Telcordia Technologies

Performance from Experience

IMS TECHNICAL CONFERENCE

# Customer Experience: Using IMS Connect and OTMA/CI

## Session E93

***Steve Nathan***
***stephen.nathan@telcordia.com***

***Alan Velasquez***
***alan.velasquez@telcordia.com***

An SAIC Company

RETURN TO INDEX

# Disclaimer

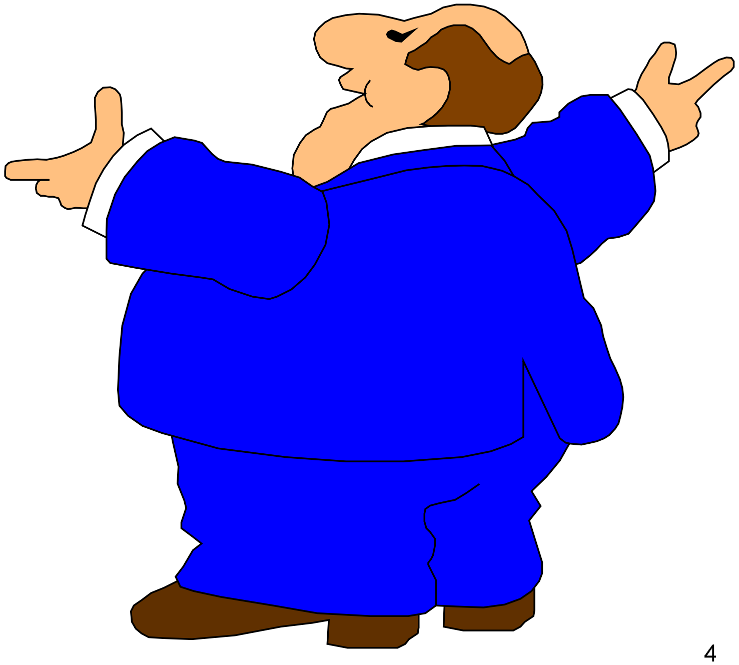
- The purpose of this presentation is to provide a technical perspective of Telcordia's experience using IMS and IMS Connect.
- Although this document addresses certain IBM products, no endorsement of IBM or its products is expressed, and none should be inferred.
- Telcordia also makes no recommendation regarding the use or purchase of IMS or IMS Connect products, any other IBM products, or any similar or comparable products.
- Telcordia does not endorse any products or suppliers. Telcordia does not recommend the purchase or use of any products by the participants.
- Each participant should evaluate this material and the products himself/herself.

# Acknowledgements

- Special thanks to:
  - Jack Yuan and his team from IBM IMS OTMA development for all the enhancements in PQ32402 and all of the other enhancements in OTMA
  - Gerald Hughes and his team from IBM ITOC and IMS Connect development for all their hard work and enhancements
  - Jack Yuan for the IMS Callable Interface and all its wonderful and timely enhancements

# Presentation Outline

- Introduction
- IMS OTMA
- IMS Connect
- CORBA
- OTMA Callable Interface
- Sources of Documentation
- Glossary



# Introduction

- Telcordia has given several presentations about its experiences using IMS Connect and OTMA Callable Interface
- There have been many changes and enhancements in the last year
- This presentation will concentrate on what is new and exciting with these OTMA clients

# Introduction

- IMS/ESA 5.1 introduced the OTMA (Open Transaction Manager Access) feature
- This feature uses the MVS cross-coupling facility (XCF) to send data to IMS from other MVS applications (OTMA clients)
  - No VTAM or TCP/IP is involved

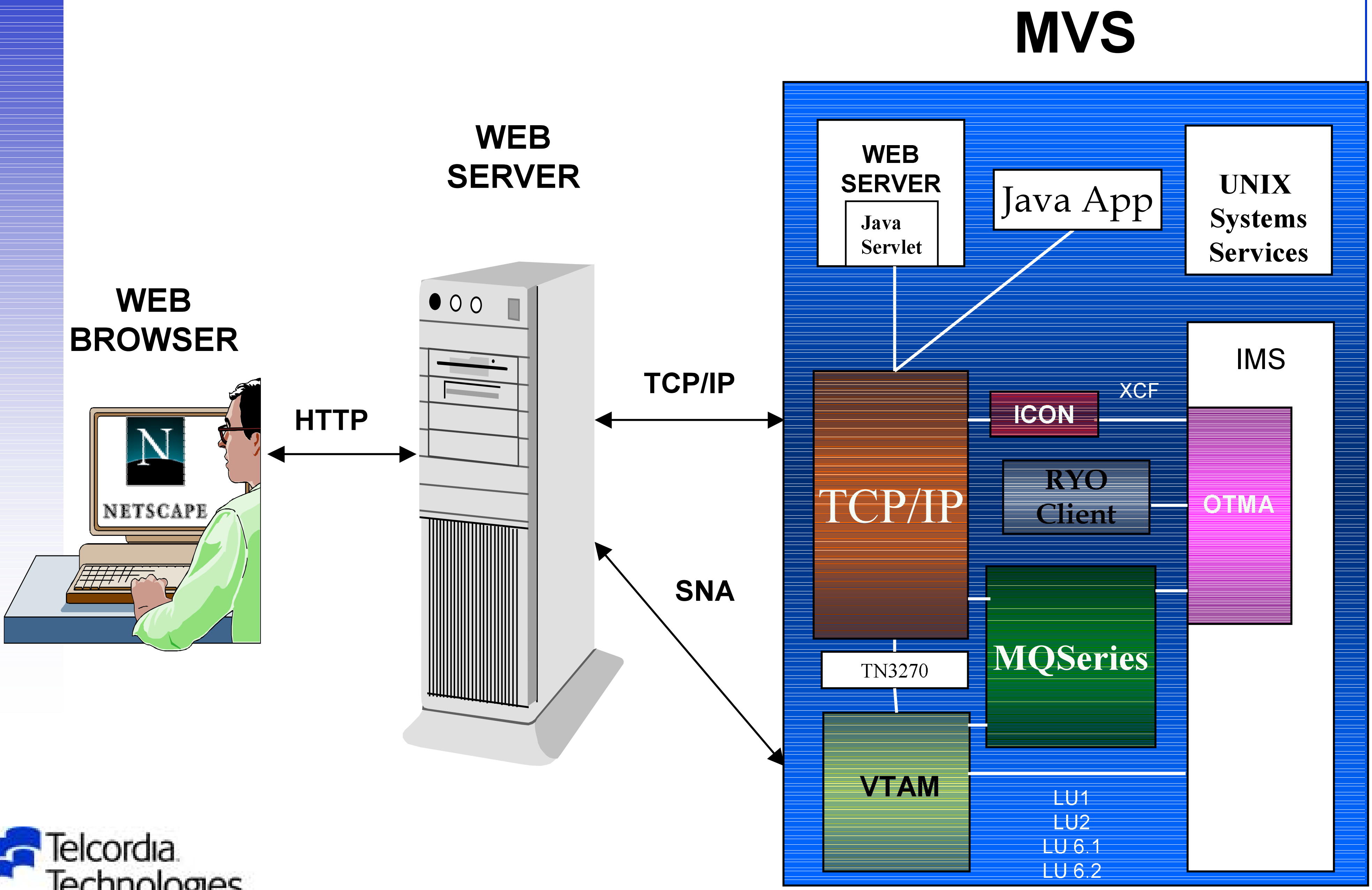
# Introduction

- IMS Connect is an IBM provided OTMA client for TCP/IP
  - This was the IMS TCP/IP OTMA Connection (ITOC)
- MQSeries includes an IMS OTMA client
  - “MQSeries-IMS Bridge”
  - See Session E92
- You can write your own OTMA client using the OTMA Callable Interface

# Introduction

- There are other vendor supplied OTMA clients
- This is a partial list I found by searching the web
  - Tuxedo - BEA Systems
  - Component Broker - IBM
  - Orbix - Iona
  - Uniface - Compuware
  - TCP/IP - Interlink
  - SNA Server - Microsoft
  - AS/IMS (DCE) - IBM

