Web Services Why, What, How

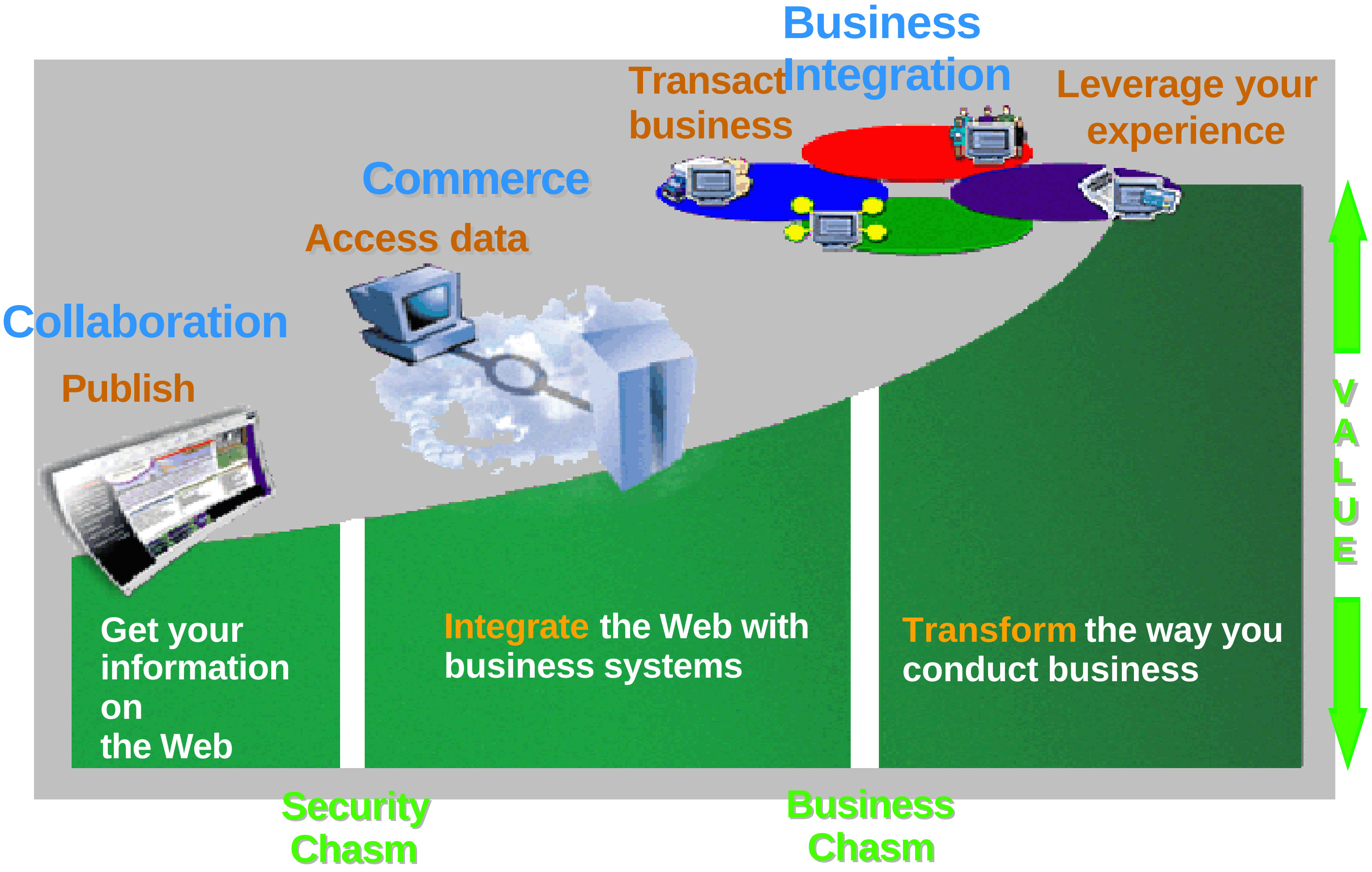
WebSphere

The Fastest Way to e-business

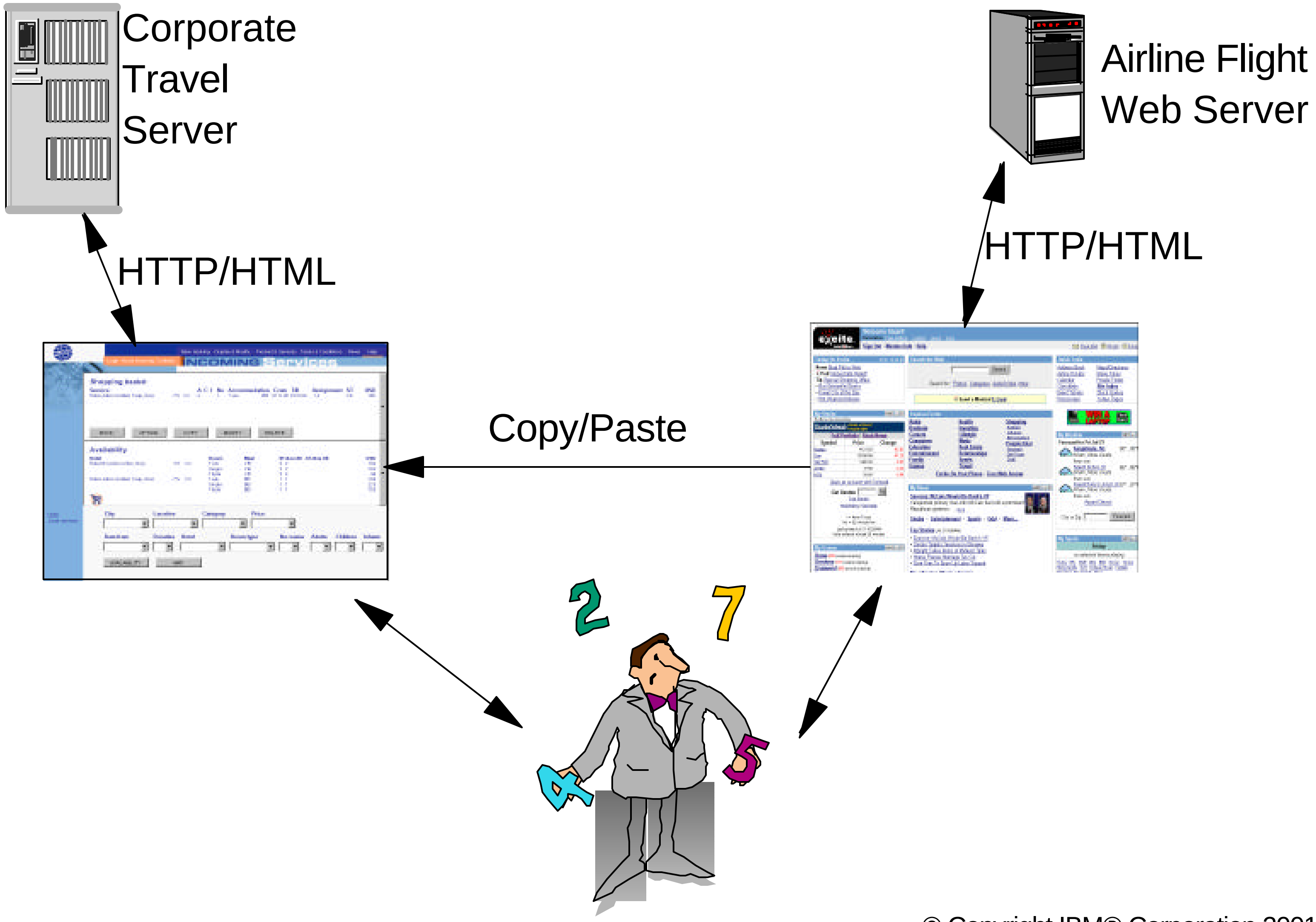
Agenda

- **Why Web Services?**
- **What are Web Services?**
- **Web Services Protocols**
- **Web Services Architecture**
- **IBM Software for Web Services**
- **IBM Tooling directions for Web Services**

