e-business

Web Service Implmentation on iSeries

SP02

ITSO iSeries Technical Forum

Yessong Johng

Redbooks
International Technical Support Organization

Abstract

The objectives of this session includes:

- Understanding the concept and benefits of Web Services
- Understanding the standards of Web Services
- Understanding a real-life example of transforming existing iSeries application into a Web Service application

Note: We use "iSeries" to refer to both AS/400e and the IBM iSeries servers.

What Web Services are

Self-contained, self-describing, networked, modular applications

Well-defined, service-based interface that describes a collection of networked accessible operations

Business functions that are available over the Web

Fairly coarse grained component models that give applications developers building blocks to build solutions

Enterprise Application Integrator

Web Service - Next Transformer

TCP/IP

- Transformed how the world gets connected

Web and HTTP

- Transformed how the connection exchanges the protocol

Browser and HTML

- Transformed the user interface

XML

- Transformed how to exchange the data

Web Service

- *Transforms how the applications find and talk each other*

Web Services for B2B

The Web revolutionized business-to-customer: B2C

- Massive extension of business opportunity
- New business models
- Dramatic reduction in infrastructure costs and complexity
- The key was a universal server-to-client model based on standards and industry support

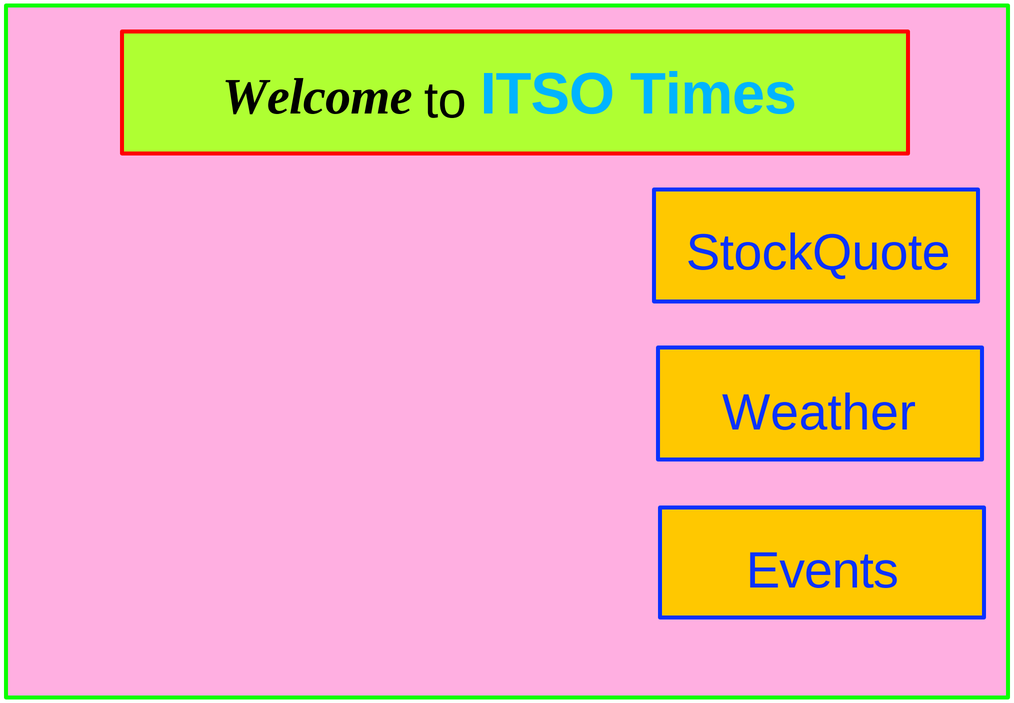
Web Services promises to do the same thing for business-to-business: B2B

- Massive extension of business opportunity
- New business models
- Dramatic reduction in infrastructure costs and complexity
- The key will be a universal program-to-program communication model based on standards and industry support

Scenario 1

You are developing a Web application...

- And you want to add a number of "services" to the user
- But you don't want to develop/deploy/maintain the background applications providing the "services"...



Welcome to ITSO Times

StockQuote

Weather

Events

Scenario 1

What if there's people....

- Who will provide the services you need
- At norminal fee or even free!

StockQuote.com

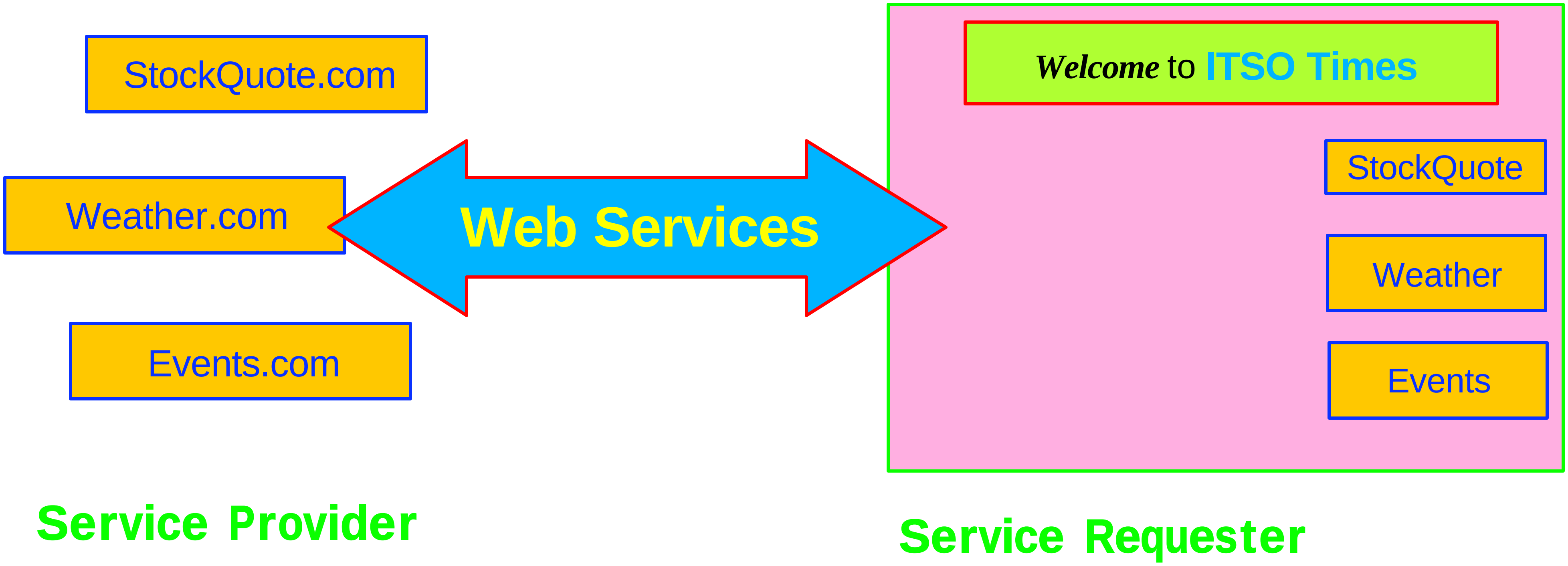
Weather.com

Events.com

Scenario 1

Web Services enable this through

- Industry standards which make dynamic connection independent of
  - Platform, language, data structure, APs, and on on



