

A19

# The World of Java from an IMS perspective

Ken Blackman

kblackm@us.ibm.com



Miami Beach, FL

October 22-25, 2001

# Trademarks

The following are trademarks of the International Business Machines Corporation in the United States and/or other countries.

AIX*	IBM logo*	RACF
CICS*	IMS	WebSphere*
DB2*	Language Environment	VisualAge*
e-business logo*	OS/390*	z/OS
ESCON*	S/390*	zSeries
IBM*	Parallel Sysplex*	

\* Registered trademarks of IBM Corporation

The following are trademarks or registered trademarks of other companies.

Lotus, Notes, and Domino are trademarks or registered trademarks of Lotus Development Corporation

LINUX is a registered trademark of Linus Torvalds

Penguin (Tux) complements of Larry Ewing

Tivoli is a trademark of Tivoli Systems Inc.

Java and all Java-related trademarks and logos are trademarks of Sun Microsystems, Inc., in the United States and other countries

UNIX is a registered trademark of The Open Group in the United States and other countries.

Microsoft, Windows and Windows NT are registered trademarks of Microsoft Corporation.

SET and Secure Electronic Transaction are trademarks owned by SET Secure Electronic Transaction LLC.

\* All other products may be trademarks or registered trademarks of their respective companies.

## Notes:

Performance is in Internal Throughput Rate (ITR) ratio based on measurements and projections using standard IBM benchmarks in a controlled environment. The actual throughput that any user will experience will vary depending upon considerations such as the amount of multiprogramming in the user's job stream, the I/O configuration, the storage configuration, and the workload processed. Therefore, no assurance can be given that an individual user will achieve throughput improvements equivalent to the performance ratios stated here.

IBM hardware products are manufactured from new parts, or new and serviceable used parts. Regardless, our warranty terms apply.

All customer examples cited or described in this presentation are presented as illustrations of the manner in which some customers have used IBM products and the results they may have achieved. Actual environmental costs and performance characteristics will vary depending on individual customer configurations and conditions.

This publication was produced in the United States. IBM may not offer the products, services or features discussed in this document in other countries, and the information may be subject to change without notice. Consult your local IBM business contact for information on the product or services available in your area.

IBM considers a product "Year 2000 ready" if the product, when used in accordance with its associated documentation, is capable of correctly processing, providing and/or receiving date data within and between the 20th and 21st centuries, provided that all products (for example, hardware, software and firmware) used with the product properly exchange accurate date data with it. Any statements concerning the Year 2000 readiness of any IBM products contained in this presentation are Year 2000 Readiness Disclosures, subject to the Year 2000 Information and Readiness Disclosure Act of 1998.

All statements regarding IBM's future direction and intent are subject to change or withdrawal without notice, and represent goals and objectives only.

Information about non-IBM products is obtained from the manufacturers of those products or their published announcements. IBM has not tested those products and cannot confirm the performance, compatibility, or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

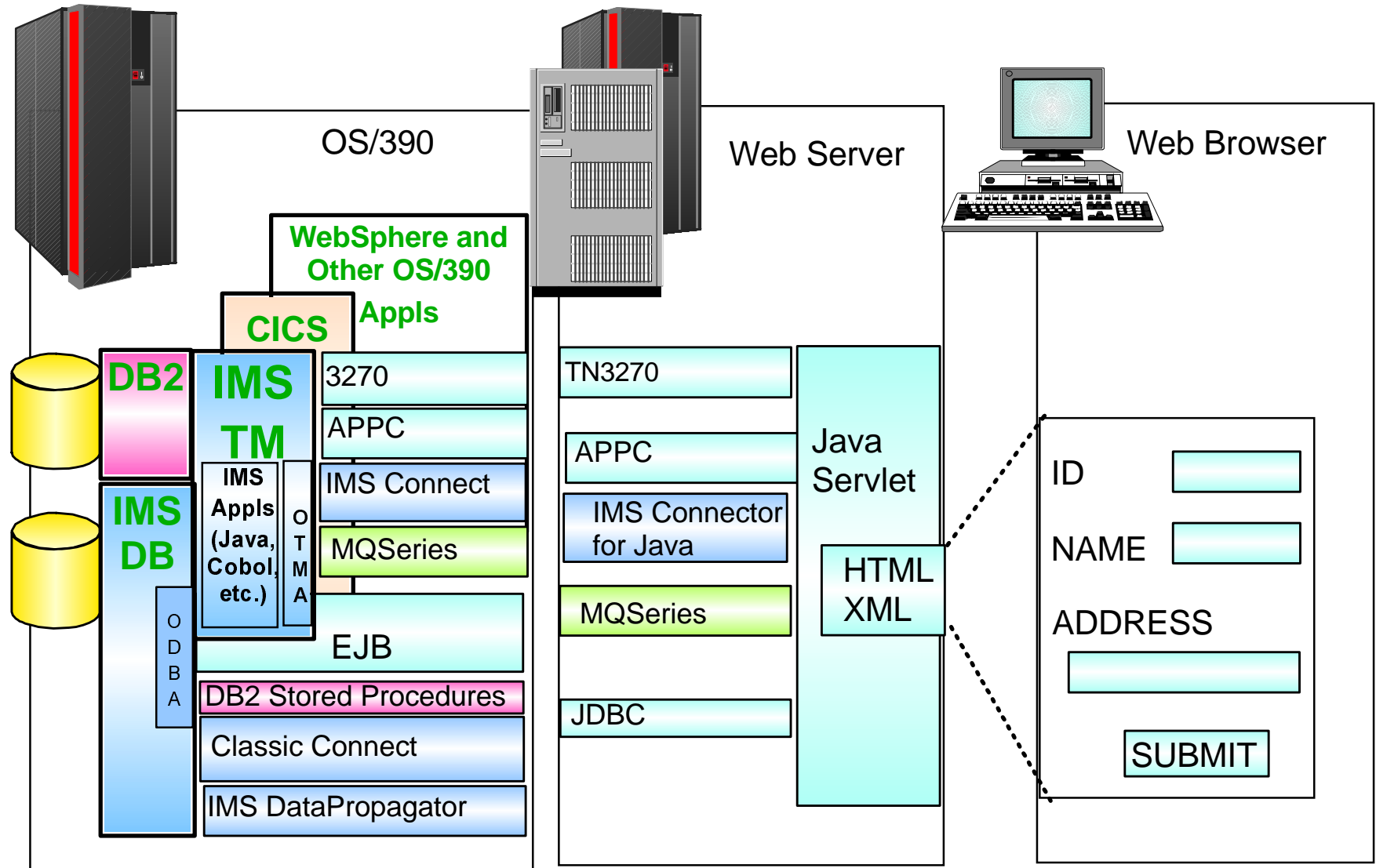
All statements regarding IBM's future direction and intent are subject to change or withdrawal without notice, and represent goals and objectives only.

Prices subject to change without notice. Contract your IBM representative or Business Partner for the most current pricing in your geography.

# Contents

- JAVA and IMS Connectors
- Building an IMS e-business Application
- Using IMS Java classes to build an IMS Application Program
- Summary

