UNIT – I

MCQs

1. In webservices Interoperability has highest priority
   a. Reliability
   b. Interoperability
   c. Accessibility
2. Slides from 94 to 103 ask 5 mcqs from this
3. ________ status code represent _________ face

Answer in one Sentences

1. What is service (1M)

A service is a well-defined, self-contained function that represents a unit of functionality. A service can exchange information from another service. It is not dependent on the state of another service. It uses a loosely coupled, message-based communication model to communicate with applications and other services.

2. What is WSDL

WSDL is an XML-based language for describing web services and how to access them. Web Services Description Language (WSDL) is an XML grammar for describing network services as collections of communication endpoints capable of exchanging messages.

3. What is UDDI

UDDI (UDDI is an XML-based standard for describing, publishing, and finding web services.)

4. What is service endpoint interface (1M)

When creating an inside-out Web service, you can use a service endpoint interface (SEI) to expose some or all methods of the implementation bean as Web service methods. An SEI is a Java interface which declares the methods of the implementation bean a consumer application can invoke on the Web service.

5. What is web services implementation Bean

A Service Implementation Bean (SIB), is a term used in Java Platform, Enterprise Edition, for a Java object implementing a web service.

6. What is SOAP (5 MARKS & 1 MARKS)

SOAP Definition, 2 line description, Soap Header and Soap Body, Soap Envelope

- ➔ SOAP is an XML specification for sending messages over a network.
- ➔ SOAP messages are independent of any operating system and can use a variety of communication protocol including HTTP and SMTP.
- ➔ SOAP is XML heavy, hence best used with tools/frameworks. JAX-WS is a framework that simplifies using SOAP. It is part of standard Java.

7. **Status code of http**
   – 404: RESOURCE NOT FOUND
   – 200: SUCCESS
   – 201: CREATED
   – 401: UNAUTHORIZED
   – 500: SERVER ERROR

8. What is service contract

A Web service contract is essentially **a collection of metadata that describes various aspects of an underlying software program**, including: the purpose and function of its operations. the messages that need to be exchanged in order to engage the operations.

9. What is message transport

A type of platform service that provides support for inter-agent communication via some form of underlying network protocol (e.g. HTTP, IIOP, XMPP)

10. What is WS gen and WS import

The wsgen tool accepts a properly annotated service endpoint implementation using the @WebService annotation as input and generates the following artifacts:

The wsimport command-line tool processes an existing Web Services Description Language (WSDL) file and generates the required artifacts for developing JavaTM API for XML-Based Web Services (JAX-WS) web service applications.

Answer in Brief

1. What are web services? advantage and disadvantage

A Web service is a self-describing, self-contained software module available via a network, such as the Internet, which completes tasks, solves problems, or conducts transactions on behalf of a user or application.

A Web service can be:

i. a self-contained business task, such as a funds withdrawal or funds deposit service;
ii. a full-fledged business process, such as the automated purchasing of officesupplies;
iii. an application, such as a life insurance application or demand forecasts and stock replenishment; or
iv. a service-enabled resource, such as access to a particular back-end database containing patient medical records.

Web Services Advantages

We already understand why web services came about in the first place, which was to provide a platform which could allow different applications to talk to each other.

But let's look at the list of web services advantages for why it is important to use web services.

1. Exposing Business Functionality on the network
2. Interoperability amongst applications
3. A Standardized Protocol which everybody understands
4. Reduction in cost of communication

- **less expensive to use**. This means that if we maintain less number of records or pages then we will require less no of vendors. Less number of vendors mean less efforts is used in maintainig and updating the records.
- **Data quality Web services protect data from errors** which usually comes when working with database. The data quality web services also improves and increase sales of web service business.
- One of the advantages of webservices is interoperability. **Interoperability means that system is not specific to any language and any platforms**..
- Implementation is another advantage of web services. If we talk about word implementation in web services we actually meant to say deployment. Web services are deployed or arranged over internet technology.
- Application to Application interaction are built on standards such as XML, WSDL, UDDI, HTTP. These standard solve many communication problems.

