Enterprise Application Modernization with the IMS ICAL

Suzie Wendler – zGrowth Washington Systems Center

Information Management software

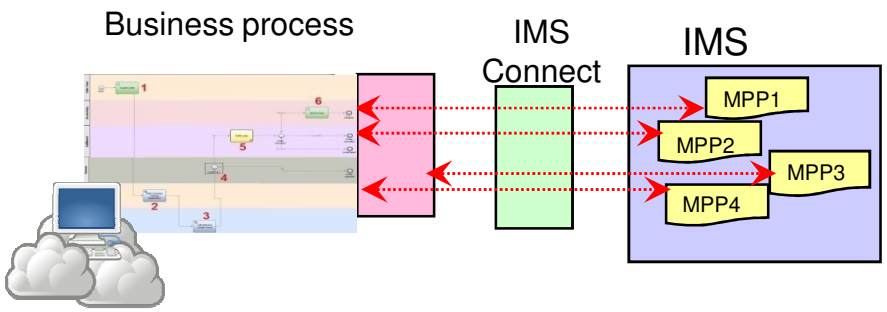
Topics

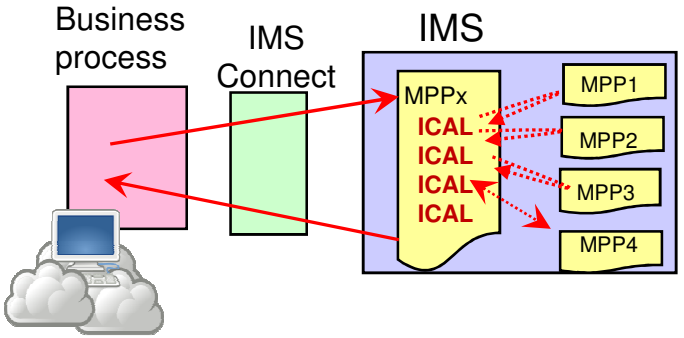
Designing an application with an internal service flow template not only supports flexibility but also facilitates the creation of new business processes. This means, for example, that a control application could call another application, analyze the results and, based on those, call any other application all within the control program's unit of work. IMS can participate in this type of an external flow but can also support the creation of a internal to extend the value of transactions that may have been written yesterday or even decades ago.

- Synchronous callout
- Synchronous program-to-program switch
- Expanding IMS – Creating an Internal Control Flow
- And even… a discussion on implementing business rules

IMS Enhancements

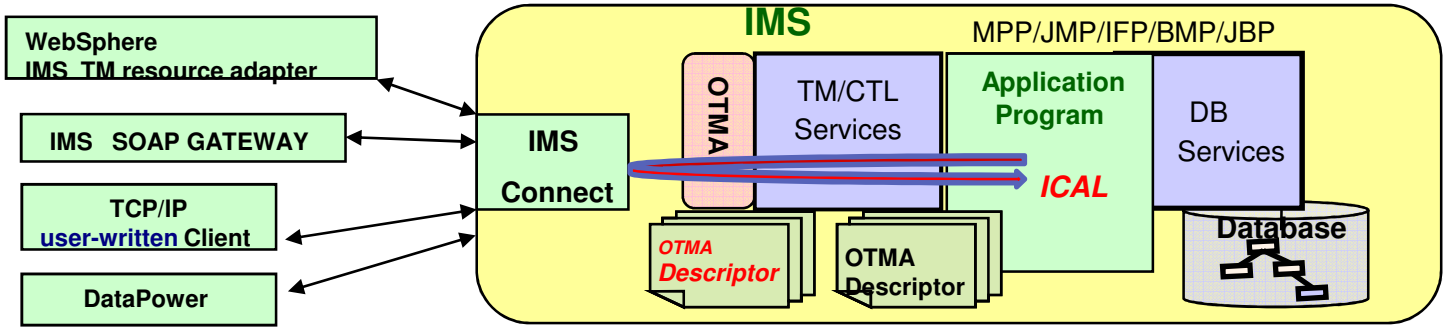
- Modernization of the IMS application infrastructure
  - Implementation of a process server or broker function inside IMS
    - Reduces unnecessary network traffic when accessing multiple applications in the same IMS or IMSplex
    - Provides an internal service flow of IMS transactions to complete a business process
      - In the same IMS or a different IMS
  - *Leverages the IMS DI/I ICAL capability*



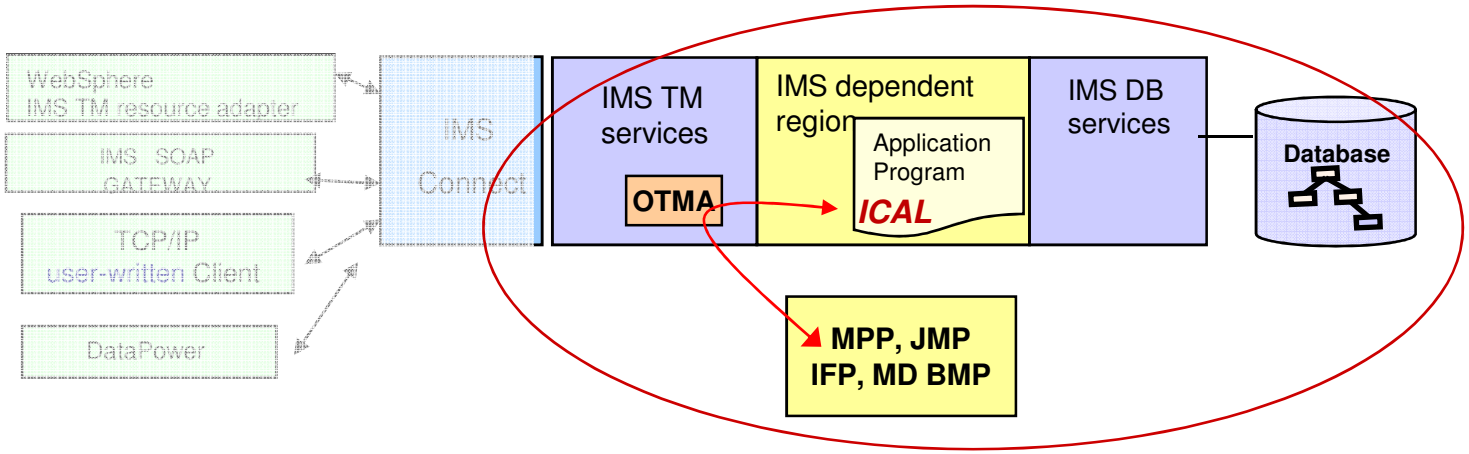


The ICAL

- Originally provided support for IMS synchronous callout capabilities



- And with IMS 13, Synchronous program-to-program switching



The ICAL – First things First

- Original Support for Synchronous Callout (IMS 10 SPE / IMS 11 base)

Call AIBTDLI USING ICAL,*aib*,REQ_area,RESP_area

  - DL/I call verb: ICAL *(sends a message and waits for a response)*
    - Uses the AIB interface

- AIBID = DFSAIBbb
- AIBLEN = AIB length
- AIBSFUNC = SENDRECV or RECEIVE (IMS 13)
- AIBRSNM1 = 8 byte OTMA Descriptor name
- AIBRSFLD = Timeout value
  - 4 byte field for time value 100th seconds. System default is 10 sec.
- AIBOALEN = **REQ_area** length
  - As an input parameter: 4 byte field contains the length of the request area
  - As an output parameter: Actual length of the response message
- AIBOAUSE = **RESP_area** length
  - As an input parameter: 4 byte field contains the length of the response area
  - As an output parameter: Length of the response message.
- AIBRETRN = AIB Return code
- AIBREASN = AIB Reason code.
- AIBERRXT = 2 byte sense code from external application
- …
- AIBMAPNM (IMS 13)
- …

The ICAL – First Things First …

- OTMA Destination Routing Descriptor
  - Defines the message destination and tells IMS how to handle the message

- DFSYDTx member of IMS.PROCLIB
  - TYPE: Destination type
  - TMEMBER: OTMA Target Client
  - TPIPE: Destination Name
  - SMEM: YES|NO
  - ADAPTER: Type of IMS Connect Adapter (for IMS Soap Gateway)
  - CONVERTR: Routine called by Adapter (For IMS Soap Gateway)
  - SYNTIMER=timeout
    - Note ICAL overrides this value

Type -2 Command:

```
UPDATE OTMADESC NAME(OTMASYN) SET(SYNTIMER(5000))
```

- IMS keeps the calling program in a wait state
  - Creates a correlation token which must be passed back to allow the response to be sent to the correct instance of the executing program

- HWSIMSCB Macro

```
*****************************************************
*  CORTKN   Synch Callout Correlator Token
*****************************************************
CORMask     DSECT           Correlator Token dsect
COR_Len     DS   H          Length of COR Structure
COR_Rsvd    DS   H          Reserved
COR_Id      DS   CL8        Str Id *CORTKN* ascii/ebcdic
COR_LL      DS   XL2        Length of Token
COR_RESV1   DS   XL2        Reserved
COR_IMSID   DS   CL4        IMSID
COR_MEMTK   DS   XL8        Member XCF Token
COR_AWETK   DS   XL8        Message Token
COR_TPIPE   DS   CL8        TPIPE Name
COR_USERID  DS   XL8        Userid
CORMask_Len EQU  *-CORMask  Size of COR Structure
```

