

Unit 9 – 2

More IMS Connectivity (cont.)

WebSphere MQ IMS Bridge - output msgs

■ IMS Output Messages

- All IMS message segments are assembled into a single MQ message which must not exceed the MAXMSGLEN parameter of the queue manager
- If the PUT to the reply-to queue fails, the message goes on the DLQ
- If the PUT to the DLQ fails, a NAK is sent to IMS and the message remains in IMS
 - When the problem has been corrected use
ALTER QL(reply-to queue) GET ENABLE
to restart the flow of messages

WebSphere MQ IMS Bridge - output msgs - 2

■ IMS Output Messages

- IOPCB Output reply messages are placed on the MQ reply-to queue specified in the original message header
 - This could be a local queue or a remote queue
- If you do nothing ALTPCB output from an IMS Bridge initiated transaction will also be placed on the MQ reply-to queue specified in the original message header
 - This can be overridden by the OTMA routing exits or OTMA Descriptors

WebSphere MQ IMS Bridge - output msgs - 3

■ IMS ALTPCB output

- In order to send ALTPCB output to MQ over the IMS Bridge you must build (or override) the OTMA user data in the OTMA DRU exit
- If the message did not originate from the WebSphere MQ IMS Bridge you must build the entire OTMA user data
- If the message did originate from the WebSphere MQ IMS Bridge the OTMA user data is set to return the message to the MQ reply-to queue specified in the original message header when the exit is invoked
 - This can be changed by updating the OTMA user data

WebSphere MQ IMS Bridge - output msgs - 4

- The format is of the OTMA User Data going to the WebSphere MQ IMS Bridge is: LL
| | MQMD | | Reply-to Format
- The LL must be exactly 2 (LL) + L'MQMD + 8 (Reply-to Format) or MQ will ignore the entire MQMD and use default values
 - You must code DCLVER=SPECIFIED,VERSION=1 on the CMQMDA macro
 - May also contain MQSeries User Headers
 - The LL would be variable in this case
 - The User Headers are placed at the end of the OTMA User Data

WebSphere MQ IMS Bridge

ALTPCB Output

- The WebSphere MQ sample exits for DFSYPRX0 and DFSYDRU0 in Appendix B of the Websphere MQ for z/OS System Setup Guide are very useful
 - Follow this code VERY carefully

WebSphere MQ IMS Bridge

ALTPCB Output (2)

- **What you are building in the OTMA User Data is actually an INPUT MQMD as if the message had come from MQ originally**
 - You must get this mindset
 - The Reply-to Queue and Reply-to Queue Manager in the MQMD become the message destination
 - There is no official way to provide a RTQ and RTQM to the receiving MQ application
 - Try using MQMD_APPLIDENTITYDATA
 - Consider using an MQ User Header

WebSphere MQ IMS Bridge

ALTPCB Output (3)

- **The MSGID and CORRELID are the INPUT MSGID and CORRELID - not necessarily what will be presented to the MQ application**
 - You must use the MQMD_REPORT field to specify that the input MSGID and CORRELID be moved to the corresponding output MQMD
 - The default is that the input MSGID is moved to the output CORRELID and a new MSGID is created

WebSphere MQ IMS Bridge

ALTPCB Output (4)

- **Setting the format name for MQ to use for asynchronous OTMA output messages is not straightforward**
 - If you are sending back LLZZdataLLZZData as is probably the case you need to set the format name to MQFMT_IMS_VAR_STRING (“MQIMSVS”)
 - The “Reply-to Format” name in the OTMA User Data is the obvious choice
 - But - the “Reply-to Format” in the OTMA User Data is ignored unless the original message was MQPUT with an MQIIH IMS prefix and the Reply-to Format in the MQIIH was valued

WebSphere MQ IMS Bridge

ALTPCB Output (5)

- **But - there was no input message – how do you tell MQ there was an MQIIH**
 - Remember you are building an INPUT MQMD in the OTMA User Data
 - Lie – set the MQMD_FORMAT to MQFMT_IMS (“MQIMS”)
 - This tells MQ the “input message” had an MQIIH
 - Then set the Reply-to Format name in the OTMA User Data to “MQIMSVS”
 - AND THIS DOES NOT WORK EITHER!!
 - MQ will create an MQIIH in front of the output message
 - > Your application is probably not expecting this
 - > The MQMD_FORMAT field will remain MQ_IMS (“MQIMS”)
 - > The Reply-to Format (“MQIMSVS”) will be in the MQIIH

WebSphere MQ IMS Bridge

ALTPCB Output (6)

- **If the Reply-to Format is not valid as previously described or is blank then field TMAMHMAP in the OTMA State Prefix is used as the MQ Format Name**
 - This is set by the MODNAME parameter in the IMS ISRT call, if present
 - CALL “CBLTDLI” USING ISRT ALTPCB MSGIO MODNAME
 - Do you trust your application to do this properly?

WebSphere MQ IMS Bridge

ALTPCB Output (7)

- **The TMAMHMAP field can be overridden in the OTMA Input/Output Edit Routine (DFSYIOE0)**
 - This lets your application be output independent
 - It can set a valid MODNAME in case the message is going to a terminal
 - The DFSYIOE0 exit can override it to “MQIMSVS”
 - If TMAMHMAP is blanks or nulls or MQIMSNON the MQSeries Format name is set to MQFMT_NONE (blanks)
 - If output is LLZZdata this will cause problems

WebSphere MQ IMS Bridge

ALTPCB Output (8)

- **Determining the MQ Persistence for ALTPCB output messages from OTMA to MQ is affected by commit mode**
 - It is also different if the ALTPCB output is going to a TPIPE originally created by MQ
 - This can happen if the CHNG call destination or the DFSYPRX0 exit or the OTMA DRU exit set the TPIPE name to an MQ created TPIPE name
 - The MQMD_PERSISTENCE field in the OTMA user data is NOT used in determining asynchronous output message persistence – it is ignored

WebSphere MQ IMS Bridge

ALTPCB Output (9)

■ Persistence Determination

- If the output is on a TPIPE created by MQ
 - If the commit mode is 1 (send-then-commit) then the output message is persistent
 - If the commit mode is 0 (commit-then-send) then persistence is determined by message recoverability
 - If the message is “recoverable” then the output message is persistent
 - If the message is not “recoverable” then the output message is nonpersistent
 - A message is “recoverable” to MQ if it is sent on a SYNChronized TPIPE AND has a valid sequence number (TMAMCRSQ not equal 0)
 - Not IMS RECOVERable

WebSphere MQ IMS Bridge

ALTPCB Output (10)

■ Persistence Determination – Phase 1

- If the output is on a TPIPE not created by MQ
 - If the commit mode is 1 or flag TMAMCASY is set in the OTMA State Header then the output message is persistent
 - If the commit mode is 0 or the TMAMCASY flag is not set then persistence is determined by message recoverability
 - If the message is “recoverable” then the output message is persistent
 - If the message is not “recoverable” then the output message is nonpersistent
 - TMAMCASY indicates “asynchronous/unsolicited queued messages”
 - It may be set on for some IMS DFS messages
 - These messages must be treated as CM0 even if the input was CM1

WebSphere MQ IMS Bridge

ALTPCB Output (11)

