10 Advanced Web Services for the Enterprises

BACKGROUND

While the first generation web services components, viz. SOAP, WSDL, and UDDI, helped creating loosely coupled web services, the process oriented web services technologies, such as WS-BPEL and WS-CDL, have helped to establish web services as a part of the business process among the collaborating partner organizations. This technology leap for the enterprises is essential, but not sufficient, to cater to the enterprise's ambitious requirement of meeting the SOA guidelines and principles. Advanced non-functional characteristics of web services technologies such as security, reliability, other qualities of service, collaboration, etc., are essential for the enterprise's needs. To that end, web services technologies such as WS-Security, WS-Reliability, WS-Collaboration, etc. collectively referred to as WS-* (pronounced WS-STAR) are some important advanced web services vocabularies that can help provide support in meeting the larger needs of enterprise requirements.

10.1 FIRST GENERATION WEB SERVICES

Web services based on SOAP, WSDL, and UDDI technologies, now dubbed as first generation web services, provided a solid foundation for the enterprises to initiate thinking in terms of "services" and "interoperability". As a part of this foundation, the SOAP protocol provided a firm basis for message exchange as part of the services, WSDL documents helped in describing the services in an interoperable way, while the UDDI specifications and services provided a brokering arena for the provider and requester for enabling business automation. Orchestration and

choreography[1] oriented web services technologies, namely WS-BPEL and WS-CDL, started evolving to provide a means to organize the business process to transpire in a smooth and elegant manner.

As we indicated earlier, as per the definition of web services by Gartner, not all three constituents of first generation web services—SOAP, WSDL, and UDDI—are required for building web services oriented applications. Similarly, process oriented web services technologies such as WS-BPEL and WS-CDL are not mandatory to build the business process. Requirement of WS-BPEL and/or WS-CDL depends on the ultimate goal of the business process needs of the enterprise. The key question that is being addressed by the enterprises now is—are enterprises attempting to bring about the services orientation within the firewalls of the intranet? or are they attempting to bring about collaborative business integration with business partners and collaborators over the Internet?

Although the core web services are essential, they are only building blocks in creating services. However, enterprises are looking to address the bigger problem, for which services oriented architecture becomes more important, and the core web services along with the process oriented web services technologies address the problem in a holistic manner. Together, the core web services technologies and process oriented web services technologies become the mainstream solution for implementing the fundamental and functional aspects of enterprise requirements.

For large enterprises, implementation of robust and elegant service-oriented architecture requires much more than the basic building blocks. These enterprises demand robust and elegant infrastructure frameworks, tools, technologies, and specifications to design and deploy loosely-coupled, service oriented and interoperable solutions. The first generation web services addressed functional requirements and the second generation web services are evolving to meet the non-functional requirements of the enterprises.

The second generation web services are essentially a set of advanced web services specifications driven by many groups or consortiums of industries and vendors. While some of the specifications have already been consolidated and have been endorsed by leading industry consortia, even more are emerging to meet the ever-growing demands of enterprise's dynamically changing requirements. Together, these specifications are referred to as WS-*. There are several second generation web services specifications; some of the more crucial ones address the following important non-functional requirements:

- Addressing
- Collaboration
- Reliability/Reliable Messaging
- Security

It should be borne in mind that not all the advanced web services technologies are necessary for all enterprises. Depending on its requirement, an enterprise might choose to implement a subset

[1]Many authors and books consider WS-BPEL and WS-CDL as the second generation web services. In this book, we are not in disagreement with that. However, we have attempted to group the orchestration and choreography elements as the essential building blocks.

of the WS-* technologies, along with some of the core web services technologies. The enterprises may also rope in process oriented web services technologies for meeting the orchestration and/or choreography requirements of the enterprise. These advanced specifications, generally termed the second generation web services technologies, are also referred to as contemporary web services extensions.

