Telcordia Technologies

Performance from Experience

IMS TECHNICAL CONFERENCE

Customer Experience: MQSeries and IMS

Session E92

Steve Nathan

stephen.nathan@telcordia.com

Gary Ward

gary.ward@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 MQSeries.
- 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 MQSeries 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.

Telcordia Technologies

Performance from Experience

Acknowledgements

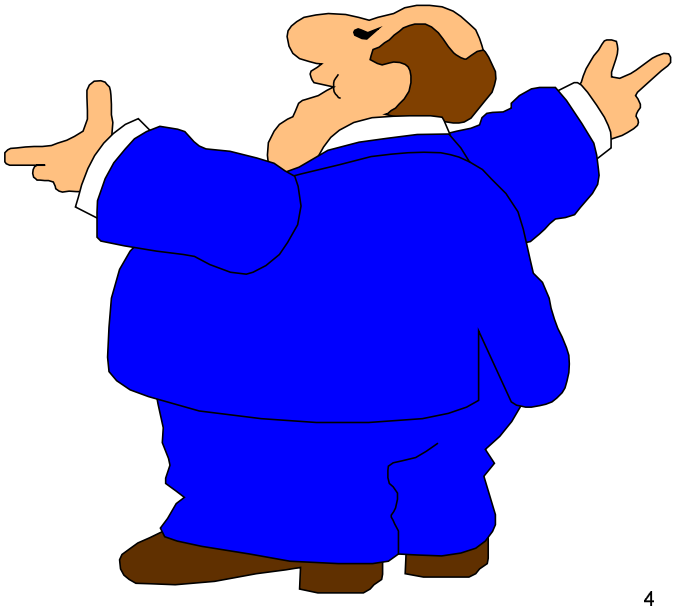
- Special thanks to:
  - Jack Yuan and his team from IBM IMS OTMA development for all the enhancements in PQ32402 and all of the other enhancements in OTMA
  - John Jones and Bill Millar and their team at IBM Hursley Labs for their work on the MQSeries IMS Bridge and sharing their knowledge of how it really works

Telcordia Technologies

Performance from Experience

Presentation Outline

- Introduction
- IMS Application Access to MQSeries Messages
- Connecting MQSeries and IMS via ESS
- Using MQSeries API with IMS Applications
- MQSeries IMS Trigger Monitor
- IMS BMP Monitor
- IMS OTMA Interface
- MQSeries IMS Bridge
- IMS OTMA Security
- Telcordia Experience with the MQSeries IMS Bridge
- Sources of Documentation



Telcordia Technologies

Performance from Experience

Introduction

- Telcordia has given several presentations about its experiences using IMS and MQSeries
- There have been many changes and enhancements in the last year
- We have learned a great deal about this topic in the last year
- This presentation will concentrate on what is new and what we have learned about IMS and MQSeries

Telcordia Technologies
Performance from Experience

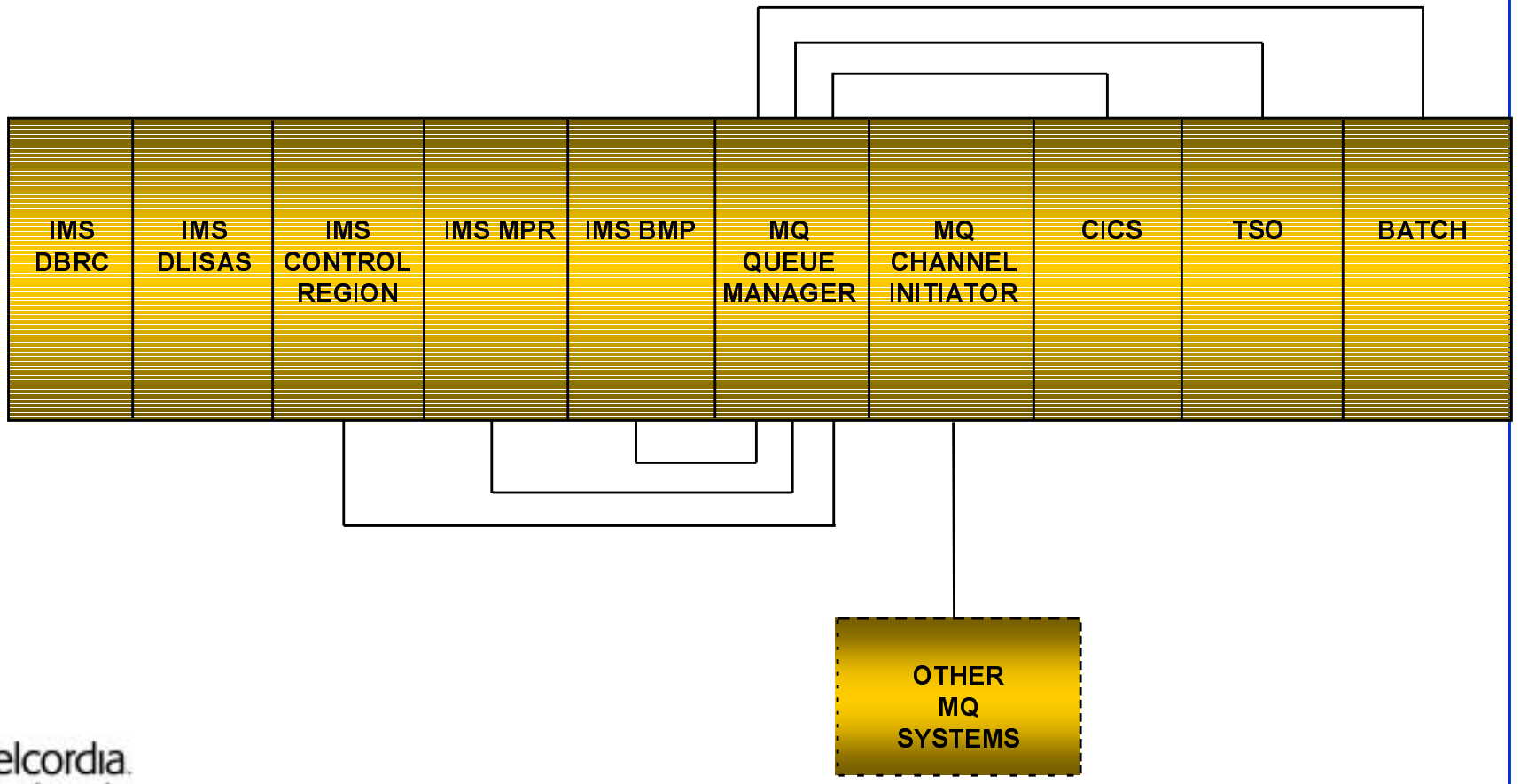
IMS Application Access to MQSeries Messages

