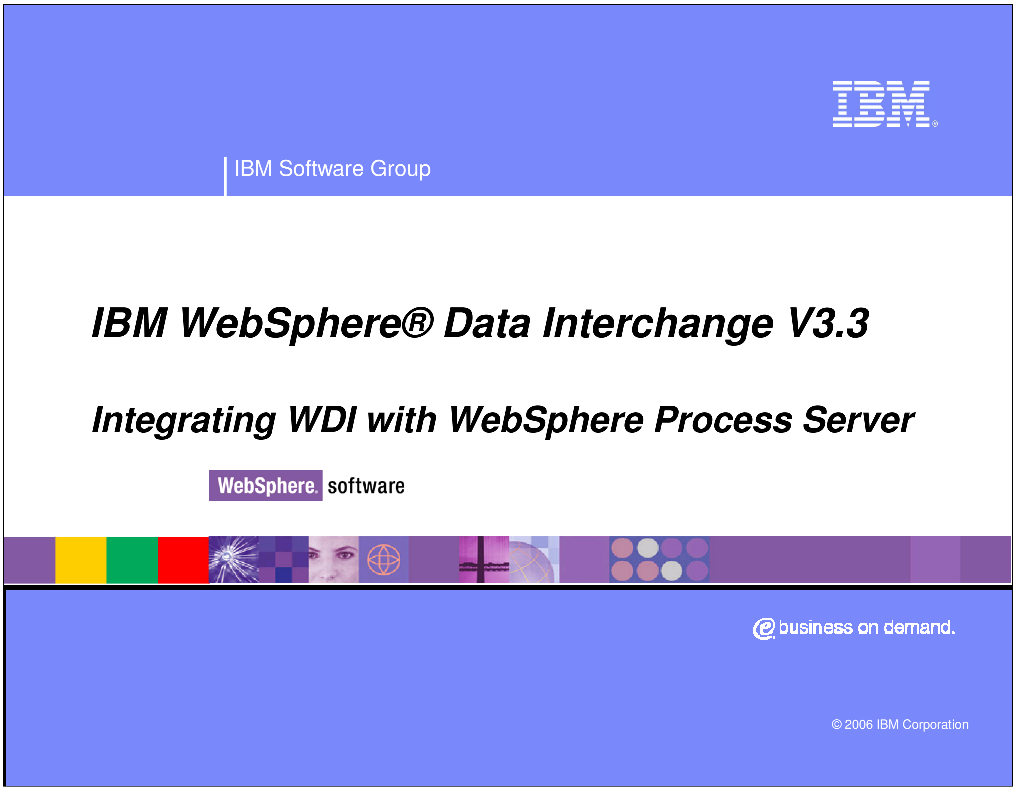


IBM Software Group

IBM WebSphere® Data Interchange V3.3

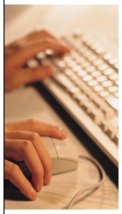
Integrating WDI with WebSphere Process Server

WebSphere. software

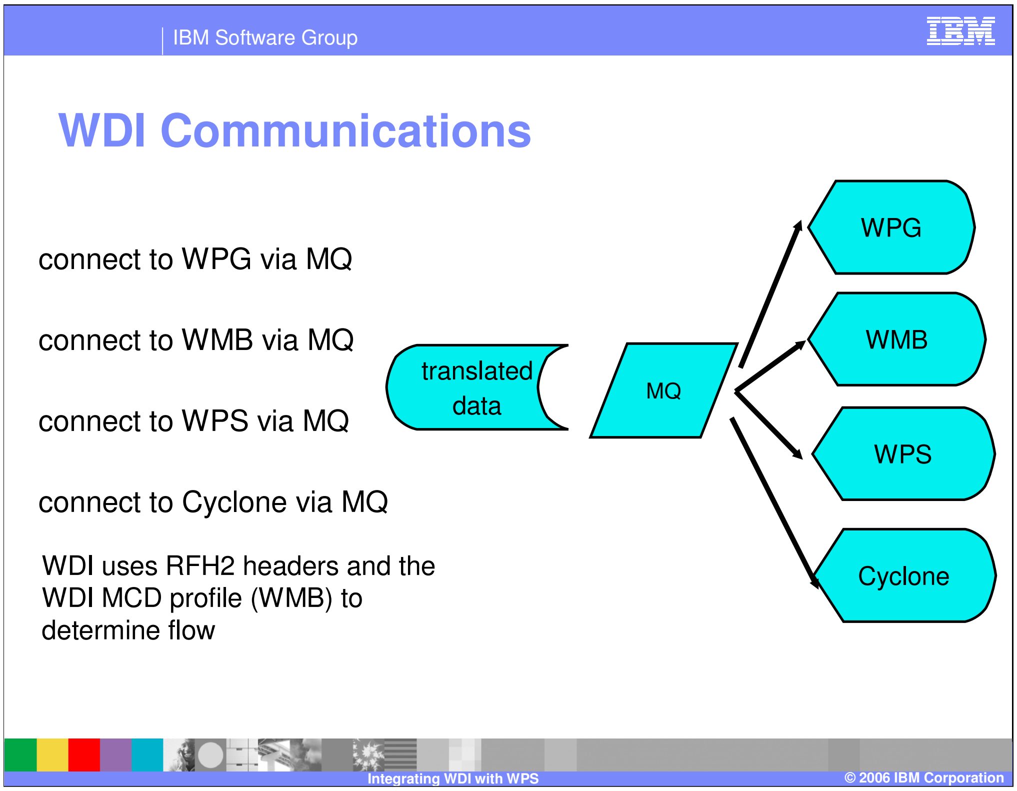
e business on demand.