# Leveraging Applications and Data

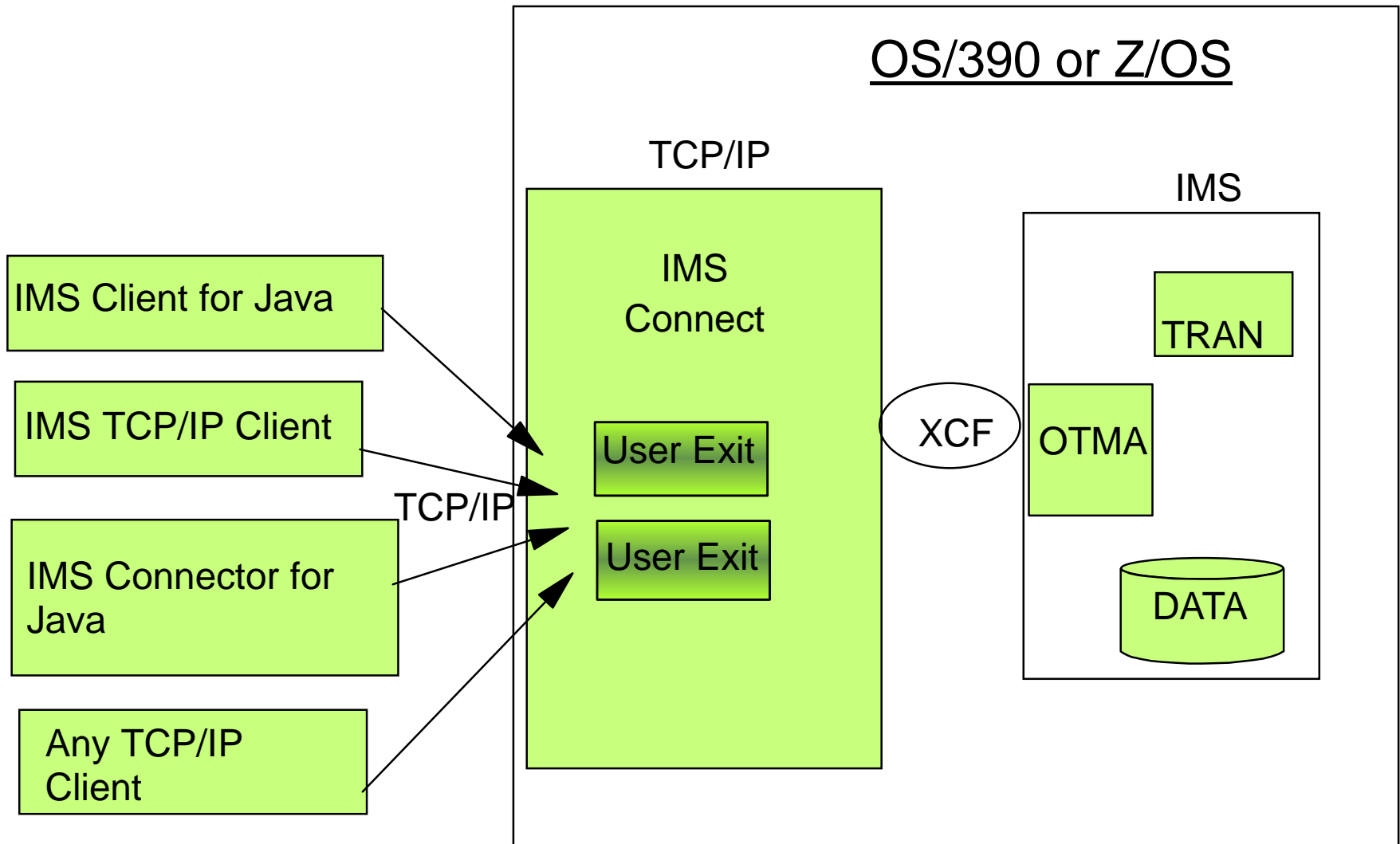


# IMS and Java

- **IMS Client for Java**
  - Applet
  - Application
- **IMS Connector for Java**
  - Application
  - Servlet
- **IMS Java Classes**
  - Application
  - Servlet
  - Enterprise Java Bean

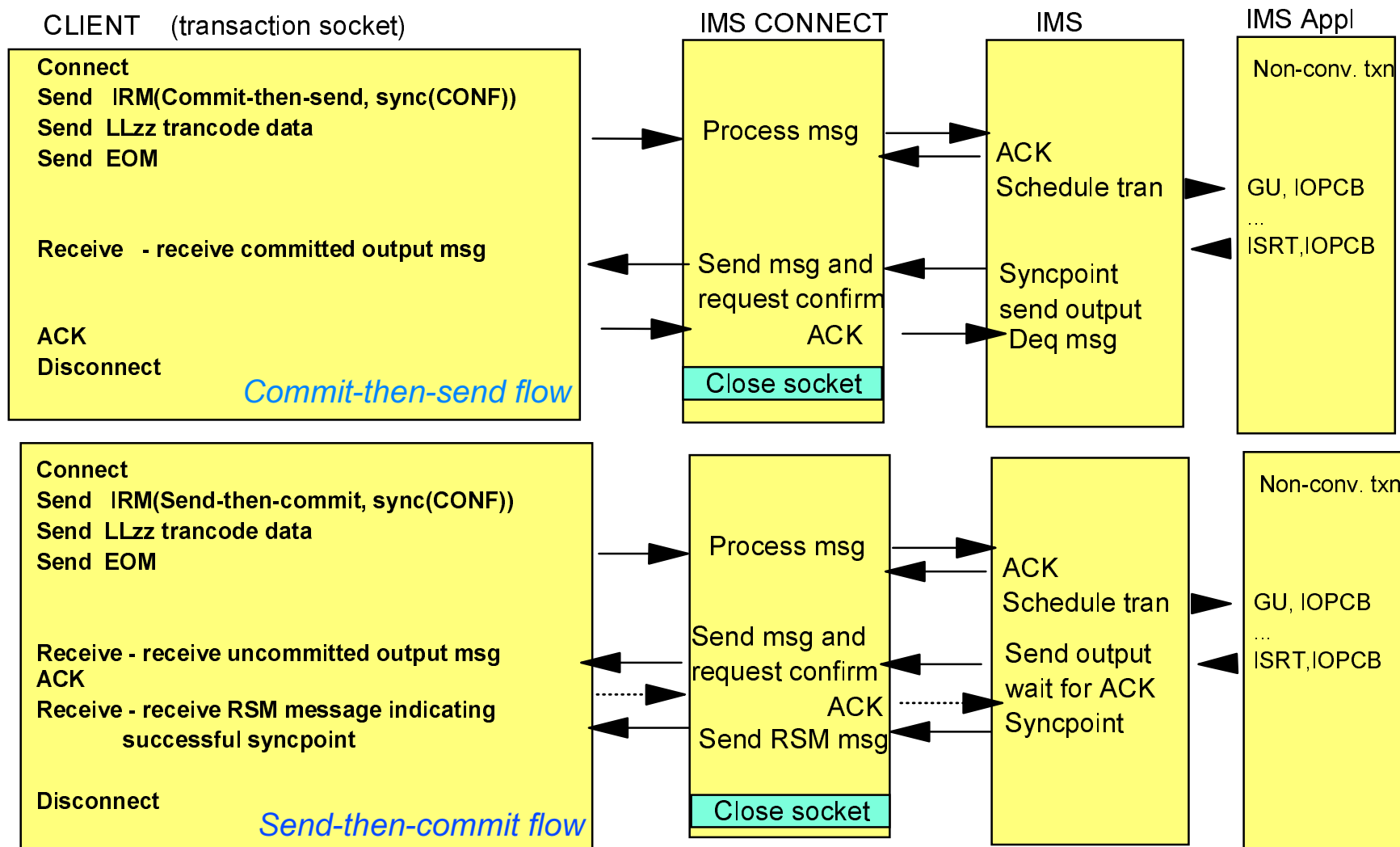
# IMS Connect

S/390 or Zseries



# IMS Connect

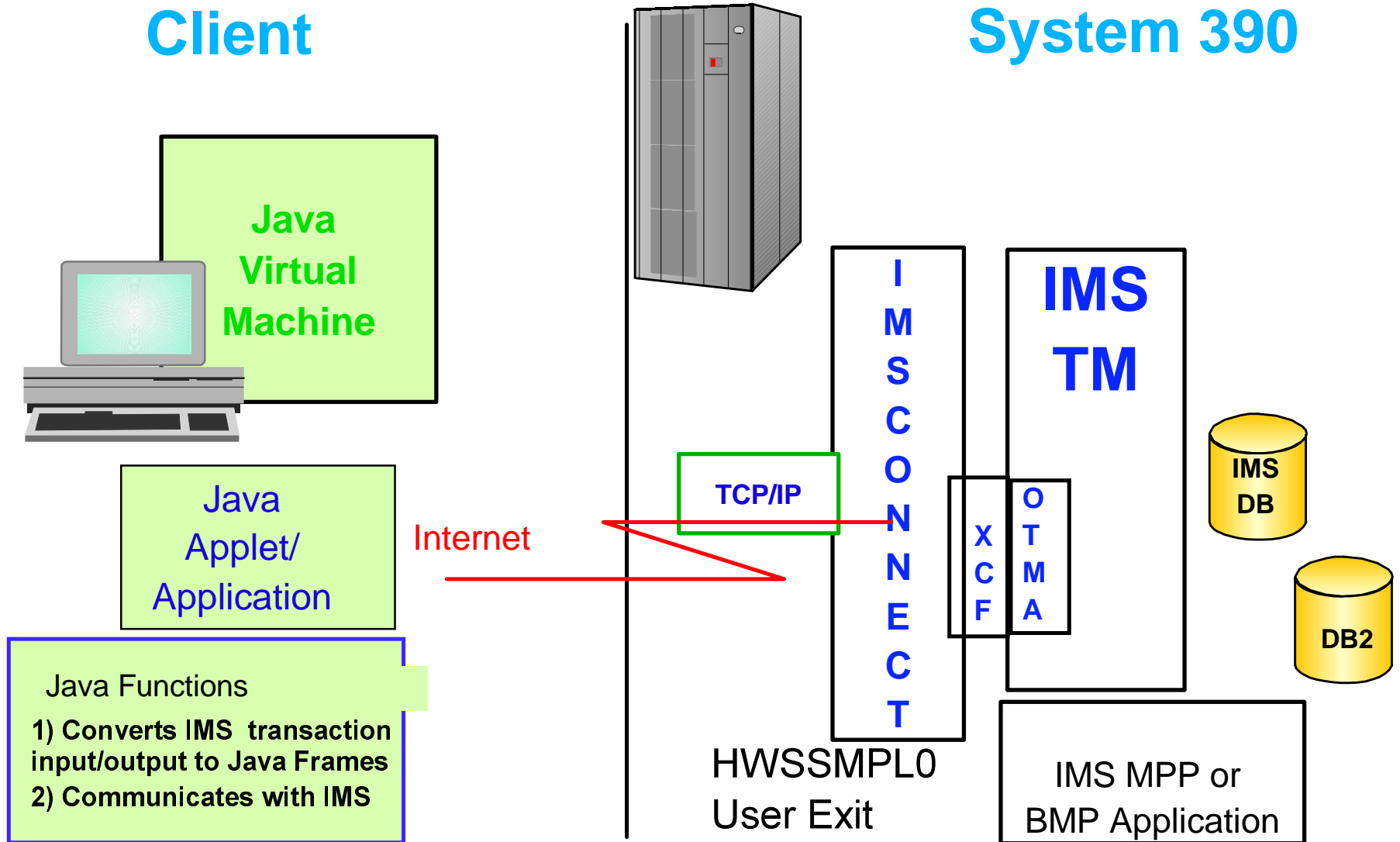
## Sockets - Basic Flow



# IMS Client for Java

<http://www-4.ibm.com/software/data/ims/about/imstoc/download/imstocfixes213.htm>

