

E86

IMS Web-Access Application Considerations



Anaheim, California

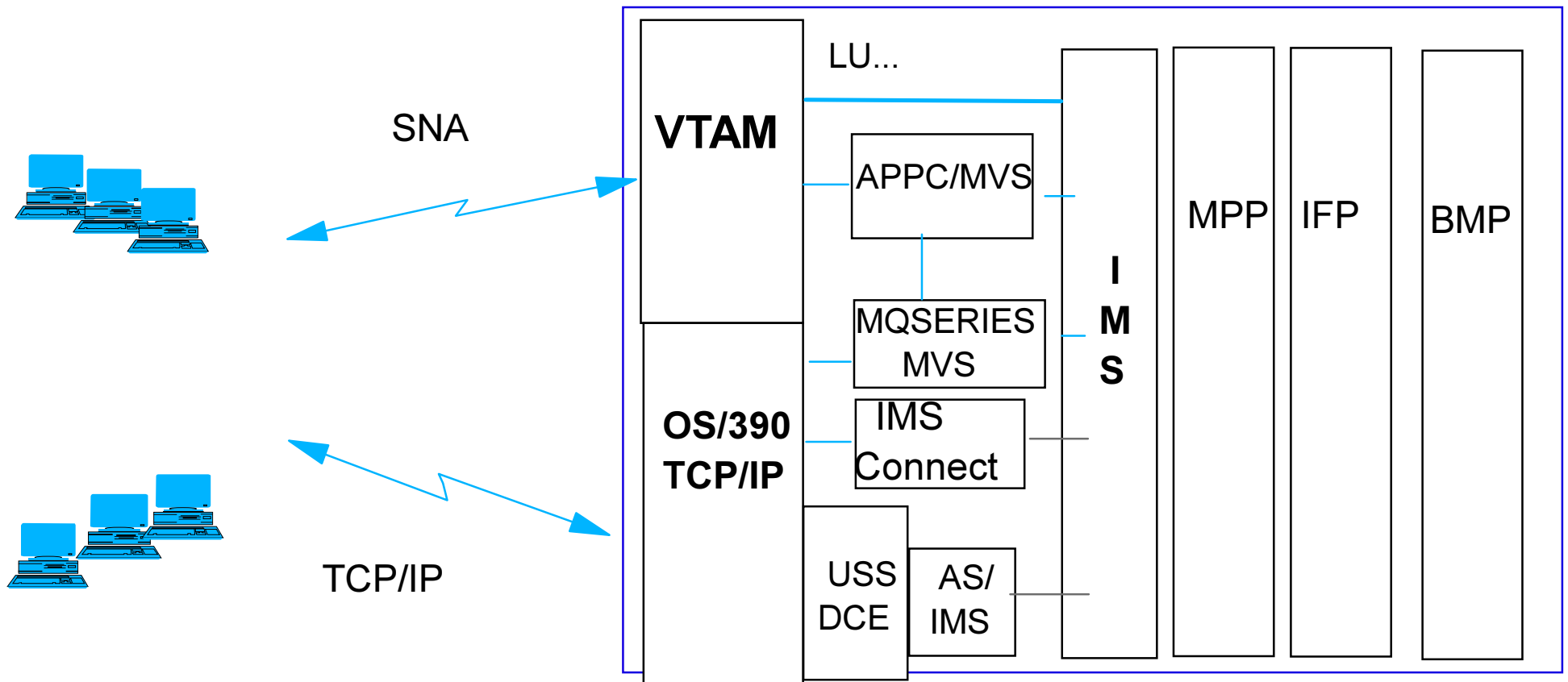
October 23 - 27, 2000

kblackm@us.ibm.com

Contents

- Connectivity
- Building an IMS e-business Application
- Existing IMS Application Program Considerations
- Summary

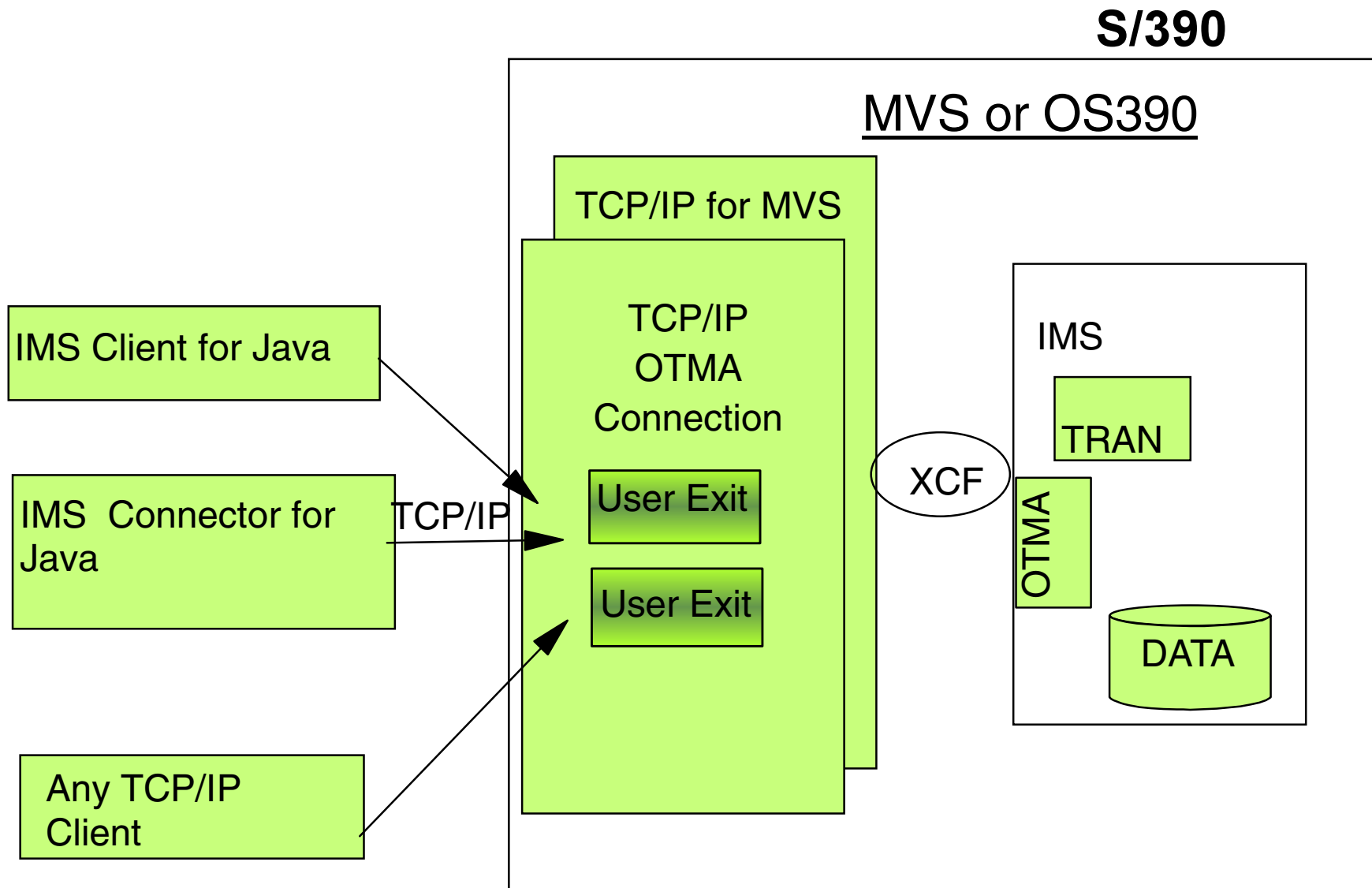
Access to IMS Application Programs



IMS e-business Connectors

- **IMS Connect**
 - TCP/IP
- **OTMA Callable Interface**
 - XCF
- **IMS Client for Java**
 - Applet
 - Application
- **IMS Connector for Java**
 - Application
 - Servlet
- **IMS Java Classes**
 - Application
 - Servlet
- **Enterprise Java Beans(EJB)**
 - Component Applications

IMS Connect Overview

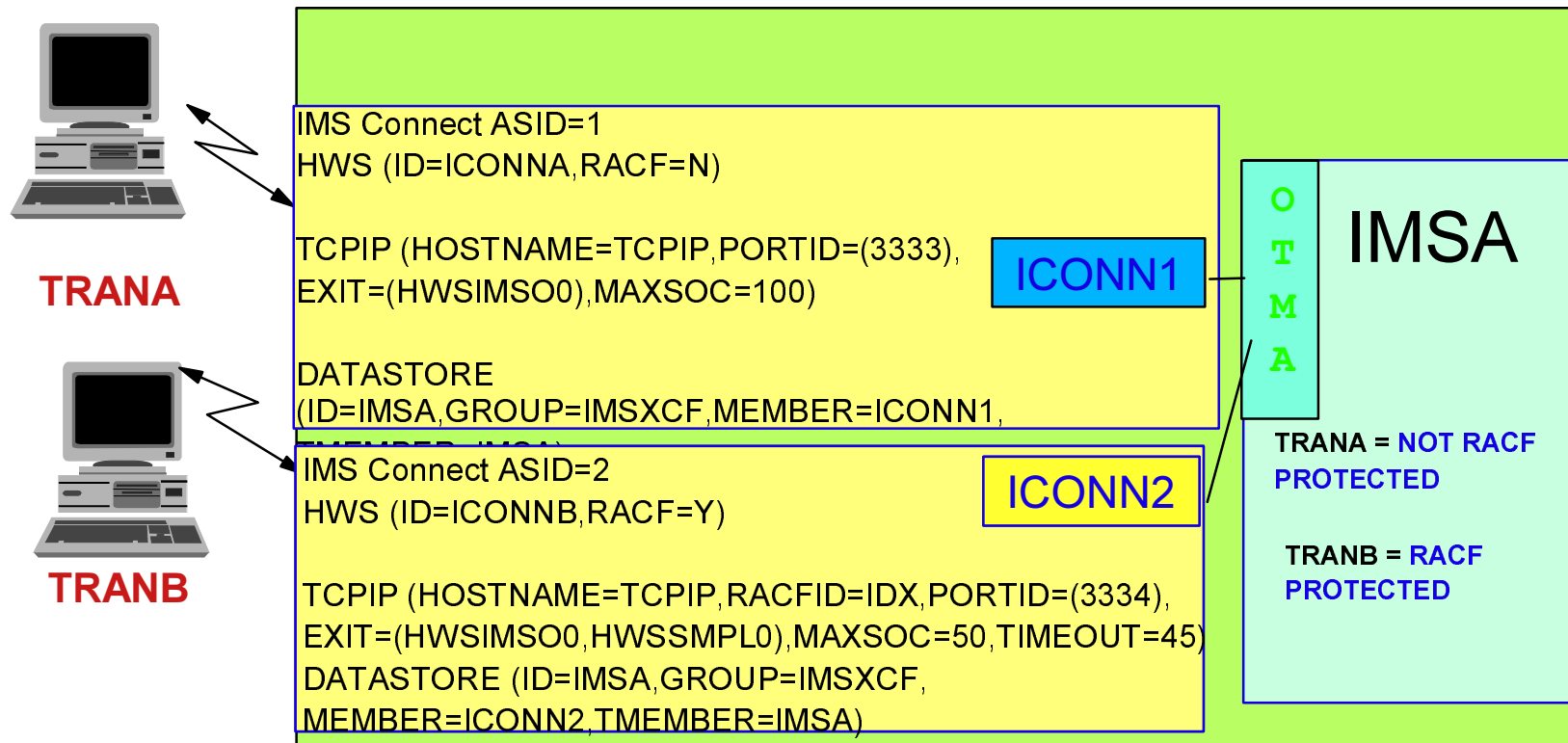


IMS Connect

Configuration Security

- Multiple IMS Connects Accessing Same IMS
 - Different IMS Connect Security Settings
 - Different Exits Included