This presentation shows how WDI could be integrated with WebSphere Process Server.

Agenda



- Introduction to SOA
- Service Component Architecture Programming Model
- Component Layering
- Integration Flow between WDI and WPS
- WDI and SOA
- Demo
- Summary



WDI Communications

connect to WPG via MQ

connect to WMB via MQ

connect to WPS via MQ

connect to Cyclone via MQ

WDI uses RFH2 headers and the WDI MCD profile (WMB) to determine flow

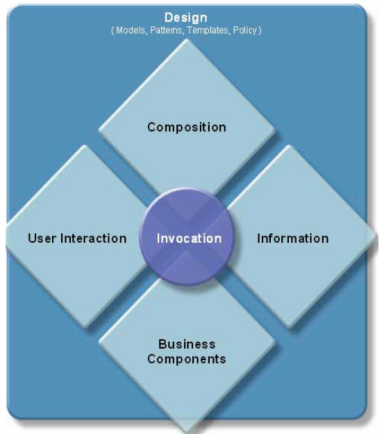
translated data
MQ
WPG
WMB
WPS
Cyclone

WDI can integrate with a number of products. WebSphere Process Server is one. WMQ is the message transport for message data.

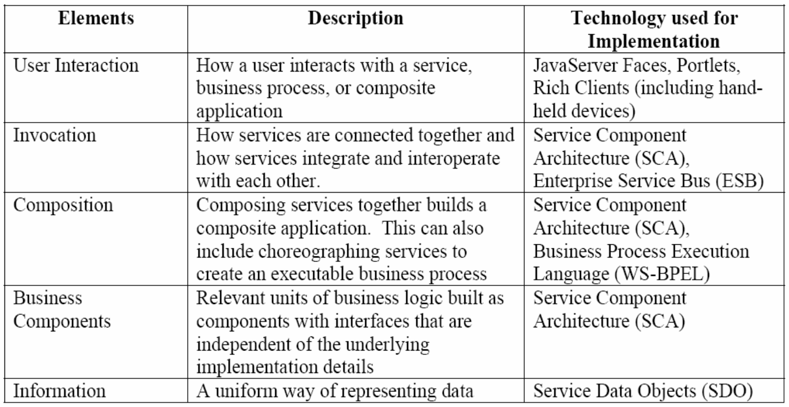
Introduction to SOA

- SOA is a framework that combines
- individual business functions and processes, called services, to implement sophisticated
- business applications and processes. SOA is an approach to IT that considers business
- processes as reusable components or services which are **loosely-coupled** and that are
- platform and implementation **neutral**. The approach allows you to design solutions as
- assemblies of services in which the assembly description is a managed, **well-defined**
- first-class aspect of the solution, and hence, amenable to analysis, change, and evolution.
- The solution can then be viewed as a choreographed set of service interactions.