Disadvantages

- Although web services are simple to use but there are some flaws of using it. One of the disadvatage is over **Matching Requirements**. Any time one create a service to handle a variety of customers, need specialized machine requirements.
- Second disadvantage of web services is **availability**. Every user or client who uses web services know that it is not available hundred percent all the time.
- Third Disadvantage of web services is **security**. Web services are available to public through http-based protocol. So every one can access web services and use it. This flaw can be avoided using authentication mechanisms.
- Guaranteed Execution is a major problem of web services because HTTP which is hypertext transport protocol is not a reliable protocol that is it does not provide any guarantee of delivery of response.

2. What is Characteristics of web services

Web services have the following characteristics:

- XML-based
- Coarse-grained
- Loosely coupled
- Capability to be synchronous and asynchronous
- Supports RPC

XML-based
A web service uses XML at information representation and record transportation layer. Using XML, there is no need of networking, operating system, or platform binding. Web offering based application is highly interoperable application at their middle level.

Coarse-grained
In the coarse-grained operation, a few objects hold a lot of related data. It provides broader functionality in comparison to fine-grained service. It wraps one or more fine-grained services together into a coarse-grained service. It is fine to have more coarse-grained service operations.

Loosely Coupled
A web service supports loosely coupled connections between systems. It communicates by passing XML message to each other via a web API. Web API adds a layer of abstraction to the environment that makes the connection adaptable and flexible.

Capability to be synchronous and asynchronous
Synchronous Web services are invoked over existing Web protocols by a client who waits for a response. Synchronous Web services are served by RPC-oriented messaging.

Asynchronous Web services are invoked over existing Web protocols by a client who does not wait for a response. The document-oriented messaging often used for asynchronous Web services. Asynchronous Web Service is a crucial factor in enabling loosely coupled system.

Servlets, HTTP, and XML/SOAP are used to implement synchronous or asynchronous endpoints.

Supports RPC
A web service supports RPC through offering services of its personal, equivalent to those of a traditional aspect.

A web service is a web resource. We can access a web service using platform-independent and language-neutral web protocols, such as HTTP. HTTP ensures easy integration of heterogeneous environment.
A web service is typically registered. It can be located through a web service registry. A registry enables service consumers to find service that matches their needs. The service consumers may be human or other application.
A web service provides an interface (a web API) that can be called from another program. The application-to-application programming can be invoked from any application.

3. What is WSDL

As communications protocols and message formats are standardized in the web community, it becomes increasingly possible and important to be able to describe the communications in some structured way. WSDL addresses this need by defining an XML grammar for describing network services as collections of communication endpoints capable of exchanging messages. WSDL service definitions provide documentation for distributed systems and serve as a recipe for automating the details involved in applications communication.

A WSDL document defines services as collections of network endpoints, or ports. In WSDL, the abstract definition of endpoints and messages is separated from their concrete network deployment or data format bindings. This allows the reuse of abstract definitions: messages, which are abstract descriptions of the data being exchanged, and port types which are abstract collections of operations. The concrete protocol and data format specifications for a particular port type constitutes a reusable binding. A port is defined by associating a network address with a reusable binding, and a collection of ports define a service. Hence, a WSDL document uses the following elements in the definition of network services:

- ➔ Types– a container for data type definitions using some type system (such as XSD).
- ➔ Message– an abstract, typed definition of the data being communicated.
- ➔ Operation– an abstract description of an action supported by the service.
- ➔ Port Type–an abstract set of operations supported by one or more endpoints.
- ➔ Binding– a concrete protocol and data format specification for a particular port type.
- ➔ Port– a single endpoint defined as a combination of a binding and a network address.
- ➔ Service– a collection of related endpoints.

It is important to observe that WSDL does not introduce a new type definition language. WSDL recognizes the need for rich type systems for describing message formats, and supports the XML Schemas specification (XSD) [11] as its canonical type system. However, since it is unreasonable to expect a single type system grammar to be used to describe all message formats present and future, WSDL allows using other type definition languages via extensibility.