- IMS applications can access MQSeries messages in 2 ways
  1. The IMS application uses the MQSeries API to Get and Put messages with syncpoint coordination with IMS
     - Requires connecting MQSeries to IMS via ESS
     - IMS Batch or BMP
       - Could be MQSeries triggered
     - IMS MPP
       - MQSeries message inserted to the IMS Message Queue by BMP
         - Trigger message
         - Real Message
  2. MQSeries IMS Bridge puts message on the IMS Message Queue via OTMA
     - Does not require connecting MQSeries to IMS via ESS
     - Requires OTMA configuration in CSQZPARM

Telcordia Technologies
Performance from Experience

Connecting MQSeries and IMS via ESS

- MQSeries for OS/390 attaches to IMS just like DB2 using the external subsystem (ESS) interface



Telcordia Technologies
Performance from Experience

Connecting MQSeries and IMS via ESS

- Define MQSeries to IMS by adding ESS information to the IMS PROCLIB

  FORMAT: SST=,SSN=,LIT=,ESMT=,RTT=,REO=,CRC=
  - SST: Subsystem Type - “MQS”
  - SSN: Subsystem Name - MQSeries for MVS/ESA subsystem
  - LIT: Language Interface Token - See CSQQDEFV
  - ESMT: External Subsystem Module Table - “CSQQESMT”
  - RTT: Resource Translation Table -Not Used by MQSeries
  - REO: Region Error Option - “R”, “Q”, or “A”
  - CRC: Subsystem Recognition Character - Not Used by MQSeries
    - The /SSR command is not supported

Telcordia Technologies
Performance from Experience

Connecting MQSeries and IMS via ESS

- Place the MQSeries authorized library (HLQ.SCSQAUTH) in the IMS control region and dependent region DFSESL concatenations
- Copy module CSQQDEFV from HLQ.SCSQASMS to be customized, assembled, and linked into an authorized library in the IMS control region STEPLIB concatenation

```
CSQQDEFV CSECT
         CSQQDEFX NAME=CSQ1,LIT=LIT1,TYPE=DEFAULT
         CSQQDEFX NAME=CSQ3,LIT=LIT2
         CSQQDEFX TYPE=END
```

Telcordia Technologies
Performance from Experience

Connecting MQSeries and IMS via ESS

- Subsystem Connection

```
/DIS SUBSYS ALL

SUBSYS   CRC  REGID  PROGRAM  LTERM     STATUS
CSQ3     !                              CONN
CSQ1     <                              CONN
DB2R     =                              CONN
                  1                     CONN
                  5                     CONN
```

Telcordia Technologies
Performance from Experience

Using MQSeries API with IMS Applications

- Calls to MQSeries, IMS and DB2 can be made within the same unit of work (UOW)
  - MQSeries API calls
  - IMS IOPCB calls
  - IMS alternate output PCB calls
  - IMS database calls
  - DB2 calls
- An IMS commit is also an MQSeries and DB2 commit
  - SYNC, CHKP, GU to IOPCB (MODE=SNGL), normal program termination
- An IMS backout (ROLB) is also an MQSeries and DB2 backout
- Any IMS abend is also an MQSeries and DB2 backout
  - ROLL, miscellaneous abends

Telcordia Technologies

Performance from Experience

Using MQSeries API with IMS Applications

- At normal syncpoint....
  - IMS input message is dequeued
  - MQSeries input messages marked with SYNCPOINT, NO_SYNCPOINT, and MARK_SKIP BACKOUT are dequeued
  - IMS EXPRESS and NON-EXPRESS output messages are sent
  - MQSeries output messages marked with SYNCPOINT and NO_SYNCPOINT are sent
  - IMS database updates are committed
  - DB2 updates are committed
  - If the IMS application is message driven the MQSeries connection handle is closed
    - For security reasons
      - Connection security is by userid
      - Each message can be from a different userid

Telcordia Technologies
Performance from Experience

Using MQSeries API with IMS Applications

- At abnormal termination or ROLx....
  - IMS input message is dequeued
    - IMS 6.1 has Non-Discardable Message Exit
  - MQSeries input messages marked with SYNCPOINT are re-queued
  - MQSeries input messages marked with NO_SYNCPOINT are dequeued
  - MQSeries input messages marked with MARK_SKIP_BACKOUT are
    - Passed to the next UOW if this was the first UOW
    - Re-queued if this was the second UOW
  - IMS NON-EXPRESS messages are discarded
  - IMS EXPRESS output messages are sent
  - MQSeries output messages marked with SYNCPOINT are discarded
  - MQSeries NO_SYNCPOINT output messages are sent
  - IMS database updates are backed out
  - DB2 updates are backed out