Standards for Scenario 1

Standard for message exchange -> SOAP

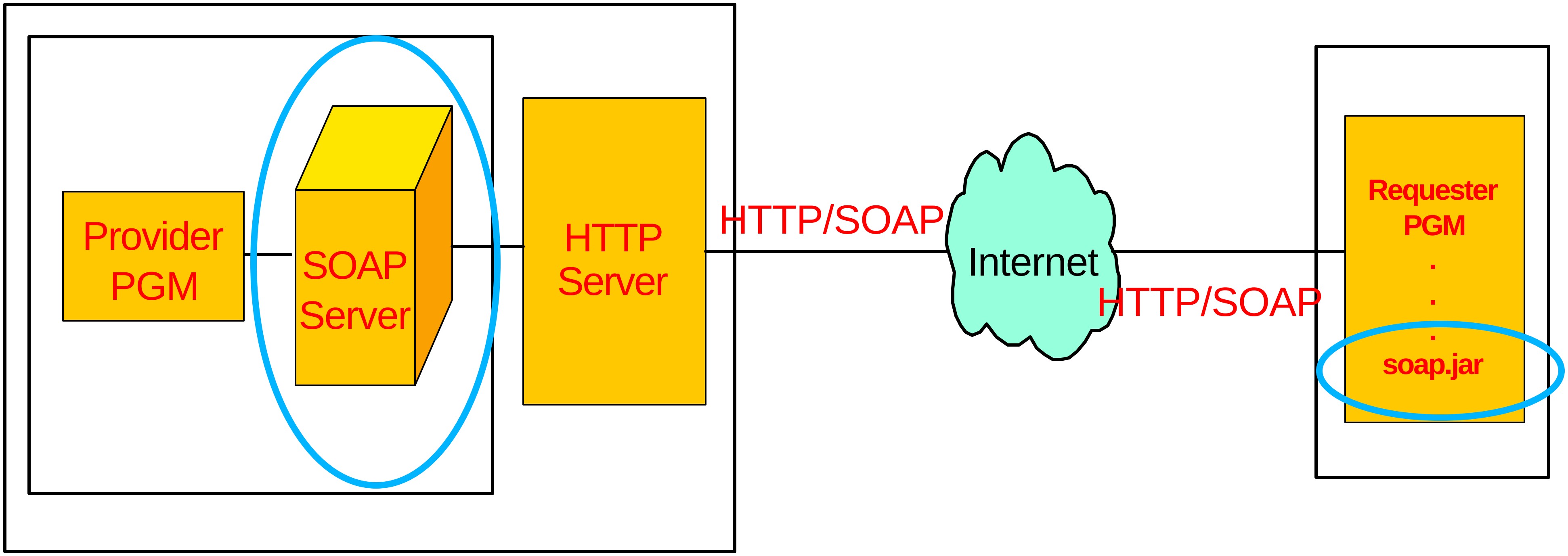
- Simple Object Access Protocol
- Allows applications to interact regardless of platform
- In commercial use today
- Simple enveloping mechanism independent of transport layer
  - alternative to using CORBA, RMI, DCOM, etc.
  - successor to XML RPC
- XML based protocol used in decentralized, distributed environments
- Currently defined to be used over HTTP (and HTTPS)
- Being standardized by the W3C under the name XML protocol

Architecture: Standards - SOAP

SOAP: 2 types of exchanges:

- Remote procedure calls (SOAP-RPC)
  - Marchalling/unmarshalling datatypes to/from XML
  - Packaging them up into a SOAP envelope
  - Send SOAP envelope across HTTP as a POST to the Application Server
  - Usually requires the response from the provider
- Document-oriented messaging
  - Allows for the transmission of an arbitrary XML document within the body of the SOAP envelope
  - Doesn't require the response from the provider

Technology View for Scenario 1



Service Provider

... StockQuote.com, for example

Service Requester

... ITSO Times, for example

Standards for Scenario 1

Standard for message exchange -> SOAP

- **S**imple **O**bject **A**ccess **P**rotocol
- Allows applications to interact regardless of platform
- RPC, Remote Procedure Call, for Web Services

Standard for service description -> **WSDL**

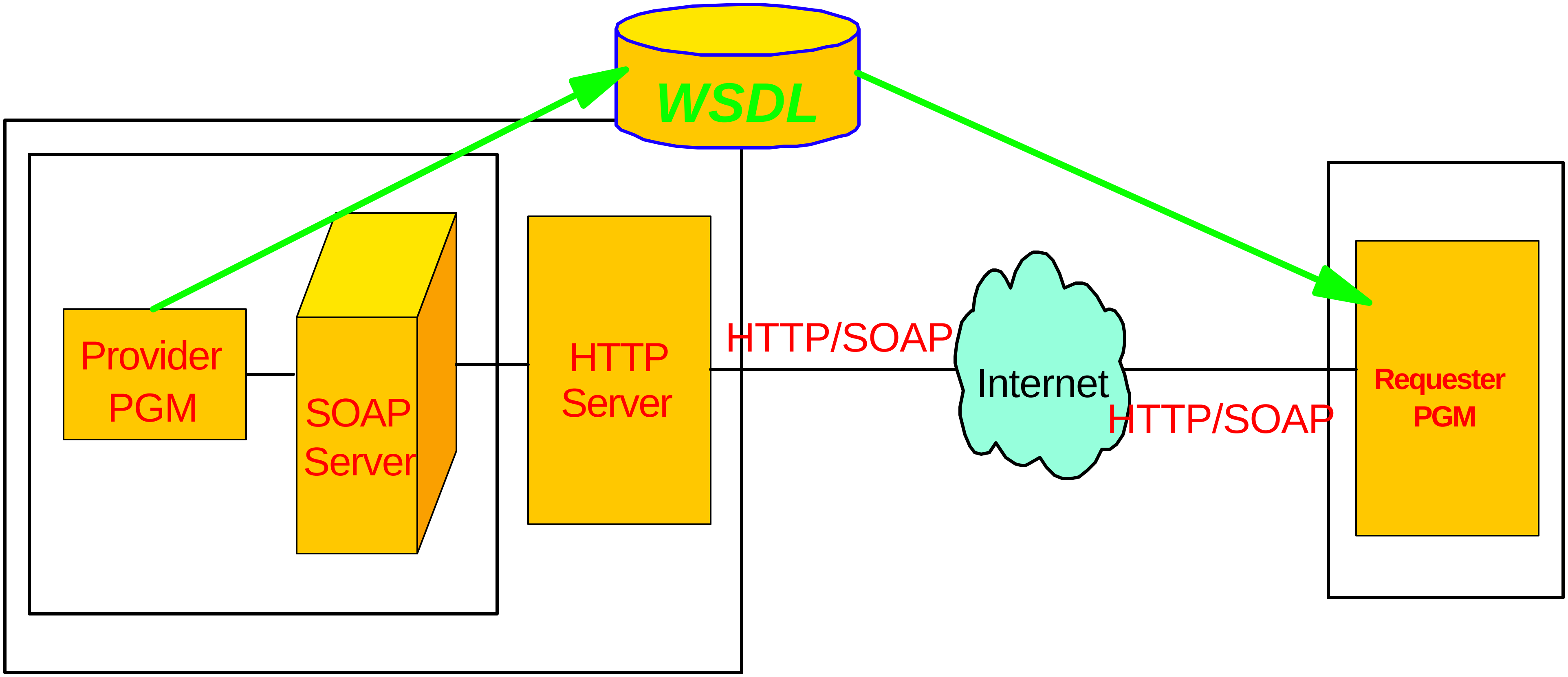
- **W**eb **S**ervices **D**efinition **L**anguage
- Defines the Web services interface
- Blueprint for an emerging architecture

Architecture: Standards - WSDL

WSDL: Web Services Description Language

- Specifies the characteristics of a Web Service
  - Name and addressing information
  - Protocol and encoding style (parameters, data types)
- Exist as two XML files
  - Service interface: abstract interface and protocol binding
    - Method that is called
    - Parameters that are passed
    - Encoding that is used
  - Service implementation: service access
    - Where the Web Service is installed
    - How it is accessed
- Tools available to generate WSDL from Java, and to generate Java from WSDL

Technology View for Scenario 1



Service Provider

... StockQuote.com, for example

Service Requester

... ITSO Times, for example

Technology View for scenario 1

Pop Quiz "One"!

Fill in the blank.

"SOAP and WSDL are all ( ).''

Technology View for scenario 1

Pop Quiz "One"!

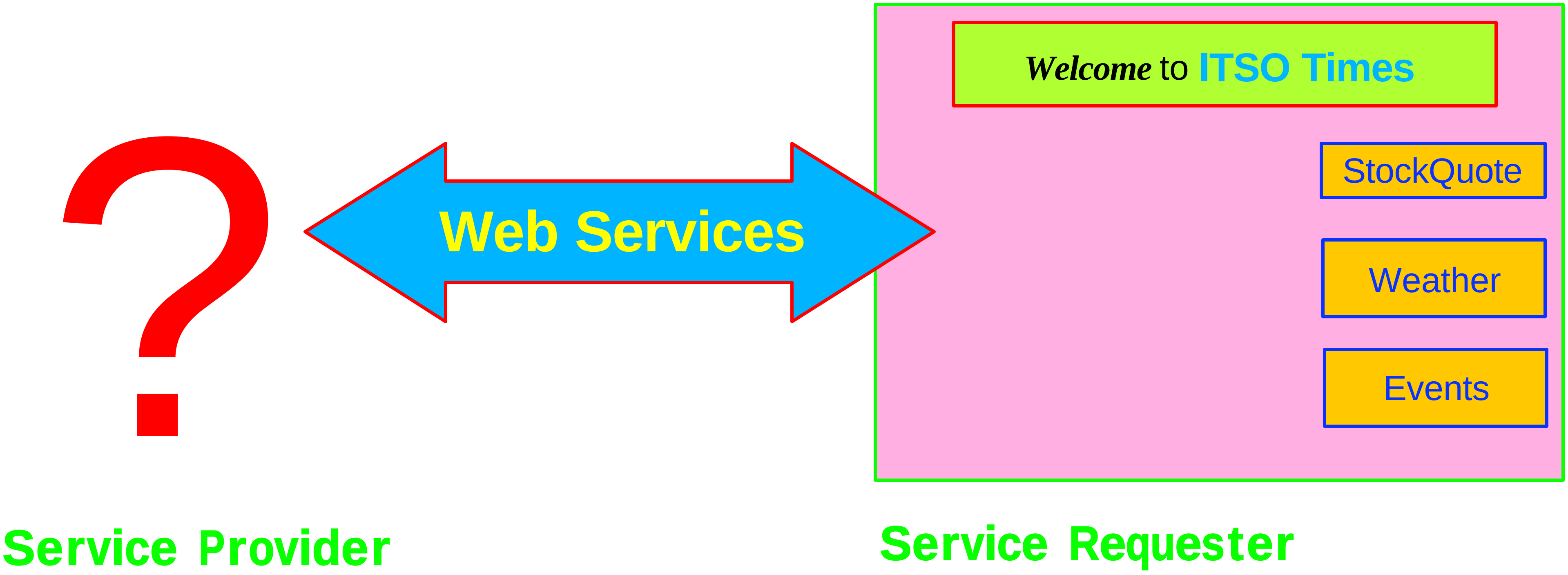
Fill in the blank.

