reply Patterns:

In the Reliable Messaging Model described in this specification, the sender node sends a message to the receiver node (i.e., intermediaries are assumed to be transparent in the WS Reliability specification). Upon receipt of the message and at the appropriate time, the receiver node sends back an Acknowledgment message or Fault message to the sender node.

There are three ways for the receiver to send back an Acknowledgment message or a Fault message to the sender. These are referred to as the "RM Reply patterns," which are defined as follows:

- Response RM-Reply Pattern

  We say that a Response RM-Reply pattern is in use if the outbound Reliable Message is sent in the underlying protocol request, and the resultant Acknowledgment message (or Fault message) is contained in the underlying protocol response message which corresponds to the original

request. In essence, the Acknowledgement is "piggybacked" onto the business response message.

- Callback RM-Reply Pattern

  We say that a Callback RM-Reply pattern is in use if the Acknowledgment message (or Fault message) is contained in an underlying protocol request of a second request/response exchange (or a second one-way message), operating in the opposite direction to the message containing the outbound Reliable Message.

- Polling RM-Reply Pattern

  We say that the Polling RM-Reply pattern is being used if a second underlying protocol request is generated, in the same direction as the one containing the outbound Reliable Message, to act as a "request for acknowledgment." The Acknowledgment message (or Fault message) is contained in the underlying protocol response to this request. This polling pattern can be used in instances where it is inappropriate for the sender of reliable messages to receive underlying protocol requests e.g. the sender behind a firewall.

These three reply patterns provide "the users" with flexibility to send reliable request/ response or one-way SOAP messages (Callback and Polling patterns). Callback is important for one-way request message patterns and for batching of acknowledgements and fault messages.

Additionally, "polling" enables reliable message delivery to extend beyond the firewall, which might otherwise block external reliable messages from reaching the intended recipient. Polling makes it possible to use the WS Reliability protocol, even when a firewall prevents 3rd parties from initiating messages or requests.

The illustrations of the basic messaging model and the reply patterns are available below.

C. WS Reliability Protocol Capabilities:

Three types of message delivery capabilities are defined in the WS Reliability protocol. One or more of these protocol capabilities may be used with each of the RM Reply patterns defined in II B above. The selection is dependent on prior end user agreements or explicitly inferred by the receiver RMP from request messages.

- Guaranteed Delivery

To successfully deliver a message from a sender RMP to a receiver RMP without failure; if this is not possible, to report the failure to the sender's application. To realize guaranteed delivery, the message MUST be persisted (i.e. stored) in the sender RMP until delivery to the receiver is acknowledged, or until the ultimate failure is reported to it's requester. (There is a requirement on the underlying transport protocol that the message MUST be transported without corruption.) If message persistence is lost for any reason, it is no longer possible to guarantee message delivery. Since the reliability of message

persistence is a property of the system implementation, the conditions under which guaranteed message delivery holds is also a property of the system implementation and is outside the scope of the specification.

Example 1. A PC Server may use a HDD for it's persistent Storage, and those messages persisted in the HDD are reliably maintained even if the the system software crashes and the system is rebooted. However, if the HDD itself crashes, it is no longer possible to guarantee message delivery.

Example 2. A message persisted in a mobile phone may be lost when it's battery is detached. In this case, message delivery is only guaranteed by proper battery maintenance of the mobile phone.

- Duplicate Elimination

A number of conditions may result in transmission of duplicate message(s), e.g. temporary downtime of the sender or receiver, a routing problem between the sender and receiver, etc. In order to provide at-most-once semantics, the ultimate RMP receiver MUST eliminate duplicate messages and never present them to the user. Messages with the same Message Identifier value MUST be treated as duplicates and not delivered to the application.

- Guaranteed Message Ordering

Some applications will expect to receive a sequence of messages from the same sender in the same order those messages were sent. Although there are often means to enforce this at the Application layer, this is not always possible or practical. In such cases, the Reliable Messaging layer is required to guarantee the message order. This specification defines a model, illustrated in Figure 3, to meet this requirement.

When the sender application sends three messages (1), (2), and (3) with Guaranteed Message Ordering, the receiver's RMP MUST guarantee that message order when it makes those messages available to the receiver's application (the user). In Figure 3, the receiver's RMP received messages (1) and (3), the receiver's RMP makes message (1) available to the application, but it persists message (3) until message (2) is received. When receiver's RMP receives message (2), it then makes message (2) and (3) available to the application, in that order.

Table 1. Milestones Reached By OASIS WS RM TC

- Dec. 9, 2003 -- Public Interop Demo at XML/2003 conference: Fujitsu, Hitachi, Oracle, NEC and Sun implemented WS-Reliability CD* 0.52.
- March 17, 2004 -- OASIS Public Review of CD 0.992 initiated.
- Aug. 24, 2004 -- TC votes to recommend CD 1.086 for OASIS Member Review.
- Oct. 16, 2004 -- OASIS Member Vote on WS-Reliability Version 1.1 -- initiated.