Telcordia Technologies

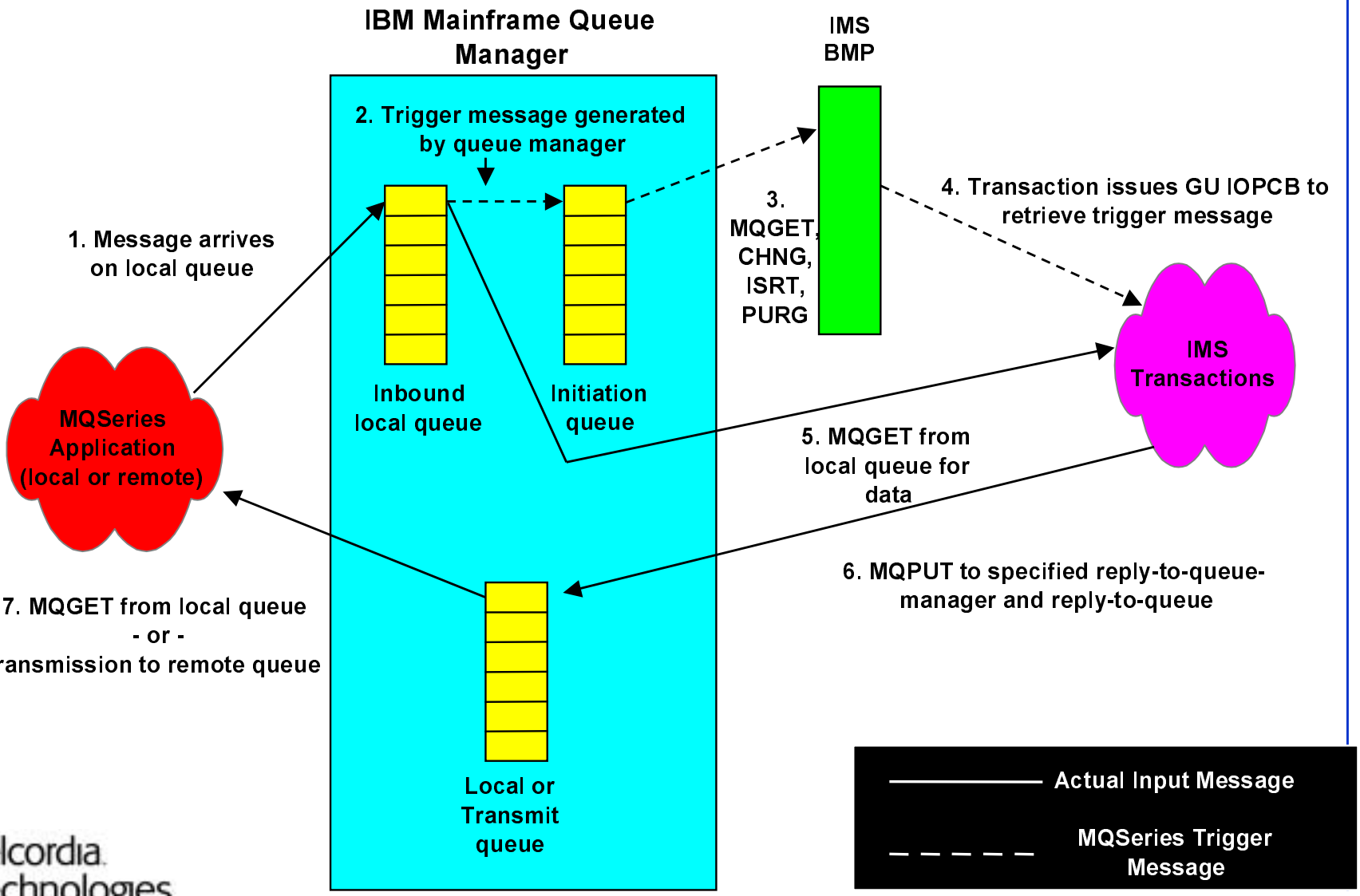
Performance from Experience

Using MQSeries API with IMS Applications

- Getting the default queue manager name is not straightforward...
  - MQCONN using default name (blank)
  - MQOPEN the Queue Manager
    - MQOD Objectype = MQOD_Q_MGR
    - MQOD Objectname = blanks
    - MQOO_INQUIRE
  - MQINQ for object name

Telcordia Technologies
Performance from Experience

MQSeries IMS Trigger Monitor



Telcordia Technologies
Performance from Experience

MQSeries IMS Trigger Monitor

- IMS triggering flow - MQSeries
  - ❶A message arrives on an MQSeries local queue
  - ❷A “trigger message” is sent to an MQSeries initiation queue
  - ❸A long-running non-message driven BMP waiting on the initiation queue reads the trigger message and inserts it to the IMS trancode named in the MQSeries process definition
    - This is an IBM supplied trigger monitor application (CSQQTRMN)

Telcordia Technologies
Performance from Experience

MQSeries IMS Trigger Monitor

- IMS triggering flow - IMS

❹The IMS transaction is scheduled and does a GU to the IOPCB, retrieving the trigger message

- The trigger message identifies the MQSeries queue manager and local queue which caused the trigger to be pulled

❺The IMS transaction connects to the queue manager and reads the message from the MQSeries local queue via MQGET

❻The IMS transaction responds by inserting a message to an MQSeries queue via MQPUT

- The MQSeries message descriptor identifies a reply-to queue manager and queue

Telcordia Technologies
Performance from Experience

IMS BMP Monitor

- User written IMS BMP monitor program
  - It is easy to write your own IMS BMP monitor
  - Telcordia developed one before the MQSeries IMS Trigger Monitor program was available
  - One enhancement could be to have one BMP wait on multiple queues using the MQGMO_SET_SIGNAL option (ECB)
  - Here's how to do it:
    - Write IMS BMP with MQI calls and an alternate output modifiable PCB (ALTPCB)
    - Wait on MQSeries initiation queue(s) or local queue(s)
      - Initiation queues hold trigger messages; local queues hold "real" messages
    - Insert messages to IMS ALTPCB

Telcordia Technologies
Performance from Experience

MQSeries - Telcordia Web IMS Application

- Telcordia had an IMS application that generated customer reports
- The customer would phone Telcordia with a request for a report
- Telcordia would generate the report and mail it to the customer

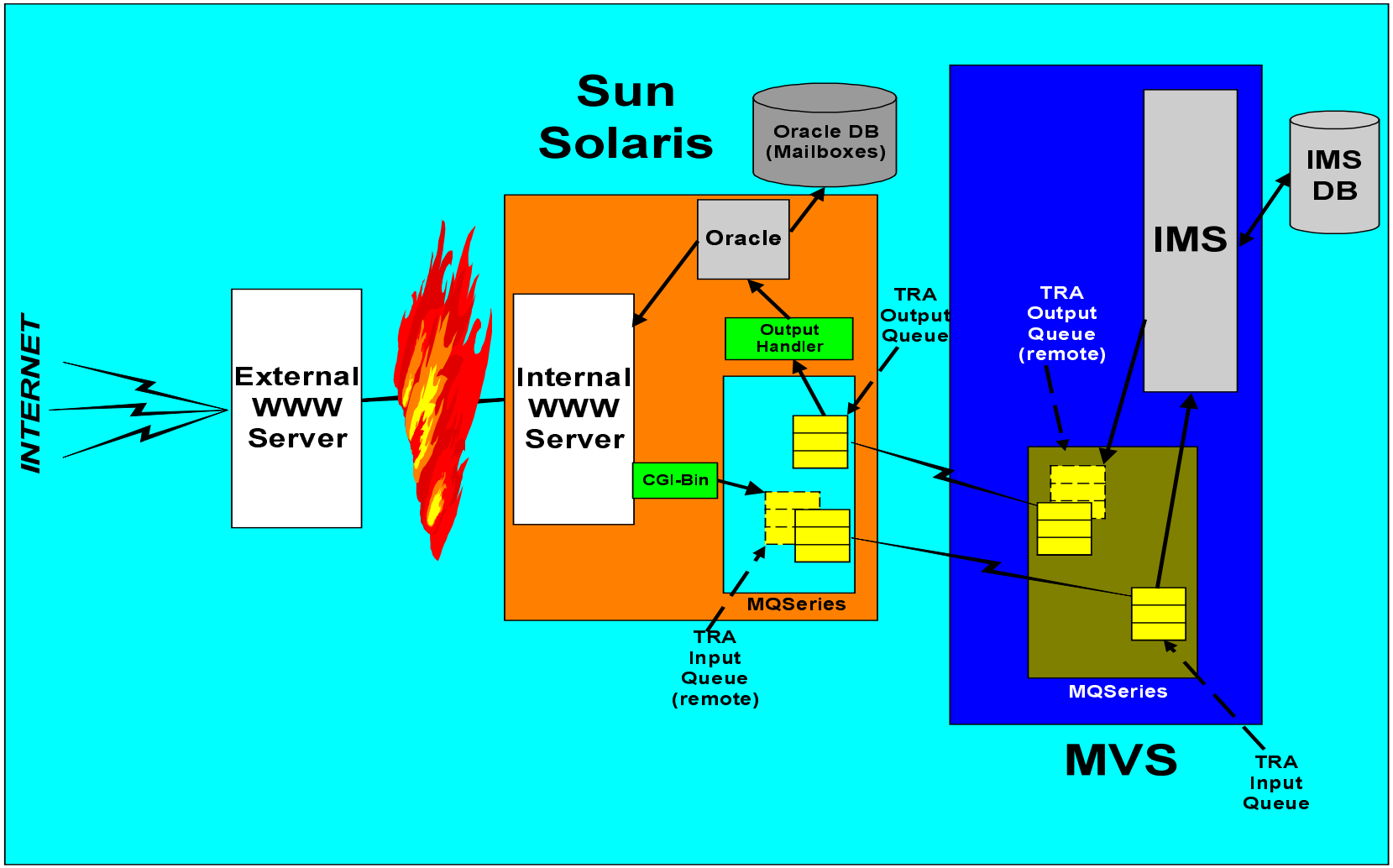
Telcordia Technologies
Performance from Experience