# Introduction

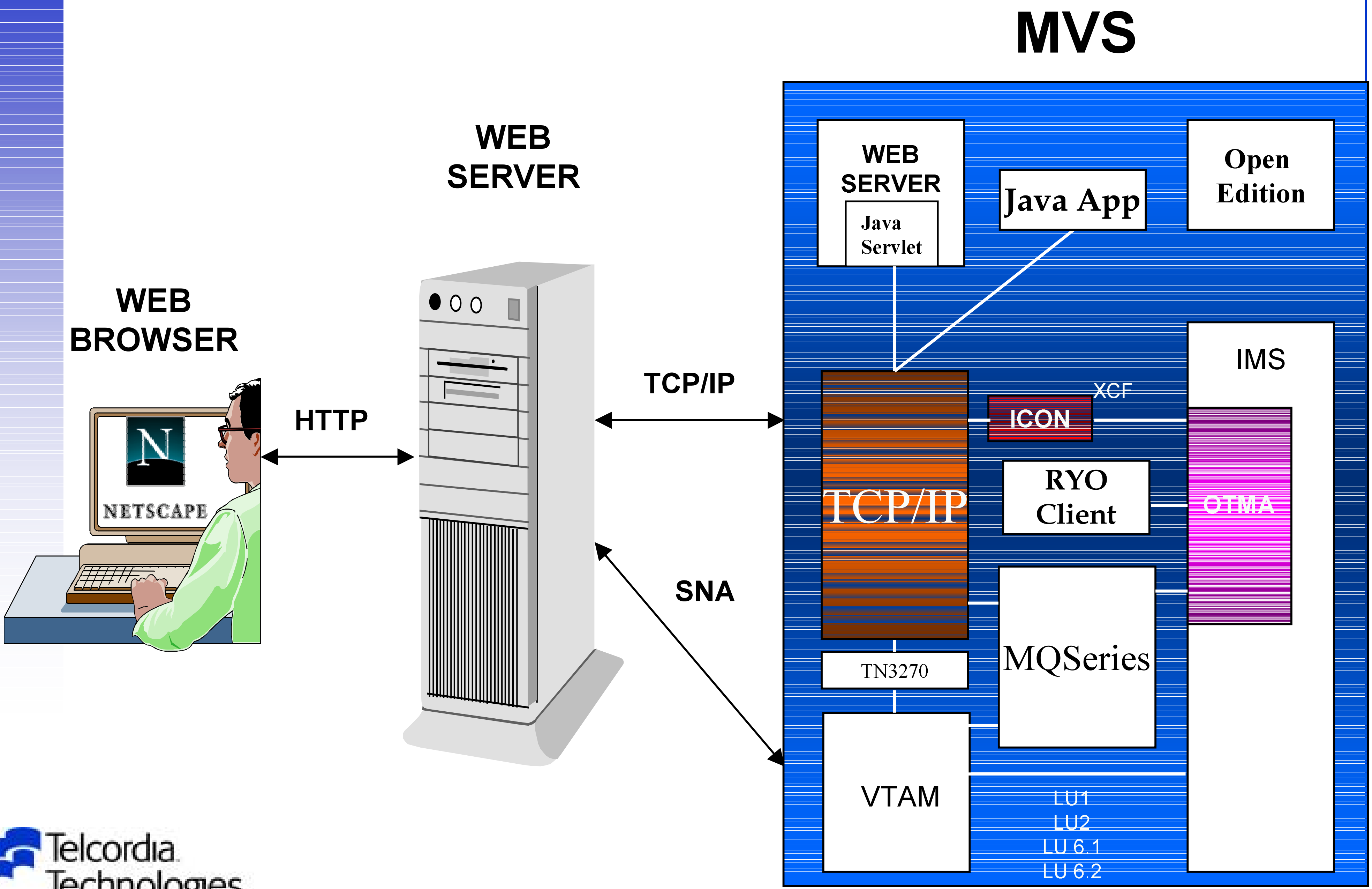


Performance from Experience

# IMS Connect

- IMS Connect is an IBM provided, TCP/IP server and an IMS OTMA client
  - Runs in a separate address space
- Accepts input messages from TCP/IP client
  - Could be any TCP/IP Socket application
  - Could be Java Application, Bean, Applet, Servlet
- Passes input message to user exit  for formatting
- Sends input message to IMS OTMA
- Receives response message from IMS OTMA
- Passes response message to user exit for formatting
- Sends response message to TCP/IP client

# IMS Connect



Telcordia Technologies

*Performance from Experience*

# IMS Connect

- Replaces the IMS TCP/IP OTMA Connection (ITOC)
- It is a separate product
  - An IMS 7.1 license is not required
- IMS Connect can be used to communicate with IMS 5.1 and IMS 6.1 and IMS 7.1
  - Some features are only available with IMS 7.1
- We transparently replaced ITOC with ICON
  - We have had no problems at all with ICON

# IMS Connect

- There are many wonderful enhancements in IMS Connect compared to ITOC
  - SMP installed and maintained
  - Persistent sockets for CM1 messages
    - ICON normally does a disconnect after each message
    - Looking to expand to CM0 messages
  - Send-Only messages
    - Prevents DFS2082 from Nonresponse Mode transaction which does not reply

# IMS Connect

- More wonderful enhancements in IMS Connect
  - Support for asynchronous output from IMS (7.1)
    - ALTPCB output
    - NAK'ed output
  - Initialization Exit
    - User can store up to 2,000 bytes of data to pass to Message Edit exits
  - Datastore Table
    - List of current IMS datastores and their status
    - Passed to Message Edit exits
    - Can be used to route messages to surviving IMS copies
  - Enhanced dump formatting

# IMS Connect

- IMS TCP/IP OTMA Connection (ITOC)
  - ITOC is scheduled to go out of service on 9/1/2001
  - There is a refresh of ITOC Version 2.1.3 available on the web
    - Was available 3/2000
    - There have been very few fixes since then
  - Migrating to IMS Connect from ITOC will require one small update to your user exits if you are storing your own data in the OTMA User Prefix
    - There are also two new values in existing flags

# IMS Connect

- One ICON can connect to multiple IMS control regions in multiple XCF groups
- One ICON can have multiple connections to the same IMS copy
- One IMS control region can connect to multiple ICON's
- ICON and IMS can be on different MVS copies in the same Sysplex
  - If you get a S9C7 abend see APAR PQ02806

# IMS Connect

- IBM made a terrible error when they assigned the 3 character prefix to ITOC modules: HWS
  - Comparable to IMS DFS or MQSeries CSQ
- ITOC developers expanded this to Host Web Services and used that phrase throughout their documentation
  - It did not help that IMS Web was the first product to use ITOC
- ITOC is a TCP/IP OTMA connector but does not necessarily have to deal with the web
  - All references to ***Host Web Services*** should be removed
    - And they have been removed for IMS Connect

# IMS Connect

- ICON runs as an MVS job or started task
- Controlled by two input files:
  - BPECFG
    - Internal trace parameters
    - Just use shipped sample
    - Only change trace parameters if requested by IMS
  - HWSCFG
    - Next page

# IMS Connect

- HWSCFG
  - HWS control card
    - Gives name to this ICON instance
    - Defines RACF processing
  - TCP/IP control card
    - HOSTNAME - TCP/IP Host name (TCP/IP JOBNAME)
    - RACFID - Default RACF ID to assign to messages
    - PORTID - List of from 1 to 50 ports for ICON to listen on
    - MAXSOC - Maximum number of concurrent sessions on each port (50 - 2000; default=50)
    - TIMEOUT - Time interval in hundredths of a second after which ICON disconnects a client if no response from IMS
      - Range = 0 to 2,147,483,647
      - Default = 0 (no timeout)
    - Exits - List of from 1 to 15 user exits to process messages