Client Programming

- IMS applications function as clients to remote web services
  - IMS DL/I call “ICAL” sends messages through IMS Connect to:
    - EJB/MDB on a JEE server, e.g., WebSphere Application Service
    - Soap clients through IMS ES Soap Gateway
    - RYO clients
  - Callout message format indicates Asynchronous or Synchronous to the partner client program

Asynch (ISRT):

LLLL |{*REQMOD*}| LLZZData	{LLZZData} |*CSMOKY*

Synch (ICAL):

LLLL	*CORTKN* |{*REQMOD*}| LLLLData |*CSMOKY*

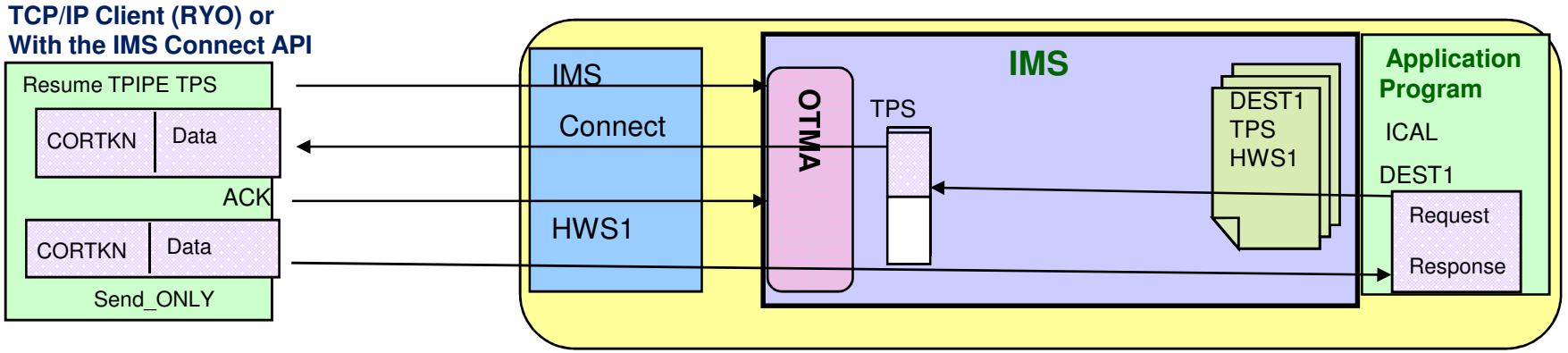
- Remote systems need to issue
  - Resume TPIPE request to retrieve messages

Client programming

- Flow

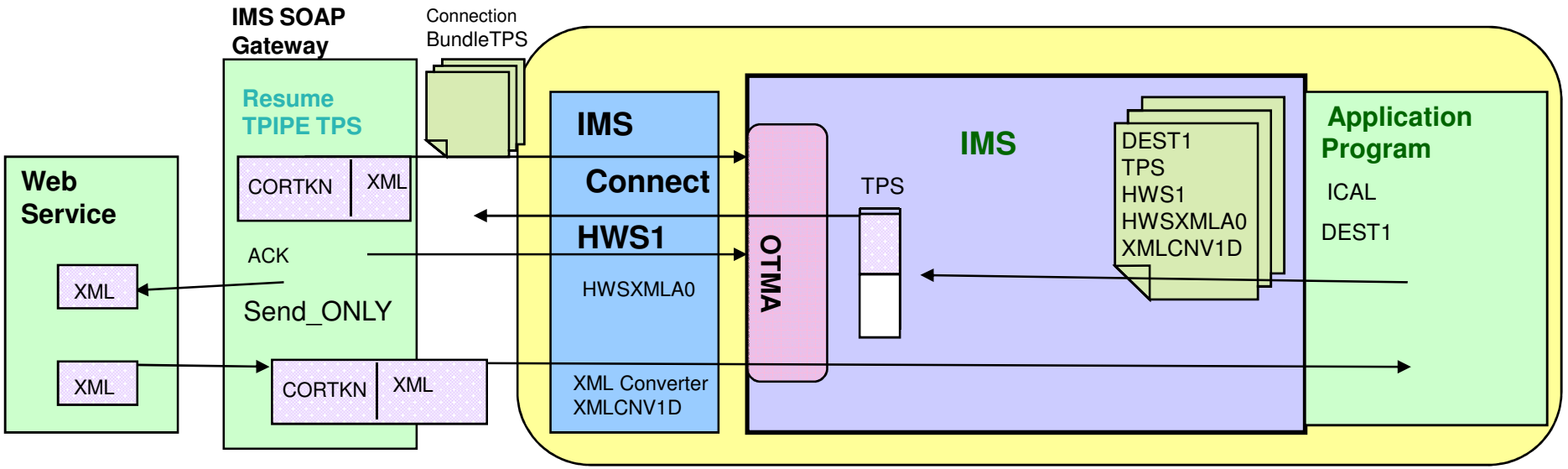
RESUME TPIPE Commit Mode=0 SyncLevel=Confirm LLLL|LLRRIRM|0400| Callout request message includes IMS CORRELATOR TOKEN(CORTKN) LLLL|*CORTKN*|LLRRCORTOKEN|{*REQMOD*}|LLLLDATA*CSMOKY*
Reply ACK or NAK LLLL|LLRRIRM|0400|
Send Response Message Callout response message for IMS application program must include the IMS CORRELATOR TOKEN(CORTKN) LLLL|IRM|LLRRCORTOKEN|LLLLDATA|0400|

  - User-Written Example
    - Client codes all the interactions with IMS Connect
      - ***Can take advantage of the IMS Connect API (Java or C)***
    - Receives the message with the Resume TPIPE call
    - Must accept and pass back the correlation token



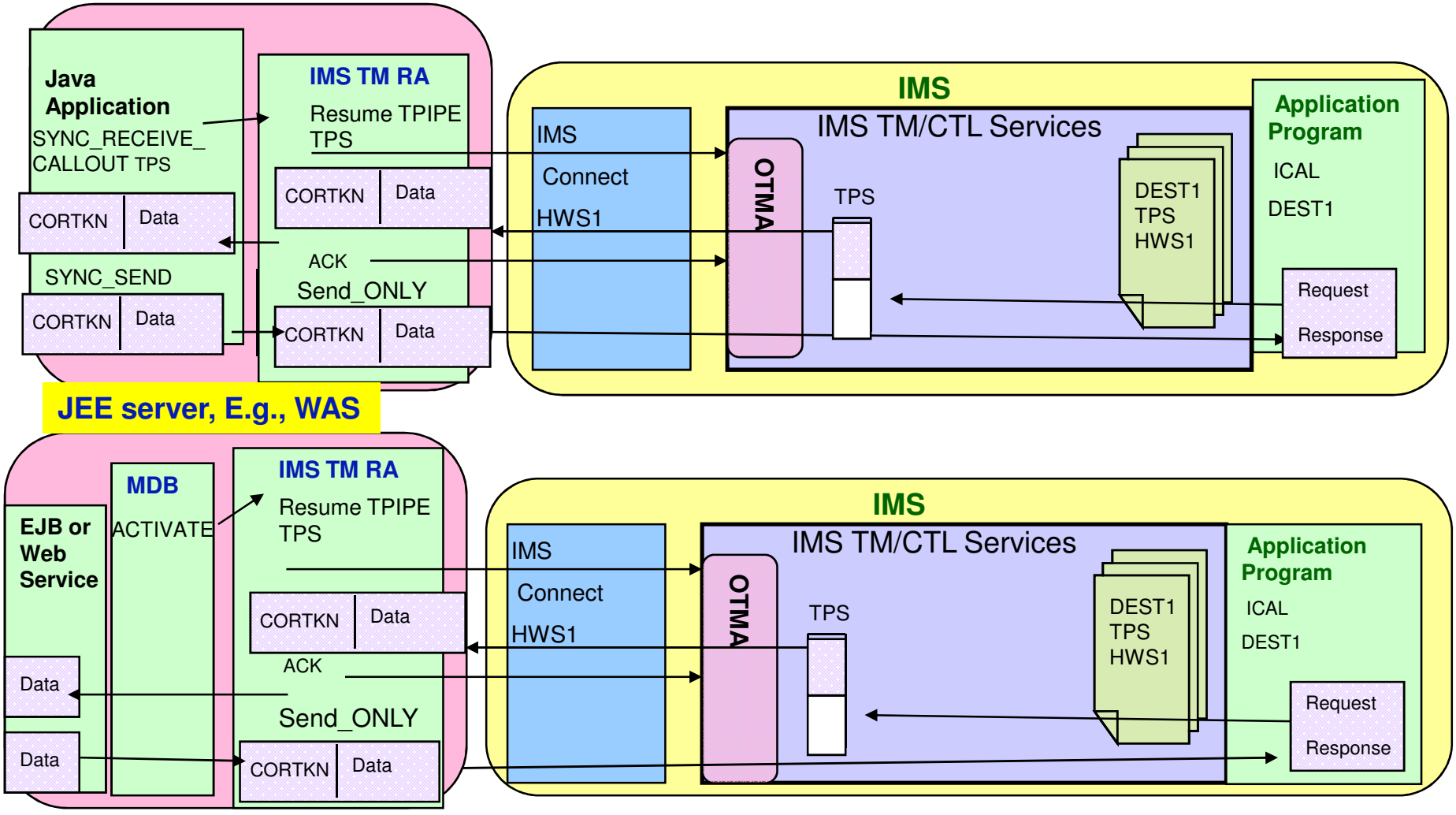
Other Client Examples

- IMS Soap Gateway (Non-JEE)
  - IMS SG *is* the Resume TPIPE client
    - Keeps track of the correlation token
  - RDz tooling
    - Generates the required artifacts – WSDL, XML files, and XML converter files



