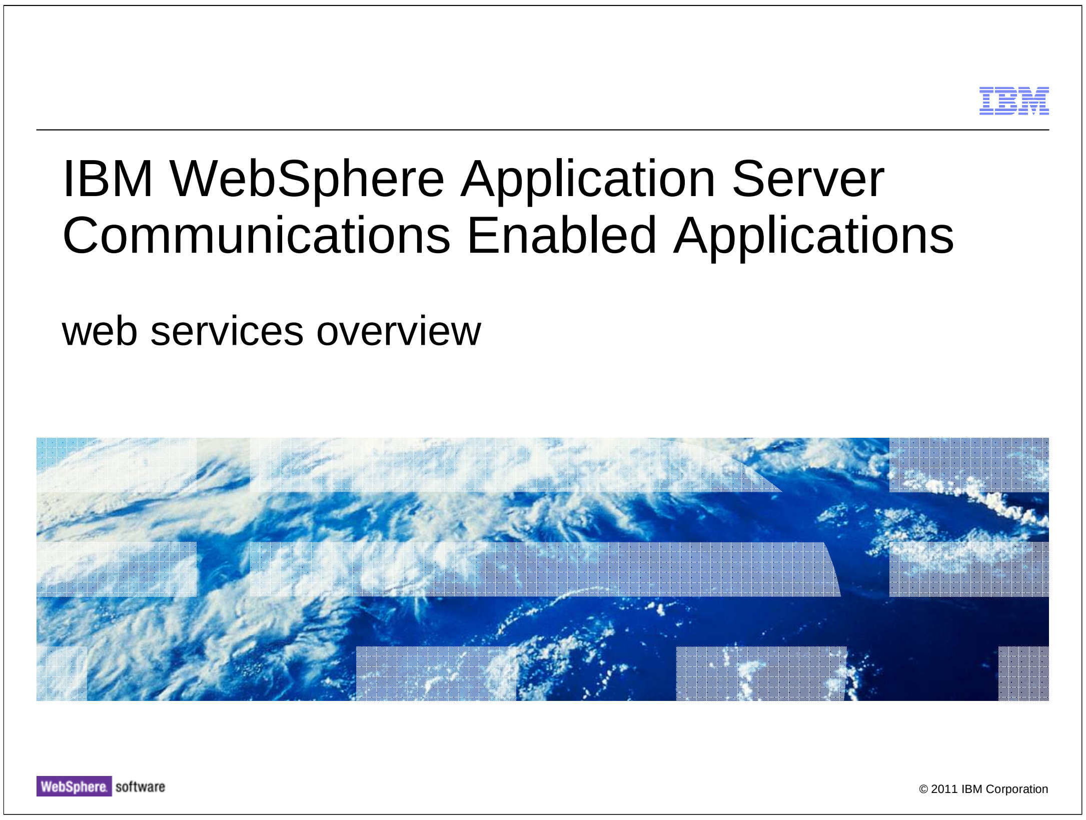
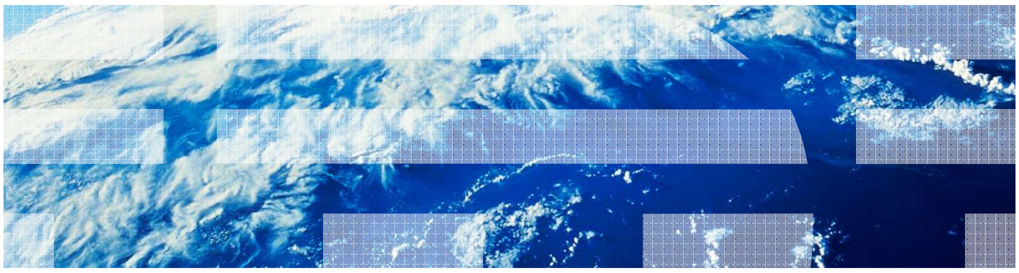


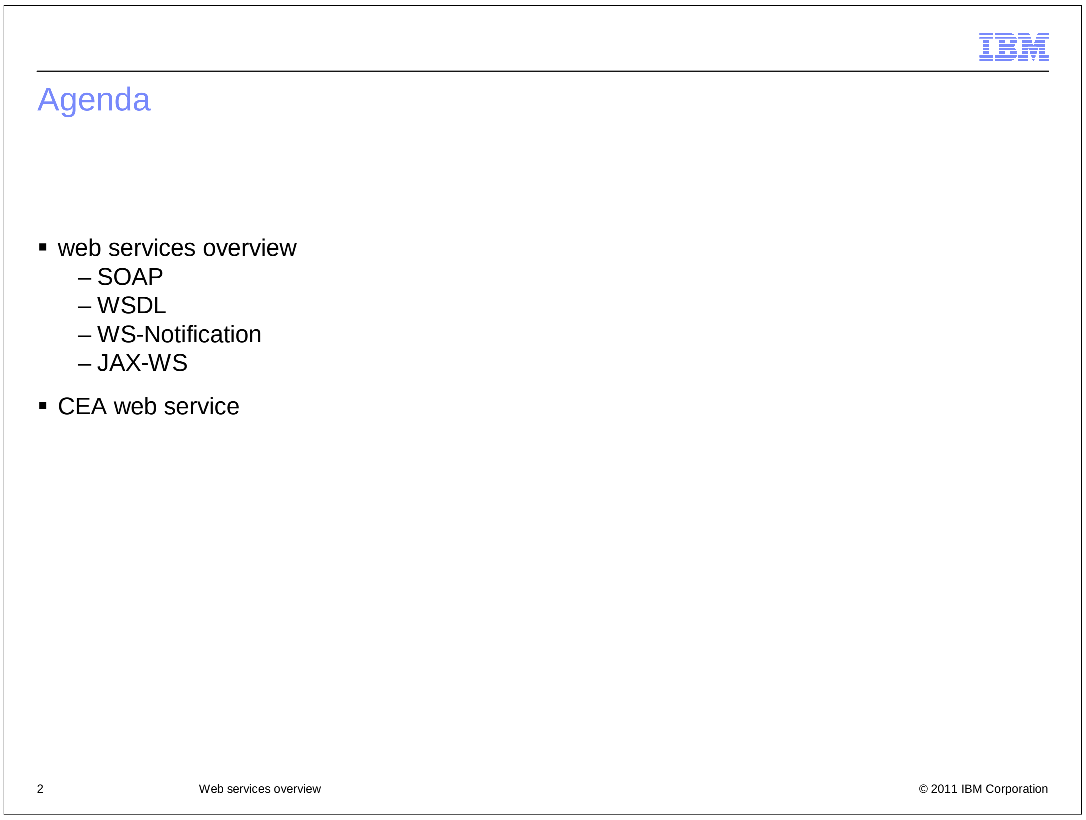
IBM WebSphere Application Server Communications Enabled Applications

web services overview



WebSphere software

This presentation goes through some of the basics for web services and then discusses how the Communications Enabled Applications feature of WebSphere Application Server allows you to access telephony services with Web service clients.