- Oct. 30 2004 -- OASIS Member Vote completed.
- Nov. 10, 2004 -- WS Reliability becomes an OASIS standard.

*CD= Committee Draft

Table 2. Relevant Standardization Bodies For Japanese Business Grid Project

- GGF
    - OGSA-WG (architecture, roadmap, WG factory).
    - CMM-WG (resource management).
    - JSDL-WG (job portability).
    - CDDLM-WG (configuration, deployment, lifecycle management).
- OASIS
    - WSDM TC.
    - WSRM TC (WS Reliability).
    - WSBPEL TC.
    - WSRF TC, WSN TC.
- DMTF
    - Server Management WG.
    - Utility Computing WG.

Figure 1 Messaging Model

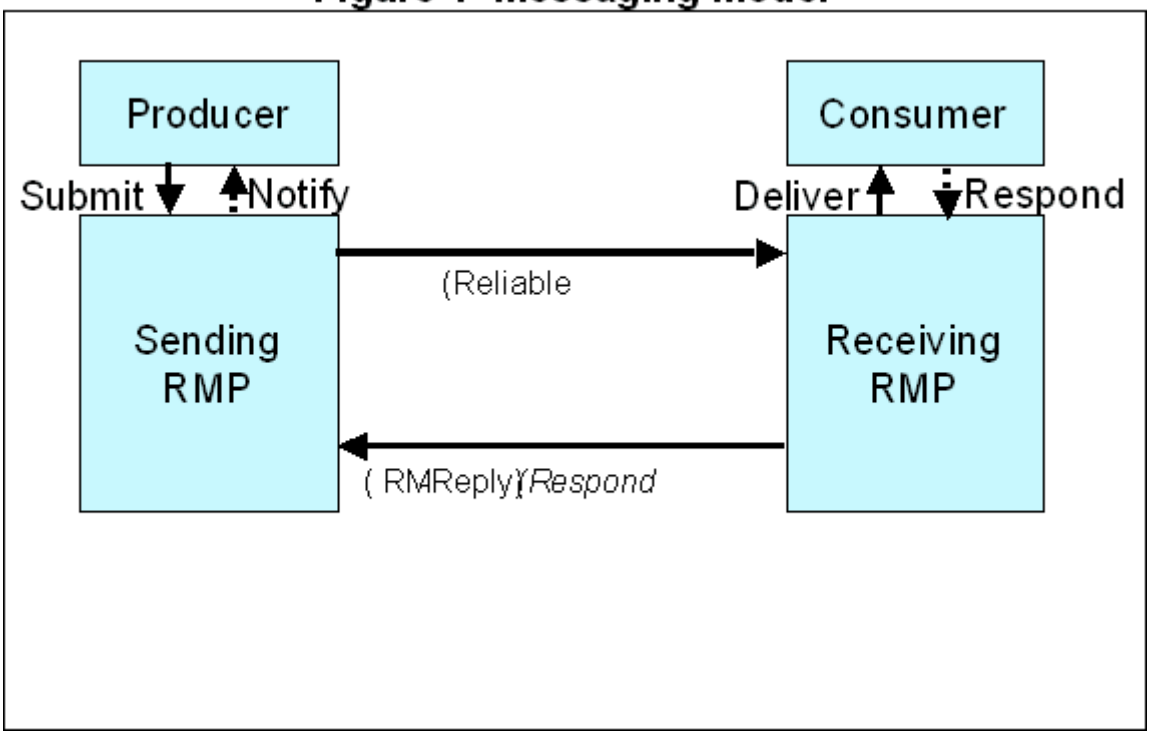

Figure 2 Response RM-Reply Pattern