# IMS Connect

- HWSCFG (continued)
  - DATASTORE control card
    - Translates logical “datastore” name passed by the TCP/IP client into IMS XCF member name and therefore IMS control region
    - Defines XCF group to join
    - Defines XCF member name of ICON for this connection
    - Datastore name becomes OTMA TPIPE name for CM1 messages
      - Client name is TPIPE name for CM0 messages
    - Defines name of DFSYDRU0 exit for messages from this OTMA client
    - There can be multiple DATASTORE cards

# IMS Connect

- To use ICON you must write one or more user exits
- Each exit gets control during ICON initialization and passes ICON two 8 character ID's that TCP/IP client applications can use to identify it
  - One is EBCDIC
  - One is ASCII
- The TCP/IP client must pass a valid exit name in the prefix of the input message
  - A bad ID will bring down the connection
  - If the ID is ASCII the exit usually assumes that the data is ASCII and should be translated to EBCDIC going to IMS and back to ASCII going to the TCP/IP client
  - You can also use a flag set by the client in the message prefix to determine if translation is required

# IMS Connect

- The exit gets control for input messages and builds the message to pass to OTMA
  - This includes the OTMA headers
- The exit translates the message from ASCII to EBCDIC if required
- The exit builds multiple segment input messages from one TCP/IP client message if required
- The exit can optionally invoke a security exit to get a RACF Utoken to pass to OTMA
  - or you can issue RACF calls directly in the ICON exit
- The exit passes message length and client name override to ICON in the HWSEXPRM parameter list
- The exit passes other information to ICON in the OTMA headers

# IMS Connect

- Format of input message from TCP/IP client to ICON to the User Exit:
  - LLLL - Total message length including suffix LLZZ
  - Prefix LLZZ - Length of message prefix
  - Required ICON fixed portion of prefix
  - Variable user portion of prefix
  - LLZZTRANCODE DATA - First message segment
  - LLZZDATA - Second message segment
  - …
  - LLZZDATA - Last message segment
  - Suffix LLZZ - x’00040000’

# IMS Connect

- Format of ICON portion of input message prefix:
  - 8 bytes - user exit ID
  - 4 bytes - reserved
  - 1 byte - flag
    - OTMA headers pre-built by client
    - Data translation done by client
  - 3 bytes - reserved
  - 8 bytes - client ID
    - Can be overridden by user exit
    - May become 16 bytes with new encoding standard

# IMS Connect

- Items you may want to include in the user portion of the prefix to pass to ICON and IMS in the OTMA headers:
  - Datastore name
    - ICON gets from OTMA User header built by the exit
  - Commit mode - 0, 1
  - Security Scope - N, C, S
  - Response Indicator - ACK, NACK, Resume TPIPE
  - Sync Level - Confirm, None
  - IOPCB LTERM override
  - IOCPB MODNAME override
  - RACF userid
  - RACF group
  - RACF new password !!!

Telcordia Technologies

Performance from Experience

# IMS Connect

- There is a sample exit provided with ICON
  - HWSSMPL0
- The Assembler Language sample exit is  full of lower case comments and variable names!!!!!!!!!!!!
  - It is just as ugly in IMS Connect as it was in ITOC
  - Real assembler programmers do not use lower case

# IMS Connect

- Format of input message from User Exit to ICON to IMS:
  - BPE Header
  - Up to first 32K bytes:
    - OTMA Headers
      - Control, State, Security, User
    - LLZZTRANCODE DATA
    - LLZZDATA’s
  - Suffix - x’0000’

  - BPE Header
  - Next 32K bytes
    - OTMA Header
      - Control
    - LLZZDATA’s
  - Suffix - x’0000’

  - As many 32K groups as needed with BPE header and  Suffix

# IMS Connect

- Format of error message from User Exit to ICON to return to client when rejecting input message:
  - BPE Header
  - LLLL - Total message length
  - Prefix LLZZ - Length of message prefix
  - Required ICON fixed portion of prefix
  - Variable user portion of prefix
    - Add Return Code, Reason Code, Error message text
  - LLZZDATA  - First message segment
  - …
  - LLZZDATA - Last message segment
  - LLZZ Suffix - x’00040000’
- I argued that since the message is going back to the client and not to IMS that a BPE header should not be required
  - I lost

# IMS Connect

- The exit gets control for output messages and strips the OTMA headers and builds the output message
- Invokes EBCDIC to ASCII translation if required
- Builds one message to the TCP/IP client from multiple segment IMS output messages if required

# IMS Connect

- Format of output message from IMS to ICON to User

  Exit:

  - OTMA Headers
    - Control, State, Security, User
  - LLZZDATA
  - OTMA Header
    - Control
  - LLZZDATA

# IMS Connect

- Format of the output message from User Exit to ICON to the TCP/IP Client can be anything you want
- Example:
  - LLLL - Total message length
  - Prefix LLZZ - Length of message prefix
  - Required ICON fixed portion of prefix
  - Variable user portion of prefix
    - Add Return Code, Reason Code, Output MODNAME, Error message text
  - LLZZDATA  - First message segment
  - …
  - LLZZDATA - Last message segment
  - LLZZ Suffix - x’00040000’

# IMS Connect

- Communication with ICON is via commands issued in response to an outstanding WTOR
- ICON Commands:
  - OPENDS - Start communication with a datastore
  - OPENPORT - Reestablish TCP/IP port communication
  - STOPDS - Stop communication with a datastore
  - STOPPORT - Stop TCP/IP port communication
  - STOPCLNT - Stop client communication
  - SETRACF - Set RACF flag to Y or N
  - VIEWDS - Display datastore status
  - VIEWPORT - Display TCP/IP port status
  - VIEWHWS - Display ICON status
  - CLOSEHWS - Terminate ICON

# IMS Connect

- Warning!!!
  - Sending a Commit Mode 0 message to a NONRESPONSE IMS transaction which really does not respond will leave the client waiting
    - Use commit mode 1 to get a DFS2082 message
    - Use the client timeout feature
    - Use IMS Connect Send-Only

# IMS Connect

- It is possible for a Socket Close issued by a client to reach ICON before all the data from the previous send
  - ICON gets very upset by this
- Affected by TCP/IP “SO_LINGER=Y/N, VALUE=N”
  - SO_LINGER=Y,VALUE=0
    - Immediate return to Client Code
    - Socket Close can lose previously sent data
  - SO_LINGER=N
    - Immediate return to Client Code
    - Socket Close can lose previously sent data
  - SO_LINGER=Y,VALUE=10
    - Return to Client Code when ACK is received from the host or after 10 seconds
    - Socket Close should not lose previously sent data
  - Thanks to Gerald Hughes of IBM ICON development for this tip

# IMS Connect

- ITOC requirements and their ICON status:
  - Updated User's Guide
    - IMS Connect is much better
  - SMP installed and maintained
    - IMS Connect
  - Create an ICON initialization exit like DFSINTX0
    - IMS Connect
  - Automatically connect/reconnect when IMS comes up
    - Known requirement
  - Support nonresponse transactions
    - IMS Connect
  - Support asynchronous output
    - IMS Connect

# IMS Connect

- ITOC requirements not fixed by ICON:
  - Command to activate exit trace
  - Command to activate ICON trace
  - Allow dynamic refresh of individual user exits
    - Known requirement
  - Allow more than 15 exits
  - Updated sample exit (WITH NO LOWER CASE)

# IMS Connect

- More ITOC requirements not fixed by ICON:
  - Support timeout request in the message prefix to tell ICON to timeout if IMS does not answer within a certain time
  - In an error situation display message in progress (dump format)
  - Better command interface (not reply to outstanding WTOR)
    - MVS modify interface
    - Command recognition character
    - Commands in a message from the client
  - Consistent commands
    - Start and Stop or Open and Close

# IMS Connect

- Telcordia is a vendor of IMS applications
- We have been building IMS applications for over 20 years
- We have already implemented 5 ICON interfaces to these applications