Agenda

- web services overview
  - SOAP
  - WSDL
  - WS-Notification
  - JAX-WS
- CEA web service

2 Web services overview © 2011 IBM Corporation

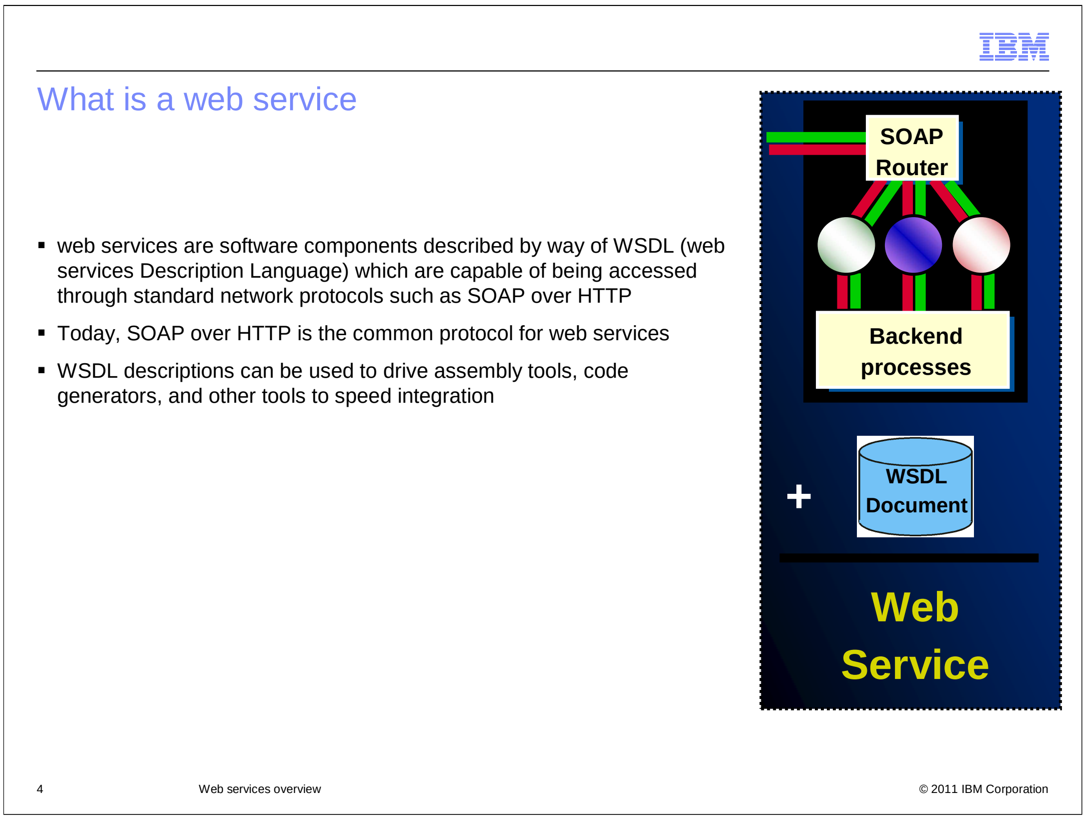
The first few slides will discuss concepts such as SOAP, WSDL, WS-Notification, and JAX-WS. Then there is a discussion of the CEA web service.

Section

web services overview

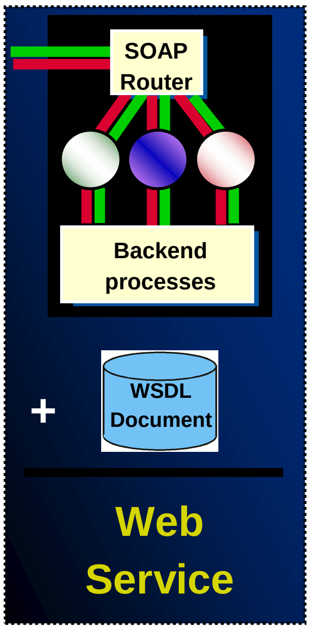
3 Web services overview © 2011 IBM Corporation

The first few slides will go over some basic web services concepts.



What is a web service

- web services are software components described by way of WSDL (web services Description Language) which are capable of being accessed through standard network protocols such as SOAP over HTTP
- Today, SOAP over HTTP is the common protocol for web services
- WSDL descriptions can be used to drive assembly tools, code generators, and other tools to speed integration

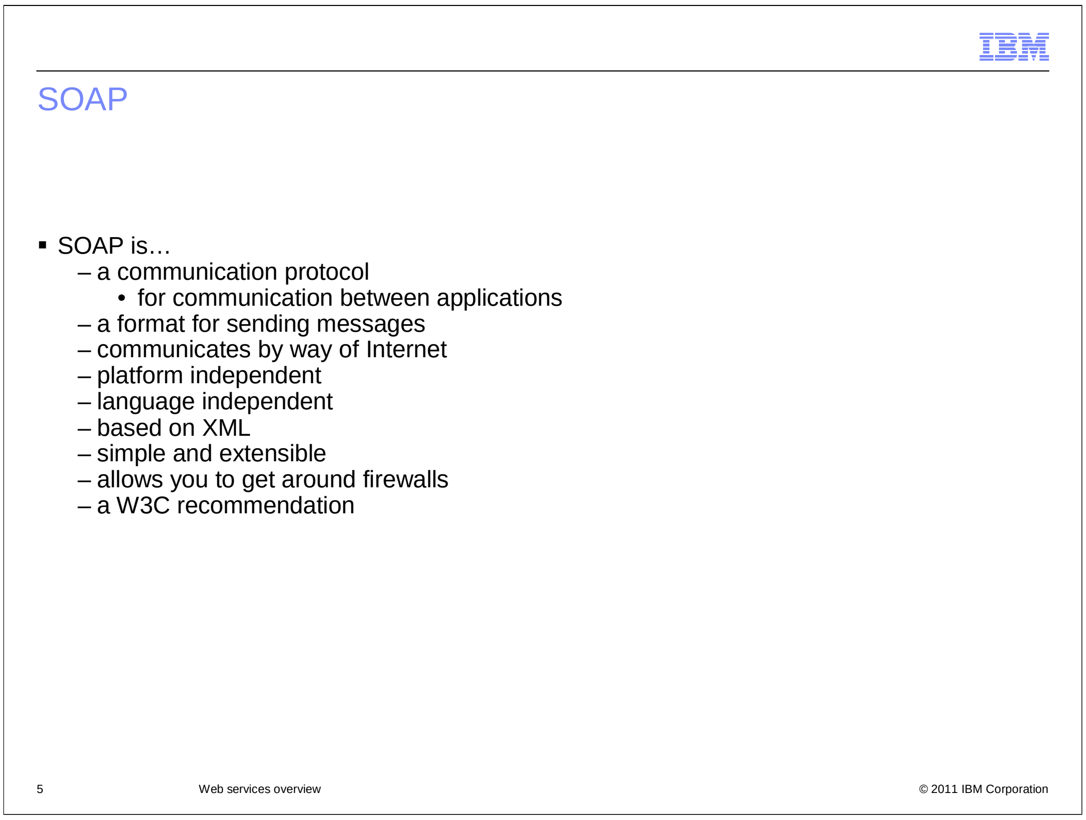


4 Web services overview © 2011 IBM Corporation

A web service is a software system designed to support interoperable machine-to-machine interaction over a network. It has an interface described in a machine-processable format, specifically WSDL. Other systems interact with the web service in a manner prescribed by its description using SOAP messages, typically conveyed using HTTP with an XML serialization in conjunction with other Web-related standards.