KUY0102 - 54887,012,649





# IMS Client for Java...

## INPUT DATA

## OUTPUT DATA

HostName	<input type="text" value="yourHost"/>	Port	<input checked="" type="radio"/> 3499
Transaction	<input type="text" value="yourTran"/>		<input type="radio"/> 9998
Client ID	<input type="text" value="clientID"/>	DataStore ID	<input type="text" value="DsName01"/>
SAFID	<input type="text" value="SAFUsrID"/>	GROUP	<input type="text"/>
Sync Level	<input type="text" value="NONE"/>	Commit Mode	<input type="text" value="1 - SEND THEN COMMIT"/>
Password	<input type="text"/>	Socket Type	<input type="text" value="1 - PERSISTENT SOCKET"/>

TRAN :	<input type="text"/>	RC =	<input type="text"/>
MOD :	<input type="text"/>	RS =	<input type="text"/>

OUTPUT  
TEXT :

Input Text

<input type="button" value="Send"/>	<input type="button" value="Receive"/>	<input type="button" value="Ack"/>	<input type="button" value="Nack"/>	<input type="button" value="Resume Tpipe"/>	<input type="button" value="Send Only"/>	<input type="button" value="Disconnect"/>	<input type="button" value="Close"/>
-------------------------------------	--	------------------------------------	-------------------------------------	---	--	---	--------------------------------------

# 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**

...

- **Commit-then-Send Sync\_Level=Confirm**
- **IRM header specifies:**

***Client id:*** to define the TPIPE name

***Persistent socket :*** to keep the connection active

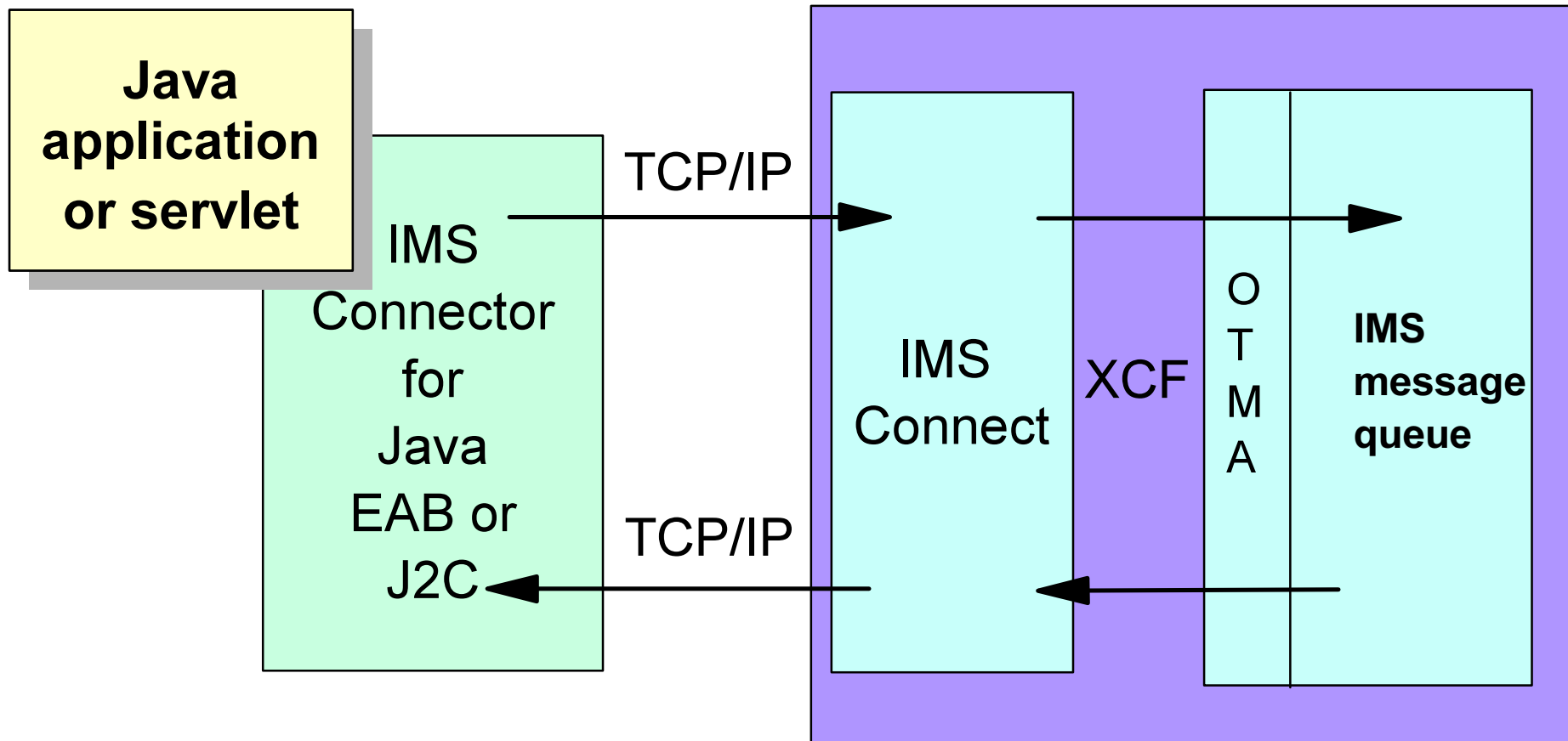
***Asynch request type:***

- ***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

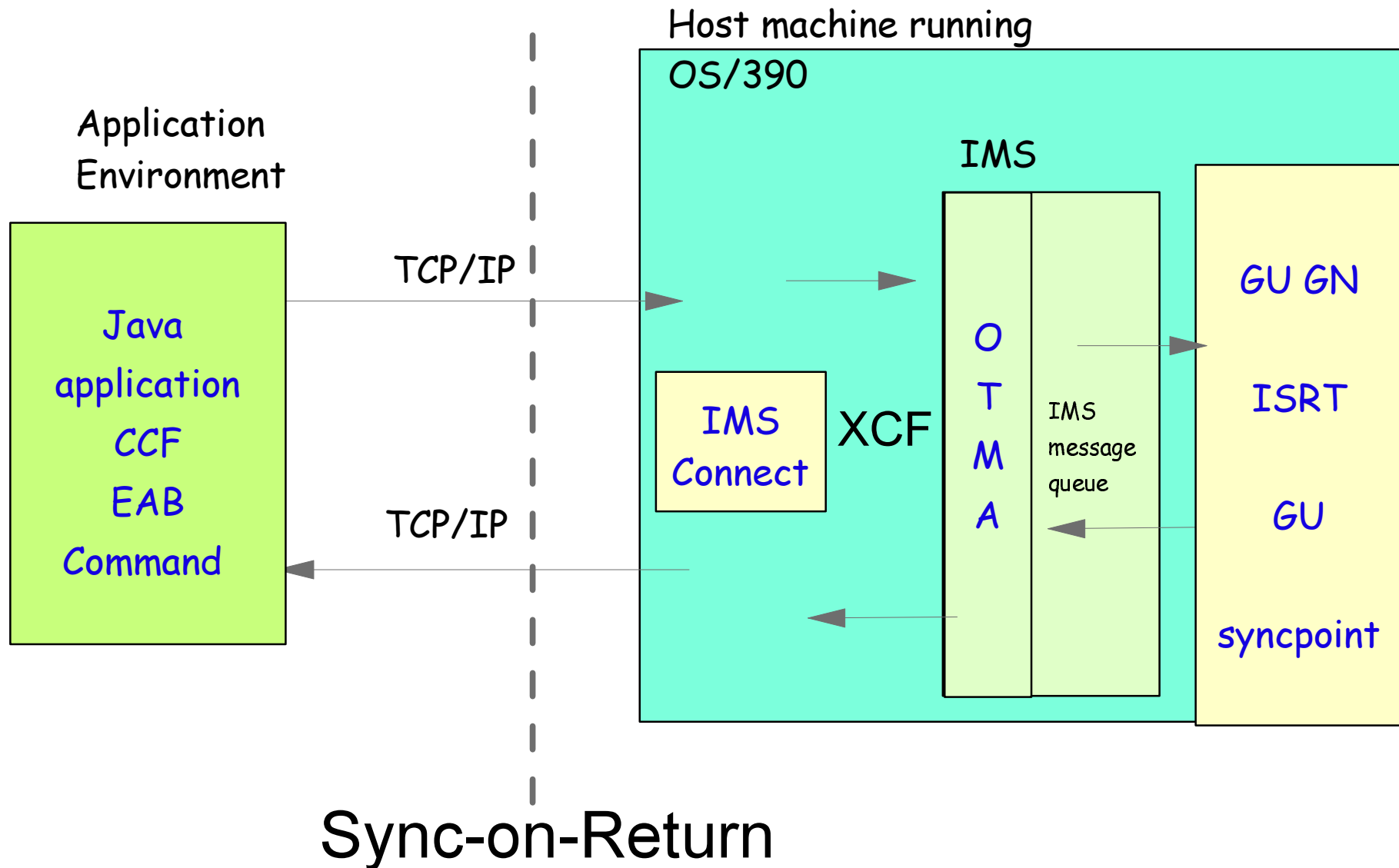
# What is IMS Connector for Java?