"SOAP and WSDL are all (**XML**)."

Scenario 2

What if we don't know

- What services are available out there?
- Who the service providers are?
- Where they live?



Standard for Scenario 2

What if there's a YellowPage of all the Web Services available where you can find...

- Web Services provider by the type of the service
- Who they are and where to contact
- What you need to know to write a Web Service client code

UDDI **(Universal Description Discovery and Integration) is the answer!**

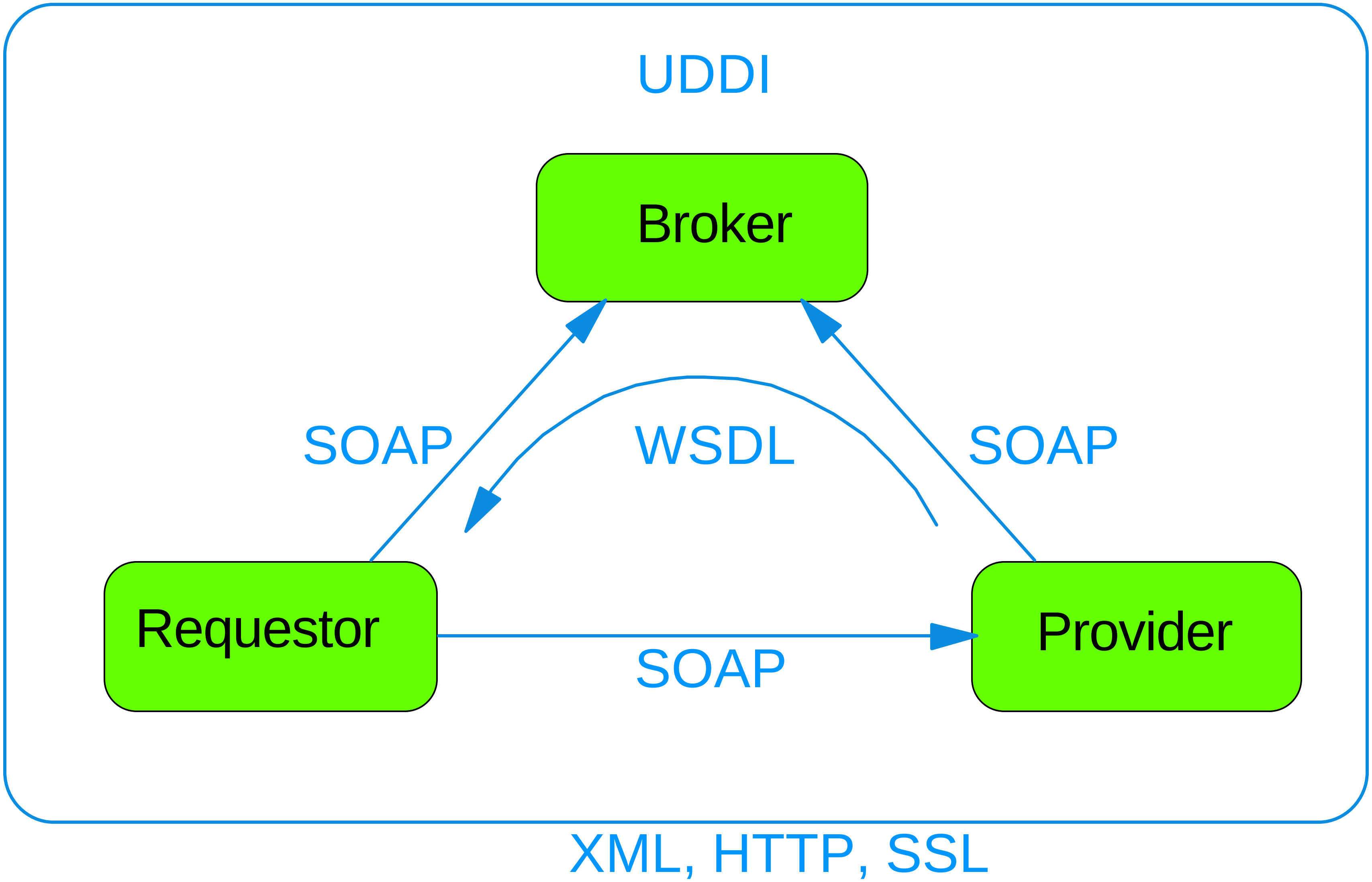
- Search engine for reusable software functions
- Web Service provider
  - Publishes their services in UDDI
- Web Service requester
  - Finds the services they need in UDDI

Architecture: Standards - UDDI

UDDI: Universal Description, Discovery and Integration

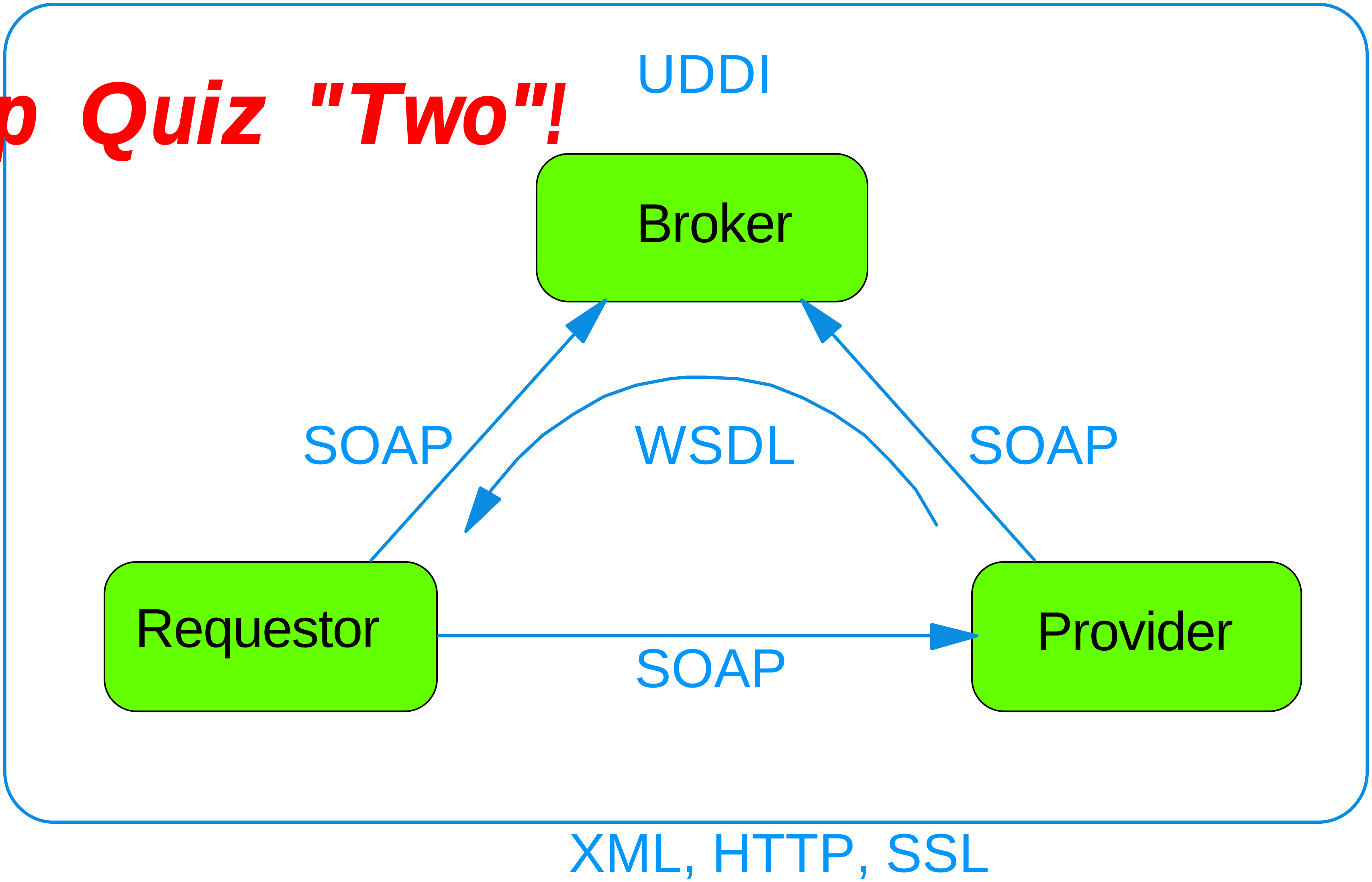
- Open industry initiative enabling businesses to discover each other
- Defines an architecture for a global registry for holding information about Web services
- Free, public, interconnected UDDI servers are deployed today
- Not playing a key role in business models yet
- A private UDDI implementation is available today from IBM alphaWorks
- Initiative defines service broker architecture
  - http://www.uddi.org/
- IBM UDDI test area
  - http://www.ibm.com/services/uddi/testregistry/
- IBM UDDI registry
  - http://www.ibm.com/services/uddi

