Web Services

WEEK 07 LECTURE 2

MURTAZA MUNAWAR FAZAL

Distributed Object Computing

- Building scalable solutions require applying the appropriate level of computing power to different parts of the application.
- Distributed object computing extends an object-oriented programming system by allowing objects to coexist across a heterogeneous network and interact as if they resided on the same machine.
- In the early 1990's many companies including Microsoft, Sun Microsystems, the Object Management Group and IBM recognized the need of distributed computing and developed their own technologies to facilitate communication among distributed components.
  - Microsoft DCOM (Distributed Component Object Model)
  - Sun Microsystems' RMI (Remote Method Invocation)
- Distributed objet systems have several problems.

OOP and Network Latency

- The style of object-oriented programming directly conflicts with the performance of distributed system.
  - Object oriented classes are fine-grained, with property and field accessors.
  - Object interactions lead to many round trips across the network. Network latency causes severe performance degradation. You want to minimize the number of interactions between client and server.

What are Web Services?

- A Web service is a message delivered over some transport mechanism.
  - XML is the most common message format.
  - HTTP has been the most common transport.
  - SOAP has become an industry standard protocol running on top of HTTP.
- The producer of the Web Service maps an object service, program or database to the XML message.
- The consumer of the Web Service maps the XML message to an object, service, program, or database
- A Web service is executed via the application that consumes it.

Why are Web Services Needed

Web services provide a neutral means to integrate programs built on different development platforms.

- Web services are superior to distributed object technologies.

Web services provide a means to evolve application as customer requirement change.

- Web services enable applications built with a Service Oriented Architecture (SOA).

Object State and Scalability

Scalability is the performance of the system under increase load.

- Systems are scalable if performance can be increased by just adding hardware.

Object oriented systems tend to have many fine-grained objects.

- Keeping these objects alive consumes resources that prevent servicing more request.

Tracking distributed object lifetimes across the network so that objects can release their proxies, increase network traffic, and degrades performance.

Persistent distributed objects also interfere with load balancing to distribute processing evenly among computers.

- Stateless objects can be created or destroyed between method calls. State is stored in the non-distributed part of the application.
- For example, an object that holds on to a database connection interferes with scalability.

Interoperability

Each distributed object system used its own transport mechanism.

- The various distributed object systems of the different vendors did not interoperate.
- Although in principle CORBA was a vendor-neutral specification, in practice customers were locked into their particular choice of vendor implementation.

The various distributed object systems used different wire protocols.

- This is similar to having different procedure calling conventions between methods.

The various distributed object systems require keeping non-standard Internet ports open for remote object invocation.

- These protocols are not transparent to the usual settings for firewall.

Industry Standard Messaging

HTTP is a ubiquitous transport standard.

- Port 80 is almost always open on a firewall

Since HTTP is a text-based protocol, XML was a natural choice for implementing an object-neutral messaging protocol.

By exchanging messages instead of object method calls, application built with different object models, and different implementation technologies can interact.

Messaging technologies are most robust and scale better in distributed scenarios.

This looser coupling, however, makes it harder for type information and transaction context to be transmitted.

SOAP and WSDL have merged as the fundamental specifications for Web services.

Additional specifications such as WS-Security and WS-Addressing are designed to solves some additional issues associated with Web services.

Future standards such as WS-Eventing (for publish and subscribe) will impact application development.

Benefits of Web Service Integration

Let’s look at some business advantages of Web services.

- Web services offer new business opportunities by marking it easy to connect with partners (Business to Business Integration)
- Reduce application development and maintenance costs (Save time and money).
- Create new sources of revenue by making the present useful functionality into Web services.
- Web services provide a solution to systems interoperability problems.
- Implement cross-platform, program-to-gram communications (application integration).
- Deliver significantly more personal, integrated experiences to users using Web services.
- Web services connect information, applications, people, systems and devices.

SOAP

SOAP (Simple Object Access Protocol) is a lightweight protocol for the exchange of information in a decentralized, distributed environment.

- SOAP is a protocol that defines how to send messages in a platform-independent manner using XML.
- SOAP, uses XML as a wire protocol to describe how the data and its associated type definitions are transmitted.
- SOAP was developed by Microsoft, IBM and others, and then handed over to the W3C for further development.

WSDL

The WSDL (Web Service Description Language) is an XML-based grammar for describing the syntax of Web Services, their functions, parameters and return values.

- WSDL defines the methods and the data associated with Web Service.
- Since WSDL is XML-based, it is both human and machine readable. However WSDL is designed to be used by machines for automated implementation of interface contracts.

WSDL Section	Use
Types	Defines types
Messages	Abstract message signatures
Operations	Abstract method definitions
Port Type	Abstract interface based on operations
Binding	Interface and method implementations
Port	Associates binding with a specific address
Service	Collection of ports

WSDL Example

```
<message name="getTermRequest">
  <part name="term" type="xs:string"/>
</message>

<message name="getTermResponse">
  <part name="value" type="xs:string"/>
</message>

<portType name="glossaryTerms">
  <operation name="getTerm">
    <input message="getTermRequest"/>
    <output message="getTermResponse"/>
  </operation>
</portType>
```

Service Oriented Architecture (SOA)

Objects do not always easily reflect real-world processes and relationships

Distributed object systems have trouble evolving to reflect the dynamic nature of customer requirements and businesses.