10.2 WS-*—OVERVIEW OF THE ADVANCED WEB SERVICES

The non-functional requirements such as security, reliability, workflow / business process, and so on were not addressed by the core specifications. However, enterprises find these requirements overwhelmingly important to be a part of their web services solution. To fill in this need, the second generation web services technologies have emerged. Any enterprise will have a list of non-functional requirements to be met as a part of the enterprise application, and to meet these requirements a number of advanced web services technologies have emerged to meet such demands. The following is partial list of some of the important infrastructure specifications and frameworks:

- WS-Reliability or WS-Reliable Messaging
- WS-Security
- WS-Policy Framework
- WS-Addressing
- WS-Eventing
- WS-Notification
- WS-Metadata Exchange
- WS-Atomic Transaction
- WS-Coordination

10.2.1 WS-Reliability/WS-Reliable Messaging

WS-Reliability or *WS-Reliable Messaging* is a SOAP[2]-based specification that enables fulfilling the reliable messaging requirements critical to a part of the whole web service. These specifications attempt to define reliability in the context of present day web services implementations.

10.2.2 WS-Security

WS-Security is a broad-ranging framework composed of several other specifications and frameworks. Many consortia have contributed to the development of these supplementary and complementary

[2]WS-Reliability and WS-Reliable Messaging are competing Web Services standards developed and supported by different industry consortia. Both these specifications have a common agenda, and the specifications overlap in many aspects. There are differences as well. In this book, we bring in appropriate aspects of either of the specifications when highlighting the reliability requirement of the services delivery.

frameworks and specifications. In the next section, we provide some working definitions of a few of the important second generation web Services specifications. We also present their scope and application scenarios and utility value proposition for the enterprises, highlighting their importance in terms of the principles of SOA. The definitions are presented in alphabetical order; note that their order of preference for any enterprise is purely requirement-driven.

10.2.3 WS-Policy Framework

The *WS-Policy Framework* is composed of three different policy-related specifications. They are WS-Policy, WS-Policy Attachments, and WS-Policy Assertions. WS-Policy is a specification that enables web services to advertise their policies (on non-functional requirements such as security, etc.) and enables web service consumers to specify their policy requirements.

10.2.4 WS-Addressing

WS-Addressing is a specification authored originally by IBM, Microsoft, BEA, Sun, and SAP. The specification was submitted to W3C for standardization. This specification enables web services to communicate addressing information. It essentially consists of two parts—a structure for communicating a reference to a Web service endpoint, and a set of Message Addressing Properties. These properties associate addressing information with a specific message.

10.2.5 WS-Eventing

The *WS-Eventing* specification, initiated by Microsoft, focuses on addressing the requirements of the *Publish/Subscribe* (or *Pub/Sub*) web services messaging infrastructure of the organization. WS-Eventing specifications help in notifications to the subscribers of specific web services when a particular web services is initiated/completed.

10.2.6 WS-Notification

The *WS-Notification* specification, initiated by IBM, addresses the requirements of the Publish/Subscribe web services messaging infrastructure of the organization.

10.2.7 WS-Metadata Exchange

The term *Metadata* in the WS-Metadata Exchange specification refers to the data about the collaborating web services. The participants in the web services exchange (particularly the requester) might need additional data related to the other web services. This specification enables the requester to send a standardized message requesting some or all information related to the web service being consumed.

10.2.8 WS-Atomic Transaction

WS-Atomic Transaction is a specification that defines protocols for distributed transaction processes to be used as part of WS-Coordination. Atomic transactions are essentially the ACID transactions—atomicity, consistency, isolation, and durability, and the specifications address and confirm that protocols meet the requirements of the enterprises.

10.2.9 WS-Coordination

The *WS-Coordination* specification describes an extensible framework that facilitates the protocols used in the enterprises to coordinate the *actions* of applications in a distributed environment. This framework includes other related specifications such as WS-BPEL, WS-Business Activity, and WS-Atomic Transaction. Developed jointly by IBM, BEA, and Microsoft, this specification helps multiple distributed applications to function correctly so that the results of the distributed transactions behave as expected.

10.3 WS-*—A Detailed Treatment

We provide here descriptive working definitions of the most frequently used second generation web services technologies by enterprises: reliability, security and addressing.