Web services are self-contained, self-describing modular applications that can be published, located, and invoked across the Web. On the client side, no additional software is required. A programming language with XML and HTTP client support is enough to get you started. On the server side, a Web server and Servlet engine are required. The client and server can be implemented in different environments. It is possible for a Web service to enable an existing application without writing a single line of code. The client and server need to recognize only the format and content of request and response messages. The definition of the message format travels with the message; no external metadata repositories or code generation tools are required. Simple Web services can be aggregated to form more complex Web services either by using workflow techniques or by calling lower layer Web services from a Web service implementation. Web services are based on a concise set of open, XML-based standards designed to promote interoperability between a Web service and clients across a variety of computing platforms and programming languages.

Web services might be anything, for example, theatre review articles, weather reports, credit checks, stock quotations, travel advisories, or airline travel reservation processes. Each of these self-contained business services is an application that can easily integrate with other services, from the same or different companies, to create a complete business process. This interoperability allows businesses to dynamically publish, discover, and bind a range of Web services through the internet.



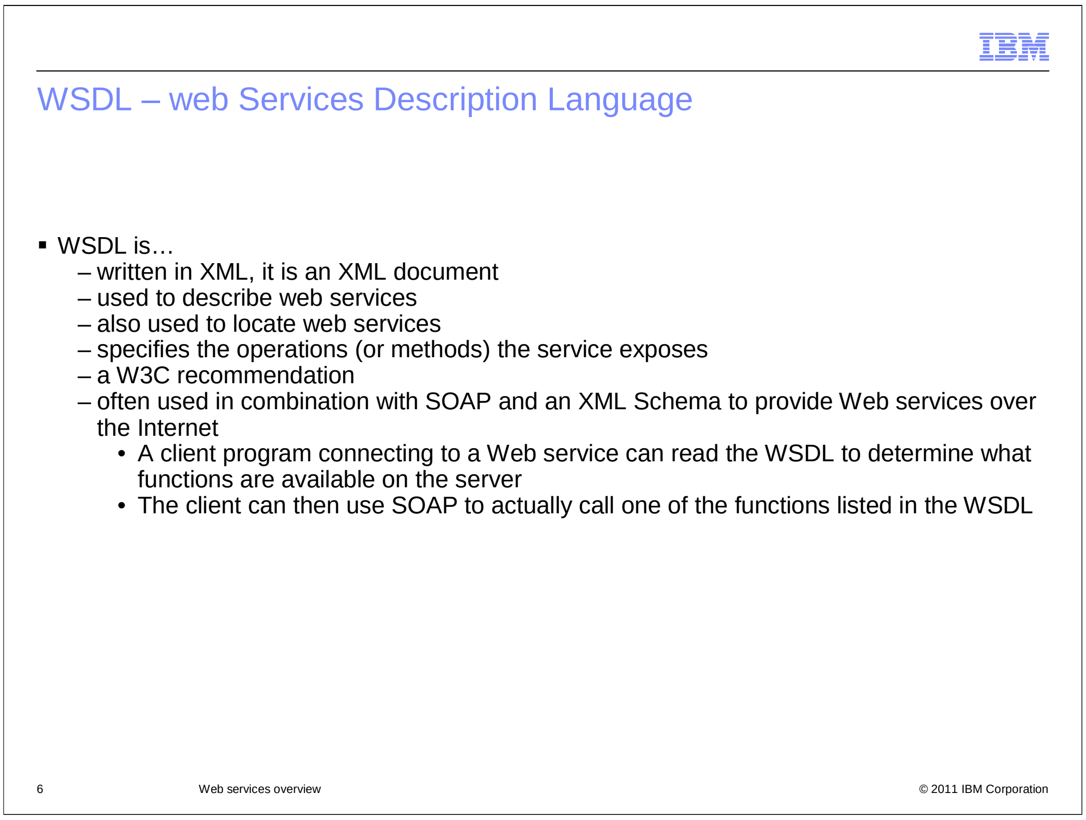
SOAP

- SOAP is…
  - a communication protocol
    - for communication between applications
  - a format for sending messages
  - communicates by way of Internet
  - platform independent
  - language independent
  - based on XML
  - simple and extensible
  - allows you to get around firewalls
  - a W3C recommendation

5 Web services overview © 2011 IBM Corporation

SOAP, originally defined as *Simple Object Access Protocol*, is a communication protocol specification for exchanging structured information in the implementation of web services in computer networks. It relies on XML as its message format, and typically relies on HTTP for message negotiation and transmission. SOAP provides a basic messaging framework upon which web services can be built.

SOAP is important for application development to allow Internet communication between programs. Today's applications communicate using Remote Procedure Calls (RPC) between objects like DCOM and CORBA, but HTTP was not designed for this. RPC represents a compatibility and security problem; firewalls and proxy servers will normally block this kind of traffic. A better way to communicate between applications is over HTTP, because HTTP is supported by all Internet browsers and servers. SOAP was created to accomplish this. SOAP provides a way to communicate between applications running on different operating systems, with different technologies and programming languages.



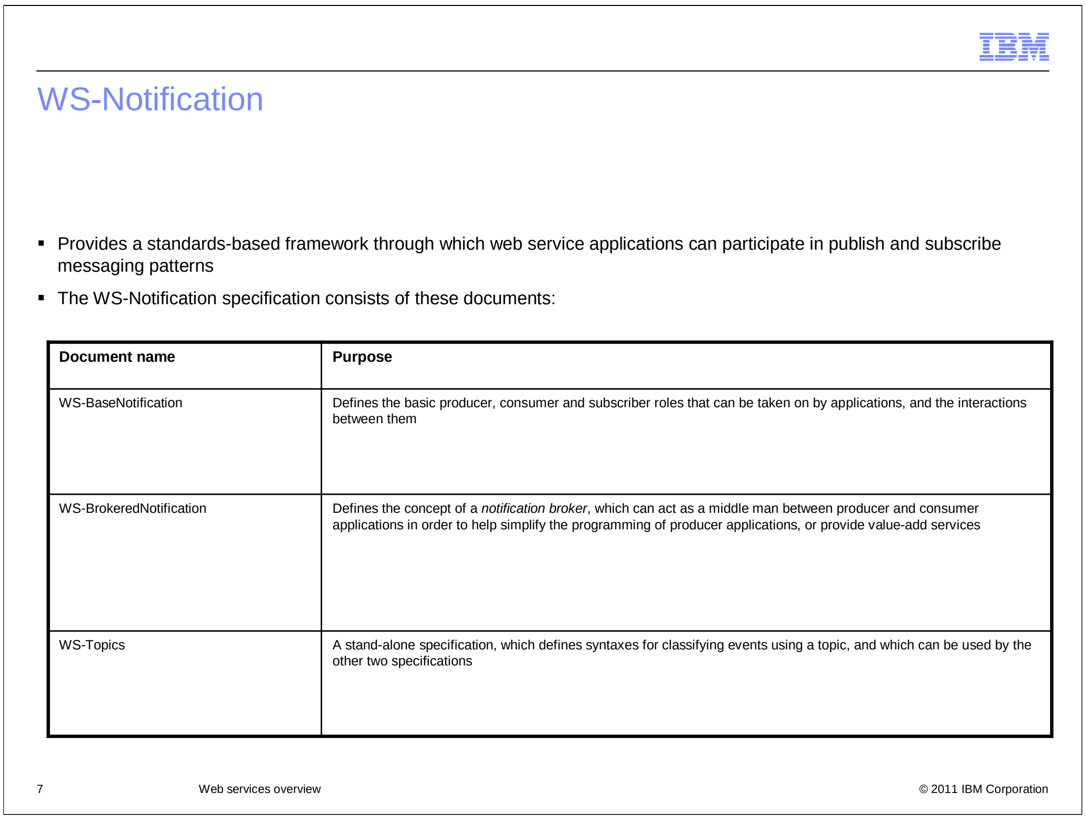
WSDL – web Services Description Language