SOA Programming Model Elements

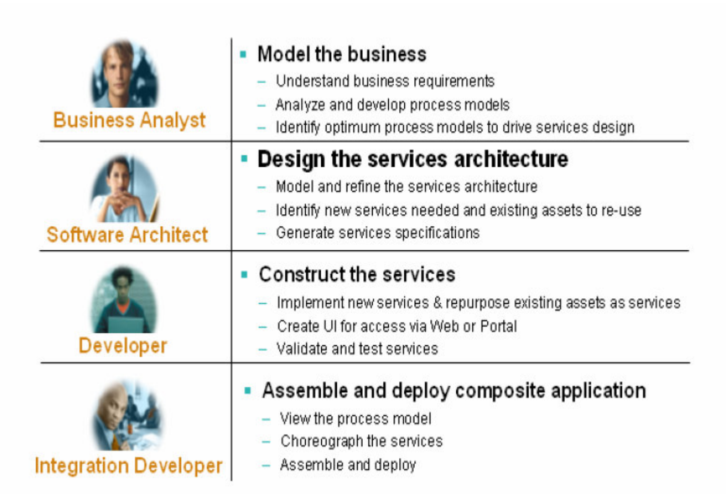


SOA Programming Model Elements



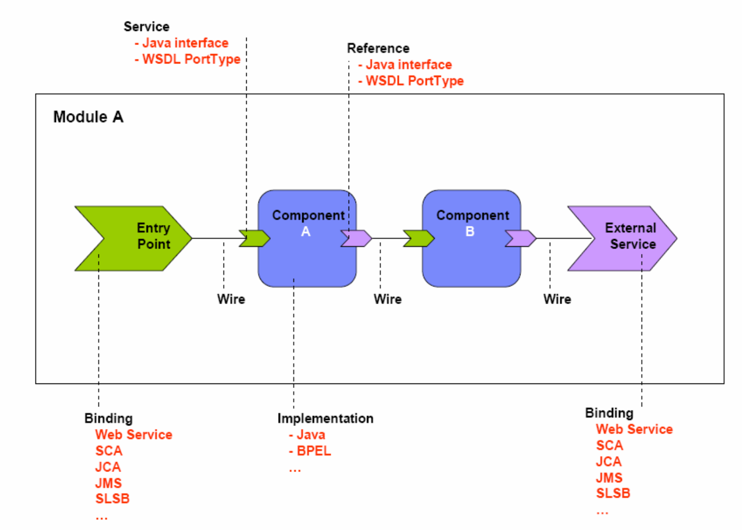
Elements	Description	Technology used for Implementation
User Interaction	How a user interacts with a service, business process, or composite application	JavaServer Faces, Portlets, Rich Clients (including hand-held devices)
Invocation	How services are connected together and how services integrate and interoperate with each other.	Service Component Architecture (SCA), Enterprise Service Bus (ESB)
Composition	Composing services together builds a composite application. This can also include choreographing services to create an executable business process	Service Component Architecture (SCA), Business Process Execution Language (WS-BPEL)
Business Components	Relevant units of business logic built as components with interfaces that are independent of the underlying implementation details	Service Component Architecture (SCA)
Information	A uniform way of representing data	Service Data Objects (SDO)

Key Roles in Service Oriented Design/Dev

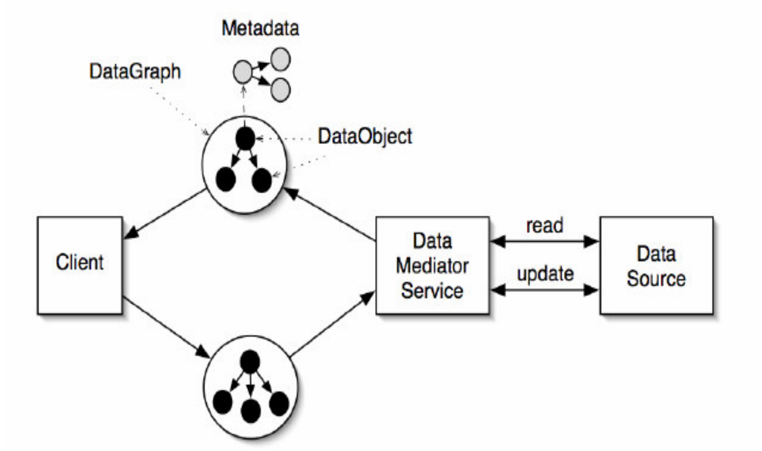


Business Analyst	▪ **Model the business** – Understand business requirements – Analyze and develop process models – Identify optimum process models to drive services design
Software Architect	▪ **Design the services architecture** – Model and refine the services architecture – Identify new services needed and existing assets to re-use – Generate services specifications
Developer	▪ **Construct the services** – Implement new services & repurpose existing assets as services – Create UI for access via Web or Portal – Validate and test services
Integration Developer	▪ **Assemble and deploy composite application** – View the process model – Choreograph the services – Assemble and deploy