■ Persistence Determination

- The commit mode is 1 and TMAMCASY is not set
 - The message is persistent

WebSphere MQ IMS Bridge

ALTPCB Output (12)

■ Persistence Determination

- There are two ways to make an ALTPCB TPIPE SYNChronized when it is created
- The OTMASP parameter in DFSPBxxx can be used to default newly created TPIPEs to SYNChronized
 - This parameter was created for MQ users who wanted persistent output messages but did not want to write OTMA DRU exits

WebSphere MQ IMS Bridge

ALTPCB Output (13)

■ Persistence Determination

- The OTMA DRU exit can specify that a newly created TPIPE is SYNChronized
 - Turn on the high-order bit in the first flag in the member override area
 - But this is not enough
 - There is still no valid sequence number
- There is a second bit in the first flag of the member override area to also indicate that a valid sequence number is required
 - MVI OUTFLAG,X'C0'

WebSphere MQ IMS Bridge

ALTPCB Output (14)

- **MQMD as seen by the MQGET for OTMA output**
 - MQMD_REPORT = (see previous)
 - MQMD_MSGTYPE = MQMT_REPLY (x'02')
 - PK61626 allows user to override in returned MQMD
 - MQMD_FORMAT = (see previous discussion)
 - MQMD_PERSISTENCE = (see previous)
 - MQMD_MSGID = (see previous)
 - MQMD_CORRELID = (see previous)
 - MQMD_REPLYTOQ = blanks
 - MQMD_REPLYTOQMGR = MVS Queue Manager

WebSphere MQ IMS Bridge

ALTPCB Output (15)

- **MQMD as seen by the MQGET (continued)**
 - MQMD_USERIDENTIFIER = blanks
 - Unless you explicitly value it in the OTMA DRU exit
 - or the input message came from the IMS Bridge
 - You can copy the USERID passed in the OTMA DRU exit PARMLIST
 - MQMD_ACCOUNTINGTOKEN = Garbage
 - MQMD_APPLIDENTITYDATA = blanks
 - Unless you explicitly value it in the OTMA DRU exit
 - or the input message came from the IMS Bridge and it was valued by the application that did the MQPUT
 - Then MQ will pass whatever you put in there
 - MQMD_PUTAPPLTYPE = MQAT_XCF (x'14')
 - MQMD_PUTAPPLNAME =
 - 8 Byte XCF Group Name
 - 16 Byte XCF Member Name of the sending IMS

WebSphere MQ IMS Bridge

XML MSGs

- **Many customers would like to send XML messages to the IMS bridge and receive XML messages on output**
 - They do not want the MQ application to have to know anything about IMS conventions
 - LLZZ
 - Transaction Code
 - Segmentation
 - MQ IMS Formats
 - IMS Lab devt. Is working hard at trying to make this easily doable within MQ or IMS OTMA
 - Until this happens there is less easy way to do this

WebSphere MQ IMS Bridge

XML MSGs (2)

- **This is a way to process XML messages to and from the IMS bridge ..**
 - The MQ application sends the XML message to a local (non-IMS Bridge queue) on z/OS
 - A long running “input” batch job will do an MQGET with Wait on this XML input queue
 - The wait time should be shorter than the z/OS wait time to avoid ABENDS522
 - If the time expires just reissue the MQGET with wait
 - The input batch job should recognize a “shutdown” message so that it can be cleanly ended
 - It should also recognize the error code when a “GET DISABLE” command is issued against the input queue and shutdown cleanly

WebSphere MQ IMS Bridge

XML MSGs (3)

■ This is a way ...

- The input batch job formats the input message for IMS
 - Add the LLZZ
 - Extract the transaction code from the XML message and add it to the front of the message
 - Or determine the transaction code in some other manner
 - If the message is greater than 32K break the input message into multiple IMS input segments
- The input batch program could optionally marshal the XML and pass a plain input message to IMS

WebSphere MQ IMS Bridge

XML MSGs (4)

- **This is a way ...**

- It is important to preserve the original reply-to queue and queue manager from the original message so that the reply message can be sent properly
 - Copy these from the input MQMD and send them to OTMA in the OTMA User Data by using an MQ User Header
 - IMS will pass back the OTMA User Data in the output OTMA prefix
 - MQ will put it on the reply-to queue

WebSphere MQ IMS Bridge

XML MSGs (5)

- **This is a way ...**

- The input batch program should then MQPUT the message to the IMS Bridge Queue
 - Set up the MQMD for the IMS Bridge
 - Set the reply-to queue and queue manager to a “local output queue”
- The format of the input message is
 - MQUserHeader+(MQIIH)+LLZZ+TRANCODE+b+Data (XML)

WebSphere MQ IMS Bridge

XML Messages (6)

- **This is a way ...**
 - MQ will send the message to IMS via OTMA
 - The IMS transaction will process the message and insert the reply to the IOPCB
 - OTMA will send the reply message to MQ
 - MQ will place the output message on the “local output queue” specified on the input MQMD
 - This includes the input MQ User Header with the reply-to queue and queue manager

WebSphere MQ IMS Bridge

XML Messages (7)

■ This is a way ...

- A long running “output” batch job will do an MQGET with Wait on this “local output queue”
 - The wait time should be shorter than the z/OS wait time to avoid ABENDS522
 - If the time expires just reissue the MQGET with wait
- The output batch job should recognize a “shutdown” message so that it can be cleanly ended
 - It should also recognize the error code when a “GET DISABLE” command is issued against the input queue and shutdown cleanly

WebSphere MQ IMS Bridge

XML Messages (8)

- **This is a way ...**
 - The output batch job formats the output message for XML
 - Remove the LLZZ
 - If this is a multi-segment output message remove the rest of the LLZZ's
 - Reformat the message into XML if the application sent plain text

WebSphere MQ IMS Bridge

XML Messages (9)

- **This is a way ...**
 - The output batch job then inserts the output XML message to the original reply-to queue and queue manager
 - These are found in the MQ User Header
 - For ALTPCB output the OTMA DRU exit will have to create this MQ User Header

WebSphere MQ IMS Bridge Security

- **Requires RACF 1.9.2 (or equivalent)**
- **Three levels of security**
 - 1. Connections from MQ to IMS (during client bid)
 - Uses the RACF Facility class
 - Not used if OTMA security is set to NONE
 - IMS checks IMSXCF.XCFgroupname.MQ-member-name
 - MQ must have read access to this class
 - MQ checks IMSXCF.XCFgroupname.IMS-member-name
 - The WebSphere MQ subsystem Userid access to this class determines Userid validation for each message that crosses the IMS Bridge

WebSphere MQ IMS Bridge Security (2)

■ Three levels of security (continued)

— 2. Userid Validation in MQ

- MQMD.UserIdentifier field is used
- Based on MQ Subsystem userid access to IMSXCF.XCFgroupname.IMS-member-name
 - CONTROL/ALTER: Userids trusted, no checks
 - UPDATE: Userid validated by RACF prior to passing to IMS
 - > Result of this check is cached and used on subsequent calls
 - READ: Userid/password validated by RACF prior to passing to IMS
 - > Result of this check is cached and used on subsequent calls
 - NONE (or no profile): Userid/password validated by RACF prior to passing to IMS
 - > No cache is performed
- Once validated, a UTOKEN is passed to IMS and can be used for normal IMS security