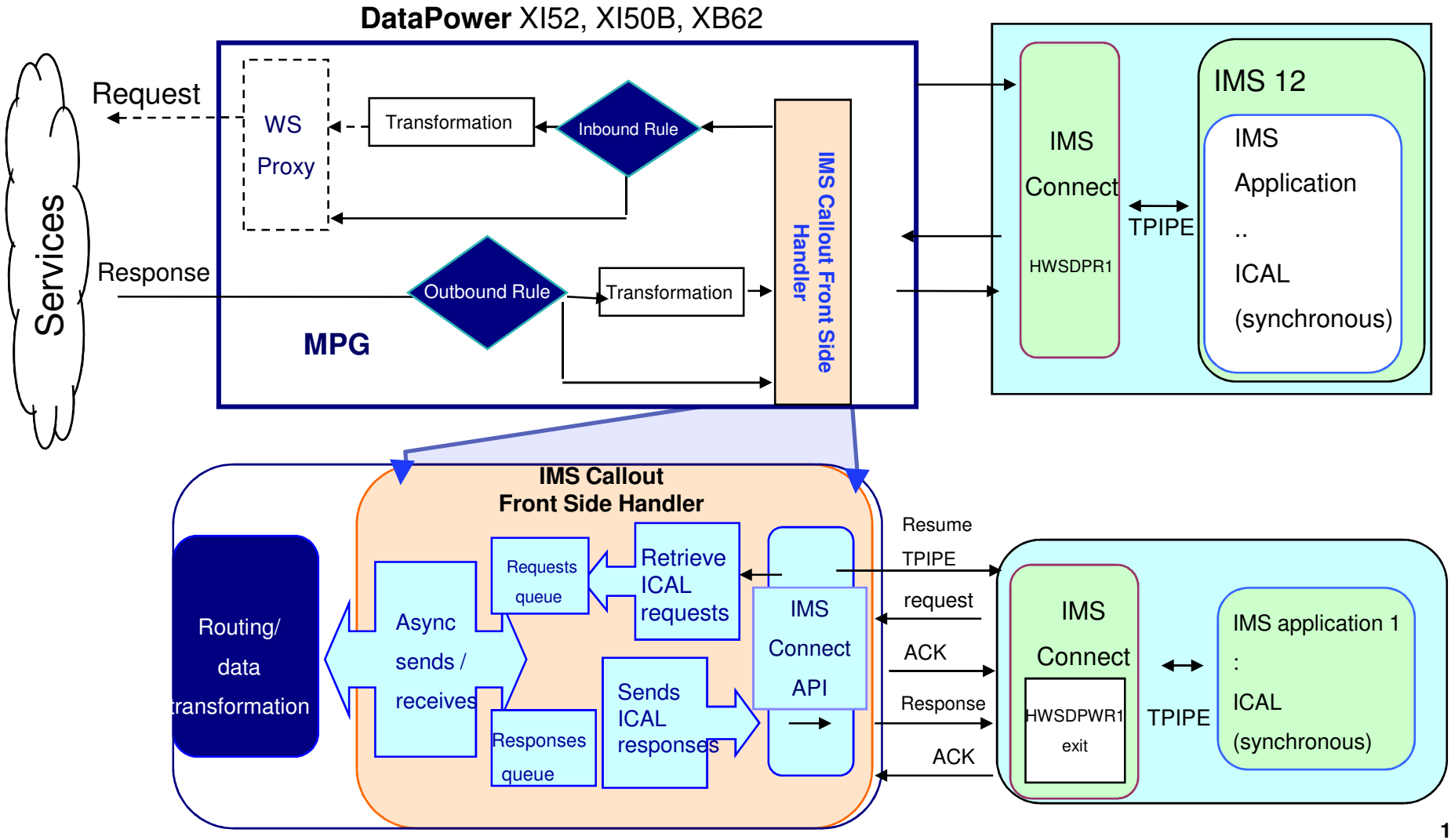
Other Client Examples …

- IMS TM Resource Adapter (JEE environments)



Other Client Examples …

- DataPower (Firmware 6.0)



IMS Application Considerations

- AIB
  - AIBID = DFSAIBbb
  - AIBLEN = AIB length
  - AIBSFUNC = SENDRECV
  - AIBRSNM1 = 8 byte OTMA Descriptor name
  - AIBRSFLD = Timeout value
    - 4 byte field for time value 100th seconds. System default is 10 sec.
  - AIBOALEN = **REQ_area** length
    - As an input parameter: 4 byte field contains the length of the request area
    - As an output parameter: Actual length of the response message
  - AIBOAUSE = **RESP_area** length
    - As an input parameter: 4 byte field contains the length of the response area
    - As an output parameter: Length of the response message.
  - AIBRETRN = AIB Return code
  - AIBREASN = AIB Reason code.
  - AIBERRXT = 2 byte sense code from external application

- OTMA Descriptor

DFSYDTx member of IMS.PROCLIB
- TYPE: Destination type
- TMEMBER: OTMA Target Client
- TPIPE: Destination Name
- SYNTIMER=timeout
  Note ICAL overrides this value
- ...

```
D TESTDPWR   TYPE=IMSCON TMEMBER=HWS1
D TESTDPWR   TPIPE=DPTEST1 ...

D SOAPGWAY  TYPE=IMSCON TMEMBER=HWS1
D SOAPGWAY  TPIPE=SGTTST, ADAPTER=XMLADPTER,
D SOAPGWAY   CONVRTR=TESTCV, ...
```

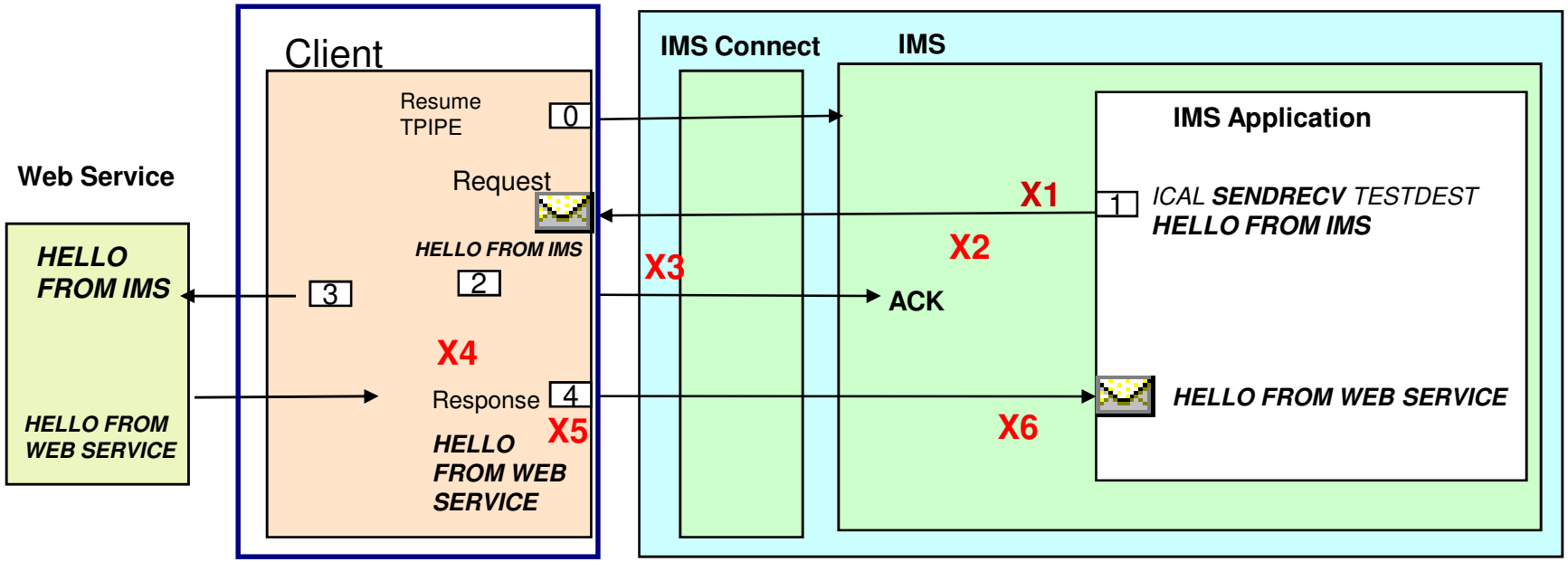
```
01 AIB.
   02 AIBRID PIC x(8)  VALUE 'DFSAIB  '.
   02 AIBRLEN PIC 9(9) USAGE BINARY.
   02 AIBRSFUNC PIC x(8) VALUE 'SENDRCV'.
   02 AIBRSNM1 PIC x(8) VALUE 'TESTDPWR'.
                      {or VALUE 'SOAPGWAY'.}
   02 AIBOALEN PIC 9(9) USAGE BINARY VALUE +12.
   02 AIBOAUSE PIC 9(9) USAGE BINARY.

01  CALLOUT-MSG.
   02  CA-DATA        PICTURE X(12) VALUE 'HELLO WORLD '
01  SCA-RESPONSE.
   02  SCA-DATA       PICTURE X(12).

CALL 'AIBTDLI' USING ICAL, AIB, CALLOUT-MSG , SCA-RESPONSE.
```