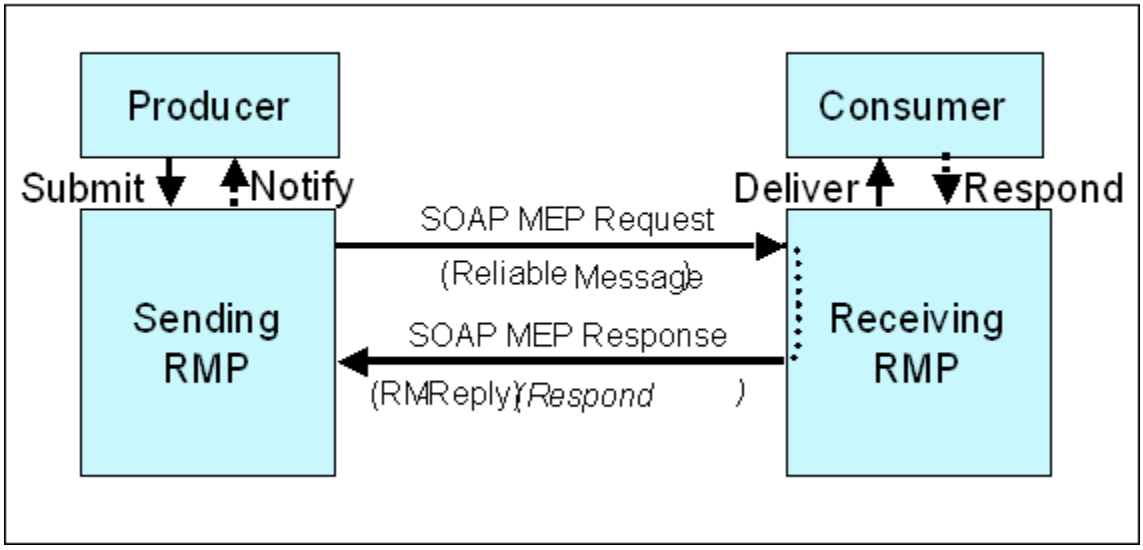

Figure 3 Callback RM-Reply Pattern

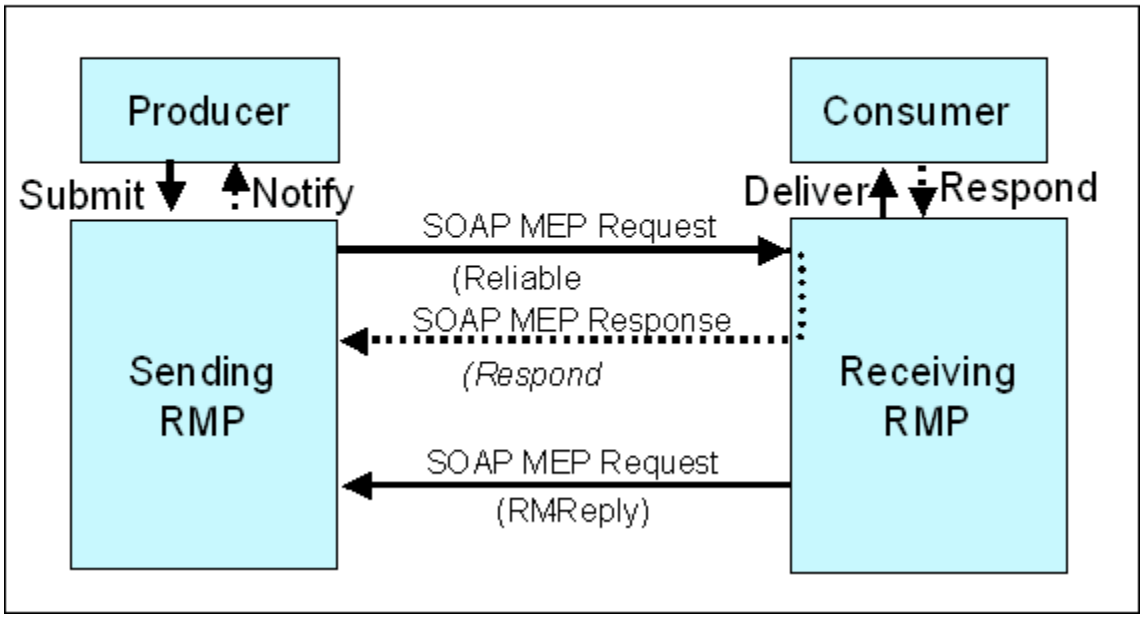

Figure 4 Poll RMReply Pattern

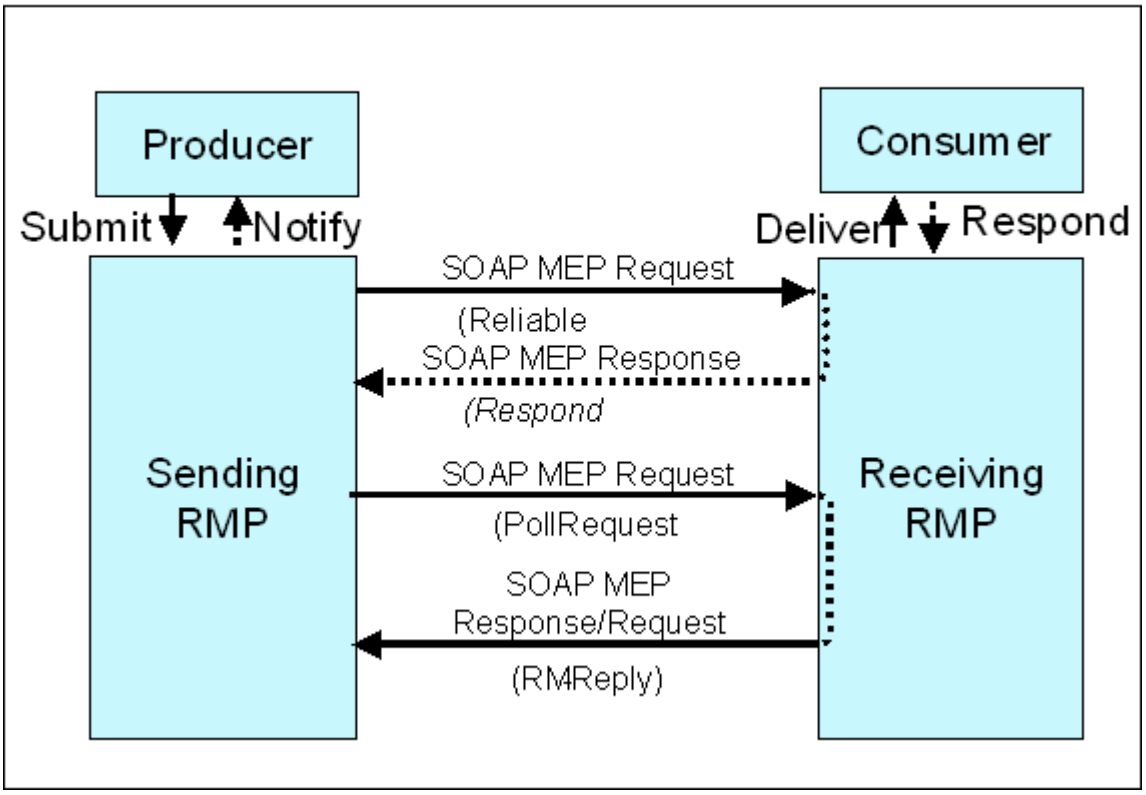

Figure 4(a) Poll RM-Reply Pattern (Synchronous)

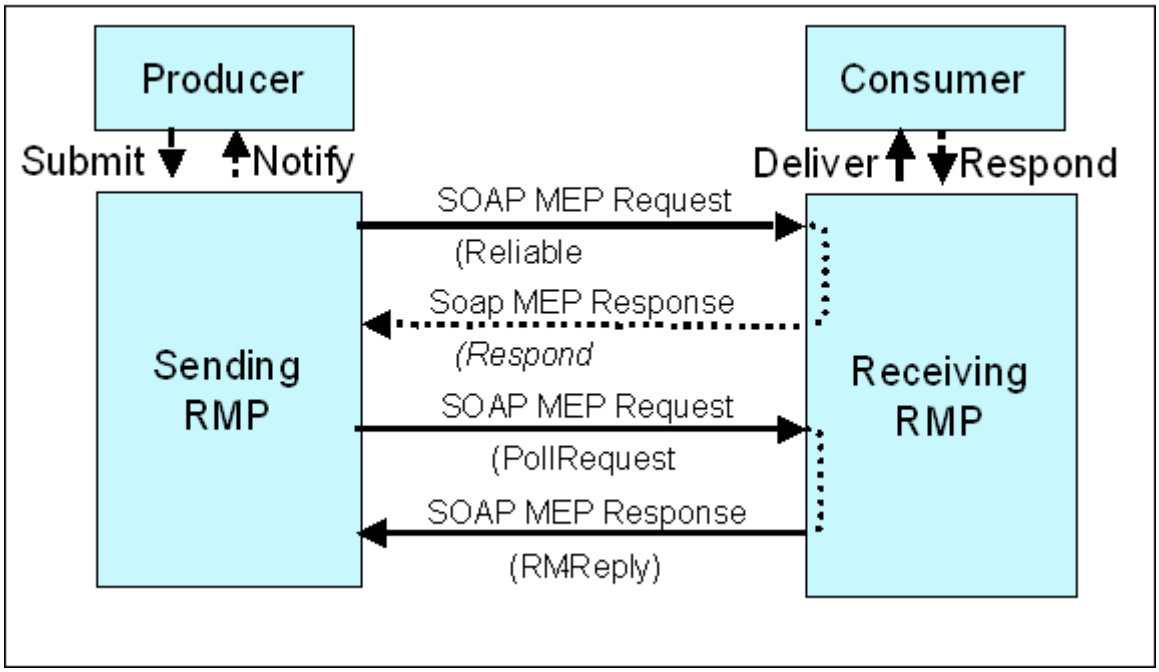

Figure 4(b) Poll RM-Reply Pattern (Asynchronous)