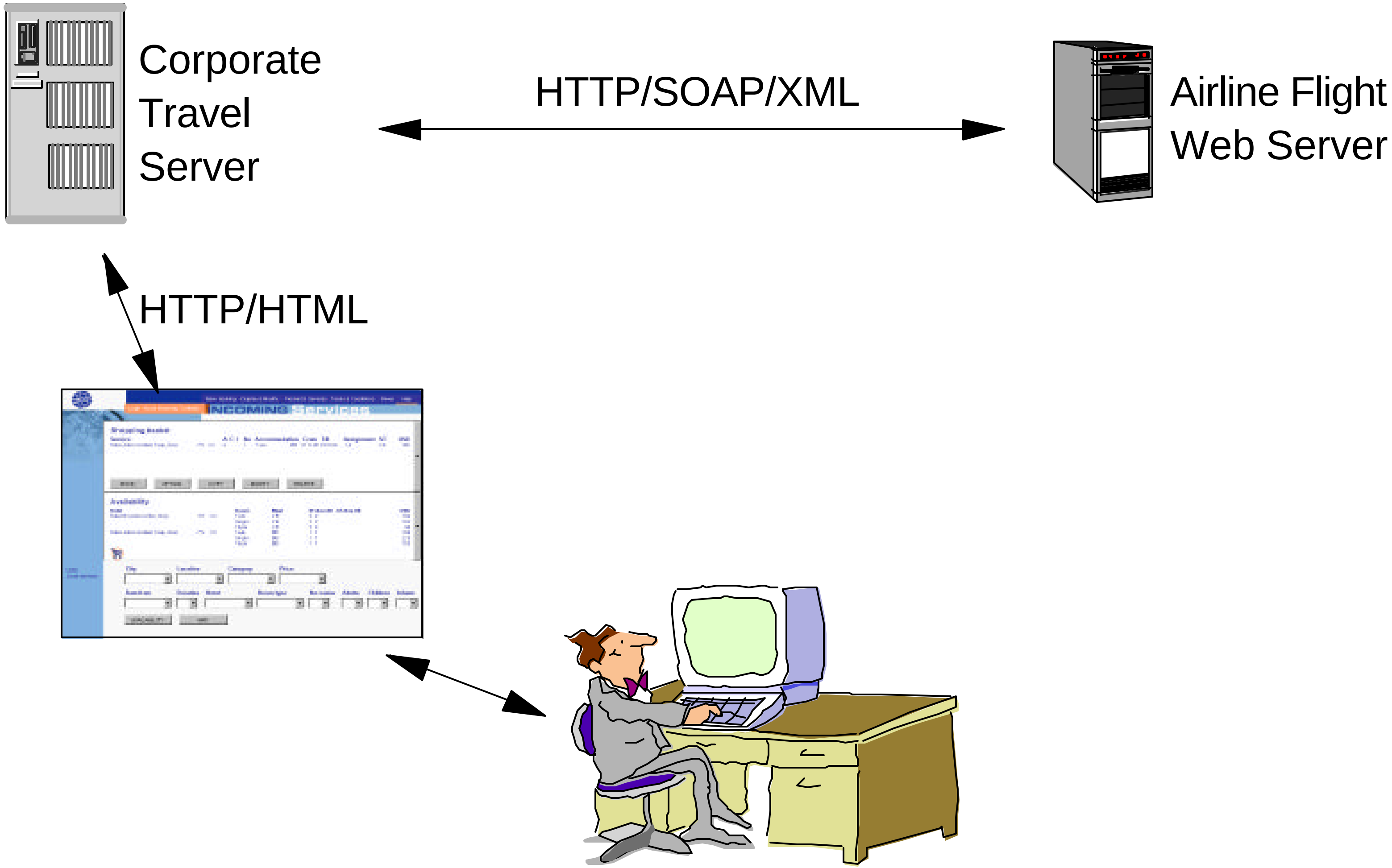
Evolution of e-business



Benefits of Web Services



Benefits of Web Services



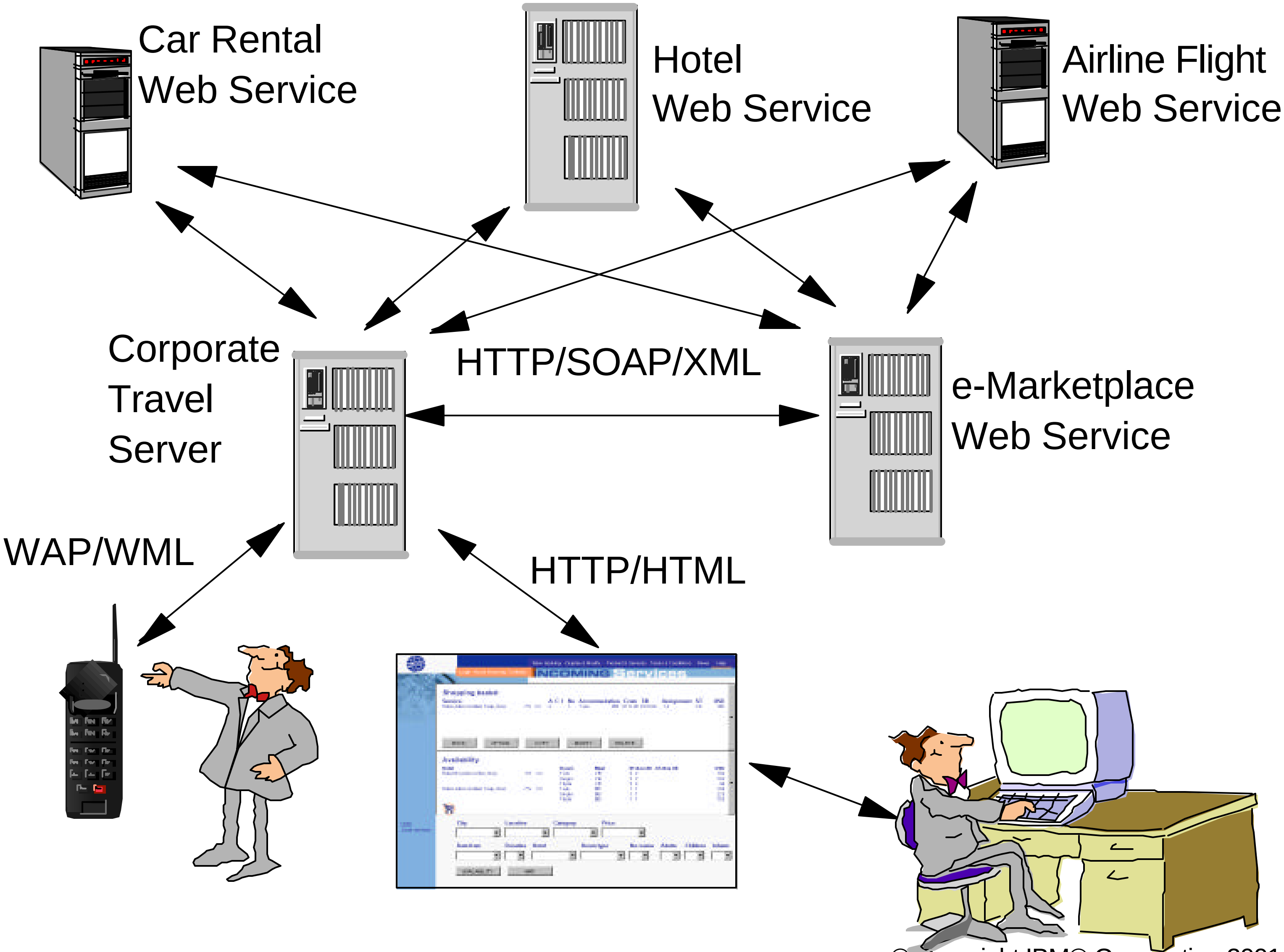
Corporate Travel Server

HTTP/SOAP/XML

Airline Flight Web Server

HTTP/HTML

Benefits of Web Services



Car Rental Web Service

Hotel Web Service

Airline Flight Web Service

Corporate Travel Server

HTTP/SOAP/XML

e-Marketplace Web Service

WAP/WML

HTTP/HTML

Evolution of the Web

SOAP	Simple Object Access Protocol
WSDL	Web Services Description Language
UDDI	Universal Description Discovery & Integration
WSFL	Web Services Flow Language

Document Web

- People get Information
- Web servers
- HTTP protocol
- HT

Application Web

- People run transactions
- Application servers
- Business logic (Java)
- Generate HTML
- Distributed processing
- WAP, WML

Service Web

- Programs request Information and run transactions
- Web services servers
- Generate XML
- SOAP, WSDL, UDDI

What are Web Services?

Web Services are self-contained, self-describing, modular applications that can be published, located, and invoked over a network, generally, the Web.

Universal program- to- program communication model based on standards and industry support

e-business is the driving force

- Merge of Web, IT, object technologies
- Highly-interoperable Web-based objects
- Object-oriented programming through SOAP messages
- Expose business functions or data access from existing enterprise code using SOAP wrappers and WSDL descriptions

► **Everything is a service, publishing an API for use by other services on the network and encapsulating implementation details**

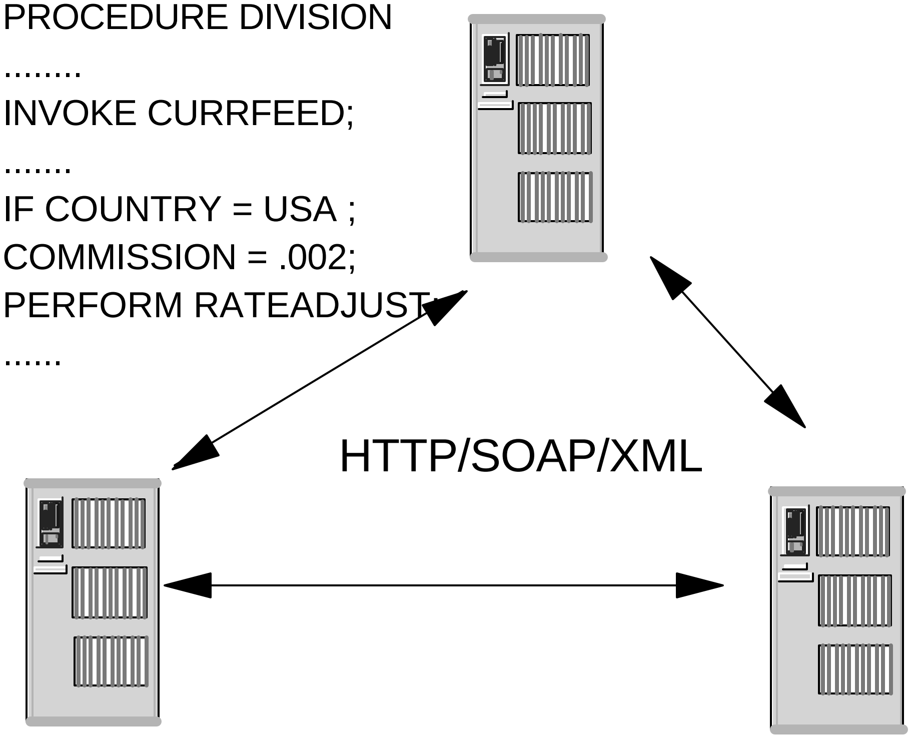
What are Web Services?

- ❑ Self-contained
  - no additional software (HTTP, XML, Application Server)
- ❑ Self-describing
  - Definition of message travels with message
- ❑ Modular
  - Callable services
- ❑ Published, located, invoked (SOAP, WSDL, UDDI)
- ❑ Language independent and interoperable
  - Different environments, can make existing code into a Web Service
- ❑ Open and standards-based
  - Web Services = HTTP+ XML
- ❑ Dynamic
  - Discovery and invocation can be automated
- ❑ Composable
  - Web service can invoke other Web Services

Examples

- ❑ Business information with rich content
  - weather reports
  - stock quotes
  - airline schedules
  - credit check
  - news feed
- ❑ Transactional Web Services for B2B, B2C
  - airline reservation
  - rental car agreement
  - supply chain mgmt
- ❑ Business process externalization
  - business linkage at workflow level
  - complete integration at process level