WebSphere MQ IMS Bridge Security (3)

■ Three levels of security (continued)

— 2. Userid Validation in MQ

- MQMD.UserIdIdentifier field is used
- If MQMD.UserIdIdentifier is blank WebSphere MQ will use the RACF default
 - This is obtained from RACF
 - This could cause security problems in IMS
- APAR PM68949 will change this
 - If the Queue Manager has UPDATE access the message will be placed on the Dead Letter Queue with a new reason code

WebSphere MQ IMS Bridge Security (4)

■ Three levels of security (continued)

— 2. Userid Validation in MQ

- The amount of time a UTOKEN in the MQ cache is valid is set by the MQ ALTER SECURITY command
 - ALTER SECURITY INTERVAL(m) TIMEOUT(n)
 - > INTERVAL = How often (in minutes) to check for expired UTOKENS
 - > TIMEOUT = How long a UTOKEN (in minutes) is valid
 - > See MQ TECHNOTE 1270319 for more information
- The MQ RVERIFY SECURITY command will expire all UTOKENs
- If the MQ Subsystem Userid access to
`IMSXCF.XCFgroupname. IMS-member-name`

is updated the IMS OTMA connection must be stopped and started for it to take effect !

- /STO OTMA - stops ALL OTMA connections – not just this MQ
- /STA OTMA

WebSphere MQ IMS Bridge Security (5)

■ Three levels of security (continued)

— 2. Userid Validation in MQ

- WebSphere MQ has a ZPARM parameter RESAUDIT(YES|NO)
- This specifies whether or not RACF audit records are written for MQ
- This parameter does NOT apply to the IMS Bridge
 - RACF audit records for the IMS Bridge can not be turned off

WebSphere MQ IMS Bridge Security (6)

■ Three levels of security (continued)

— 3. OTMA Security (set by /SECURE OTMA)

- CHECK
 - Existing RACF calls are made
 - IMS commands are checked against the CIMS class
 - IMS transactions are checked against the TIMS class
- FULL
 - Same as CHECK, but the ACEEs are built in the dependent regions as well as the control region
 - > Default at IMS cold start
- NONE
 - No calls to RACF are made
- PROFILE
 - Each message defines the level of security checking to be done
 - `MQIIH.SecurityScope` field allows for FULL or CHECK to be specified

WebSphere MQ IMS Bridge Security (7)

- **WebSphere MQ security profiles in the MQADMIN class**
 - If `subsysid.NO.SUBSYS.SECURITY` is defined...
 - MQ does not pass Userid to IMS
 - Client bid only succeeds when /SEC OTMA NONE is in effect (no Userid is passed to the IMS transaction)
 - Good for early testing

WebSphere MQ IMS Bridge Security (8)

- **WebSphere MQ security profiles in the OPERCMDS class**
 - If you are using RACF to protect resources in the OPERCMDS class
 - Make sure that MQ has the authority to issue the MODIFY command to an IMS system to which it might connect

WebSphere MQ IMS Bridge - Requirements

- **These are some of the requirements which customers have wanted for the MQSeries IMS Bridge**
 - Use the XCF Group name on the Storage Class definition
 - Allow Member name, DRU exit name, ACEE aging value and TPIPE prefix to be specified at the Storage Class level
 - Allow the OTMA DRU exit to easily specify the MQ format name
 - Capture SMF statistics for the IMS Bridge
 - Support Confirm on Delivery
 - Support OTMA transaction expiration
 - Support Version 2 MQMD

WebSphere MQ IMS Bridge Information Sources

<https://www.ibm.com/products/mq>

There are some useful Supportpacs / blogs / websites , as of this one:

[MP16](#) – *Capacity Planning and Tuning for MQSeries on z/OS*
The new version does not have IMS performance data
This may be corrected in the future

(older stuff : <http://www.redbooks.ibm.com/> (REDP-3959-00) , WebSphere for z/OS to CICS and IMS Connectivity Performance)

WebSphere MQ Connection to IMS

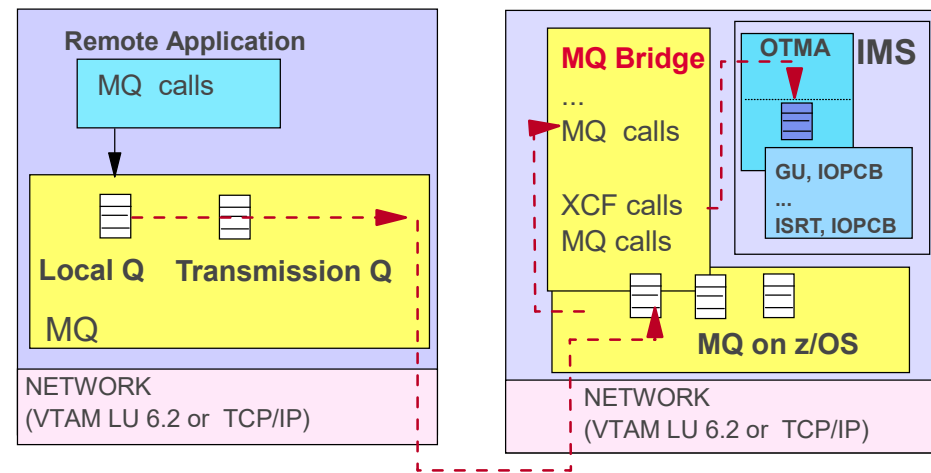
- Supports the use of MQ API
 - Remote program is not sensitive to the network typeMQ provides its own high-level standard API
- Supports the use of JMS (Java Message Service) API
 - Messaging standard that allows application components based on J2EE to create, send, receive, and read messages