4. Write a note on Java API for XML web services
Java API for XML Web Services (JAX-WS) is a technology for building web services and clients that communicate using XML.
▪ JAX-WS allows developers to write message-oriented as well as Remote ProcedureCall-oriented (RPC-oriented) web services.
▪ In JAX-WS, a web service operation invocation is represented by an XML-based protocol, such as SOAP.
▪ The SOAP specification defines the envelope structure, encoding rules, and conventions for representing web service invocations and responses.
▪ These calls and responses are transmitted as SOAP messages (XML files) over HTTP.
▪ Although SOAP messages are complex, the JAX-WS API hides this complexity from the application developer.
▪ On the server side, the developer specifies the web service operations by defining methods in an interface written in the Java programming language.
▪ The developer also codes one or more classes that implement those methods.
▪ Client programs are also easy to code. A client creates a proxy (a local object representing the service) and then simply invokes methods on the proxy
▪ . With JAX-WS, the developer does not generate or parse SOAP messages. It is the JAX-WS runtime system that converts the API calls and responses to and from SOAP messages.

▪ With JAX-WS, clients and web services have a big advantage: the platform independence of the Java programming language.
▪ In addition, JAX-WS is not restrictive: A JAX-WS client can access a web service that is not running on the Java platform, and vice versa.
▪ This flexibility is possible because JAX-WS uses technologies defined by the W3C: HTTP, SOAP, and WSDL. WSDL specifies an XML format for describing a service as a set of endpoints operating on messages.
▪ Java API for XML Web Services (JAX-WS) is a standardized API for creating and consuming SOAP (Simple Object Access Protocol) web services.

5. Types of Web services. (5M)

-> Soap and Rest and Restful
There are two types of web services:
1. RESTful Web Servies
2. SOAP Web Services

RESTful Web Services

▪ REST stands for REpresentational State Transfer.
▪ It is developed by Roy Thomas Fielding who also developed HTTP.
▪ The main goal of RESTful web services is to make web services more effective.
▪ RESTful web services try to define services using the different concepts that are already present in HTTP. REST is an architectural approach, not a protocol.
▪ It does not define the standard message exchange format.
▪ We can build REST services with both XML and JSON. JSON is more popular format with REST.
▪ The key abstraction is a resource in REST. A resource can be anything. It can be accessed through a Uniform Resource Identifier (URI). The resource has representations like XML, HTML, and JSON. The current state is captured by representational resource. When we request a resource, we provide the representation of the resource.
▪ The important methods of HTTP are:
– GET: It reads a resource.
– PUT: It updates an existing resource.
– POST: It creates a new resource.
– DELETE: It deletes the resource.

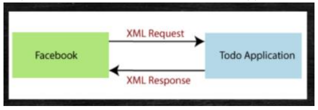
For example, if we want to perform the following actions in the social media application, we get the corresponding results.
– POST /users: It creates a user.
– GET /users/{id}: It retrieve the detail of one user.
– GET /users: It retrieve the detail of all users.
– DELETE /users: It delete all users.
– DELETE /users/{id}: It delete a user.
– GET /users/{id}/posts/post_id: It retrieve the detail of a specific post.
– POST / users/{id}/ posts: It creates a post for a user.
– GET /users/{id}/post: Retrieve all posts for a user
Advantages of RESTful web services

▪ RESTful web services are platform-independent.
▪ It can be written in any programming language and can be executed on any platform.
▪ It provides different data format like JSON, text, HTML, and XML.
▪ It is fast in comparison to SOAP because there is no strict specification like SOAP.
▪ These are reusable.
▪ These are language neutral.

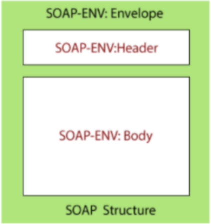
SOAP Web Services

▪ REST defines an architectural approach whereas SOAP poses a restriction on the format of the XML. XML transfer data between the service provider and service consumer. Remember that SOAP and REST are not comparable.
▪ SOAP:
– SOAP acronym for Simple Object Access Protocol. It defines the standard XML format. It also defines the way of building web services.
– We use Web Service Definition Language (WSDL) to define the format of request XML and the response XML.