Benefits of Web Services : micro level

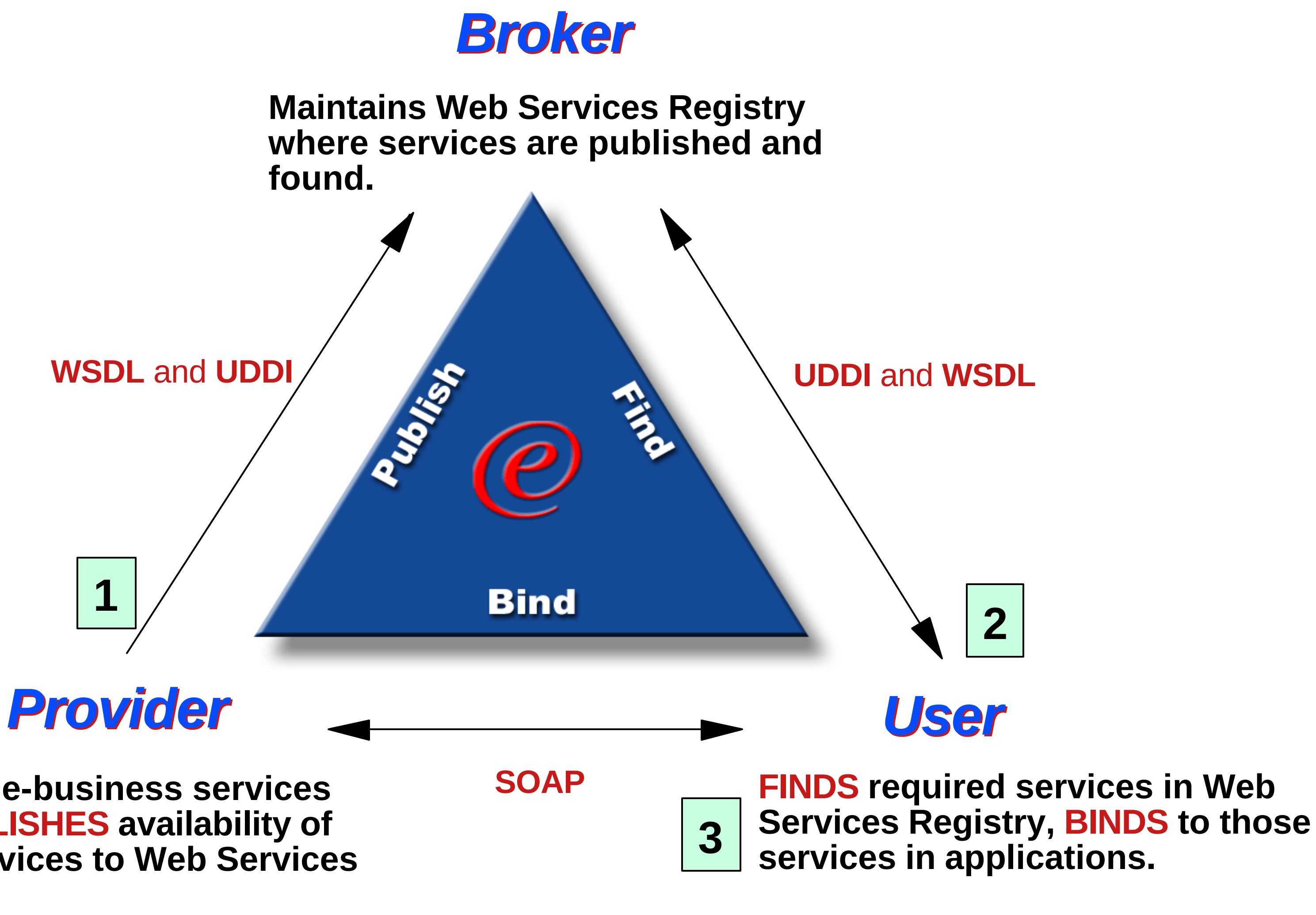


```
PROCEDURE DIVISION
........
INVOKE CURRFEED;
.......
IF COUNTRY = USA ;
COMMISSION = .002;
PERFORM RATEADJUST;
......
```

```
{.....
  if cust.citizenof = "France";
    {if cust.status = "Gold"
         { discount = 0.003);
      .......
}
```

```
{.....
  float rate = exchange.getRate("USA", "Germany");
........
  customer.getDiscount("France", "Gold");
.......
}
```

Web Services Participants and their Roles



Broker

Maintains Web Services Registry where services are published and found.

WSDL and UDDI

UDDI and WSDL

1

2

Provider

Provides e-business services and PUBLISHES availability of these services to Web Services Registry.

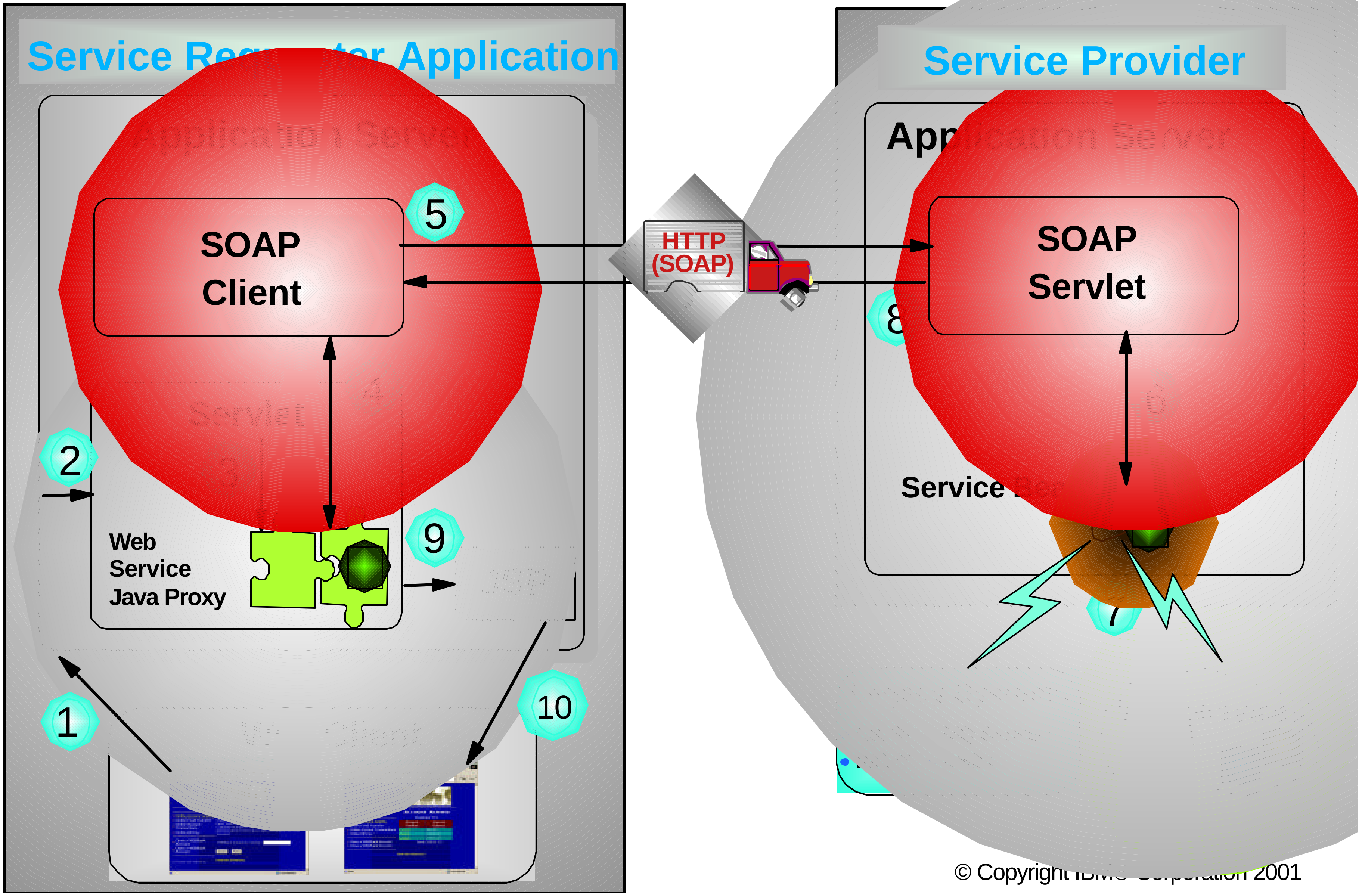
SOAP

3

User

FINDS required services in Web Services Registry, BINDS to those services in applications.

Web Services Runtime View



What is SOAP?

- Simple Object Access Protocol
- SOAP is an XML based protocol for communication between two remote applications:
  - is based on RPC messaging
  - is language independent (<u>de-couples</u> interface from implementation)
  - represents remote procedure calls and responses
- A SOAP message consists of:
  - envelope
    - wraps the message itself
    - defines rules for decoding the message
  - message
    - request
      - method to invoke on a remote object and parameters
    - response
      - result of running the method and exceptions

SOAP Example - Request

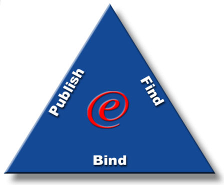
```
POST /soap/servlet/rpcrouter HTTP/1.0
Host: localhost:8081
Content-Type: text/xml
Content-Length: 510

SOAPAction: "urn:LowestMortgageRateWebService"



<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:submitRateInquiry xmlns:ns1="urn:LowestMortgageRateWebService"
                         SOAP-ENV:encodingStyle="http://schemas.xmlsoap.org/soap/
                                                encoding/">

   <meth1_inType1 xsi:type="xsd:string">1</meth1_inType1>
   <meth1_inType2 xsi:type="xsd:string">12</meth1_inType2>

  </ns1:submitRateInquiry>

 </SOAP-ENV:Body>

</SOAP-ENV:Envelope>
```

e-business

SOAP Example - Response

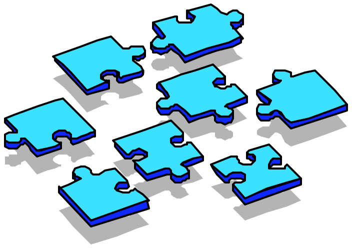
```
HTTP/1.0 200 ok
Content-Type: text/xml;; charset=UTF-8
Set-Cookie: sesessionid=CSIQFDYAAAAAC5YAAAAX5ZI;Path=/
Cache-Control: no-cache="set-cookie,set-cookie2"
Expires: Thu, 01 Dec 1994 16:00:00 GMT
Content-Length: 460
Content-Language: en


<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:submitRateInquiryResponse xmlns:ns1="urn:LowestMortgageRateWebService"
                                 SOAP-ENV:encodingStyle="http://schemas.xmlsoap.
                                                        org/soap/encoding/">

   <return xsi:type="xsd:boolean">false</return>

  </ns1:submitRateInquiryResponse>

 </SOAP-ENV:Body>

</SOAP-ENV:Envelope>
```

What is WSDL?