10.3.1 Reliability and Reliable Messaging

One of the concerns of asynchronous-oriented communication is the reliability of the message delivery. There could be several issues including

- Delivery of the message to the target system
- Detecting whether there is a failure of delivery of the message
- Sequencing and prioritizing a series of messages that must be delivered

WS-Reliable Messaging specifications are designed to help in effectively managing various message delivery requirements of enterprises. These specifications also help in tackling issues related to message sequencing, delivering, notification, and so on. If a message cannot be delivered, a suitable fault message is generated so that redelivery of the lost message is ensured. Reliability in message delivery can be ensured by building suitable rules by the WS-Reliable Messaging specifications. These reliability rules are implemented as SOAP header entries. This concept is illustrated in Fig. 10.1. Reliable Messaging specification identifies four participants involved in the delivery of messages: *Message Source*, *Reliable Message Source* (or *RMSource*), *Reliable Message Destination* (or *RM Destination*), and *Message Destination*. Any message delivery between the Message Source and Message Destination cannot transpire without the involvement of the RM Source and RM Destination. The Message Source always transmits the message (or a sequence of messages) to the RM Source, and the rest of the message transmission task is taken over by the latter. When the

transmission of the message (or sequence of messages) is completed, the RM Destination takes care of delivering the same to the Message Destination.

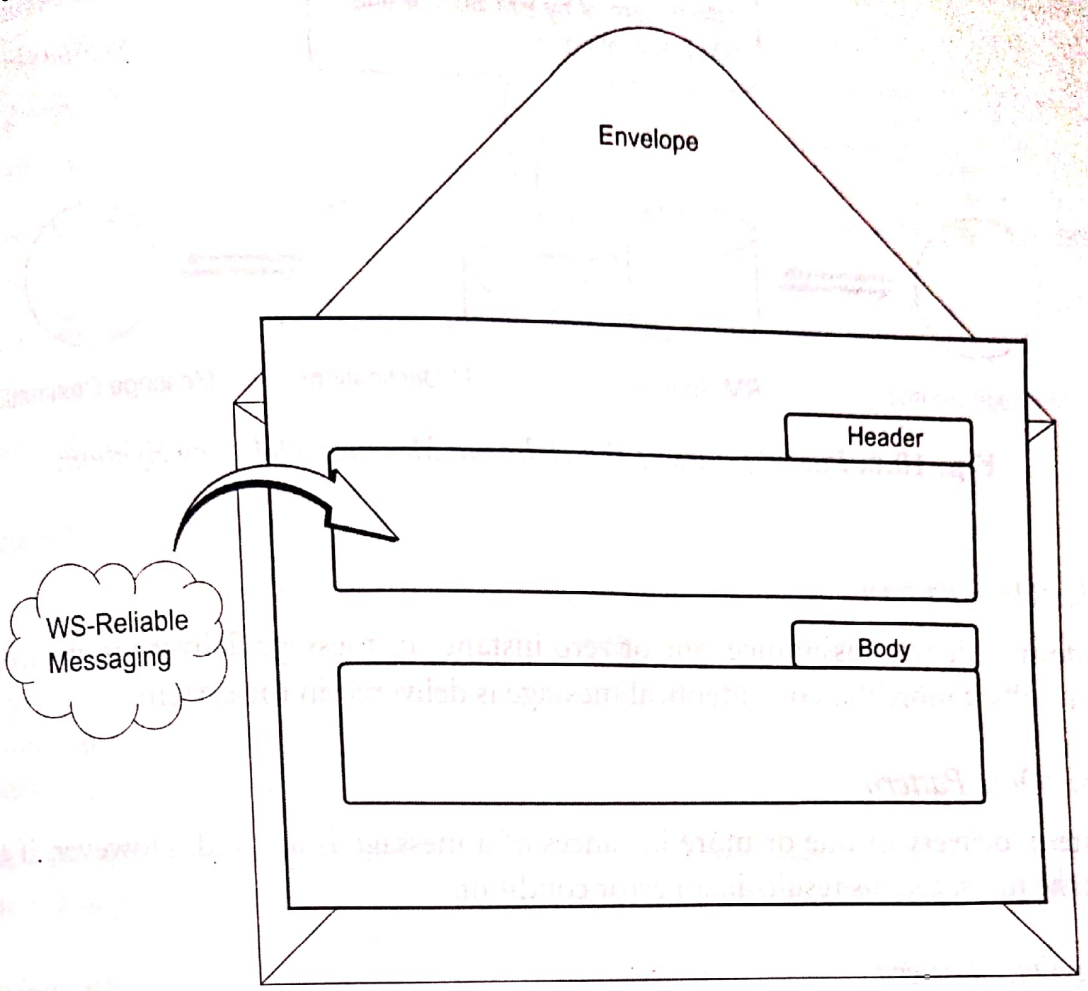

Fig. 10.1: *WS-Reliable Messaging and its Relationship with SOAP Message Structure.*

Acknowledgment of the message delivery is one of the important features of reliable messaging. WS-Reliability enables a variety of acknowledgment scenarios. *Acknowledgment* of the receipt of the messages, *Sequence Acknowledgment* of the delivery of a set of messages, *Request Acknowledgment* by the RM Source and *Negative Acknowledgment* by RM Destination enable a robust quality of service for delivering the messages reliably. The entire process—from the initiation of the message to the delivery of the message in a reliable scenario—is illustrated in Fig.10.2.