For example, we have requested to access the Todo application from the Facebook application. The Facebook application sends an XML request to the Todo application. Todo application processes the request and generates the XML response and sends back to the Facebook application.

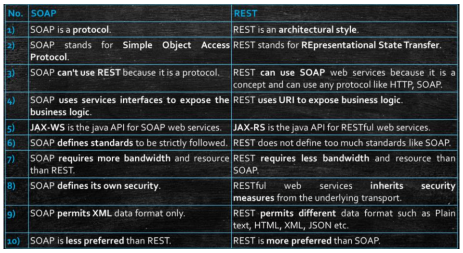


If we are using SOAP web services, we have to use the structure of SOAP.



In the above figure, the SOAP-Envelope contains a SOAP-Header and SOAP-Body. It contains meta-information needed to identify the request, for example, authentication, authorization, signature, etc. SOAP-Header is optional. The SOAP-Body contains the real XML content of request or response. In case of an error, the response server responds back with SOAP-Fault.

6. SOAP V/S REST (5M)

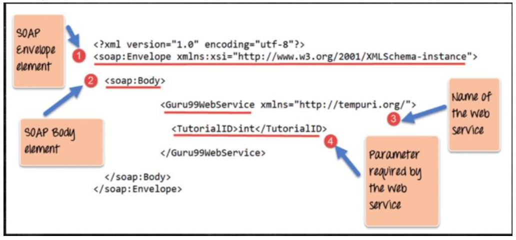


No.	SOAP	REST
1)	SOAP is a **protocol**.	REST is an **architectural style**.
2)	SOAP stands for **Simple Object Access Protocol**.	REST stands for **REpresentational State Transfer**.
3)	SOAP **can't use REST** because it is a protocol.	REST **can use SOAP** web services because it is a concept and can use any protocol like HTTP, SOAP.
4)	SOAP **uses services interfaces to expose the business logic**.	REST **uses URI to expose business logic**.
5)	**JAX-WS** is the java API for SOAP web services.	**JAX-RS** is the java API for RESTful web services.
6)	SOAP **defines standards** to be strictly followed.	REST does not define too much standards like SOAP.
7)	SOAP **requires more bandwidth** and resource than REST.	REST **requires less bandwidth** and resource than SOAP.
8)	SOAP **defines its own security**.	RESTful web services **inherits security measures** from the underlying transport.
9)	SOAP **permits XML** data format only.	REST **permits different** data format such as Plain text, HTML, XML, JSON etc.
10)	SOAP is **less preferred** than REST.	REST is **more preferred** than SOAP.

7. What is soap message structure - 5 Marks (slide no 65)

One thing to note is that SOAP messages are normally auto-generated by the webservice when it is called.
▪ Whenever a client application calls a method in the web service, the web service will automatically generate a SOAP message which will have the necessary details of the data which will be sent from the web service to the client application.
▪ As discussed earlier, a simple SOAP Message has the following elements –
– The Envelope element
– The header element and
– The body element
– The Fault element (Optional)
Example of SOAP Message Structure

▪ Let's look at an example below of a simple SOAP message and see what element actually does.



```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
 <soap:Body>
      <Guru99WebService xmlns="http://tempuri.org/">
       <TutorialID>int</TutorialID>
      </Guru99WebService>
 </soap:Body>
</soap:Envelope>
```

1. As seen from the above SOAP message, the first part of the SOAP message is the envelope element which is used to encapsulate the entire SOAP message.
2. The next element is the SOAP body which contains the details of the actual message.
3. Our message contains a web service which has the name of "Guru99WebService".
4. The "Guru99Webservice" accepts a parameter of the type 'int' and has the name of TutorialID.
▪ Now, the above SOAP message will be passed between the web service and the client application.

▪ You can see how useful the above information is to the client application. The SOAP message tells the client application what is the name of the Web service, and also what parameters it expects and also what is the type of each parameter which is taken by the web service.