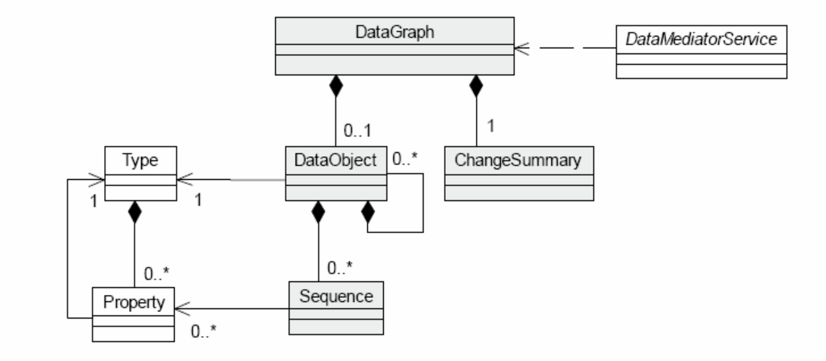
A Simple Module



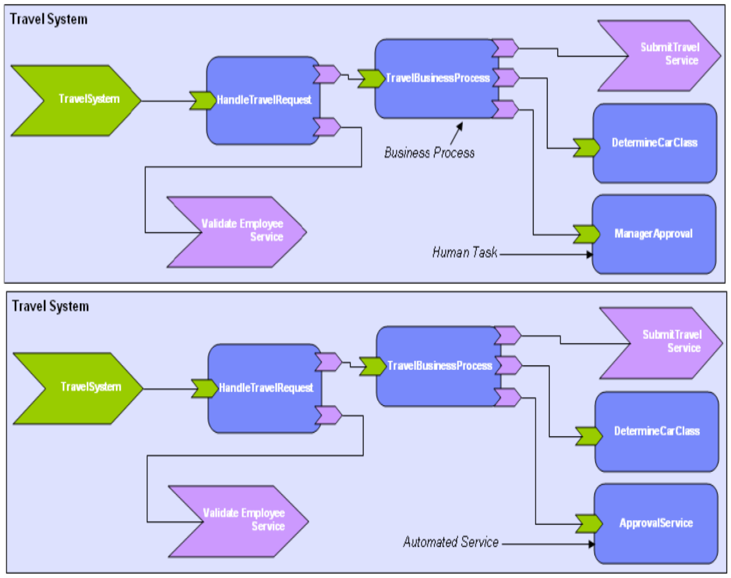
Components of an SDO Solution



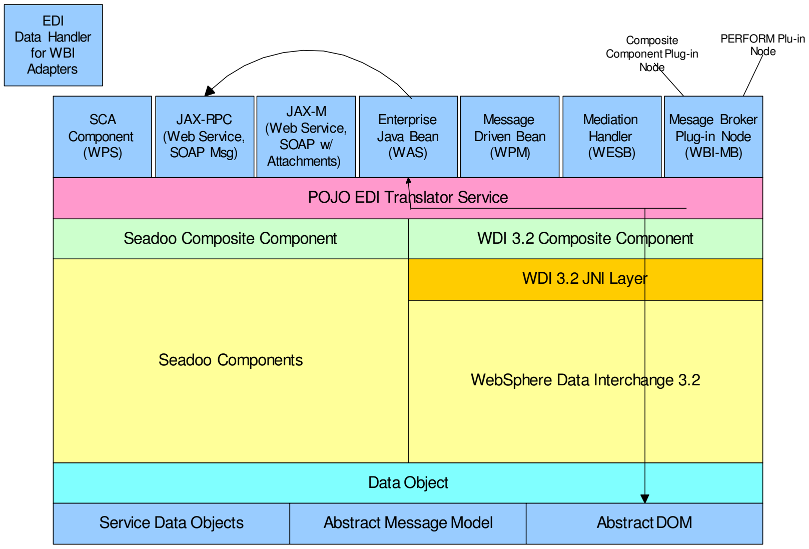
UML Model of Core SDO Components



Example of rewiring an assembly



Component Layering



EDI Data Handler for WBI Adapters

Composite Component Plug-in Node

PERFORM Plu-in Node

SCA Component (WPS)

JAX-RPC (Web Service, SOAP Msg)

JAX-M (Web Service, SOAP w/ Attachments)

Enterprise Java Bean (WAS)

Message Driven Bean (WPM)

Mediation Handler (WESB)

Mesage Broker Plug-in Node (WBI-MB)