COBOL example

Callout message will need to be mapped according to the target program, WSDL, etc.,

IMS Application Considerations …

- Error Conditions
  - The IMS ICAL is informed of error conditions using AIB return/reason codes



Examples of where some errors might occur: *(complete list in the manuals)*

X1: ICAL cannot be sent out
X2: ICAL time out
X3: late ACK received after time out
X4: XML converter in error
X5: external server already committed before time out
X6: external server already committed, but IMS fails to process the response

And in IMS 13

- New "RECEIVE" subfunction code
  - With an expanded response area
    - Retrieves the response message after an ICAL "SENDRECV" is issued with an inadequate response area specification and gets partial data (AIB RC X'100', AIB RS X'00C')
  - IMS 13 keeps a copy of the entire response message in the control region private storage
    - Until a subsequent ICAL "SENDRECV", syncpoint, or application termination
- Addresses
  - Partial response message due to inadequate application specification
- Benefit
  - Provides the ability to complete the retrieval of a reply message
    - Without having to re-issue a complete ICAL "SENDRECV" and associated network transmission costs

And in IMS 13 …

- Usage example:
  - **ICAL --*aib—request area, response area***
    - AIBSFUNC (SENDRECV)
    - AIBOAUSE – Response area length

    - CALL is issued → AIBRETRN=x'100', AIBREASN='00C'
      - Specified length of the output response area is too small
      - AIBOAUSE= length of the data that was returned in the response area
      - AIBOALEN = the actual length of the entire response message

    - Using the value in the previous AIBOALEN and leveraging the new support which keeps the message in IMS CTL region private, retrieve the entire response!

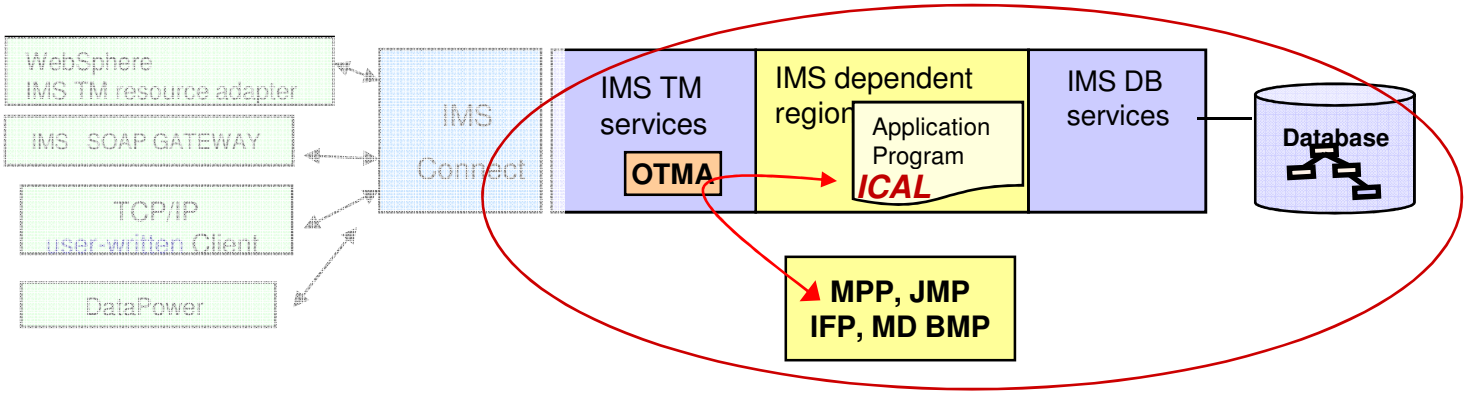
  - **ICAL --*aib— response area***
    - Where *response area* has been expanded to contain the entire message
    - AIBSFUNC (RECEIVE)
    - AIBOAUSE – new response area length

    - CALL is issued successfully
      - ✓ AIBOAUSE – length of the response in the response area
      - ✓ AIBOALEN – set to 0 because the call successfully returned the entire response

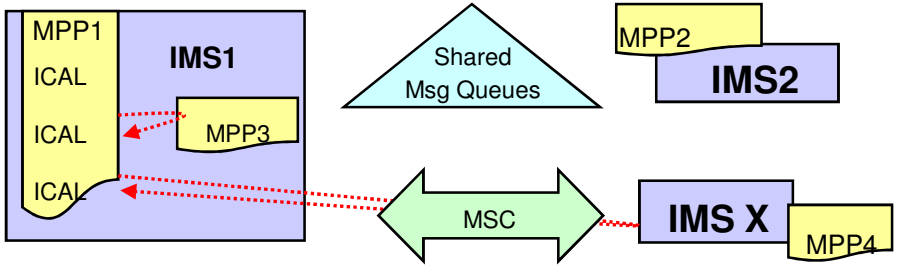
And also in IMS 13, the extension of the ICAL to invoke IMS transactions

Synchronous Program Switch

- New capability that enhances the DL/I ICAL support
  - Allows an IMS application program to *synchronously* call and wait for a reply from *another IMS application program*
    - Within the calling program's UOW



  - Regardless of where the called application resides
    - In the same IMS
    - In a different IMS
      - In a Shared Queues back-end
        - *All IMS's are IMS 13*
      - Across an MSC link
        - *Any supported release*



Synchronous Program Switch Highlights

- Automatic invocation of OTMA
  - Without requiring OTMA to be defined or commands to be issued
  - New OTMA destination descriptor TYPE
    - IMSTRAN
- Enhancements to the DL/I ICAL
  - Allows an IMS transaction to be the target destination
    - Accepts multi-segment requests/responses
  - Provides additional AIB return and reason codes
- Support for Late Reply messages
  - Can be purged or rerouted
- Security authorization
  - Ensures userid of program issuing ICAL can access the target transaction

Considerations

- No ICAL support for BMP or JBP applications running in DBCTL environments
  - ICAL is part of the IMS TM capability
- IMS application program issuing ICAL for a synchronous program switch
  - Can be a protected transaction
    - But the target transaction of the ICAL is not part of the RRS commit scope
- The switched-to program (target of the ICAL)
  - Has read-only access to the main storage data base (MSDB)
  - Cannot be an IMS Conversational transaction
  - Does not invoke IMS Message Format Service (MFS)

The DI/I ICAL

Same Format as the ICAL originally introduced for synchronous callout support

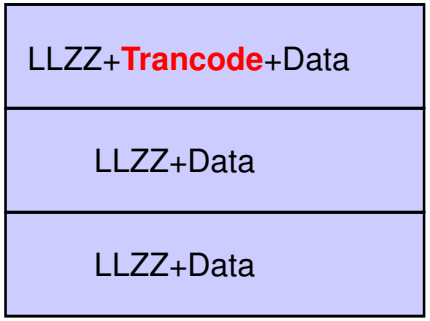
>>-ICAL--aib--request_area--response_area---------->><

Call Name	DB/DC	DBCTL	DCCTL	DB Batch	TM Batch
ICAL	X		X		

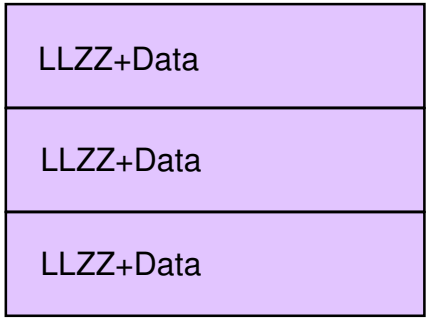
```
01 AIB.
   02 AIBRID   PIC x(8) VALUE 'DFSAIB  '.
   02 AIBRLEN  PIC 9(9) USAGE BINARY.
   02 AIBSFUNC PIC x(8) VALUE 'SENDRECV'.
   02 AIBRSNM1 PIC x(8) VALUE 'OTMDEST1'.
   02 AIBRSNM2 PIC x(8).
   02 AIBRESV1 PIC x(8).
   02 AIBOALEN PIC 9(9) USAGE BINARY VALUE 28.
   02 AIBOAUSE PIC 9(9) USAGE BINARY VALUE 30.
   02 AIBRSFLD PIC 9(9) USAGE BINARY VALUE 5000.
   02 AIBRESV2 PIC X(8).
   02 AIBRETRN PIC 9(9) USAGE BINARY.
   02 AIBREASN PIC 9(9) USAGE BINARY.
   02 AIBERRXT PIC 9(9) USAGE BINARY.
```

What is different is that the format of the request and response reflect standard IMS transaction message format

Request Data (example of multi-segment):



Response Data in multi-segment:



The DL/I ICAL Call …

- Examples of common Return codes for synchronous program switch
  - Full list in the IMS documentation