The acknowledgment of the receipt of a message enables the RM Source to know that the message has been successfully delivered. Similarly, the Sequence Acknowledgment of the receipt indicates that the ordering of the delivered message was appropriately done. Similarly, RM Destination can be allowed to send a Negative Acknowledgment indicating that the delivery of the message to RM Source was unsuccessful. WS-Reliable Messaging also allows defining four different patterns of delivery sequences. These delivery sequences are referred to as Delivery Assurances. They are *AtMostOnce*, *AtLeastOnce*, *ExactlyOnce*, and *InOrder*.

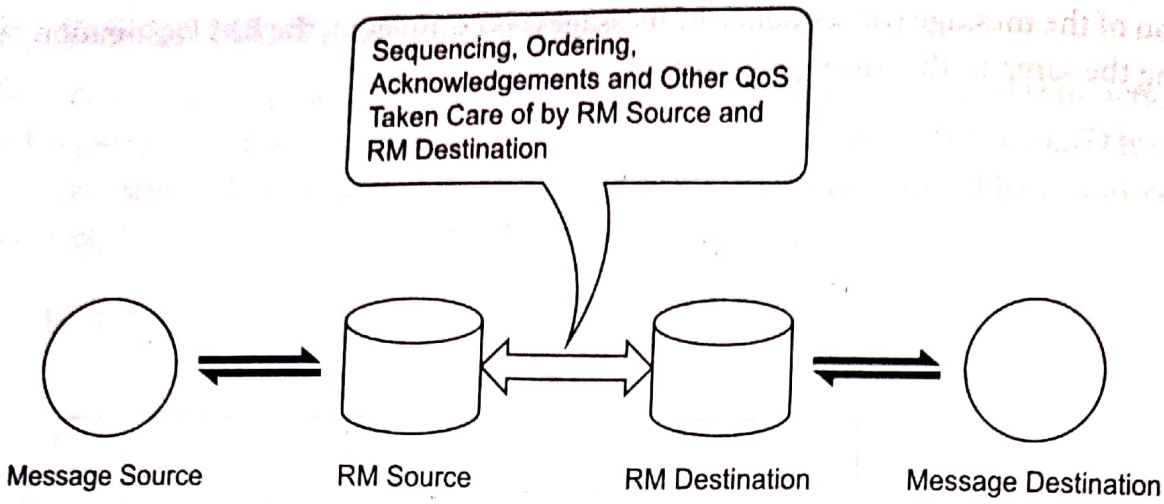

Fig. 10.2: *Participants of the Reliable Message Delivery System.*

The AtMostOnce Pattern

In this pattern of delivery assurance, one or zero instance of message delivery is ensured. An error condition results if more than one identical message is delivered in the system.

The AtLeastOnce Pattern

In this pattern, delivery of one or more instances of a message is assured. However, if there is no delivery of the message, this results in an error condition.

The ExactlyOnce Pattern

This pattern ensures delivery of just one instance of the message. An error condition results in the system if zero or more than one identical message is delivered.

The InOrder Pattern

This pattern of delivery assurance is useful to ensure the order of delivery of a set of messages. If the set of messages is not delivered as per the predefined sequence, an error condition results. This pattern can also be clubbed with that of the other delivery assurance patterns previously discussed.