- WSDL is…
  - written in XML, it is an XML document
  - used to describe web services
  - also used to locate web services
  - specifies the operations (or methods) the service exposes
  - a W3C recommendation
  - often used in combination with SOAP and an XML Schema to provide Web services over the Internet
    - A client program connecting to a Web service can read the WSDL to determine what functions are available on the server
    - The client can then use SOAP to actually call one of the functions listed in the WSDL

6 Web services overview © 2011 IBM Corporation

The mechanics of the message exchange are documented in a web service description (WSD) for a web service. The WSD is a machine-processable specification of the Web service's interface, written in WSDL. WSDL is an XML-based language that provides a model for describing Web services. A WSDL document specifies the location of the service and the operations (or methods) the service exposes. It defines the message formats, data types, transport protocols, and transport serialization formats that should be used between the requester agent and the provider agent. It also specifies one or more network locations at which a provider agent can be invoked, and can provide some information about the message exchange pattern that is expected. In essence, the service description represents an agreement governing the mechanics of interacting with that service.

A WSDL document is often used in combination with SOAP and an XML Schema to provide Web services over the Internet. A client program connecting to a Web service can read the WSDL to determine what functions are available on the server. Any special data types used are embedded in the WSDL file in the form of XML Schema. The client can then use SOAP to actually call one of the functions listed in the WSDL.



WS-Notification

- Provides a standards-based framework through which web service applications can participate in publish and subscribe messaging patterns
- The WS-Notification specification consists of these documents:

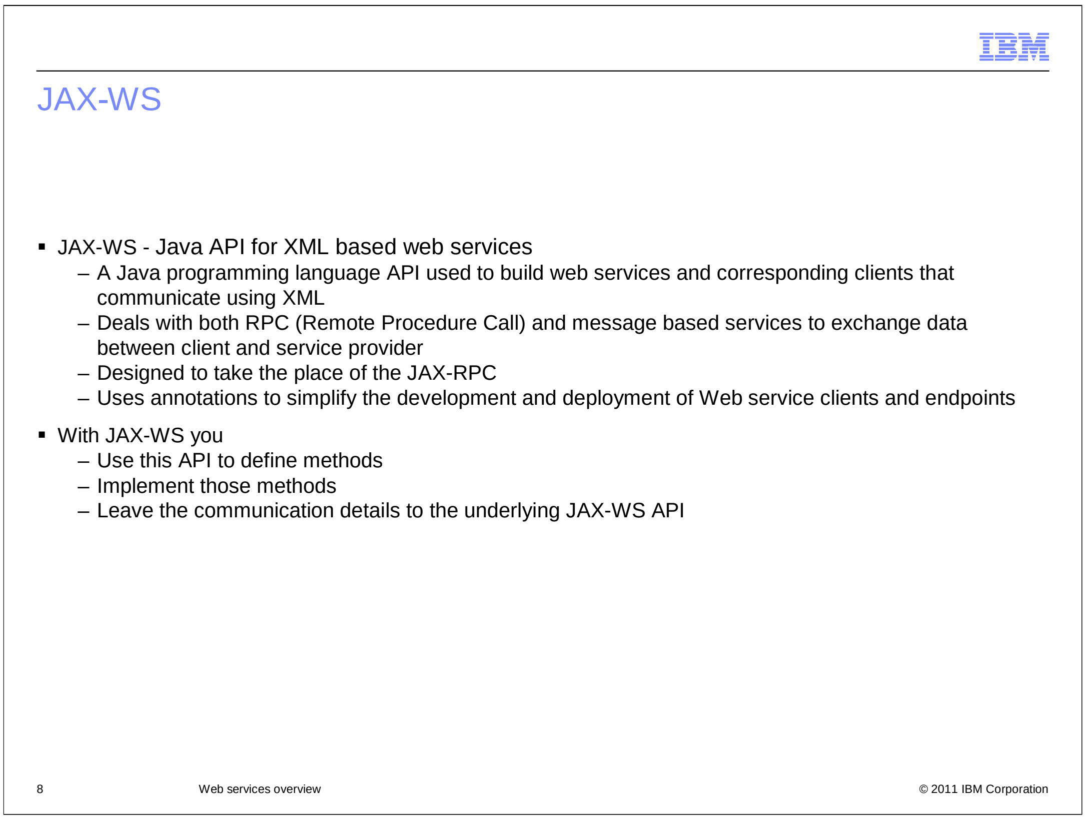
Document name	**Purpose**
WS-BaseNotification	Defines the basic producer, consumer and subscriber roles that can be taken on by applications, and the interactions between them
WS-BrokeredNotification	Defines the concept of a *notification broker*, which can act as a middle man between producer and consumer applications in order to help simplify the programming of producer applications, or provide value-add services
WS-Topics	A stand-alone specification, which defines syntaxes for classifying events using a topic, and which can be used by the other two specifications

7 Web services overview © 2011 IBM Corporation

When creating a web application you might want to have an event-driven, or Notification-based, interaction pattern for inter-object communications. Examples exist in many domains, for example in publish/subscribe systems are provided by Message Oriented Middleware vendors, or in system and device management domains. This notification pattern is increasingly being used in a web services context.

WS-Notification is a family of related specifications that define a standard Web services approach to notification using a topic-based publish/subscribe pattern. It includes: standard message exchanges to be implemented by service providers that want to participate in Notifications, standard message exchanges for a notification broker service provider, operational requirements expected of service providers and requestors that participate in notifications, and an XML model that describes topics. The WS-Notification family of documents includes three normative specifications: WS-BaseNotification, WS-BrokeredNotification, and WS-Topics.