POJO EDI Translator Service

Seadoo Composite Component

WDI 3.2 Composite Component

WDI 3.2 JNI Layer

Seadoo Components

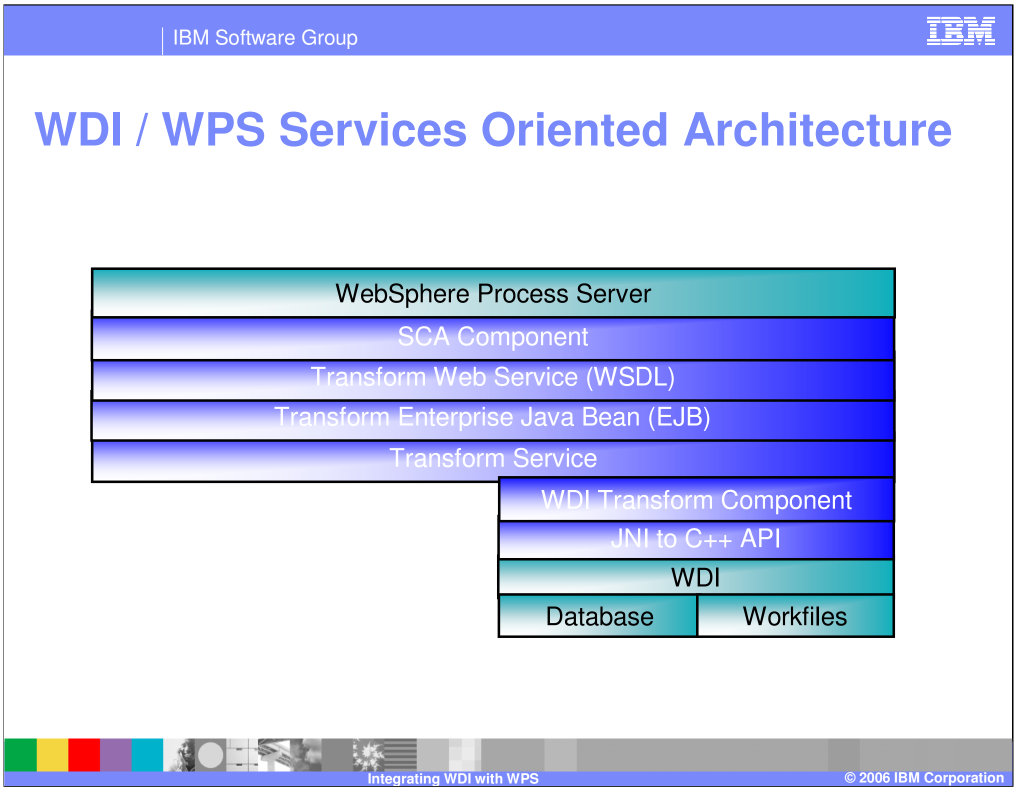
WebSphere Data Interchange 3.2

Data Object

Service Data Objects

Abstract Message Model

Abstract DOM



WDI / WPS Services Oriented Architecture

WebSphere Process Server

SCA Component

Transform Web Service (WSDL)

Transform Enterprise Java Bean (EJB)

Transform Service

WDI Transform Component

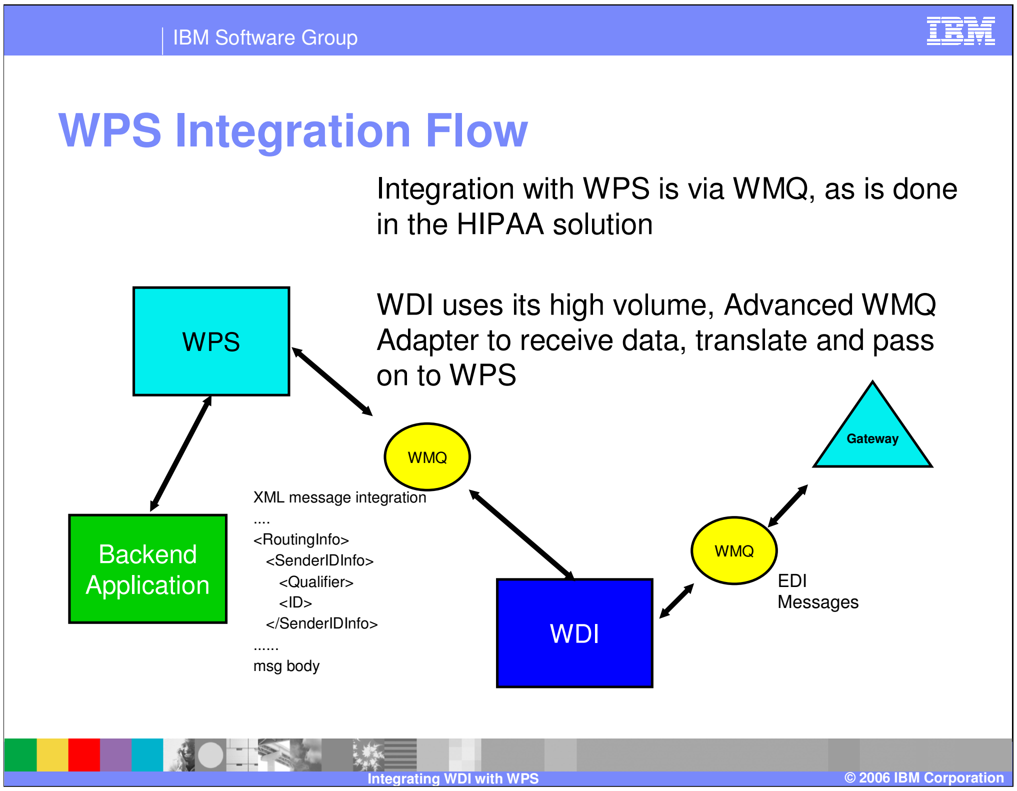
JNI to C++ API

WDI

Database

Workfiles

Services Oriented Architecture has layers of integration. The Service Component Architecture component, web service, Enterprise Java Bean, and Transform Service are the general means of Integrating WPS to a service. WDI provides a JNI to C++ API and a sample Transform component to help integrate into the WDI product. The WDI product usess database and workfile access during translation or transformation.



