Chapter 14: Web Services

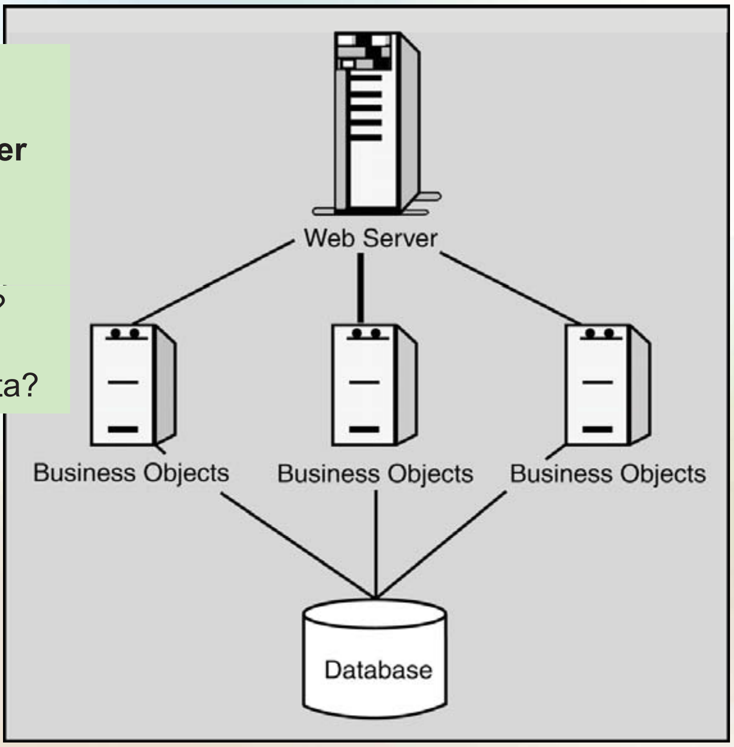
Chapter 14 Objectives

- What a Remote Procedure Call (RPC) is, and what RPC protocols exist currently
- Why Web services can provide more flexibility than previous RPC protocols
- How XML-RPC works
- Why most Web services implementations should use HTTP as a transport protocol, and how HTTP works under the hood
- How the specifications that surround Web services fit together

What Is an RPC?

A program on a computer calls procedure on another machine.

You have to know:
Where does it reside?
Code parameters?
Handling of return data?



RPC Protocols

- **Distributed Component Object Model (DCOM)**
  - **Microsoft**
- **Common Object Request Broker Architecture (CORBA) & Internet Inter-ORB Protocol (IIOP)**
  - **Object Management Group (OMG)**
- **Remote Method Invocation**
  - **Java**

The New RPC Protocol – Web Services

- A software system identified by a **URI**, whose **public interfaces and bindings** are defined and described using **XML**. Its definition can be discovered by other software systems. These systems may then interact with the Web service in a manner prescribed by its definition, using **XML based messages** conveyed by **Internet protocols** (W3C Web Services Architecture Requirements Working Group Notes, http://www.w3.org/TR/2004/NOTE-wsa-reqs-20040211/)
- A software system designed to support **interoperable machine-to-machine** interaction over a **network** (W3C Web Services Architecture Working Group Notes, http://www.w3.org/TR/2004/NOTE-ws-arch-20040211/)
- A service that accepts a request and returns data or carries out processing task over the Internet (our textbook)

Web Services

- **Three approaches**
  - **XML-RPC**
  - **RESTful**
  - **SOAP-Based**

XML-RPC

- XML-RPC
  - Simple
  - Specifies
    - Procedure to call
    - Parameters to pass
  - Calls remote procedure
    - XML requests
    - XML responses

Example: Internet Topic Exchange

- The Target API
  - Three methods:

```
struct topicExchange.getChannels()
// return a list of existing channels

struct topicExchange.ping(string topicName, struct details)
// add a new entry to a particular topic

struct topicExchange.getChannelInfo(string topicName)
// retrieve information on a specific channel
```

XML-RPC Request

- **Simple request**

```
<methodCall>
  <methodName>topicExchange.getChannels</methodName>
</methodCall>
```

- **Parameter passing**

```
<methodCall>
 <methodName>topicExchange.getChannelInfo</methodName>
 <params>
  <param>
   <value><string>books</string></value>
  </param>
 </params>
</methodCall>
```

XML-RPC Request

- **Using *struct*, a group of named values passed**

```
<methodCall>
 <methodName>topicExchange.ping</methodName>
 <params>
  <param>
   <value><string>books</string></value>
  </param>
  <param>
   <value>
    <struct>
     <member><name>…</name><value>…</value>
     </member>
    </struct>
   </value>
  </param>
 </params>
</methodCall>
```