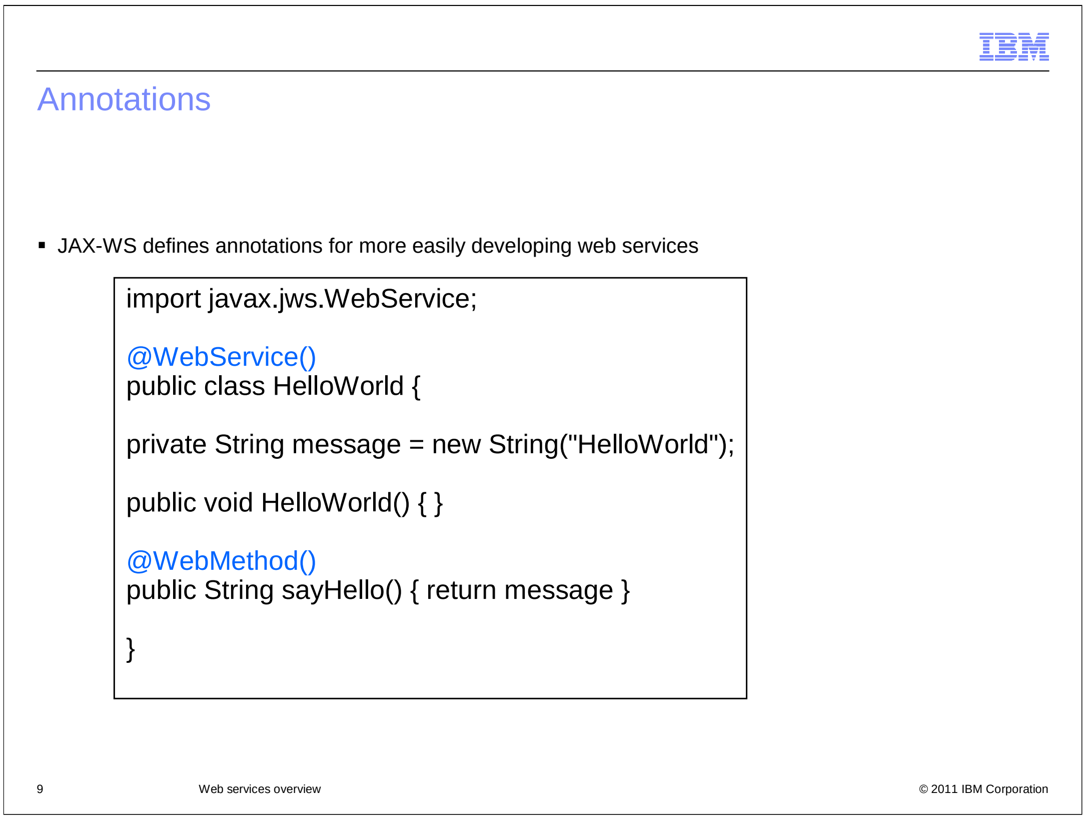
The goal of WS-BaseNotification is to standardize the terminology, concepts, operations, WSDL and XML needed to express the basic roles involved in Web services publish and subscribe for notification message exchange. It defines the basic producer, consumer and subscriber roles that can be taken on by applications, and the interactions between them. WS-BrokeredNotification defines the concept of a notification broker, which can act as a middle man between producer and consumer applications in order to help simplify the programming of producer applications, or provide value-add services. It includes standard message exchanges to be implemented by NotificationBroker service providers along with operational requirements expected of service providers and requestors that participate in brokered notifications. WS-Topics define a mechanism to organize and categorize items of interest for subscription known as "topics. " It defines a set of topic expression dialects (syntaxes) for classifying events using a topic that can be used as subscription expressions in subscribe request messages and other parts of the WS-Notification system.


JAX-WS

- JAX-WS - Java API for XML based web services
  - A Java programming language API used to build web services and corresponding clients that communicate using XML
  - Deals with both RPC (Remote Procedure Call) and message based services to exchange data between client and service provider
  - Designed to take the place of the JAX-RPC
  - Uses annotations to simplify the development and deployment of Web service clients and endpoints
- With JAX-WS you
  - Use this API to define methods
  - Implement those methods
  - Leave the communication details to the underlying JAX-WS API

8 Web services overview © 2011 IBM Corporation

The Java API for XML web services (JAX-WS) is a Java programming language API for creating web services. It is part of the Java EE platform from Sun Microsystems. Like the other Java EE APIs, JAX-WS uses annotations, introduced in Java SE 5, to simplify the development and deployment of Web service clients and endpoints. JAX-WS began as a follow on to the JAX-RPC specification, but evolved further into its own specification. JAX-WS goes further by dealing with message based services in addition to remote procedure call (RPC) based services. JAX-WS can be used to build Web services and corresponding clients that communicate using XML to send messages or use RPC to exchange data between client and service provider. JAX-WS represents RPC or messages using XML-based protocols such as SOAP, but hides SOAP's innate complexity behind a Java-based API. Developers use this API to define methods, then code one or more classes to implement those methods and leave the communication details to the underlying JAX-WS API. Clients create a local proxy to represent a service, then invoke methods on the proxy. The JAX-WS runtime system converts API calls and matching replies to and from SOAP messages.



Annotations

- JAX-WS defines annotations for more easily developing web services

```
import javax.jws.WebService;

@WebService()
public class HelloWorld {

private String message = new String("HelloWorld");

public void HelloWorld() { }

@WebMethod()
public String sayHello() { return message }

}
```

9 Web services overview © 2011 IBM Corporation

JAX-WS uses annotations, introduced in Java SE 5, to simplify the development and deployment of web service clients and endpoints. Some of the annotations used come from JSR-181 rather than the JAX-WS specification. JAX-WS defines its own set of annotations to cover the portions that are not addressed in JSR-181. When combined they allow a developer to specify configuration data within the source files. This can simplify the development process.

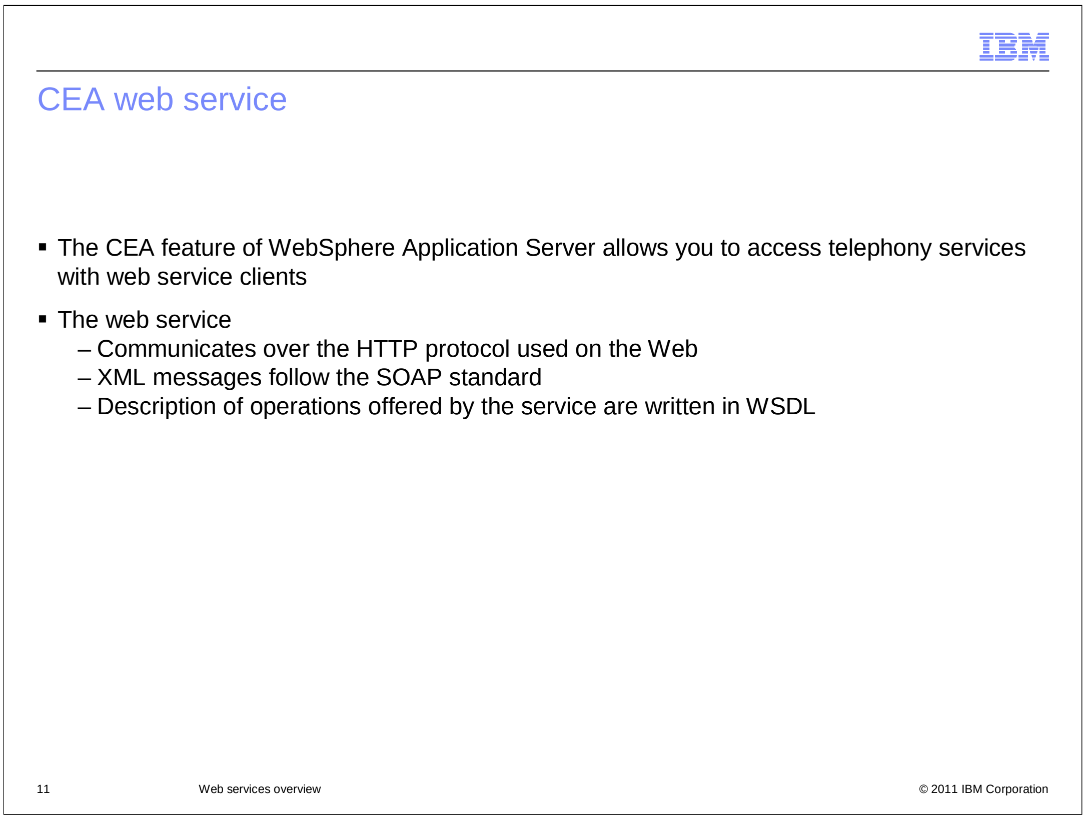
Sample annotations are shown above. The @WebService and @WebMethod annotations come from JSR-181, and are the most common on the server side. By default, you probably will not need to modify these annotations, but they contain functionality that you might find useful. If you have done any programming with web services in the past, most of the parameters will look familiar. The @WebService annotation is used to specify that the class is a Web service or that the interface defines a Web service. The @WebMethod annotation customizes a method that is exposed as a Web service operation. The associated method must be public, otherwise, you will receive an error. In addition, the parameters, return value, and exceptions of the associated method must follow the rules defined.