WPS Integration Flow

Integration with WPS is via WMQ, as is done in the HIPAA solution

WDI uses its high volume, Advanced WMQ Adapter to receive data, translate and pass on to WPS

WPS

Backend Application

WMQ

XML message integration

```
....
<RoutingInfo>
  <SenderIDInfo>
    <Qualifier>
    <ID>
  </SenderIDInfo>
......
msg body
```

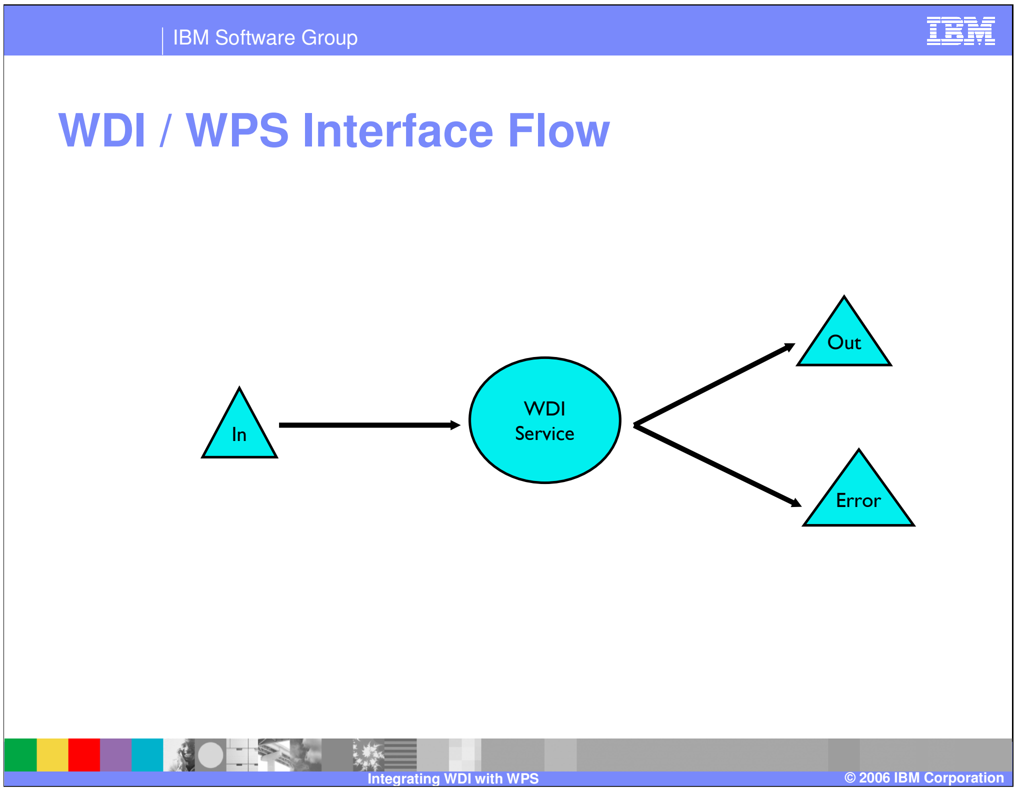
WDI

WMQ

EDI Messages

Gateway

In a typical integration flow, a customer's backend application might provide data for a WPS flow. WPS would use place message data on a WMQ queue. The WMQ trigger mechanism would start WDI's Advanced Adapter and invoke WDI. WDI would then translate the data and place it on a WMQ queue which is monitored by WPS or a gateway, like WPG.



WDI / WPS Interface Flow

In

WDI Service

Out

Error

A typical WPS flow is simple. The input message interacts with a service. In this case the components of the prior page are combined into something labeld the “WDI Service”. This service then produces output and errors.

Sample WSDL

```
<?xml version="1.0" encoding="UTF-8"?>
<wsdl:definitions xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
xmlns:tns="http://ibm.com/TransformService/"
xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema" name="TransformService"
targetNamespace="http://ibm.com/TransformService/">
<wsdl:types>
<xsd:schema targetNamespace="http://ibm.com/TransformService/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">
<xsd:simpleType name="tSyntax">
  <xsd:restriction base="xsd:string">
      <xsd:enumeration value="xml"/>
      <xsd:enumeration value="edi"/>
      <xsd:enumeration value="rod"/>
  </xsd:restriction>
</xsd:simpleType>
<xsd:simpleType name="tDictionary">
  <xsd:restriction base="xsd:string">
  </xsd:restriction>
</xsd:simpleType>
<xsd:simpleType name="tDocument">
  <xsd:restriction base="xsd:string">
  </xsd:restriction>
</xsd:simpleType>
<xsd:complexType name="tProperties">
   <xsd:choice>
      <xsd:element name="Syntax" type="tns:tSyntax" minOccurs="1" maxOccurs="1"/>
      <xsd:element name="Dictionary" type="tns:tDictionary" minOccurs="0" maxOccurs="1"/>
      <xsd:element name="Document" type="tns:tDocument" minOccurs="0" maxOccurs="1"/>
   </xsd:choice>
</xsd:complexType>
<xsd:simpleType name="tMessage">
  <xsd:restriction base="xsd:string">
  </xsd:restriction>
</xsd:simpleType>
```