- Web Services Description Language
- WSDL is an XML based vocabulary for defining a Web Service:
  - interfaces
    - operation types (i.e. one-way, request-response, notification)
    - messages defining a Web Service interface
    - definition of data types (XML Schema)
  - access protocol (i.e. SOAP over HTTP)
  - contact endpoints (i.e. Web Service URL and URNs[1])

- A Web Service URL returning WSDL makes Web Services self-describing
- Similar in purpose to IDL (Interface Definition Language)
  - From a WSDL file, wizards can generate:
    - proxy classes for calling Web Service
    - skeleton classes to implement a Web Service



WSDL Example - Outline

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

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

  <message>
  ...
  </message> ...


  <portType>
  ...
  </portType>

  <binding>
  ...
  </binding>

  <service>
  ...
  </service>

</definitions>
```

WSDL Example - Message Section

```
<message name="InsubmitRateInquiryRequest">
  <part name="meth1_inType1" type="xsd:string"/>
  <part name="meth1_inType2" type="xsd:string"/>
</message>

<message name="OutsubmitRateInquiryResponse">
  <part name="meth1_outType" type="xsd:boolean"/>
</message>

<message name="InreceiveRateRequest">
  <part name="meth2_inType1" type="xsd:int"/>
</message>

<message name="OutreceiveRateResponse">
  <part name="meth2_outType"type="xsd:string"/>
</message>

<message name="InreceiveCommentRequest">
  <part name="meth3_inType1" type="xsd:int"/>
</message>

<message name="OutreceiveCommentResponse">
  <part name="meth3_outType" type="xsd:string"/>
</message>
```

WSDL Example - PortType Section

```
<portType name="LowestMortgageRateWebService">
  <operation name="submitRateInquiry">
    <input message="InsubmitRateInquiryRequest"/>
    <output message="OutsubmitRateInquiryResponse"/>
  </operation>

  <operation name="receiveRate">
    <input message="InreceiveRateRequest"/>
    <output message="OutreceiveRateResponse"/>
  </operation>

  <operation name="receiveComment">
    <input message="InreceiveCommentRequest"/>
    <output message="OutreceiveCommentResponse"/>
  </operation>
</portType>
```

WSDL Example - Binding Section

```
<binding name="LowestMortgageRateWebServiceBinding" type="LowestMortgageRateWebService">
  <soap:binding style="rpc" transport="http://schemas.xmlsoap.org/soap/http"/>
  <operation name="submitRateInquiry">
    <soap:operation soapAction="urn:LowestMortgageRateWebService"/>
    <input>
      <soap:body encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
                 namespace="urn:LowestMortgageRateWebService" use="encoded"/>
    </input>
    <output>
      <soap:body encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
                 namespace="urn:LowestMortgageRateWebService" use="encoded"/>
    </output>
  </operation>
  <operation name="receiveRate">
    <soap:operation soapAction="urn:LowestMortgageRateWebService"/>
    <input>
      <soap:body encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
                 namespace="urn:LowestMortgageRateWebService" use="encoded"/>
    </input>
    <output>
      <soap:body encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
                 namespace="urn:LowestMortgageRateWebService" use="encoded"/>
    </output>
  </operation>
  <operation name="receiveComment">
    <soap:operation soapAction="urn:LowestMortgageRateWebService"/>
    <input>
      <soap:body encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
                 namespace="urn:LowestMortgageRateWebService" use="encoded"/>
    </input>
    <output>
      <soap:body encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
                 namespace="urn:LowestMortgageRateWebService" use="encoded"/>
    </output>
  </operation>
</binding>
```

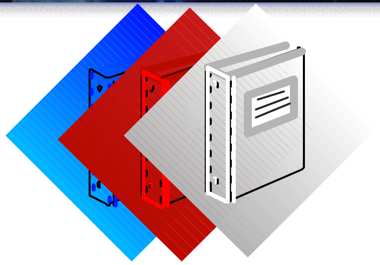
WSDL Example - Service Section