MQ IMS Adapter

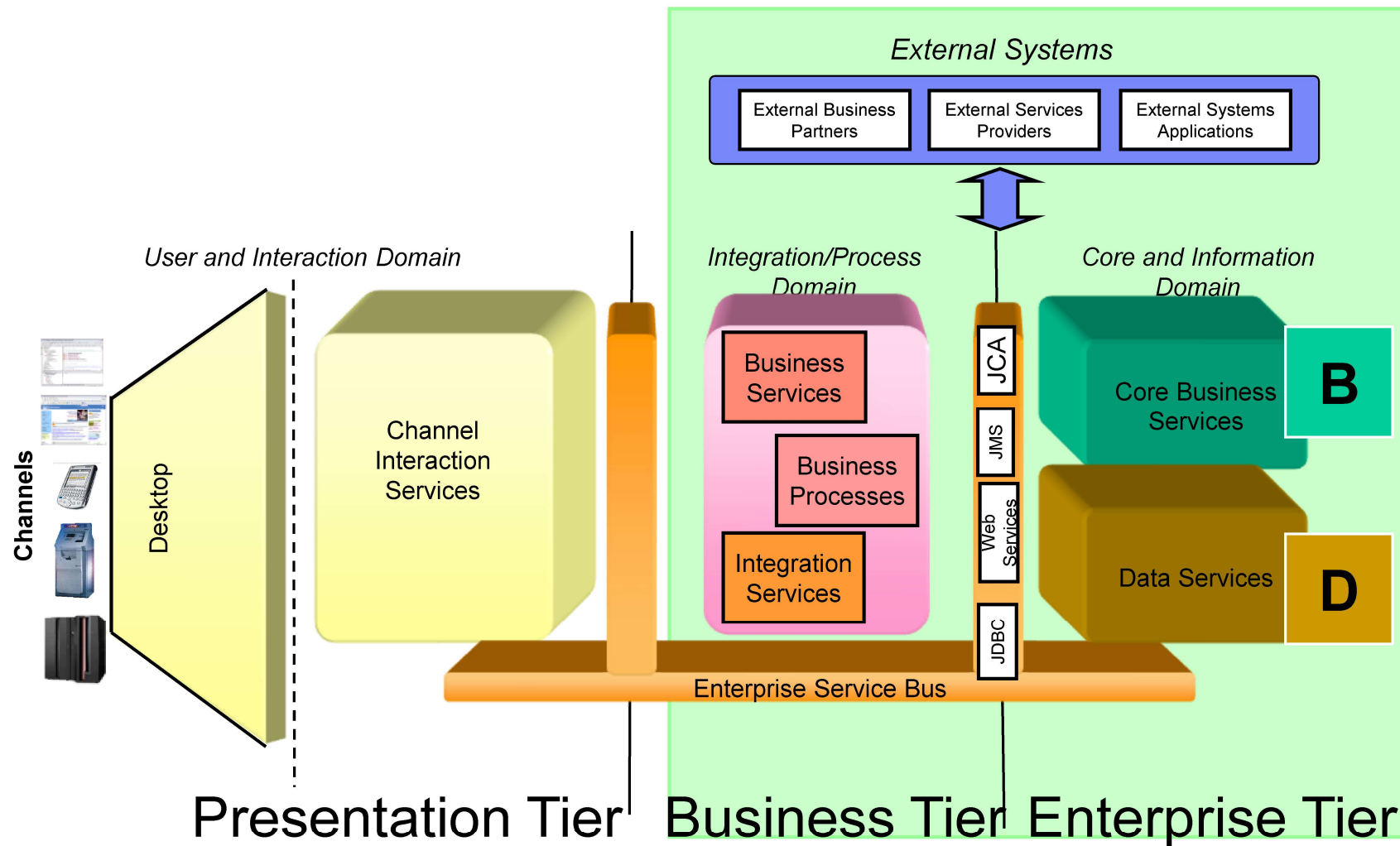
- uses the IMS ESS interface
 - *Supports the use of explicit MQ calls in the IMS application*

MQ IMS Bridge

- uses the OTMA interface
 - *Takes advantage of the DL/I call interface in the IMS application*



Positioning IMS Assets in SOA Architecture



SOA with IMS Applications

- **When designing an SOA, much of the business logic to be deployed as services is already implemented in existing IT application systems**
 - And much of this runs on the mainframe - Bottom-Up approach
- **New services may also be written**
 - It might be appropriate to write this as an EJB or Web Service.
 - For example, using IMS JDBC to directly access the IMS Databases.
 - But in many cases the best solution will be to create new IMS transactions – Top-Down approach
 - IMS as high performance business logic container
 - WAS & IMS co-location with WOLA for optimum performance

Business rule mining is the process of extracting essential intellectual business logic in the from packaged or application software, recasting them in natural or formal language, and managing them in a BRMS.

SOA with IMS Applications

- **When the existing transaction does not exactly match the business requirement, the most efficient solution is to modify the existing transaction.**
 - or add COBOL or PL/I logic
 - Add JAVA classes to existing COBOL or PL/I programs
 - Take benefit of a Business Rules management system
 - Business Rules mining using Rational Asset Analyzer
 - Creating rules in COBOL with “Rules for COBOL” feature
 - Or using Callout to execute Rules Services
 - Take benefit of Business Event management system
 - Generate events from IMS application
 - Study all Call in/Callout capabilities
 - Be creative 😊