Primary Web Service Technologies

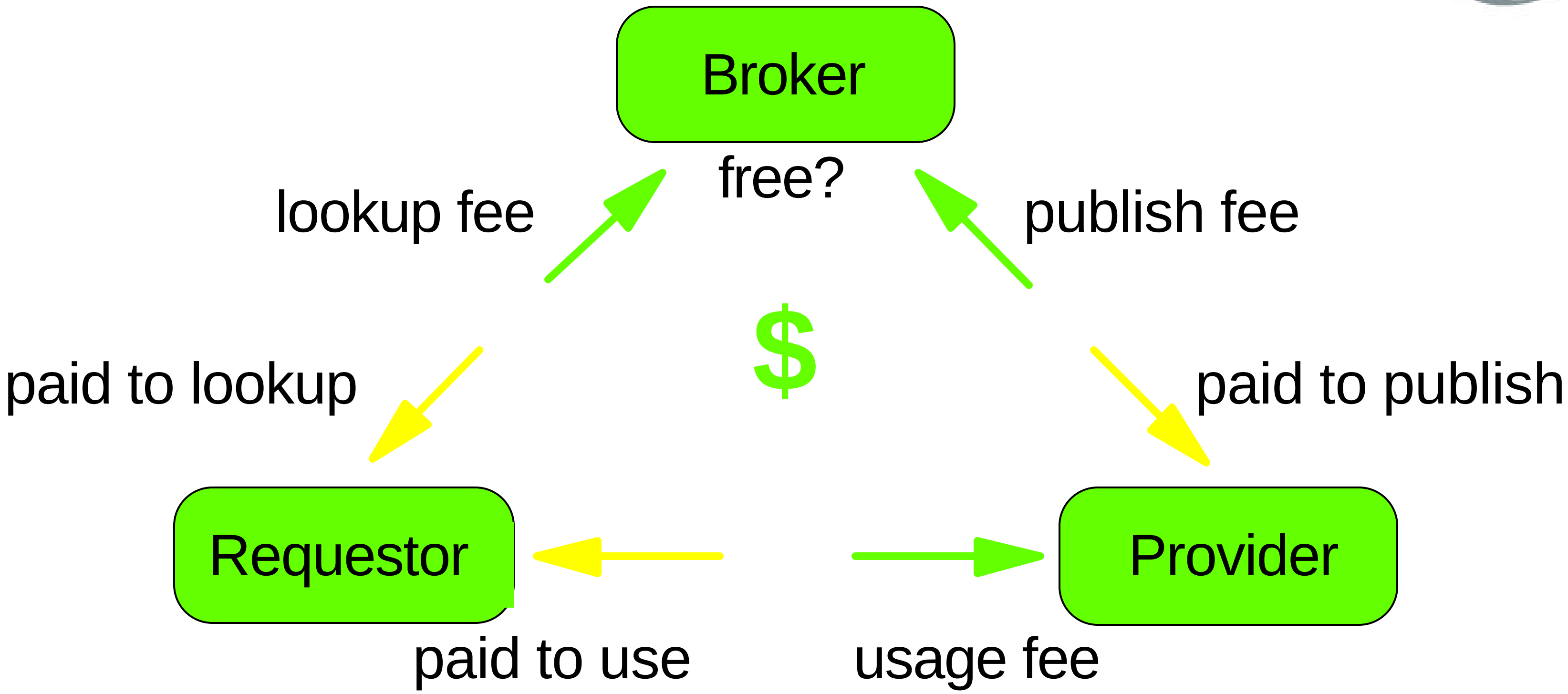


Fee Flow

Pop Quiz "Two"!



Who pays who ?



Broker

free?

lookup fee

publish fee

$

paid to lookup

paid to publish

Requestor

Provider

paid to use

usage fee

Service requester pays. Yes

Service provider pays. Questionable dot-com model (except publish?)

Ads are hard -- programs don't impulse buy

Business and Technical Benefits

Revenue Benefits

- Improved relationships with customers/partners
- Work force productivity
- Innovation and reduced cycle time
- Share processes without sharing technology

Cost Reduction

- Deliver new business solutions faster
- Better information flow and knowledge
- Consistent infrastructure
- Based on industry standards avoiding costly proprietary implementation
- Standards hiding underlying implementation allowing for more efficient debugging/testing

Business and Technical Benefits

Unification of Applications

- New ways of accessing old applications
- Applications can be integrated without regard to implementation details
- Applications can dynamically navigate, discover, and interact over the Internet (loosely coupled)

Organized Information

- Targeted, more relevant

IBM and Web Services Standards

UDDI - Universal Description, Discovery and Integration

- Leader in creation of UDDI project
- Host of UDDI Business Registry

SOAP - Simple Object Access Protocol

- Co-author of specification
- Chair of XML Protocol working group in W3C
- Contributor of SOAP4J to Apache open source project

WSDL - Web Services Description Language

- Co-author of specification
- First WSDL toolkit implementation on alphaWorks

Application Servers and Development Tools

On iSeries - available today

Servers

- Web Services first supported with WebSphere Application Server V4.0
  - Web Services shipped with both editions
    - Advanced Edition (AE)
    - Advanced Edition Single Server (AEs)
- WebSphere Application Server V5.0
  - WSGW: Web Service Gateway
  - UDDI Registry
- WebSphere Application Server V5.0 Express
- HTTP Server Powered By Apache Tomcat

WebSphere Runtime Base Web Services Support

Rich support for SOAP based Web Service hosting and invocation

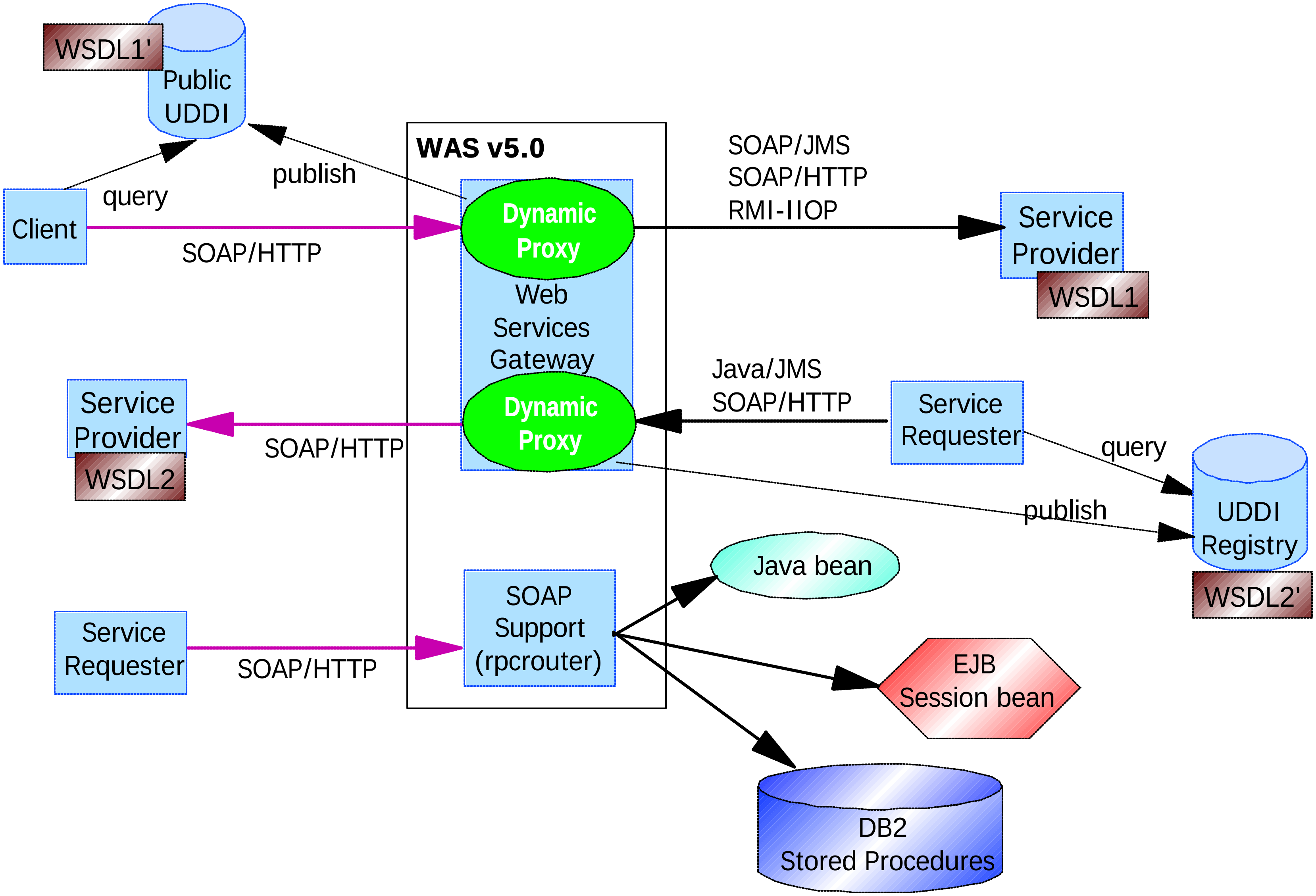
Web Service support for SOAP/HTTP as service provider