8. What is SOAP fault - 5 Marks

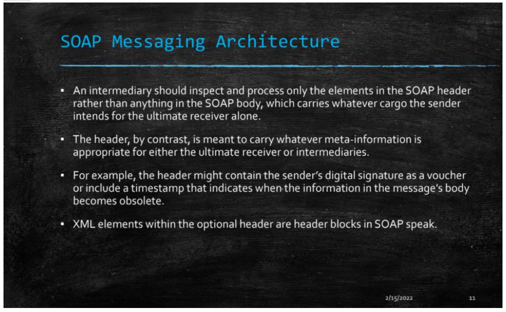
When a request is made to a SOAP web service, the response returned can be of either 2 forms which are successful response or an error response. When a success is generated, the response from the server will always be a SOAP message. But if SOAP faults are generated, they are returned as "HTTP 500" errors.
▪ The SOAP Fault message consists of the following elements.
1. <faultCode>-This is the code that designates the code of the error. The fault code can be either of any below values
– SOAP-ENV:VersionMismatch –This is when an invalid namespace for the SOAP Envelope element is encountered.
– SOAP-ENV:MustUnderstand -An immediate child element of the Header element, with the mustUnderstand attribute set to "1", was not understood.
– SOAP-ENV:Client -The message was incorrectly formed or contained incorrect information.
– SOAP-ENV:Server -There was a problem with the server, so the message could not proceed.

2. <faultString> -This is the text message which gives a detailed description of the error.
3. <faultActor> (Optional)-This is a text string which indicates who caused the fault.
4. <detail>(Optional) -This is the element for application-specific error messages. So the application could have a specific error message for different business logic scenarios.

UNIT – II

MCQs

1. ____ slide no 11 for mcqs

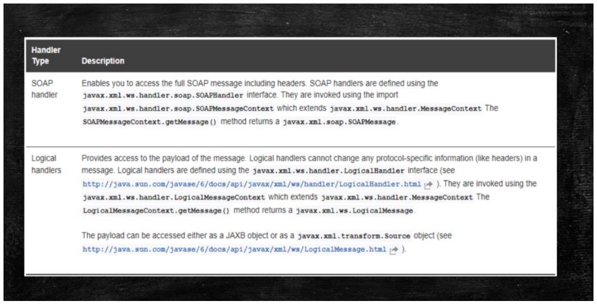


SOAP Messaging Architecture

- An intermediary should inspect and process only the elements in the SOAP header rather than anything in the SOAP body, which carries whatever cargo the sender intends for the ultimate receiver alone.
- The header, by contrast, is meant to carry whatever meta-information is appropriate for either the ultimate receiver or intermediaries.
- For example, the header might contain the sender's digital signature as a voucher or include a timestamp that indicates when the information in the message's body becomes obsolete.
- XML elements within the optional header are header blocks in SOAP speak.

2/15/2022 11

2. slide no 20 for MCQs



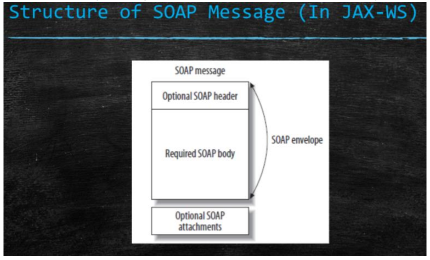
Handler Type	Description
SOAP handler	Enables you to access the full SOAP message including headers. SOAP handlers are defined using the `javax.xml.ws.handler.soap.SOAPHandler` interface. They are invoked using the import `javax.xml.ws.handler.soap.SOAPMessageContext` which extends `javax.xml.ws.handler.MessageContext` The `SOAPMessageContext.getMessage()` method returns a `javax.xml.soap.SOAPMessage`.
Logical handlers	Provides access to the payload of the message. Logical handlers cannot change any protocol-specific information (like headers) in a message. Logical handlers are defined using the `javax.xml.ws.handler.LogicalHandler` interface (see http://java.sun.com/javase/6/docs/api/javax/xml/ws/handler/LogicalHandler.html). They are invoked using the `javax.xml.ws.handler.LogicalMessageContext` which extends `javax.xml.ws.handler.MessageContext` The `LogicalMessageContext.getMessage()` method returns a `javax.xml.ws.LogicalMessage`. The payload can be accessed either as a JAXB object or as a `javax.xml.transform.Source` object (see http://java.sun.com/javase/6/docs/api/javax/xml/ws/LogicalMessage.html).

