

# Distributed Objects and Web Services Assignment

101057

October 6, 2020

## Introduction to Web Services

The original WWW consisted of documents (i.e., Web pages) and links between documents. The initial idea of the WWW was to develop a universal information database to publish information that could be accessed in a reliable and simple way by consumers. The linking of information would allow easier and quicker browsing of information by users.

### Evolution of business usage of WWW

Organizations started to implement business-to-customer and e-commerce solutions and with the help of SSL(Secure Sockets Layer) allowed organizations to obtain confidential information e.g. credit card numbers securely. The creation of Enterprise Application Integration(EAI) allowed for the integration of incompatible and distributed systems like ERPs, CRMs etc.B2B(Business to Business) infrastructures were directed to help organizations to streamline their processes so they could carry out business transactions more efficiently with their business partners (such as resellers and suppliers). XML was used to represent data since it allows one to model data at any level of complexity since it is extensible with the addition of new tags. However, B2B strategies exhibited tight coupling thus Service Oriented Architectures were created as a method of designing, developing, deploying and managing discrete pieces of computer logic (i.e., services) within the web application.

### History of Distributed Systems

Socket interface which supported Transmission Control Protocol (TCP) or User Datagram Protocol (UDP) transport protocols for low-level messaging over Internet Protocol (IP) network were used. In the 1980s, Remote Procedure Calls (RPC) were created which uses the client/server model and extends the capabilities of traditional procedure calls across a network however it cannot provide flexibility functionality for N-tiered architectures thus Common Object Re-

quest Broker Architecture (CORBA) and Distributed Common Object Model (DCOM). Later, Java Remote Method Invocation (RMI) were developed.

## Service Oriented Architecture

To cope with the restrictions(Integration) of more traditional distributed objects architectures, in the early 2000's, the concept of Service-Oriented Architecture (SOA) was introduced facilitates the development and composition of modular services that can be easily integrated and reused to create distributed applications.

### Advantages

1. **Scalable:** work in a variety of settings, such as within an organization, between business partners and across the world.
2. **Loosely-coupled:** The source can send a message independently of the target
3. **Interoperability:** One party can be able to communicate with another party regardless of the machine they are running on.
4. **Discovery:** One party can be able to communicate with a second party selected from a set of competent candidates.
5. **Abstraction:** abstracts the underlying technology.
6. **Standards:** based on a peer-to-peer paradigm, have symmetric interaction models, mixes synchronous and asynchronous protocols, assigns public contracts to network accessible objects, and supports capability based service discovery

## Service Oriented Architecture and Web Services

There are three common actions associated with a service in SOA - discovery, request, and response. Discovery is the process of finding the service provides the functionality that is required. A request provides the input to the service. The response yields the output from the service. It follows easily that this architecture must have three primary actors: requestor, provider, and registry. Web services are modular, self-describing, self-contained applications that are accessible over the Internet

### SOA and Web Service Standards

The use of standard protocols is one of the aspects that allow SOA to deploy technically compatible services. . Currently, Web service standards are the preferred solution to develop SOA-based products.

## **Basic Web Service Standards**

XML is the standard for data representation; SOAP specifies the transport layer to send messages between consumers and providers; WSDL describes Web services; and UDDI is used to register and lookup for Web services.

## **Other Web Services Standards and Protocols: WS-\***

### **WS-Policy**

WS-Policy is a specification of a framework for defining the requirements and capabilities of a service. Once a provider has a policy defined in XML, then he must publish that information by referencing it in the description of the service.

### **WS-PolicyAttachment**

This defines the method for attaching a policy to a WSDL file so that it can be published to the UDDI and thus used in deciding on services.

## **Web Service Security**

A basic definition of security can be thought of as ‘keeping unauthorized access minimal’. Encryption plays a large role in providing integrity of messages between the requester and the provider while traversing intermediaries.

### **WS-Security Framework**

Provides a framework and vocabulary for requesters and providers to secure messaging as well as communicate information regarding security and privacy. XML Encryption specifies the process of encrypting data and messages. XML Signature provides a mechanism for messages integrity and authentication, and signer authentication. XACML is an XML representation of the Role Based Access Control standard (RBAC). XACML will likely play an important function in Web services authorization. Security Assertion Markup Language, or SAML, is an OASIS framework for conveying user authentication and attribute information through XML assertions.

### **WS-SecurityPolicy**

Policies for Web services that describe the access permissions as well as actions which a requester or provider are required to perform.

### **WS-Trust**

It offers extensions to the WS-Security elements to exchange security tokens and establishing trust relationships.

### **WS-SecureConversation**

Uses SOAP extensions to define key exchange and key derivation from security context so that a secure communication can be ensured.

### **WS-Authorization**

Assertion based authorization uses assertions about the requester to decide on the level of authorization. In a role based approach, requesters are given ‘user’ labels and these labels are associated with roles, which in turn have permissions assigned to them. Context based authorization examines the context in which a requester is acting.

### **WS-Privacy**

Privacy is in the context of data and can be associated with the requester or the provider. The requester may be concerned that the information given to a provider will be propagated to other entities. A provider may be concerned with the proliferation of information which they have sold to a requester.

## **Transaction Processing**

Web services transaction management enables participating services to have a greater degree of confidence in that the actions among them will progress successfully, and that in the worst case, such transactions can be cancelled or compensated as necessary.

### **WS-Transaction**

WS-Coordination (WS-C) defines a coordination context, which represents an instance of coordinated effort, allowing participant services to share a common view. WS-AtomicTransaction (WS-AT) targets existing transactional systems with short interactions and full ACID properties. WS-BusinessActivity, on the other hand, is intended for applications involved in business processes of long duration, whose relaxed properties increase concurrency and suit a wider range of applications.

### **WS-Composite Application Framework**

Container-Managed Transactions (CMT) for some time, a way to procure increased Web service reliability could be through their deployment in managed environments, in which the hosting application server becomes responsible for support activities such as event logging and system recovery.

## **Messaging**

### **WS-ReliableMessaging**

WS-Reliability and WS-ReliableMessaging have rules that dictate how and when services must respond to other services concerning the receipt of a message and its validity.

### **WS-Eventing**

Web Services Eventing (WS-Eventing) is a specification that defines a list of operations that should be in a Web service interface to allow for asynchronous messaging.

### **WS-Notification**

Web Service Notification (WS-Notification) is a family of specifications that provide several capabilities.

1. Standard message exchanges for clients
2. Standard message exchanges for a notification broker service provider
3. Required operations for services that wish to participate in notifications
4. An XML model that describes topics.

## **Developing Web Services**

### **Create a UML Class Diagram**

UML provides a succinct representation of modeling classes.

### **Generate Java Code**

Using a UML tool such as Poseidon, the UML Class Diagram can easily be converted into a Java class skeleton.

### **Adding in Web Services Annotations**

Java includes annotations so that the compiler will know that the program code is a Web Service.

### **Generate WSDL**

The annotations from the previous step indicate to the Annotation Processing Tool or the Java compiler that a WSDL is to be generated at compile-time.

## **Implement Methods**

Getters and setters are used during invocation by the SOAP engine to serialize and deserialize the data that is in the SOAP messages into Java objects and back to SOAP.

## **Deploy Service**

Deploying a service is accomplished using a Web application server and a SOAP engine, like Tomcat and Axis2 respectively. If using Axis2, deploying a service is as simple as dropping the .aar files, which are .jar files with a different extension, into the -INF directory.

## **Test Service**

A java program can be used to test the web service.

## **Publish Service**

Publishing a service requires the use of UDDI registries. After deployment and testing, the service is open to the world and ready to accept request.