Section

CEA web service

10 Web services overview © 2011 IBM Corporation

The next two slides will talk about more specific concepts related to the CEA web service.

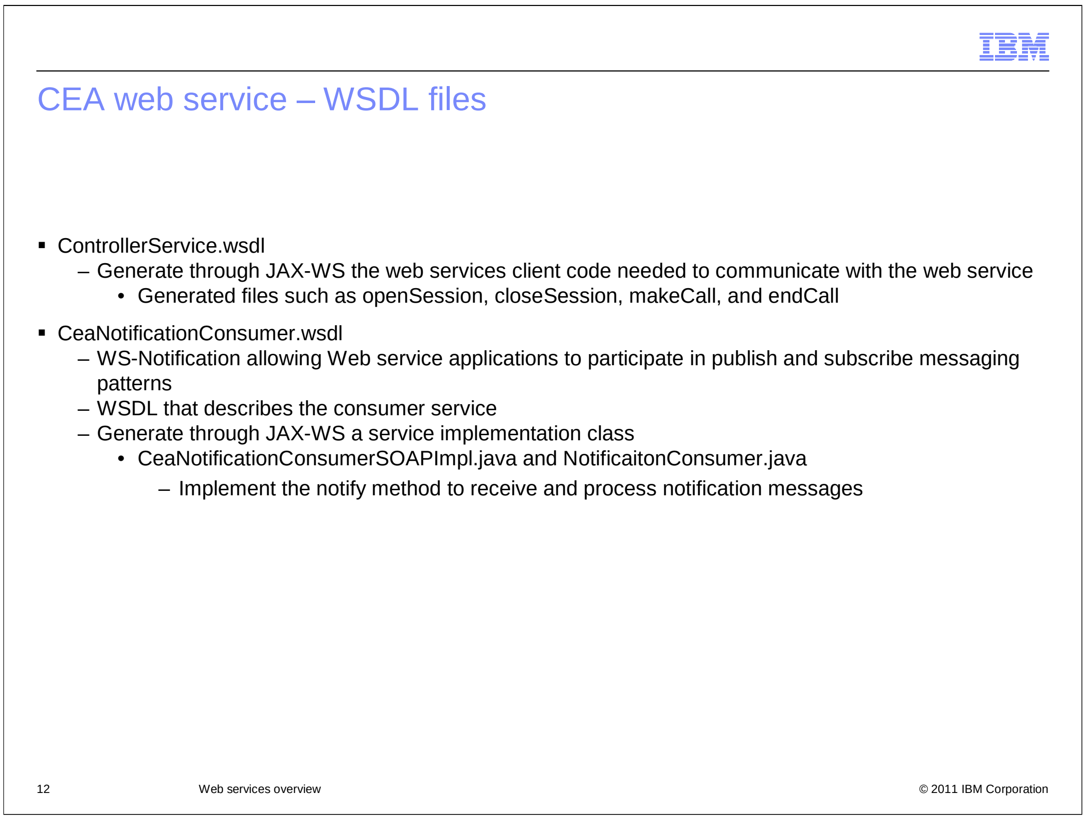


CEA web service

- The CEA feature of WebSphere Application Server allows you to access telephony services with web service clients
- The web service
  - – Communicates over the HTTP protocol used on the Web
  - – XML messages follow the SOAP standard
  - – Description of operations offered by the service are written in WSDL

11 Web services overview © 2011 IBM Corporation

The Communications Enabled Applications feature of WebSphere Application Server allows you to access telephony services with web service clients. The web service is designed to support interaction over a network and communicates over the HTTP protocol used on the Web. In CEA a Web service uses XML messages that follow the SOAP standard where there is a machine-readable description of the operations offered by the service written in WSDL. The WSDL file can be interpreted by Web service tools to generate the Web services client code needed to communicate with the Web service. As a result, an application developer need only to call the correct set of Java APIs to manage telephone calls in an application.



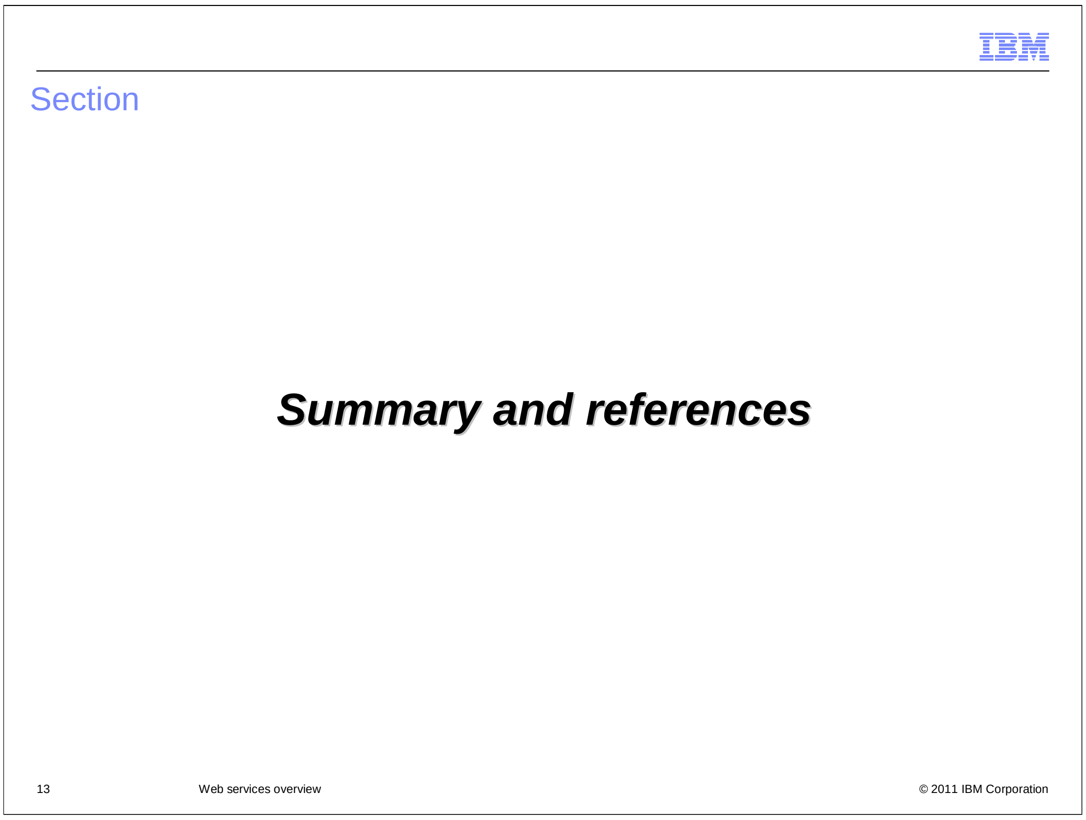
CEA web service – WSDL files

- ControllerService.wsdl
  - Generate through JAX-WS the web services client code needed to communicate with the web service
    - Generated files such as openSession, closeSession, makeCall, and endCall
- CeaNotificationConsumer.wsdl
  - WS-Notification allowing Web service applications to participate in publish and subscribe messaging patterns
  - WSDL that describes the consumer service
  - Generate through JAX-WS a service implementation class
    - CeaNotificationConsumerSOAPImpl.java and NotificaitonConsumer.java
      - Implement the notify method to receive and process notification messages

12 Web services overview © 2011 IBM Corporation

There are two main WSDL-based descriptions of the operations offered by the Communications Enabled Applications (CEA) web service. The ControllerService.wsdl file generates through JAX-WS the web services client code needed to communicate with the Web service. Generated files include openSession, closeSession, makeCall, and endCall. As a result, an application developer need only call the correct set of Java APIs to manage telephone calls in an application.