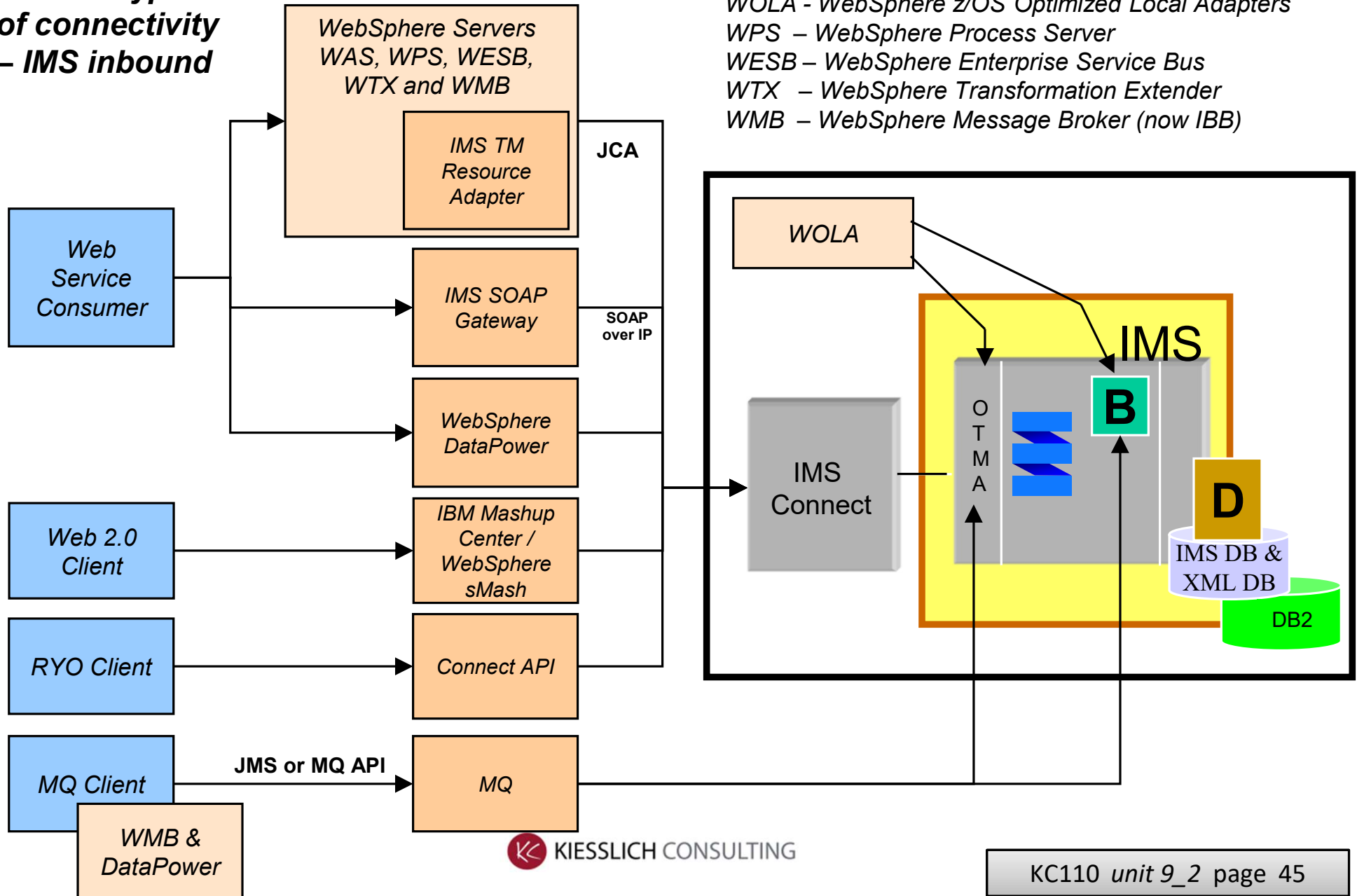
IMS Transactional Program Flows

- **All input messages into IMS go onto a queue**
 - There is one input queue for each transaction code
- **Without OTMA, output messages (ISRT IOPCB) get put onto a queue.**
- **With OTMA, client chooses a COMMIT MODE (1 or 0)**
 - Send reply directly (synchronously), bypassing the queues (called “Commit Mode 1” or “Send then Commit”)
 - If SEND fails, transaction program is backed-out.
 - Synchronisation flow depends on sync-level (None, Confirm, Syncpt)
 - Put reply on a named output queue and send it after application has committed (called “Commit Mode 0” or “Commit Then Send”)
 - If SEND fails, reply is still on queue and can be retrieved later
 - Assured Delivery of replies is available*
 - Sync-level is always Confirm
- **IMS applications can create messages on any named output queue (using alternate PCB)**
 - Client can create CM0 services to read these asynchronous messages (Resume Tpipe)
 - For failed CM0 IOPCB reply or ALTPCB output
 - Integration with Business Event solutions

SOA Connectivity with IMS TM (Inbound)

**Various types
of connectivity
– IMS inbound**

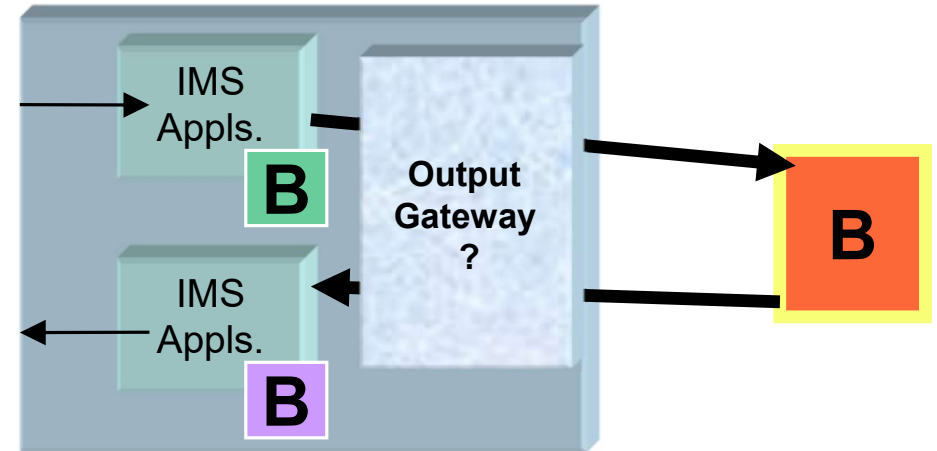
WAS – WebSphere Application Server
WOLA – WebSphere z/OS Optimized Local Adapters
WPS – WebSphere Process Server
WESB – WebSphere Enterprise Service Bus
WTX – WebSphere Transformation Extender
WMB – WebSphere Message Broker (now IBB)



SOA Connectivity with IMS TM (Outbound)

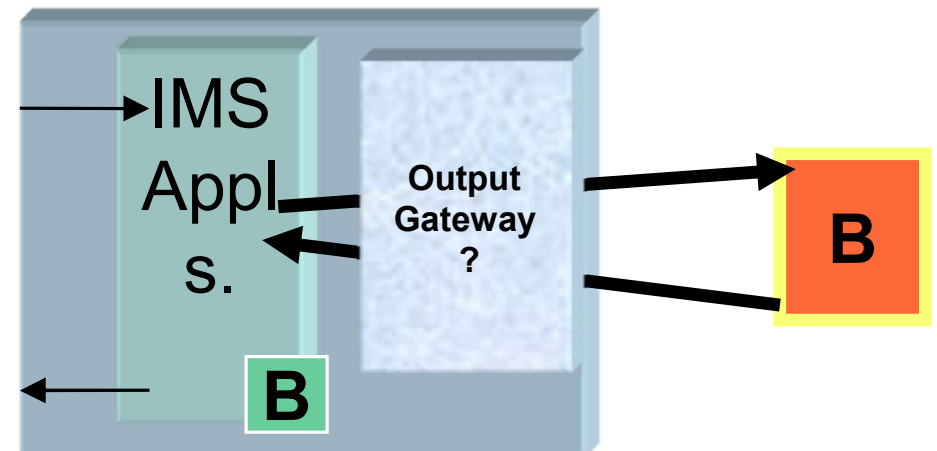
■ Asynchronous support with

- IMS API (ISRT ALTPCB)
- MQ API
- APPC API
- TCP/IP calls with IMS Connect



■ Synchronous (not in 2PC scope) support with

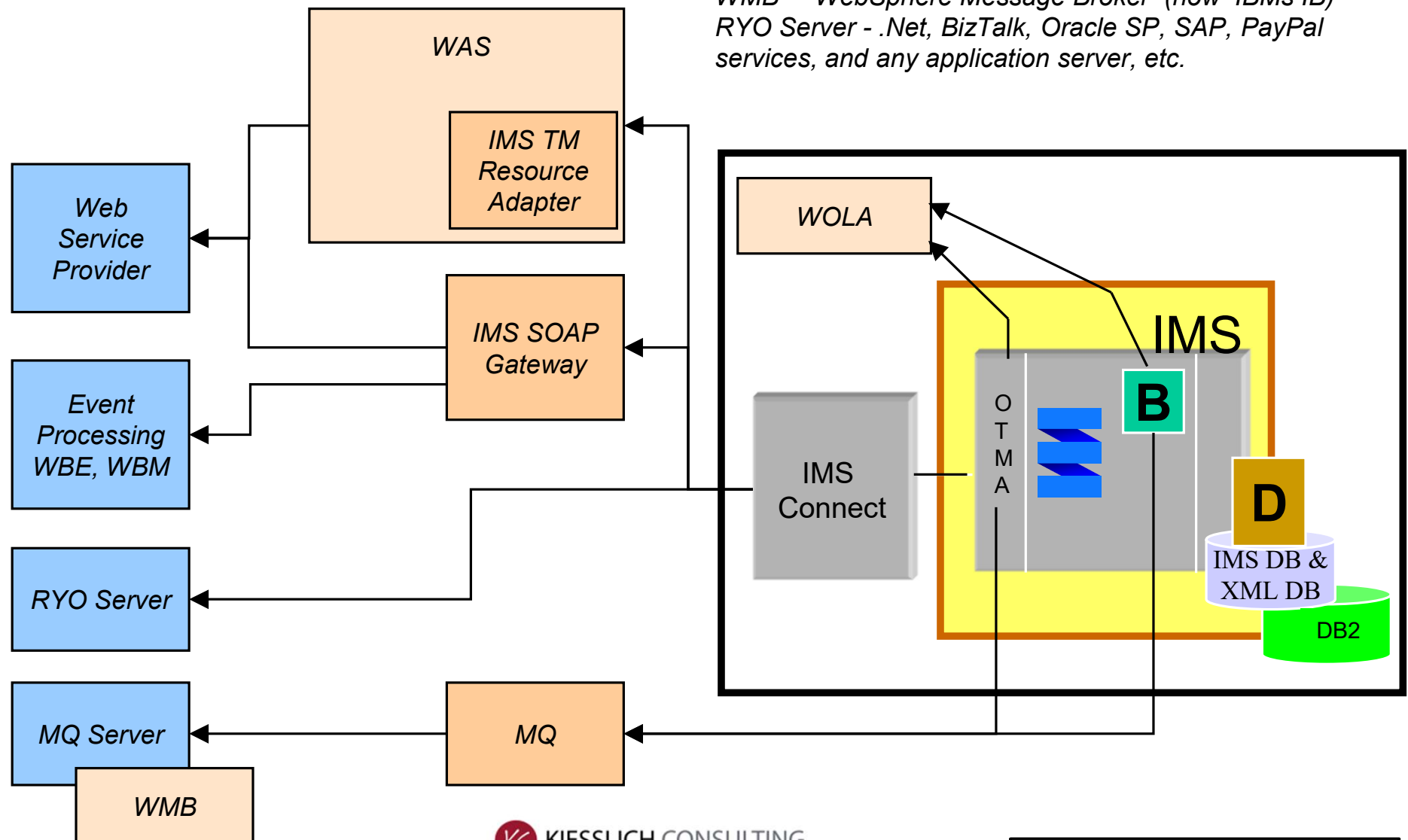
- IMS API – “New” ICAL
- MQ API
- APPC/IMS (also in 2PC scope)
- SQL calls to DB2 StoredProcs



