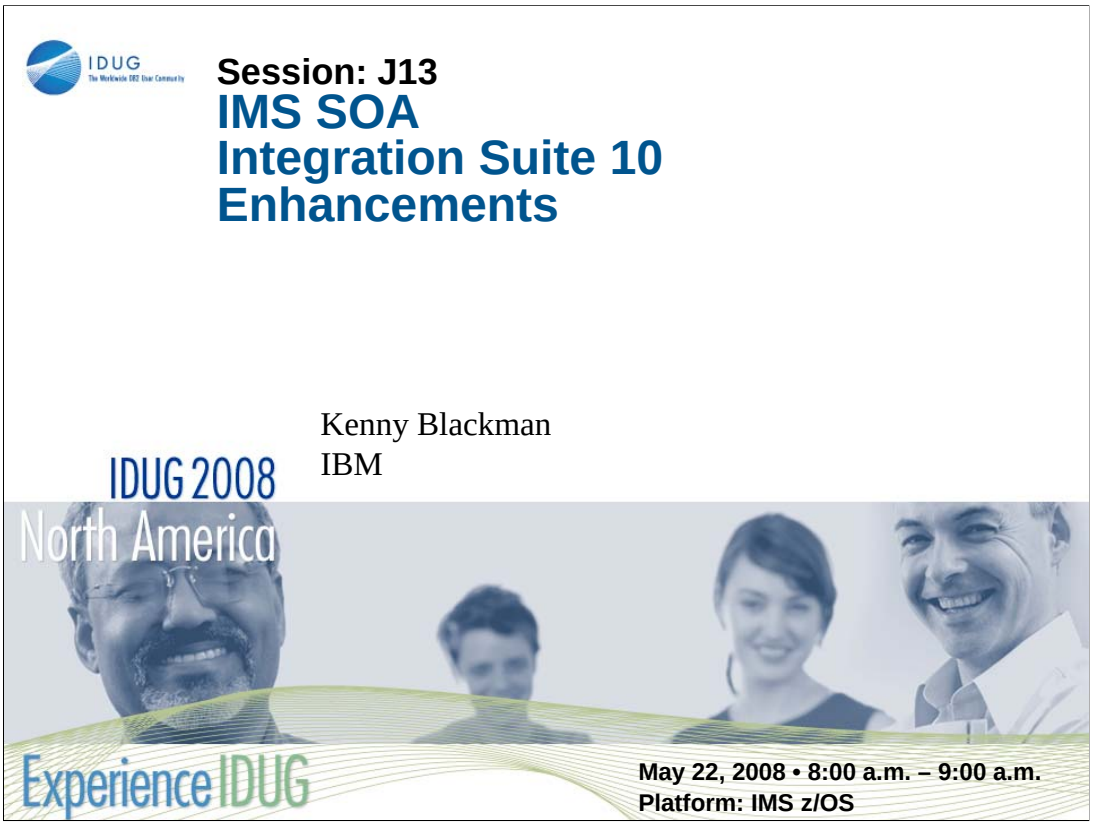


This presentation will discuss the IMS SOA Integration Suite with a focus on how IMS customers can leverage their existing IMS applications and data for an SOA solution. It describes the IBM development tools and the IMS integration technologies that can be used to access IMS TM application programs and databases. This presentation will also describe the IMS V10 support that enables IMS TM applications access to Web Services Development and Connectivity.

Agenda

- SOA review
- What is IMS
- What are the SOA patterns for IMS
- What is OTMA and ODBA
- What is the IMS SOA Integration Suite

IDUG 2008 North America 2

Gain an understanding of types of SOA integration models

Objective 2: Learn how to apply IMS Integration Suite to support SOA initiatives

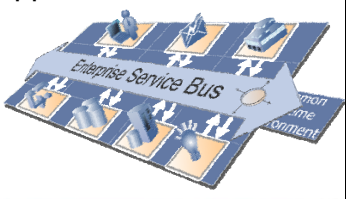
Objective 3: Acquire understanding of IMS base functions to support the Integration Suite

Objective 4: Discover how IMS applications can access services

Objective 5: Have a better understanding of how to integrate with IMS applications and data

Service Oriented Architecture (SOA)

- A *Service Oriented Architecture* enables flexible connectivity of applications or resources by
  - Representing every application or resource as a service with a standardized interface
    enabling them to exchange
    Structured information (messages,
    documents, 'business objects')
  - mediating the message exchange through a service integration bus
  - providing on-ramps to the bus for existing application environments

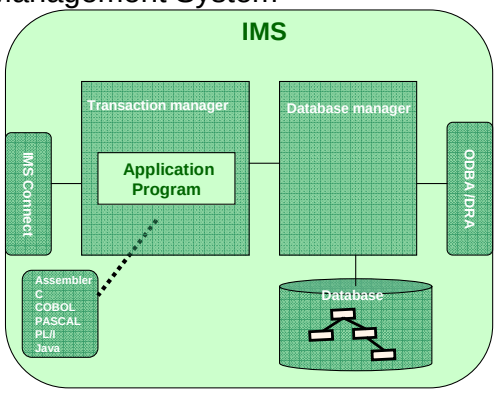


IDUG 2008 North America 3

The goal for a Service Oriented Architecture (SOA) is a world wide mesh of collaborating services, which are published and available for invocation on the Service Bus. Adopting SOA is essential to deliver the business agility and IT flexibility promised by Web Services. These benefits are delivered not by just viewing service architecture from a technology perspective and the adoption of Web Service protocols, but require the creation of a Service Oriented Environment that is based on specific key principals.

What is IMS?

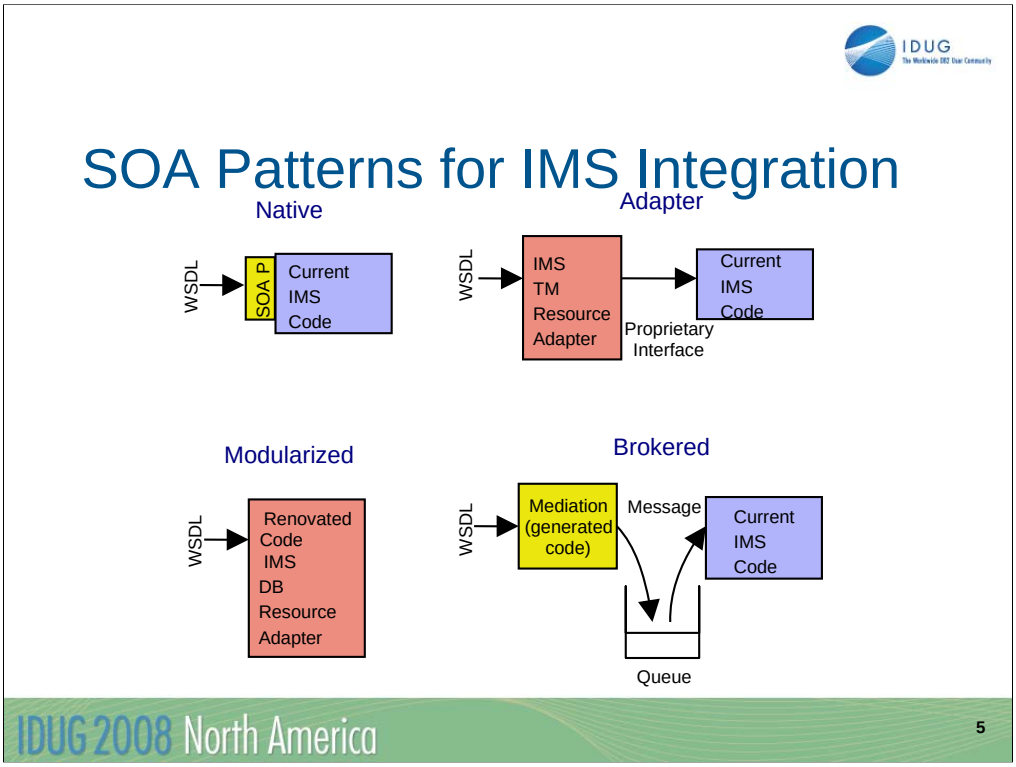
- Transaction Management System
- Multilingual Application Program server
- Database Management System



Services accessed without knowledge of underlying implementation

Services written in C# running on .Net platforms and services written in Java running on Java EE platforms can both be consumed by a common composite application

According to IBM's own estimates, SOA has a market opportunity of $65 billion this year. Gartner says that by 2008, SOA will provide the basis for 80 percent of new IT transformation projects.



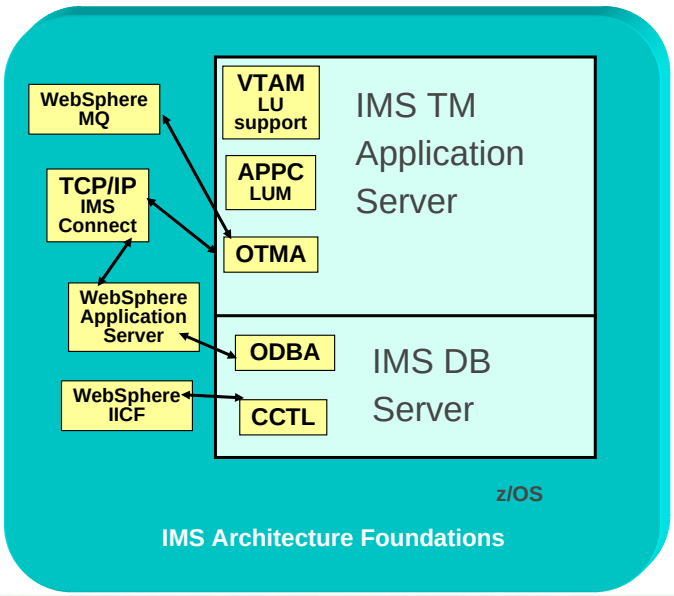
•Business Application Services

- •Native pattern is used when the existing code is already well modularized and technology is available to put a native web services interface in front of the native interface.
- •When the existing code does not expose the services needed in any accessible fashion, the code must be renovated to restructure the access to the business rules and data appropriately. This pattern provides the opportunity to directly expose the service.

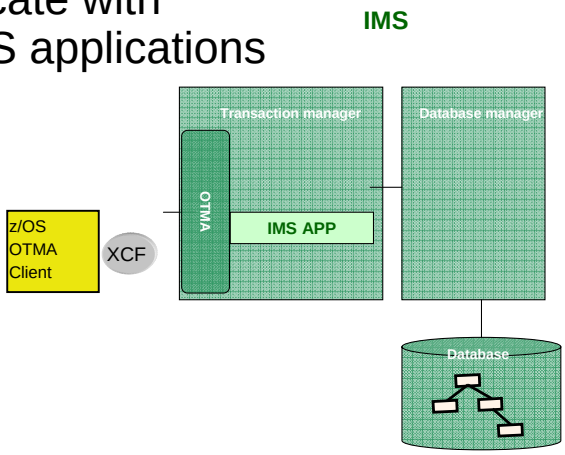
•Access Services

- •Adapter pattern is used to expose existing code with the changes necessary to expose the desired services made primarily in the custom adapter with modest changes to the existing code.
- •Brokered pattern is used when the existing code exposes the desired services as messages when this can be done with modest changes to the existing code. The service can often be created by using generated code in a message broker or ESB.

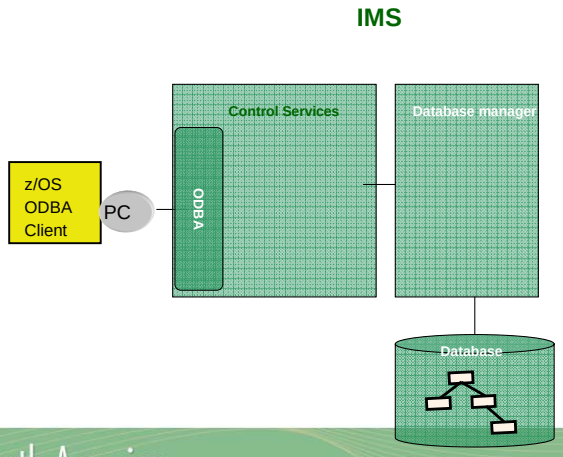
IMS Architecture - Understand the Building Blocks



OTMA is the IMS client/server protocol that uses z/OS XCF to communicate with other z/OS applications



ODBA is the IMS DB callable interface that provides access to IMS databases for z/OS applications



Application Protocols

- **Synchronization level (Sync_level)**
  - NONE
  - CONFIRM
  - SYNCPOINT
- **Commit modes**
  - *Commit_then_send (Commit mode 0)*
    - Output is sent as a result of syncpoint
    - Always uses sync_level of CONFIRM
    - Output is queued until client sends an ACK
  - *Send_then_commit (Commit mode 1)*
    - IOPCB output is sent before syncpoint
    - Sync_level can be either NONE , CONFIRM or SYNCPOINT

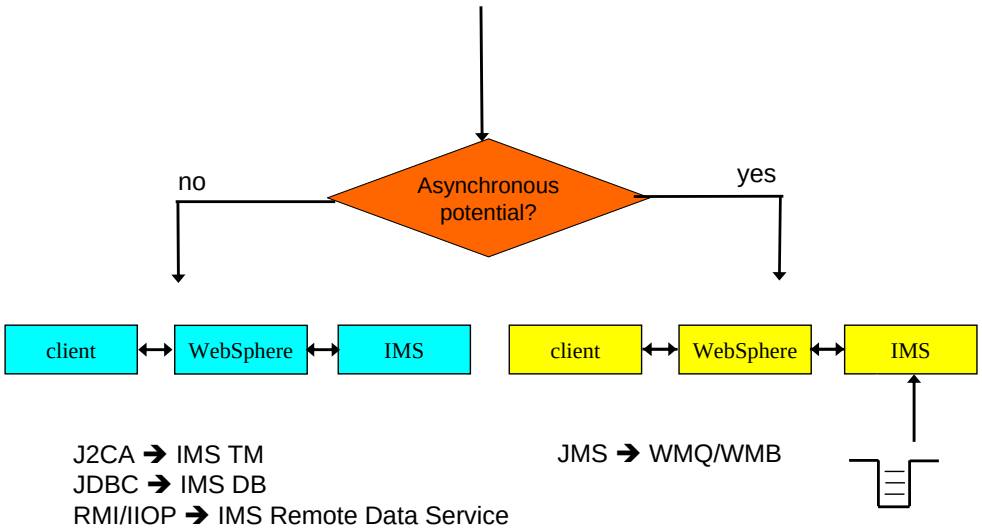
Remote applications can request one of two synchronization levels for message interaction with IMS. NONE assumes that the other partner will receive the message and CONFIRM allows acknowledgement of message receipt (ACK/NAK flows).

In addition to synchronization levels, remote applications can also specify the commit mode, i.e., when IMS processes syncpoint in relation to sending the output reply.

- Commit_then_send (Commit mode 0) is the IMS standard flow and requests that IMS process syncpoint before sending the output reply. This mode cannot be used with IMS conversational and Fast Path (EMH) transactions which require Commit mode 1. Commit mode 0 output messages always use a sync_level of CONFIRM for guaranteed message delivery. The output remains queued in IMS until the client sends an ACK to acknowledgement the receipt of the message.