10.3.2 SECURITY

Security is, perhaps, one of the most important and most widely required non-functional requirements for any enterprise. There are wide varieties of technologies, rules, and practices used to provide security cover to enterprise applications. In the world of web services and SOA, security might be provided through a set of frameworks and specifications called WS-Security. There are many security-related frameworks, supplementary frameworks, and specifications that constitute WS-Security specifications. Some of the important ones are

- WS-Security Policy
- WS-Federation
- XML-Signature
- XML-Encryption
- Secure Socket Layer

Provisioning of security to an enterprise application can be conceptually divided into three different levels:

- The User/Requester-Level Security
- The Transport-Level Security
- The Message-Level Security

10.3.2.1 *Requester-Level Security*

For ensuring the security in an enterprise application, it is important that some checks are implemented as a security blanket. They are *Identification*, *Authentication*, and *Authorization*. An enterprise's goal is to ensure the all of the related security aspects identified above are properly established through a process such as single sign-on. A detailed treatment of these aspects is presented below:

Identification: *Identification* is essentially the property of identity of a person. The employee ID of an employee or *Social Security number* (*SSN*) of a citizen of a country are examples of identification.

Authentication: *Authentication* is the process of verification of a requester trying to use a service that is secured. Authentication confirms the identification of the requester trying to access the service or information. The process of authentication could involve passwords, *Personal Identification Number* (*PIN*), *Trading Partner Identification Number* (*TPIN*), and so on.

Authorization: *Authorization* is the process of permitting the requester to access a service (or set of services) in an appropriate way. Authorization to use a service (or a part of it) could be based on an *Access Control List* (*ACL*).

Single Sign-On

An enterprise solution that has been architected using the principles of SOA and implemented using web services could encounter some unique challenges. Different services and service-oriented applications might require security checks such as Authentication and Authorization. Such security checks, although desirable, can prove expensive and might turn nasty when human users are repeatedly prompted to authenticate (and be authorized) while related and interconnected services are being sought. Such situations can be avoided if a blanket security clearance, such as *Single Sign-On* (SSO), is implemented. The technology of SSO enables the service requester to identify and

authenticate for the first time during a particular session. After this is done, the security context generated during the SSO is shared among different services. There are several technologies of SSOs available that are implemented and used by large enterprises. Some of the popular ones are Kerberos and Federation.

Kerberos: Kerberos is a popular SSO technology that is available on a variety of platforms such as Microsoft Windows, different flavors of UNIX systems, and the mainframe systems. In this technology, the process of authentication is entirely externalized. When the requester signs into the Kerberos server, a ticket is issued to the requester. This ticket is produced by the client applications for using services in different servers.

Federation: Federation technology use standards-based protocols to enable an application to assert the identification of a service requester to another application. This process eliminates the need for redundant authentication. *Security Assertion Markup Language* (*SAML*) and WS-Federation employ the Federation technology. *Liberty Alliance* or *Project Liberty* is a broad-based industry consortium involved in defining federated identification management for web services communication protocols. Many industry leaders such as IBM, Sun Microsystems, Novell, Intel, Oracle, VeriSign, RSA Security, Hewlett-Packard, etc. are involved in the development efforts of SSO based on the concepts of federated identity management. This federated SSO technology is based on SAML and is suitable for intranet and Internet business/application environments.

10.3.2.2 Transport-Level Security

Security at the transport level is essentially providing protection to the message while it is transported from one system to the other. In the context of enterprises using web services technology, the message needs to be protected along the SOAP message path, and this level of security can be achieved by using *Secure Socket Layers (SSL)*. It is important to note that synchronous operations between the requester and provider can be effectively implemented by providing message level security as there is no involvement of intermediaries. However, when asynchronous communications are involved and intermediaries are required, it is important that the intermediaries do not view or modify the message in any way. Such scenarios require implementation of transport-level security.

10.3.2.3 Message-Level Security

Confidentiality is important for any enterprise. The confidentiality of the message is said to be maintained when the content of the message is not viewed unauthorized. Likewise, the integrity of the message is said to be maintained when the message is not altered any way by the intermediaries or any unauthorized persons. Enterprises that are involved in frequently exchanging confidential information need to implement message-level security. Message-level security includes not only the transport-level security, but also maintains the confidentiality and integrity of the message. Technologies such as encryption and digital signature help protect the message. Under the WS-Security frameworks, there are many ways of providing the confidentiality and integrity; however,

for our purposes, we focus on two WS-Security extensions: *XML Encryption* and *XML Signature*. These extensions can provide secure and robust message level security.

XML Encryption: *XML Encryption* is an encryption technology specifically designed for encrypting XML data. The advantage of using XML Encryption is that the encryption can be applied to either the entire message or to specific parts of a specific message. For example, when credit card information is sent, it is possible that just the card number and the expiration date can be encrypted and sent rather than encrypting the complete details of the credit card information.