Return Code (Hex)	Reason Code (Hex)	Extended Reason code (Hex)	Brief Description
0100	000C	0000	Partial data returned
0100	0100	0000	Error message returned
0100	0104	0020	ICAL timed out
0100	010C	0000	/PSTOP cmd issued
0100	0110	0000	Invalid trancode
0100	0110	0004	Security violation
0100	0110	0005	DFSYICAL stopped
0100	0110	0020	Input length invalid
0100	0110	0031	Trans is stopped
0100	0110	0061	Target trans does not insert back to IOPCB

- If the ICAL “SENDRECV” receives a partial data status
  - The ICAL “RECEIVE” subfunction can retrieve the entire message

OTMA Support

- Continues to use OTMA Destination Routing Descriptors (AIBRSNM1)
  - TYPE= IMSTRAN for synchronous program switches

D entry_name keywords *Where* entry_name is descriptor entry name and can be masked by ending in an *

keywords are: **TYPE=IMSTRAN**
LTERMOVR=name
TMEMBER=name
TPIPE-name
SMEM=NO|YES
EXIT= NO|YES
REPLYCHK=YES|NO
SYNCTP=NO|YES
SYNTIMER=timeout value

For example: D OTMDEST1 TYPE=IMSTRAN SYNTIMER=500

D OTMDEST2 TYPE=IMSTRAN TMEMBER=SCOTTHWS1 TPIPE=BRYCE EXIT=YES

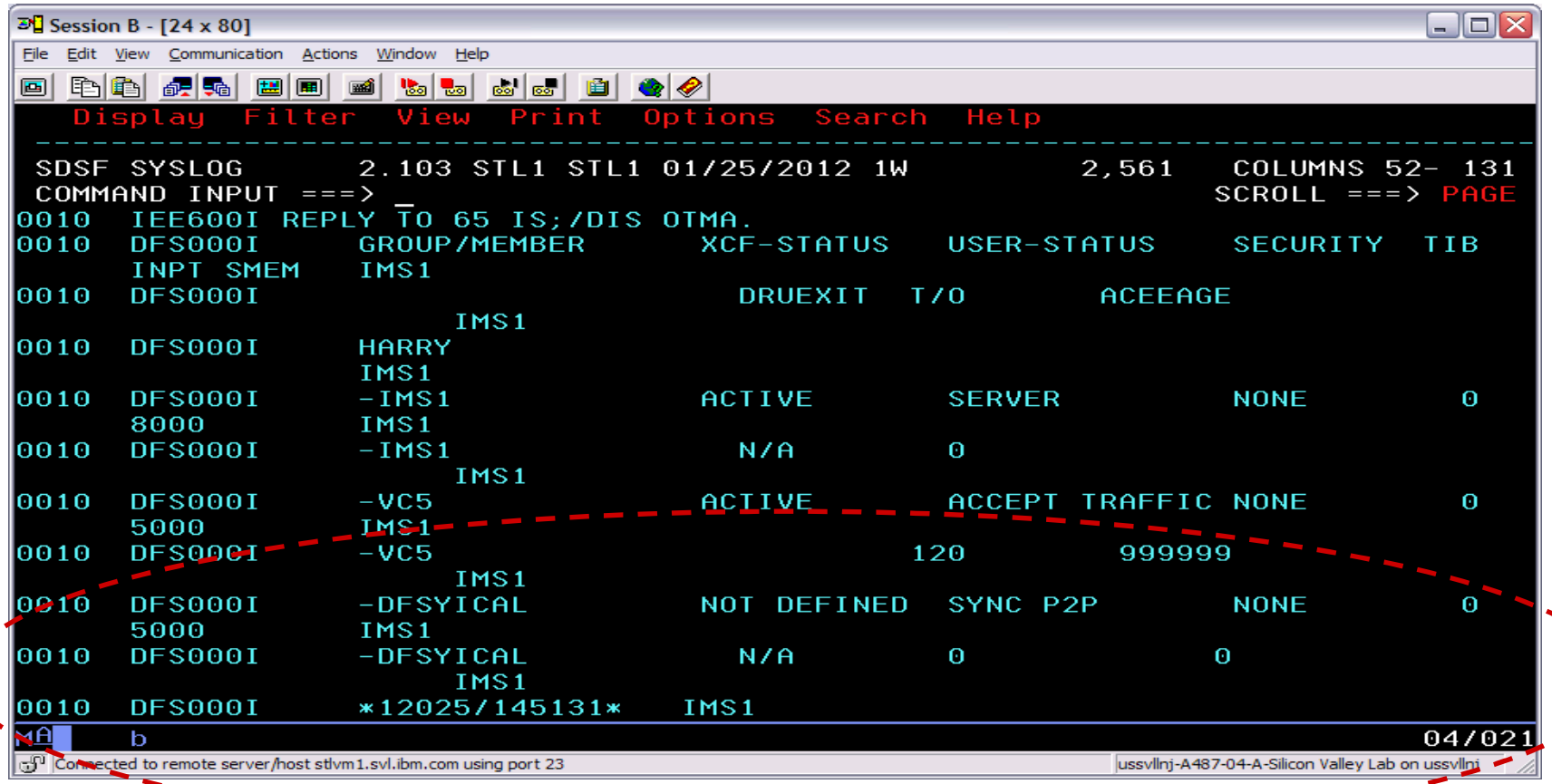
  - The same TYPE=IMSTRAN descriptor can be used on behalf of multiple target transactions
    - *Actual trancode is in the request_area of the ICAL and not in the descriptor*
    - *Need for different descriptors is based on different processing requirements*
    - *Minimizes the number of descriptors needed for synchronous program switch support*

OTMA Support …

- Non-XCF related OTMA services are used
  - Internal invocation of OTMA services to process the target transaction request
    - Without specifying OTMA=Y in the DFSPBxxx member of IMS PROCLIB and without issuing /START OTMA command
      - No need to start the XCF connection with any OTMA client for the synchronous program switch
  - The target transaction is processed as an OTMA transaction
    - If authorization is required, IMS checks to see if user can access target
    - Using OTMA send-then-commit (CM1) protocol with SyncLevel=CONFIRM
      - Target transaction of the ICAL is processed as an OTMA transaction
    - *IMS creates an internal OTMA member DFSYICAL and internal tpipe DFSTPIPE to process the transaction*
      - And generates internal ACK/NAK for the CONFIRM request

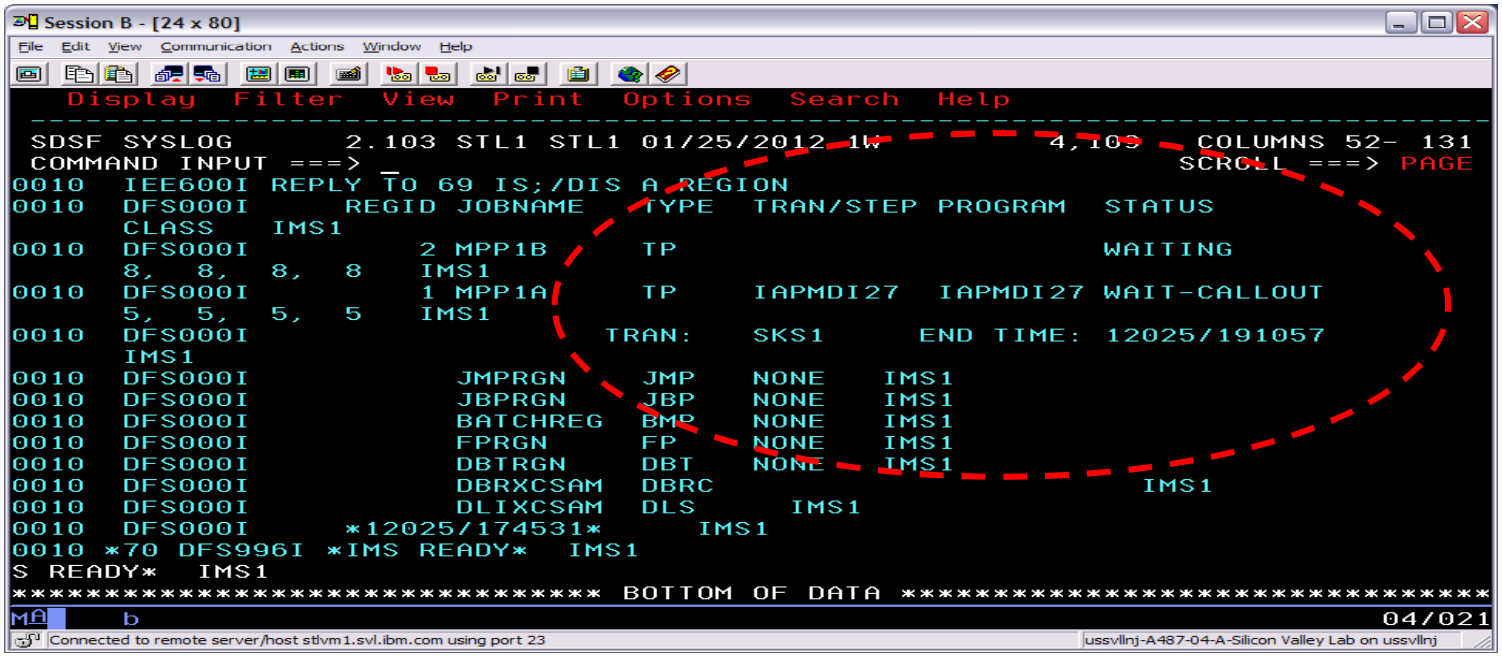
OTMA Support …

- /DIS OTMA can be used to see the existence of the DFSYICAL TMEMBER and associated properties.