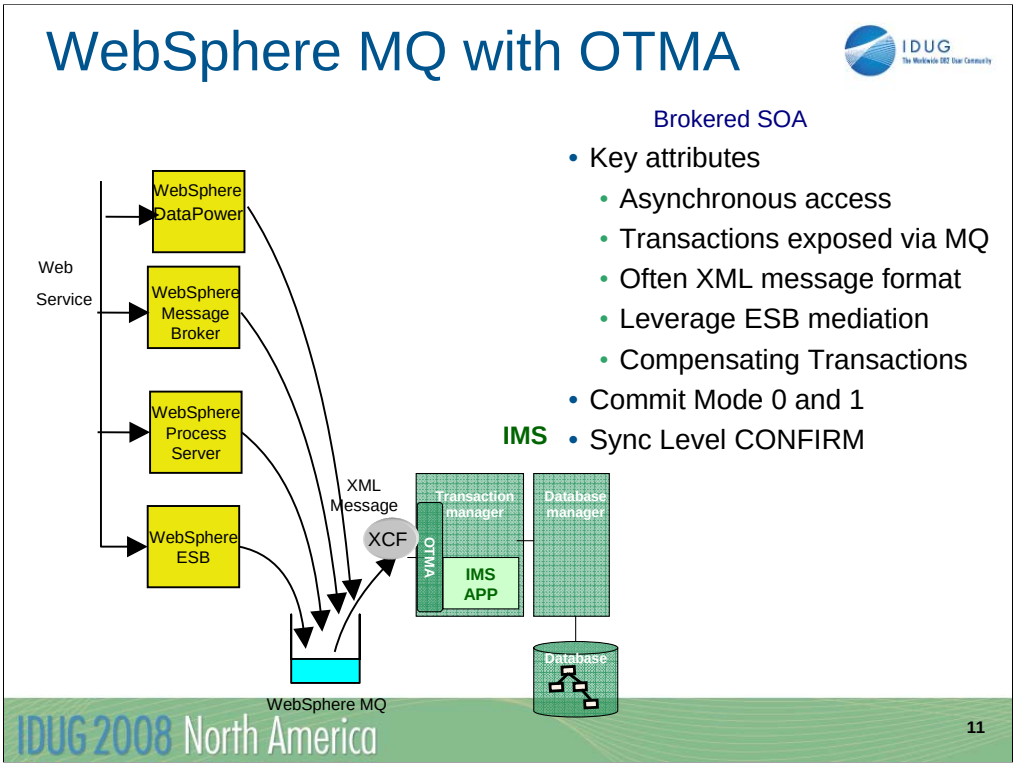
- Send_then_commit (Commit mode 1) requests that IMS send the output message (IOPCB) prior to syncpoint. The output is not queued. The sync_level can be either NONE or CONFIRM. If IMS cannot successfully deliver the message (e.g., IMS Connect is not there) or a NAK is received on the CONFIRM request, then IMS issues a Abend U0119 and the transaction is backed out.

Synchronous vs. asynchronous options



J2CA ➔ IMS TM
JDBC ➔ IMS DB
RMI/IIOP ➔ IMS Remote Data Service

JMS ➔ WMQ/WMB



•Legacy integration using XML via MQ is a common practice.

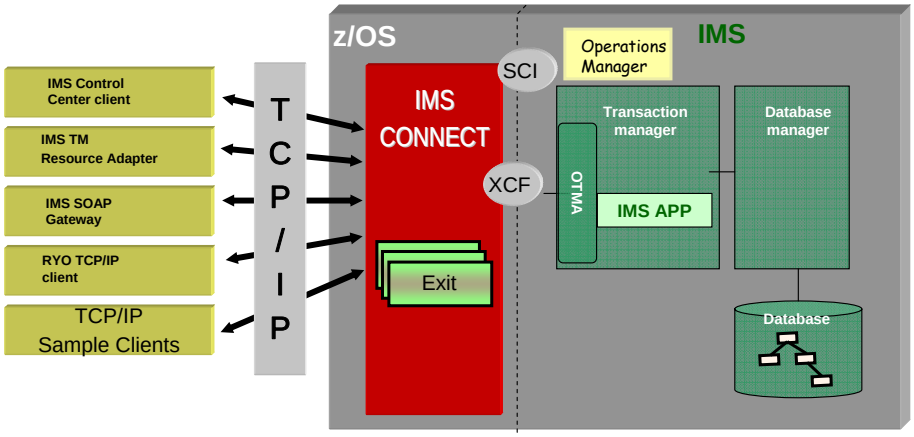
•Need ESB to mediate between Web Service and message format.

•WebSphere Message Broker can mediate to non-XML message formats.

•Exposed transactions good match to required services (possibly with minor mediation in the ESB).

IMS Connect with OTMA

- IMS Connect is the TCP/IP gateway into IMS TM



Commit Mode 0 and 1
Sync Levels NONE, CONFIRM, SYNCPOINT

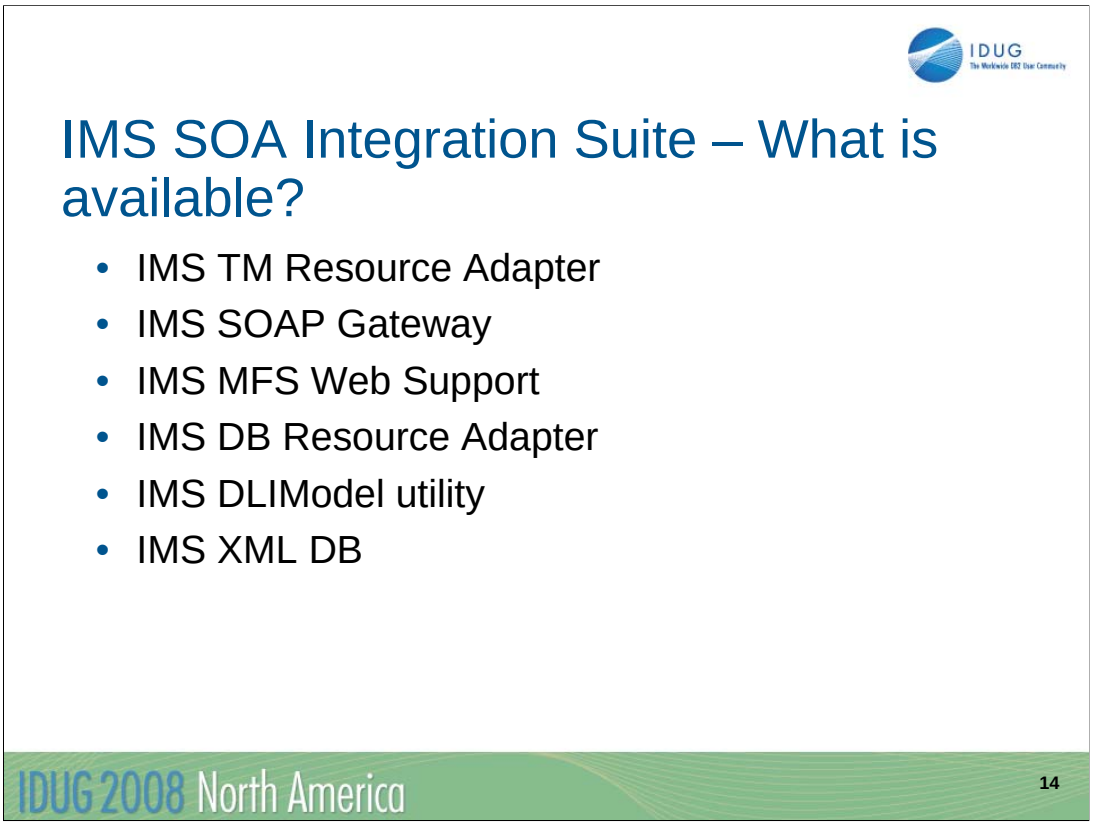
Comparing Solution Types – IMS Connect and WebSphere MQ

Direction Connection

- Natively synchronous (connection-oriented), supports asynchronous (connectionless)
- Direct correlation between input and output
- Potential issues with program-to-program switches when spawning multiple transactions
- Easily supports IMS conversational transactions (relatively transaparent)
- Designing for failure:
  - If connection can not be made, try later
  - Decide what to do when the connection breaks - understand IMS actions

Messaging and Queuing

- Natively asynchronous (connectionless), simulates synchronous (connection-oriented)
- Need to consider how to correlate output to input
- Easily supports program-to-program switches even when spawning multiple transactions
- Requires keeping track of the conversation id to continue an IMS conversation
- Designing for failure:
  - No knowledge of whether entire connection path is available
  - Handle Late reply messages and the dead letter queue



The IMS Integration Suite includes the following IMS middleware functions and tools:

IMS TM Resource Adapter - formally known as IMS Connector for Java

IMS SOAP Gateway

IMS MFS Web support

IMS DB Resource Adapter – formally known as IMS JDBC Connector

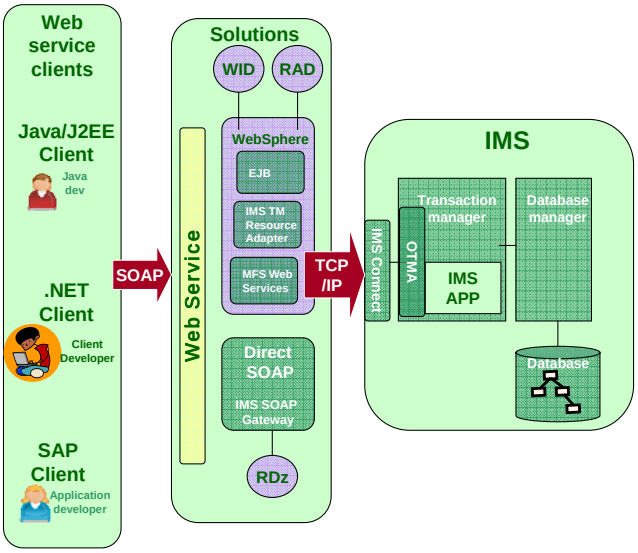
IMS DLIModel utility

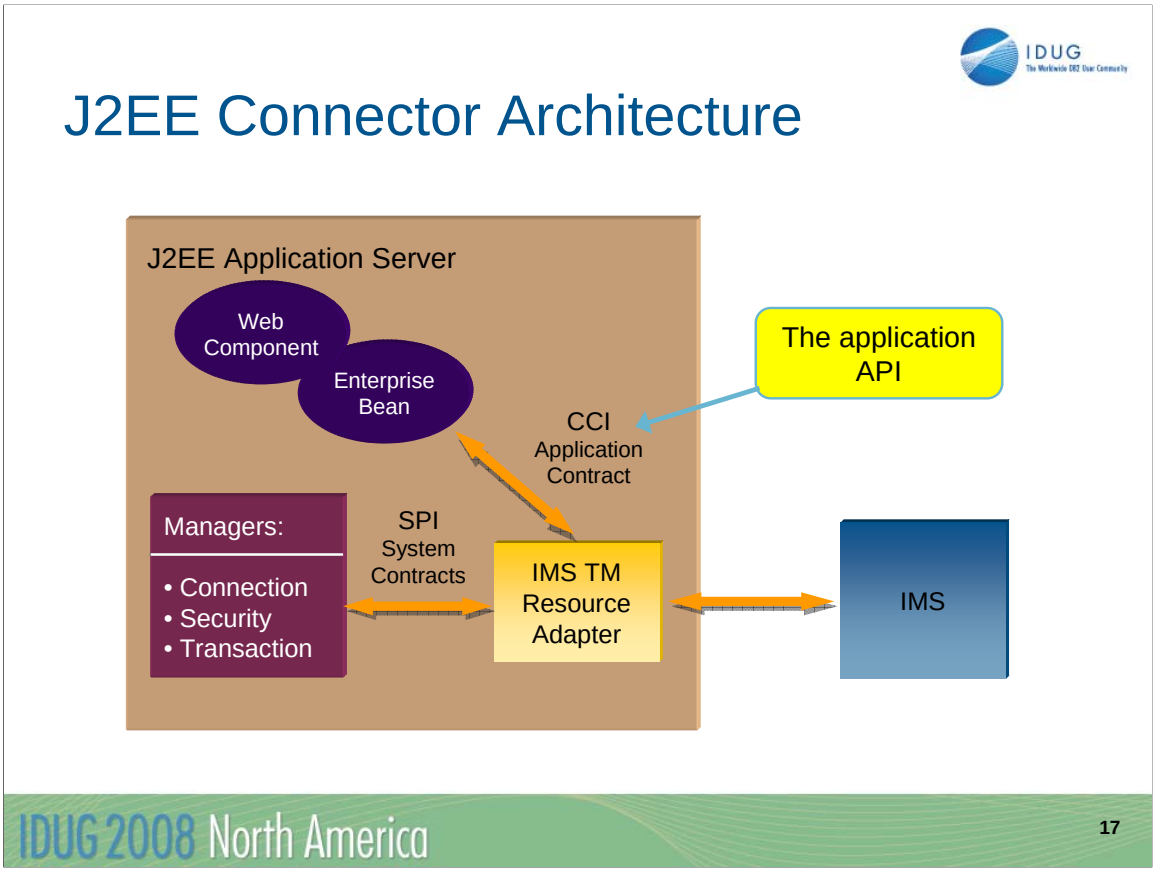
IMS XML DB

IMS TM Resource Adapter

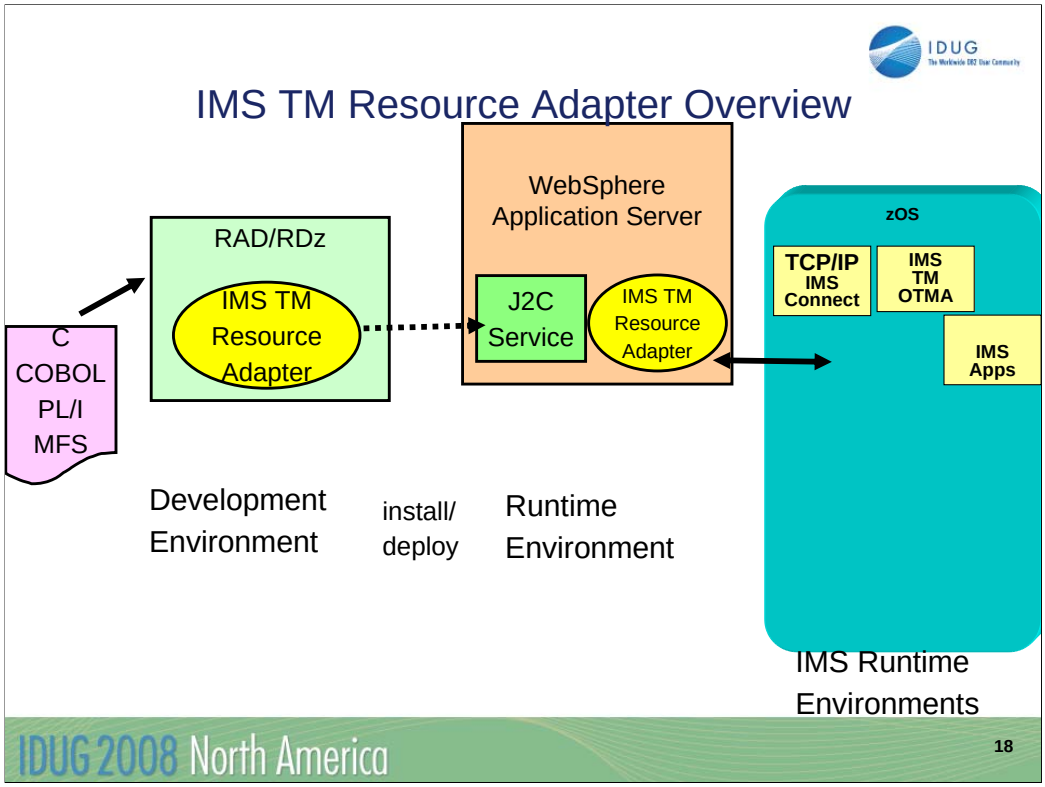
SOA for IMS TM

Adapter SOA





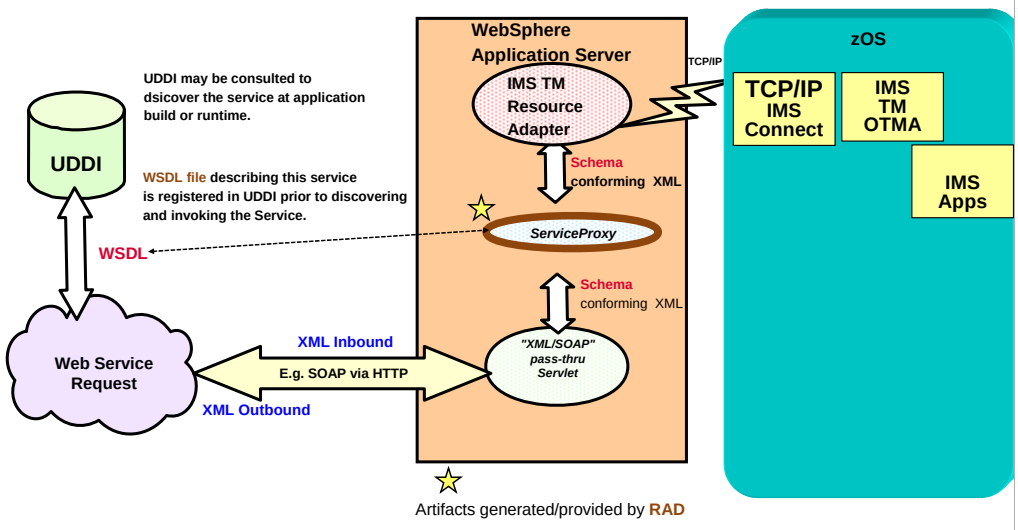
The SPI system contract defines the types of interactions that take place between a resource adapter and the various qualities of services provided by the J2EE container such as connection, security and transaction management. The interactions between an EJB which contains an application's business logic and a JCA resource adapter is governed by the CCI or Common Client Interface. In the case of IMS, the activity between the IMS TM Resource Adapter and its back-end EIS, IMS, takes place over a proprietary interface.