XML Signature: This technology helps provide authenticity to the accompanying message. The XML document containing the data is accompanied by a code generated by a special algorithm, based on the content of the message. This code is called the *message digest*. To make it a *digital signature*, it has to be encrypted using the private key of the sender. The XML document with the digital signature will together be delivered to the recipient of the message. The recipient needs to verify the digital signature, and the verification of the signature will succeed if and only if the contents of the message accompanying the digital signature have not been altered in any way. XML Encryption and XML Signature are included as part of the SOAP message. Although the digital signature can be sent as a part of the `<Header>` elements, encrypted messages are encoded into the `<Body>` elements, as represented in Fig. 10.3.

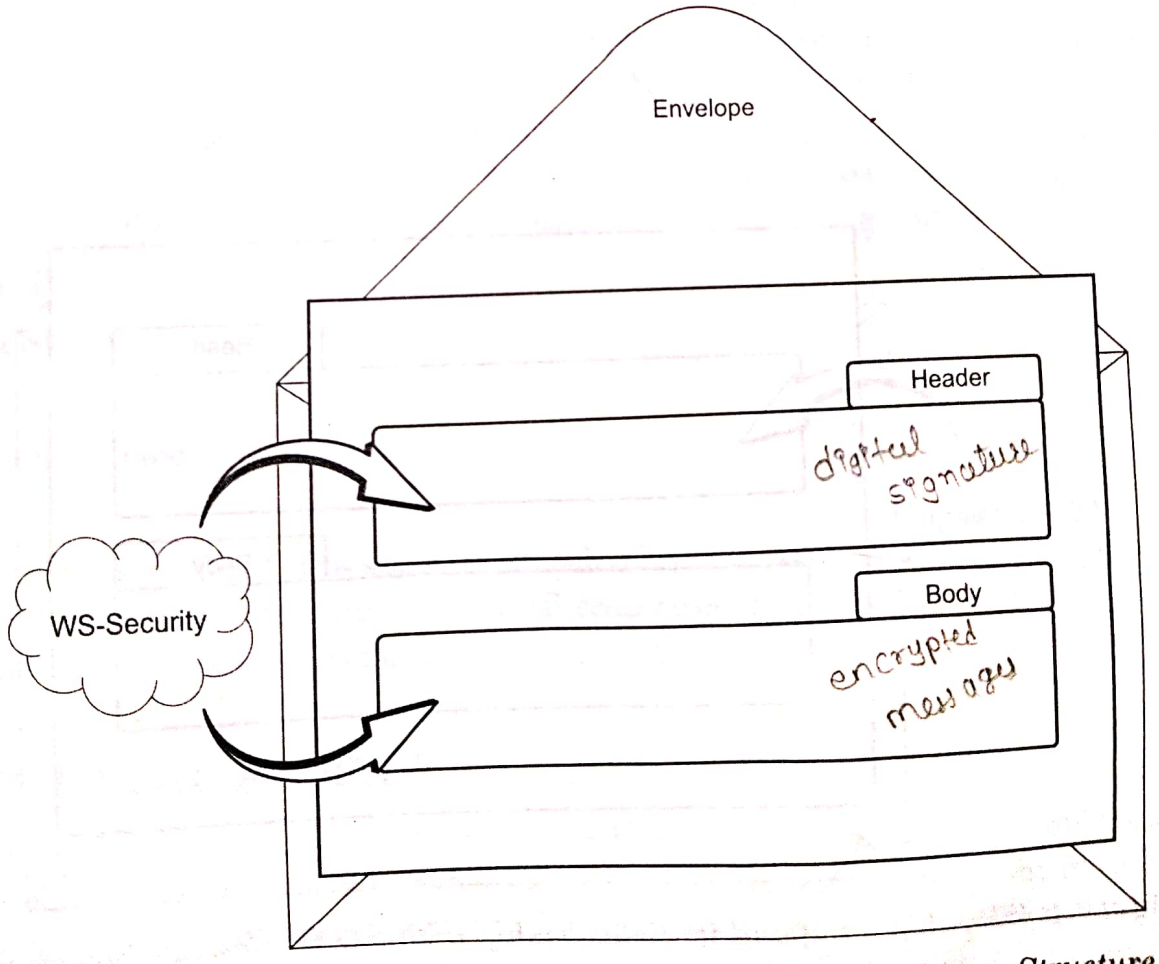

Fig. 10.3: *WS-Security and its Relationship with SOAP Message Structure.*

10.3.3 ADDRESSING

Communication between two applications needs to take place in a trusted, reliable, and secure environment. When such communications occur in the extranet or Internet environment, and these communications are associated with transactional context, it is important that each of the applications know about the location of the application, whereabouts, and other necessary details. In essence, the addressing part should provide location transparency and transport neutral addressing capabilities to the application in a secure manner. The WS-Addressing specification attempts to impart the whereabouts or address features to the communicating applications. The address features include the following:

- Source of the message
- Destination of the message
- Routing details of the message
- Instructions for what needs to be done in case of faults and non-delivery, and so on.

WS-Addressing specifications implement these features within the scope of the SOAP header feature of the SOAP specification. This is presented in Fig. 10.4. The WS-Addressing specification defines two types of SOAP headers – *Message Addressing Properties* and *Endpoint References* (EPR).