# IMS Connect

- In the first application a Windows client passes requests for data in the form of tag-value contracts to an existing NONRESPONSE IMS MPP
- The MPP puts the request in an IMS database
- A BMP job reads the request and returns the data via FTP
- The ICON exit does RACF Userid and Password verification
- The message is set to Commit Mode 1 so that IMS returns a DFS2082 message which is translated by the ICON exit into a successful completion message

# IMS Connect

- A second application stores very complex telephone network work requests in the form of text documents in an IMS database which customers wanted displayed at a workstation
- The workstation builds graphic representations of telephone network requests
- Screen scraping required many interactions and IMS transaction schedules
- New IMS contract interface transactions were developed to send large amounts of data to the workstation in one transaction
- Initial response time was reduced from 60 seconds to less than 5 seconds
- Complex process completion time was reduced from several minutes to 15 seconds
- Users LOVE it

# IMS Connect

- This same application wanted to provide a Web front-end to its normal IMS transactions
- They tried using 3270 screen scraping and vendor screen customization tools but were not satisfied
- They use ICON to send the normal transaction output to a workstation where it is formatted by application Java code
- Uses a special ICON exit to send and receive data from the workstation in EBCDIC

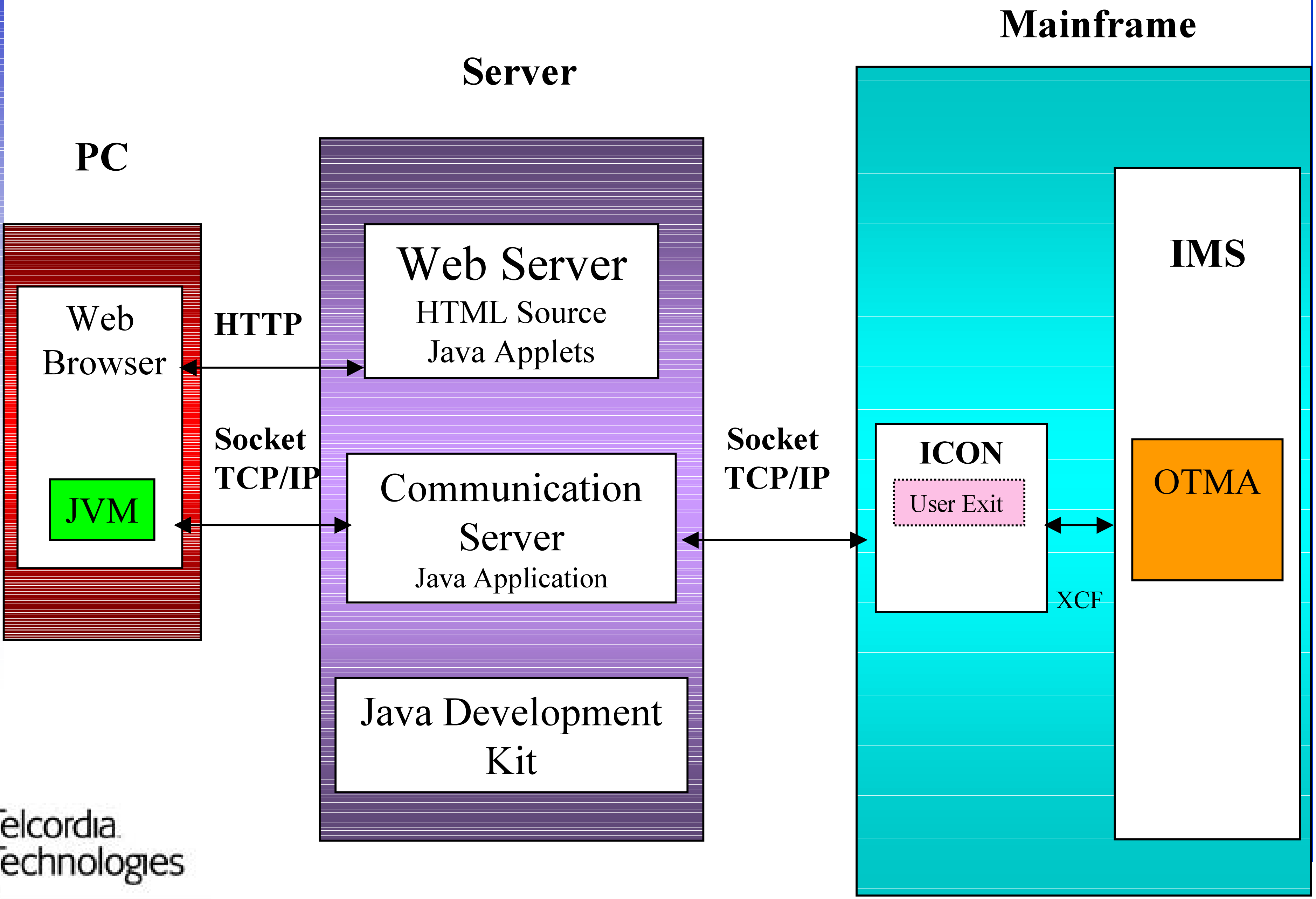
# IMS Connect

- A third application also wanted to build a Web front-end and were similarly dissatisfied with screen scrapers and customization tools
- New IMS contract interface transactions were developed to send large amounts of data to a workstation
- Created more efficient work flows
- Uses the ICON exit to provide data translation and security checking

# IMS Connect

- The Java application ICON client runs on a Sun Web server
- The application used the Java Client for IMS example downloaded from the Web
  - This sample was updated based on our experiences
- Java JDK 1.1
- Apache Web Server on a SUN Workstation