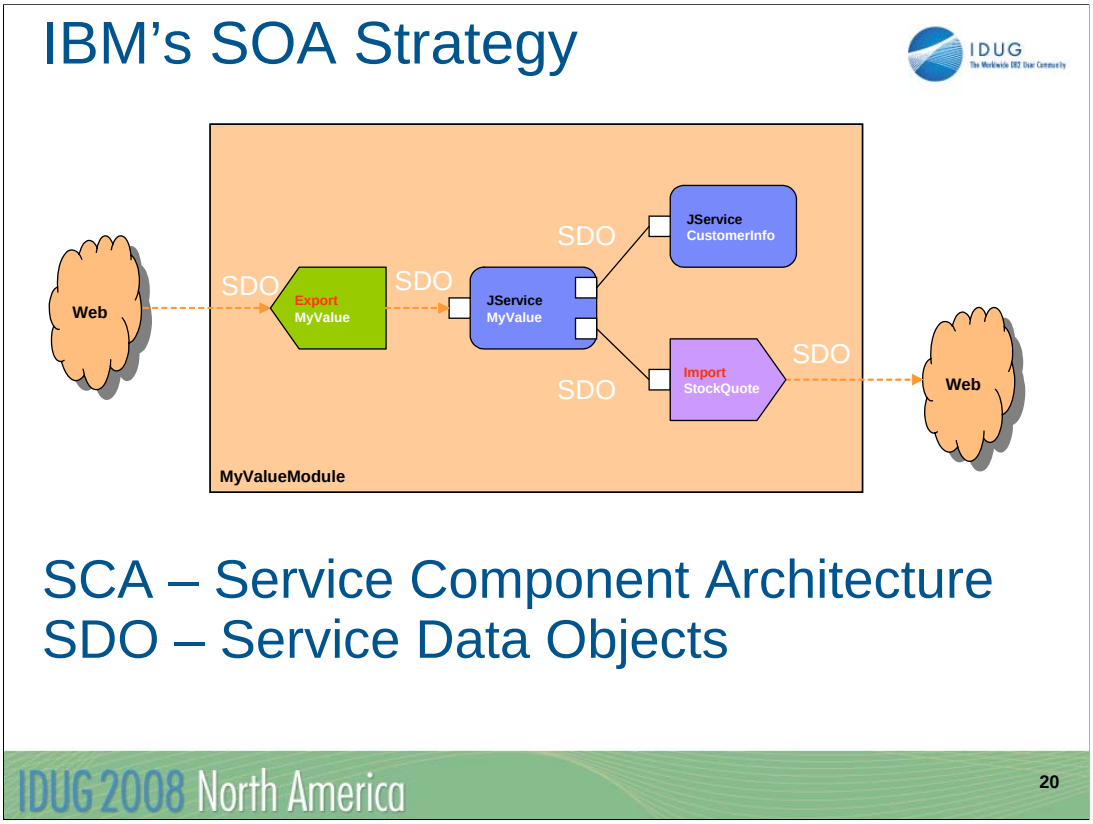


Existing MFS Web Services customers using MFS tooling in WebSphere Studio Application Developer Integration Edition (WSADIE) need to move to the new tooling (RAD / WID / WDz) supporting the latest J2EE Connector (J2C) programming model and the Service Component Architecture (SCA) programming model to continue developing and running MFS Web Services

IMS MFS SOA uses IMS TM Resource Adapter that supports IBM Java integrated development environments (IDEs).

SOAP using IMS TM Resource Adapter





Application programmers are confronted with too many technology choices, too many interfaces to master, and too many particular ways of doing something that apply in one arena or another but not universally.

SCA:

A Service Component is a logical six-tuple comprising:

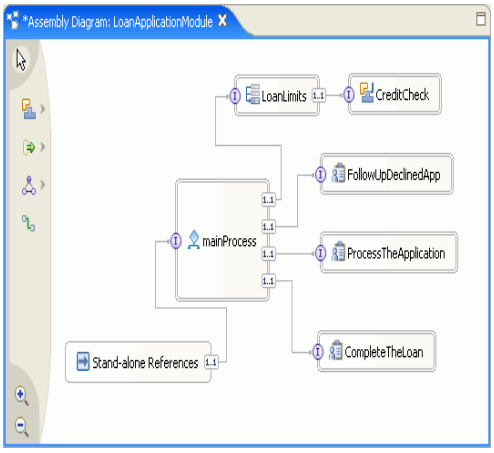
1. An interface definition
2. An implementation
3. An enumeration of the other service interfaces that it calls (if any)
4. A set of policies that it expects (for example security)
5. A description of the state data that it manages (optional)
6. Its proper invocation sequences (optional).

SDOs allow programmers to:

manipulate data while being unaware of the underlying data access technology (JDBC, DLI, etc)

process and manipulate messages without knowing the message transport technology (JMS, SOAP, etc)

IBM WebSphere Integration Developer

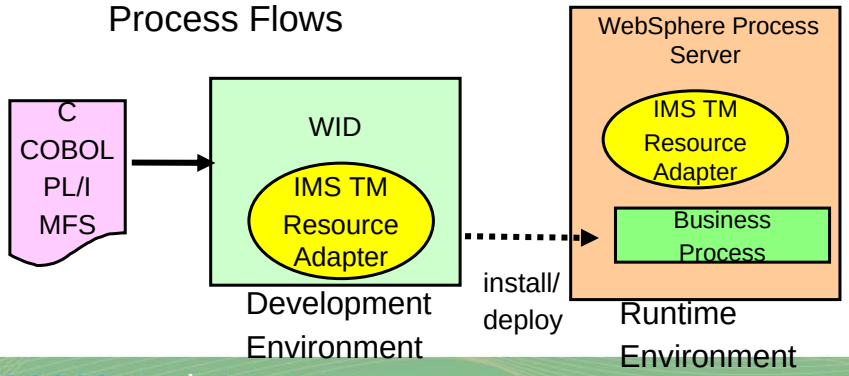


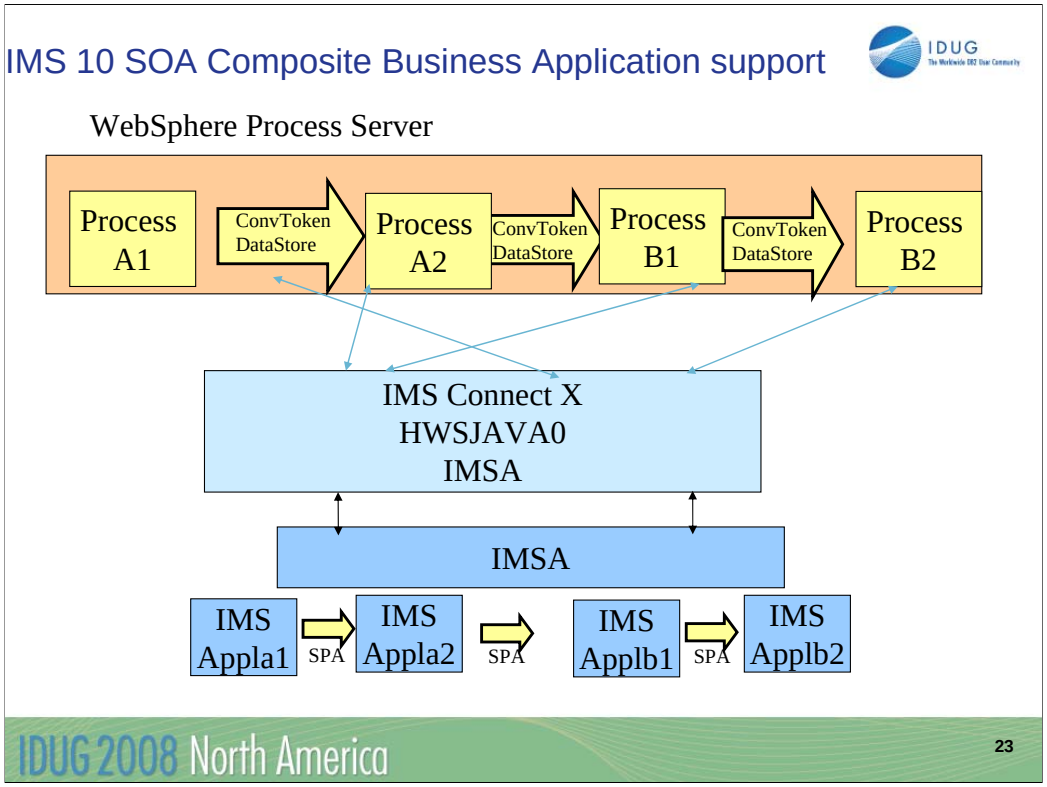
- Simplify and accelerate the development of integrated applications
- Implements Service Component Architecture technology and business process choreography
- Industry-standard service-oriented architecture

IBM WebSphere Integration Developer (WID) is IBM's premier integrated development environment for creating business process flows, state machines and business rules. It simplifies and accelerates the development of integrated applications. WID implements Service Component Architecture and has full support for the Business Process Execution Language (BPEL,) the language of business process choreography. WID also has support for SOA including a wiring editor for assembling service components as you see on the left.

IMS TM Resource Adapter

- IMS SOA Composite Business Application support
  - Provides multi-connection IMS Conversational Transaction support
    - Only available to IMS TM Resource Adapter clients
    - Supports WebSphere Process Server Business Process Flows





In this scenario Process A1 starts conversational tran for PGM Appla1 using a generic datastore ID IMS.

SD routes the request to IMS Connect Y. HWSJAVA0 User Exit determines that generic datastore ID should be routed to IMSA. This establishes an affinity to IMSA for subsequent conversational input message routing.

PGM Appla1 processes the request, updates the SPA and responds to Process A1

The ConvToken and IMSA datastoreID are returned to IMS TM Resource Adapter.

Process A1 passes the ConvToken and IMSA datastoreID to Process A2 which continues the IMS conversation and invokes PGM Appla1.

PGM Appla1 does immediate pgm to pgm switch to Appla2 running in IMSA.

Appla2 responds to Process A2 and does deferred pgm to pgm switch to Applb1 running in IMSB via MSC or Shared Queues.

Process A2 passes the ConvToken and IMSA datastoreID to ProcessB1

ProcessB1 continues the IMS conversation and invokes Applb1 running on IMSB via routing from IMSA

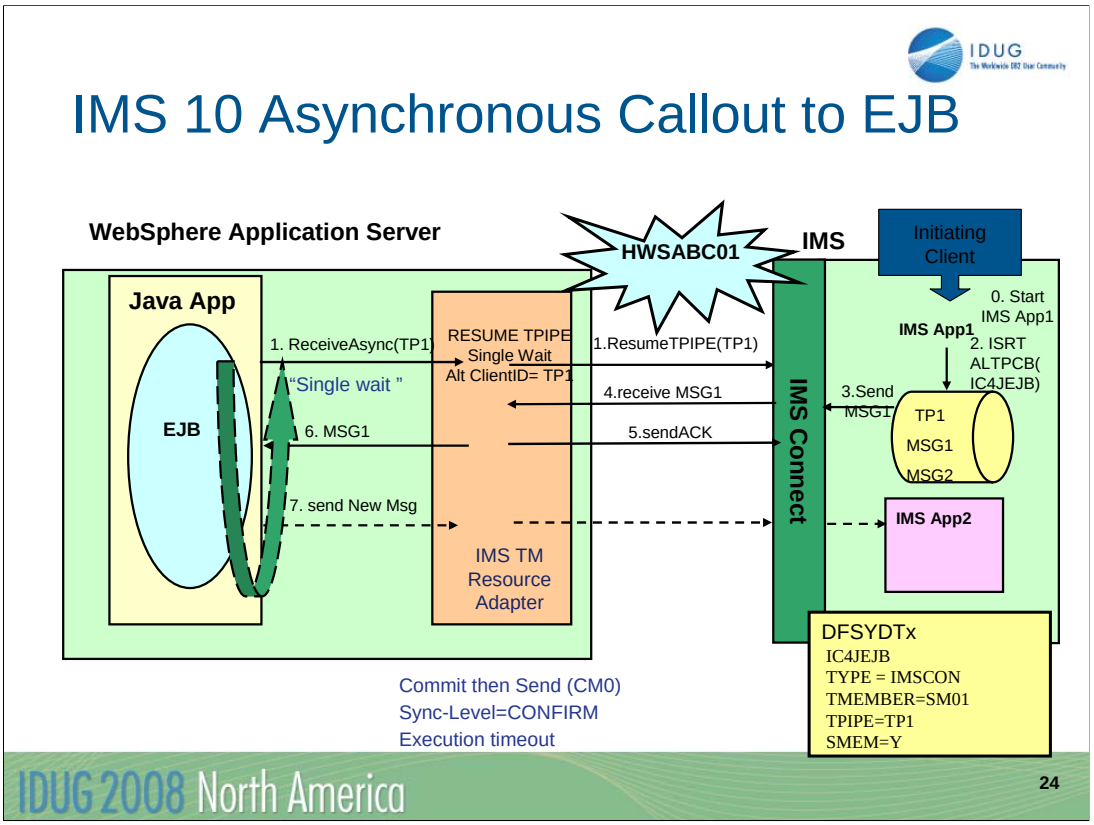
PGM Applb1 processes the request, updates the SPA and responds to Process B1

ProcessB1 passes the ConvToken and IMSA datastoreID to ProcessB2

ProcessB2 invokes Applb1

Applb1 does immediate pgm to pgm switch to Applb2

Applb2 responds to ProcessB2 and ends the conversation.