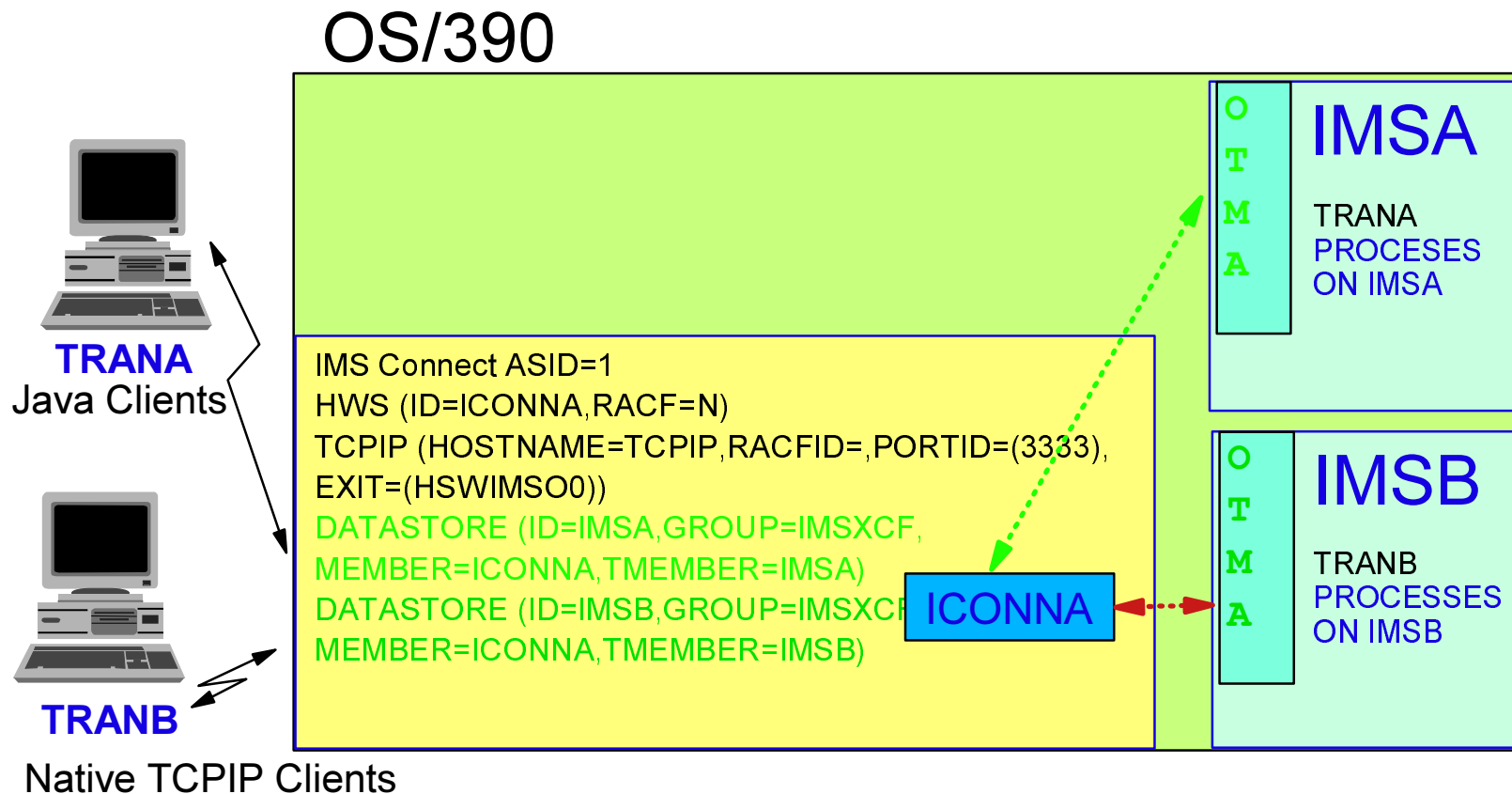
OS/390



IMS Connect

Configuration Generic Datastore

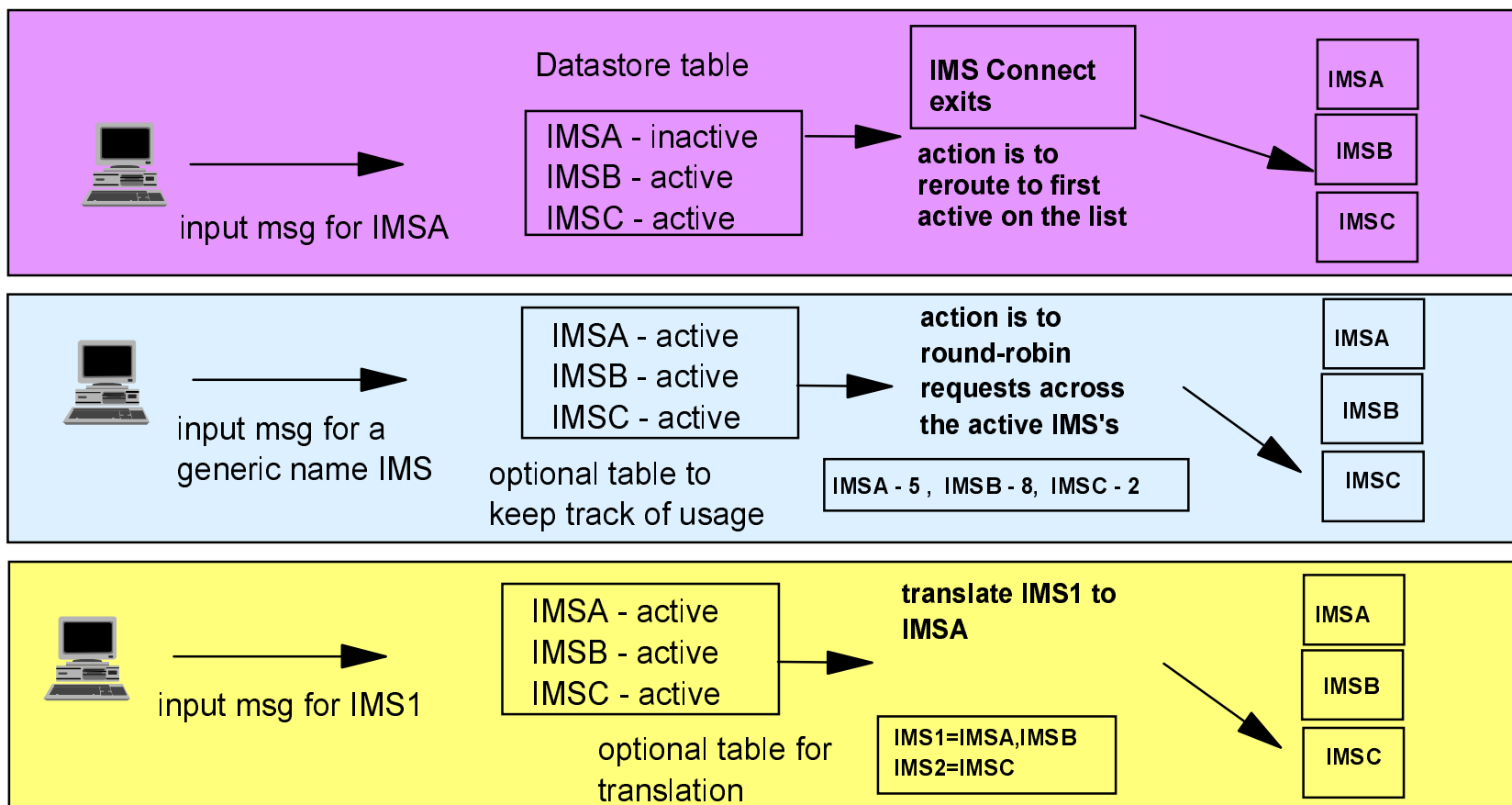
- One IMS Connect Accessing Different IMSs
 - DATASTORE identifies IMS
 - Different Exits Can be Selected



IMS Connect

Datastore Table

HWS (ID=ICONNA,RACF=Y)
 TCPIP (HOSTNAME=TCPIP,RACFID=IDX,PORTID=(3333),EXIT=(HWSIMSO0,HWSSMPL0)
 DATASTORE (ID=IMSA,GROUP=IMSXCF,MEMBER=ICONN1,TMEMBER=IMSA)
 DATASTORE (ID=IMSB,GROUP=IMSXCF,MEMBER=ICONN1,TMEMBER=IMSB)



IMS Connect

Troubleshooting

- **If you receive errors back in the remote client**
 - Review IMS Connect address space or OS syslog for associated error messages
- **Possible return codes received in the remote client**
 - OTMA - documented in IMS OTMA Guide & Reference
 - TCP/IP - documented in IMS TCP/IP Application Development Guide and Reference SC31-7186
- **HWSnnnnn Error Messages documented at:**
 - [IMS Connect Messages and Codes](#)

IMS Connect

Troubleshooting ..

- **Recorder Trace**

- Shows the message layout
 - Input message received from the client
 - Input message after the user exit / before IMS
 - Output reply from IMS before user exit
 - Output reply after is processed by the user exit but before it gets sent

- **Internal Trace**

- Provides information about the activity through the IMS Connect components
 - CMDT - Command component
 - HWSI - IMS Connect to Datastore component
 - HWSW - Server to IMS Connect component
 - TCPI - TCPI Component
 - OTMA - OTMA Component
 - ENVT - ITOC environment component

- **Formatted Dump**

IMS Connect

Asynchronous Output

- **Remote client environment**

- Retrieval of messages is a client application responsibility
- Client program must be written to:

- **RESUME TPIPE**

RECEIVE - receive first output msg

ACK - acknowledge receipt of first msg

RECEIVE - receive second output message

ACK

...