# IMS Connect

## Send-then-Commit

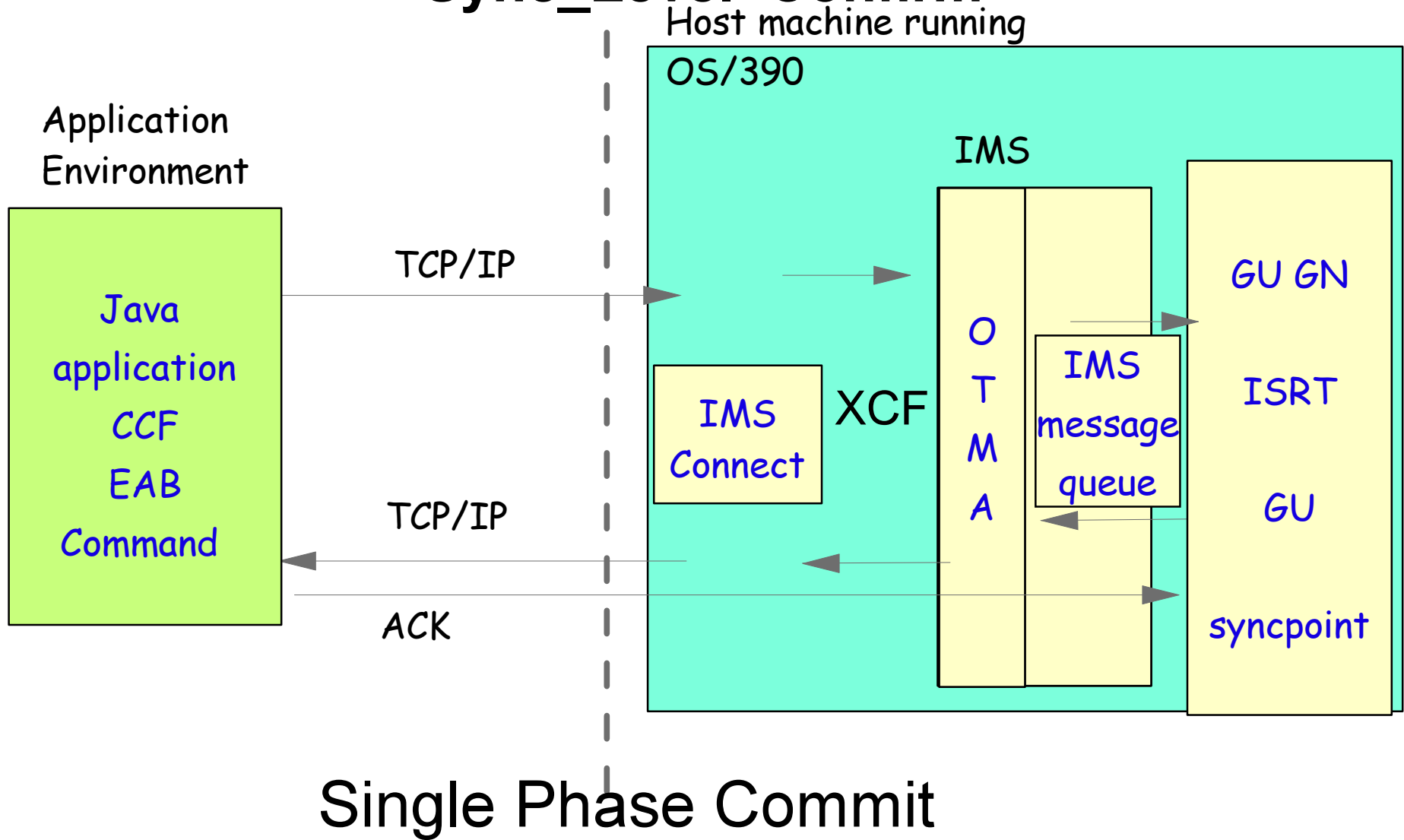
### Sync\_Level=None



# IMS Connect

## Send-then-Commit

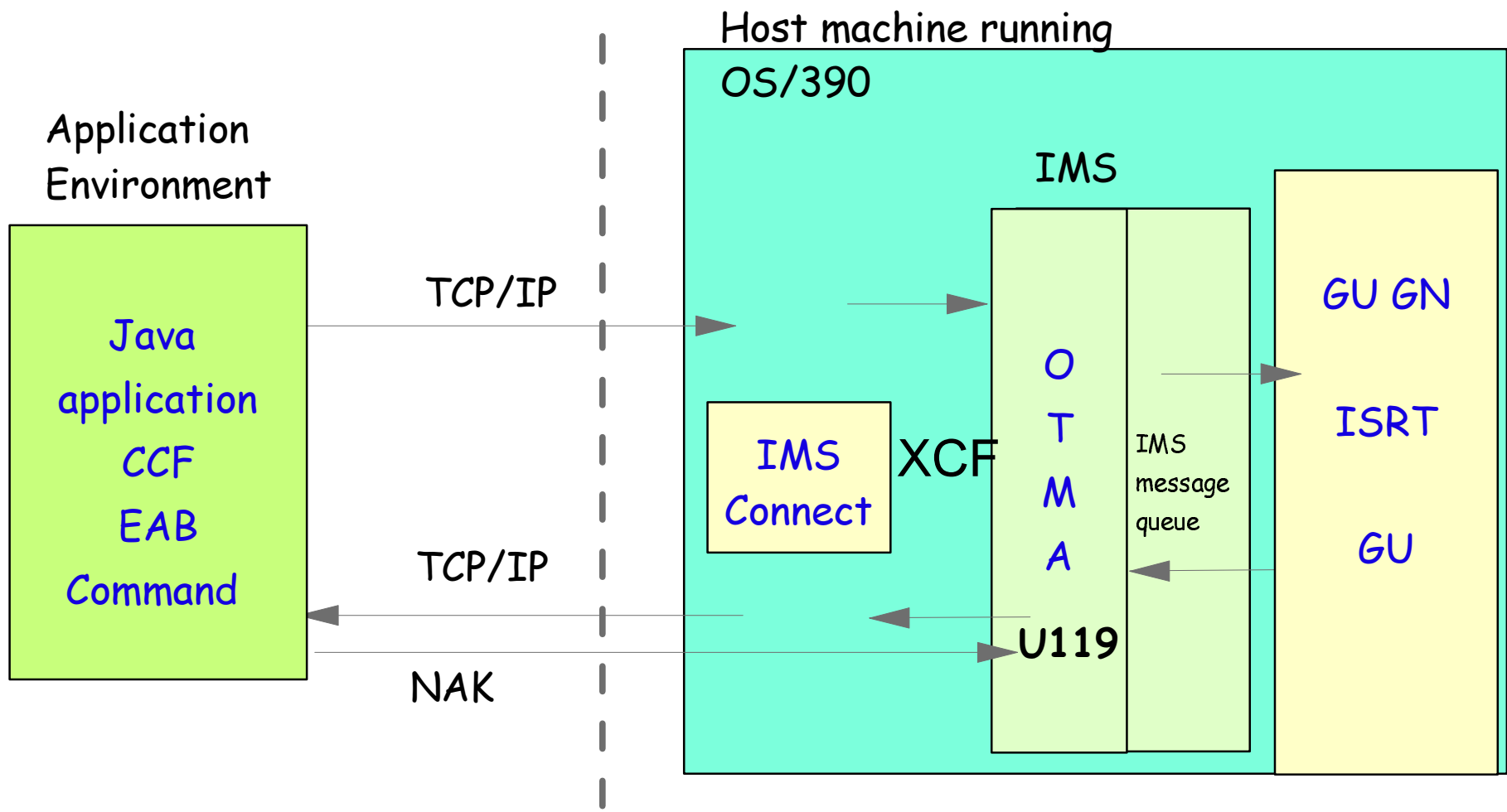
### Sync\_Level=Confirm



# IMS Connect

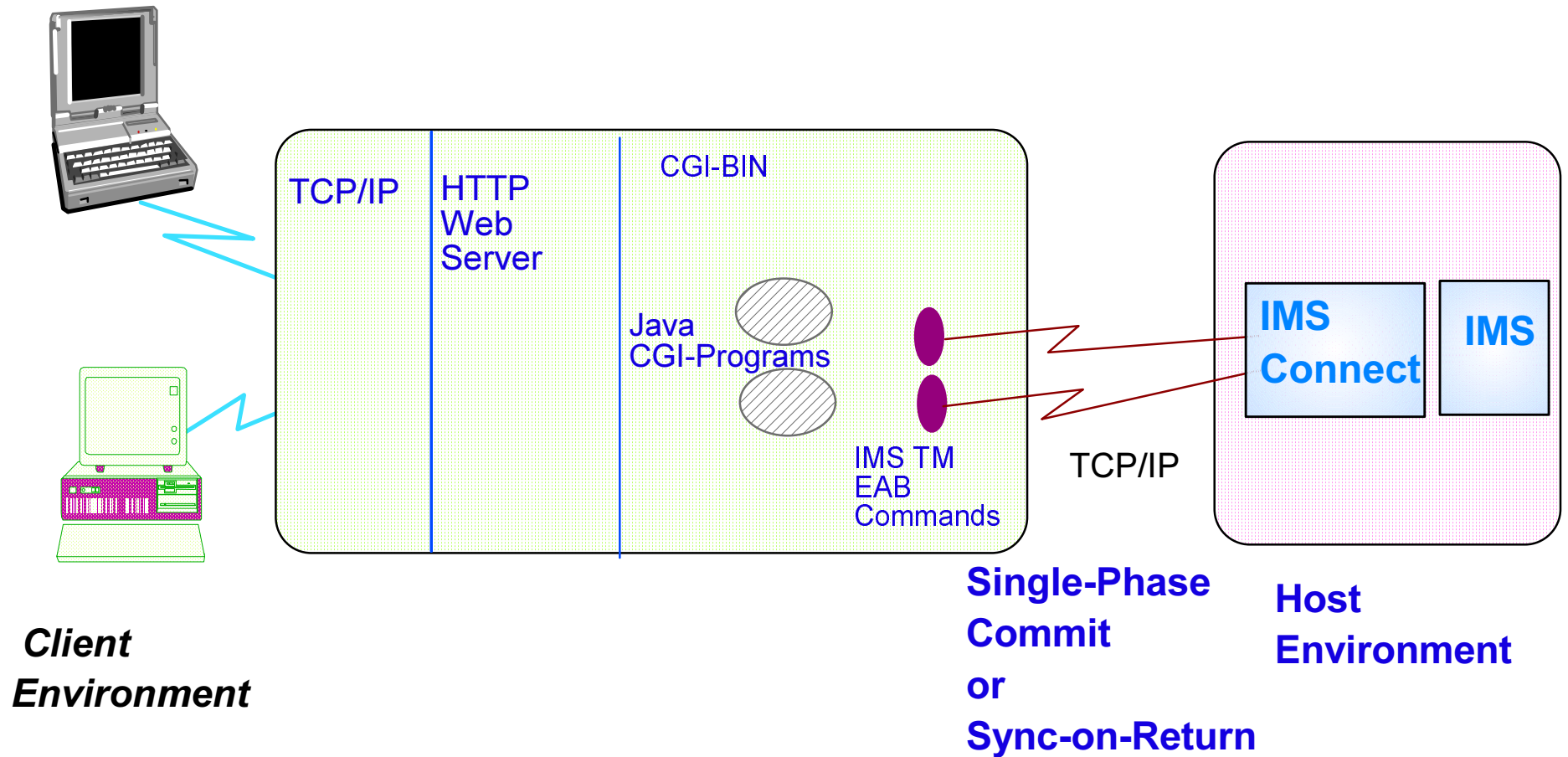
## Send-then-Commit

### Sync\_Level=Confirm-U119

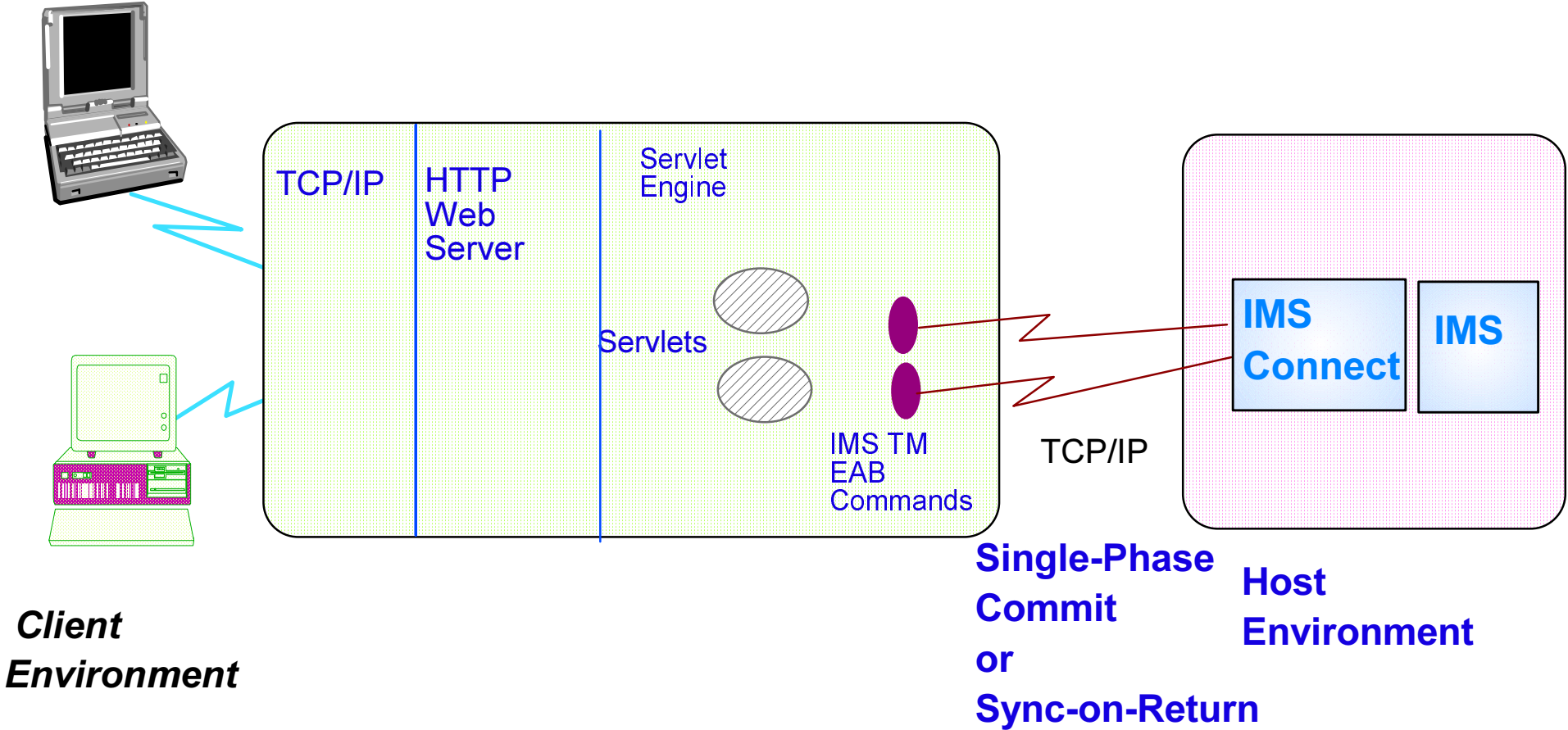


Single Phase Backout

# HTTP Server

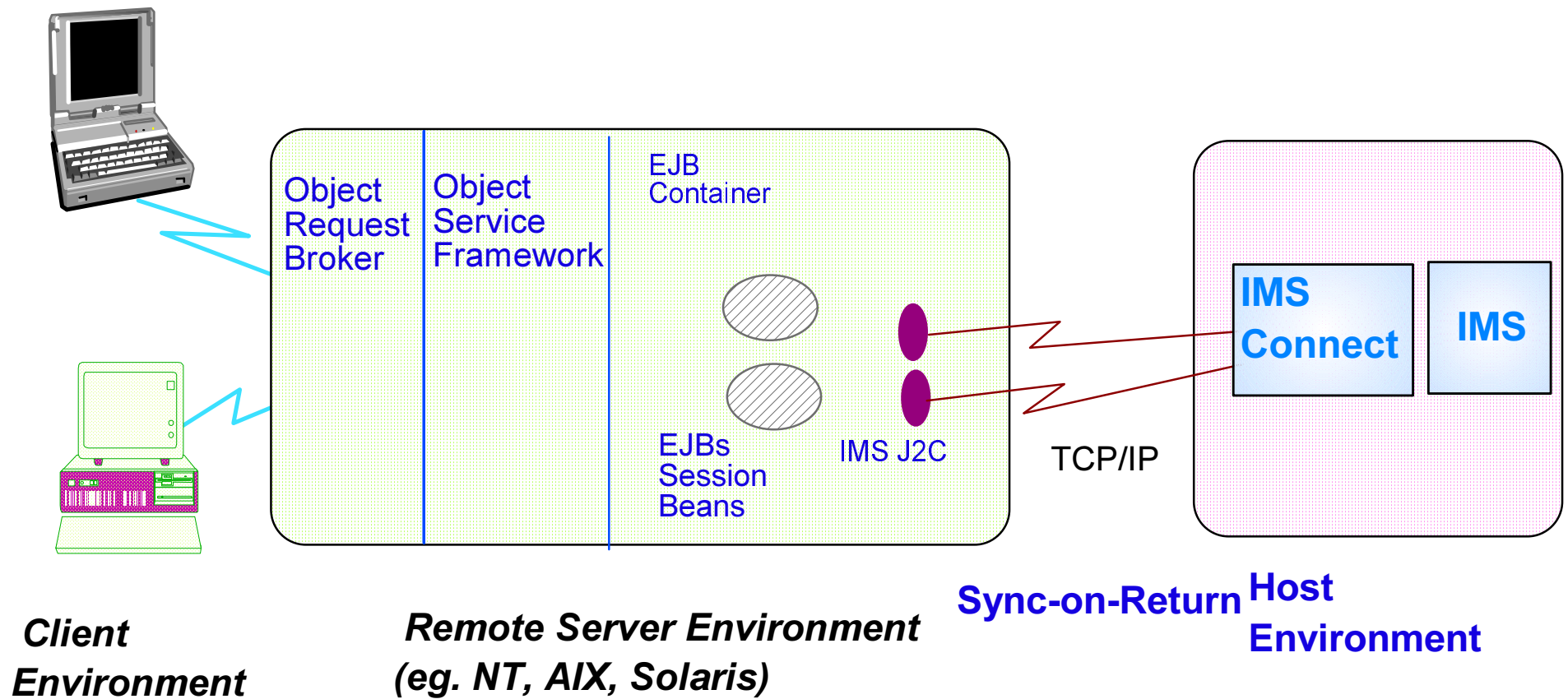


# WebSphere Application Server

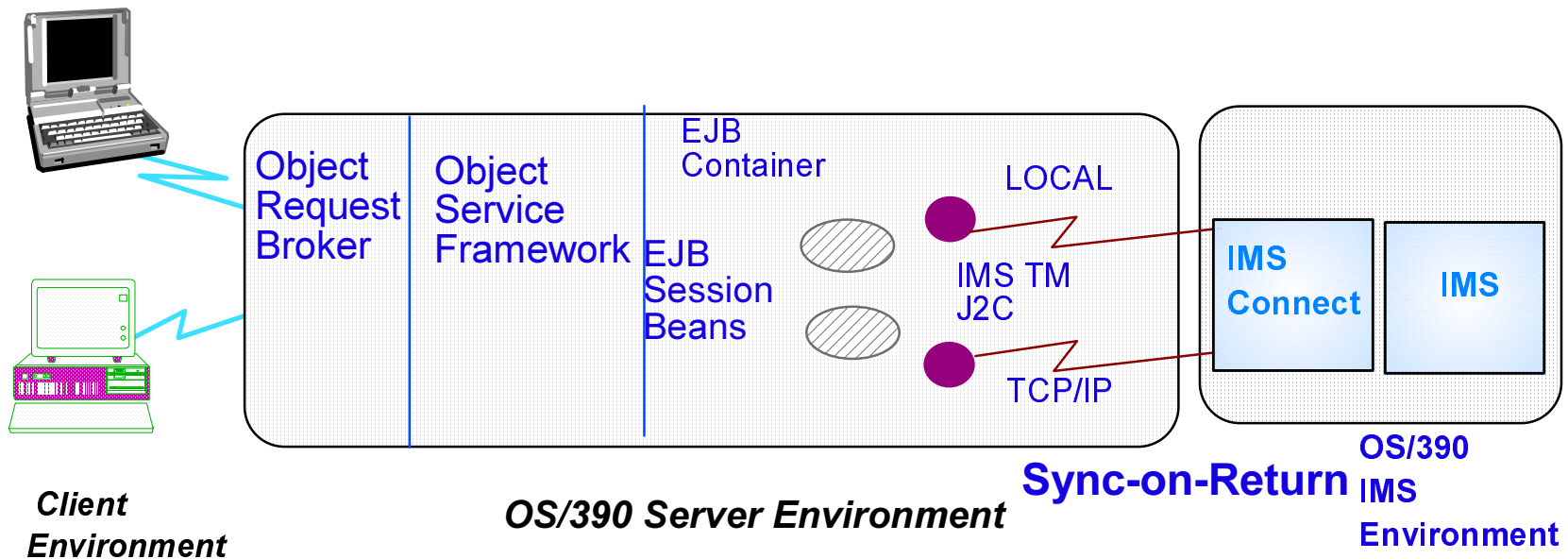




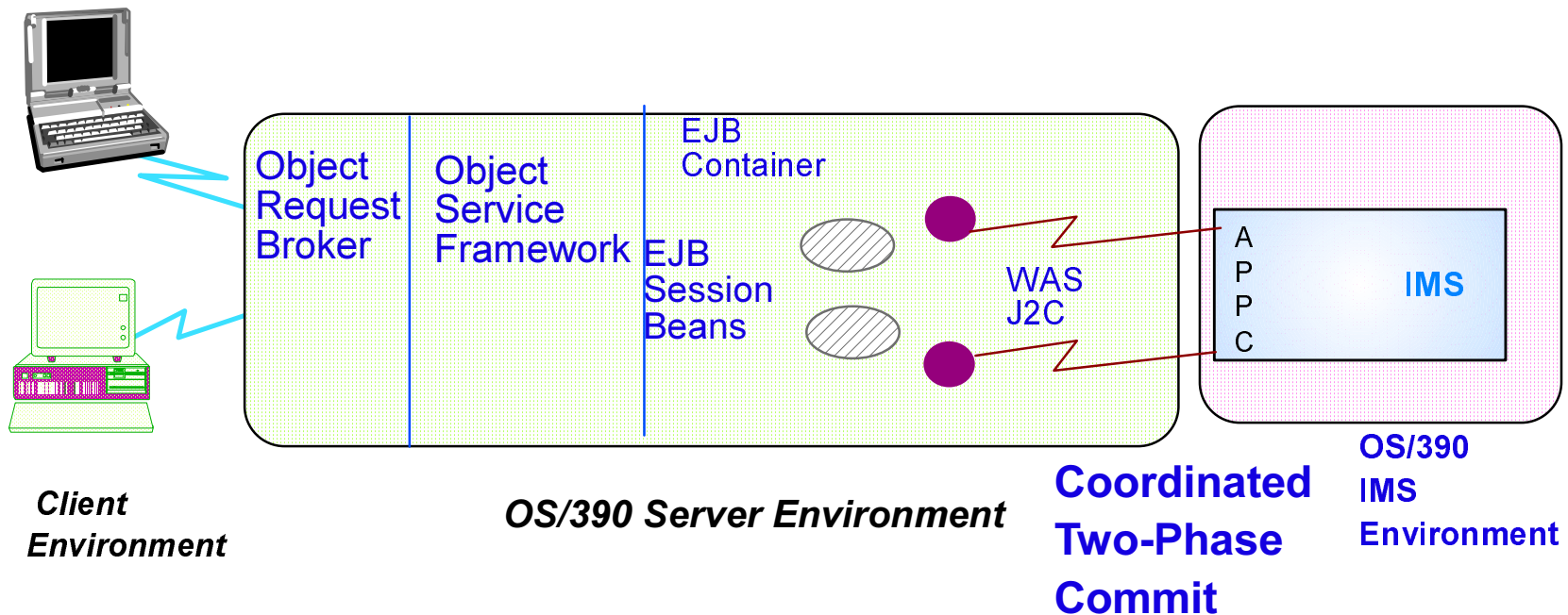
# WebSphere Application Server



# WebSphere Application Server Enterprise Edition - OS/390



# WebSphere Application Server Enterprise Edition - OS/390

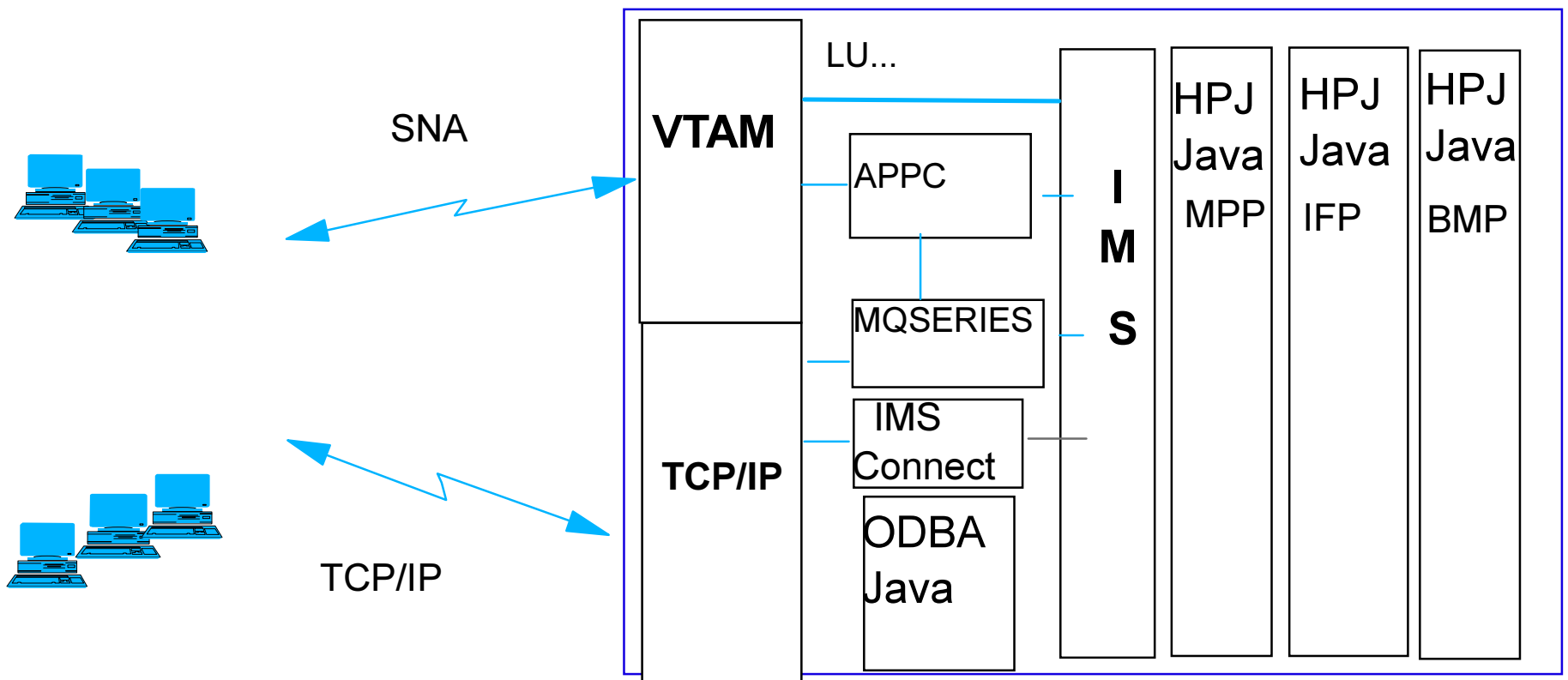


*APPC Protected Conversation  