# IMS Connect



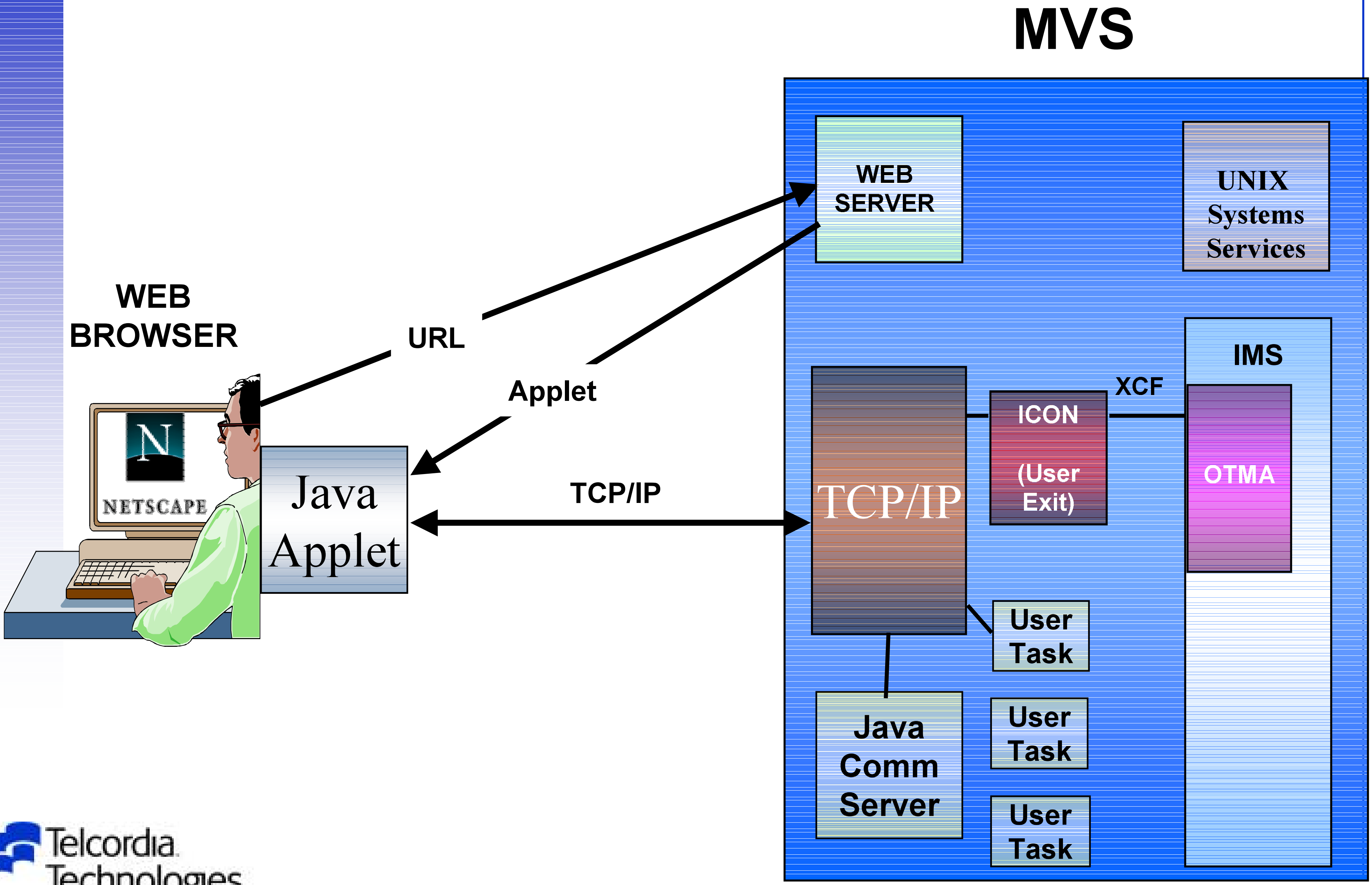
Telcordia Technologies

Performance from Experience

# IMS Connect

- The fifth ICON interface is used to provide a Web front-end to the IMS application that administers 800 numbers
- The user at a browser invokes a starting URL
- The Web server on MVS downloads the Java applet
- From there on the Java applet communicates directly with a user written Java communications server via TCP/IP
  - The Web server is no longer used
- The Java communications server spawns one task per user
- The user task communicates with IMS via TCP/IP and the ICON

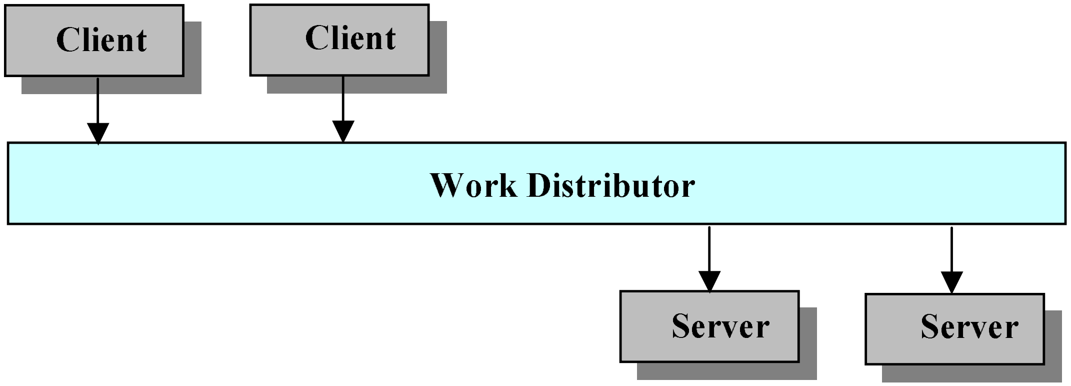
# IMS Connect



Telcordia Technologies
Performance from Experience

# CORBA - Overview

- Client / Server with a twist
- CORBA adds a “work distributor” (Object Request Broker)

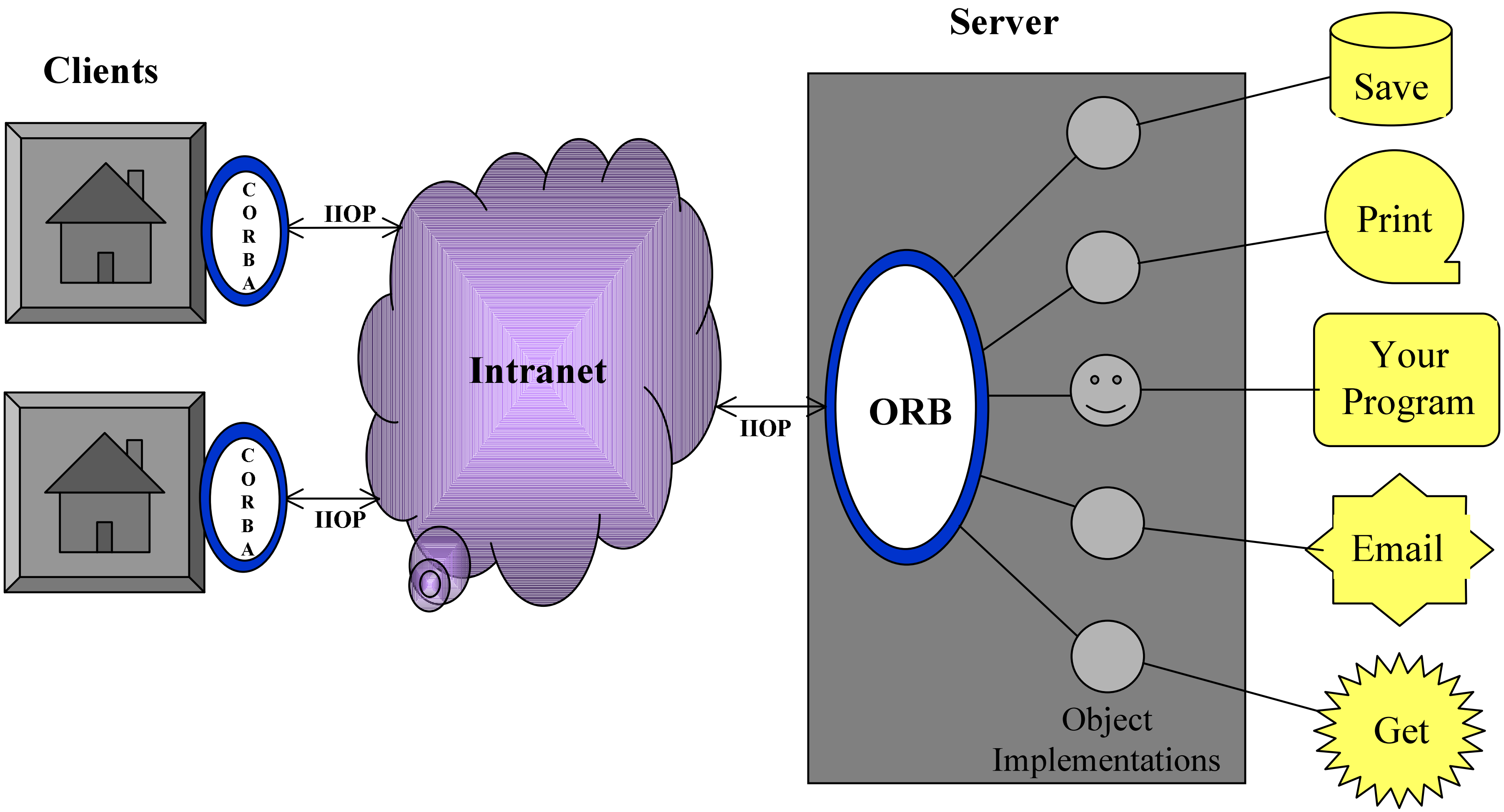


- Everyone can use it
  - Platforms
    - PC, Sun, HP, OS/390
  - Program Languages
    - Java, C++, etc.
- Based on the Object Oriented Data modeling