- **IRM header specifies:**

Client id: to define the TPIPE name

Persistent socket : to keep the connection active

Asynch request type:

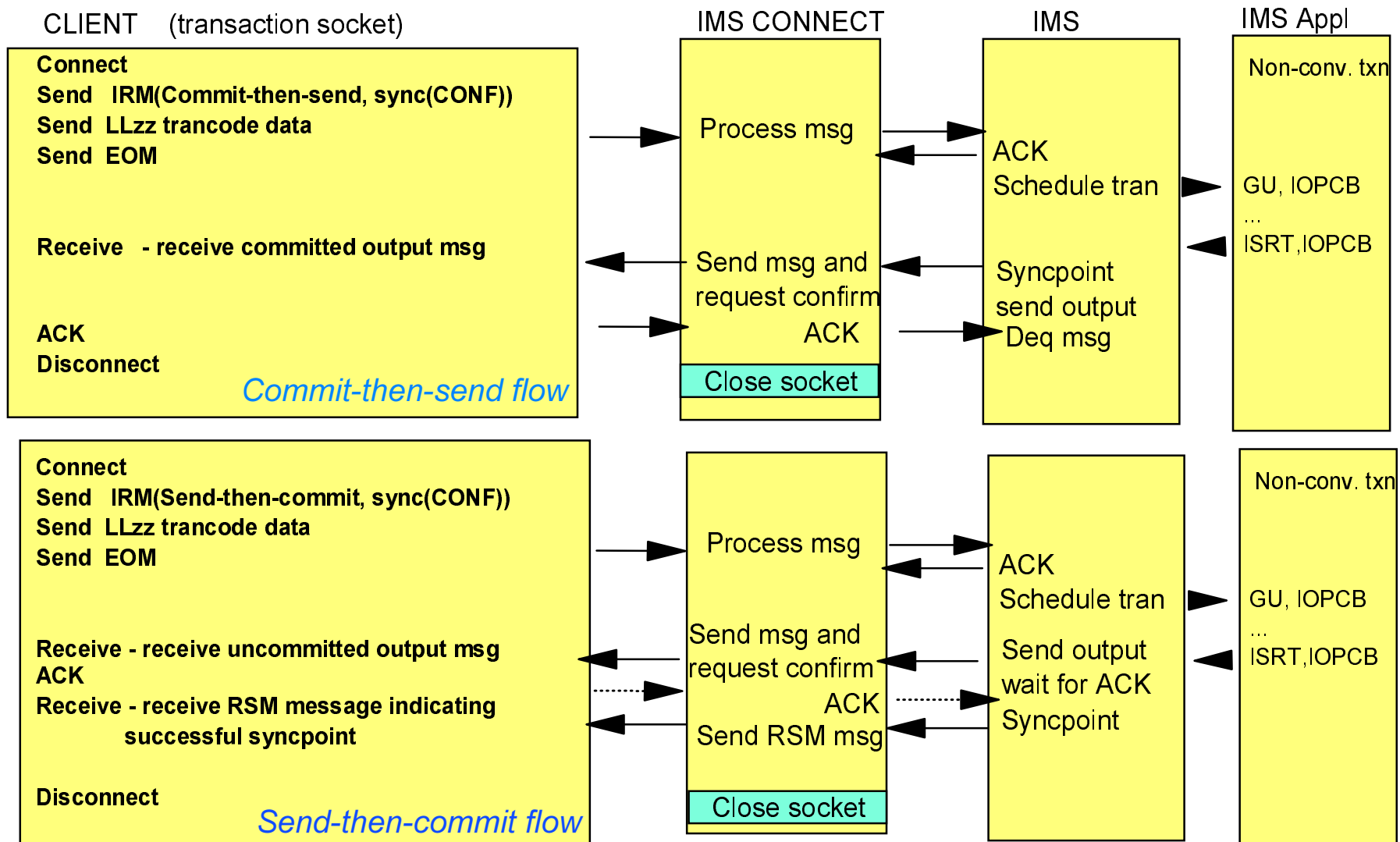
- ***Single*** - receive one msg and disconnect the socket

- ***No-Auto*** - receive all available msgs, wait a specified time
and disconnect if no more messages