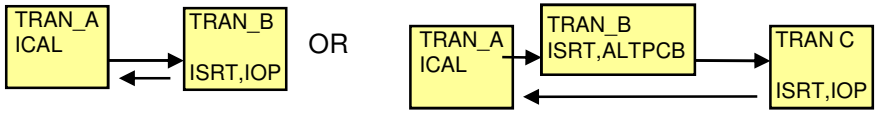
OTMA Support …

- /DIS ACTIVE REGION
  - Displays the target transaction for the synchronous program switch
  - Displays the calculated end time of the ICAL
    - Based on the timeout value

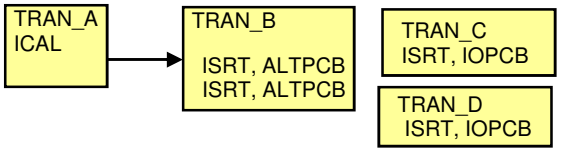


Application Considerations

- Design
  - Simple is best
    - One response to the request



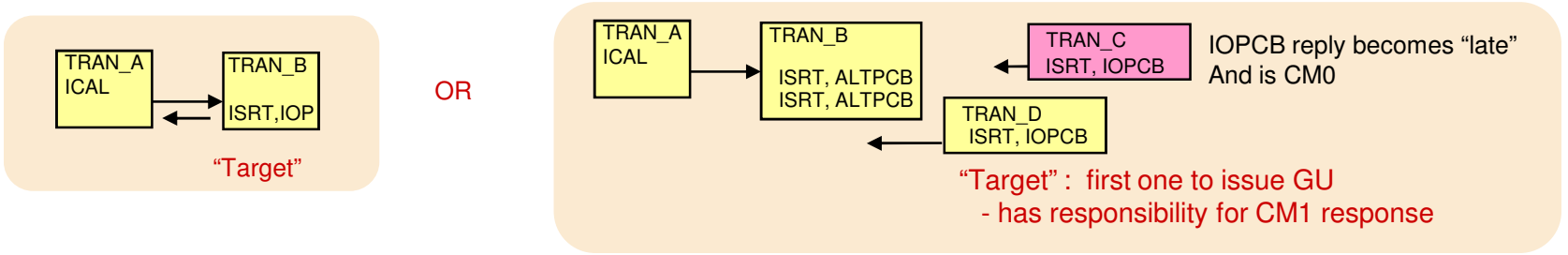
  - But complexity is possible
    - Calling an existing transaction, e.g., TRAN_B might spawn other transactions instead of replying



    - Design needs to determine which reply message is the response to the ICAL and which is to be considered “late”
      - Late responses can be automatically purged or re-routed
        - *Controlled by the REPLYCHK parameter on the IMSTRAN descriptor*

Application Considerations…

- Design…
  - REPLYCHK parameter (YES |NO) in the IMSTRAN descriptor
    - Specifies whether or not IMS should check that the "target" IMS transaction responds to the IOPCB
      - *"Target"* transaction has responsibility for the send-then-commit CM1 reply

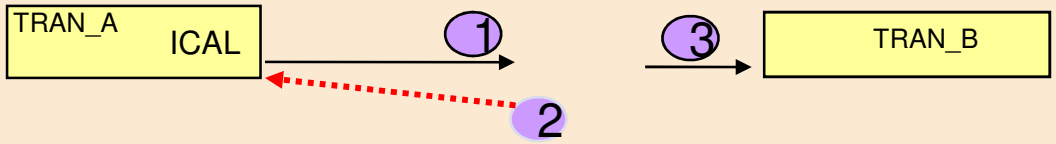


    - Controls the actions IMS performs
      - E.g., Rely on the 'target' transaction to reply or allow the first response from any of the spawned transactions

Detailed information is available in the IMS 13 GA class materials – Application section
http://www-01.ibm.com/software/os/systemz/telecon/8jul/

Application Considerations …

- Transaction Expiration
  - Uses the timeout value of the synchronous program switch
    - Lower of ICAL/Descriptor timeout value
  - Invoked during application GU time



ICAL times out!!

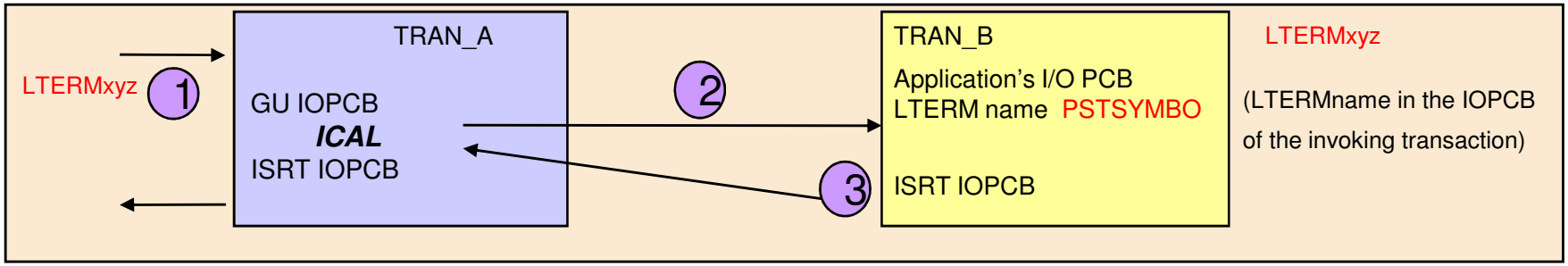
Application issuing ICAL receives a timeout with:
- AIB return code X'00000100'
- reason code X'00000104'
- extended reason code X'00000020'

Application can decide the next action

Transaction Expiration is detected during Application GU:
- Input message is discarded
- A 67D0 log record is written
- No symptom dump
- No DFS554 message issued.

Application Considerations …

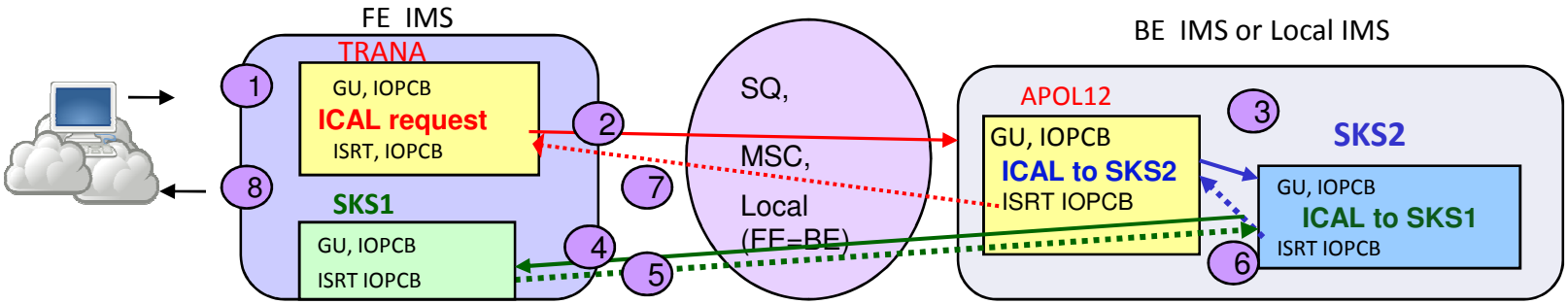
- What about calling applications that rely on the LTERM name in the IOPCB?
  - Default in the IOPCB of the called program is the origin symbolic (PSTSYMBO)
  - It can be overridden by the following 2 methods:
    - AIBRSNM2 in AIB block of the ICAL
    - LTERMOVR value in the OTMA descriptor for the ICAL call
  - If both are provided, the name in the AIB will be used.