```
<service name="LowestMortgageRateWebService">
  <documentation>
    WSTK 1.2 wrappered class LowestMortgageRateWebService as service
  </documentation>
  <port binding="LowestMortgageRateWebServiceBinding"
name="LowestMortgageRateWebServicePort">
    <soap:address location="http://localhost:8080/soap/servlet/rpcrouter"/>
  </port>
</service>
```

What is UDDI?

- **Universal Description, Discovery and Integration**
- **UDDI is a specification for <u>publishing and discovery</u> of businesses and the services they provide**
- **UDDI specifications define how to construct UDDI Business Registries**
- **UDDI specifications are based on XML and SOAP:**
  - **API to communicate with a UDDI Registry are SOAP based**
    - **UDDI4J (UDDI for Java) - Open Source implementation in Java**
    - **JAXR (Java API for XML Registries) - Sun**
  - **data structures that define Web Service in UDDI Registry are XML based**
- **UDDI registration for a Web Service contains**
  - A Business Entity description
  - Description of the service (semantics and technical)
  - Access Locator

UDDI Business Registry

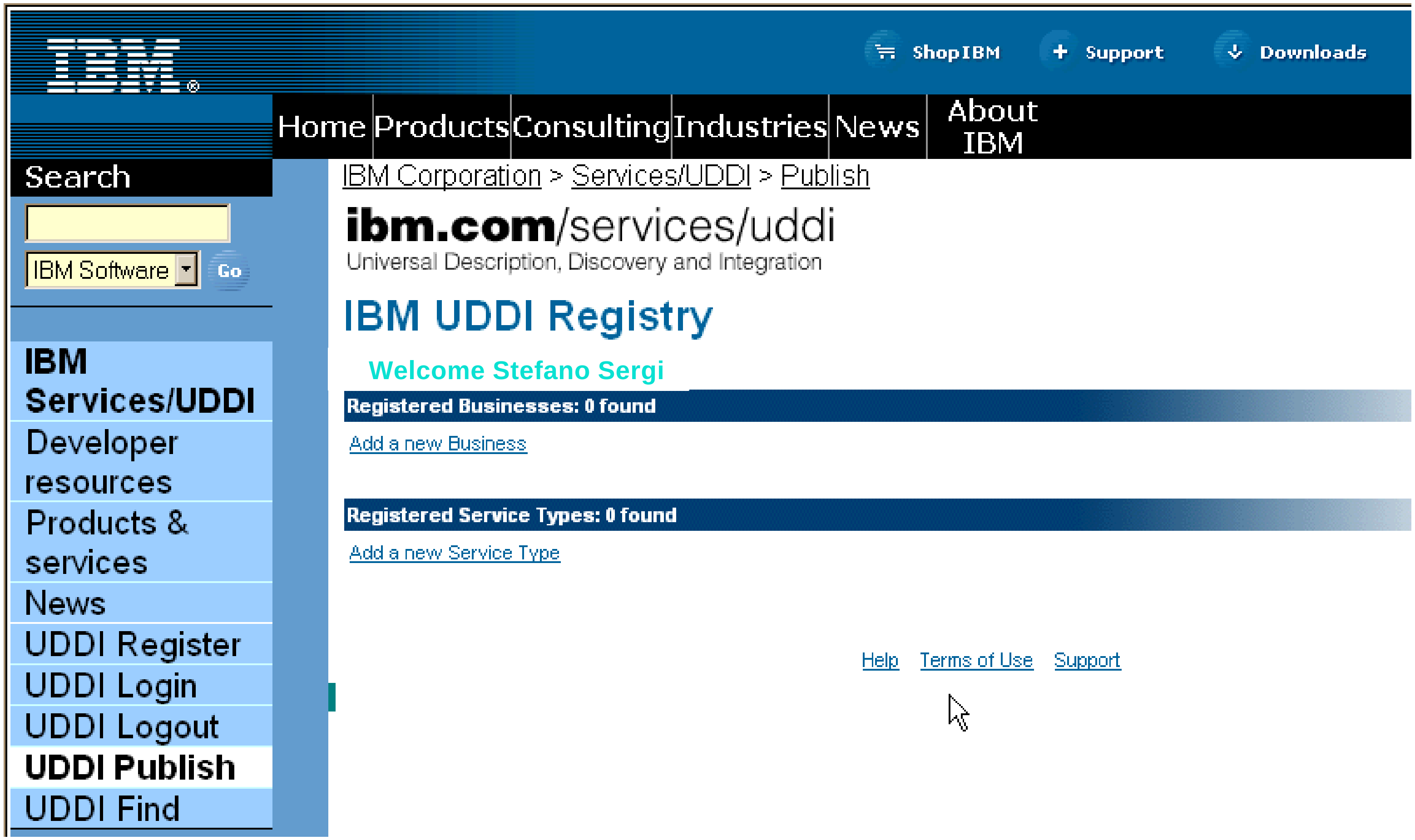


- There are three public registry nodes on the web
  - sponsored by IBM, Microsoft and HP
  - in UDDI Business Registries, businesses register information about themselves and their services
  - registration is free
  - registration entries are replicated to other nodes on daily basis
  - businesses can discover each other's services:
    - automated searches (UDDI APIs)
    - manual searches (search engines in UDDI Registry web sites)
- UDDI Registries are organized into three areas:

White Pages	business information	business name, description, contact information
Yellow Pages	service information	industrial categorizations based on standard taxonomies
Green Pages	binding information	the technical information about services that are exposed by the business

UDDI Registries are real!

https://www.ibm.com/services/uddi/protect/home.jsp



IBM Corporation > Services/UDDI > Publish

ibm.com/services/uddi
Universal Description, Discovery and Integration

IBM UDDI Registry

Welcome Stefano Sergi

Registered Businesses: 0 found

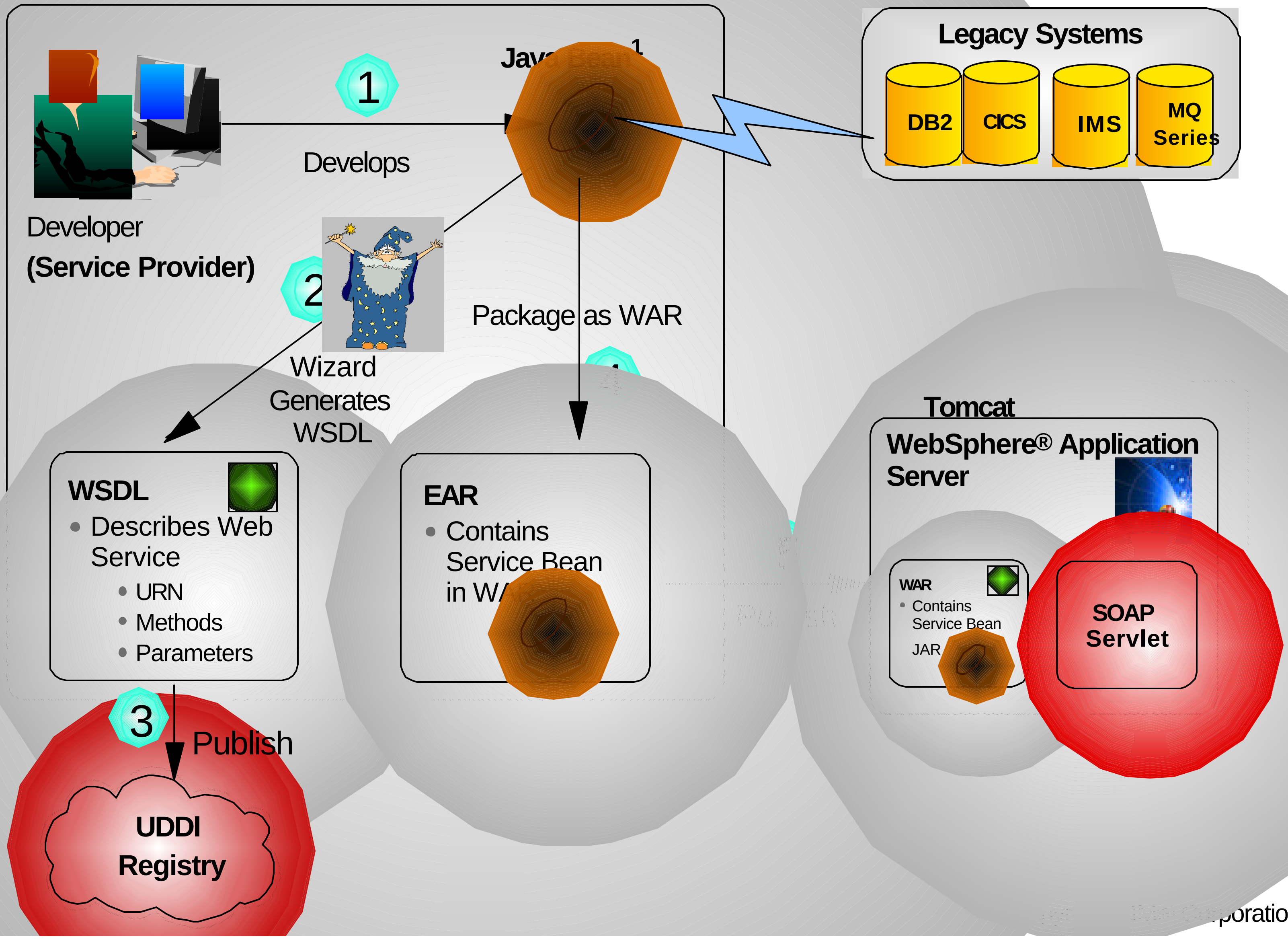
Add a new Business

Registered Service Types: 0 found

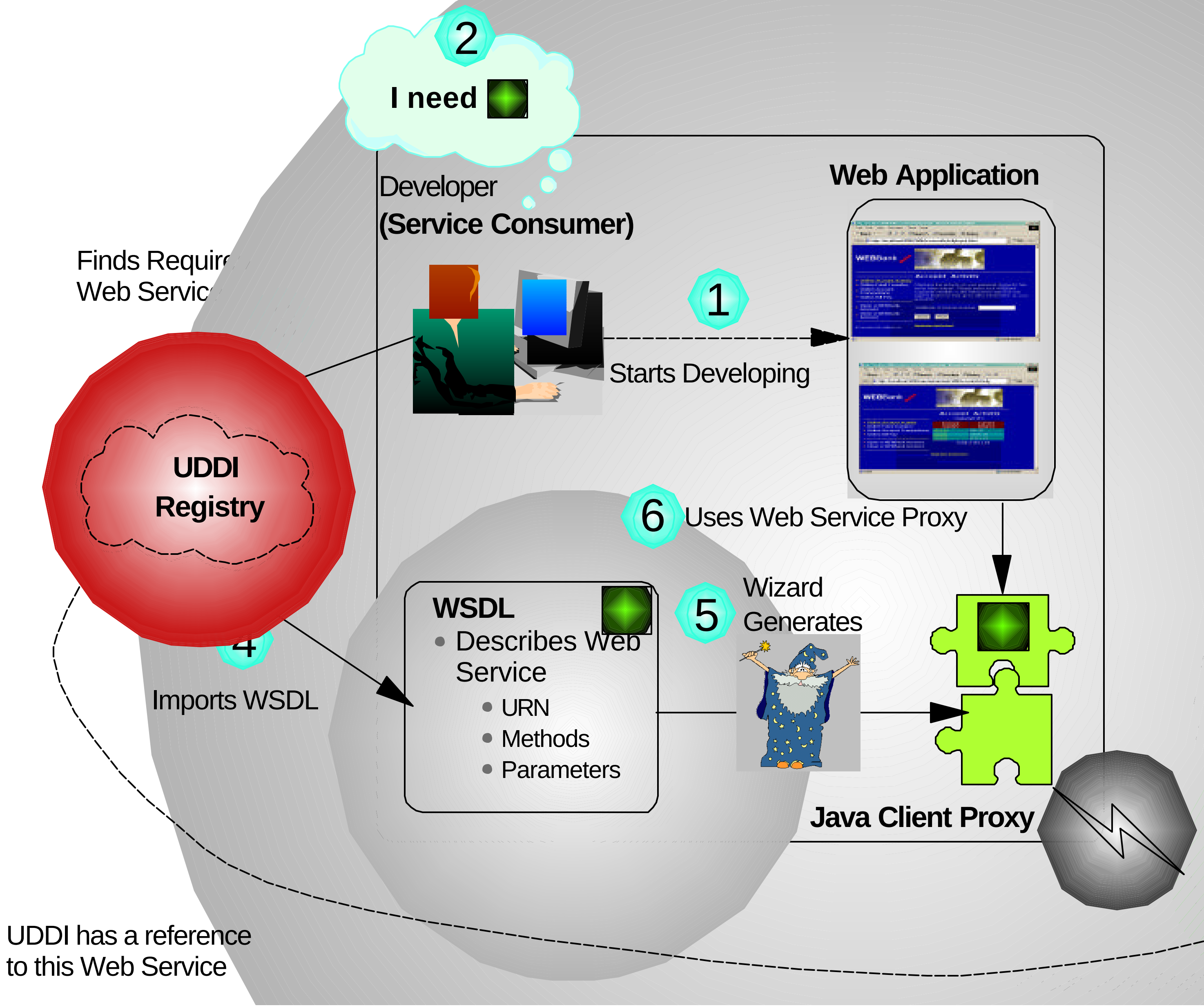
Add a new Service Type

Help Terms of Use Support

B ith IBM Tools



Using a Web S Tools



2

I need

Developer
(Service Consumer)

Web Application

1

Starts Developing

Finds Require
Web Servic

UDDI Registry

6

Uses Web Service Proxy

5

Wizard
Generates

WSDL

- Describes Web Service
  - URN
  - Methods
  - Parameters

4

Imports WSDL

Java Client Proxy

UDDI has a reference
to this Web Service

Web Services - Technical Perspective

1. **Technology is backed by key software vendors (IBM®, Microsoft®, Oracle,Ariba etc.) and Open Source organizations (Apache)**
   - ► **all cooperate to develop Web Services standards**
     - – **Web Services protocols continue to evolve rapidly**
     - – **Web Services are based on open protocols**