- ***Auto*** - receive all available msgs, wait for next message
with no timeout

IMS Connect

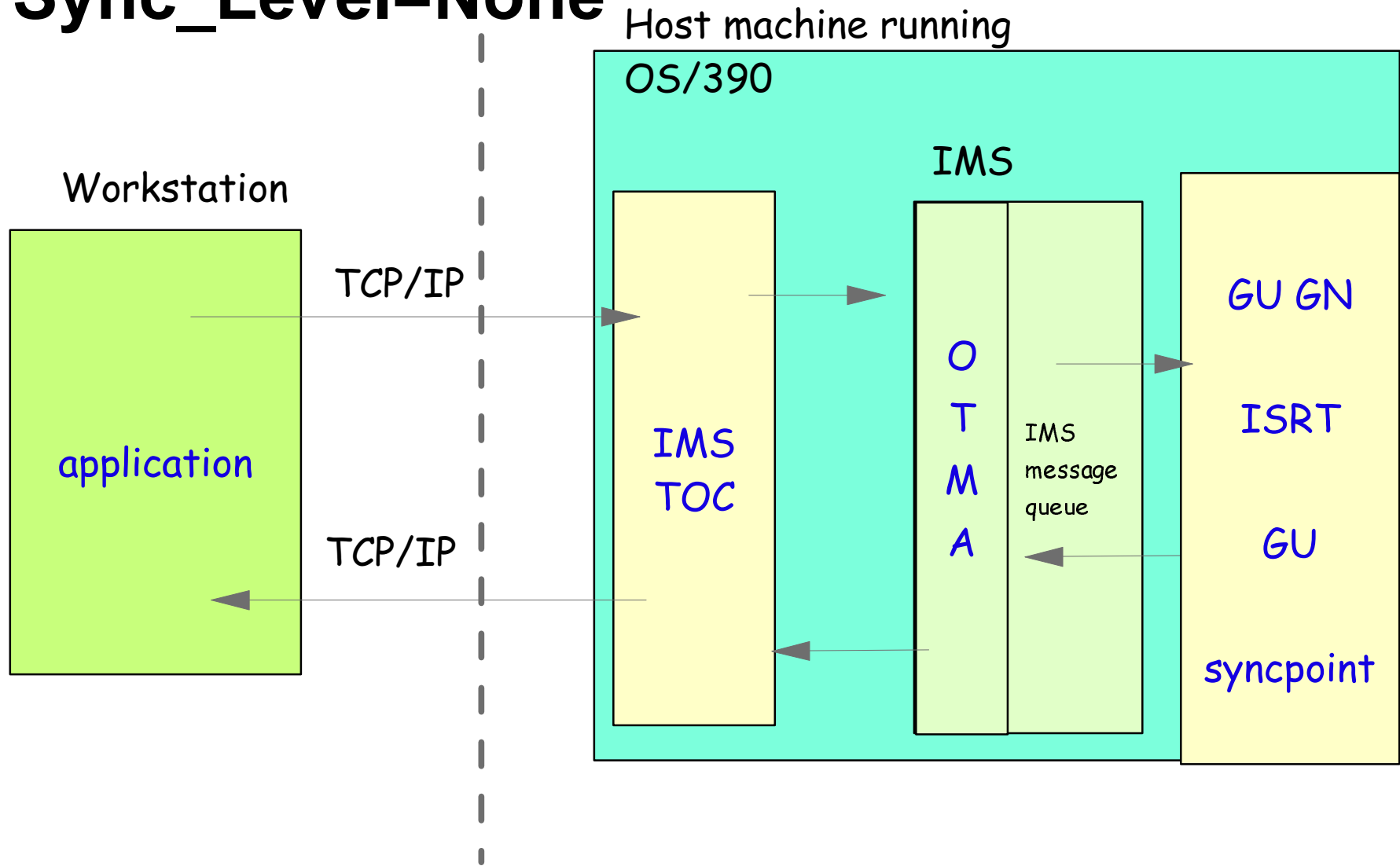
Sockets - Basic Flow



IMS Connect

Send-then-Commit

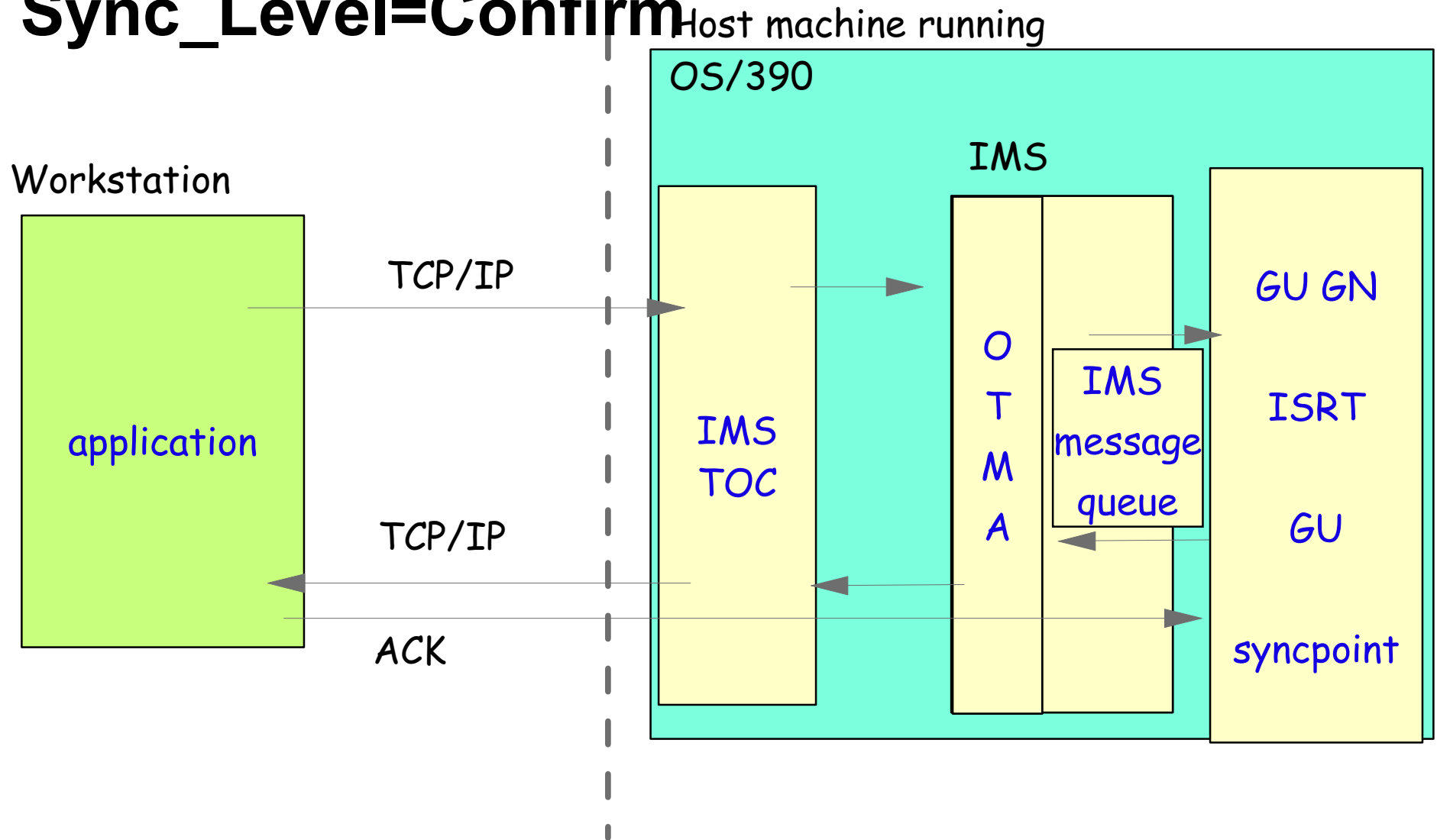
Sync_Level=None



IMS Connect

Send-then-Commit

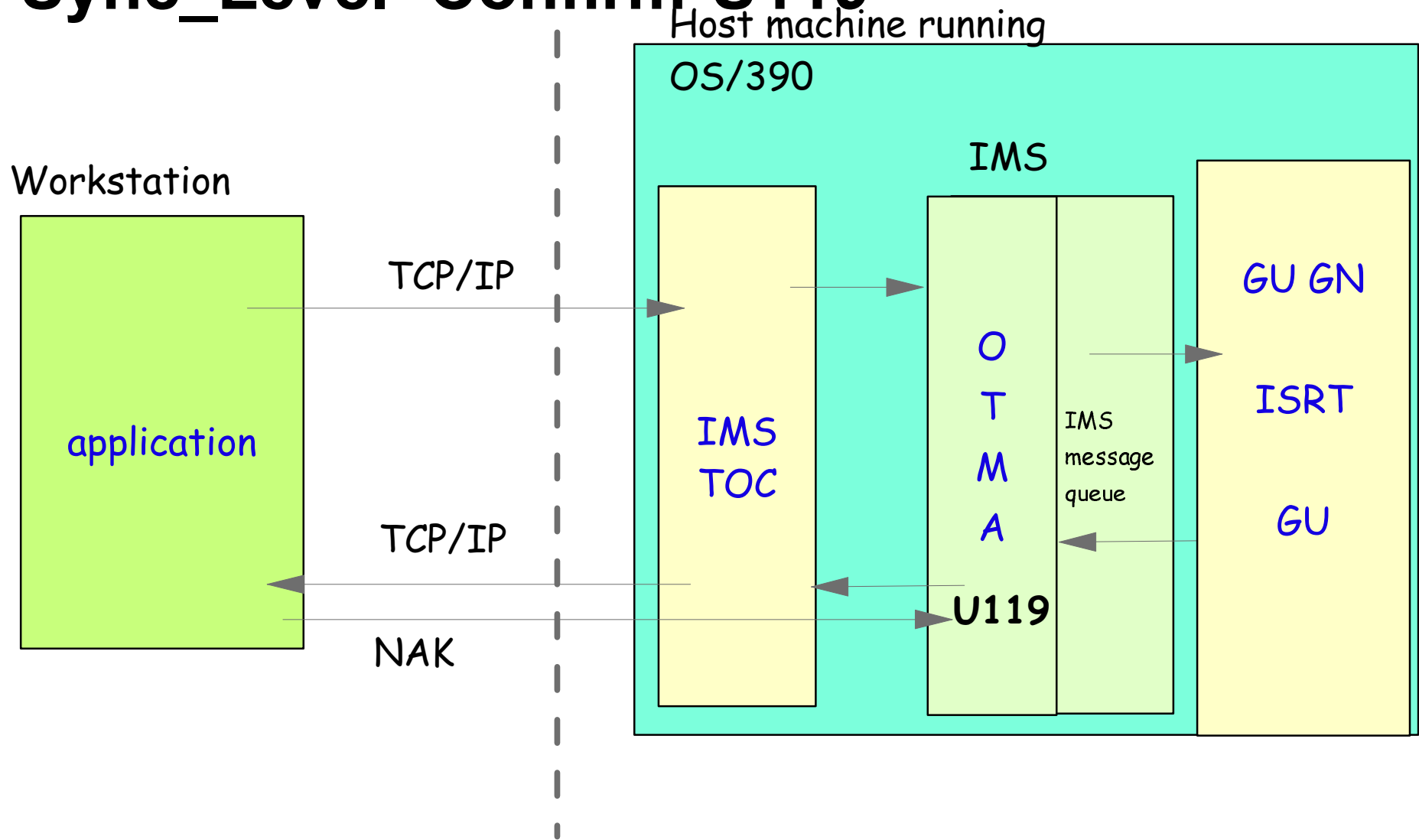
Sync_Level=Confirm



IMS Connect

Send-then-Commit

Sync_Level=Confirm-U119

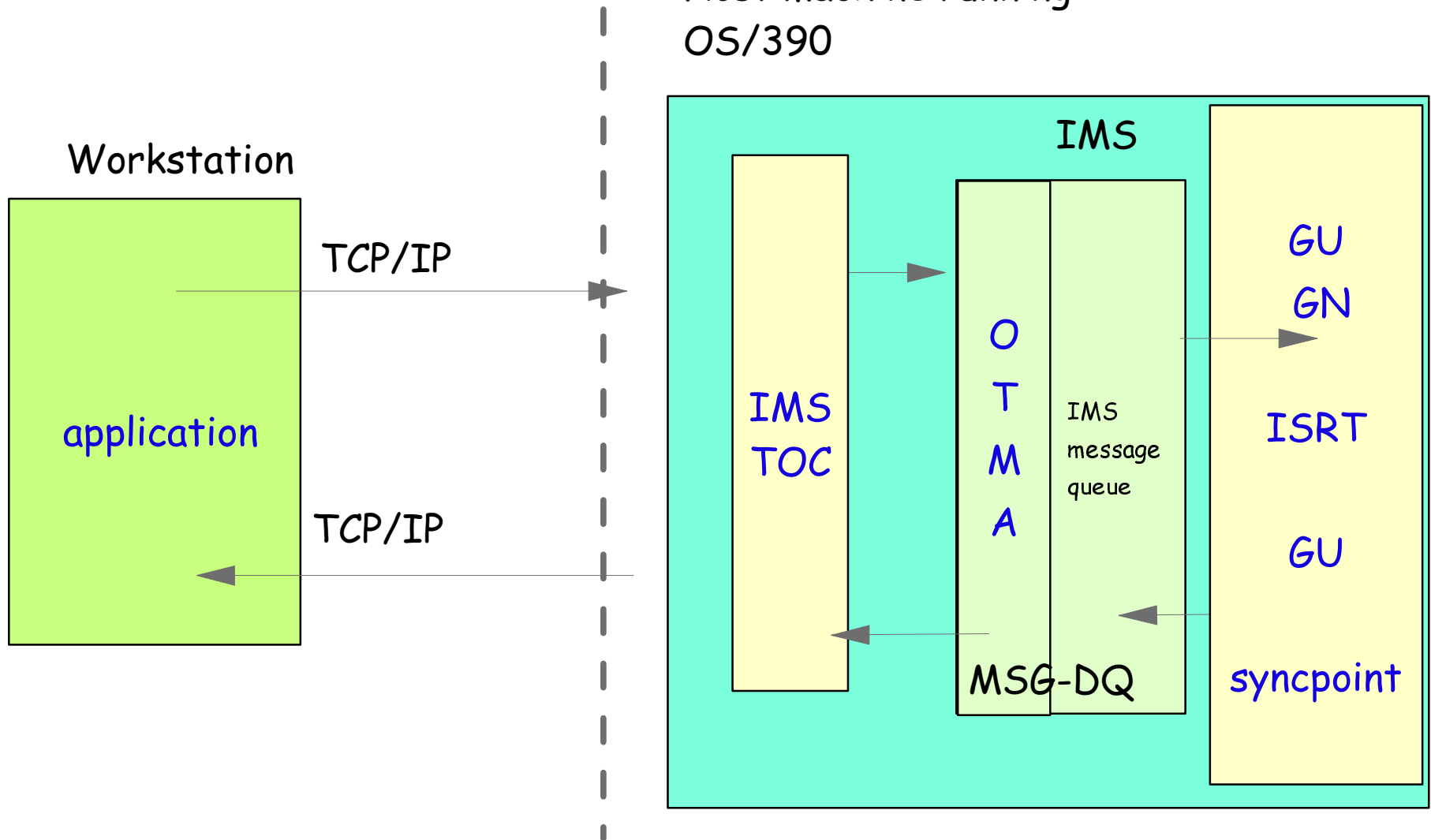