MQSeries - Telcordia Web IMS Application

- A new customer Web interface was built
  - Customer requests report via HTML
  - Request is sent to MQSeries queue on MVS
  - Long running BMP reads MQSeries queue for request
  - BMP reads data from IMS and formats report
  - BMP sends report via MQSeries to server
  - CGI-BIN program on server puts report in customer mailbox

Telcordia Technologies
Performance from Experience

MQSeries - Telcordia Web IMS Application



Telcordia Technologies

Performance from Experience

IMS OTMA Interface

- 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

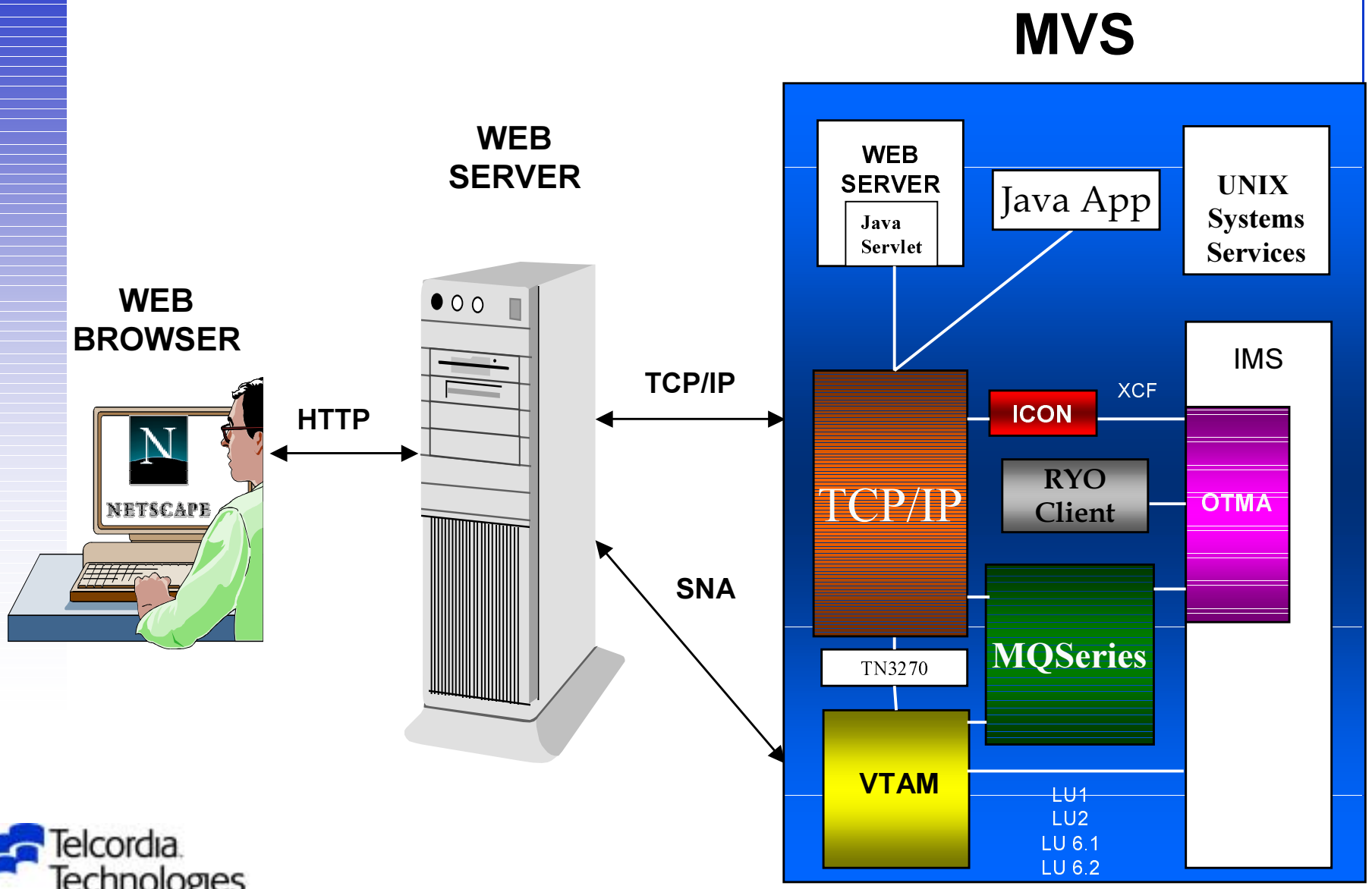
Telcordia Technologies
Performance from Experience

IMS OTMA Interface

- IMS Connect is an IBM provided OTMA client for TCP/IP
  - This was the IMS TCP/IP OTMA Connection (ITOC)
  - See Session E93
- MQSeries includes an IMS OTMA client
  - "MQSeries-IMS Bridge"
- You can write your own OTMA client using the OTMA Callable Interface
  - See Session E93

Telcordia Technologies
Performance from Experience

IMS OTMA Interface



Telcordia Technologies
Performance from Experience

IMS OTMA Interface

- Tpipes (Transaction Pipes)
  - OTMA equivalent of LTERMs
  - Input Tpipe names are specified by the client
  - Asynchronous output Tpipe names
    - The destination in the CHNG call for modifiable ALTPCB's
    - The destination name in static ALTPCB's
    - Can be overridden by the DFSYDRU0 exit
  - Control blocks are dynamically created by IMS
  - Tpipe name must be unique only within a client (multiple clients can use the same Tpipe name)
    - IMS recognizes Tpipes as **XCFmember**.**Tpipename**
  - A Tpipe name cannot be the same name as an IMS transaction
  - Client manages the use of Tpipes (number, routing, etc.)