Telcordia Technologies

Performance from Experience

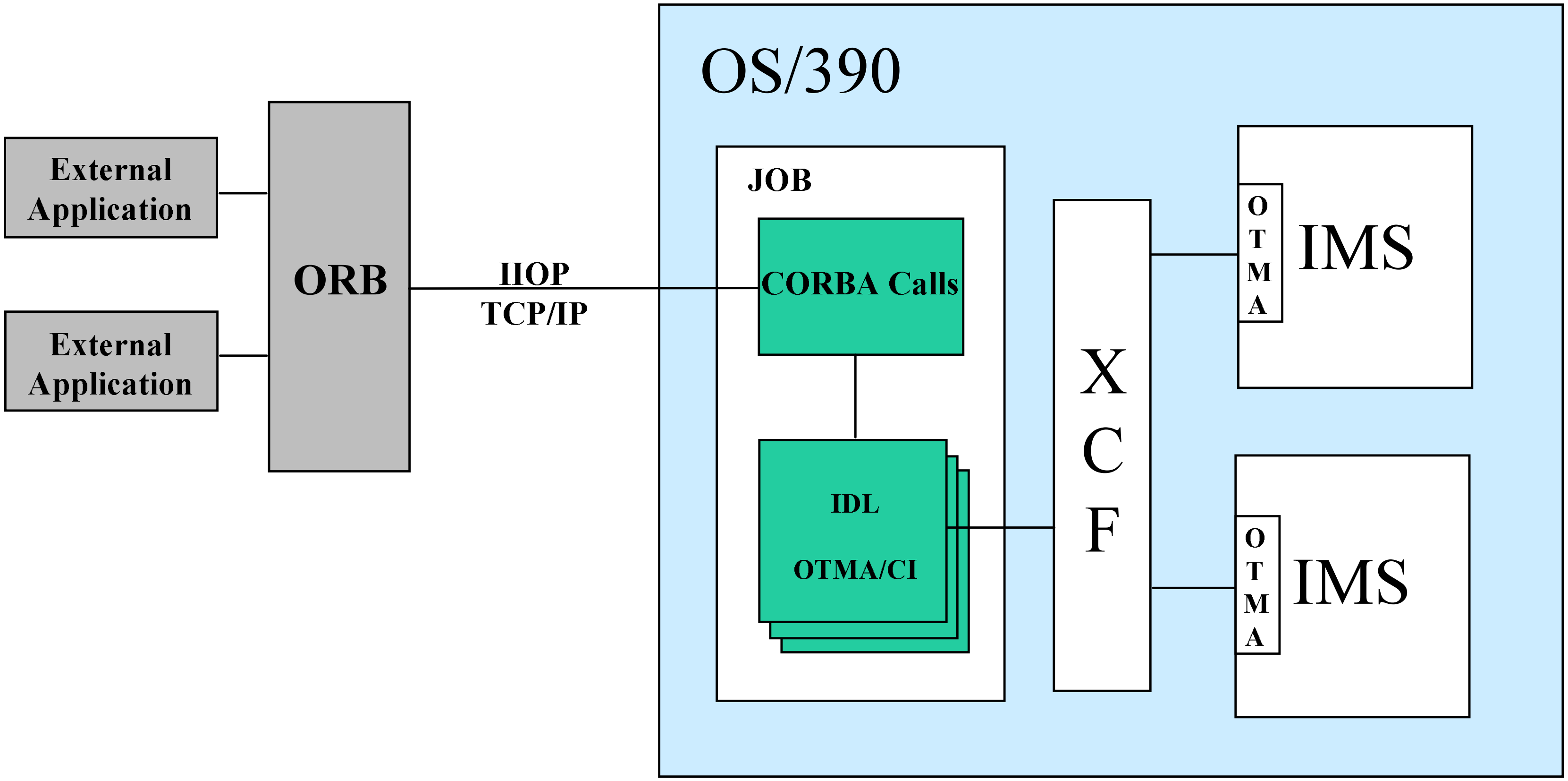
# CORBA - The Picture



Telcordia Technologies

Performance from Experience

# CORBA to IMS OTMA Interface



Telcordia Technologies

Performance from Experience

# OTMA Callable Interface

- Writing your own OTMA client without OTMA Callable Interface is extremely difficult
  - You have to know OTMA protocols
  - You have to know XCF protocols
  - You have to write in Assembler
  - You have to be authorized
- The OTMA Callable Interface makes it easy to write an OTMA client
  - You do not need to know OTMA or XCF protocols
  - You can write in C or C++
    - More languages will come later
  - You do not have to be authorized

# OTMA Callable Interface

- Requires IMS 6.1 + PQ17203 +PQ19424 + PQ20680 + PQ39044 + + +
- Requires running an SVC initialization program each time MVS is IPL’ed
- See www.software.ibm.com/data/ims/otmaci.html for more details and documentation
- See www.share.org/proceedings/sh94/share00w.htm for a  SHARE presentation
  - Look in the IMS Project

# OTMA Callable Interface

- OTMA/CI functions
  - otma_create
    - Create storage structures to support communications but does not establish a connection with IMS
  - otma_open
    - Establishes a connection with IMS
  - otma_alloc
    - Create an independent transaction session
  - otma_send_receive
    - Sends to IMS and passes parameters for receive functions
  - otma_free
    - Releases the independent transaction session
  - otma_close
    - Ends the connection with IMS

# OTMA Callable Interface

- APAR PQ32398 introduces major enhancements to the OTMA Callable Interface
  - otma_send_async
    - Send an IMS transaction or command using the commit-then-send protocol
  - otma_receive_async
    - Receives an unsolicited message or a commit-then-send output message from IMS
  - otma_openx
    - Establishes a connection with IMS + DFSYDRU0 Exit name override
  - otma_send_receivex
    - Sends to IMS and passes parameters for receive functions + supports OTMA user data

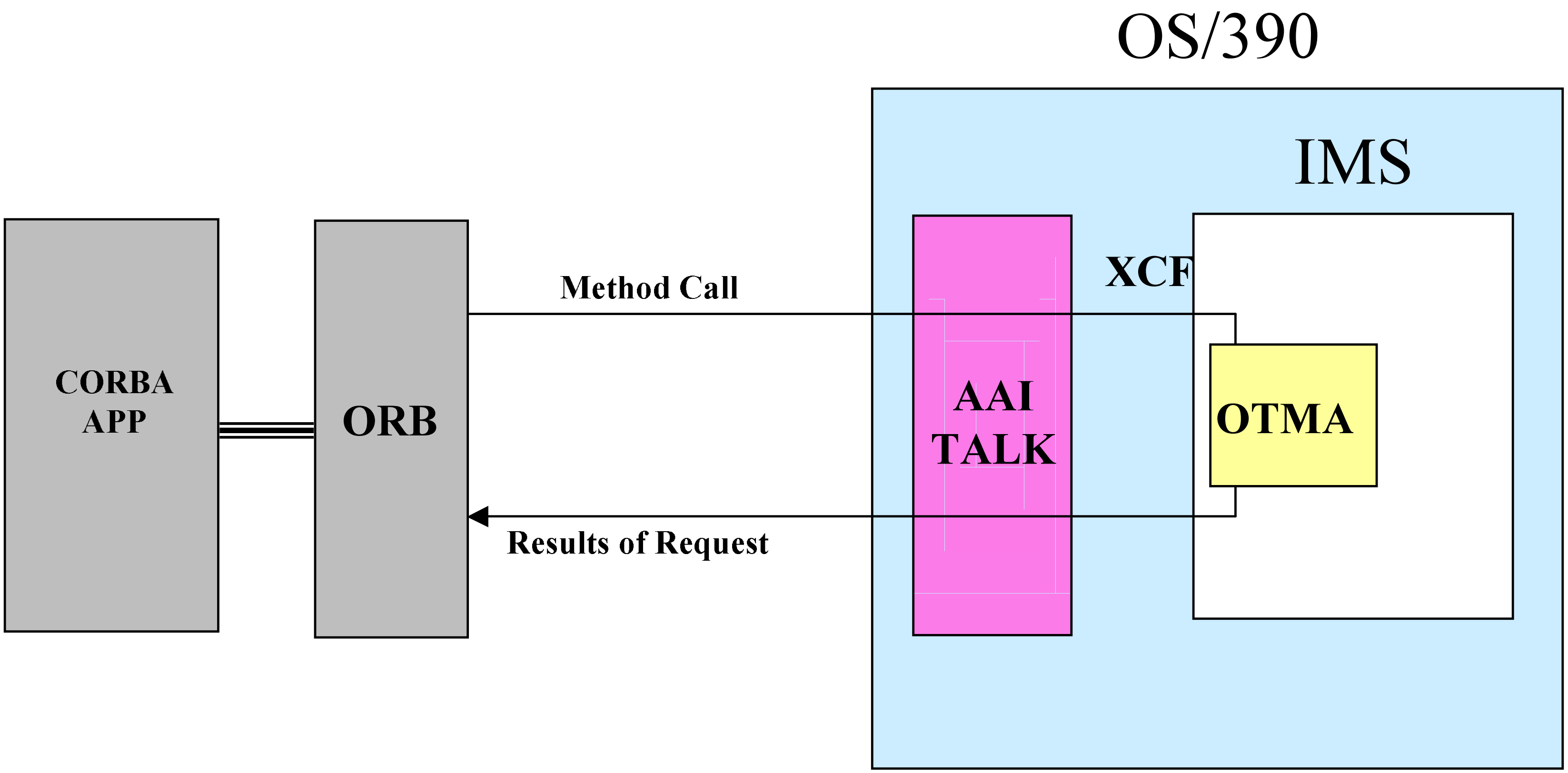
# Telcordia™ Advanced Application Interface