Application Server can act as a Web Service requester using SOAP/xxx, for example, SOAP/JMS

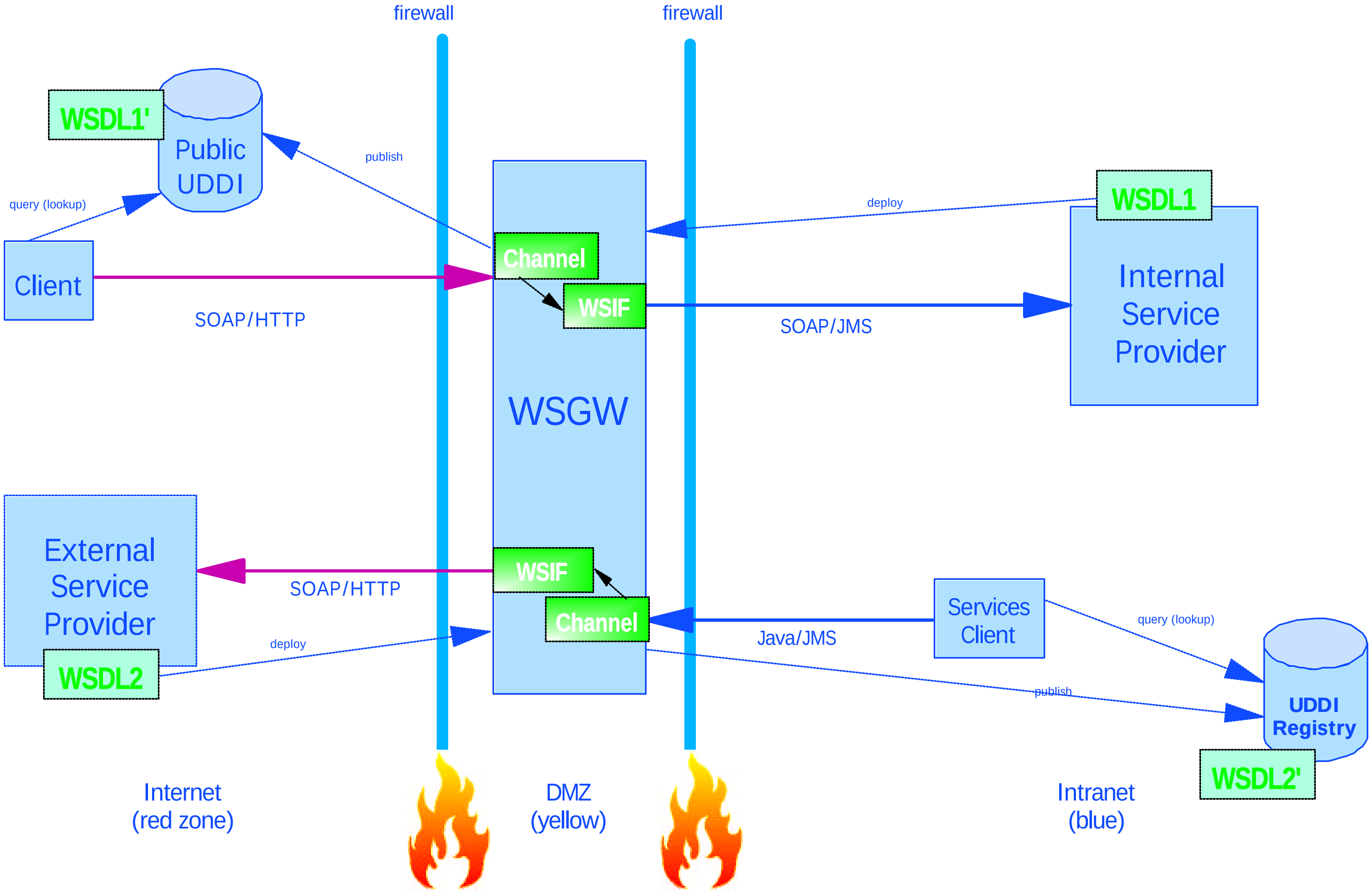
Supported Levels

- Web Services Definition Language (WSDL) Version 1.1
  - XML format for describing Web Services
- SOAP Version 2.3
- UDDI Version 2.0
  - Java APIs for UDDI actions (e.g. retrieve, process results, query, send, publish, etc.)
- WSIL 1.0

WebSphere V5.0 Web Services



Web Services Gateway



Web Services Gateway (continued)

Benefits of using the Gateway

- J2EE application
- Application server hosts the service proxy
- Provides centralized management of Web Services (including 3 party Web Services)
- Handles protocol translation

Deploying a Web Service to the Gateway

- The Web Service must exist
- Deploy existing binding information for Web Service and deploy to Web Services Gateway
- Gateway acts as a dynamic proxy for the service and a new WSDL file is created
- Web Services client utilizes new WSDL binding to access

On iSeries - available today

Development

- WebSphere Studio Development Environment
  - WebSphere Studio Site Developer for Windows (beta)
  - WebSphere Studio Application Developer for Windows (WSAD)
  - WebSphere Development Studio Client for iSeries (WDSc)
- Web Services Toolkit
  - Implementations of emerging Web Services technologies
  - Available on IBM alphaWorks: *www.alphaworks.ibm.com*

iSeries Example

iSeries Example

Pop Quiz "Three"!

Can you leverage Web Services with existing iSeries applications?