Telcordia Technologies
Performance from Experience

IMS OTMA Interface

- Tpipes
  - There are two kind of TPIPES: SYNChronized and Non-SYNChronized
    - SYNChronized Tpipes exchange sequence numbers and can be resynchronized across failures
    - SYNChronized Tpipes will survive an IMS warm start but will be deleted during an IMS cold start
    - Non-SYNChronized Tpipes will be deleted during an IMS warm start

Telcordia Technologies
Performance from Experience

IMS OTMA Interface

- OTMA Commit Mode
  - Specified by OTMA client for each message
  - Similar to APPC-IMS Commit Mode
  - Commit Mode 0 - Commit-then-send
    - IMS sends output after syncpoint is complete
    - OTMA requires ACK to dequeue message
  - Commit Mode 1 - Send-then-commit
    - IMS sends output and waits for ACK before syncpoint is complete
    - Increases region occupancy
    - Messages are not queued

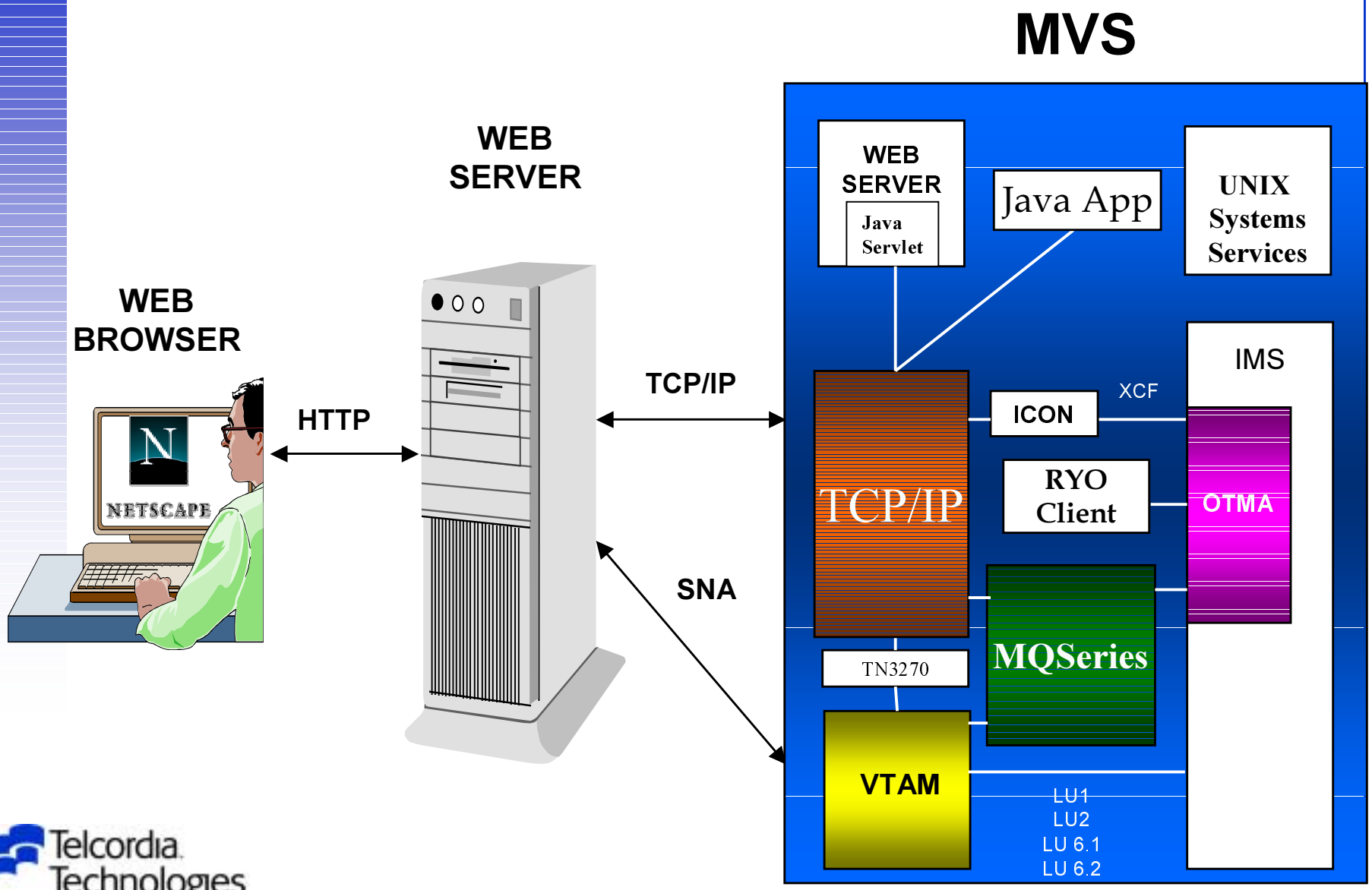
Telcordia Technologies
Performance from Experience

IMS OTMA Interface

- The maximum OTMA input segment size is 32767
  - LLZZ + 32763 bytes of user data
  - IMS will break up the segment into pieces that fit into the LGMSG and chain them together
- The maximum OTMA total message length to input to MQSeries is the MQ Max Message Length
  - This is usually 4MB
  - MQSeries long message support increases this to 100MB
    - This has not been tested with OTMA
- Thanks to Bob Millar of IBM Hursley for this information

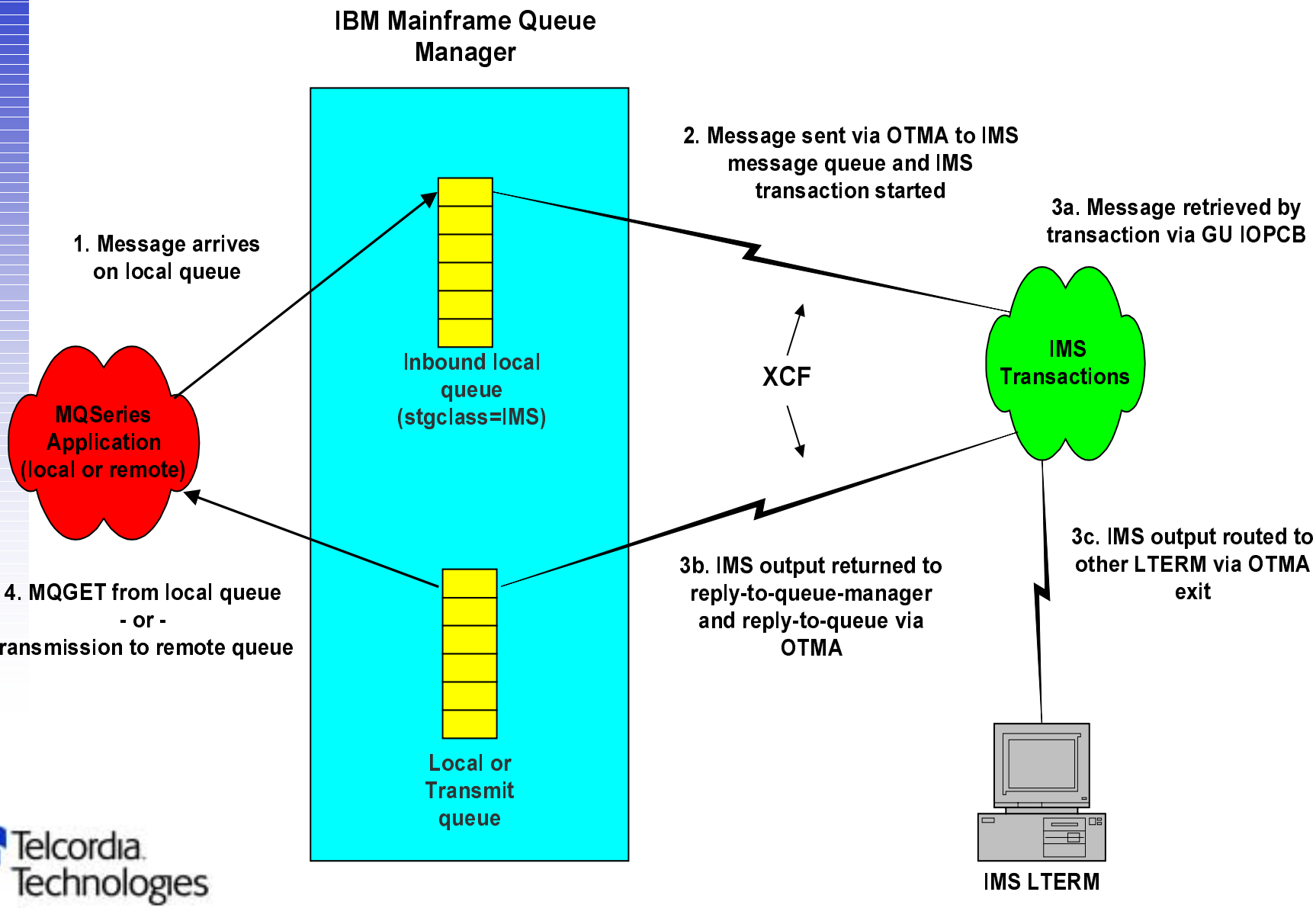
Telcordia Technologies
Performance from Experience