(Telcordia™ Advanced Application Interface)

- Telcordia IMS CORBA Interface
  - Visibroker ORB running native under OS/390
  - Methods written to schedule IMS transactions
    - Send with no reply
    - Send and return reply
  - Methods written to receive unsolicited IMS transaction requests
  - In production in first location

# Telcordia™ Advanced Application Interface - TALK

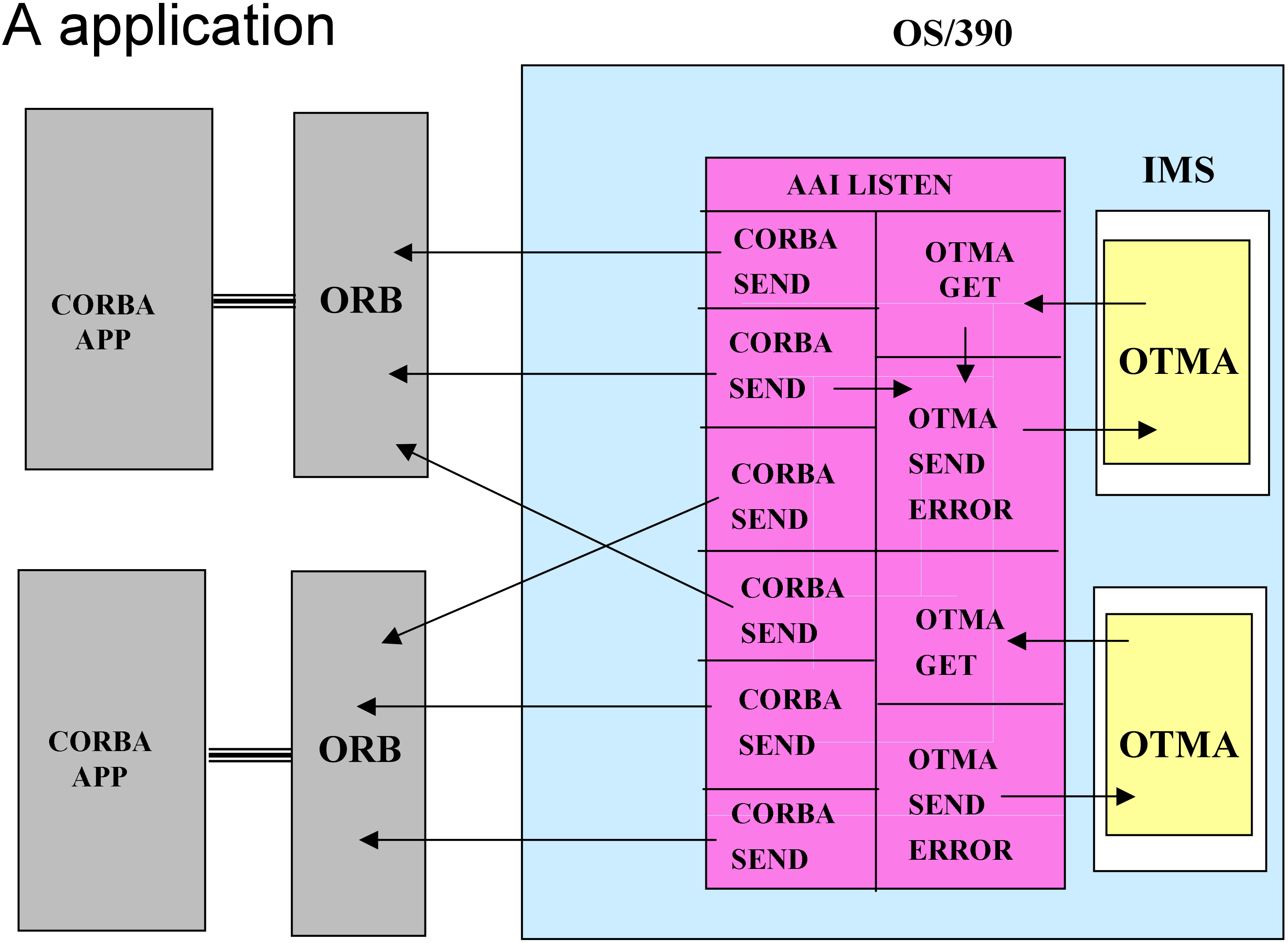
- Allows a CORBA application to send a request to IMS and receive the reply



Performance from Experience

# Telcordia™ Advanced Application Interface - LISTEN

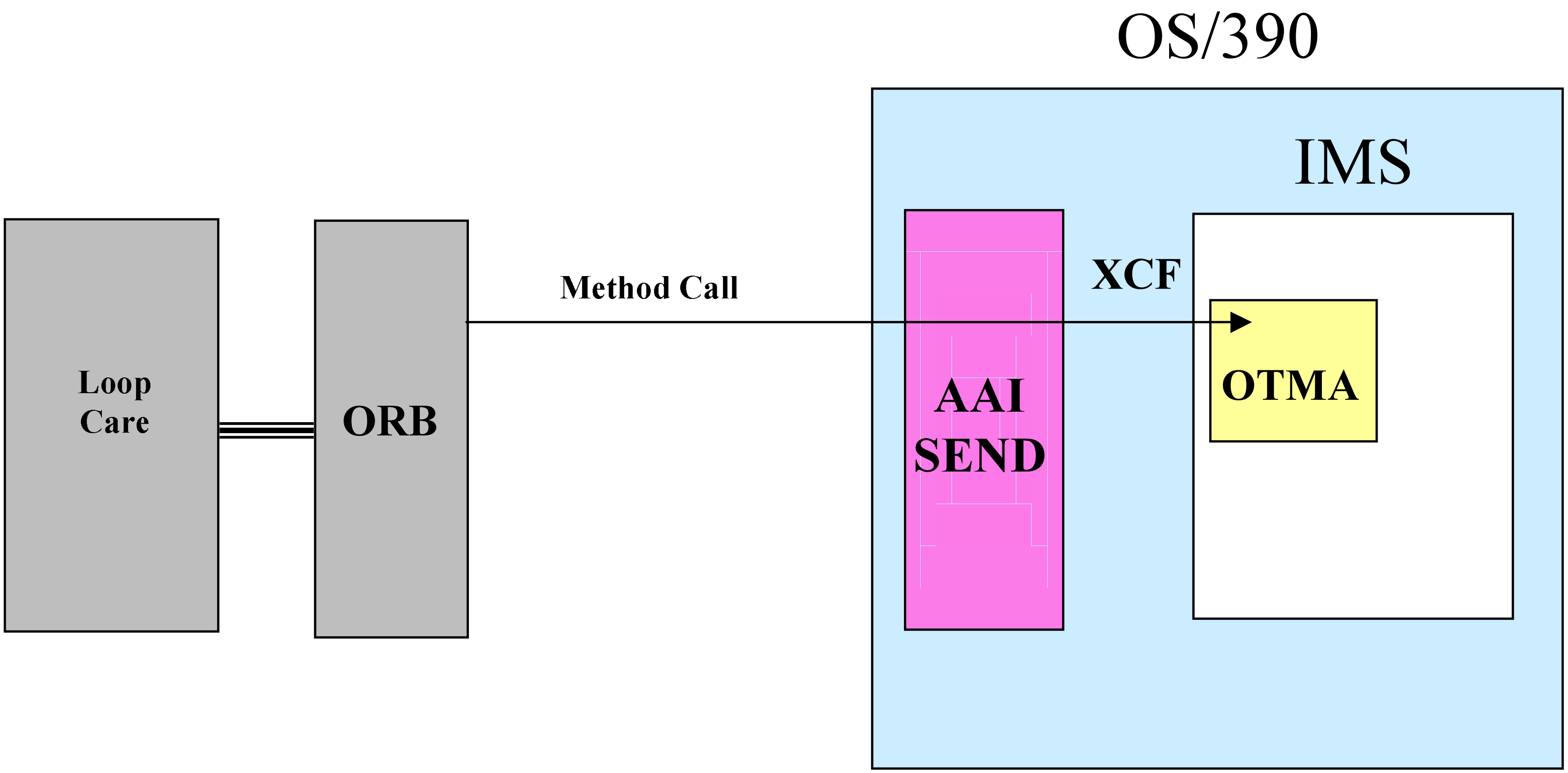
- Allows IMS applications to send requests to a remote CORBA application