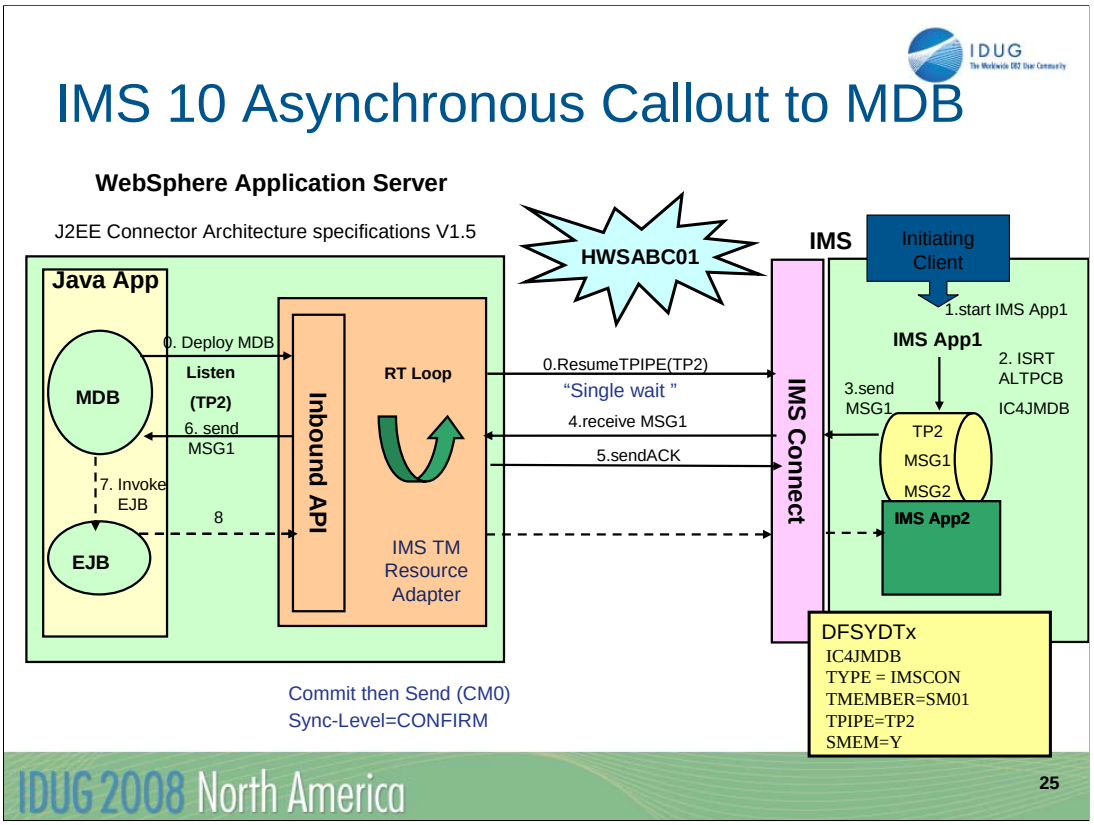


IMS TM Resource Adapter has been enhanced to support access to Enterprise Java Beans (EJB) from an IMS application program. In this scenario an EJB uses the SYNC_RECEIVE_ASYNC_OUTPUT_SINGLE_WAIT (1) alt-clientID=TP1 to request a message queued to TP1 . IMS TM Resource Adapter uses the IMS Connect "Resume tpipe with alternate clientid" protocol to wait for a message to retrieve (1). If there are no messages in the IMS OTMA asynchronous queue for the clientID TP1when the request is made, IMS Connect waits for OTMA to return a message. IMS Connect waits the length of time specified in the executionTimeout property of the

SYNC_RECEIVE_ASYNCOUTPUT_SINGLE_WAIT interaction before returning an exception. IMS Connect receives the request and notifies OTMA to send a message. An already scheduled (0) IMS Application program does an insert to an ALTPCB for a destination defined in an OTMA destination routing descriptor (2). The OTMA destination routing descriptor is used to define the OTMA TPIPE for the en-queue of the message (TP1). OTMA sends the message to IMS Connect (3) and IMS Connect will return the message to IMS TM Resource Adapter (4). IMS TM Resource Adapter receives the message and notifies IMS Connect with an ACK(5). IMS Connect will forward the ACK to OTMA and the message will be de-queued.

IMS TM Resource Adapter receives the message and returns it to the EJB (6).

An optional step is the EJB creates a response to the message request. IMS TM Resource Adapter sends the response to IMS (7). This results in the scheduling of an IMS Application program to process the response.



IMS TM Resource Adapter also supports access to Message Driven Beans (MDB) from an IMS application program. The J2EE Connector Architecture specifications V1.5 describes how an enterprise application can access a J2EE application environment. This is known as Inbound processing. IMS TM Resource Adapter is providing Inbound processing for IMS application programs. From an IMS application perspective this is Asynchronous Callout processing.

In this scenario IMS TM Resource Adapter uses the IMS Connect "Resume tpipe with alternate clientid" protocol to wait for a message to retrieve (0).

An already scheduled(1) IMS Application program does an insert to an ALTPCB for a destination defined in an OTMA Descriptor (2). The OTMA Descriptor is used to define the OTMA TPIPE for en-queue of the message.

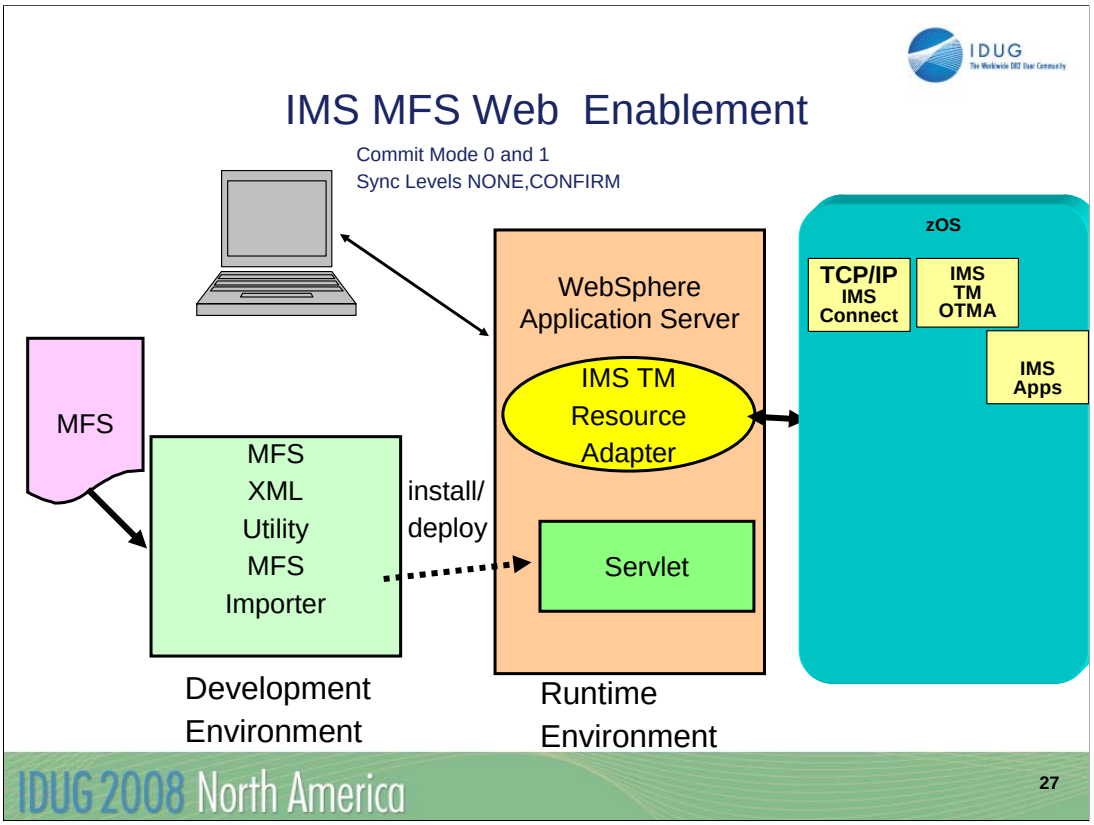
OTMA sends the message to IMS Connect (3) and IMS Connect will return the message to IMS TM Resource Adapter (4). IMS TM Resource Adapter receives the message and notifies IMS Connect with an ACK(5). IMS Connect will forward the ACK to OTMA and the message will be de-queued.

An MDB running inside WebSphere Application server will use the standard JCA 1.5 interfaces to listen to callout request from IMS applications. IMS TM Resource Adapter receives the message send it to the MDB (6).

The MDB will invoke the EJB (7) to process the message.

An optional step is the EJB creates a response to the message request. IMS TM Resource Adapter sends the response to IMS (8). This results in the scheduling of an IMS Application program to process the response.

IMS MFS Web Support



IMS MFS Web Enablement provides the tooling utility and runtime support to Web-enable existing or new IMS MFS-based applications in IBM WebSphere Application Server, and interactively render them for display in standard browsers such as Microsoft Internet Explorer and Mozilla Firefox.

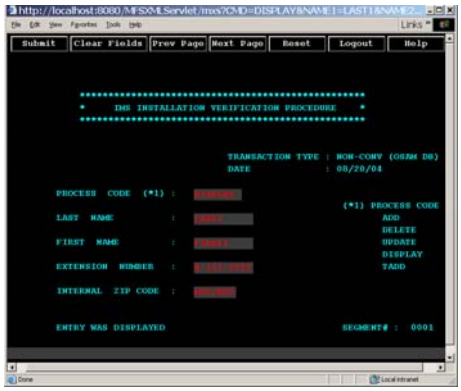
MFS Web Enablement tooling utility support is comprised of the MFS XML Utility and the MFS Importer. The tooling utility support also requires WebSphere Application Server. The MFS XML Utility invokes the MFS Importer to parse MFS source files and generates XML Metadata Interchange (XMI) files for each MID and DIF (Message Input Descriptor or Device Input Format) pair, MOD and DOF (Message Output Descriptor or Device Output Descriptor) pair, and MFS table.

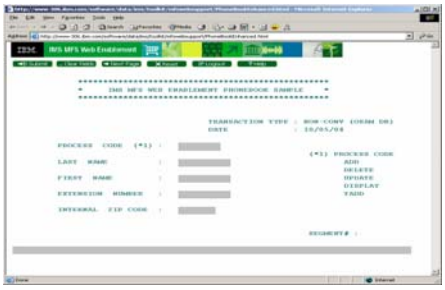
Servlets are generated by the MFS XML Utility tool and deployed to WebSphere Application Server.

IMS TM Resource Adapter is used to provide access to IMS transactions through IMS Connect.

IMS MFS Web Enablement Functional Overview

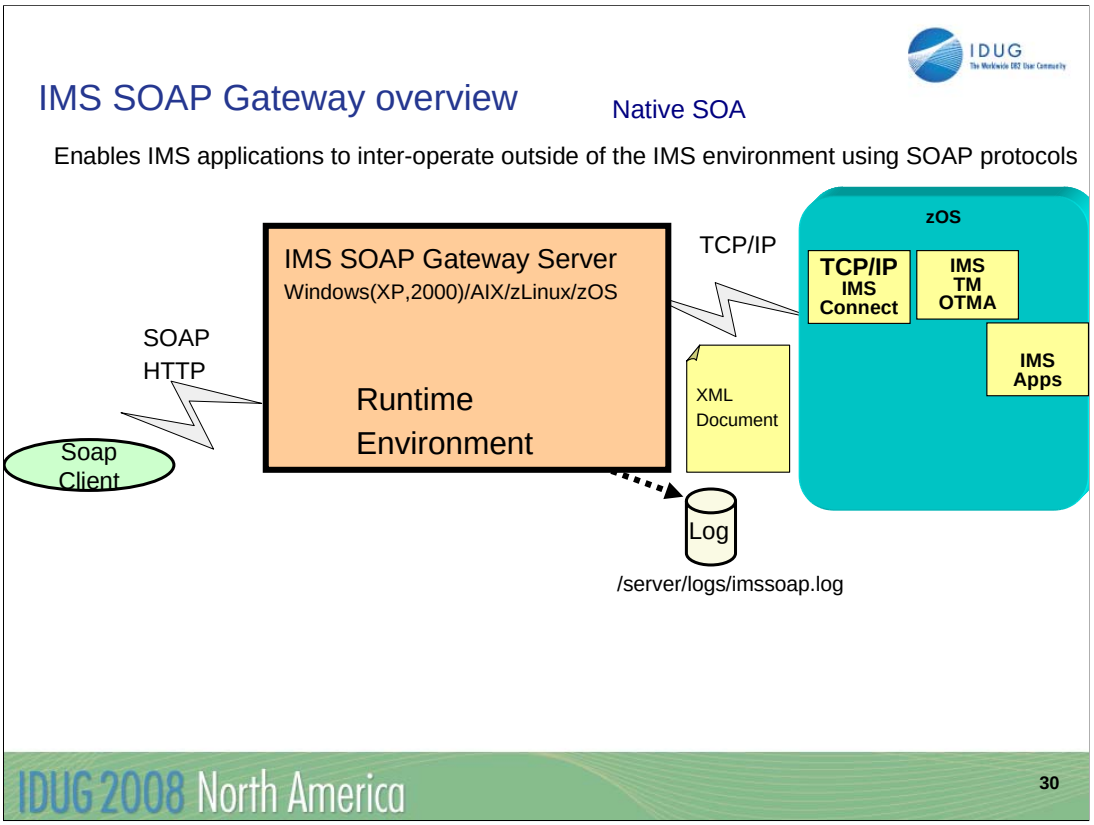
- Provide ***B2C*** solution to web-enable existing MFS-based IMS business logic.
- Maintain conversational iterations.
- Provide simple and user-friendly user-interface development tool
- Render displays on new modern devices, e.g. browsers





Two kinds of different look and feel of the MFS Web Enablement (from different stylesheets)

IMS SOAP Gateway is a separate product that is available free of charge from the IMS Web site



The IMS SOAP Gateway is an XML-based connectivity Web service solution that integrates IMS applications in a Service-Oriented Architecture (SOA) environment.

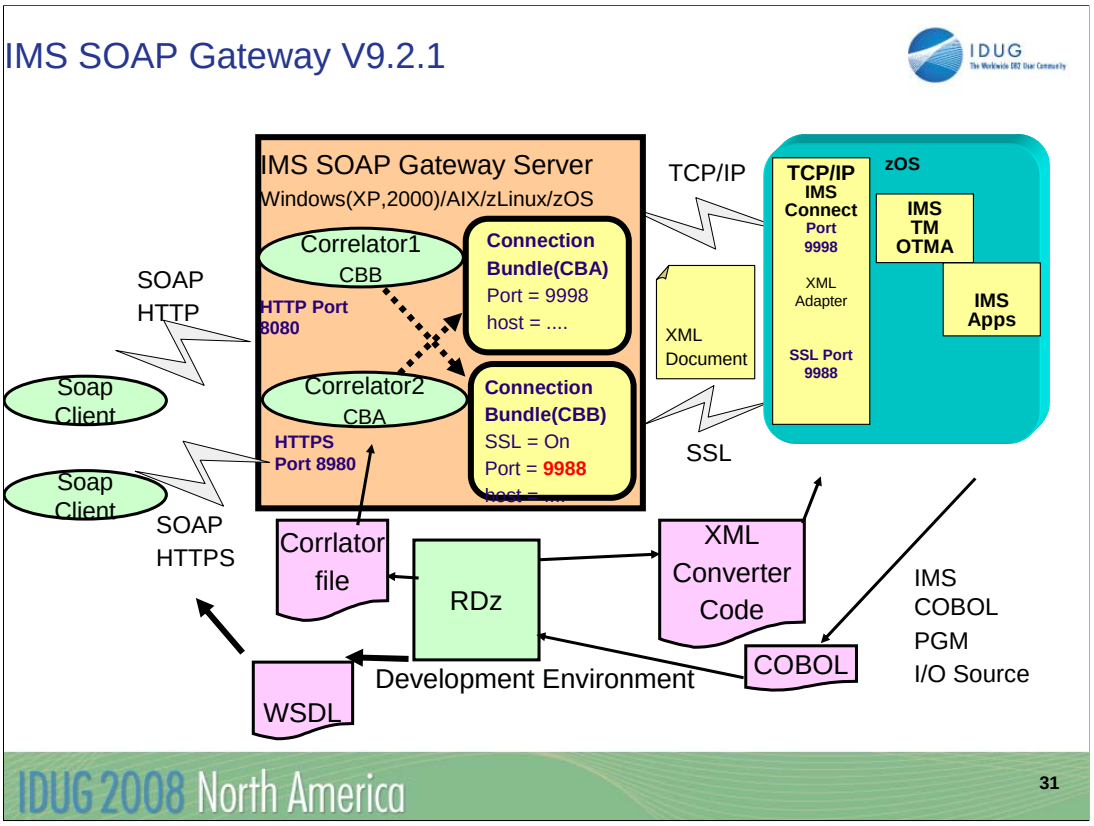
This diagram shows a SOAP client accessing IMS via the IMS SOAP Gateway Server.

IMS SOAP Gateway Server is the HTTP endpoint for the SOAP Client. SOAP is a protocol for exchanging the XML-based messages over a network using HyperText Transfer Protocol (HTTP). HTTP is a protocol used to transfer information over TCP/IP.