SOA Connectivity with IMS TM (Outbound)..

- Asynchronous and synchronous capabilities

*WAS – WebSphere Application Server
WOLA - WebSphere z/OS Optimized Local Adapters
WBE – WebSphere Business Events
WBM – WebSphere Business Monitor
WMB – WebSphere Message Broker (now IBMs IB)
RYO Server - .Net, BizTalk, Oracle SP, SAP, PayPal services, and any application server, etc.*



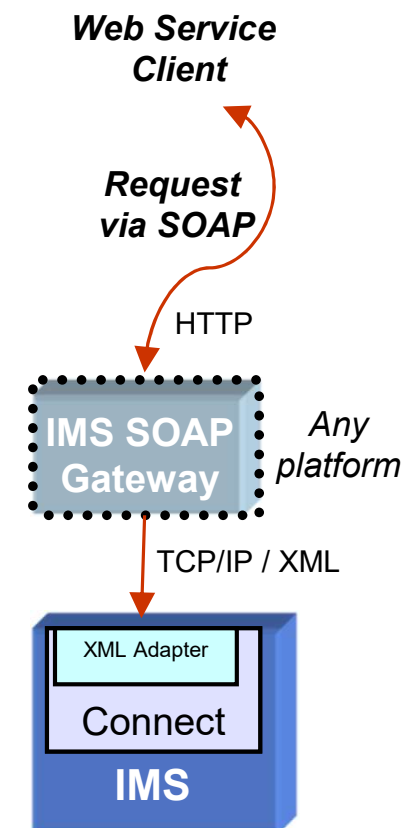
IMS Transactions SOA Integration - Solution Summary

Standard architecture	Middleware	Capabilities	Recommendation
SOAP	IMS Connect (enhanced with IMS Connect Extension Tool) & IMS SOAP Gateway	Synchronous access over HTTP Inbound and outbound (with IMS 10) CM1 with sync-level=None No CM0 support No support for IMS conversational transaction	Still has some limitation today; work with IMS lab if you have specific customer requests
JCA	IMS Connect (enhanced with IMS Connect Extension Tool) WAS Server	Synchronous with Asynchronous output retrieve options Inbound and outbound (with IMS 10) CM0 and CM1 support NO maximum message size: IMS Connect supports multi segment message (32K limit for one single segment)	Most appropriate solution when service requester is JEE component and when high QoS required (2PC, connection pooling, identity propagation etc.)
JMS	MQ IMS Bridge WAS Server	Asynchronous, with almost-synchronous capabilities Inbound and outbound CM0 and CM1 support Assured delivery	Exploit JMS and WMQ for basic messaging and flowing Web services.
DataPower	IMS Connect (enhanced with IMS Connect Extension Tool) DP Appliance	Synchronous and Asynchronous Inbound CM1 with sync-level=None No CM0 support 32 KB limit (single segment) No support for IMS conversational transaction	Use as ESB gateway for security functions, message transformation and routing
WebSphere Message Broker / IBM Integration Bus (IIB)	IMS Connect (enhanced with IMS Connect Extension Tool) WMB server	IMS Connect node available in addition to MQ support	Consider as option to service enable IMS applications when WMB is already used as enterprise ESB.