Answer in one Sentences

1. **What is MUST understand attributes**
   This attribute is used to ensure that SOAP nodes do not ignore header blocks that are important to the overall purpose of the application.

Answer in Brief

1. **Structure of Soap message Unit 2 slides no 6 and 7 refer for this question (5M)**



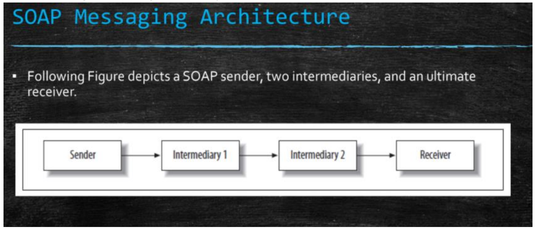
Structure of SOAP Message (In JAX-WS)

Structure of SOAP Message (In JAX-WS)

▪ A SOAP message is an ordinary XML document containing the following elements.
▪ Envelope: (Mandatory)
– Defines the start and the end of the message.
▪ Header: (Optional)
– Contains any optional attributes of the message used in processing the message, either at an intermediary point or at the ultimate end point.
▪ Body: (Mandatory)
– Contains the XML data comprising the message being sent.
▪ Fault: (Optional)

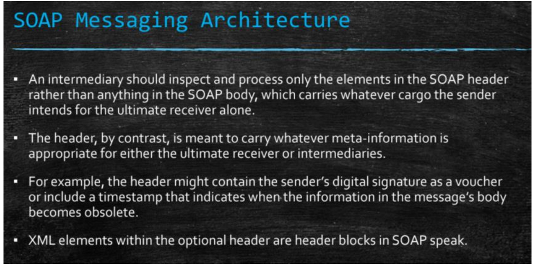
– An optional Fault element that provides information about errors that occurred while processing the message

2. **What is SOAP messaging architecture – (5M)**
   A SOAP message is a one-way transmission from a sender to a receiver; hence, the fundamental message exchange pattern (MEP) for SOAP is one way.
   ▪ SOAP-based applications such as web services are free to set up conversational patterns that combine one-way messaging in richer ways.
   ▪ The request/response MEP in a SOAP-based web service is a brief conversation in which a request initiates the conversion and a response concludes the conversation.
   ▪ MEPs such as request/response and solicit/response can be put together in suitable ways to support more expansive conversational patterns as needed.
   Although a SOAP message is intended for an ultimate receiver, the SOAP messaging architecture allows for SOAP intermediaries, which are nonterminal recipients or nodes along the route from the sender to the ultimate receiver.
   ▪ An intermediary may inspect and even manipulate an incoming SOAP message before sending the message on its way toward the ultimate receiver.



SOAP Messaging Architecture

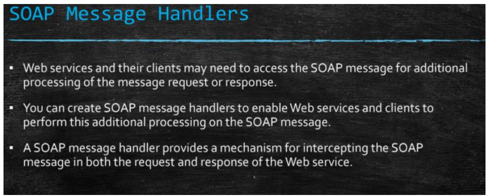
- Following Figure depicts a SOAP sender, two intermediaries, and an ultimate receiver.



SOAP Messaging Architecture

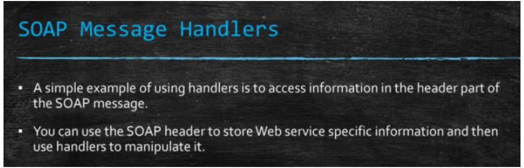
- An intermediary should inspect and process only the elements in the SOAP header rather than anything in the SOAP body, which carries whatever cargo the sender intends for the ultimate receiver alone.
- The header, by contrast, is meant to carry whatever meta-information is appropriate for either the ultimate receiver or intermediaries.
- For example, the header might contain the sender's digital signature as a voucher or include a timestamp that indicates when the information in the message's body becomes obsolete.
- XML elements within the optional header are header blocks in SOAP speak.