To deploy an IMS application as a Web service, you need to create a WSDL (Web service description language) file.

A WSDL file is an XML document that describes a Web service. WSDL files are used by others (for example, the client that invokes the service) to discover the service and to understand how to invoke the service. It specifies the location of the service and the operations that the service exposes.

WebSphere Developer for zSeries (WDz) can be used to generate a WSDL file.



The IMS SOAP Gateway is an XML-based connectivity Web service solution that integrates IMS applications in a Service-Oriented Architecture (SOA) environment.

This diagram shows SOAP clients accessing IMS via IMS SOAP Gateway Server.

In the diagram the XML Adapter function provided by IMS Connect APAR PK24912 is available via the IMS service process. IMS SOAP Gateway Version 9.2 provides the function to support the XML Adapter.

IMS SOAP Gateway Server runs on Windows, AIX and zLinux and is the HTTP endpoint for the SOAP Client. SOAP is a protocol for exchanging the XML-based messages over a network using HyperText Transfer Protocol (HTTP). HTTP is a protocol used to transfer information over TCP/IP.

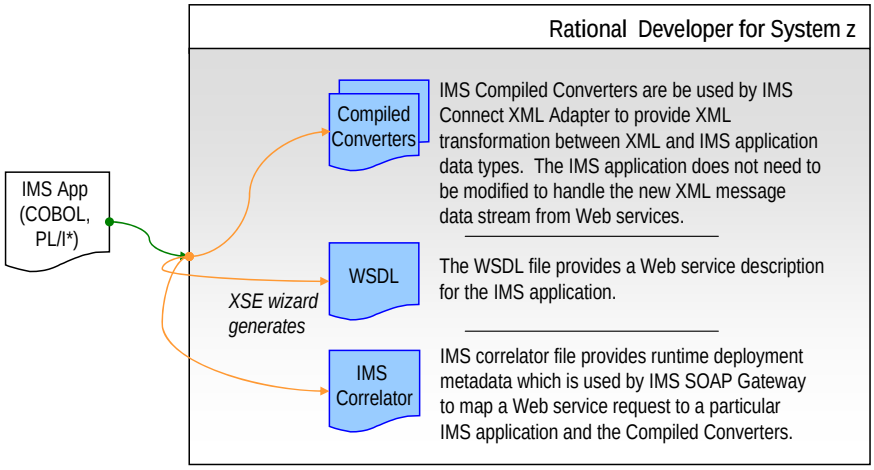
IMS SOAP Gateway provides a deployment utility that generates the connection bundle and the correlator file. The connection bundle specifies the connection and security properties between IMS SOAP Gateway, IMS Connect, and IMS.

The correlator file specifies transaction, and runtime properties and the information that the IMS SOAP Gateway needs to match incoming requests to the appropriate back-end IMS application. It also identifies the Connection Bundle.

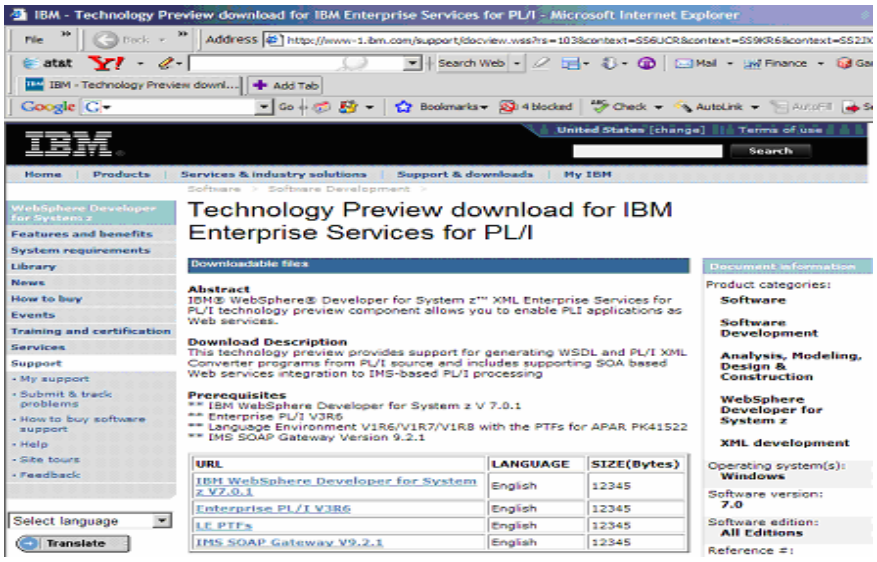
WebSphere Developer for zSeries XML can be used to generate the COBOL converter drivers to transform the XML data in IMS Connect.

Rational Developer for System z

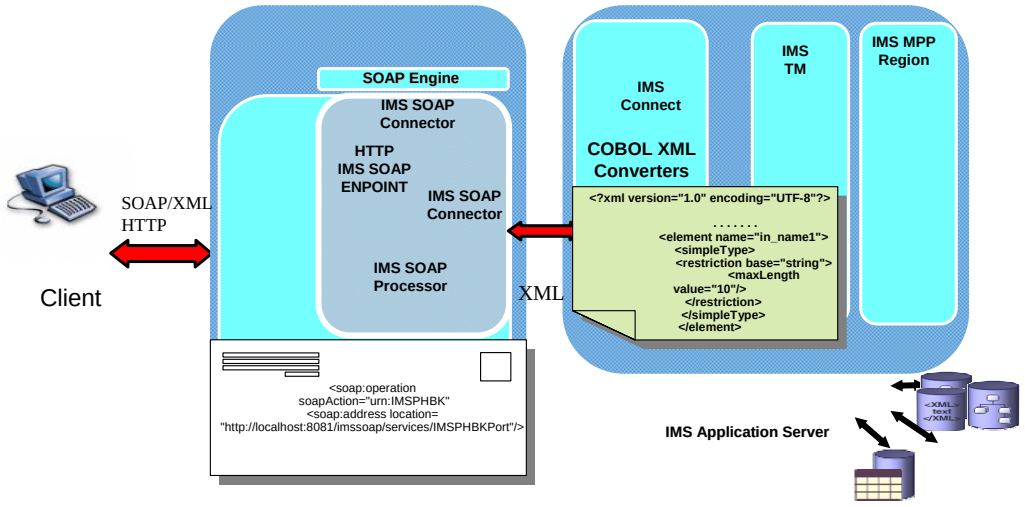
- *Application development tool for modernizing and developing System z application*



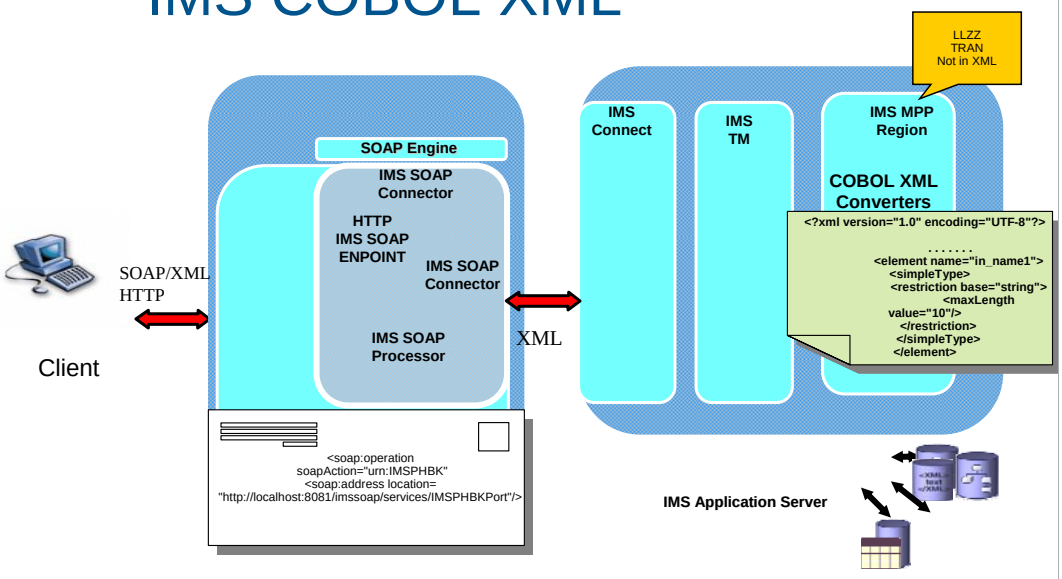
PL/I Support

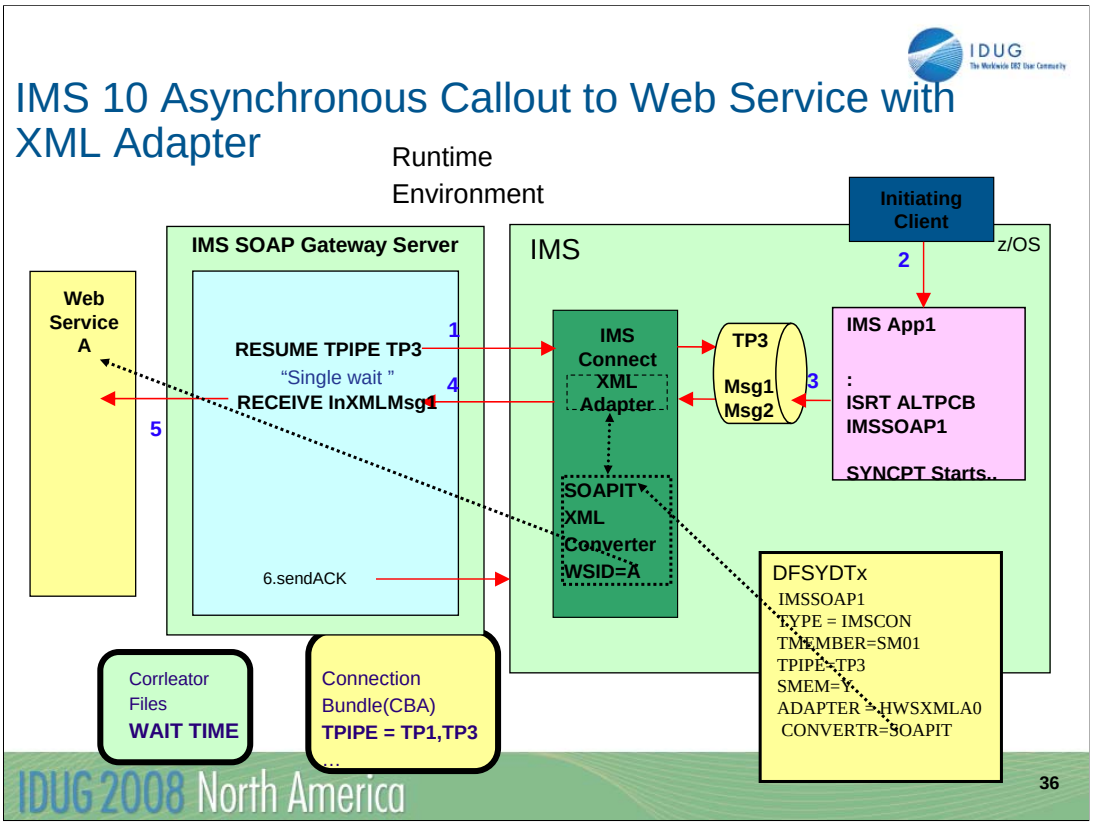


IMS CONNECT COBOL XML



IMS COBOL XML





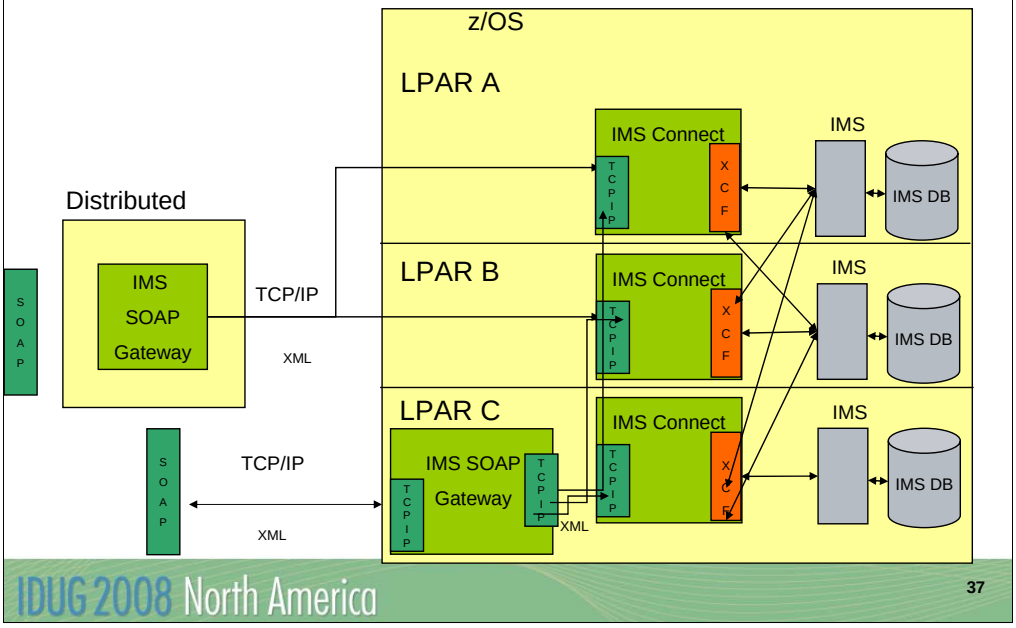
In this scenario an IMS Application program does an insert to an ALTPCB whose destination is an OTMA Descriptor. The OTMA Descriptor is used to define the OTMA TPIPE for en-queue of the message.

Steps 1, 2 and 3 schedule the IMS application which results in the insert to ALTPCB defined by the OTMA Descriptor that represents IMS SOAP Gateway.

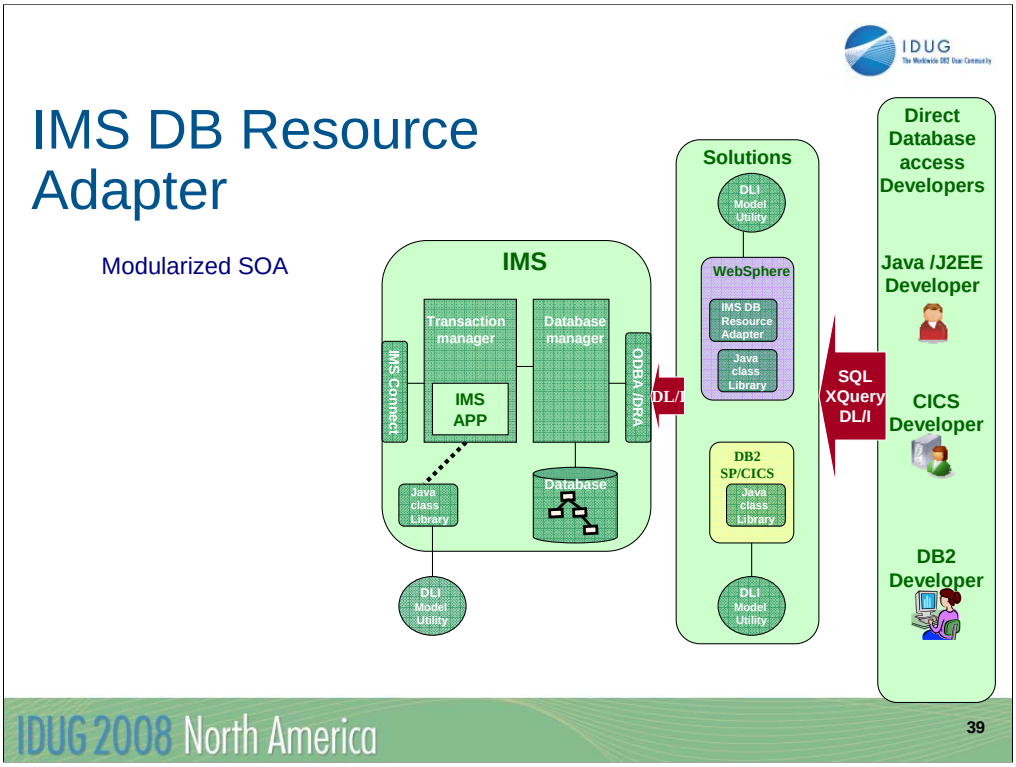
IMS SOAP Gateway will use the Resume TPIPE Alt-ClientID protocol to retrieve the message(1). IMS Connect will invoke the COBOL adapter to perform XML transformation of the message and send the message to the IMS SOAP Gateway(4). IMS SOAP Gateway receives the messages and sends it to the Web Service(5).