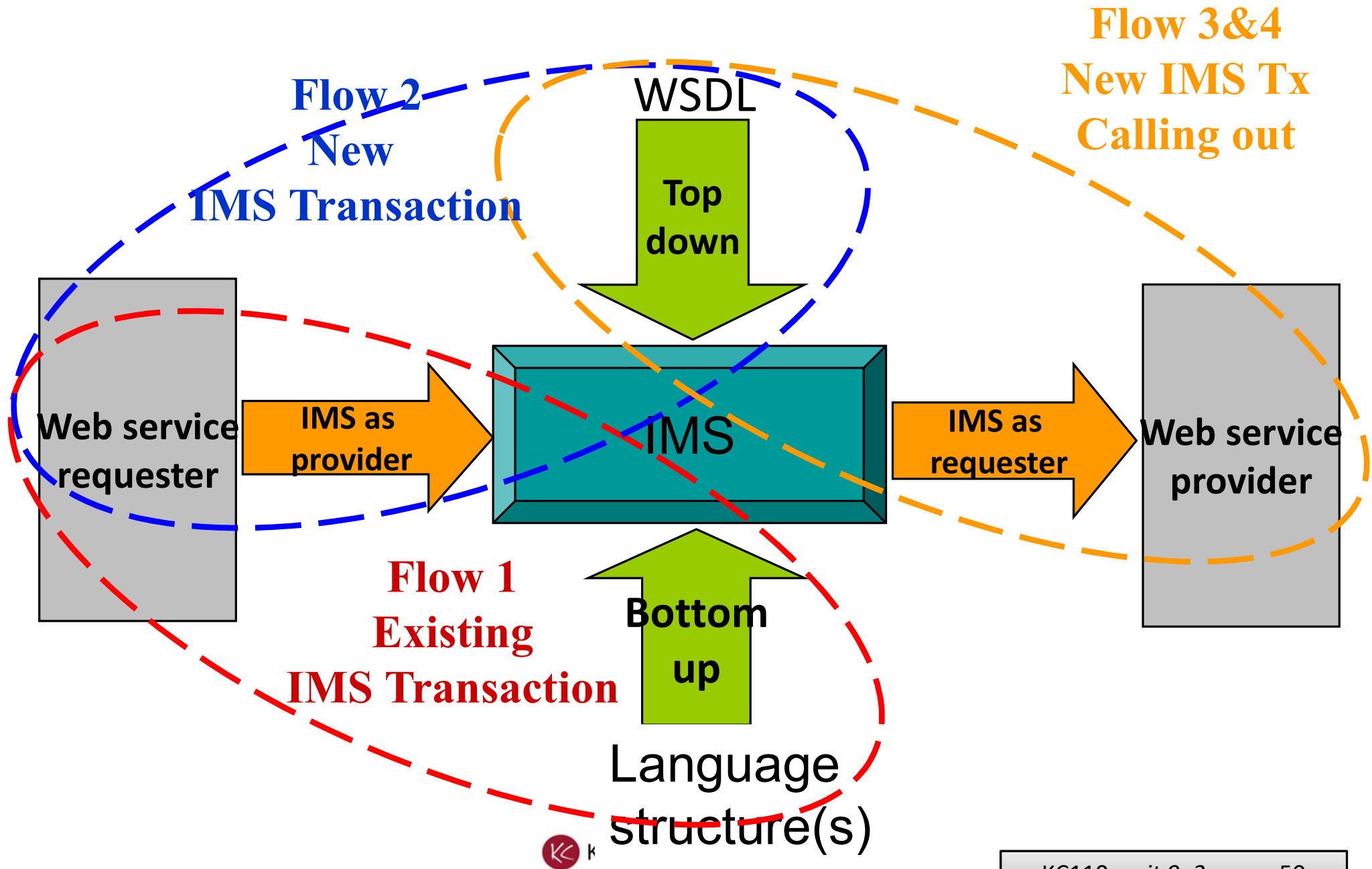
IMS SOAP Gateway

IMS SOAP Gateway is a light-weight application server to offer an easy solution to make IMS Transactions as SOAP Web Services available

- SOAP support for IMS without the need of a full J2EE Server
- Supported on all Runtime Environments (z/OS, zLinux, AIX & Windows)
- IMS SOAP Gateway is for free (part of the IMS Enterprise Suite)
- Offers standardized access ways via Web Services
- Offers easy integration of IMS assets into a SOA environment
- Little to No IMS Specific knowledge needed, just Web Service calls
- SOAP Gateway can act as a Web Service Provider or Requester for IMS
- Easy Integration with Help of Rational Developer for System z (RDz)



SOAP - Flow Overview



IMS Connect API

Simple, easy-to-use, light weight solution for developing client applications that interact with IMS through IMS Connect over a TCP/IP network.

Shields client application developers from the complexities of TCP/IP socket programming, IMS Connect message protocol, and message interpretations. Facilitates remote access to IMS transactions and commands.

To be made available in high level languages:

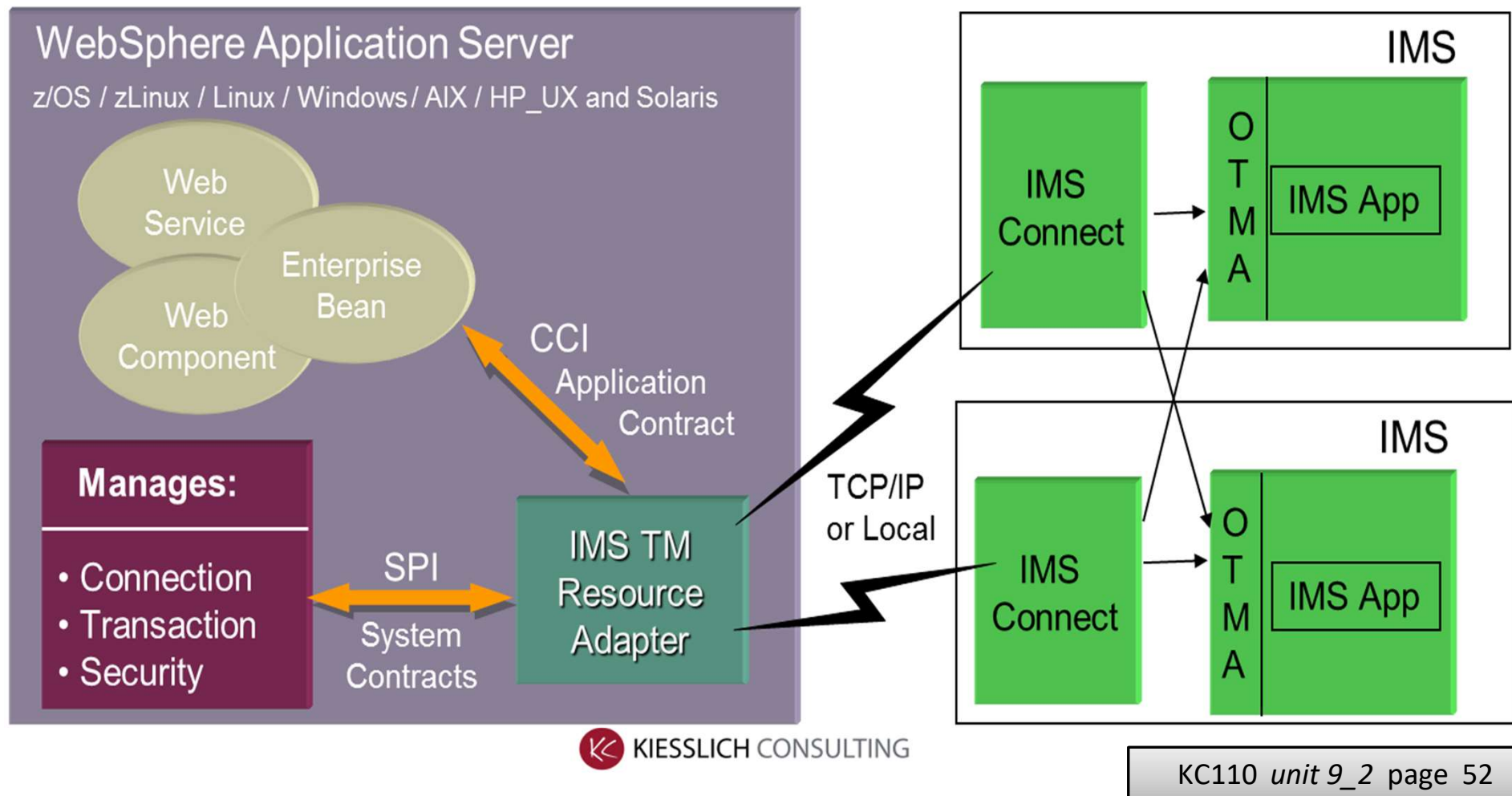
Java

C

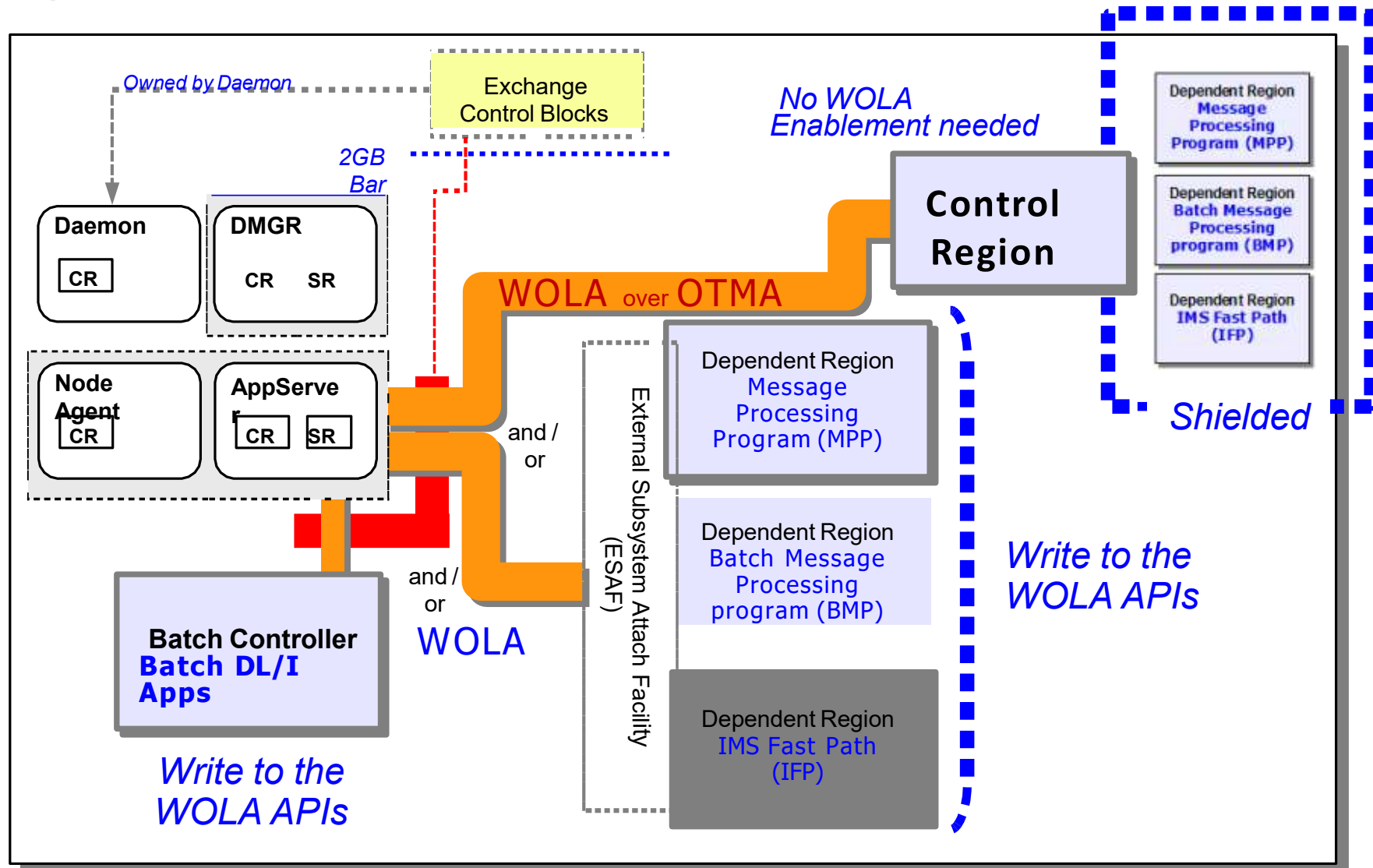
Targets users who need to write and maintain **RYO** applications and utilizing the user message exits SMPL1 or SMPL0.

IMS TM Resource Adapter

- Offers easy IMS integration from different Java platforms like WebSphere
- Handles many things for the application developer like Connection Handling and Security
- Works perfectly together with Rational Tooling and comes with IMS

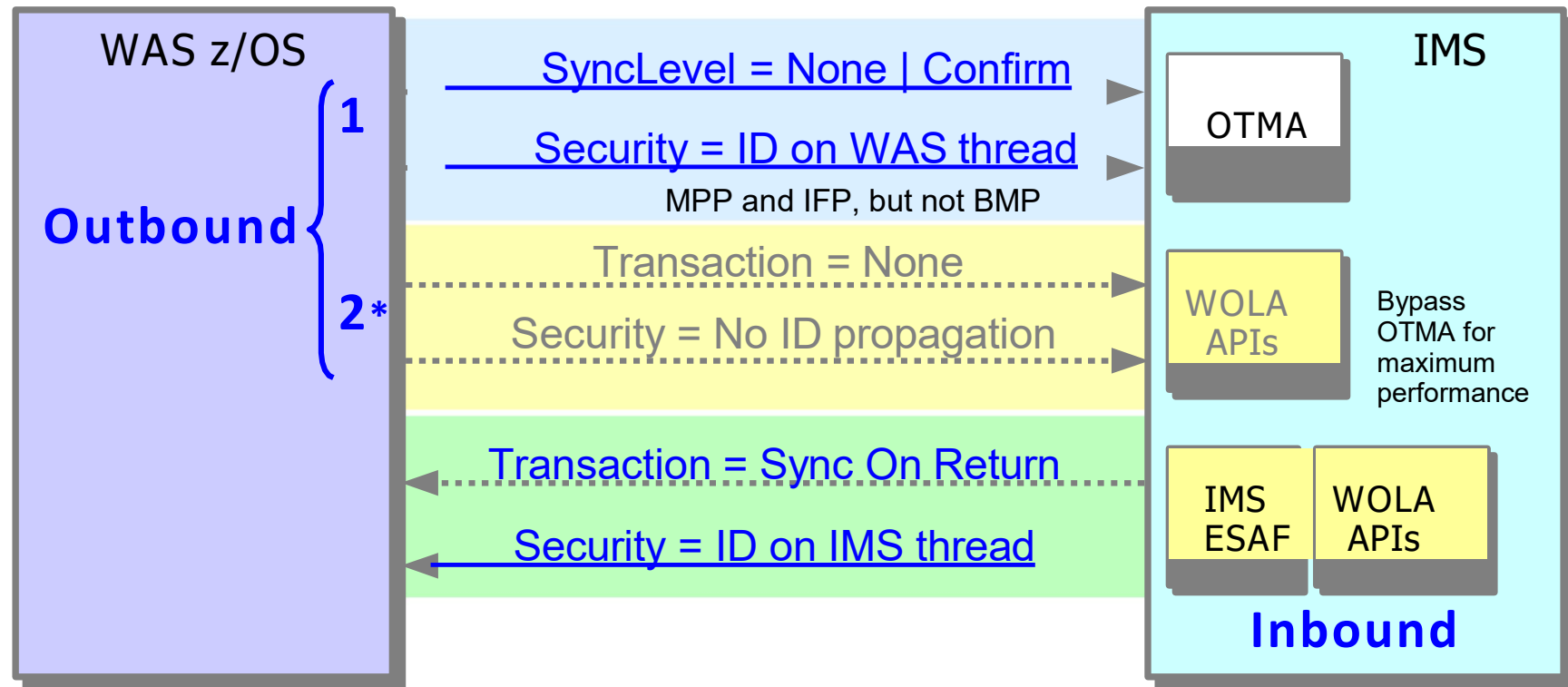


High Level Overview - IMS Support for WOLA



- WebSphere Optimized Local Adapter for WebSphere auf z/OS
- TCP/IP Overhead saved by ASID – ASID communication (Cross Memory); but XM isn't strategic nor supported anymore
- Exploited via OTMA or directly by implementing of OLA APIs

WOLA and IMS – Transaction and Security



(*) OUTBOUND 2 – less or no use