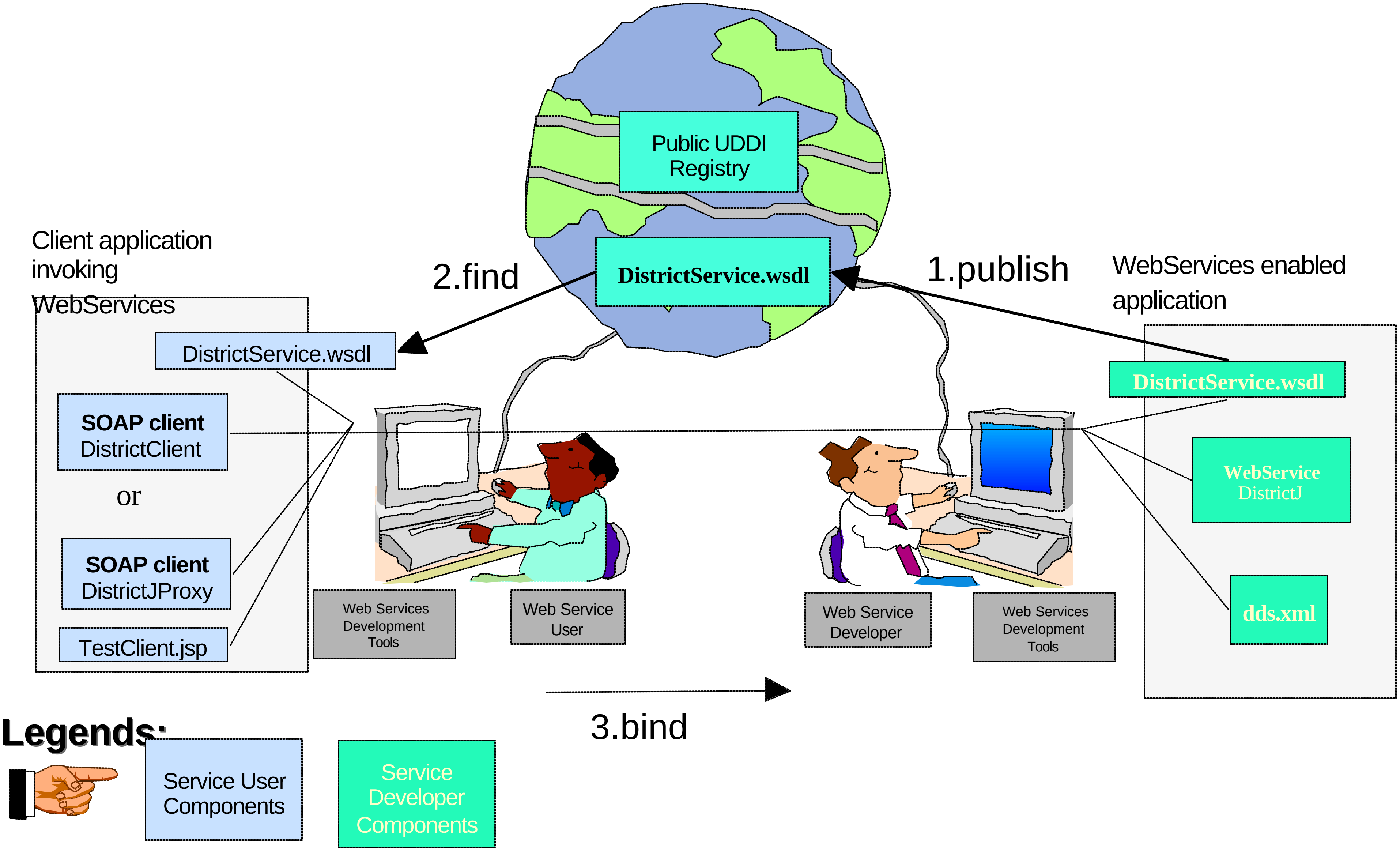
Answer: ***"Of course"!***

iSeries Example

Our example,

- turned an RPG program into a Web Service
- started with a existing RPG ILE service program
  - no recompile necessary
  - looks up and returns the year to date balance for a specified sales district
- created a simple Java 'wrapper' program to invoke the RPG service program
- used WDSc wizards to:
  - create a WSDL from the Java program
  - create a Web service proxy for use with by client Web applications
    - also used this to develop a Java (thick) client application
  - create a JSP for testing
  - deployed the Web service application to a WebSphere application server
    - application included: Java program, JSPs, proxy, SOAP server

iSeries Example: Development Environment



Public UDDI Registry

DistrictService.wsdl

2.find

1.publish

Client application invoking WebServices

DistrictService.wsdl

SOAP client DistrictClient

or

SOAP client DistrictJProxy

TestClient.jsp

Web Services Development Tools

Web Service User

Web Service Developer

Web Services Development Tools

WebServices enabled application

DistrictService.wsdl

WebService DistrictJ

dds.xml

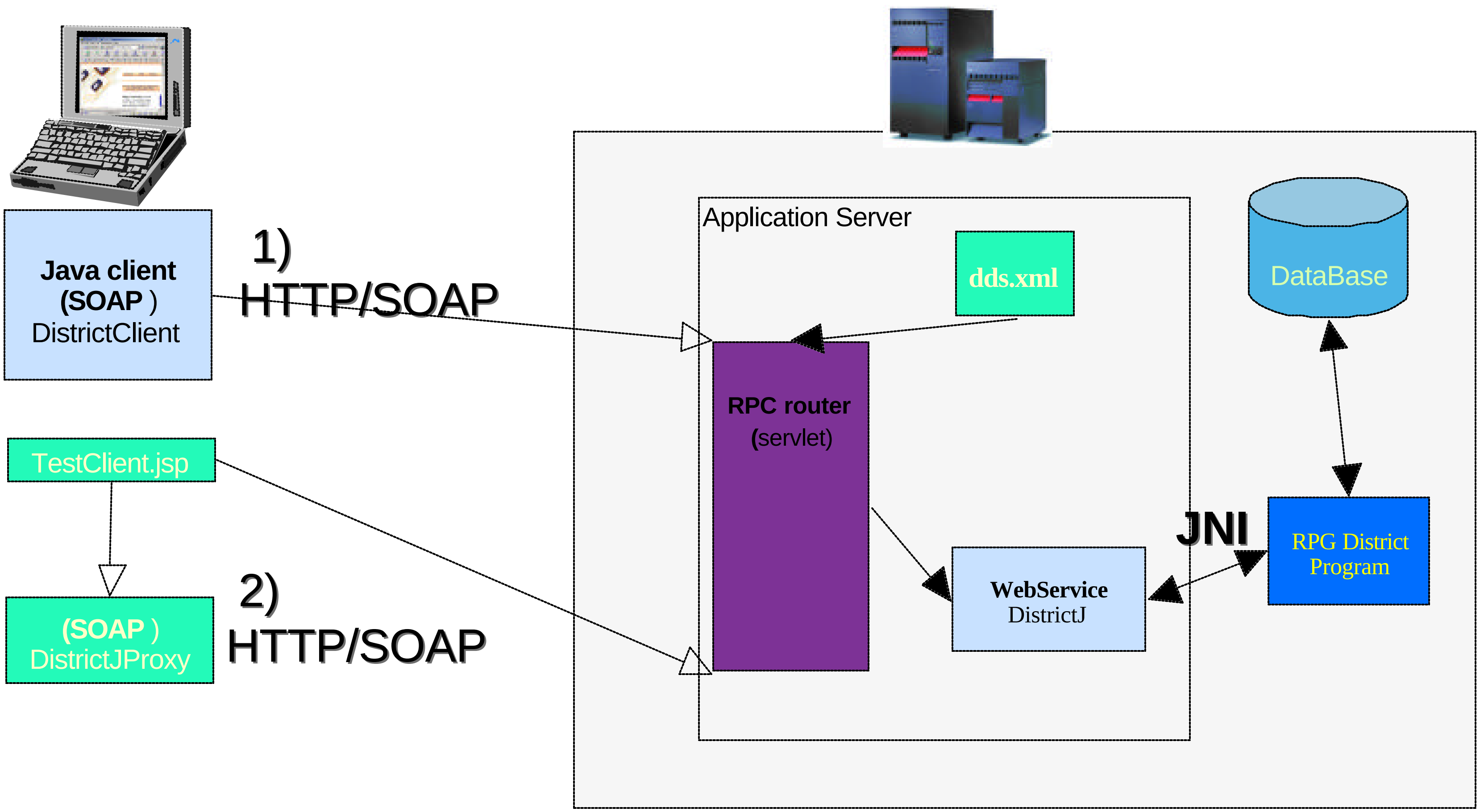
3.bind

Legends:

Service User Components

Service Developer Components

iSeries Example: Components Overview



Java client (SOAP) DistrictClient

1) HTTP/SOAP

TestClient.jsp

(SOAP) DistrictJProxy

2) HTTP/SOAP

Application Server

dds.xml

RPC router (servlet)