An optional step is the Web Service creates a response to the message request. IMS SOAP Gateway sends the response to IMS (6). This results in the scheduling of an IMS Application program to process the response.

IMS SOAP Gateway runtimes
IMS 10 z/OS



The IMS DB Resource Adapter is a function of IMS that enables a programmer with minimal IMS knowledge to write Java application programs that access IMS databases

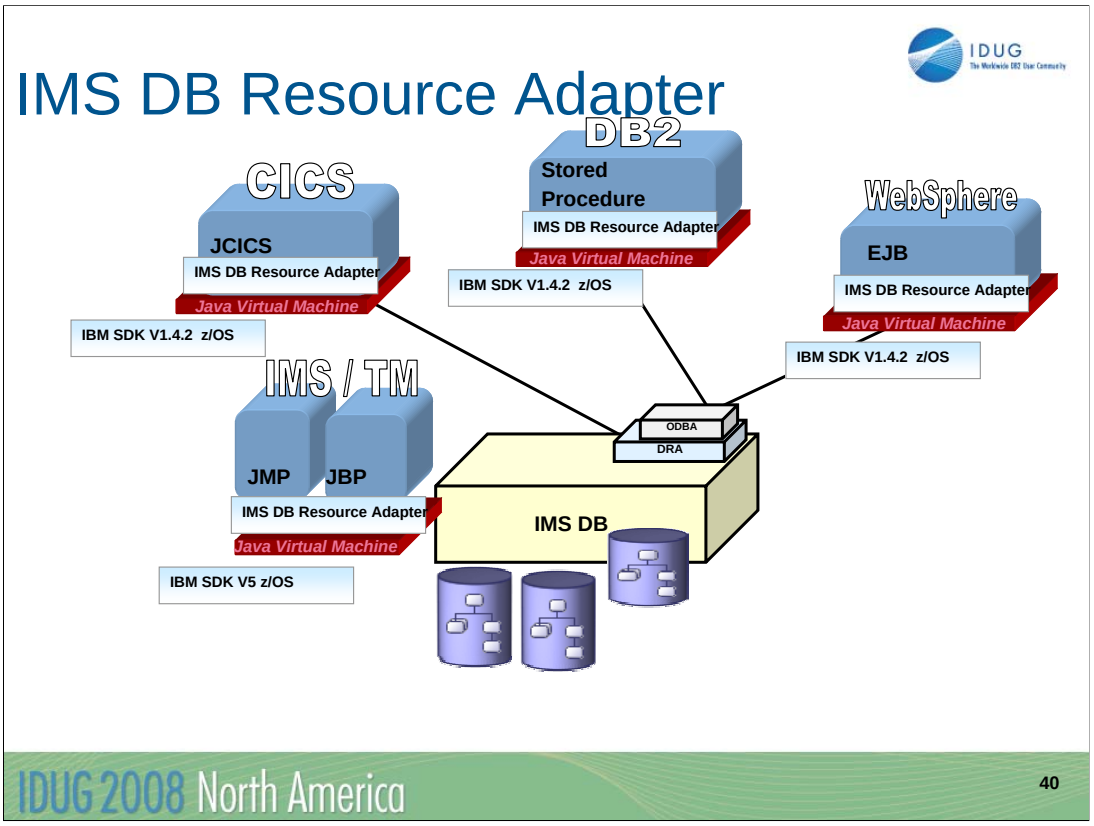


Animated version:

These are all the developers who want to access IMS data (click)

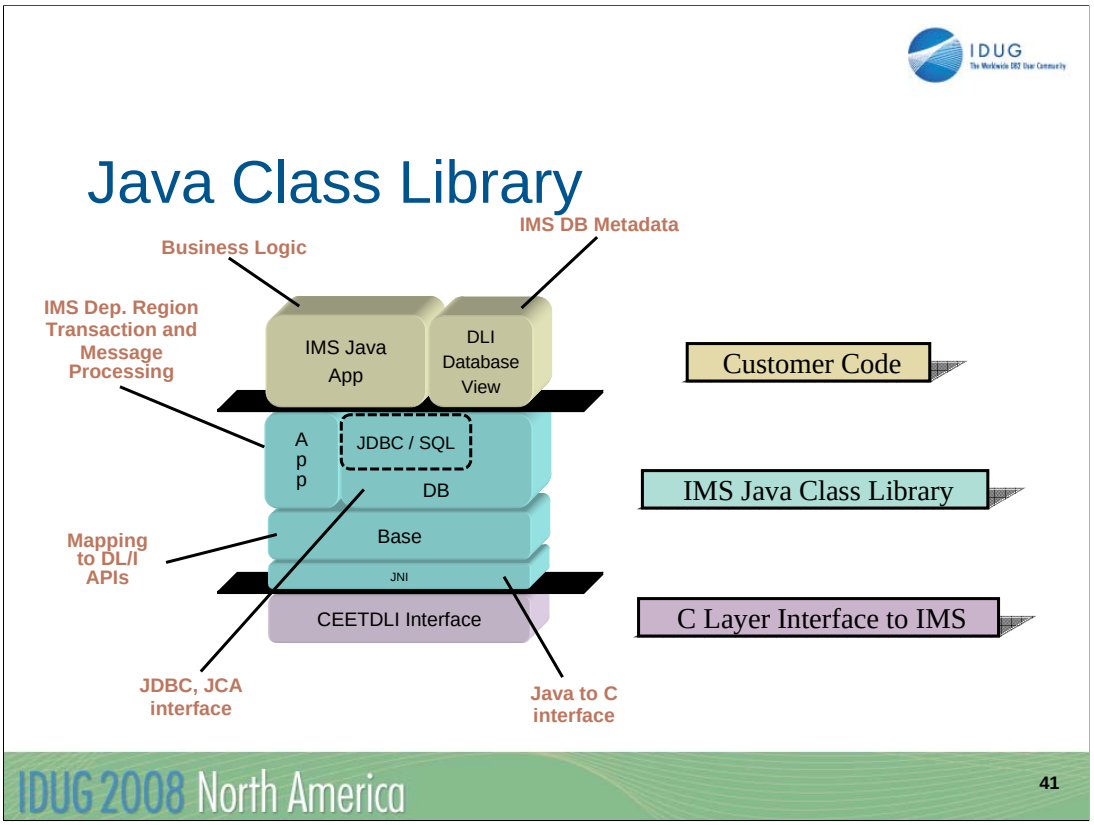
These are the solutions we offer (click)

They can access IMS using SQL, XQuery or DL1



The IMS DB Resource Adapter enables JDBC access to IMS DB from IMS TM JMP/JBP environments, CICS Java application, DB2 Java Stored procedure, and Enterprise Java Beans running on WebSphere distributed and z/OS environments.

IMS V10 requires SDK V5 for JMP and JBP regions, IMS DB Resource Adapter for CICS, DB2 or WAS requires SDK V1.4.2 or higher.



C Layer interface via CEETDLI use Java Native Interface (JNI)

Base – 1-1 Java mapping of the IMS DL/I calls

Db – IMS DB Resource Adapter JDBC driver supporting sql calls

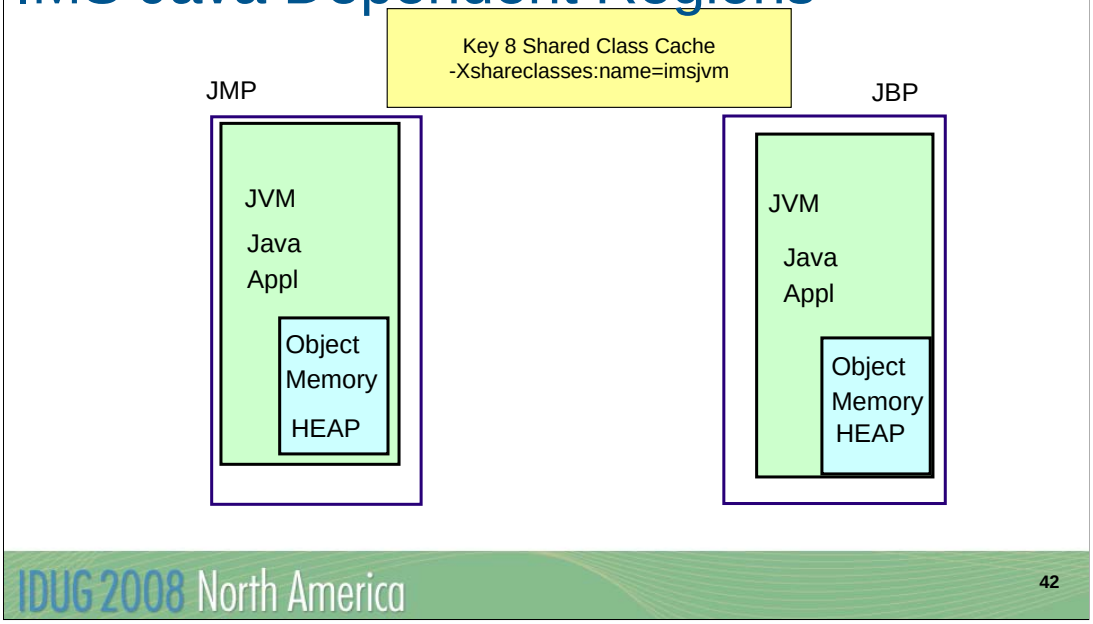
App - running in an IMS Java dependent region and offers reading/writing messages to ims message queue

Customer code – Java code that does uses IMS DB Resource Adapter APIs

DLI DatabaseView- ims does not have online metadata to define DB structure

Tooling – DLIModel utility generates database view

IMS DB Resource Adapter - SDK V5
IMS Java Dependent Regions



Java Message Processing (JMP) region

JMP regions enable the scheduling of only Java message-processing applications. A JMP application is started when there is a message in the queue for the JMP application and IMS schedules the message to be processed (similar to MPP applications). It can access the following resources:

– IMS message queue for input and output messages
– IMS databases
– DB2 for z/OS databases

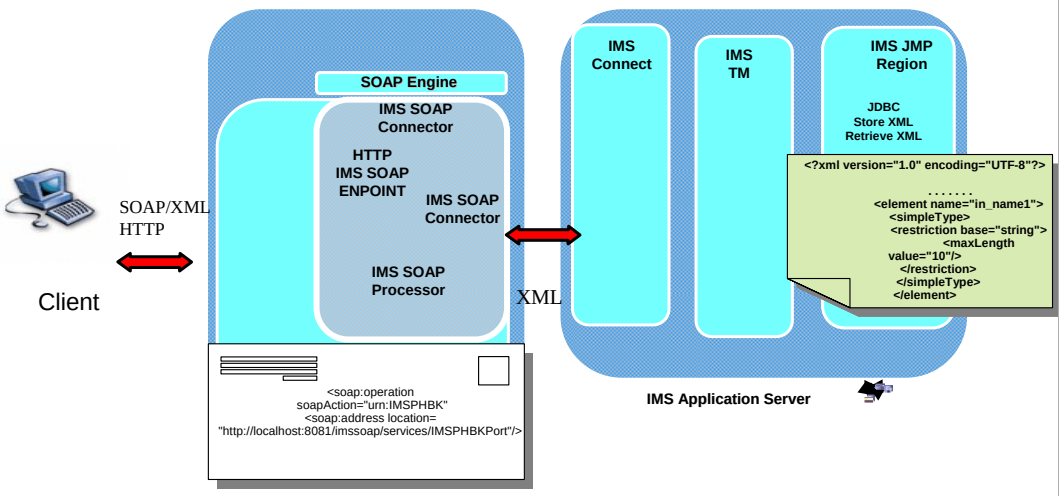
Java Batch Processing (JBP) region

JBP regions run the Java applications that perform batch-type processing online and can access the IMS message queues for output (similar to non-message-driven BMP applications). JBP applications are started by submitting a job with JCL or from the Time Sharing Option (TSO). It can access the following resources:

– IMS message queue for output messages
– IMS databases
– DB2 for z/OS databases

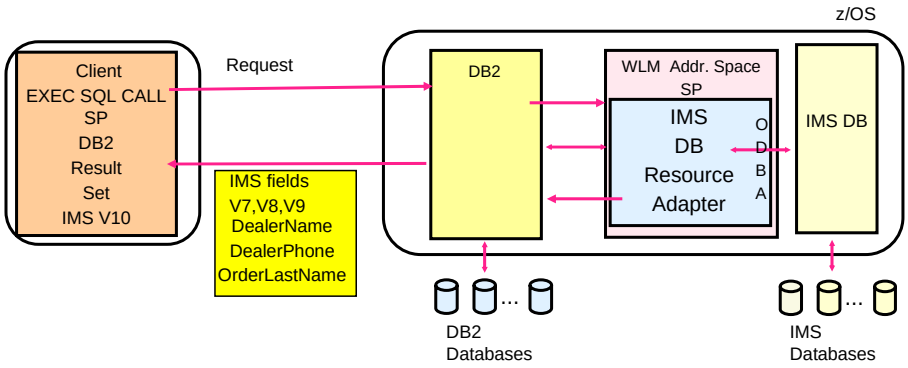
These Java-dependent regions are attached to an IMS control region directly.

IMS DB Resource Adapter XML and IMS SOAP Gateway



IMS DB Resource Adapter

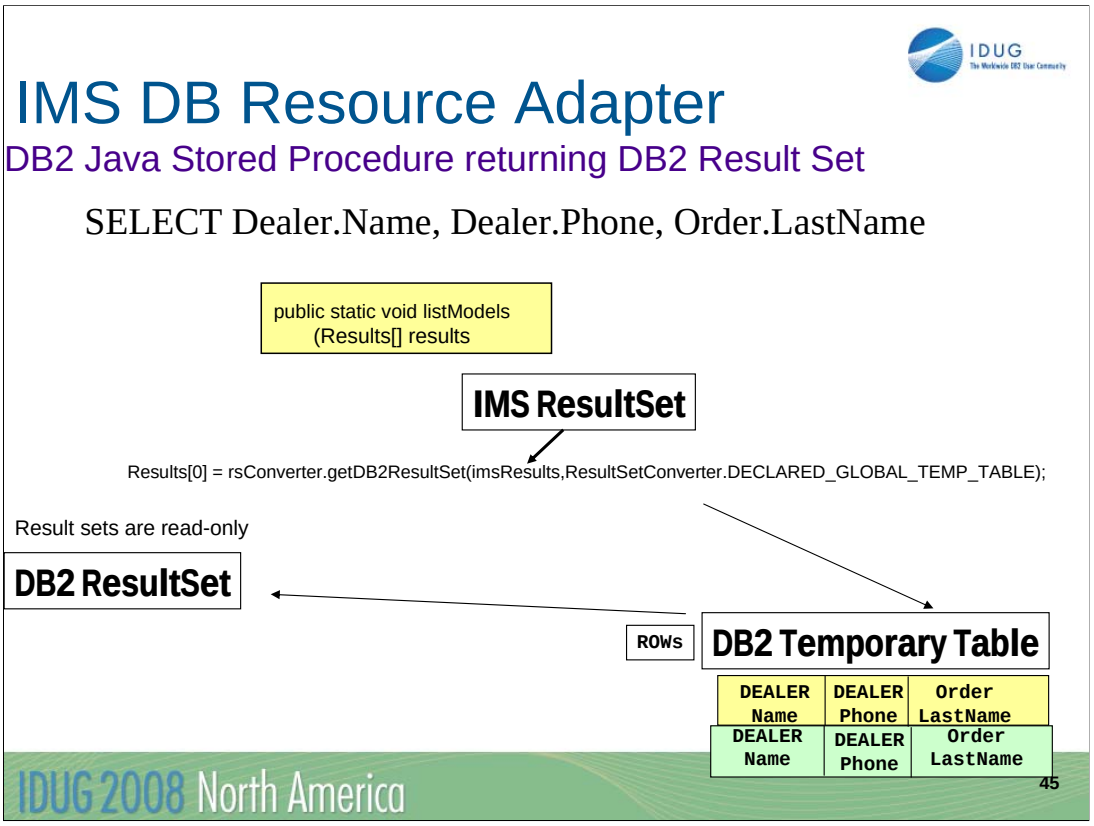
DB2 Java Stored Procedure Support



Stored procedures are application programs executed by DB2 in response to an SQL CALL statement

When the DB2 Stored Procedure needs to access IMS DB the IMS Open Database Access (ODBA) function is activated.