Send-Then-Commit  
Sync\_Level=syncpoint*

# IMS Java

- **New function of IMS V7**
- **Capability to write, compile and run IMS Java programs**
  - Development environment
    - Provides a set of packages (groups of classes)
      - Allow access to IMS services
      - Support APIs familiar to Java programmers
    - Uses the S/390 HPJ (high performance java) compiler
      - Compiles bytecode into high-performance executables/DLLs
  - Runtime environment supports
    - Dependent regions (MPP, IFP, and BMP)
    - DB2 Stored Procedures, CICS, WebSphere
- **Benefit**
  - Incorporation of the Java programming model into the IMS environment

# Access to IMS Java Application Programs



# IMS Java...

## ■ IMS Java Application Pgm Interfaces (APIs)

### ▲ IMS Java APIs support different programming styles

#### ■ High-level standard database access methods

##### ▶ **JDBC** to access IMS data

Ex. "SELECT \_\_ FROM \_\_ WHERE \_\_"

- Interface maps segments to tables and fields to columns

##### ▶ **JDBC/SQLJ** interfaces to access DB2 data

#### ■ The use of IMS DB Package classes to access IMS DBs in a more familiar IMS programming style

Ex. "connection.getUniqueSegment (mySegment , ssalist)"

##### ▶ Supports building SSAs

##### ▶ Provides ability to get, insert, update, replace, delete segments

##### ▶ Supports full navigation of the DB hierarchy

# IMS Java...

## ■ IMS Java Application Pgm Interfaces (APIs)

### ▲ IMS Java APIs support different programming styles ...

- The use of the IMS Application Package classes to establish the application environment and request IMS services