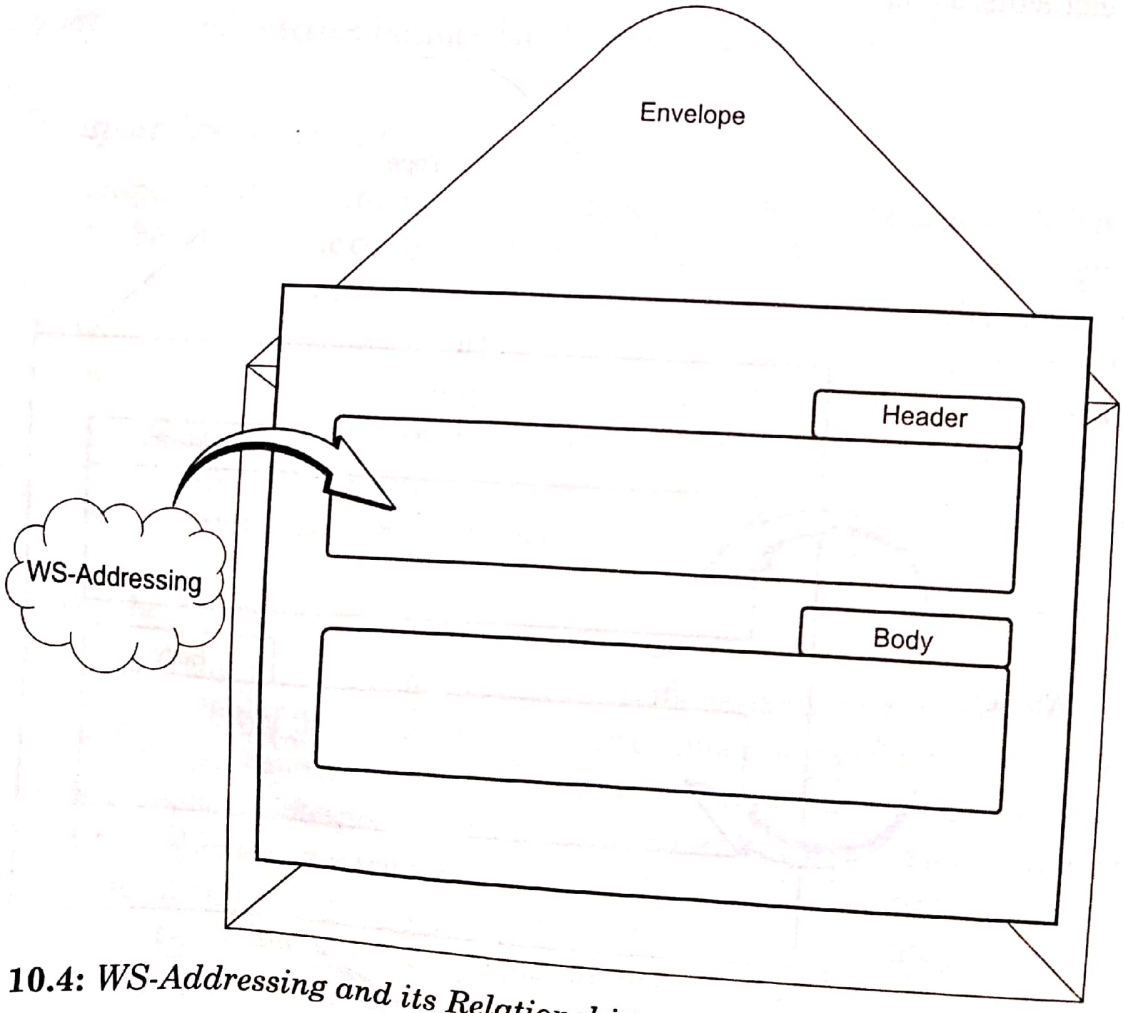

Fig. 10.4: *WS-Addressing and its Relationship with SOAP Message Structure.*

10.3.3.1 *Message Addressing Properties*

In this type of SOAP header, addressing information can be included using the following standardized header entries: *destination*, *source endpoint*, *reply endpoint*, *fault endpoint*, *message ID*, *relationship*, and *action*. The message ID is a very special and important header entry that is specifically used in WS-Reliable Messaging specification.

10.3.3.2 *Endpoint References*

An Endpoint reference type of SOAP header encapsulates the information useful for addressing a message to a specific web service, using the following header entries: *address*, *reference properties*, *reference parameters*, *service port*, *port type*, and *policy*.

10.4 IMPORTANCE OF WEB SERVICES AND WS-* IN SOA

The first generation web services—SOAP, WSDL, and UDDI provided the foundation for interoperability and message interchange using XML. SOAP specification played a key role in message interchange through the synchronous and asynchronous route. As the enterprises and organizations realized the power of web services and realized the possibility of achieving business automation, the requirements of the non-functional requirements for business integration through web services loomed large. Several industrial consortia started building and contributing towards achieving the same. This resulted in the second generation of web services. They are also termed *web services extensions* because many of them are inherently the extensions in SOAP and other advanced XML vocabularies. Particularly, these extensions are needed by industry consortia because they help organizations to architect (or re-architect) the enterprise solutions to extract the maximum benefit for service orientation. Service orientation helps achieve the following goals:

- Reliability
- Interoperability
- Extensibility
- Loose coupling
- Security

You've now read, in detail, how specifically each of the WS-* specifications is designed, and how they help in achieving different non-functional requirements. In the next section, we cover how some of these new generation specifications help in promoting SOA.

10.5 WS-I BASIC PROFILE