Notice in IMS V7,V8 and V9 individual IMS fields are returned to the client. In IMS V10 a DB2 Result Set is returned to the client.



By embedding the IMS data within a DB2 result set the type of data and the amount of data does not need to be known when the stored procedure is developed.

The DB2 global temporary table (GTT) is either created or declared to be used in the process of converting an IMS result set to a DB2 result set.

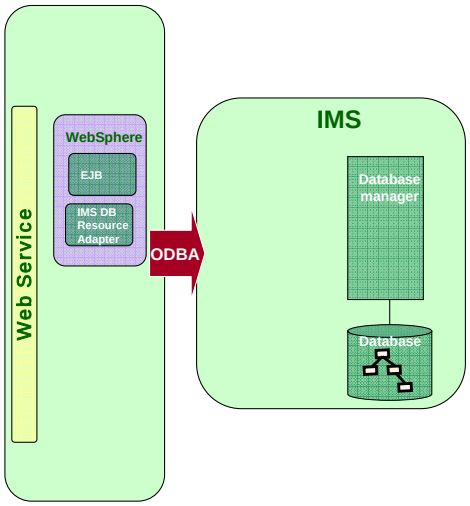
The IMS DB Resource Adapter ResultSetConverter class is used to convert an IMS result set instance into a DB2 result set instance. This method takes in an instance of com.ibm.ims.db.DLIResultSet object which is IMS DB Resource Adapter's implementation of the java.sql.ResultSet interface and converts it into an instance of the DB2 implementation of the java.sql.ResultSet interface

Note the ResultSet[] results replaces

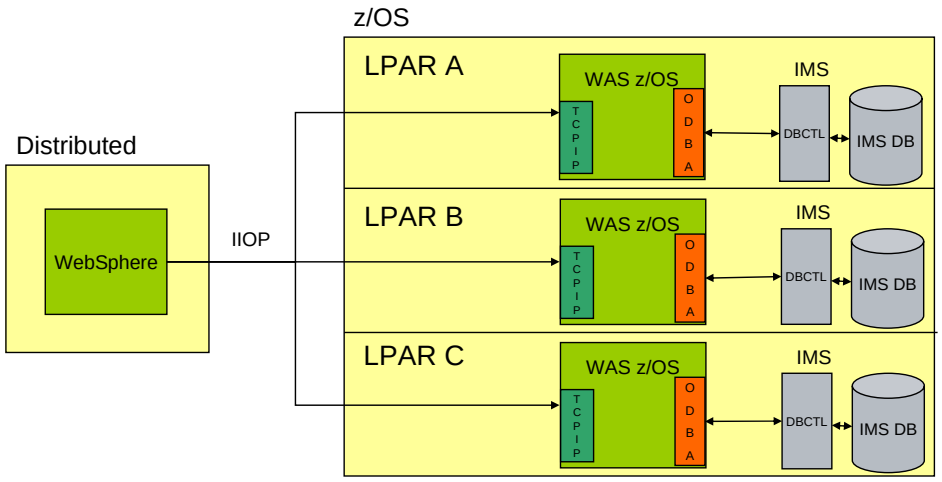
(String[] dealername1, String[] dealerphone1, String[] orderlastname1,

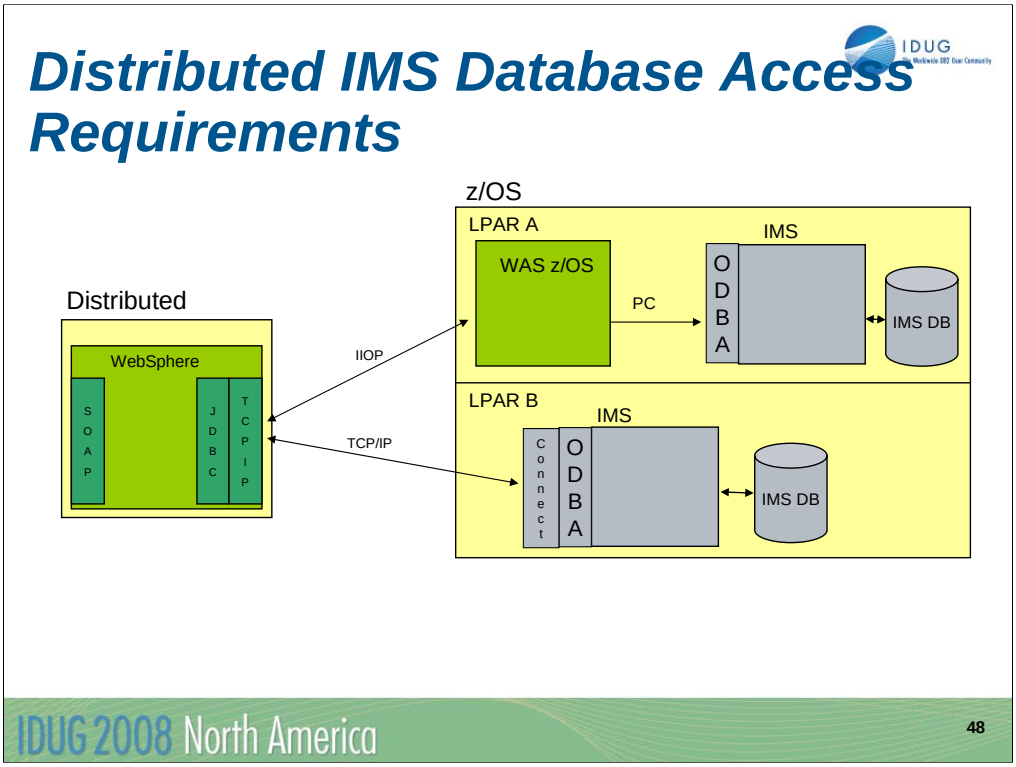
String[] dealername2, String[] dealerphone2, String[] orderlastname2)

IMS DB Resource Adapter with WAS on z/OS



IMS Distributed DB Resource Adapter

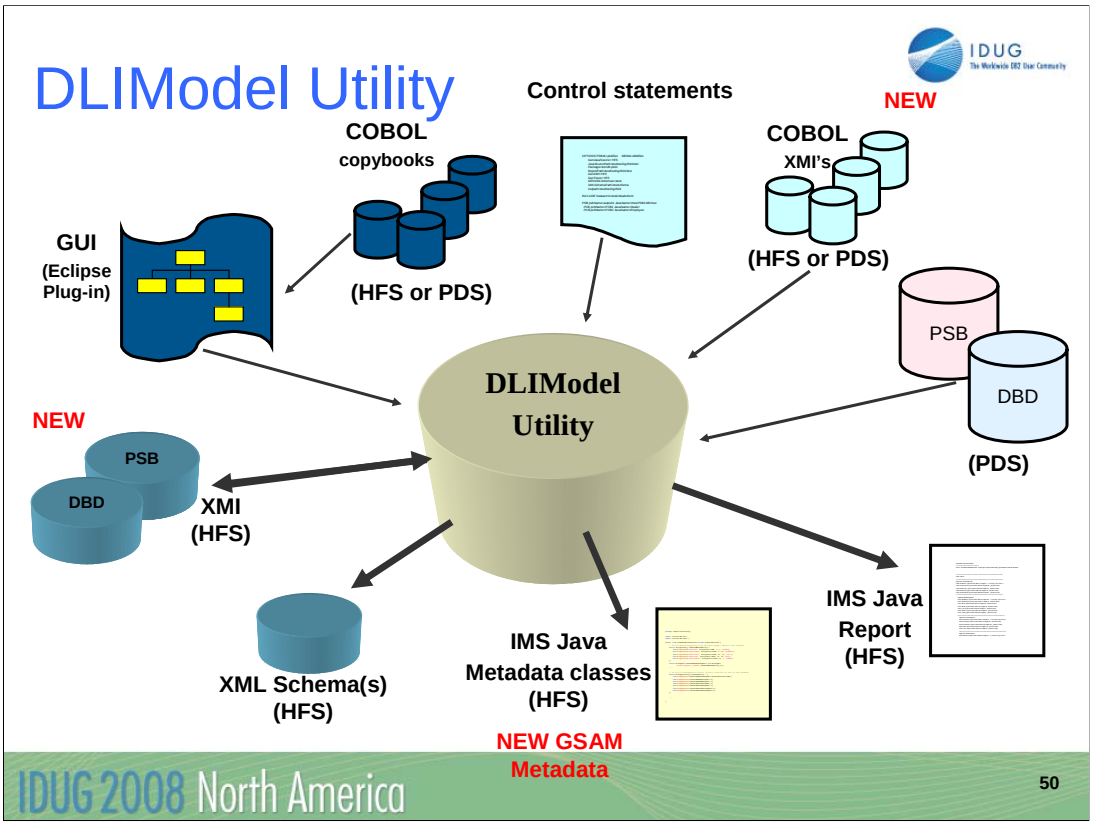




Key Message: IMS is also addressing requirements for extended distributed database access.

In the top box on the right labeled LPAR A you'll see our current solution. Once again, we leverage WebSphere z/OS as the TCP/IP endpoint on the mainframe and our libraries use the ODBA API to access IMS databases. The requirement we have is to leverage IMS Connect as the TCP/IP endpoint on the mainframe. This nicely positions IMS Connect as the complete gateway to IMS resources…both TM and DB related. This is shown in LPAR B.

IMS DLIModel utility



The IMS DLIModel utility has been enhanced to generate XMI from PSB and DBD source. The generated XMI can also be used as input to the DLIModel utility.

GSAM now uses the GSAMDLIDatabaseView IMS Java class for metadata information about the GSAM database. The DLIModel Utility now supports GSAM databases.

Input:

Shows the inputs and outputs of the DLIModel utility. The actions of the utility are directed by control statements that you supply. PSB and DBD source members are read from their PDS or PDSE data sets and parsed by the utility to build an in-memory object model of the database structure and the PSB's view of that structure.

Note IMS COBOL copybooks can only be processed by the GUI and the BPXBATCH utility can only process COBOL XMI representations of the COBOL copybooks.

Output:

The utility generates various outputs that were requested through control statements. You can specify to have an IMS Java metadata class be generated for the PSB processed, together with a corresponding easy-to-read DLIModel Java Report for the Java programmer to use.

You can specify an XMI description of the entire in-memory model (one description covers the PSB and all DBDs processed in the run).

You can also request a detailed trace file of the utility execution if one is necessary for problem resolution.

IMS Java metadata classes

The DLIModel utility produces the necessary metadata classes needed to develop IMS Java applications. However, the Java developer needs only to reference the DLIModel Java Report for information about the classes.

DLI Model Java Report

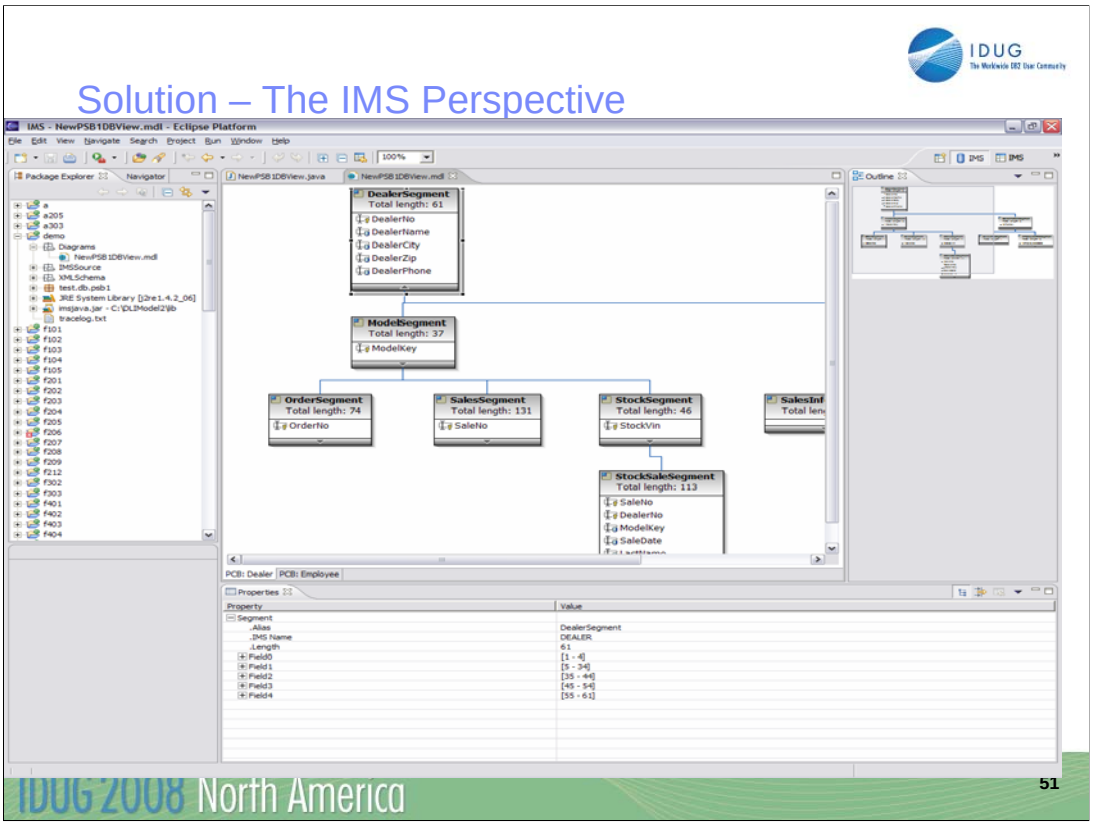
The DLIModel Java Report summarizes the structure of the IMS databases in a way that allows you to create IMS Java applications and to code SQL queries against the databases. With the DLIModel Java Report, you do not have to interpret the syntax of the IMS Java classes or refer to the DBD or PSB source.