Ex. `"messageQueue.getUniqueMessage (inputMessage)"`

Ex. `"trans.commit()"`

- ▶ Interface with the IMS message queues
- ▶ Request sync point services
- ▶ Java pgms must explicitly commit or rollback prior to retrieving another message or terminating

# IMS Java...

## ■ IMS Java Application Pgm Interfaces (APIs)

### ▲ IMS Java APIs support different programming styles ...

- The use of the low-level JavaToDLI interface

Ex. `JavaToDLI.execute("GU,dbaib,ioArea,ssa.getBytes()")`

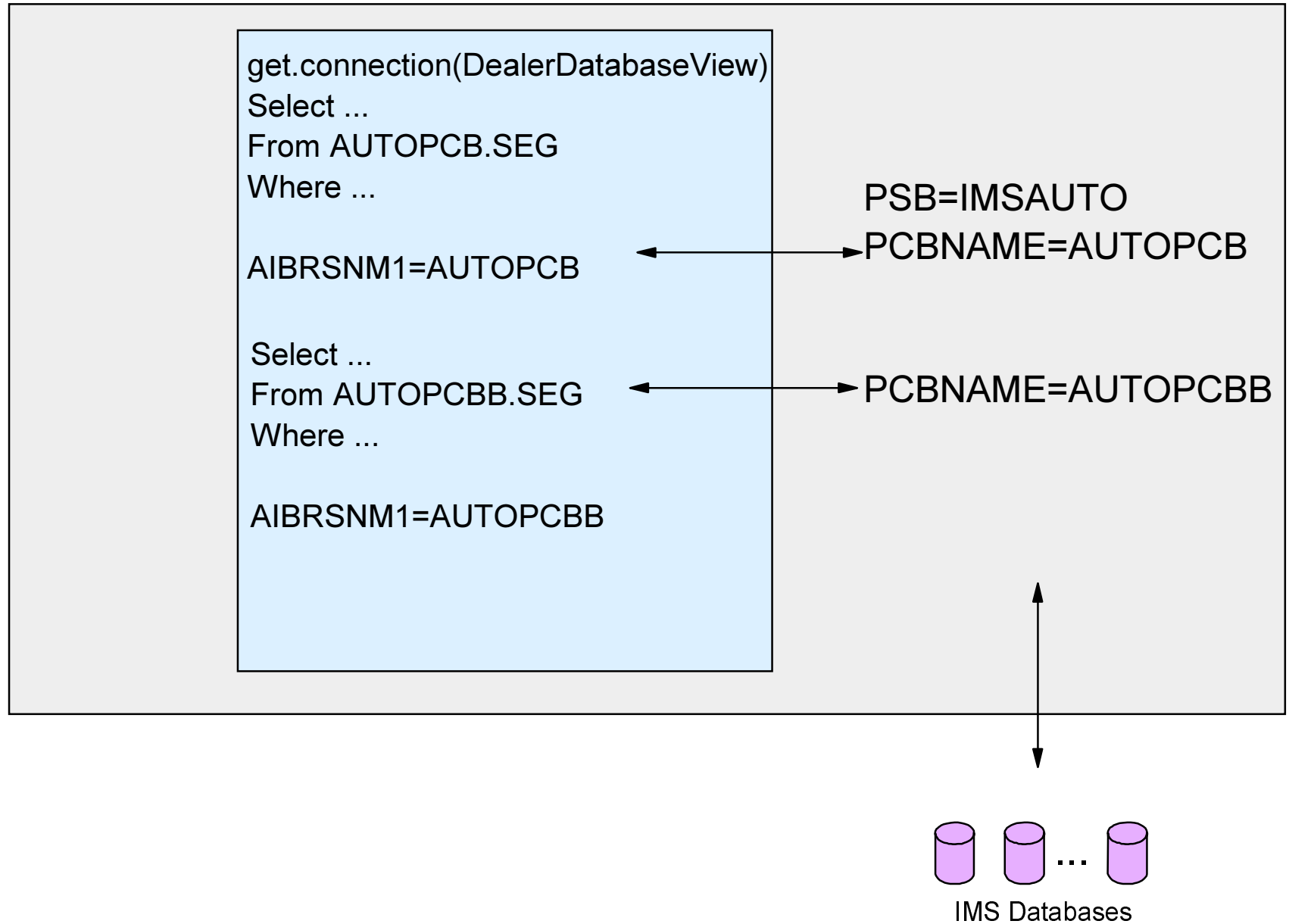
- ▶ Used to support the higher-level APIs
- ▶ Can be used for call functions not supported by higher levels
- ▶ Not meant for fully coding an application