The Network Transport

- HTTP
- Why HTTP for Web Services?
  - Widely Implemented
  - Request/Response
  - Firewall-Ready
  - Security (SSL)
- Using HTTP for XML-RPC
  - HTTP method: POST
  - Message body: XML-RPC request

Taking a REST

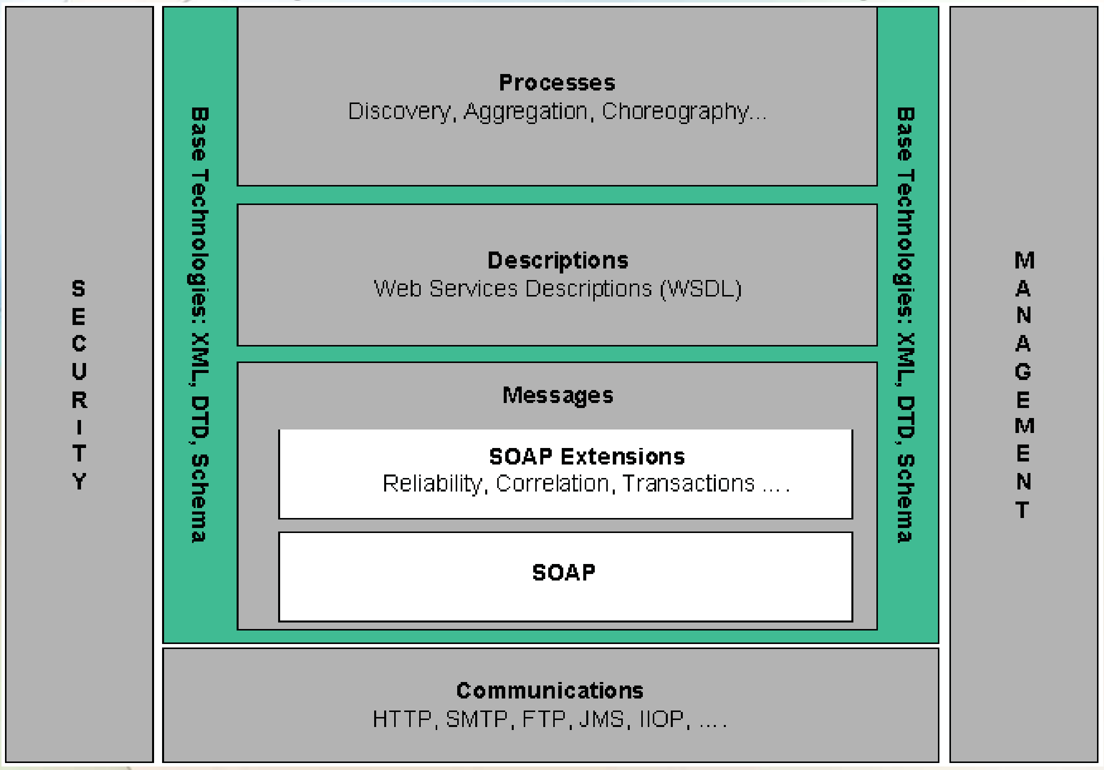
REpresentational State Transfer

- Introduced by Roy Fielding
- URI + XML + HTTP
- Not really a specification
- Is an architecture style
- Many who think they are using Web Services are really using REST

Using the REST Interface to Access eXist

- HTTP GET method
  - View /db collection: http://localhost:8080/exist/rest/db/
  - View /db/blog collection: http://localhost:8080/exist/rest/db/blog
  - View /db/blog/blogItem1.xml: http://localhost:8080/exist/rest/db/blog/blogItem1.xml
  - Execute XQuery "//a": http://localhost:8080/exist/rest/db/blog/?_query=//a
  - Transform the result of "//a" using /db/xslt/rest-query-results.xsl: http://localhost:8080/exist/rest/db/blog/?_query=//a&_xsl=/db/xslt/rest-query-results.xsl
- Other HTTP methods (requests)
  - POST, PUT, DELETE

Web Service Architecture Stack


SECURITY
Base Technologies: XML, DTD, Schema
Processes
Discovery, Aggregation, Choreography...
Descriptions
Web Services Descriptions (WSDL)
Messages
SOAP Extensions
Reliability, Correlation, Transactions ....
SOAP
Base Technologies: XML, DTD, Schema
MANAGEMENT
Communications
HTTP, SMTP, FTP, JMS, IIOP, ...