IMS Connect

Commit-then-Send

Sync_Level=None

Host machine running
OS/390



IMS Connect

Commit-then-Send

Sync_Level=Confirm

Host machine running OS/390

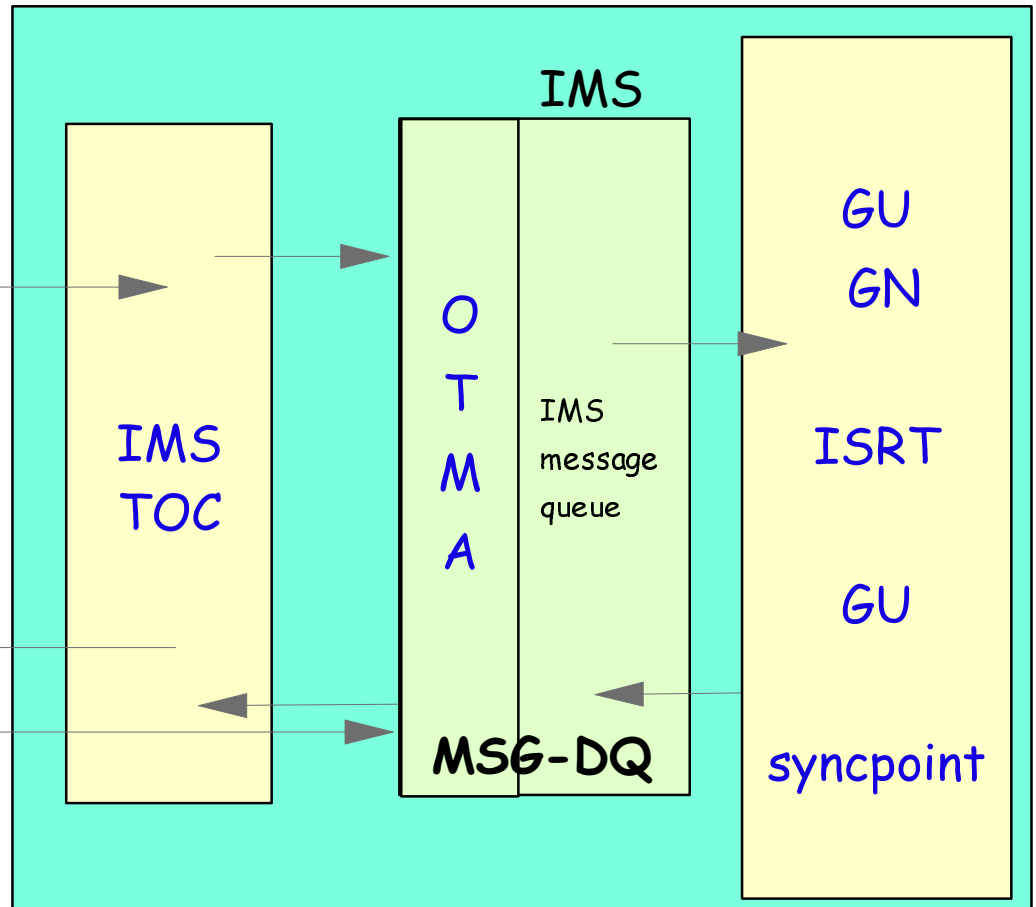
Workstation



TCP/IP

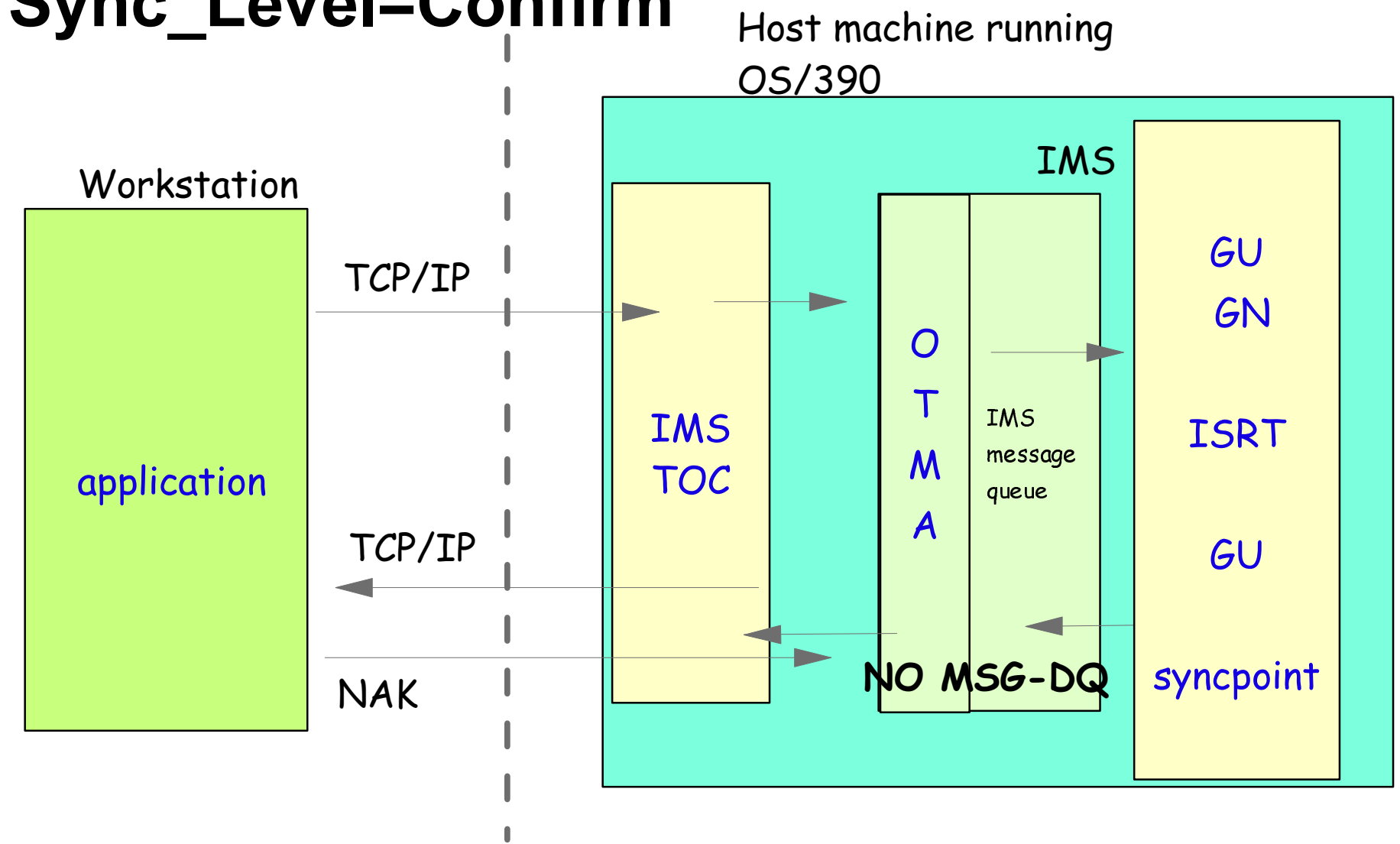
TCP/IP

ACK



IMS Connect

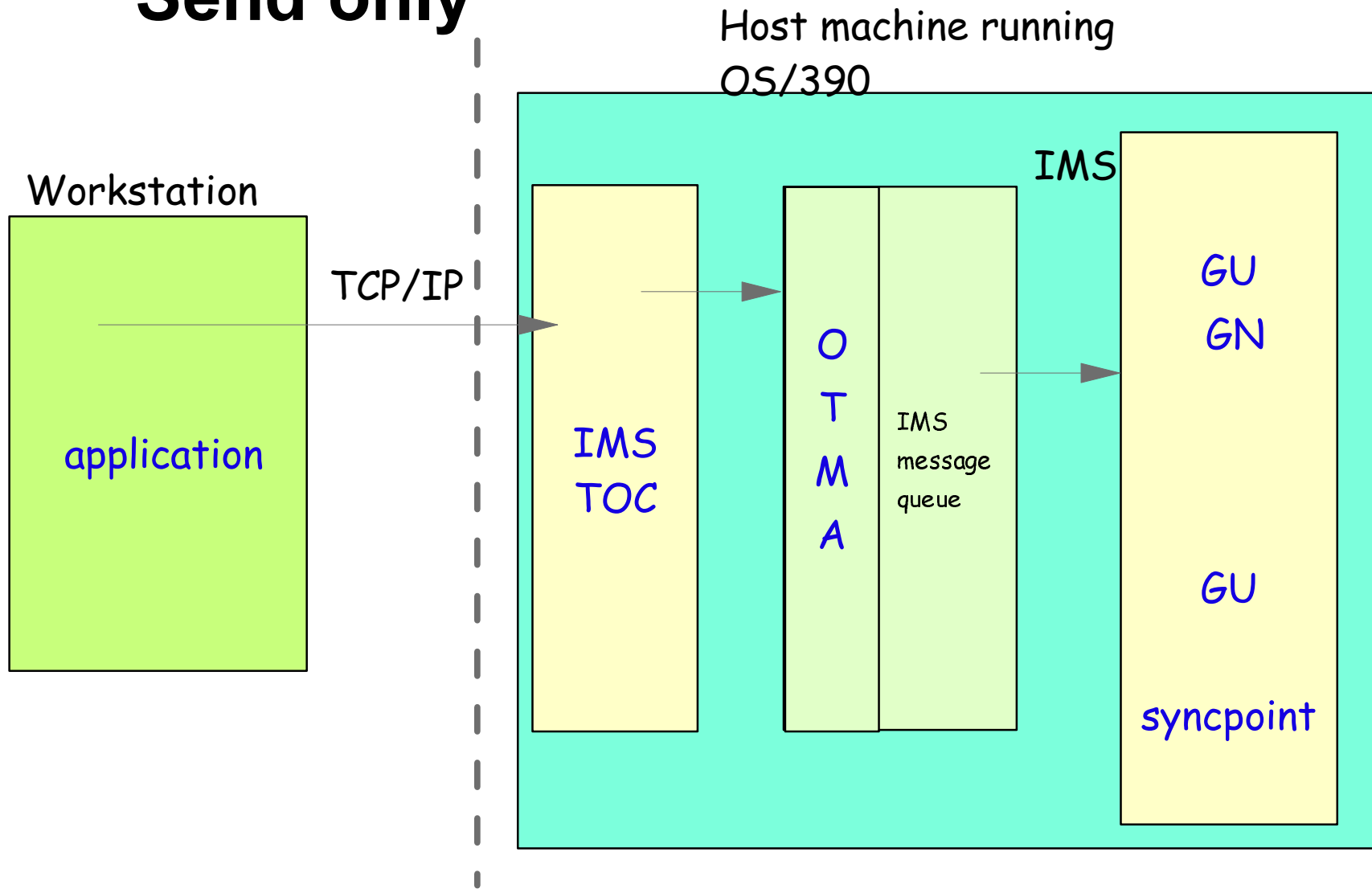
Commit-then-Send Sync_Level=Confirm



IMS Connect

Commit-then-Send

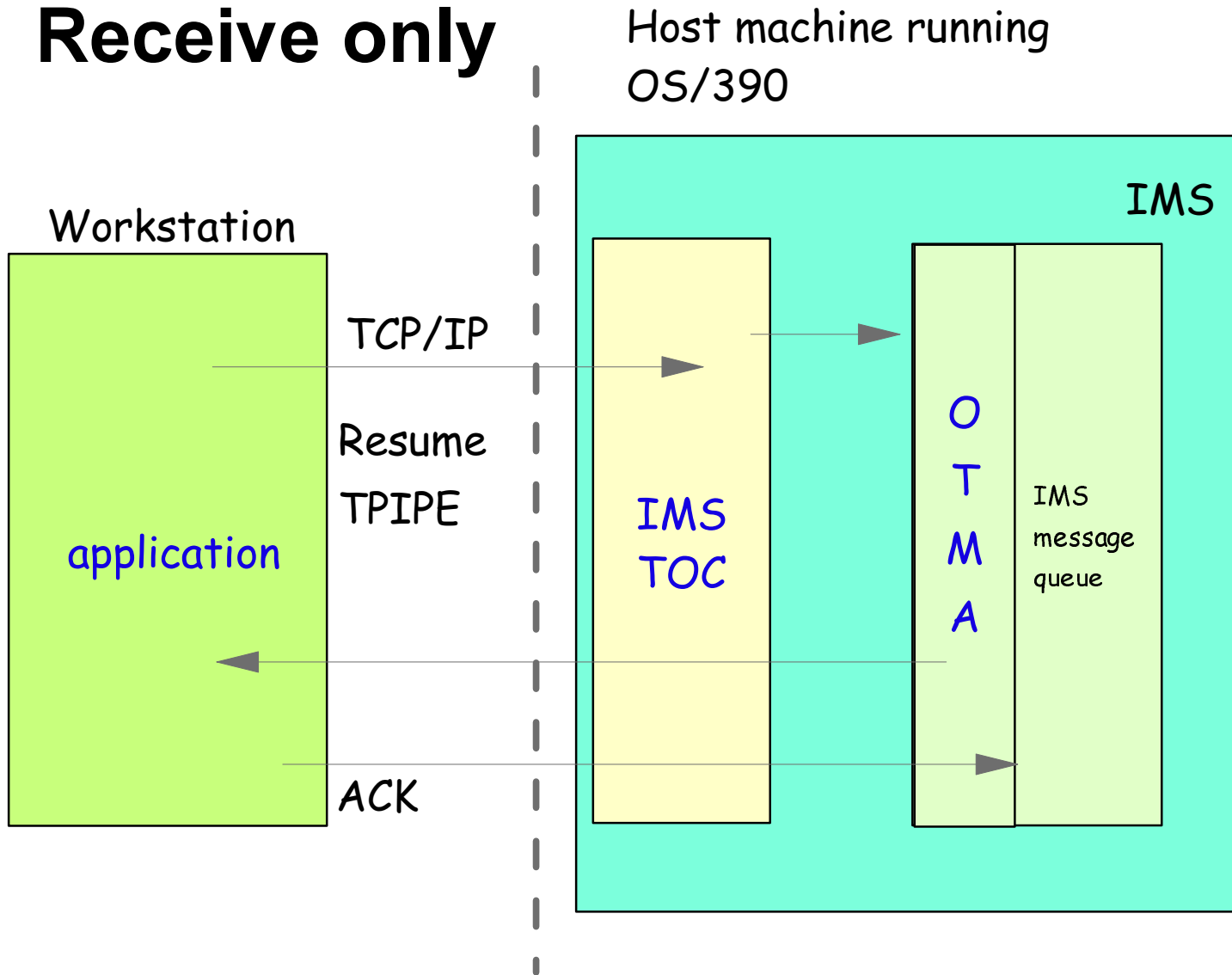
Send only