The CeaNotificationConsumer.wsdl follows WS-Notification allowing the CEA Web service to participate in publish and subscribe messaging patterns. The WSDL describes the consumer service. The CeaNotificationConsumer.wsdl file generates through JAX-WS a service implementation class, CeaNotificationConsumerSOAPImpl.java and a NotificaitonConsumer.java file. As a result, you just need to Implement the notify() method to receive and process notification messages, notifying you of the call status.

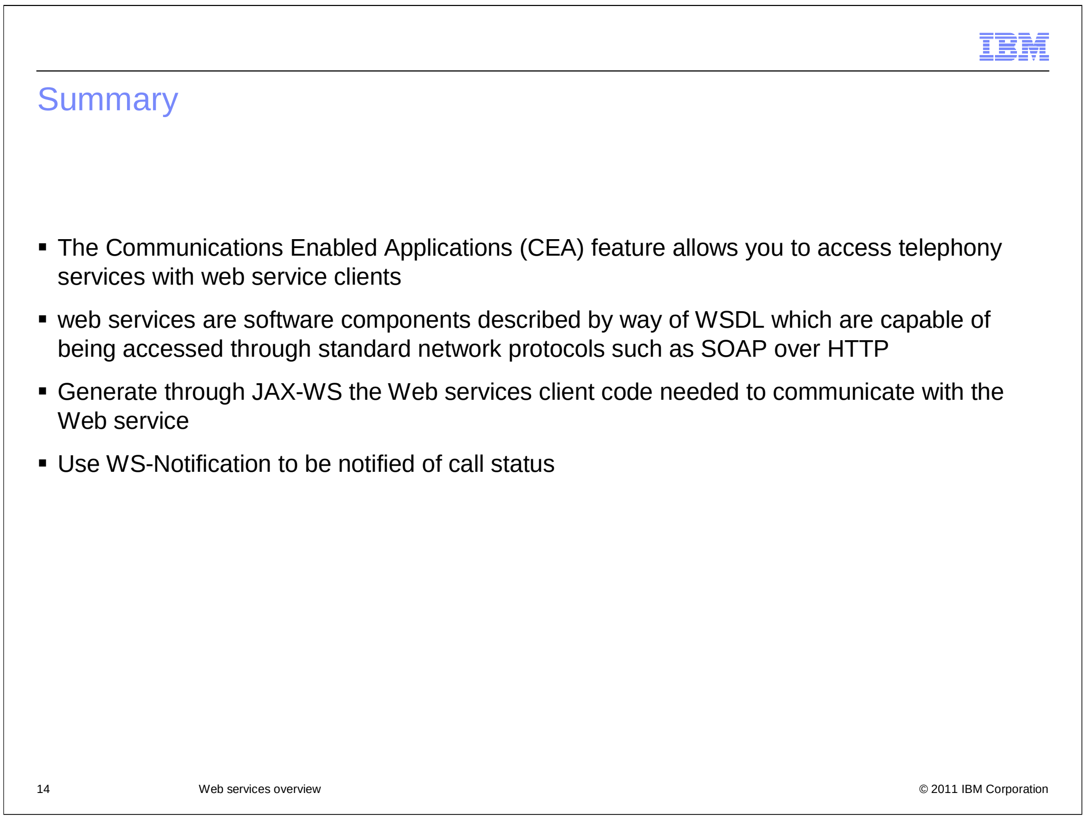


Section

Summary and references

13 Web services overview © 2011 IBM Corporation

The next slides will contain the summary and references.

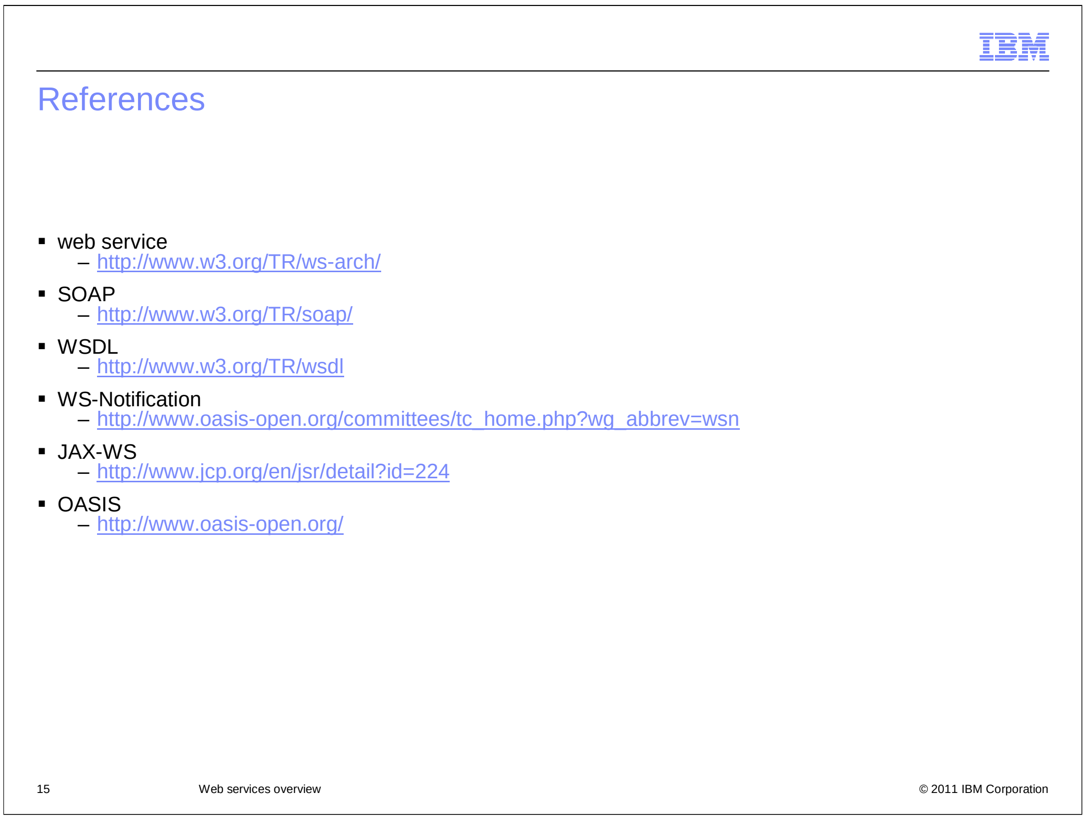


Summary

- The Communications Enabled Applications (CEA) feature allows you to access telephony services with web service clients
- web services are software components described by way of WSDL which are capable of being accessed through standard network protocols such as SOAP over HTTP
- Generate through JAX-WS the Web services client code needed to communicate with the Web service
- Use WS-Notification to be notified of call status

14 Web services overview © 2011 IBM Corporation

The Communications Enabled Applications feature allows you to access telephony services with web service clients. web services are software components described by way of WSDL which are capable of being accessed through standard network protocols such as SOAP over HTTP. WSDL files describing the Web service are included. Use the WSDL files and generate through JAX-WS the Java files needed to communicate with the Web service to open a session, make a call, end a call and close a session. The CEA feature also includes a WSDL file for monitoring call status notifications through WS-Notification.