WebService DistrictJ

JNI

RPG District Program

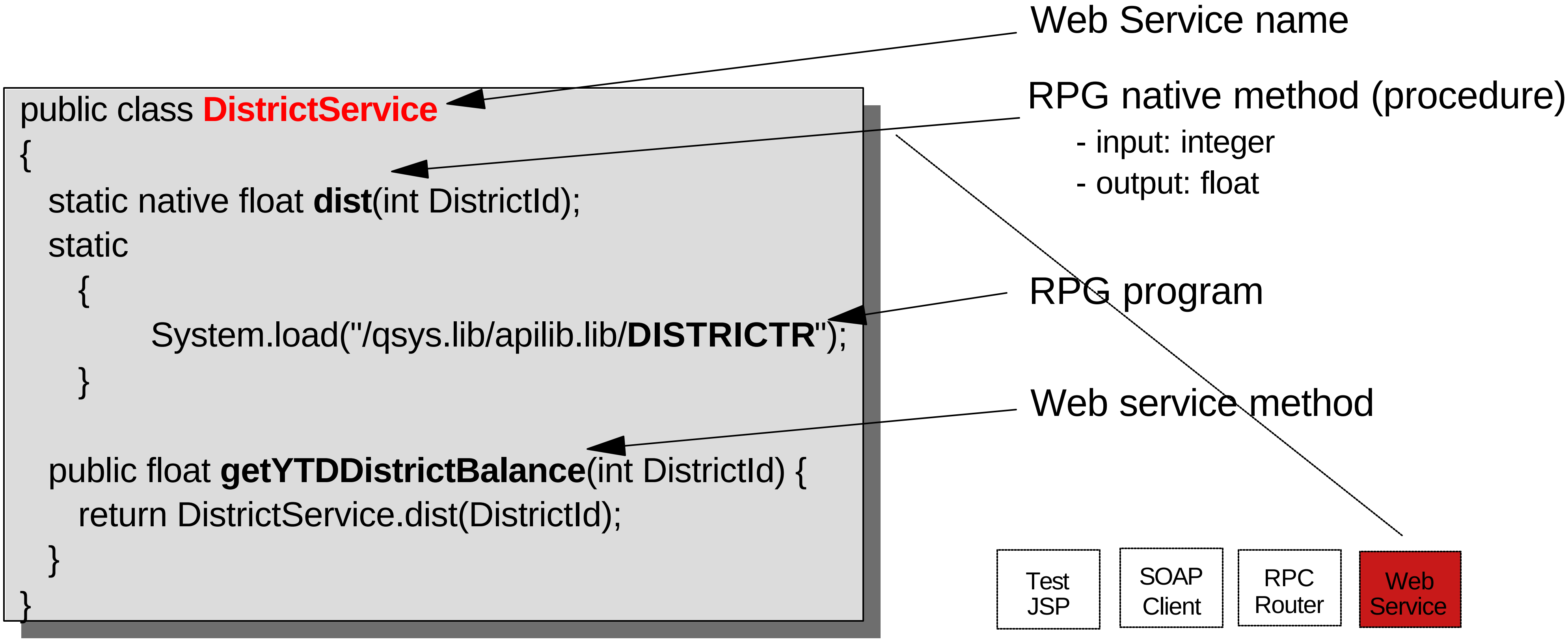
DataBase

Legends:

User code | WAS code | Other code | WDSc code

iSeries Example: RPG Web Service

DistrictService is a Web service that returns the YTD balance for the specified district. The Web service is implemented as a Java servlet that invokes an ILE RPG program that retrieves the YTD balance from a DB.



```
public class DistrictService
{
   static native float dist(int DistrictId);
   static
      {
            System.load("/qsys.lib/apilib.lib/DISTRICTR");
      }

   public float getYTDDistrictBalance(int DistrictId) {
      return DistrictService.dist(DistrictId);
   }
}
```

Web Service name

RPG native method (procedure)

- input: integer
- output: float

RPG program

Web service method

iSeries Example - WSDL

1 Type Container

- XSD=XML Schema

2 Message Definition

- Input and output
  - with parameters

3 Port Type

- Operation and method
  - Points to Input/Output messages

4 Binding

- Style and encoding

5 Location of Web Service

- Where it is installed

iSeries Example - WSDL

1 Type Container

XSD=XML Schema

```
 <?xml version="1.0" encoding="UTF-8"?>
 <definitions name="DistrictService"

targetNamespace="http://www.DistrictService.wsdl.com/wrapperedService"
   xmlns="http://schemas.xmlsoap.org/wsdl/"
   xmlns:dis="http://www.DistrictService.wsdl.com/wrapperedService"
   xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
     xmlns:xsd="http://www.w3.org/1999/XMLSchema">
```

iSeries Example - WSDL

2

Message Definition

Input and output

with parameters

```
<message
     name="IngetYTDDistrictBalanceRequest">
  <part
     name="meth1_inType1" type="xsd:int"/>
</message>

<message
     name="OutgetYTDDistrictBalanceResponse">
  <part
     name="meth1_outType" type="xsd:float"/>
</message>
```

iSeries Example - WSDL

3

Port Type

Operation and method

Points to Input/Output messages

```
<portType
      name="DistrictService">
  <operation
        name="getYTDDistrictBalance">
    <input
        message="IngetYTDDistrictBalanceRequest"/>
    <output
    message="OutgetYTDDistrictBalanceResponse"/>
  </operation>
</portType>
```

iSeries Example - WSDL

4 Binding

Style and encoding

```
<binding
    name="DistrictServiceBinding"
      type="DistrictService">
  <soap:binding style="rpc"
      transport="http://schemas.xmlsoap.org/soap/http"/>
  <operation
        name="getYTDDistrictBalance">
    <soap:operation
          soapAction="urn:DistrictService"/>
    <input>
      <soap:body
          encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
          namespace="urn:DistrictService"
          use="encoded"/>
    </input>
    <output>
      <soap:body
          encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
          namespace="urn:DistrictService"
    use="encoded"/>
    </output>
  </operation>
</binding>
```

iSeries Example - WSDL

5 Location of Web Service

Where it is installed

```
 <service
       name="DistrictService">
   <documentation>WSTK 1.2 wrappered class DistrictService as
service</documentation>
   <port
       binding="DistrictServiceBinding"
         name="DistrictServicePort">
     <soap:address location="http://brs2am:10102/soap/servlet/rpcrouter"/>
   </port>
 </service>
 </definitions>
```

iSeries Example: Java client SOAP application

The following is an example of code found in a Java client application that directly uses SOAP to invoke the **getYTDDistrictBalance** method of the Web service **DistrictService**.

```
 // Build the call.
   Call call = new Call ();
   call.setTargetObjectURI ("urn:DistrictService");
   call.setMethodName ("getYTDDistrictBalance");
   call.setEncodingStyleURI(Constants.NS_URI_SOAP_ENC);
   Vector params = new Vector ();
   params.addElement (new Parameter("districtID", Int.class, districtID, null));
   call.setParams (params);

// Make the call
   Response resp = call.invoke ("localhost:port/servlet/rpcrouter", "" );
   Parameter result = resp.getReturnValue ();
   System.out.println (result.getValue ());
catch(SOAPException e){
System.err.println("SOAPException =" +e.getFaultCode()
                   + "" +e.getMessage()); }
```