IMS Connect

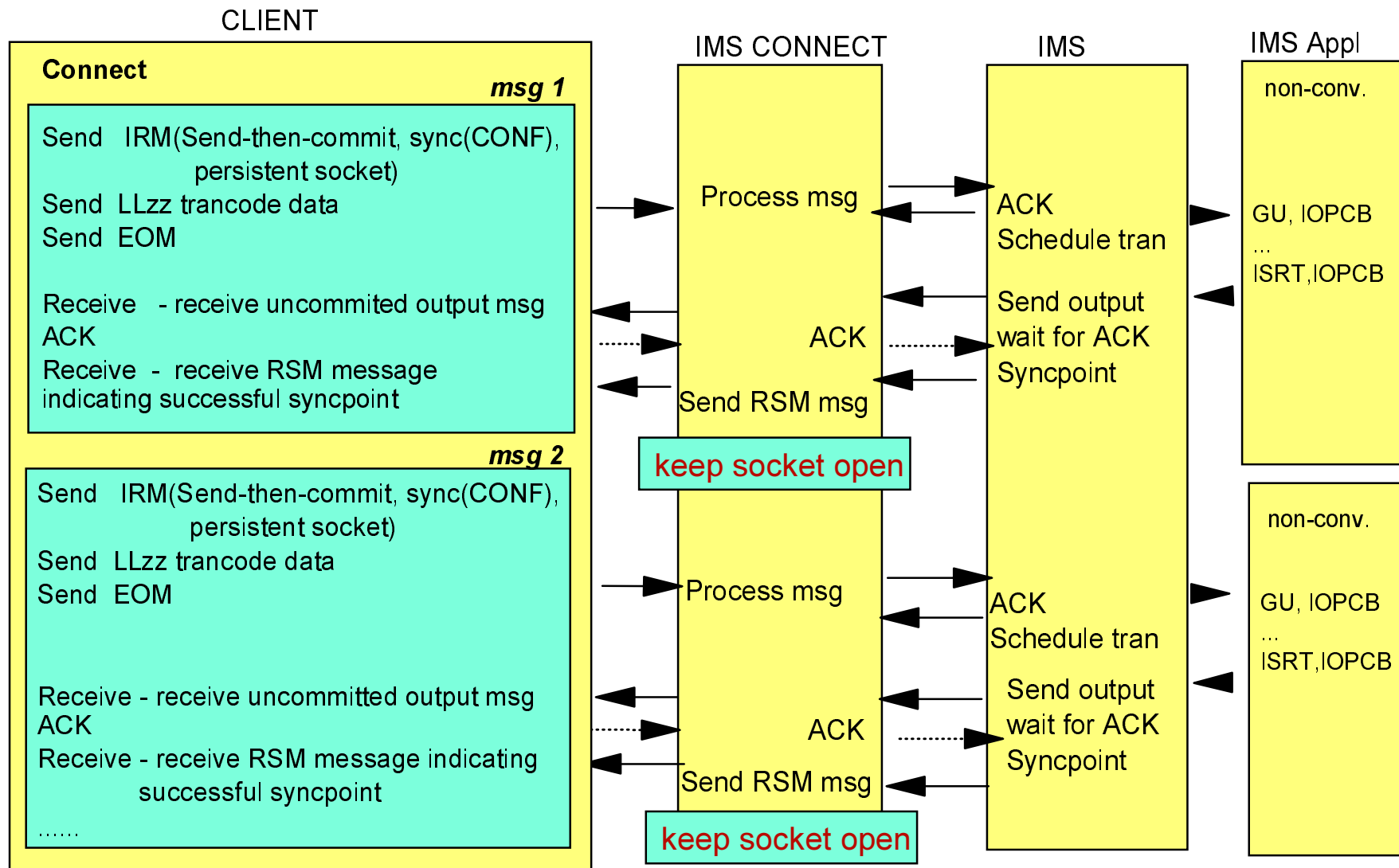
Commit-then-Send

Receive only



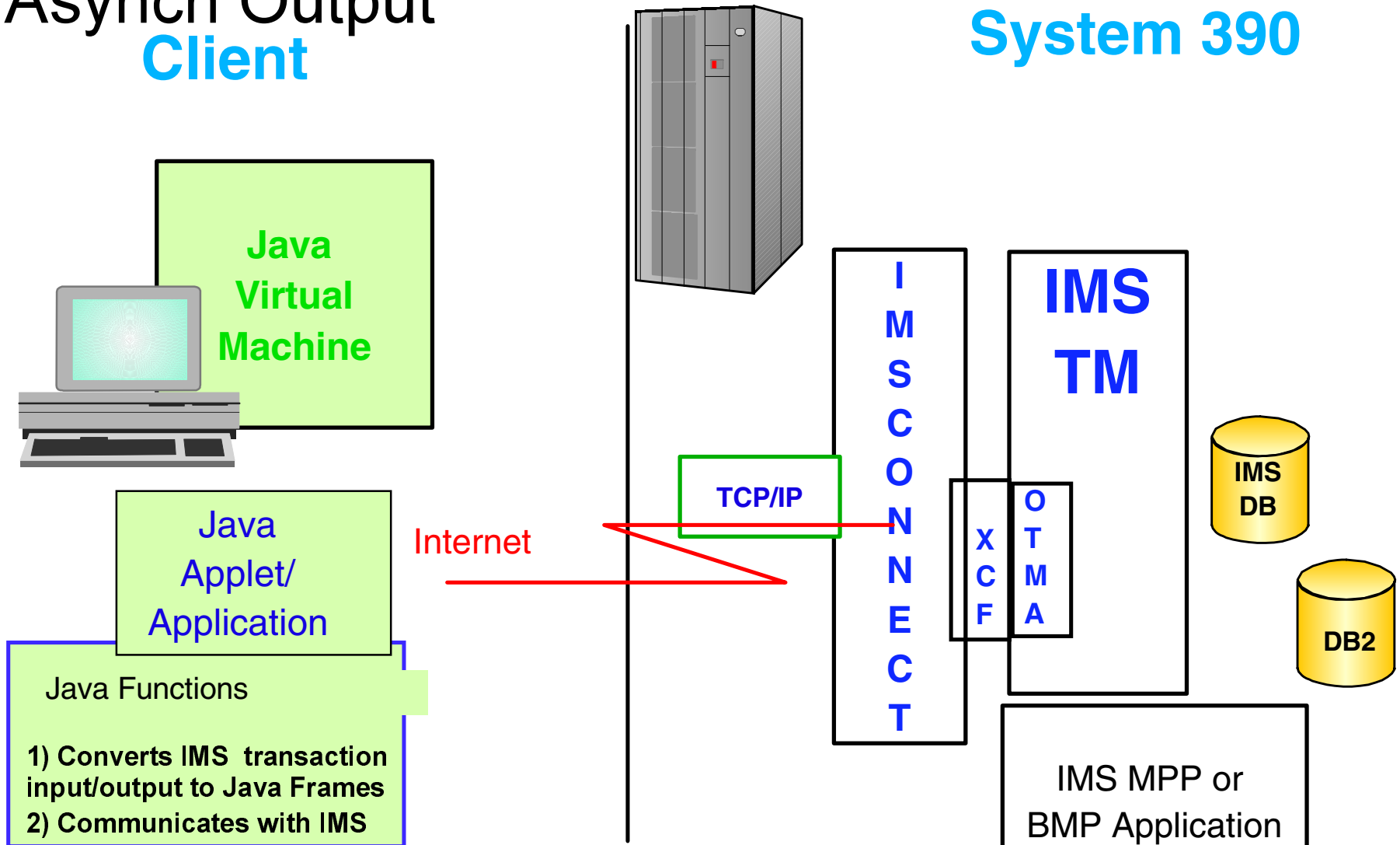
IMS Connect

Persistent Sockets



IMS Client for Java

Note: Can be used to retrieve
Asynch Output
Client



IMS Client for Java...

Transaction Data

INPUT DATA

HostName: Port: Port 34xx

Transaction: Port: Port 34xx

Client ID: DataStore ID:

RACFID: GROUP:

Sync Level: Commit Mode:

Password:

Input Text:

OUTPUT DATA

TRAN: RC =

MOD: RS =

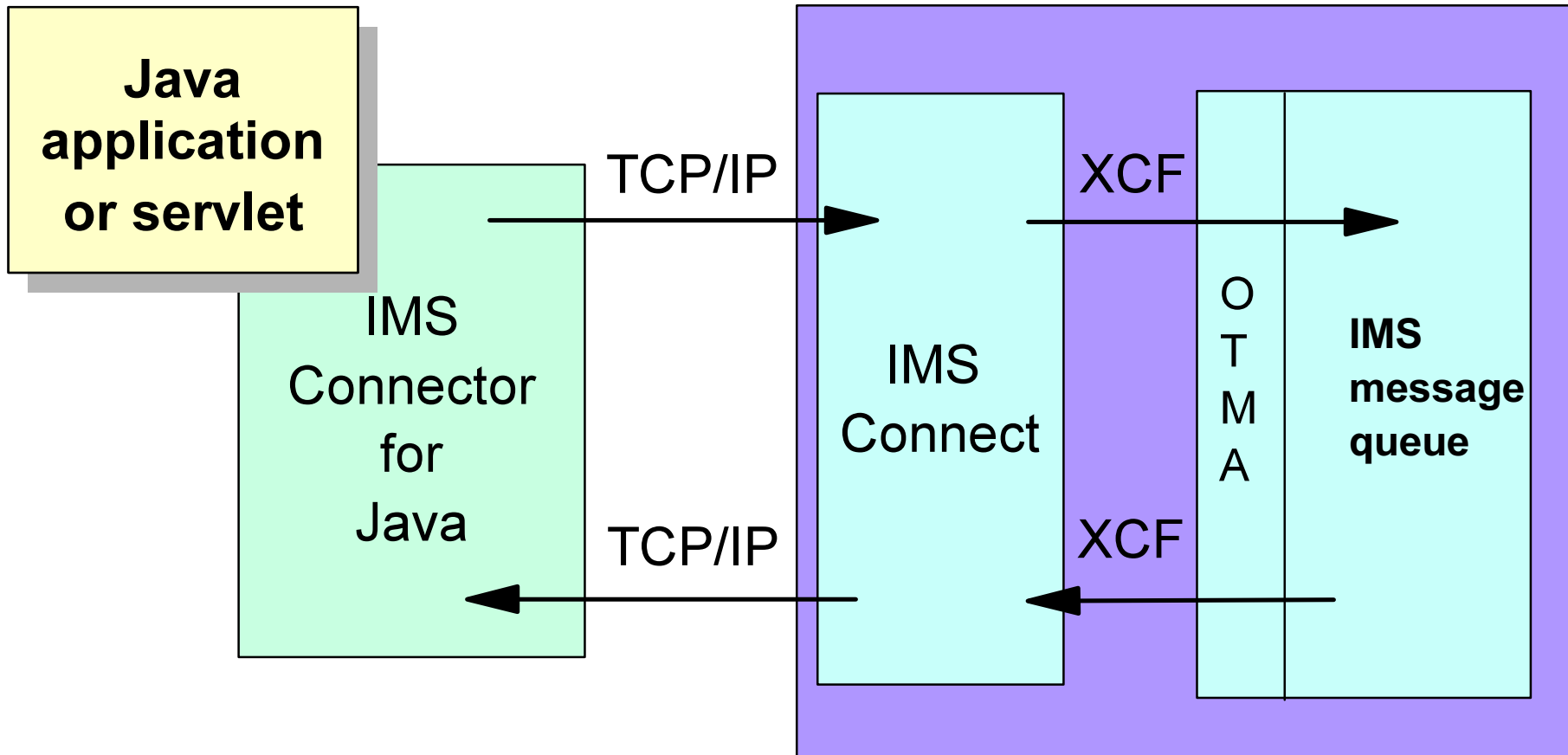
OUTPUT TEXT:

IMS Connector for Java

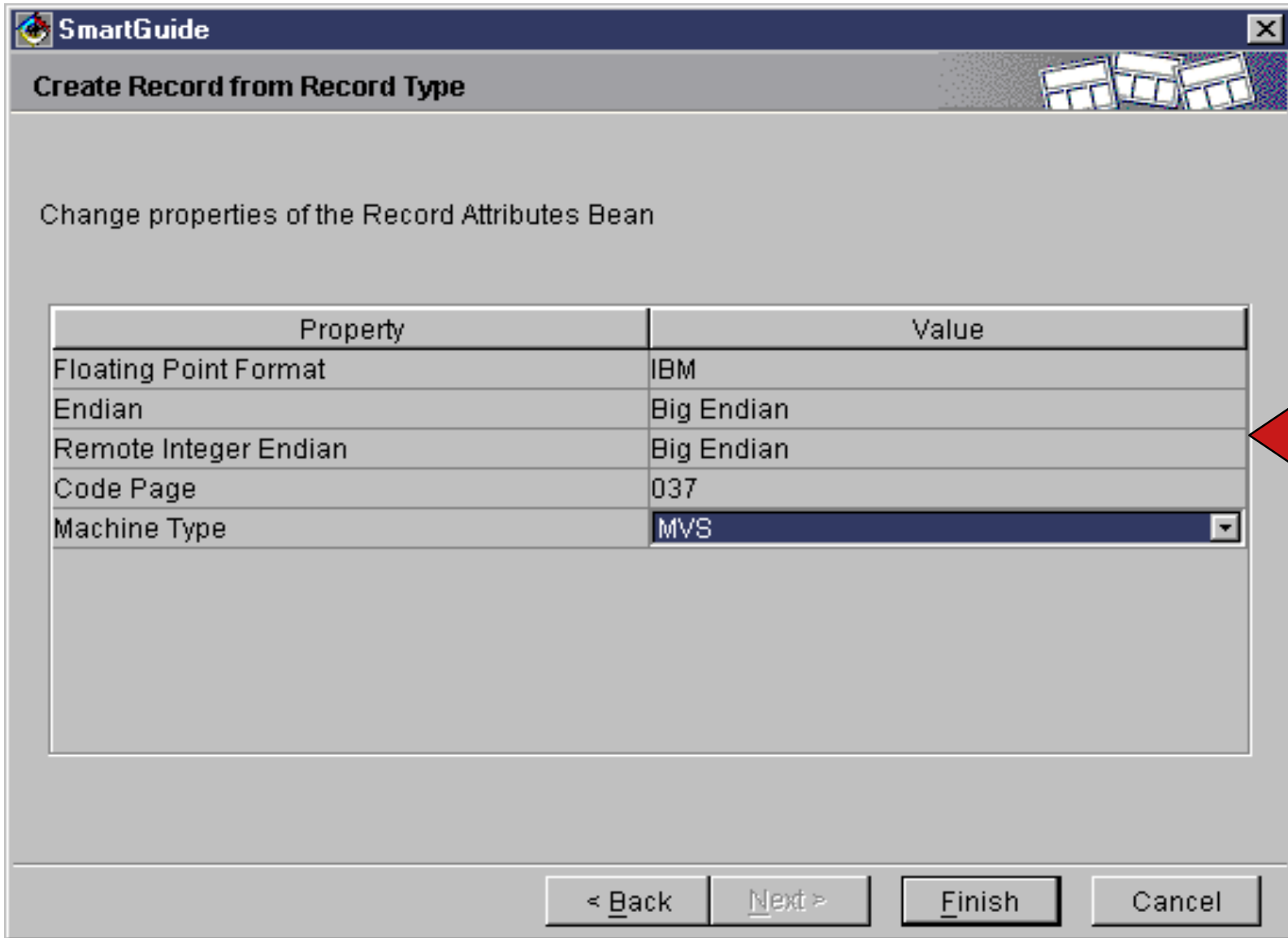
Send-Then-Commit

None or Confirm

Persistent Sockets



IMS Connector for Java



SmartGuide

Create Record from Record Type

Change properties of the Record Attributes Bean

Property	Value
Floating Point Format	IBM
Endianness	Big Endian
Remote Integer Endian	Big Endian
Code Page	037
Machine Type	MVS

< Back Next > Finish Cancel

Example BIG Endian= 0801

Little Endian= 8010

IMS Connector for Java

- **Visual Age For Java COBOL parser**
- **COBOL layout of the input or output message**
 - Note PL/1 uses LLLLZZTRANCODE only use LLZZTRANCODE for COBOL parser

01 INPUT-MSG.

02 IN-LL PICTURE S9(3) COMP.

02 IN-ZZ PICTURE S9(3) COMP.

02 IN-TRCD PICTURE X(10).

02 IN-CMD PICTURE X(8).

02 IN-STATIONID PICTURE X(10).

IMS Connector for Java

Synchronization Level and Mode

- **SYNCH_LEVEL_NONE**

- Default
- Command with `MODE_SEND_RECEIVE`
- Transaction is committed when output message is sent

- **SYNCH_LEVEL_CONFIRM**

- Command with `MODE_SEND_RECEIVE`
- DB changes are not committed when output is sent
- Application must send positive or negative acknowledgment, either:
 - EAB command with `MODE_ACK` or `MODE_NOACK`
 - `Coordinator.commit` or `rollback` call

IMS Connector for Java

Use generic IMS datastore name

MAPNAME(MOD Name IMS application requirement)

LTERM security checking

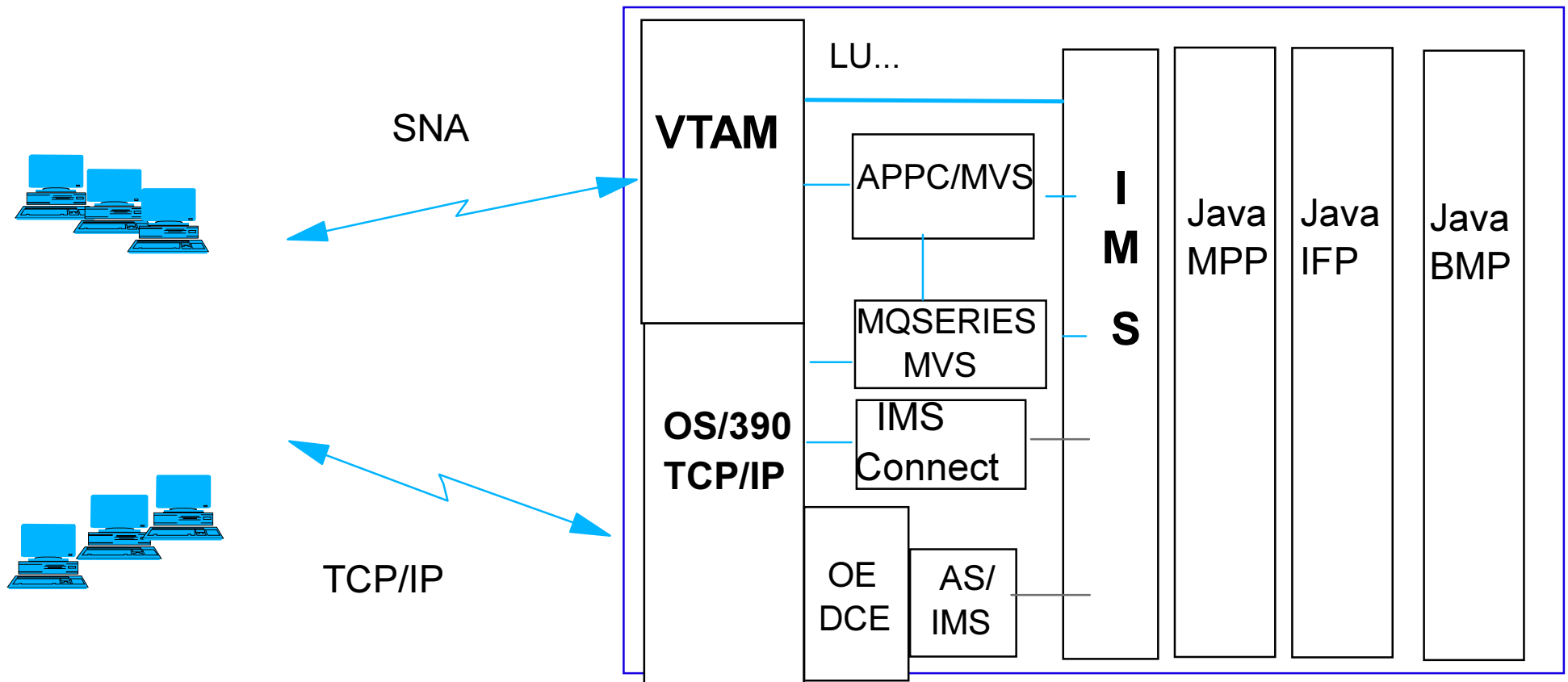
Output message length modified by COBOL program