Sample WSDL

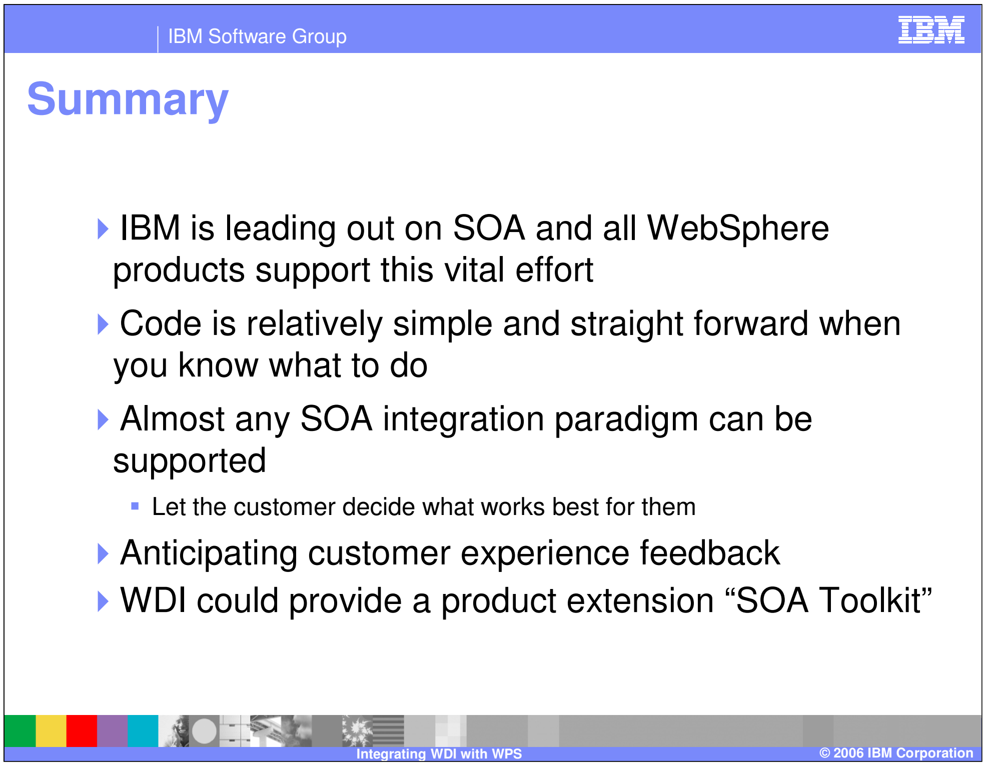
```
<xsd:complexType name="tMsgAssembly">
   <xsd:sequence>
      <xsd:element name="Properties" type="tns:tProperties"/>
      <xsd:element name="Message" type="tns:tMessage"/>
   </xsd:sequence>
</xsd:complexType>
<xsd:simpleType name="tFileURI">
  <xsd:restriction base="xsd:anyURI">
     <xsd:pattern value="file:.*"/>
  </xsd:restriction>
</xsd:simpleType>
<xsd:complexType name="tURIAssembly">
   <xsd:sequence>
      <xsd:element name="Properties" type="tns:tProperties"/>
      <xsd:element name="InputURI" type="tns:tFileURI"/>
      <xsd:element name="OutputURI" type="tns:tFileURI"/>
   </xsd:sequence>
</xsd:complexType>
<xsd:complexType name="tTransformFault">
<xsd:sequence>
<xsd:element name="ErrorCode" type="xsd:string"/>
<xsd:element name="ErrorMsg" type="xsd:string"/>
<xsd:element name="MessageID" type="xsd:string"/>
<xsd:element name="CorrelationID" type="xsd:string"/>
</xsd:sequence>
</xsd:complexType>
<xsd:element name="MessageAssembly" type="tns:tMsgAssembly"/>
<xsd:element name="TransformMsgResp" type="xsd:string"/>
<xsd:element name="TransformURIResp" type="xsd:string"/>
<xsd:element name="URIAssembly" type="tns:tURIAssembly"/>
<xsd:element name="TransformMsgFault" type="tns:tTransformFault"/>
<xsd:element name="TransformURIFault" type="tns:tTransformFault"/>
</xsd:schema>
</wsdl:types>

<wsdl:message name="TransformMsgResp">
<wsdl:part name="TransformMsgResp" element="tns:TransformMsgResp" />
</wsdl:message>
```

Example – SOAP Request

- <?xml version="1.0" encoding="UTF-8" ?>
- - <SOAP-ENV:Envelope xmlns:SOAP-ENV="**http://schemas.xmlsoap.org/soap/envelope/**" xmlns:q0="**http://ibm.com/TransformService/**" xmlns:xsd="**http://www.w3.org/2001/XMLSchema**" xmlns:xsi="**http://www.w3.org/2001/XMLSchema-instance**">
- - <SOAP-ENV:Body>
- - <q0:MessageAssembly>
- - <Properties>
- <Syntax>**xml**</Syntax>
- </Properties>
- <Message>**<OrderSR><Header typecode="00"><PONum>PO12345678901234</PONum><PODate>03232001</PODate><Sender><Id>OfTheBeast</Id><Qualifier>ST</Qualifier></Sender><Receiver><Id>Lewitt</Id><Qualifier>BT</Qualifier></Receiver></Header><DetailLoop><ItemNumber>89988760964</ItemNumber><SubDetail><Description>LEG OF LAMB</Description><Quantity>1.00</Quantity><UnitPrice>5.01</UnitPrice></SubDetail></DetailLoop><Trailer><ItemCount>6</ItemCount><TotalBucks>1304.55</TotalBucks></Trailer></OrderSR>**</Message>
- </q0:MessageAssembly>
- </SOAP-ENV:Body>
- </SOAP-ENV:Envelope>