Building application out of services allows for a more flexible architecture in building complex applications, specially if the services are owned by different divisions, or event different business.

Service orientation allows you to build applications out of new services, services cleaved out of old applications, or application wrapped as services.

SOA is important when application have to cross trust boundaries.

Object technology can be used in implementing services

Web Services are the natural implementation technology for SOA.

Services are Independent

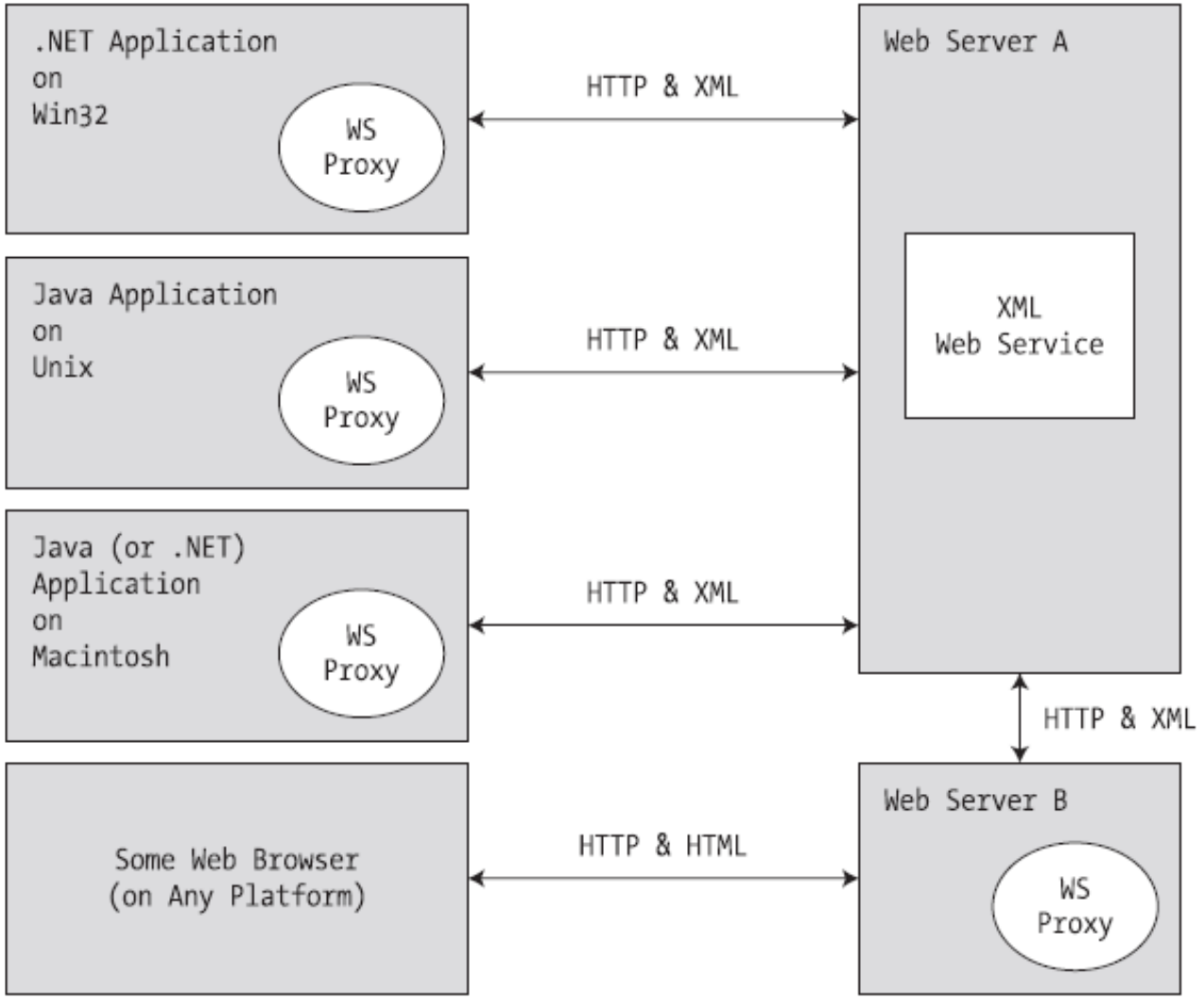
A service is deployed independently of other services.

A Service is separate from the User Interface.

ASP.Net Web Services

ASP.Net Web services are built on top of ASP.Net; therefore you can benefit from the features of ASP.Net to build Web Services.

ASP.Net Web services facilitate Web Services' accessing the many features of the .NET framework, such as authentication, caching, tracing and state management.


.NET Application
on
Win32
WS
Proxy
HTTP & XML
Web Server A
Java Application
on
Unix
WS
Proxy
HTTP & XML
XML
Web Service
Java (or .NET)
Application
on
Macintosh
WS
Proxy
HTTP & XML
HTTP & XML
Some Web Browser
(on Any Platform)
HTTP & HTML
Web Server B
WS
Proxy