   - ► **software vendors compete in the tooling and runtimes arena**
     - – **high quality Web Services tooling is available**
       - • **developers can concentrate only on writing the business logic**
       - • **developers do not need to write the Web Services "plumbing" code**
     - – **most runtimes (Web Application Servers) support consumption and provision of Web Services**

2. **Implementation is de-coupled from interface**
   - ► **Web Services developers (produces and consumers) are not restricted to any particular language, operating system or object model**
     - – **a Web Service built with .net technology can be called from J2EE application**

Why will Web Services Technology Succeed?

- Other distributed technologies are not optimized for the Internet because they required strong coupling:
  - required the use of the same transport technologies
  - required knowledge of each others implementation
- What is wrong with other distributed technologies?
  - DCOM - requires Windows at each endpoint
  - CORBA - requires compatible ORBs at each endpoint
  - RMI - requires Java at each endpoint
- Why is XML/WSDL/SOAP/UDDI so much better?
  - implementation choices are de-coupled from interfaces
    - any language can be used to write and consume Web Services
    - more than one open standard transport technology can be specified
    - operating system differences are not a factor

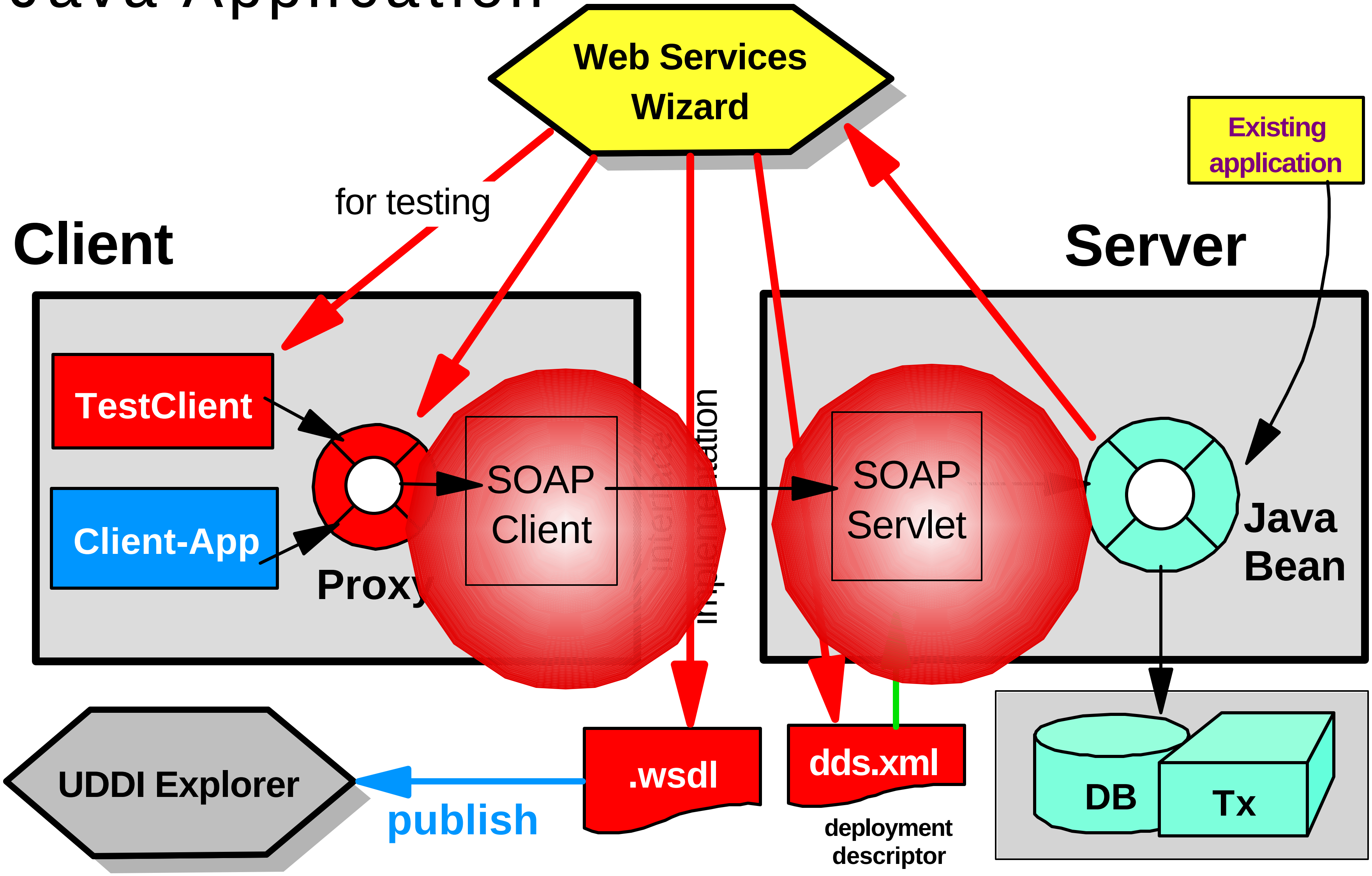
Complete IBM Software Portfolio for Web Services

WebSphere	**Web Services Runtime and Deployment** **Web Application Server 4.0. Includes SOAP Servlet and support for access to UDDI Registries.** **WebSphere Business Integrator. New software to manage web services workflow within and between enterprises.**
WebSphere Studio Family	**Development and Integration of Web Services Applications** **WebSphere Studio Application Developer IDE hosting a complete web application development environment. Includes major Web Services functions such as:** **▪ Web Service creation: JavaBeans, DADX, DB2 XML Extender, WSDL, URLs;** **▪ Test Client generation for Web Services test and verification** **▪ Web Services consumption: generate Java Proxy from WSDL** **▪ Full interface to UDDI Registries to browse, discover, download, publish.** **Futures: WebSphere Studio Enteprise Developer**

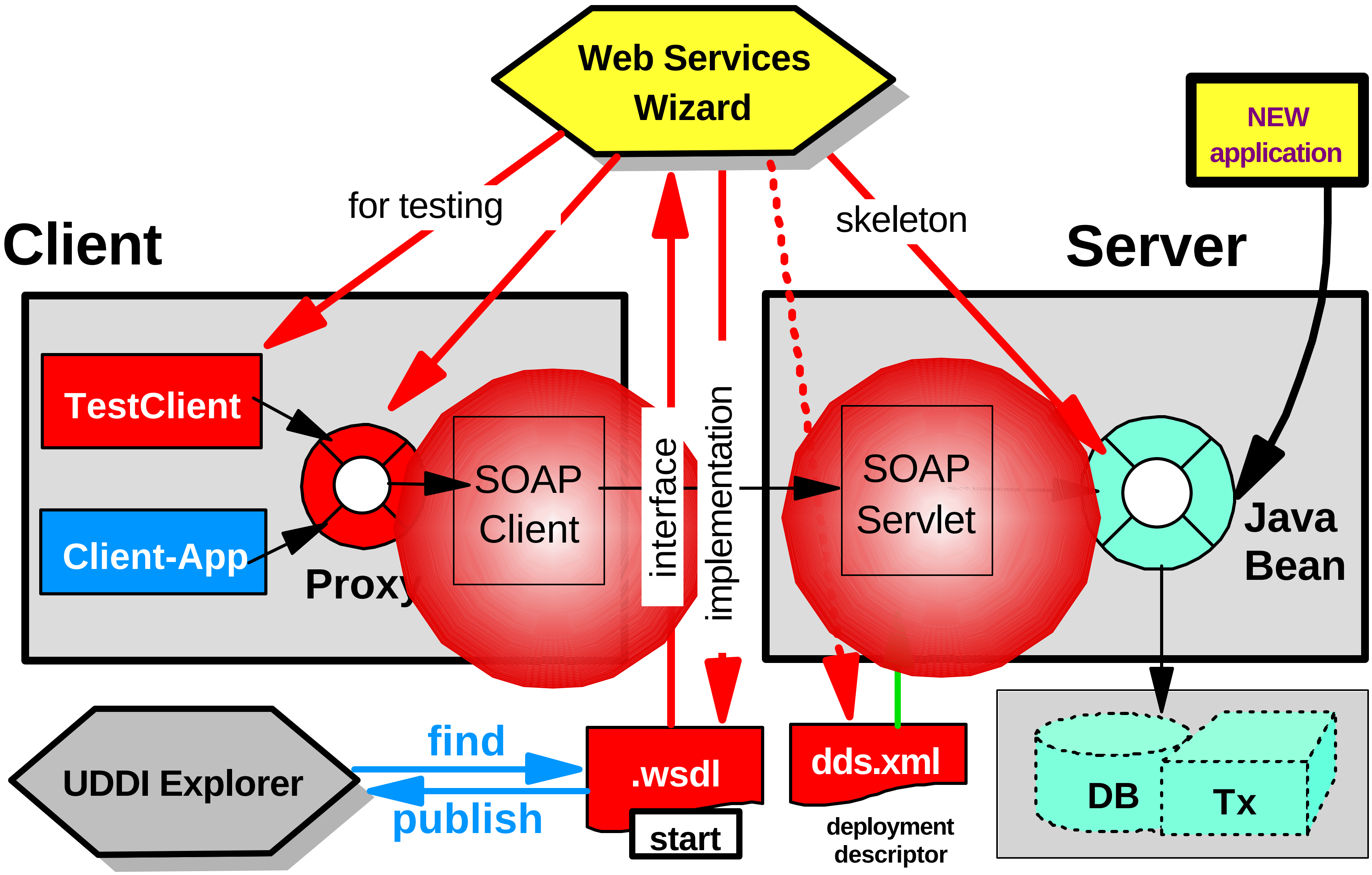
Complete IBM Software Portfolio for Web Services

DB2 UDB 7.2	**Manage Data in Web Services Environment** DB2 XML Extender. Stored procedures and SQL statements to be exposed as invokable Web Service operations. Enables Web services applications to access data stored in DB2 as an XML structured document.
Tivoli®	**Manage the IBM Middleware that Runs the Web Services** Tivoli Web Services Manager. Software for performance monitoring of Web Services. Tivoli Manager for WebSphere Application Server. Software providing a single point of control to manage Web Services. Tivoli SecureWay Policy Director. Software to build consistent security policy into Web Services.