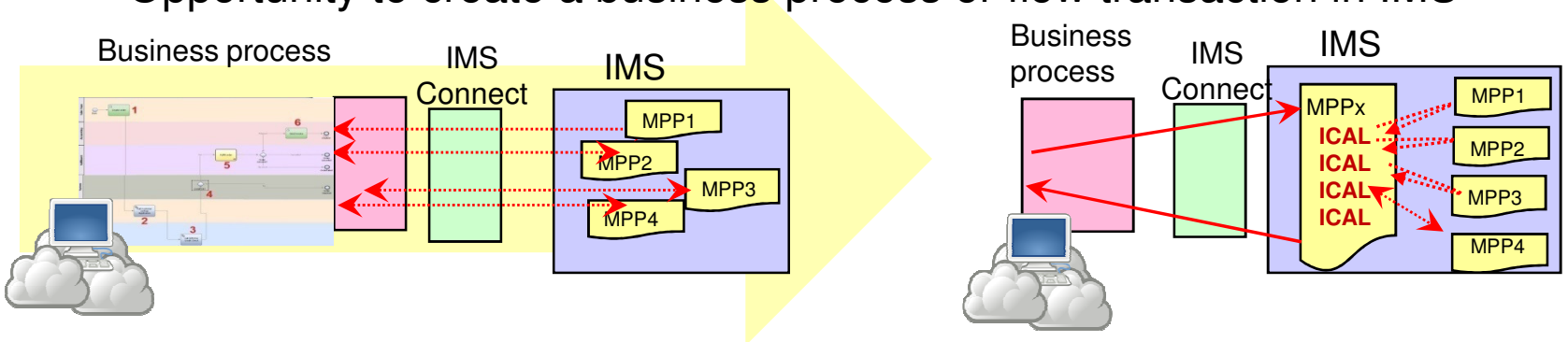
Again: More detailed information on application considerations for the ICAL can be found in the IMS 13 GA class materials – Application section http://www-01.ibm.com/software/os/systemz/telecon/8jul/

So…

- The synchronous program switch capability expands the IMS environment to suit your evolving needs
  - Ability to call other IMS transactions in the calling program's UOW to integrate multiple transactions into a composite transaction
    - Across a single or multiple IMS systems

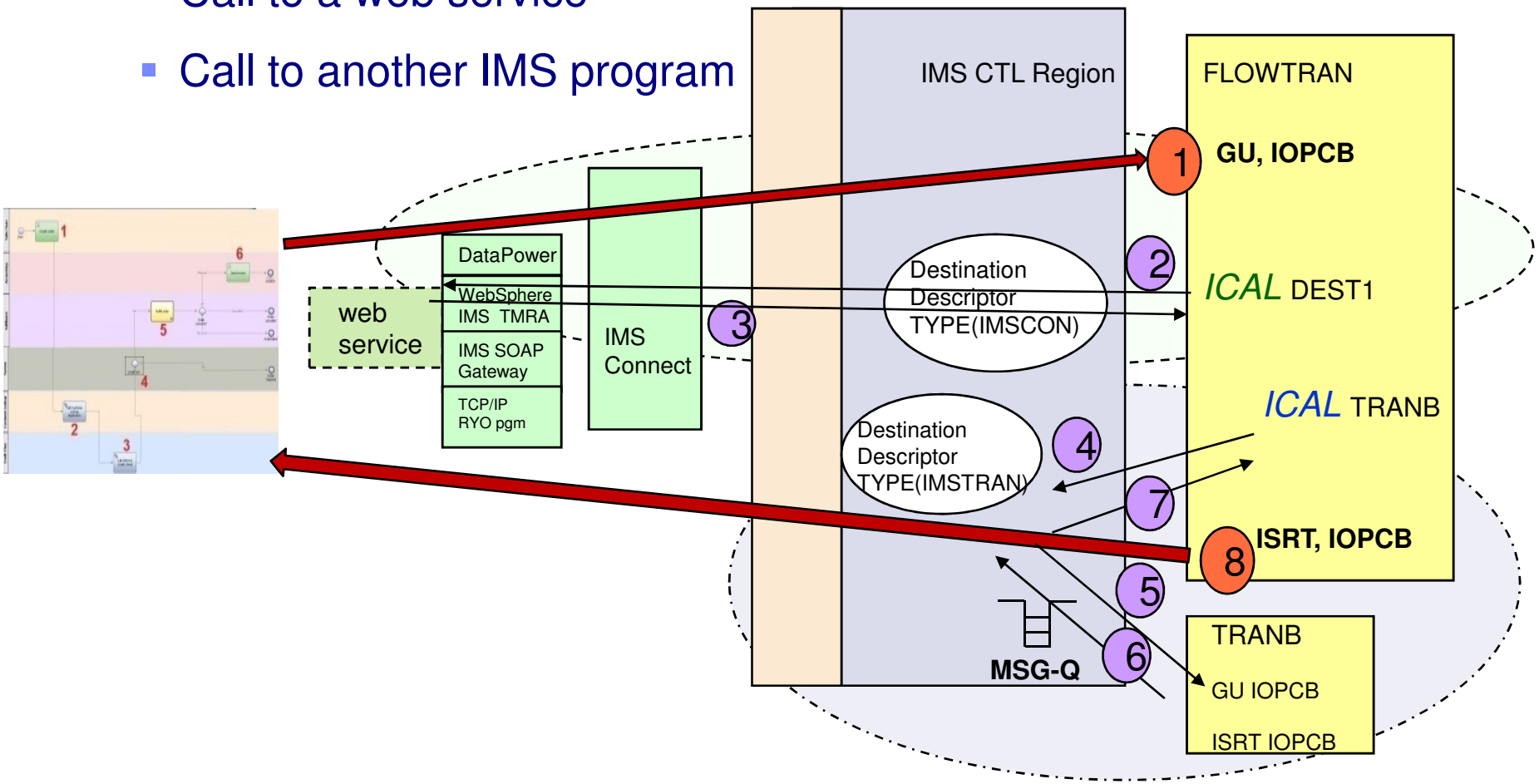


  - Opportunity to create a business process or flow transaction in IMS



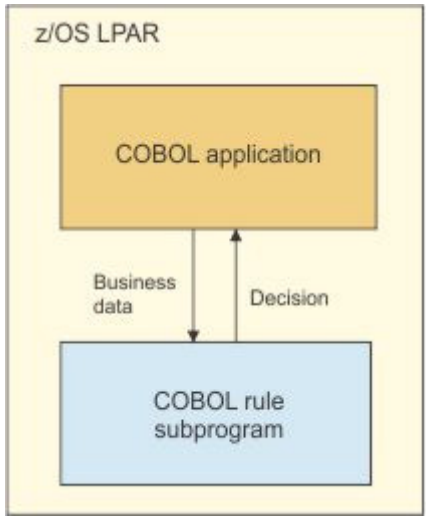
And even

- Integration of the two ICAL capabilities
  - Call to a web service
  - Call to another IMS program



But what if Changes need to be made?

- Part of effective business process management is the ability to stay up to date with business needs
  - Requires the ability to change processes quickly and effectively
    - Older systems: decision making is buried deep within the code, requiring an application lifecycle process change

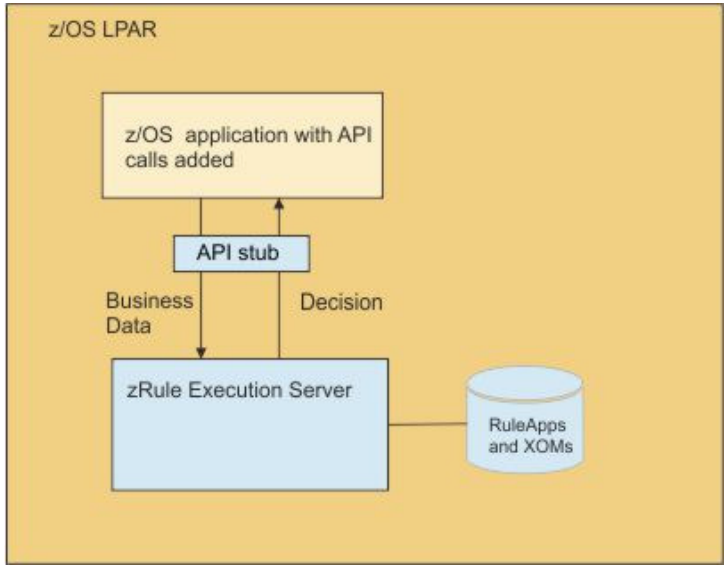


- To change rules that are implemented as a COBOL rule subprogram
  - Regenerate the rule subprogram
  - Recompile the COBOL application
  - Rebuild the load module

But what if Changes need to be made? …

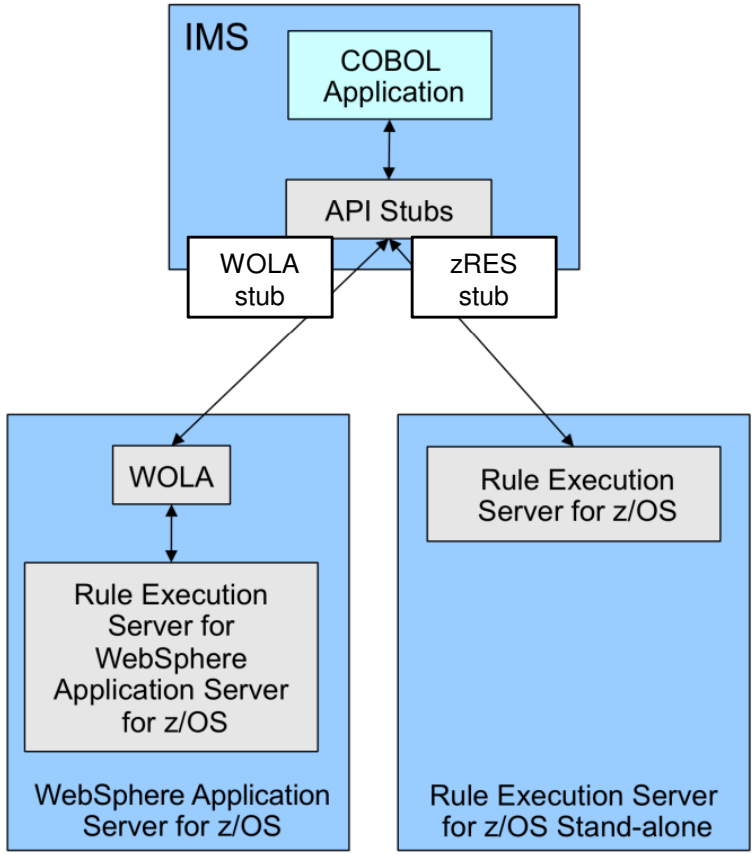
- New model: separation of policies from the software itself
  - By keeping the decision-making details separate, they can be changed more conveniently, without modifying the software itself
    - Allows different software modules to access the same rules with less risk of contradicting policies