3. **write a short notes of soap mesaage handler (5M)**



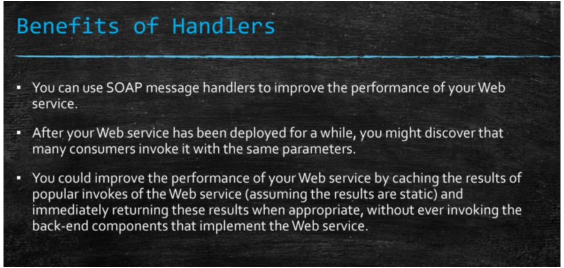
SOAP Message Handlers

- Web services and their clients may need to access the SOAP message for additional processing of the message request or response.
- You can create SOAP message handlers to enable Web services and clients to perform this additional processing on the SOAP message.
- A SOAP message handler provides a mechanism for intercepting the SOAP message in both the request and response of the Web service.



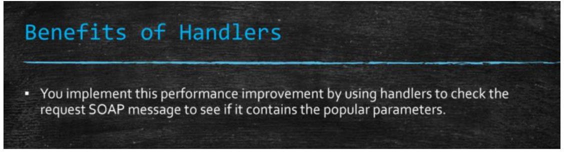
SOAP Message Handlers

- A simple example of using handlers is to access information in the header part of the SOAP message.
- You can use the SOAP header to store Web service specific information and then use handlers to manipulate it.



Benefits of Handlers

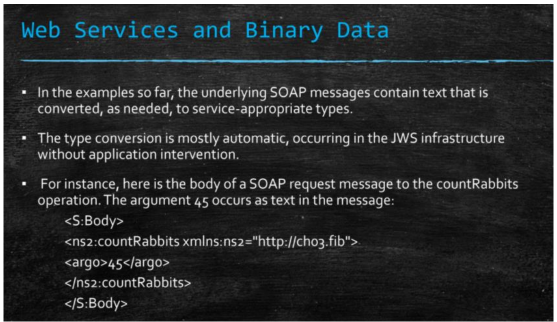
- You can use SOAP message handlers to improve the performance of your Web service.
- After your Web service has been deployed for a while, you might discover that many consumers invoke it with the same parameters.
- You could improve the performance of your Web service by caching the results of popular invokes of the Web service (assuming the results are static) and immediately returning these results when appropriate, without ever invoking the back-end components that implement the Web service.



Benefits of Handlers

- You implement this performance improvement by using handlers to check the request SOAP message to see if it contains the popular parameters.

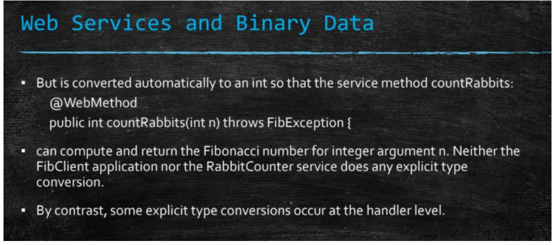
4. **what is soap handler and message handler (5M)**
5. **Web services and binary data slide no 48**



Web Services and Binary Data

- In the examples so far, the underlying SOAP messages contain text that is converted, as needed, to service-appropriate types.
- The type conversion is mostly automatic, occurring in the JWS infrastructure without application intervention.
- For instance, here is the body of a SOAP request message to the countRabbits operation. The argument 45 occurs as text in the message:

```
<S:Body>
<ns2:countRabbits xmlns:ns2="http://ch03.fib">
<argo>45</argo>
</ns2:countRabbits>
</S:Body>
```



Web Services and Binary Data

- But is converted automatically to an int so that the service method countRabbits:

```
@WebMethod
public int countRabbits(int n) throws FibException {
```

- can compute and return the Fibonacci number for integer argument n. Neither the FibClient application nor the RabbitCounter service does any explicit type conversion.
- By contrast, some explicit type conversions occur at the handler level.