Telcordia Technologies
*Performance from Experience*

# Telcordia™Advanced Application Interface - SEND

- Allows a CORBA application to send a responses to IMS



Telcordia Technologies
Performance from Experience

# References

- Telcordia Technologies Inc.
  - www.telcordia.com
- IMS
  - www.software.ibm.com/data/ims
- OTMA/CI
  - www-4.ibm.com/software/data/ims/about/otmaci/document/IL0TMCI.HTM
- MQSeries
  - www.software.ibm.com/ts/mqseries
- OMG
  - www.omg.org
- Orbix
  - www.iona.com
- Component Broker
  - www.s390.ibm.com/cb/
- VisiBroker
  - www.inprise.com/visibroker

# Glossary

- OTMA
  - Open Transaction Manager Access
    - OTMA is a transaction based, connectionless client / server protocol interfacing with IMS
    - Part of IMS 6.1
      - PTFs required for advanced OTMA/CI and OTMA exits
- OTMA Client
  - Program that exists outside of IMS that communicates with IMS through OTMA
- OTMA/CI
  - OTMA Callable Interface
    - OTMA/CI is a high-level interface for easy access to IMS transactions and commands for other OS/390 subsystems

# Glossary

- ITOC
  - The IMS TCP/IP OTMA Connector is an OTMA Client
    - The ITOC is a TCP/IP server that enables TCP/IP clients to exchange messages with IMS
  - Beginning with IMS Version 7, this product will be called IMS Connect, and will be offered as a separate product

# Glossary

- CORBA
  - **C**ommon
    - “Belonging equally to or shared equally by two or more”
  - **O**bject
    - Object Oriented Programming
  - **R**equest
    - Ask for work to get done
  - **B**roker
    - Distribute work
  - **A**rchitecture
    - An architecture, idea, definition

Telcordia Technologies

Performance from Experience

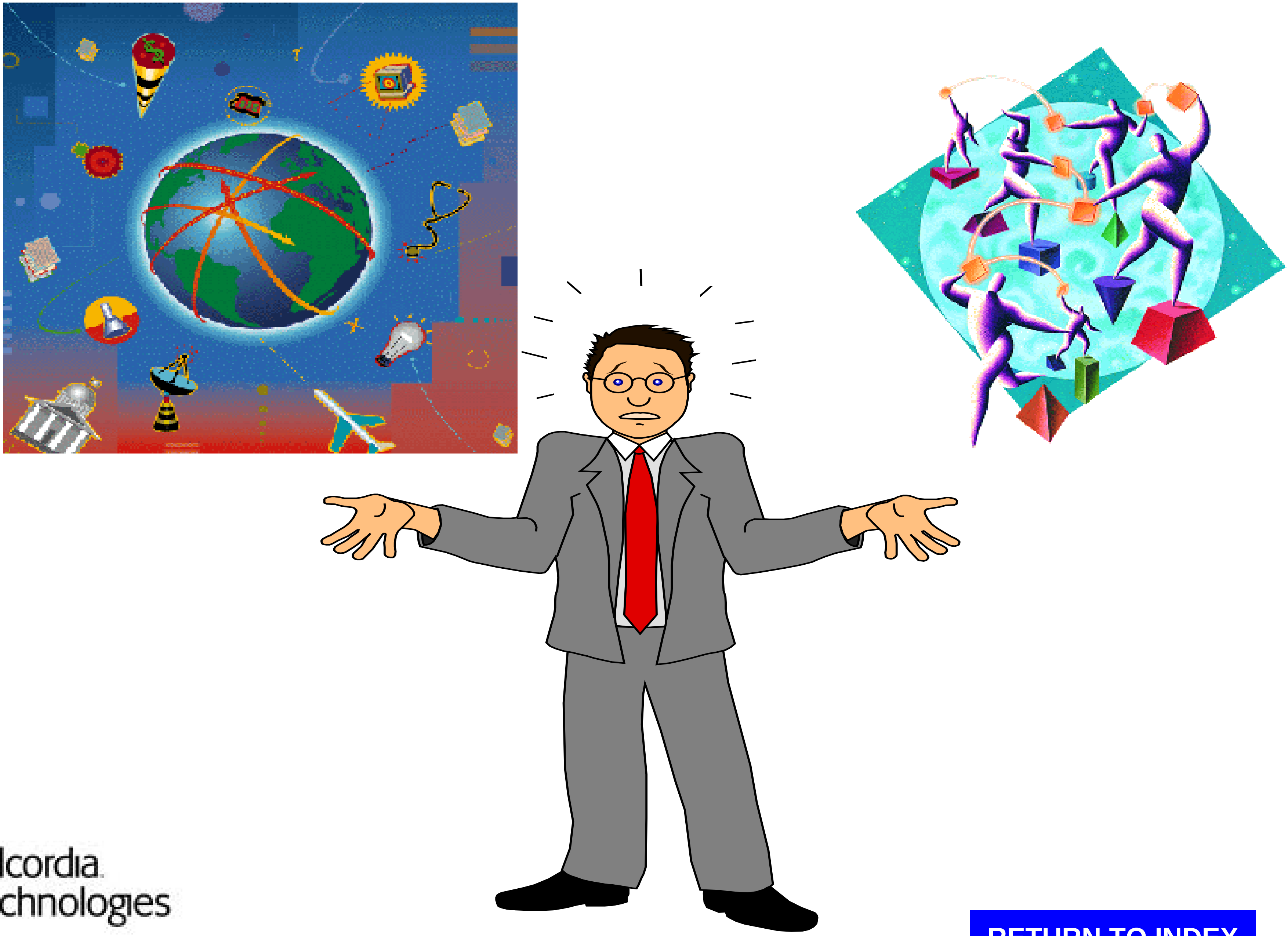
# Glossary

- ORB
  - Object Request Broker
    - If facilitates the communication between clients and servers
- GUI
  - Graphical User Interface
    - Typically refers to a graphical windows point and click front end
- IDL
  - Interface Definition Language
    - A standard language to describe application interfaces

Telcordia Technologies

Performance from Experience

# Questions?



Telcordia Technologies

*Performance from Experience*

RETURN TO INDEX