Lotus®	**Web Services Enablement in Lotus Products** Lotus Web Services Enablement Kit. Software tools to build Web Services with Lotus products. Lotus Domino Server. Collaboration, workflow and messaging software that can be accessed by Web Services. Lotus Knowledge Discovery System. Knowledge management software to access expertise from Web Services.

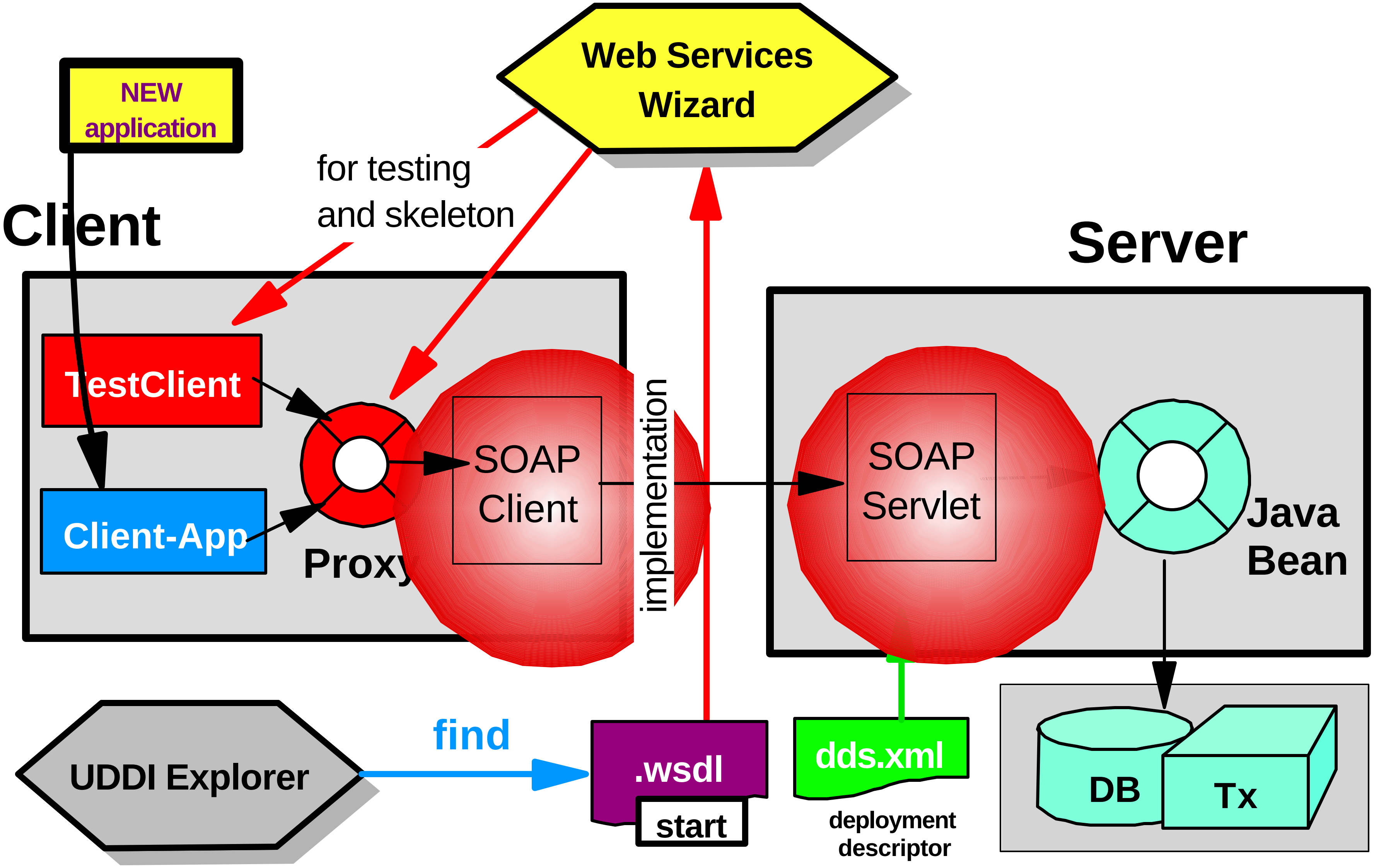
WSAD: Create Web Service from existing Java Application



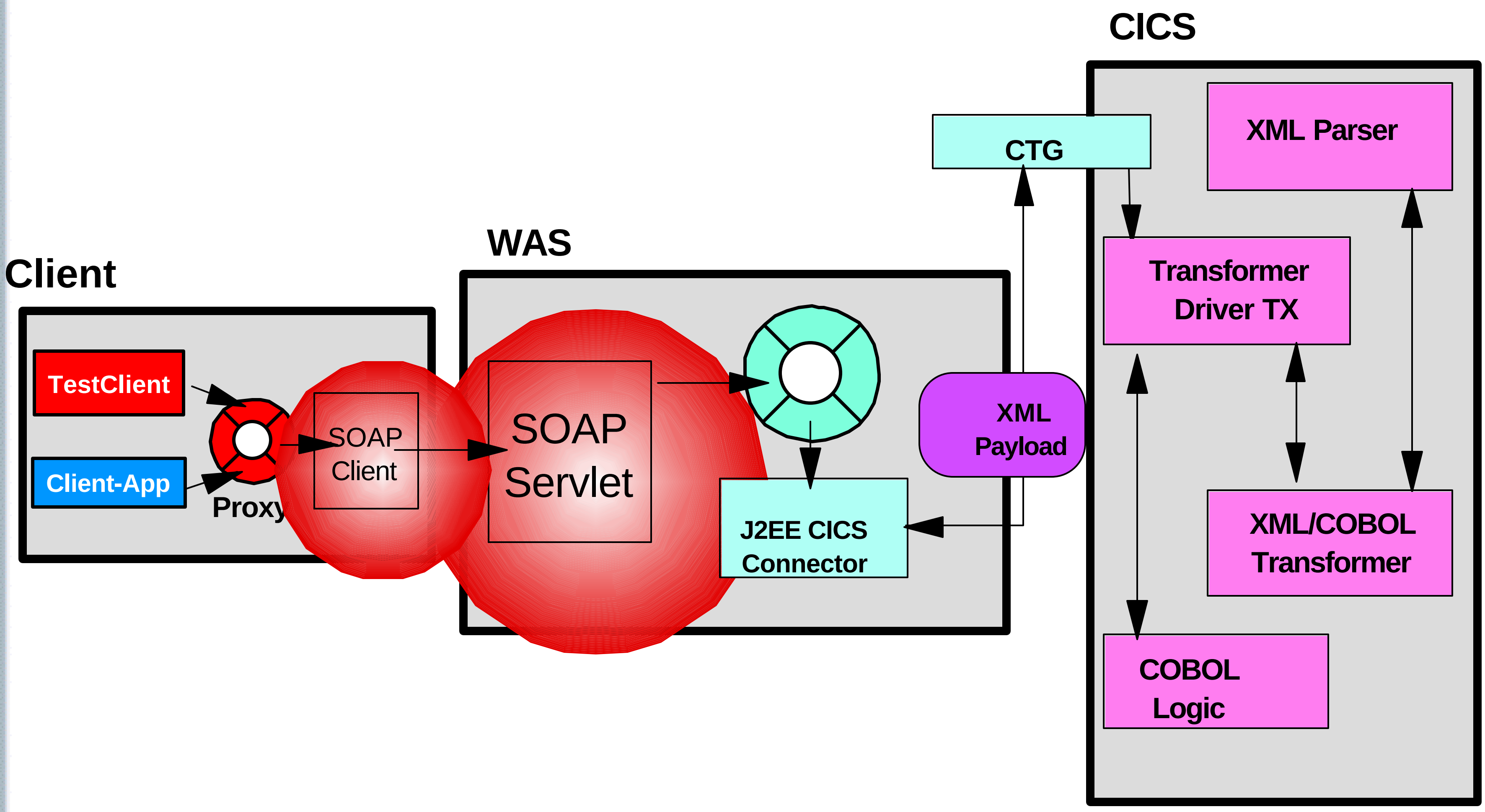
WSAD: Create Web Service from WSDL



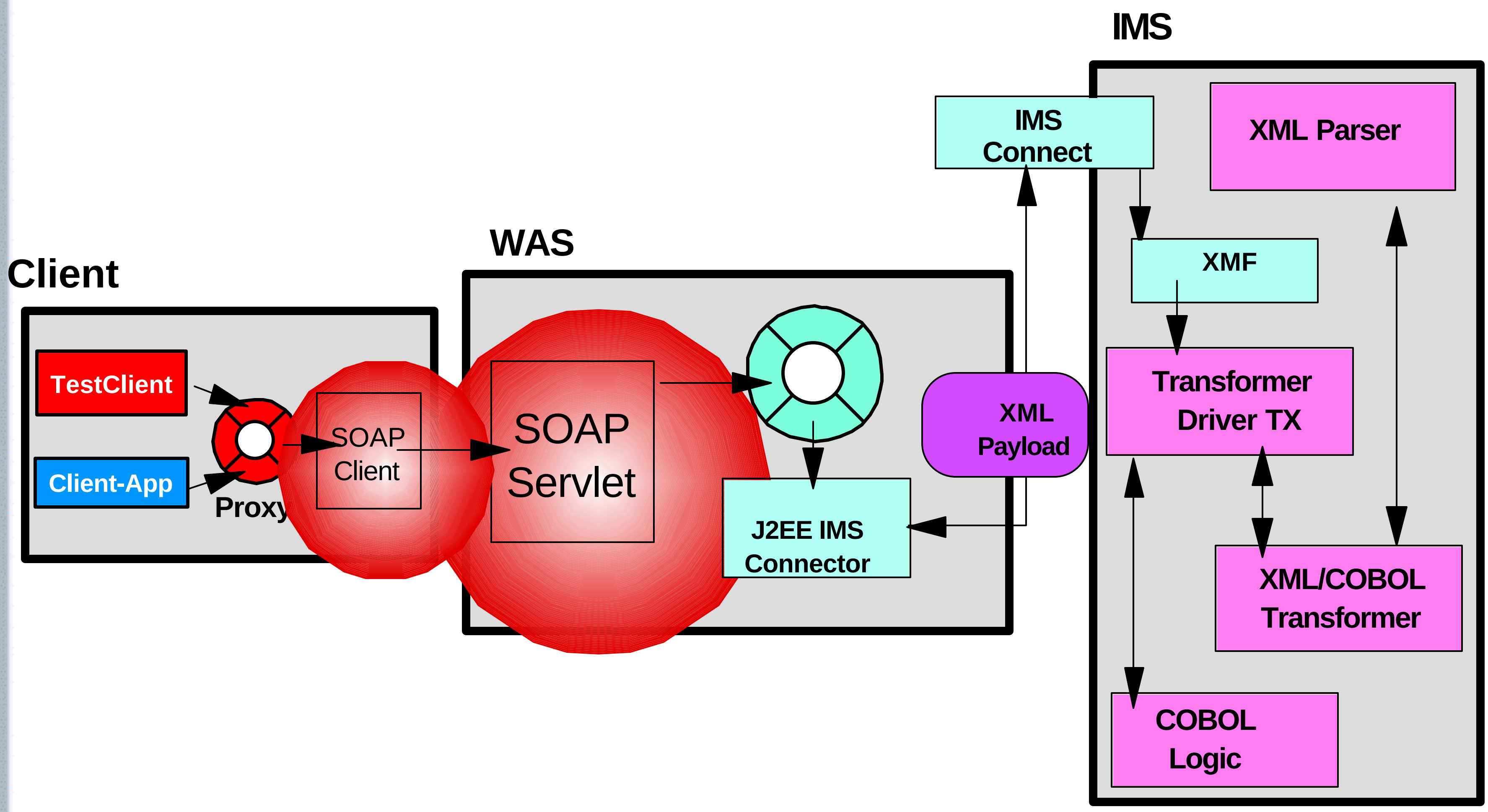
WSAD: Create Client from WSDL



Future directions: CICS Program as Web Service



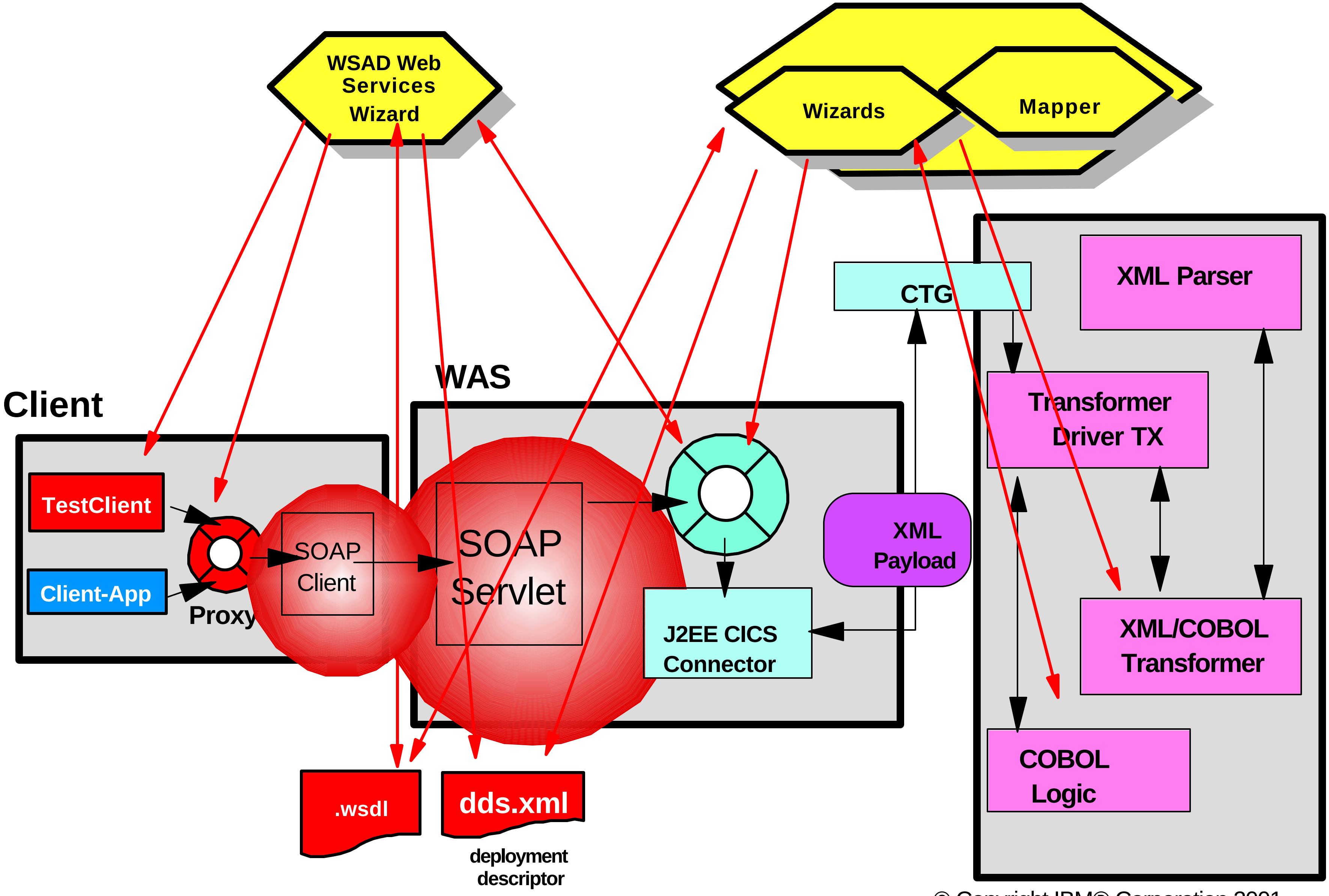
Future directions: IMS Program as Web Service



Directions for Web Services implementations in Enterprise tooling

- **Bottom Up: Existing COBOL/EGL, no Web Service**
  - ► **Facility for mapping COBOL/Enterprise Generation Language(EGL) to XML**
  - ► **Wizard for generation of all required WebSphere(WS) support artifacts**
- **Top Down: Existing Web Service**
  - ► **Facility for mapping XML to COBOL/EGL**
  - ► **Wizard for generation of all required WS support artifacts plus COBOL/EGL templates**
- **Meet in the middle: Existing WS, existing COBOL/EGL**
  - ► **Facility to map XML and COBOL/EGL**
  - ► **Wizard for generation of CICS/IMS WS support artifacts**

WSED: Web Services implementations



Concluding Remarks

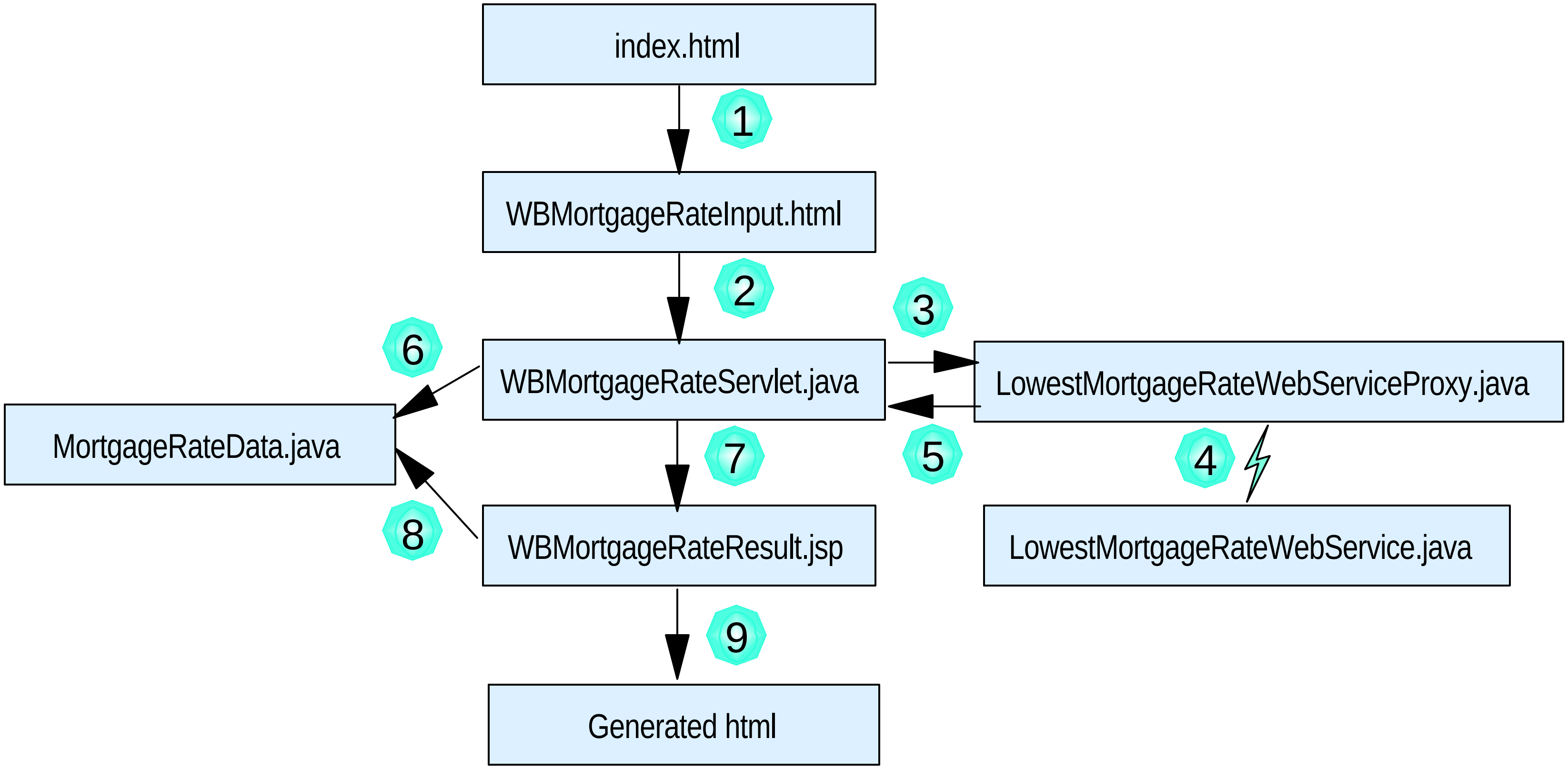
- **Web Services technology will enable businesses to:**
  - **deliver new IT solutions faster and at lower cost**
  - **protect their investment in IT legacy systems**
  - **externalize their business processes and integrate them with business processes of their customers and partners at a much lower cost**
  - **enter new markets and widen customer base**
- **Web Services technology is poised to become a big business for IBM:**
  - **Gartner: by 2004 the Web Services market will be worth $15.1 billion**
    - "WS is an incremental technology advance and has low entry costs , significant potential for cost savings from B2B and B2C integration"
- **IBM has the software to help customers with this "Web Services Revolution" !**

More Information...

developerWorks - Web Services Zone IBM sponsored site that contains articles, tutorials and latest news related to Web Services.	**www-106.ibm.com/developerworks/webservices**
AlphaWorks - Web Services Toolkit IBM sponsored site to provide early adopters access to emerging "alpha-code" technology. The Web Services Toolkit exploits new technologies that may be adopted in future releases of IBM Web Services Tools.	**www.alphaworks.ibm.com/tech/webservicestoolkit**
XMethods An organization dedicated to promoting the development, deployment, and use of web services.	**xmethods.com**
UDDI Organization UDDI Project and Community web site contains UDDI specification, whitepapers, FAQ. IBM is a member.	**www.uddi.org**
SOAP The latest version of SOAP specification hosted by the W3C Organization web site.	**www.w3.org/TR/SOAP**