IMS OTMA Interface



Telcordia Technologies
Performance from Experience

MQSeries IMS Bridge



Telcordia Technologies
Performance from Experience

MQSeries IMS Bridge

- When the message arrives in MQSeries it will be sent via XCF to the IMS OTMA interface.
- Message may be:
  - an IMS transaction
  - an IMS command
  - NOT a message to an IMS LTERM
- IMS will put it on the IMS message queue
- The application will do a GU to the IOPCB to retrieve the message
- Output to the IOPCB returns to MQSeries which sends it to the ReplyToQueue
- This is very similar to the implicit LU6.2 process
- A remote queue manager can send a message to a local queue destined for IMS via OTMA

Telcordia Technologies
Performance from Experience

MQSeries IMS Bridge

- One MQSeries queue manager can connect to multiple IMS control regions
- One IMS control region can connect to multiple MQSeries queue managers
- MQSeries and all IMS control regions must be in the same XCF group
  - I am hoping to get this fixed
- MQSeries and IMS can be on different MVS copies in the same Sysplex
- MQSeries IMS Bridge start and stop events are sent to the SYSTEM.ADMIN.CHANNEL.EVENT.QUEUE

Telcordia Technologies
Performance from Experience

MQSeries IMS Bridge

- Define MQSeries to XCF in CSQZPARM
  - OTMACON keyword on CSQ6SYSP macro
    - OTMACON(Group,Member,Druexit,Age,TpipePrefix)
      - Group=XCF group
      - Member=XCF member
      - Druexit=IMS exit to route output (overrides DFSYDTx)
      - Age=how long a userid from MQSeries is considered previously verified in IMS
      - TpipePrefix=three character prefix for Tpipe name
        - To avoid collision with IMS transaction code names
  - Member CSQ4ZPRM in data set hlq.SCSQPROC has default CSQZPARM members you can use to build your members

Telcordia Technologies
Performance from Experience

MQSeries IMS Bridge

- Define one or more storage classes with the XCFGNAME and XCFMNAME parameters of the IMS systems to which you will connect
  - DEFINE STGCLASS(IMSA) -
    PSID(02) –
    XCFGNAME(XCFGROUP)
    XCFNAME(XCFIMSA)
- Define local queue(s) referencing the storage classes
  - DEFINE QLOCAL(MQID_TO_IMSA) –
    STGCLASS(IMSA)
- Define reply-to-queue(s)
  - DEFINE QLOCAL(MQID_FROM_IMSA)

Telcordia Technologies
Performance from Experience

MQSeries IMS Bridge

- Confirm On Arrival (COA) is provided when the message reaches the IMS queue
- Confirm On Delivery (COD) is not available
  - The message is rejected if this option is specified
- Expiry
  - A message can expire before reaching the IMS Bridge
  - It can not expire once it has reached the IMS Message Queue

Telcordia Technologies
Performance from Experience

MQSeries IMS Bridge

- The sending application can optionally provide an IMS sub-header (MQIIH)
  - Several reserved fields...
  - IMS LTERM name to put in IOPCB
  - MFS Format name to put in IOPCB (no MFS formatting is actually done)
  - Reply To Format (MQSeries format name)
  - Authenticator (RACF password or PassTicket)
  - Transaction Instance ID (if IMS conversational)
  - Transaction State (conversational or not, architected command)
  - Commit Mode (commit-then-send or send-then-commit)
  - Security Scope (ACEE in control region or in control region and dependent region)
    - Only honored if /SEC OTMA PROFILE used

Telcordia Technologies
Performance from Experience

MQSeries IMS Bridge

- Specify the presence of the MQIIH sub-header by using the MQMD.Format=MQFMT_IMS (“MQIMS”)
  - WARNING: MQIIH must be fullword aligned
- If no MQIIH is presented to the bridge (MQMD.Format=MQFMT_IMS_VAR_STRING) (“MQIMSVS”) default values are assumed:
  - LTERM=TpipeName
  - MODNAME=MQMD.Format
    - Default is “MQIMSVS”
  - MQMD.Format is used as the MFSMapName
  - Non-conversational
  - Commit-then-send (commit mode 0)
  - Security mode is MQISS_CHECK (ACEE only in CR)
- Data is passed to IMS in the OTMA prefix

Telcordia Technologies
Performance from Experience

MQSeries IMS Bridge

- MQSeries creates two TPIPE control blocks per local queue defined as using the IMS bridge
  - One for “asynchronous” messages
    - Commit mode 0 - commit-then-send
    - Output is “asynchronous” to transaction completion
    - Default if no IIH
    - This is a SYNChronized Tpipe
  - One for “synchronous” messages
    - Commit mode 1 - send-then-commit
    - Output is “synchronous” with transaction completion
    - This is a non-SYNChronized Tpipe
  - See MQSeries Support Pack MP14 for TPIPE performance tuning
- The TPIPE control blocks in IMS are created when the first message for the type (sync, async) arrives on the queue

Telcordia Technologies
Performance from Experience

MQSeries IMS Bridge

- There are special requirements for Commit Mode 0 input messages to IMS
  - If the IMS transaction is defined as RECOVER then the MQSeries message must be persistent
  - If the IMS transaction is NORECOV then the MQSeries message must be non-persistent
  - If this is not the case IMS will reject the message with sense code 00230000