XMI Description of databases

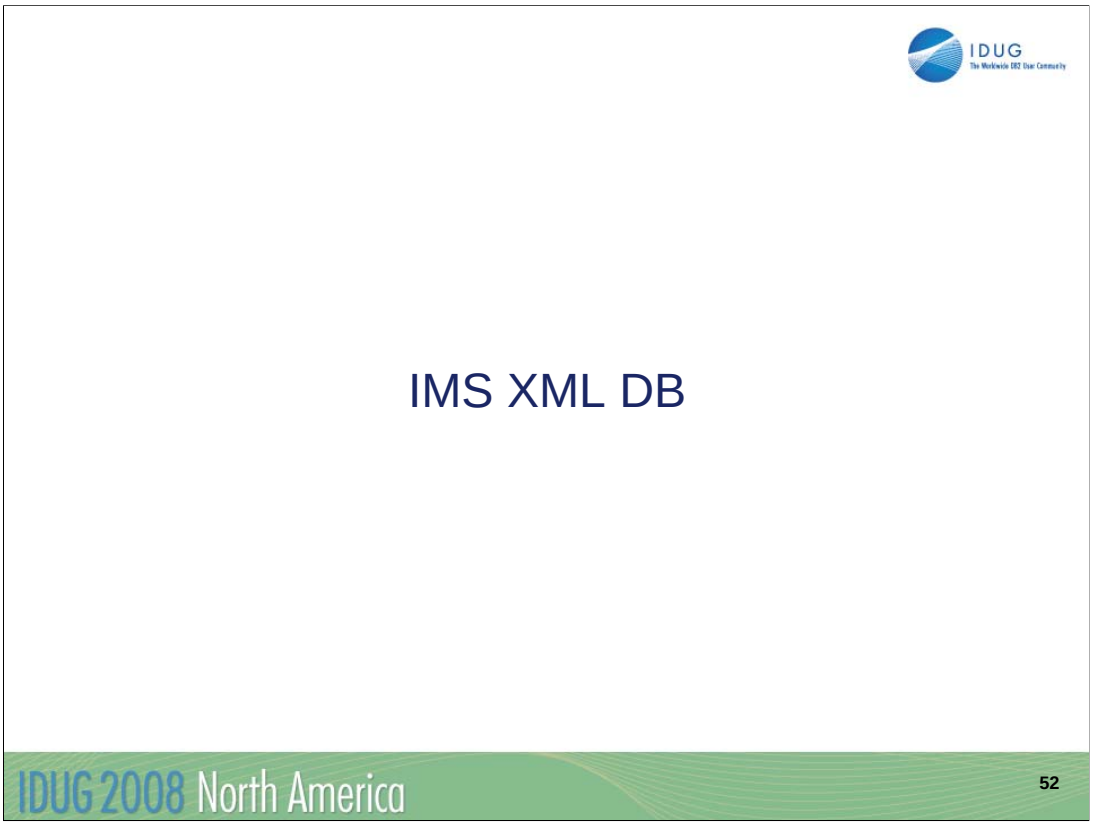
An XMI file, written in UTF-8 encoding, is produced by the utility if you specify genXMI=YES in the OPTIONS control statement. It describes all of the PCBs and their referenced DBDs processed in the entire run of the utility. The XMI that is produced by the utility is based on a metamodel of IMS database defined in UML. This model is a package with a number of inheritance relationships to the OMG Common Warehouse Metamodel (CWM). However, only the IMS package itself is included and used in the DLIModel utility.

XML Schema

The generated XML schema, written in UTF-8 encoding, is an XML document that describes an IMS database based on a PCB. An XML schema is required to retrieve or store XML in IMS. IMS uses an XML schema to validate an XML document that is either being stored into IMS or being retrieved from IMS. The XML schema, not the application program, determines the structural layout of the parsed XML document in the database during storage and the of the generated XML document during retrieval.



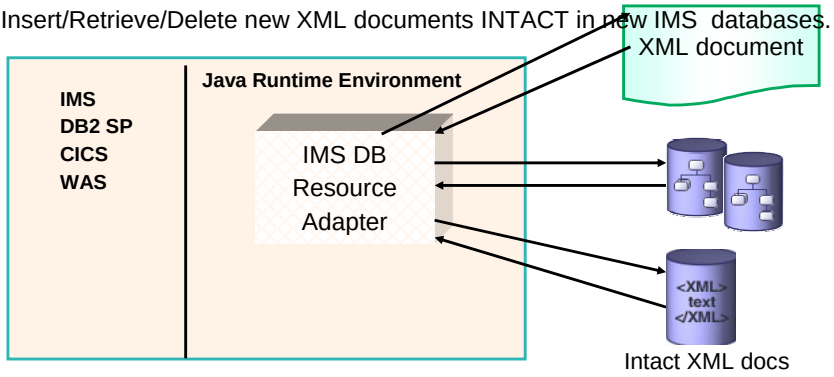
Sample screen shot of IMS Perspective

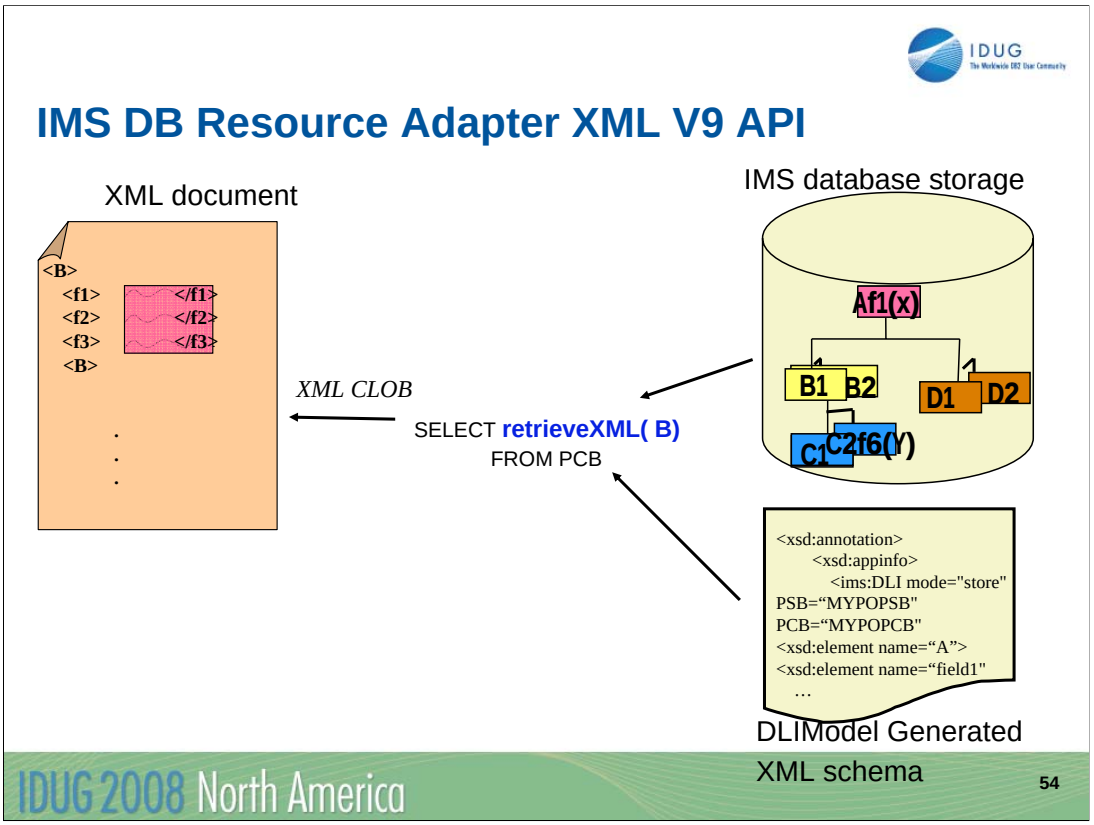


IMS XML DB allows applications to view a traditional IMS database as an XML database and to use an IMS database to store XML documents.

IMS DB Resource Adapter XML-DB

- Retrieve - Compose XML document from any existing traditional database.
- Insert - Decompose XML documents back into same DB.
- Insert/Retrieve/Delete new XML documents INTACT in new IMS databases.





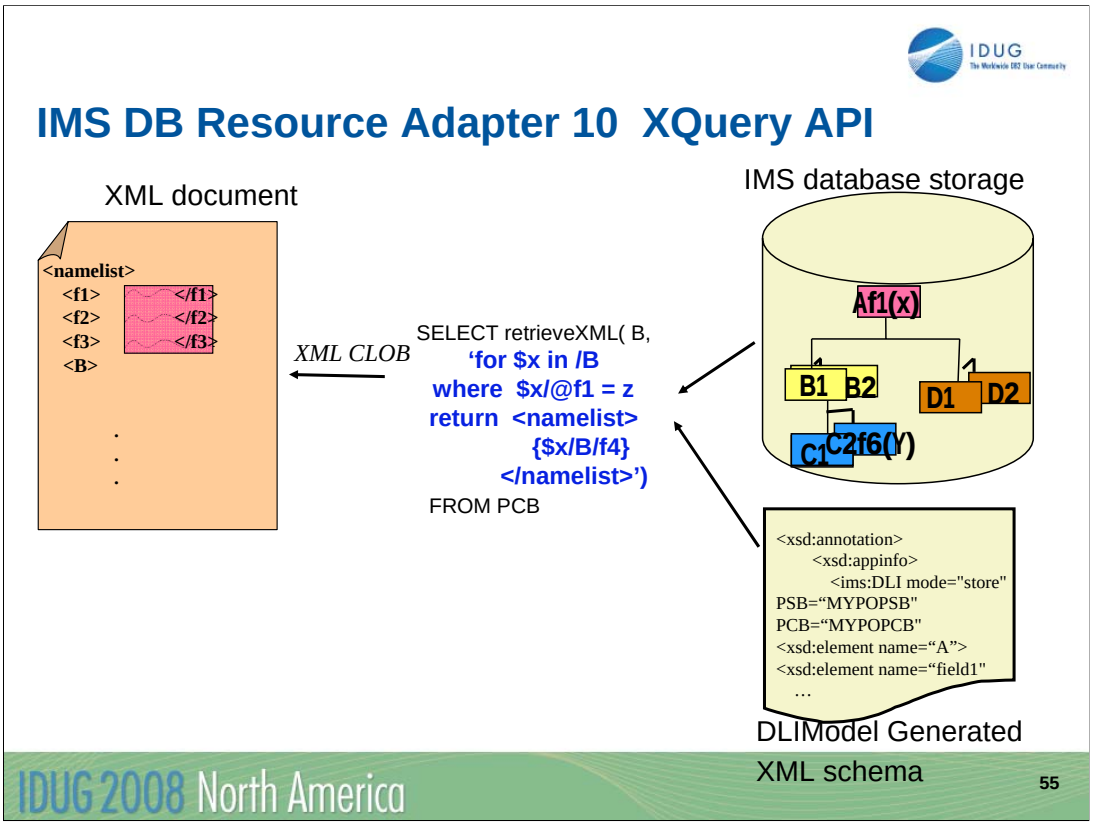
An XML document can be composed from an IMS database.

The retrieveXML is used to access the IMS database and create an XML document based on the DLIModel utility generated XML Schema. The XML root element segment is provided as input parameter to the retrieveXML UDF. The FROM clause identifies which PCB in the PSB is to be used and the lowest level segment in the path. The WHERE clause predicate qualifies which segments to be used to build the XML document.

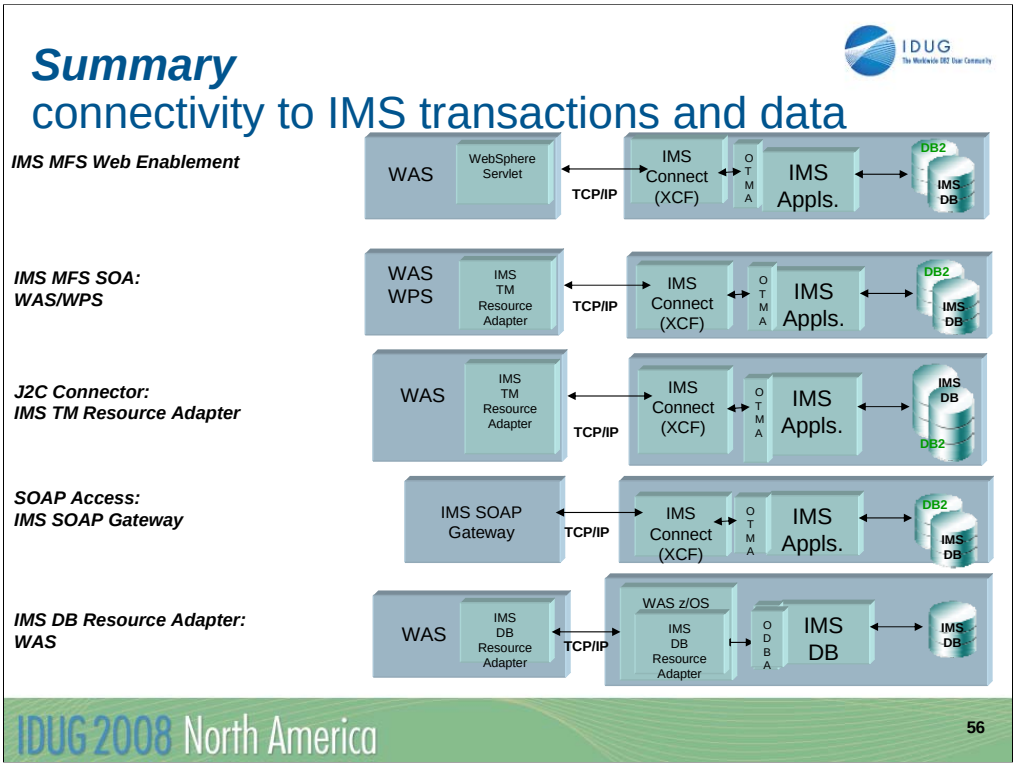
During XML retrieval, IMS database segments are retrieved, fields are converted to the destination XML encoding, tags and XML syntactic information defined in the DLIModel generated XML schema are added, and the XML document is composed.

The Root of an XML document can be any segment in the IMS hierarchy.

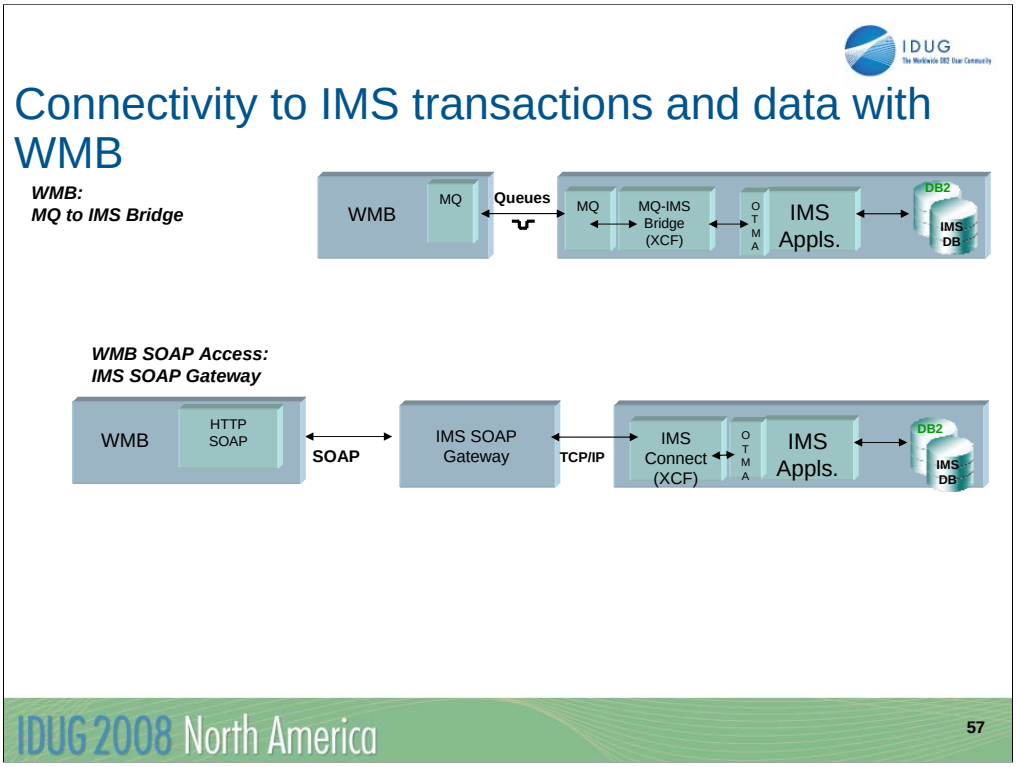
Note the retrieveXML is used to access the IM DB and create an XML document based on the DLIModel Utility generated XML Schema, but you can only retrieve the selected IMS records and build an entire XML document from the IMS records.



Since IMS XQuery is an extension to IMS XML DB, existing DLIModel generated XML Schemas can be used by the IMS XQuery processor to compose XML documents.



Key Message: IMS is providing a variety of solutions for providing access to IMS applications and data as web services.



Key Message: IMS is providing a variety of solutions for providing access to IMS applications and data as web services.

Summary

- IMS and SOA are a team
- Architected interfaces support standard access from the web
- New interfaces, products and tools from a variety of vendors provide access to IMS transactions and data

J13

IMS SOA Integration Suite 10 Enhancements

Kenny Blackman

IBM

kblackm@us.ibm.com