Create a new call

Specify the target object

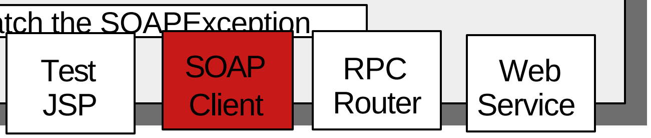
Specify the method name

Specify the method parameters

Invoke the service

Get the result

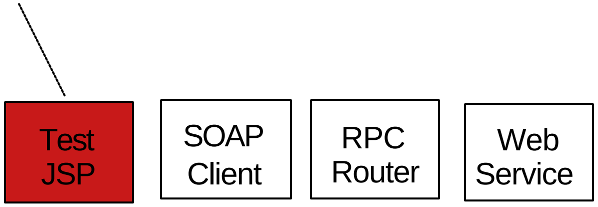
Catch the SOAPException



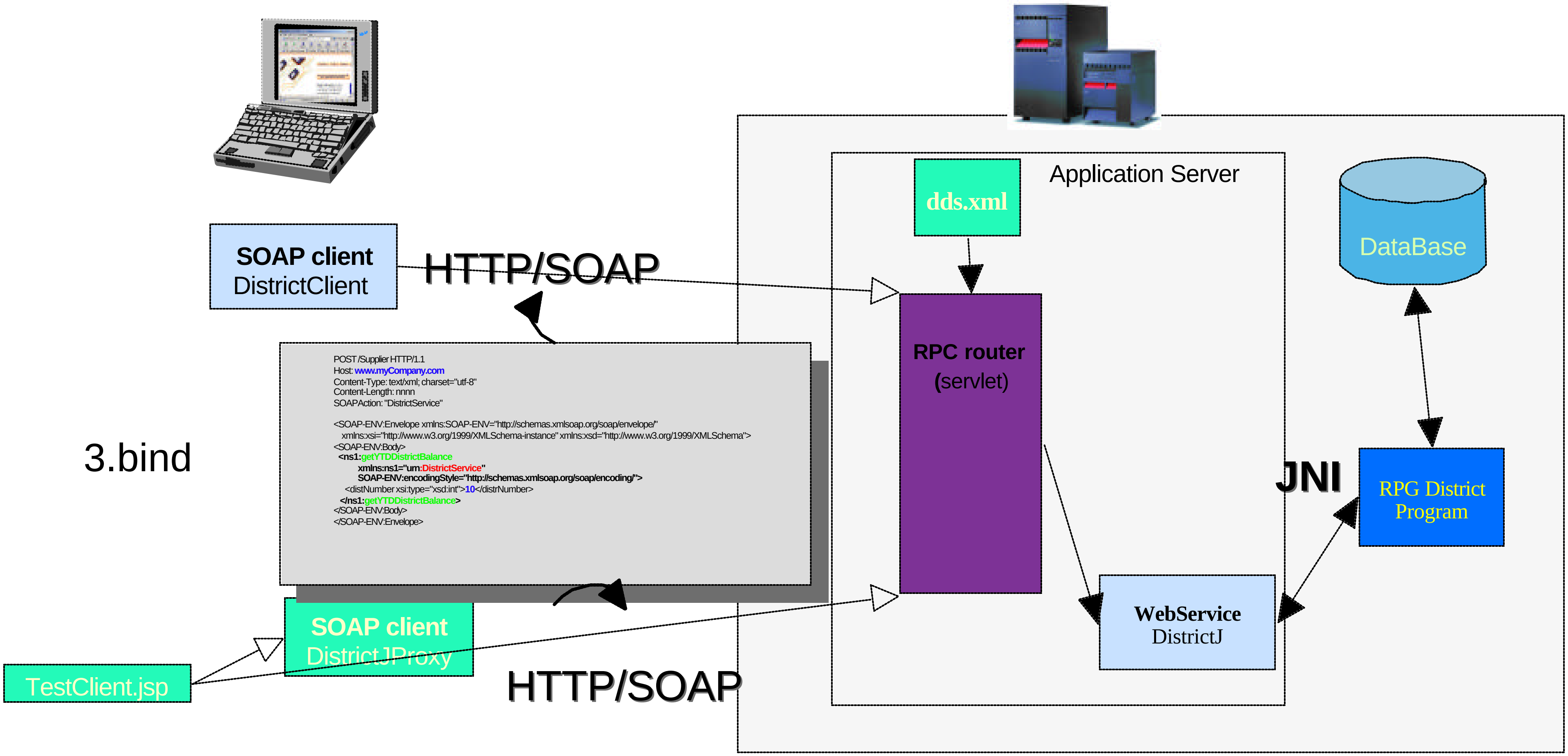
iSeries Example: JSP example

The following is code found in a JSP that invokes the Web service proxy **DistrictServiceProxy.** The proxy will generate a SOAP request similar to the request made by our Java client application.

```
<HTML>
<HEAD>
<TITLE>Result</TITLE>
</HEAD>
<BODY>
<H1>Result</H1>
<jsp:useBean id="proxy" scope="session" class="services.DistrictServiceProxy" />
int temp1  = 1;
float mtemp = proxy.getYTDDistrictBalance(temp1);
result = markup(String.valueOf(mtemp));
</BODY>
</HTML>
```



iSeries Example: Runtime Environment



```
POST /Supplier HTTP/1.1
Host: www.myCompany.com
Content-Type: text/xml; charset="utf-8"
Content-Length: nnnn
SOAPAction: "DistrictService"

<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
   xmlns:xsi="http://www.w3.org/1999/XMLSchema-instance" xmlns:xsd="http://www.w3.org/1999/XMLSchema">
<SOAP-ENV:Body>
  <ns1:getYTDDistrictBalance
          xmlns:ns1="urn:DistrictService"
          SOAP-ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">
     <distNumber xsi:type="xsd:int">10</distrNumber>
  </ns1:getYTDDistrictBalance>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Legends:

User code	WAS code	Other code	WDSc code

iSeries Example: SOAP Request Message

The SOAP request indicates that the **getYTDDistrictBalance** method, from the **DistrictService** namespace, should be invoked from **www.myCompany.com.** Upon receiving this request, the supplier application at www.myCompany.com executes the business logic that corresponds to **getYTDDistrictBalance.**

```
POST /Supplier HTTP/1.1
Host: www.myCompany.com
Content-Type: text/xml; charset="utf-8"
Content-Length: nnnn
SOAPAction: "DistrictService"

<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
    xmlns:xsi="http://www.w3.org/1999/XMLSchema-instance" xmlns:xsd="http://www.w3.org/1999/XMLSchema">
<SOAP-ENV:Body>
  <ns1:getYTDDistrictBalance
          xmlns:ns1="urn:DistrictService"
          SOAP-ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">
     <distNumber xsi:type="xsd:int">10</distrNumber>
   </ns1:getYTDDistrictBalance>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Business Cases

New Channel Business Case

Goal

- Add web service front-end to bring in additional revenue

Assumptions

- Existing legacy application
- Callable procedures, not intertwined with 5250 display ("event-driven")
- Existing development and iSeries production machines
  - 10% available capacity
  - Connected to Internet
- 2 developers with consulting help
  - Developers have some level of Java skills
  - 1 week training, 1 week planning, 2 month development

New Channel Business Case

Development Expenses (one time):	**$60,000**
Development costs (5 pm @ 100K$/yr)	$41,000
Consulting	$19,000
New hardware and software	$0
Operating Expenses (per year):	**$18,000**
System cost of ownership (10% of 43K$/yr)	$4,300
Help Desk (training + 10% @ 40K$/yr)	$5,500
Billing operation (10% @ 40K$/yr)	$4,000
Program maintenance (1/2 pm @ 100K$/yr)	$4,200
Intangible Benefits:	
Enhanced customer loyalty	
Increased employee retention	

New Channel Business Case

	Year 1	Year 2	Year3		
Fixed Exp	$60,000				
Var Exp	$18,000	$18,000	$18,000		
Total Exp	$78,000	$18,000	$18,000		
Revenue					
$100/day	$25,000	$25,000	$25,000		
$200/day	$50,000	$50,000	$50,000		
$300/day	$75,000	$75,000	$75,000		
Cash Flow				**NPV**	**break-even**
$100/day	($53,000)	$7,000	$7,000	**($36,191)**	**103**
$200/day	($28,000)	$32,000	$32,000	**$20,889**	**23**
$300/day	($3,000)	$57,000	$57,000	**$77,970**	**13**

Cost Savings Business Case

Previous business case did not assume any

- Decreased costs
- Enhanced efficiencies

Goal

- Redirect existing customers to web service front-end

Assumptions

- Same as previous case, except
- Free one person, half-time ($20,000/yr) to do something else
  - Less data re-entry
  - More accurate input data

Cost Savings Business Case

	Year 1	**Year 2**	**Year3**		
Fixed Exp	$60,000				
Var Exp	$18,000	$18,000	$18,000		
Savings	($20,000)	($20,000)	($20,000)		
Total Exp	$58,000	($2,000)	($2,000)		
Revenue					
$100/day	$25,000	$25,000	$25,000		
$200/day	$50,000	$50,000	$50,000		
$300/day	$75,000	$75,000	$75,000		
Cash Flow				**NPV**	**break-even**
$100/day	($33,000)	$27,000	$27,000	**$9,473**	**27**
$200/day	($8,000)	$52,000	$52,000	**$66,554**	**14**
$300/day	$17,000	$77,000	$77,000	**$123,634**	**9**

Most Likely Scenario 1

Between businesses

- Providing service to your customers
  - e. g.., How long & how much does it cost an Insurance company to link its systems to the systems of a new institutional customer?
- Accessing services from your partners and suppliers
  - e. g.., Dynamically link to new partners and suppliers to offer their services to complement the value you provide?
- Standards and common infrastructure reduce the barriers
- Simplicity accelerates deployment
- Dynamics opens new business opportunities

Most Likely Scenario 2

Within a business

- Accelerate and reduce the cost of integration
- Save on infrastructure deployment and management costs
- Reduce skill requirements
- Improve reuse

Most Likely Scenario 3

Between a business and end-users

- Deliver a better user experience
- Integrate diverse content
- Reduce the cost of content delivery

Web Services Resources

WebSphere Application Server for iSeries

- http://www.ibm.com/servers/eserver/iseries/software/websphere/wsappserver/index.html
- http://www.iseries.ibm.com/developer/websphere/

WebSphere Application Server

- http://www.ibm.com/software/webservers/
- IBM HTTP Server Powered By Apache
  - www.ibm.com/servers/eserver/iseries/software/http/

Development tools

- WebSphere Studio
  - www.ibm.com/software/webservers/studio
- WebSphere Studio Development Environment Preview (WSSD and WSAD Betas)
  - www.ibm.com/software/webservers/studio/preregister.html
- IBM Web Services Toolkit (WSTK)
  - www.alphaworks.ibm.com

Web Services Information

- www.ibm.com/developerworks
- UDDI: Specifications and architecture:
  - www.uddi.org
- UDDI: Java client (UDDI4J):
  - www.alphaworks.ibm.com
- UDDI: IBM
  - www.ibm.com/services/uddi
  - business registry: www.ibm.com/services/uddi/protect/registry.html
  - business test registry: www.ibm.com/services/uddi/testregistry/protect/registry.html
- SOAP: Specifications
  - www.w3.org/TR/SOAP
- Case Studies
  - www.ibm.com/software/ebusiness/jstart/casestudies