Example – SOAP Response Message

- <soapenv:Envelope xmlns:soapenc="**http://schemas.xmlsoap.org/soap/encoding/**" xmlns:soapenv="**http://schemas.xmlsoap.org/soap/envelope/**" xmlns:xsd="**http://www.w3.org/2001/XMLSchema**" xmlns:xsi="**http://www.w3.org/2001/XMLSchema-instance**">
- <soapenv:Header />
- - <soapenv:Body>
- <p412:TransformMsgResp xmlns:p412="**http://ibm.com/TransformService/**">**ISA*00* *00* *ST*OFTHEBEAST *BT*LEWITT *060110*1049*U*00401*000000003*0*P*:! GS*PO* * *20060110*1049*3*X*004010! ST*850*0003! BEG*00*NE*PO12345678901234**03232001! N1*ST*OfTheBeast! N1*BT*Lewitt! PO1*1*****BP*89988760964! PO3*ZZ***5.01*FX*1*YY*LEG OF LAMB! CTT*6*1304.55! SE*8*0003! GE*1*3! IEA*1*000000003!**</p412:TransformMsgResp>
- </soapenv:Body>
- </soapenv:Envelope>



Summary

- IBM is leading out on SOA and all WebSphere products support this vital effort
- Code is relatively simple and straight forward when you know what to do
- Almost any SOA integration paradigm can be supported
  - Let the customer decide what works best for them
- Anticipating customer experience feedback
- WDI could provide a product extension “SOA Toolkit”

Trademarks, copyrights, and disclaimers

The following terms are trademarks or registered trademarks of International Business Machines Corporation in the United States, other countries, or both:

IBM
IBM(logo)
e(logo)business
AIX
CICS
Cloudscape
DB2
DB2 Universal Database
IMS
Informix
iSeries
Lotus
MQSeries
OS/390
OS/400
pSeries
Tivoli
WebSphere
xSeries
zSeries

Java and all Java-based trademarks are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.

Microsoft, Windows, Windows NT, and the Windows logo are registered trademarks of Microsoft Corporation in the United States, other countries, or both.

Intel, ActionMedia, LANDesk, MMX, Pentium and ProShare are trademarks of Intel Corporation in the United States, other countries, or both.

UNIX is a registered trademark of The Open Group in the United States and other countries.

Linux is a registered trademark of Linus Torvalds.

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

Product data has been reviewed for accuracy as of the date of initial publication. Product data is subject to change without notice. This document could include technical inaccuracies or typographical errors. IBM may make improvements and/or changes in the product(s) and/or program(s) described herein at any time without notice. Any statements regarding IBM's future direction and intent are subject to change or withdrawal without notice, and represent goals and objectives only. References in this document to IBM products, programs, or services does not imply that IBM intends to make such products, programs or services available in all countries in which IBM operates or does business. Any reference to an IBM Program Product in this document is not intended to state or imply that only that program product may be used. Any functionally equivalent program, that does not infringe IBM's intellectual property rights, may be used instead.

Information is provided "AS IS" without warranty of any kind. THE INFORMATION PROVIDED IN THIS DOCUMENT IS DISTRIBUTED "AS IS" WITHOUT ANY WARRANTY, EITHER EXPRESS OR IMPLIED. IBM EXPRESSLY DISCLAIMS ANY WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NONINFRINGEMENT. IBM shall have no responsibility to update this information. IBM products are warranted, if at all, according to the terms and conditions of the agreements (e.g., IBM Customer Agreement, Statement of Limited Warranty, International Program License Agreement, etc.) under which they are provided. Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products in connection with this publication and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. IBM makes no representations or warranties, express or implied, regarding non-IBM products and services.

The provision of the information contained herein is not intended to, and does not, grant any right or license under any IBM patents or copyrights. Inquiries regarding patent or copyright licenses should be made, in writing, to:

IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785
U.S.A.

Performance is based on measurements and projections using standard IBM benchmarks in a controlled environment. All customer examples described are presented as illustrations of how those customers have used IBM products and the results they may have achieved. The actual throughput or performance that any user will experience will vary depending upon considerations such as the amount of multiprogramming in the user's job stream, the I/O configuration, the storage configuration, and the workload processed. Therefore, no assurance can be given that an individual user will achieve throughput or performance improvements equivalent to the ratios stated here.

Note to U.S. Government Users - Documentation related to restricted rights-Use, duplication or disclosure is subject to restrictions set forth in GSA ADP Schedule Contract and IBM Corp.