The creation of *Web Services Interoperability (WS-I)* was one of the significant milestones in the history of web services. The birth of WS-I signaled the beginning of one of the most important aspect of web services—the agreement on different aspects of interoperability among several different consortia. One of the first significant steps initiated by WS-I was the introduction of a

set of specifications called the WS-I Basic Profile. This is an important step for assembling a set of mature core web services specifications to define a commonly supported web services platform.

The WS-I Basic Profile is standardized on the following core specifications:

- XML 1.0
- XML Schema 1.0
- SOAP 1.1
- WSDL 1.1
- UDDI 2.0

This Basic Profile specification also provides a set of design standards and recommends how a specific feature needs to be implemented. It also covers information regarding dos and don'ts in web services implementation.

Summary

While the first generation web services technologies addressed the functional requirements, the second generation web services technologies gradually evolved addressing the non-functional requirements. As a result, a number of advanced web services extensions have emerged, and the specifications and frameworks towards this end have been contributed by competing industry consortia. For instance, WS-Reliability and WS-Reliable Messaging specifications have a common agenda, and the specifications have duplicating and overlapping proposals. Nevertheless, some of the important extensions that are commonly accepted by most of the industry consortia are WS-Security, WS-Reliability, WS-Addressing and so on. Each of these specifications helps include specific non-functional requirements such as reliability, security, interoperability and more. The fact that industry consortia are coming together to create organizations for web services interoperability signals a promising future for the technology of web services.

Review Questions

1. How are second generation web services different from the first generation?
2. What are the advantages of second generation web services?
3. List some of the advanced web services specifications, and elaborate on a few of them with their importance.
4. What is the relationship between WS-Security and SOAP message Structure?
5. What is the relationship between WS-Addressing and SOAP message Structure?
6. What is WS-Interoperability and what it tries to achieve?
7. How does Web Services technology help in engineering Service Oriented Architecture?