Telcordia Technologies
Performance from Experience

MQSeries IMS Bridge

- Commit Mode 0 input messages (continued)
  - How does IMS know whether a message is persistent or nonpersistent?
    - All CM0 messages arrive on SYNChronized TPIPES
    - There are no flags in the OTMA headers for this
  - The answer is that IMS and MQSeries “cheat”
    - If the message is persistent it is sent on the SYNChronized TPIPE with a valid sequence number
    - If the message is non-persistent it is sent on the SYNChronized TPIPE with the sequence number set to zero

Telcordia Technologies
Performance from Experience

MQSeries IMS Bridge

- The data stream passed to MQSeries by the application MQPUT is in LLZZTrancodeDataLLZZData... format
  - This allows for multi-segment input messages via OTMA to IMS
- The sending message is always treated by MQSeries as “in syncpoint”
  - If the message is “not in syncpoint” it is treated as “in syncpoint” followed by an immediate commit

Telcordia Technologies
Performance from Experience

MQSeries IMS Bridge

- If the input message cannot be put to the IMS queue because:
  - Invalid message - input message goes on DLQ and warning sent to system console
  - IMS rejected message (sense 001A) - DFS message from IMS is put into MQSeries reply message and put on reply-to queue
    - If reply message cannot be PUT, it is placed on the DLQ
    - Input message now goes to DLQ (PQ05963)
  - IMS rejected message (message error) - input message goes on DLQ and warning sent to system console
  - IMS rejected message (other) - messages go back to their original queue, the bridge is stopped, and warning sent to system console
- If any return messages cannot be PUT to the DLQ, messages go back to their original queue, the bridge is stopped, and warning sent to system console

Telcordia Technologies
Performance from Experience

MQSeries IMS Bridge

- IMS Output Messages
  - Reply messages are placed on the MQSeries reply-to-queue specified in the original message header
  - All IMS message segments are assembled into a single MQSeries message which must not exceed the MAXMSGLEN parameter of the queue manager
  - If PUT to reply-to queue fails, the message goes on the DLQ
  - If PUT to DLQ fails, a NAK is sent to IMS and the message remains in IMS

Telcordia Technologies
Performance from Experience

MQSeries IMS Bridge

- Building the MQMD in the OTMA user data for asynchronous OTMA output messages in the DFSYDRU0 exit is not straightforward
- The format is LL | | MQMD | | Reply-to-Format
- The LL must be exactly 2 (LL) + L'MQMD + 8 (Reply-to-Format) or MQSeries will ignore the entire MQMD and use default values
- The MQSeries sample exit in Appendix B of the *MQSeries for OS/390 System Management Guide* is very useful
  - Follow this code VERY carefully

Telcordia Technologies
Performance from Experience

MQSeries IMS Bridge

- What you are building is actually an INPUT MQMD as if the message had come from MQSeries originally
  - The Replyto Queue and Replyto Queue Manager are the message destination
  - The MSGID and CORRELID are the INPUT MSGID and CORRELID - not what will be presented to the MQSeries application
    - 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

Telcordia Technologies
Performance from Experience

MQSeries IMS Bridge

- Setting the format name for MQSeries 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 OTMA User Data is valid and the message was input with an MQIIH IMS prefix and the Reply-to-Format is not blanks

Telcordia Technologies
Performance from Experience

MQSeries IMS Bridge

- But - there was no input message – how do you tell MQSeries 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 MQSeries the “input message” had an MQIIH
- Then set the Reply-to-Format name in the OTMA User Data to “MQIMSVS”

Telcordia Technologies
Performance from Experience

MQSeries IMS Bridge

- If after all that you:
  - set the Reply-to-Format to blanks (MQFMT_NONE)
  - or if the MQMD_FORMAT is not MQFMT_IMS
  - or if there is not valid OTMA User Data
  - **THEN** field TMAMHMAP in the OTMA State Prefix is used as the MQSeries Format Name
- This is set by the MODNAME parameter in the IMS ISRT call, if present
  - CALL “CBLTDLI” USING ISRT ALTPCB MSGIO MODNAME

Telcordia Technologies
Performance from Experience

MQSeries IMS Bridge

- 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 the MQSeries Format name is set to MQFMT_NONE (blanks)

Telcordia Technologies
Performance from Experience

MQSeries IMS Bridge

- The MQSeries application that MQGET's the reply message must be able to process the LLZZ's
  - If the application is written in C you must "cast" the LL field as MQ_BYTE or very bad things will happen
- You can also consider using MQSeries channel exits to remove the LLZZ's and set the Format Name to MQFMT_NONE

Telcordia Technologies
Performance from Experience

MQSeries IMS Bridge

- Determining the MQSeries Persistence for output messages from OTMA to MQSeries is complicated and is affected by commit mode
- The determination takes place in two phases…

Telcordia Technologies
Performance from Experience

MQSeries IMS Bridge

- Persistence Determination – Phase 1
  - If the output is on a TPIPE created by MQSeries
    - 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 if there is a valid sequence number in the TMAMCRSQ field in the OTMA State Header
      - More later

Telcordia Technologies
Performance from Experience

MQSeries IMS Bridge

- Persistence Determination – Phase 1
  - If the output is on a TPIPE not created by MQSeries
    - If the commit mode is 1 and flag TMAMCASY in not set in the OTMA State Header then the output message is persistent
    - If the commit mode is 0 or the TMAMCASY flag is 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

Telcordia Technologies
Performance from Experience

MQSeries IMS Bridge

- Persistence Determination – Phase 1
  - You can control the presence of valid sequence numbers in the TMAMCRSQ field in the OTMA State Header for asynchronous output messages from IMS
  - Tell IMS to make the TPIPE SYNChronized when it is created
  - This can be set explicitly in the DFSYDRU0 exit
  - The OTMASP parameter in DFSPBxxx can be used to default newly created Tpipes to Synchronized
    - This parameter was created for MQSeries customers who wanted persistent output messages but did not want to write DFSYDRU0 exits

Telcordia Technologies
Performance from Experience

MQSeries IMS Bridge

- Persistence Determination – Phase 2
  - If there is a valid OTMA User Data Prefix
    - If the output is on a TPIPE created by MQSeries
      - If the message is Commit Mode 1
        - Then the output message persistence is the persistence of the input message as reflected in the MQMD_PERSISTENCE field in the OTMA User Data

Telcordia Technologies
Performance from Experience

IMS OTMA Security

- Requires RACF 1.9.2 (or equivalent)
- Three levels of security
  1. Connections from MQSeries to IMS (during client bid)
     - Uses the RACF Facility class
     - IMS checks ***IMSXCF.XCFgroupname.MQ-member-name***
       - MQSeries must have read access to this class
     - MQSeries checks ***IMSXCF.XCFgroupname.IMS-member-name***
       - The MQSeries subsystem userid access to this class determines userid validation for each message that crosses the bridge

Telcordia Technologies
Performance from Experience

IMS OTMA Security

- Three levels of security (continued)
  2. Userid Validation in MQSeries
     - MQMD.UserIdentifier field is used
     - Based on MQSeries 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
       - 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, the userid is passed to IMS and can be used for normal IMS security