WSDL The latest version of WSDL specification hosted by the W3C Organization web site.	**www.w3.org/TR/wsdl**

Backup

WBOnline Mortage Rate Web Service



Web Services Technology Values

- **Web services promote interpretability**
  - The interaction between a service provider and a service requester is designed to be completely platform and language independent. This interaction requires a WSDL document to define the interface and describe the service, along with a network protocol (usually HTTP). Because the service provider and the service requester have no idea what platforms or languages each other are using, interpretability is a given.
- **Web services enable just-in-time integration**
  - As service requesters use service brokers to find service providers, the discovery takes place dynamically. Once the requester and provider have found each other, the provider's WSDL document is used to bind the requester and the service together. This means that requesters, providers, and brokers work together to create systems that are self-configuring, adaptive, and robust.
- **Web services reduce complexity through encapsulation**
  - Service requesters and providers concern themselves with the interfaces necessary to interact with each other. As a result, a service requester has no idea how a service provider implements its service, and a service provider has no idea how a service requester uses its service. Those details are encapsulated inside the requesters and providers. That encapsulation is crucial for reducing complexity.
  - Web services give new life to legacy applications It's relatively straightforward to take an application, generate a SOAP wrapper, then generate a WSDL document to cast the application as a Web service. This means that legacy applications can be used in interesting new ways. In addition, the infrastructure associated with legacy applications (security, directory services, transactions, etc.) can be "wrapped" as a set of services as well.

Trademarks and Disclaimers

The following terms are trademarks or service marks of the IBM(R) Corporation in the United States or other countries or both: Tivoli(R), DB2(R), CICS(R),WebSphere(TM), MQSeries(R), and IMS(TM).

The following terms and phrases are trademarks or service marks of other companies:

Java and all Java-based trademarks and logos are trademarks or registered trademarks of Sun Microsystems, Inc. in the U.S.and other countries.

Microsoft is a registered trademarks of Microsoft Corporation.

Lotus is a registered trademark of Lotus Corporation.

Other company, product, and services names may be trademarks or service marks of others.

This presentation is provided by IBM, its Affiliates and Suppliers "as is" without warranty of any kind, either express or implied, statutory or otherwise, including, without limitation, the implied warranties of merchantability, fitness for a particular purpose, or non-infringement. Customer experiences may be different from those described here. This presentation discusses future plans which are subject to change.