(W3C Web Services Architecture http://www.w3.org/TR/2004/NOTE-ws-arch-20040211/)

SOAP

Simple Object Access Protocol
or just
SOAP

- Provides a simple and lightweight mechanism for exchanging structured and typed information between peers in a decentralized, distributed environment using XML
- Three parts
  - Envelope
  - Encoding rules
  - RPC representation

SOAP

- SOAP envelope construct
  - Defines overall framework for expressing
    - what is in a message
    - who should deal with it
    - whether it is optional or mandatory
- SOAP encoding rules
  - Defines serialization mechanism that can be used to exchange instances of application-defined datatypes
- The SOAP RPC representation
  - Defines convention that can be used to represent remote procedure calls and responses

SOAP

SOAP Envelope

```
<?xml version="1.0" encoding="UTF-8"?>
<SOAP:Envelope xmlns:SOAP="http://www.w3.org/2003/05/soap-envelope">
 <SOAP:Header></SOAP:Header>
 <SOAP:Body>
  <totals xmlns="http://www.wiley.com/SOAP/accounting">
    <dept id="2332">
      <gross>433229.03</gross>
      <net>23272.39</net>
    </dept>
    <dept id="4001">
      <gross>993882.98</gross>
      <net>388209.27</net>
    </dept>
  </totals>
  </SOAP:Body>
</SOAP:Envelope>
```

WSDL

Web Services Description Language

- **Provides a model and an XML format for describing Web services**
- **Describes a Web service in two fundamental stages**
  - **Abstract**
  - **Concrete**

(http://www.w3.org/TR/2007/REC-wsdl20-20070626/)

Toolkits are available from:

http://www-128.ibm.com/developerworks/views/webservices/downloads.jsp

WSDL

- At an abstract level
  - WSDL 2.0 describes a Web service in terms of messages it sends and receives
  - Messages are described independent of a specific wire format using a type system, typically XML Schema
- At a concrete level
  - *Binding* specifies transport and wire format details for one or more interfaces
  - *Endpoint* associates a network address with a binding
  - *Service* groups together endpoints that implement a common interface.

UDDI

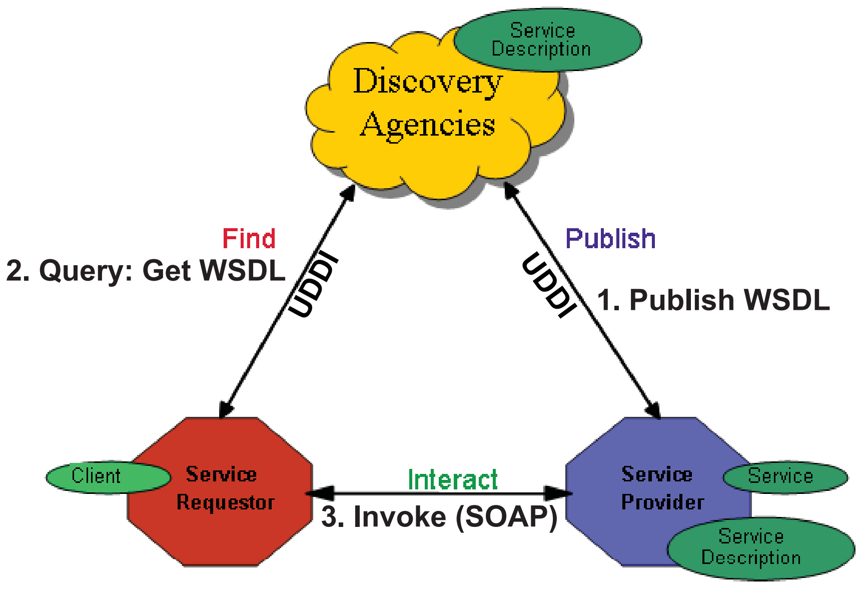
Universal Discovery, Description, and Integration

- Defines a standard method for publishing and discovering the network-based software components of a service-oriented architecture (SOA)
- OASIS (Organization for the Advancement of Structured Information Standards) approved standard
- Global UDDI Registry

  http://www.uddi.xml.org

Working Together

Service Oriented Architecture



Surrounding Specifications

- Interoperability
  - Web Services Interoperability Organization (WS-I) (www.ws-i.org)
  - Define “profiles” for web services and provide testing tools
- Coordination
  - WS-Choreography (www.w3.org/2002/ws/chor)
- Security
  - XML Encryption, XML Signature
  - Identity recognition
  - Reliable messaging
  - Overall security policy