Telcordia Technologies
Performance from Experience

IMS OTMA Security

- 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
     - 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

IMS OTMA Security

- MQSeries security profiles
  - If ***subsysid.NO.SUBSYS.SECURITY*** is defined...
    - MQSeries 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
  - If ***subsysid.NO.SUBSYS.SECURITY*** is not defined...
    - Define the ***subsysid.NO.xxxx.CHECKS*** profiles
    - Give MQSeries Subsystem userid at least UPDATE access in the ***IMSXCF.XCFgroupname.IMS-member-name*** profile
    - Client bid will succeed and userids will be passed to the IMS transactions

Telcordia Technologies
Performance from Experience

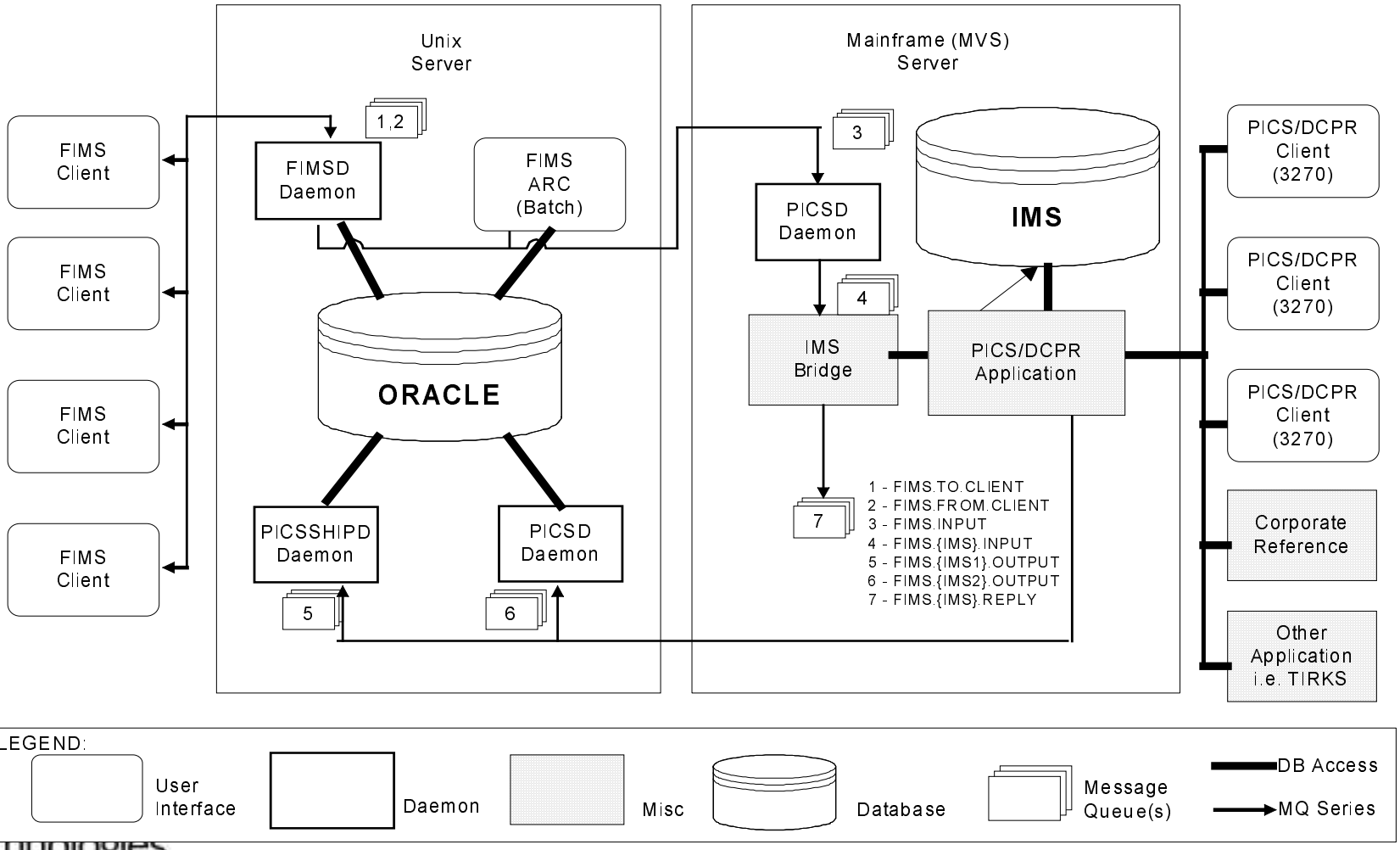
Telcordia Experience with the MQSeries IMS Bridge

- Two Telcordia applications are currently using the MQSeries IMS bridge in production
- A client had been sending data to one of our applications via tape
- Regulators dictated that the turnaround be reduced from 3 days to 3 hours
- The client already had experience using MQSeries and had a general purpose MQSeries client on a UNIX platform
- The UNIX client sends the messages to MQSeries on MVS which sends them to IMS
- Replies are sent back to the client asynchronously using the OTMA routing exits

Telcordia Technologies
Performance from Experience

Telcordia Experience with the MQSeries IMS Bridge

Telcordia FIMS Architecture (Production)



Tel Technologies

Performance from Experience

Information on the Web

- http://www.software.ibm.com/ts/mqseries/
- http://www.software.ibm.com/data/ims
- http://www.redbooks.ibm.com/

Telcordia Technologies
Performance from Experience

IMS OTMA Documentation

- IMS/ESA V6 OTMA Guide and Reference - SC26-8743
- IMS/ESA Application Programming : Transaction Manager - SC26-8729
- IMS/ESA Customization Guide - SC26-8732

Telcordia Technologies
Performance from Experience

MQSeries IMS Bridge Documentation

- MQSeries for MVS/ESA System Management Guide - SC33-0806
  - Chapter 2.3 - Customizing the MQSeries-IMS Bridge
  - Chapter 3.5 - The MQSeries-IMS Bridge
  - Chapter 3.6 - Controlling the MQSeries-IMS Bridge
  - Appendix B - Using OTMA exits in IMS
  - Appendix C - Mapping MQ messages to IMS transactions (will move to APG)
  - Appendix C - Resetting TPIPES (will move to Command Reference)
  - Appendix C - Messages and Codes (will move to Messages and Codes)

Telcordia Technologies
Performance from Experience

MQSeries IMS Bridge Documentation

- MQSeries Application Programming Guide - SC33-0807
  - Chapter 15 - Writing MQSeries-IMS bridge applications
- MQSeries Application Programming Reference - SC33-1673
  - Chapters 2 and 6 - MQIIH structure and constants
- SupportPac MP14: MQSeries for MVS/ESA - IMS Bridge Performance Report #2
  - www.software.ibm.com/ts/mqseries/txppacs/txpsumm.html

Telcordia Technologies
Performance from Experience

Questions?

Telcordia Technologies

Performance from Experience

RETURN TO INDEX