- The answer: a ***business rules and events engine*** implementation



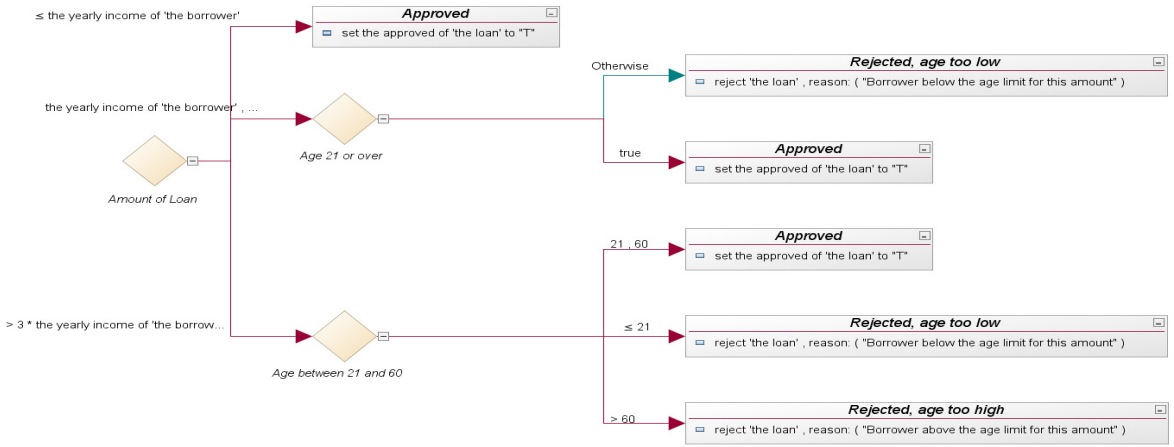
IBM Operational Decision Manager (ODM)

- IBM ODM and IMS
  - Supports rules management for IMS COBOL message processing programs (MPPs), batch message processing programs (BMPs), and DLIBATCH programs
  - Runtimes:
    - Stand-alone on z/OS
    - Within a WebSphere Application Server environment
      - Accessed via WebSphere z/OS Optimized Local Adapters (WOLA)



Steps

- Translate a *business policy into a set of rules that can be* stored inside IBM Operational Decision Manager (ODM).
  - Create all rules in the Rules Designer GUI, which is shipped as part of ODM
    - The Rules Designer works on an Eclipse platform
  - Deploy the rules to the Rules Server
    - Rules can be
      - Simple: if balance < 0 then account_status = overdrawn
      - Complex: Loan value < $1M and customer age < 65 and borrower income and borrower income after tax > $24K and ….



Steps…

- Example of using the Rule Execution Server for z/OS
  - Connect the program to the server
    - E.g., Call 'HBRCONN' using HBRA-CONN-AREA

```
01 HBRA-CONN-AREA.
10 HBRA-CONN-EYE PIC X(4) VALUE 'HBRC'.
10 HBRA-CONN-LENGTH PIC S9(8) COMP VALUE +3536.
10 HBRA-CONN-LENTH REDEFINES HBRA-CONN-LENGTH
```

  - Invoke the Rule Execution Server for rules checking as often as needed
    - Specify which rules (or RuleApps) to check
      - The location of the rules is specified in the following path:

        HBRA-CONN-RULEAPP-PATH
    - Specify the input and output parameters associated with the rules
      - E.g., in HBRA-RA-PARMS of the HBR-CONN-AREA

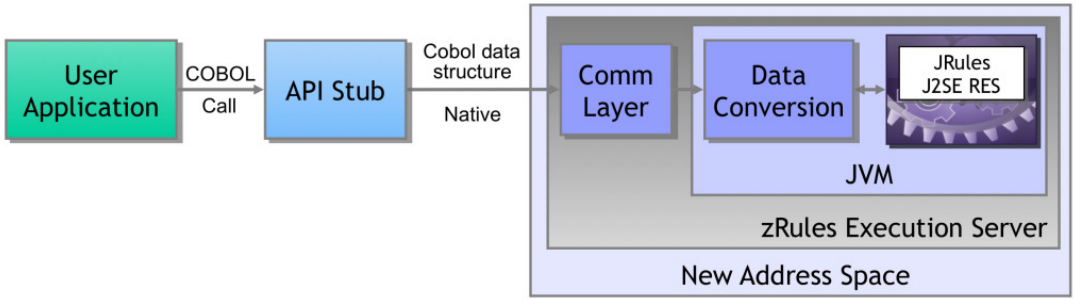
```
15 HBRA-RA-PARMS OCCURS 32.
20 HBRA-RA-PARAMETER-NAME PIC X(48).
20 HBRA-RA-DATA-ADDRESS USAGE POINTER.
20 HBRA-RA-DATA-LENGTH PIC 9(8) BINARY
```

    - Invoke the rule
      - E.g., call 'HBRRULE' using HBRA-CONN-AREA

Steps…

- Example of using the Rule Execution Server for z/OS (Rule Execution Server) …
  - Disconnect from the server
    - call 'HBRDISC' using HBRA-CONN-AREA

- To resolve the API calls (HBRCONN, HBRRULE & HBRDISC), the IMS program also needs to be link-edited with the HBRISTUB module.
  - INCLUDE HBRLIB (HBRISTUB)



- Alternatively, you can use the Rule Execution Server with WebSphere z/OS Optimized Local Adapters (WOLA) and the ESS interface in IMS

IBM Operational Decision Manager (ODM) …

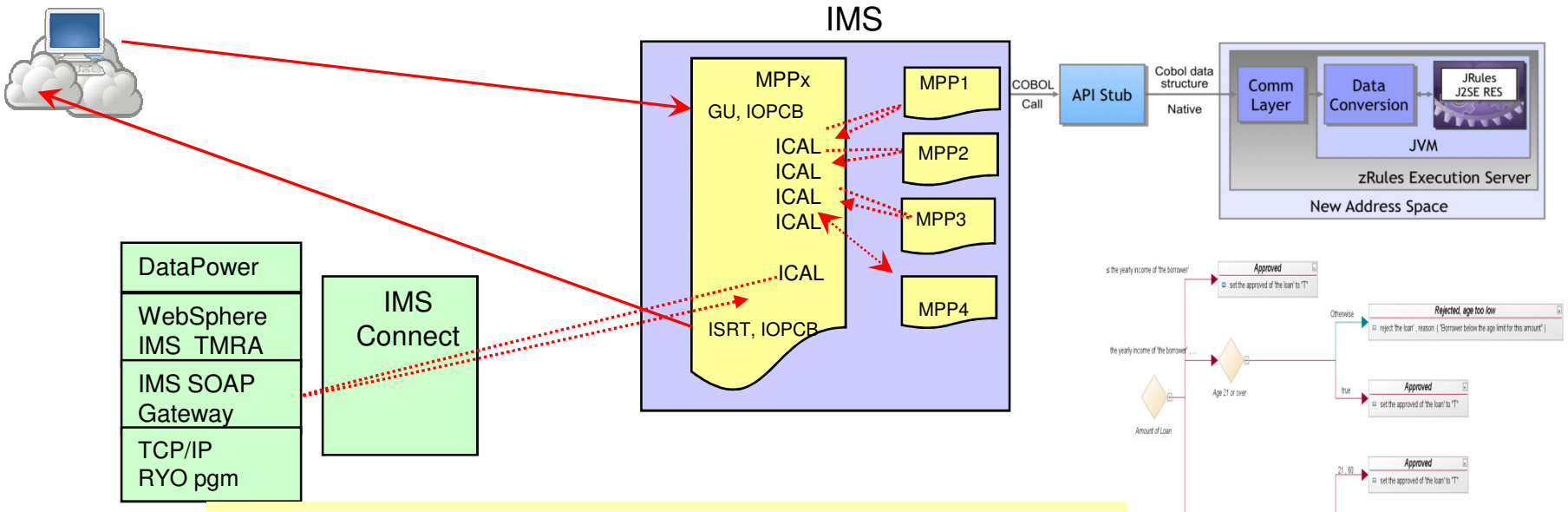
- Implementing IBM ODM
  - Allows IMS applications to exploit a powerful decision engine
    - Take advantage of rules-based processing
    - And quickly address evolving business conditions

- Code can be modified in the rules engines without having to change the IMS application

- More information including implementation steps and examples
  - http://www.redbooks.ibm.com/abstracts/redp4997.html?Open

And so … The Final Message

- IMS application Modernization includes
  - The ability to “call out” to a remote service
  - The ability to “synchronously program switch” to another IMS application
  - The ability to implement change using business rules



Ensuring that IMS continues to be part of
leading technology solutions
*** the platform for today, tomorrow, and the future ***