PFK usage

V3.5 supports IMS Conversational Transactions

IMS Java Application Programs

Supports access Send-then-Commit or Commit-then-Send
None or Confirm

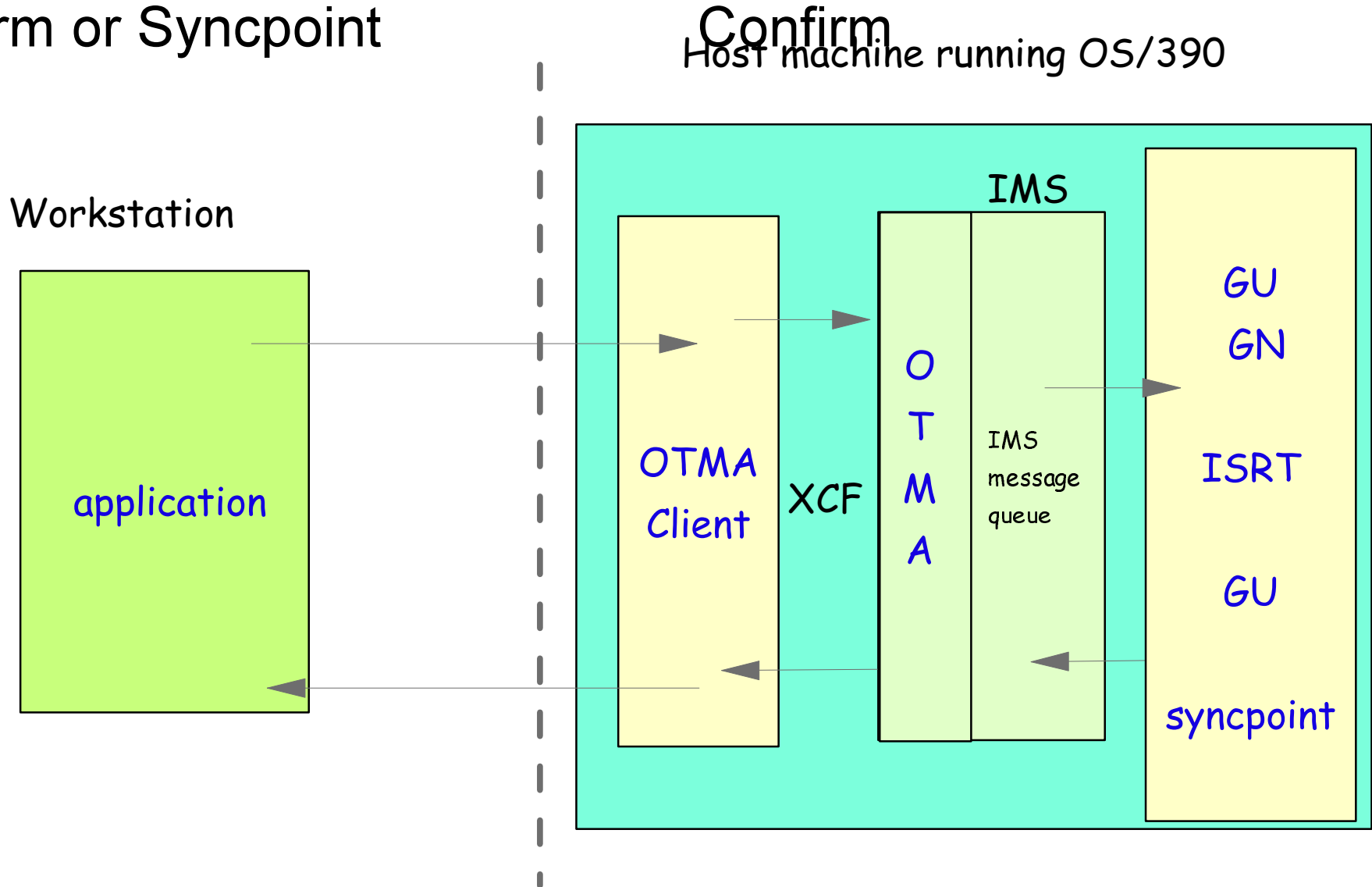


IMS Java Application Model

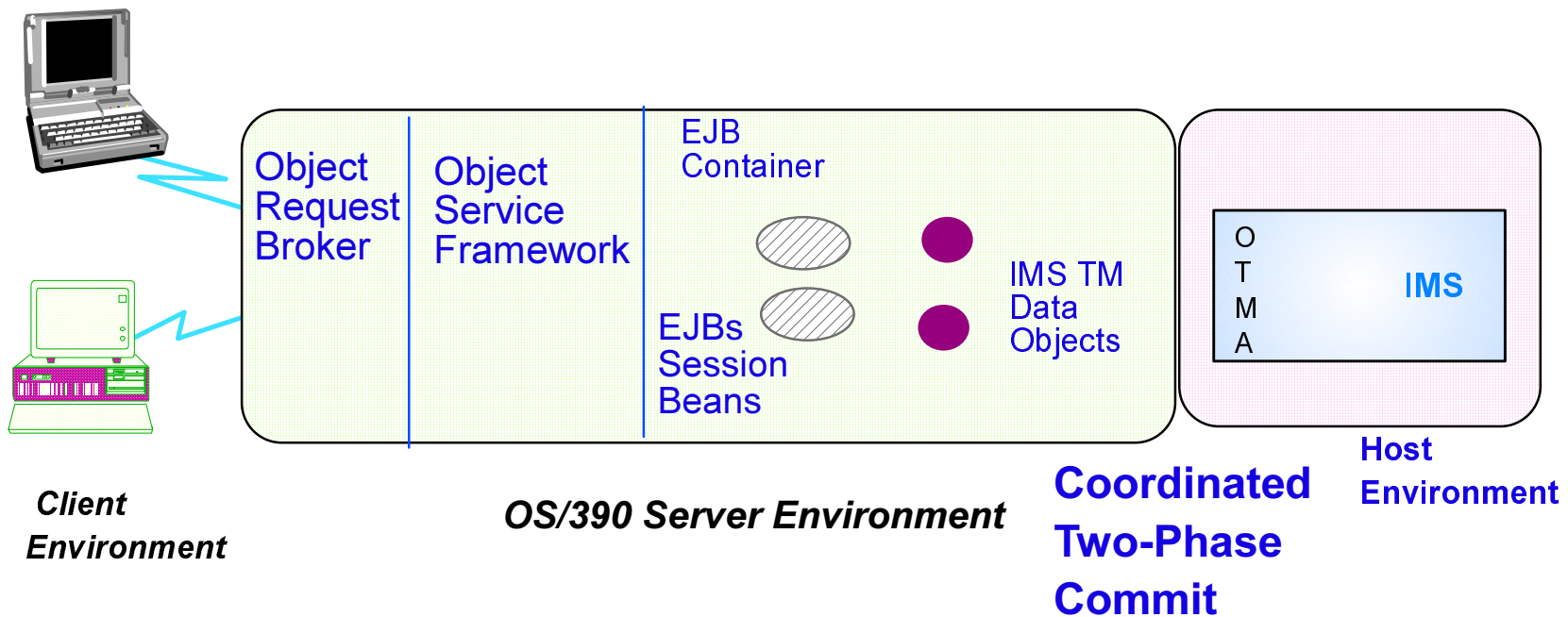
- Define classes for input and output message segments
- Define classes to access IMS DB segments
- Subclass **IMSApplication** and implement **doBegin** method
- Use **IMSMessageQueue** to receive and send messages
- Use JDBC to access IMS DB
- Use JDBC to access DB2
- Use **IMSTransaction** to commit or rollback resources
 - ▶ else get U118 Abend

OTMA Callable Interface

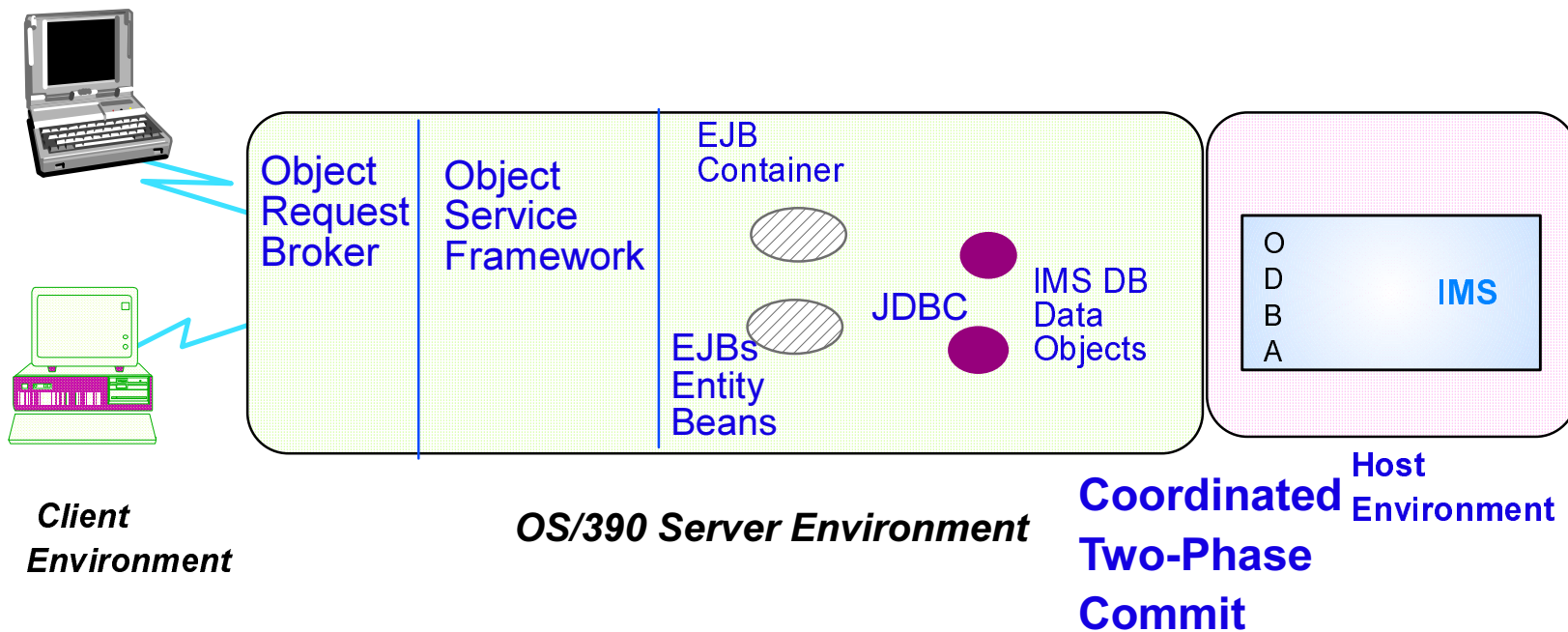
Supports Send-then-Commit or Commit-then-Send
Confirm or Syncpoint



OS/390 WebSphere Enterprise Edition



OS/390 WebSphere Enterprise Edition- Future IMS Java function



Summary

Commit Mode options

- Send-then-Commit

- Commit-then-Send

Sync Level options

- None

- Confirm

- Syncpoint