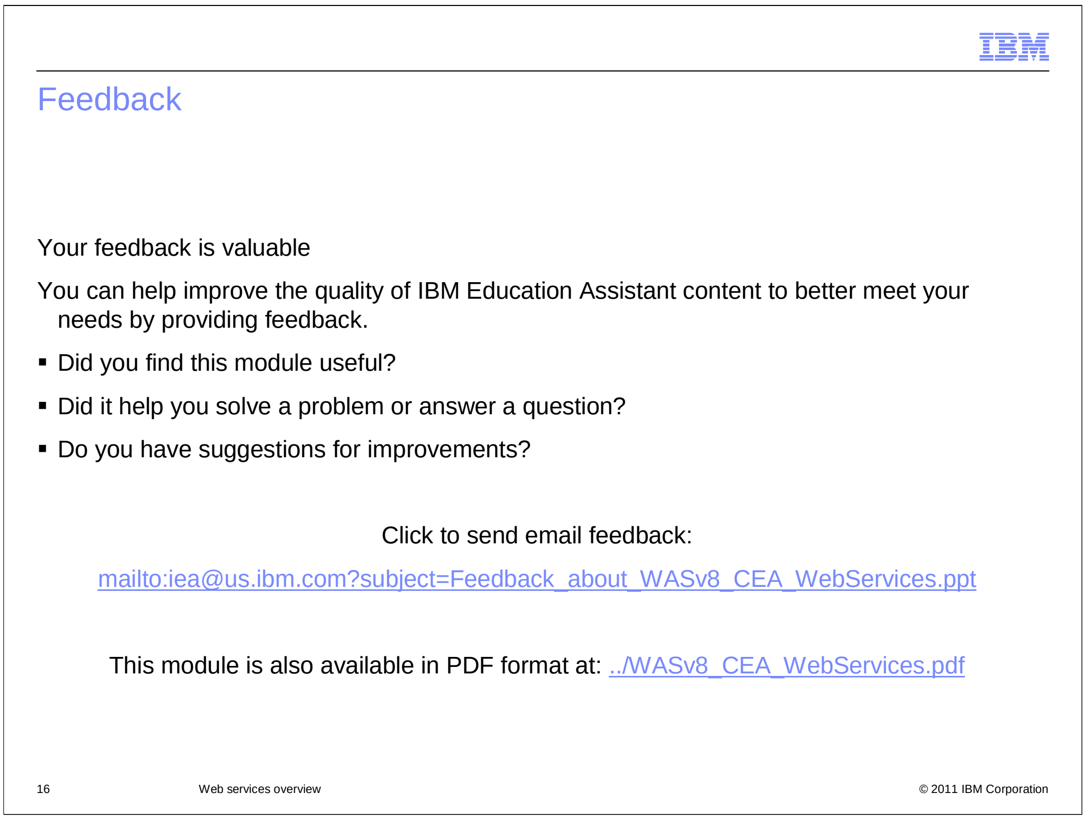


References

- web service
  - http://www.w3.org/TR/ws-arch/
- SOAP
  - http://www.w3.org/TR/soap/
- WSDL
  - http://www.w3.org/TR/wsdl
- WS-Notification
  - http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=wsn
- JAX-WS
  - http://www.jcp.org/en/jsr/detail?id=224
- OASIS
  - http://www.oasis-open.org/

15 Web services overview © 2011 IBM Corporation

Here are some useful links.



Feedback

Your feedback is valuable

You can help improve the quality of IBM Education Assistant content to better meet your needs by providing feedback.

- Did you find this module useful?
- Did it help you solve a problem or answer a question?
- Do you have suggestions for improvements?

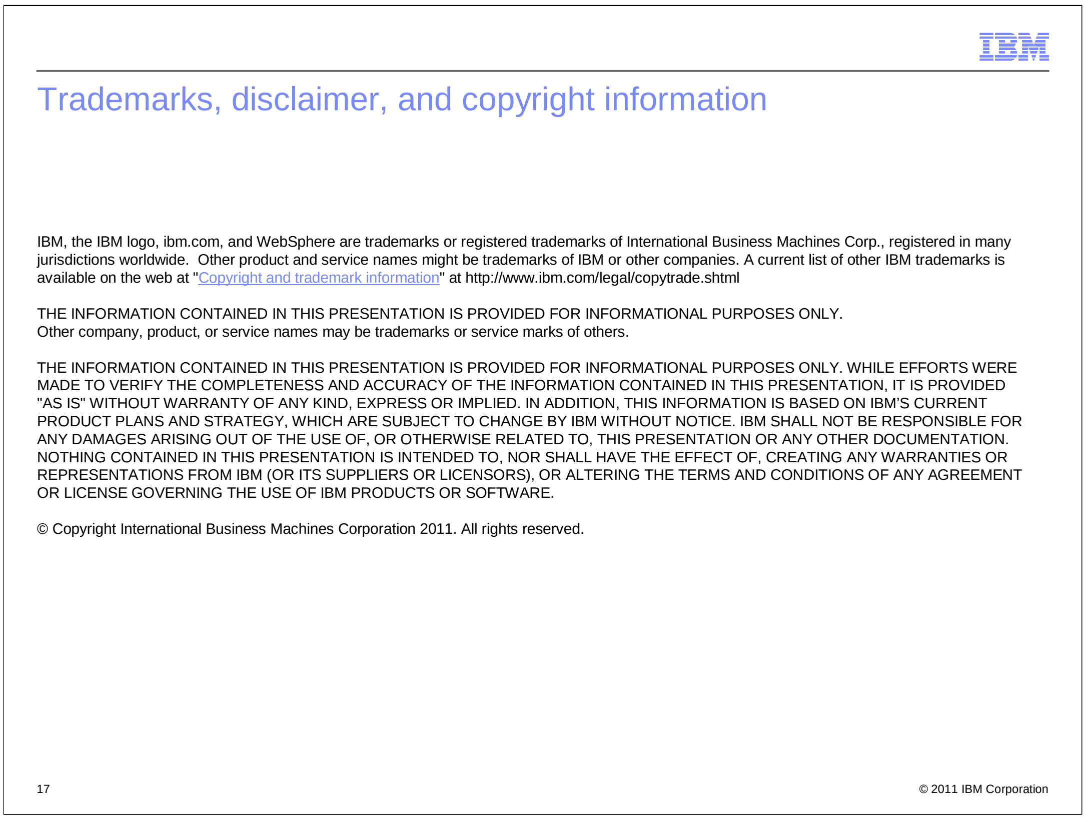
Click to send email feedback:

mailto:iea@us.ibm.com?subject=Feedback_about_WASv8_CEA_WebServices.ppt

This module is also available in PDF format at: ../WASv8_CEA_WebServices.pdf

16 Web services overview © 2011 IBM Corporation

You can help improve the quality of IBM Education Assistant content by providing feedback.



Trademarks, disclaimer, and copyright information