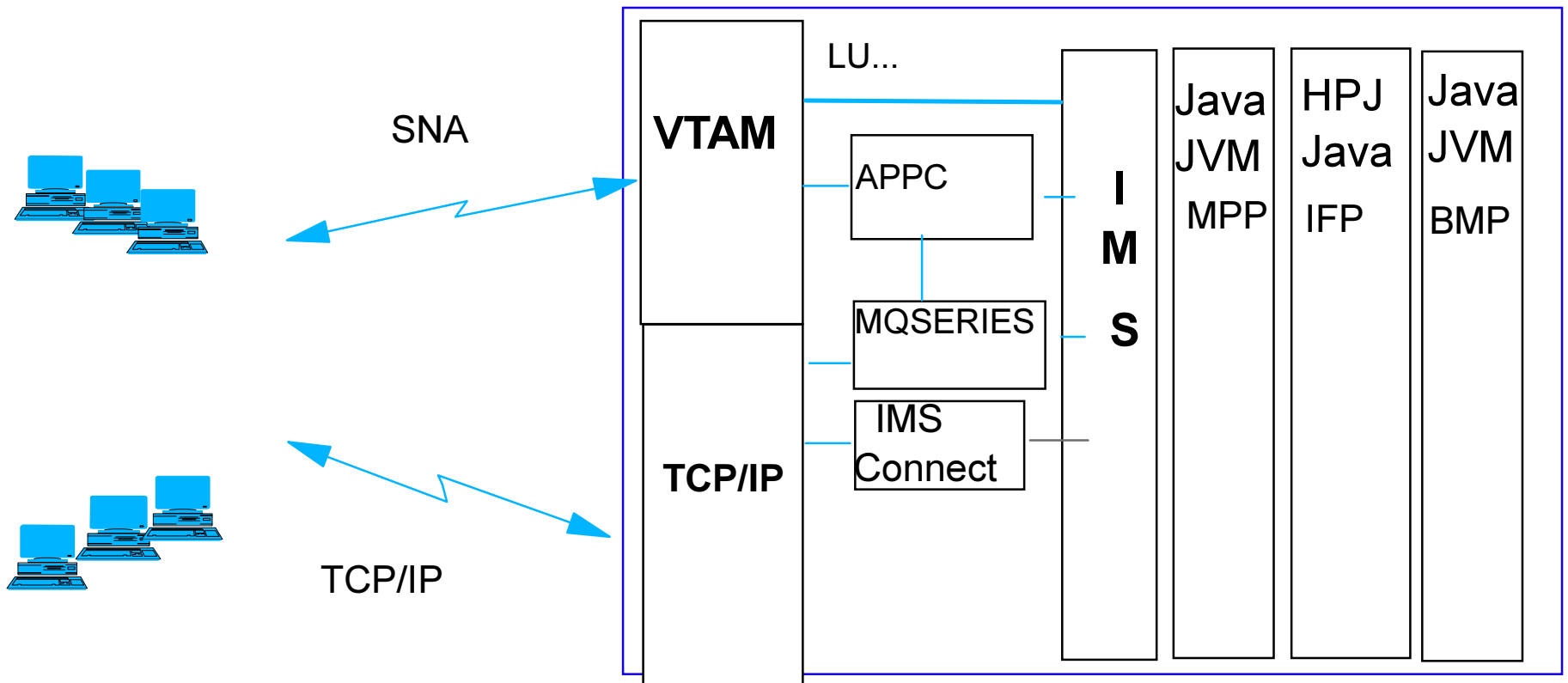
# Building an IMS Java Application

- Define IMS Trancode as **MODE=SNGL**
- Define classes for input and output message segments
- Define classes to access IMS DB segments
- 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
  - Note: **U118 abend** if you do not commit/rollback

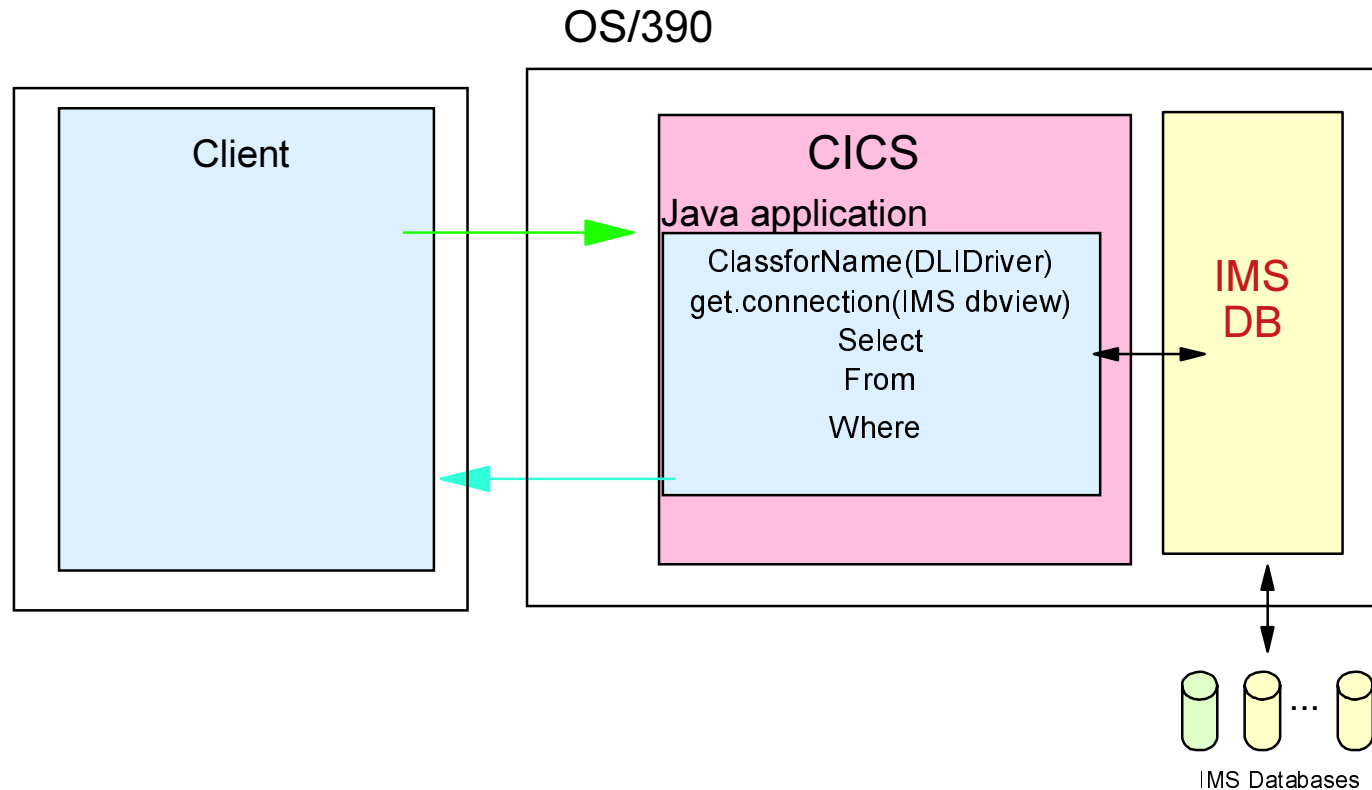
# JDBC Connection to IMS Database



# Access to IMS JVM Regions Java Application Programs

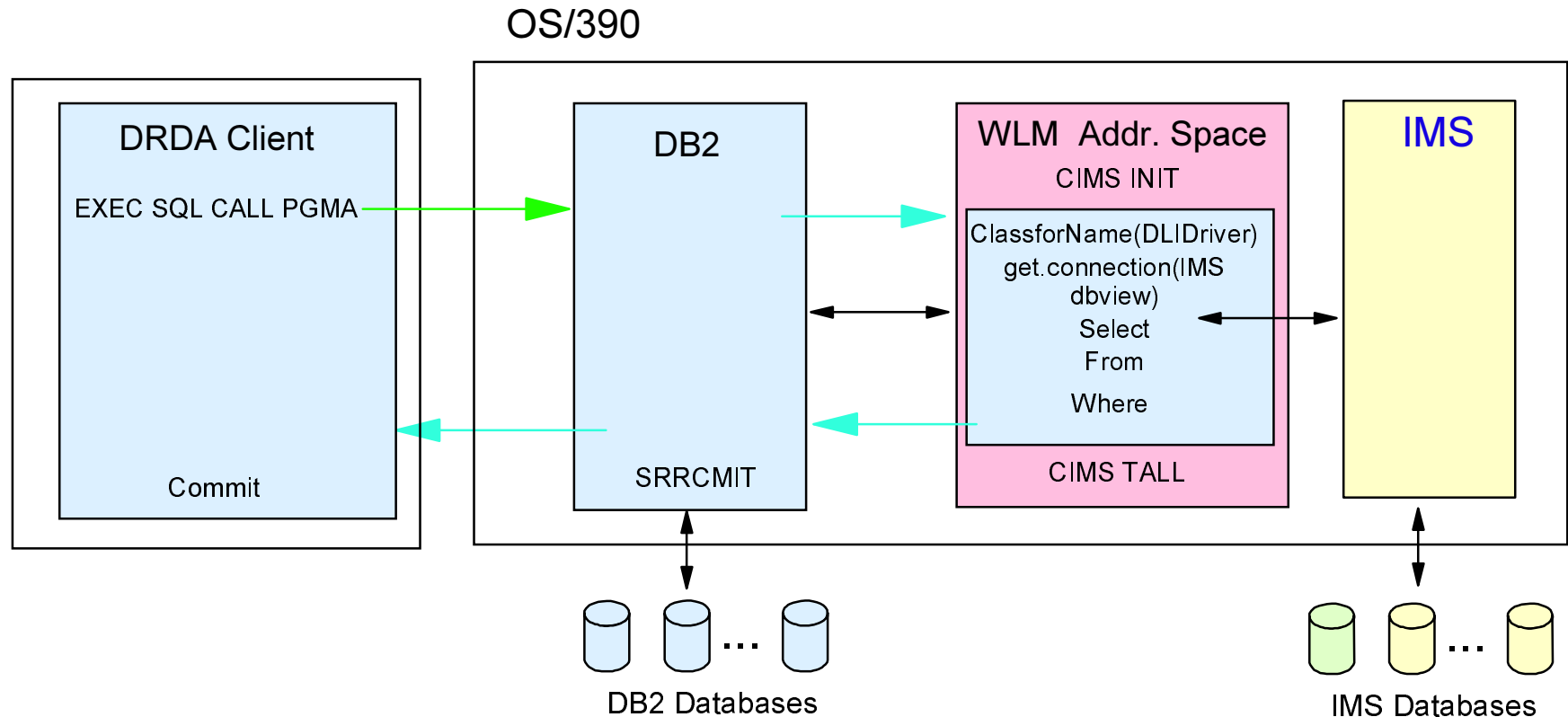


# Using IMS V7 Java Classes JDBC interface



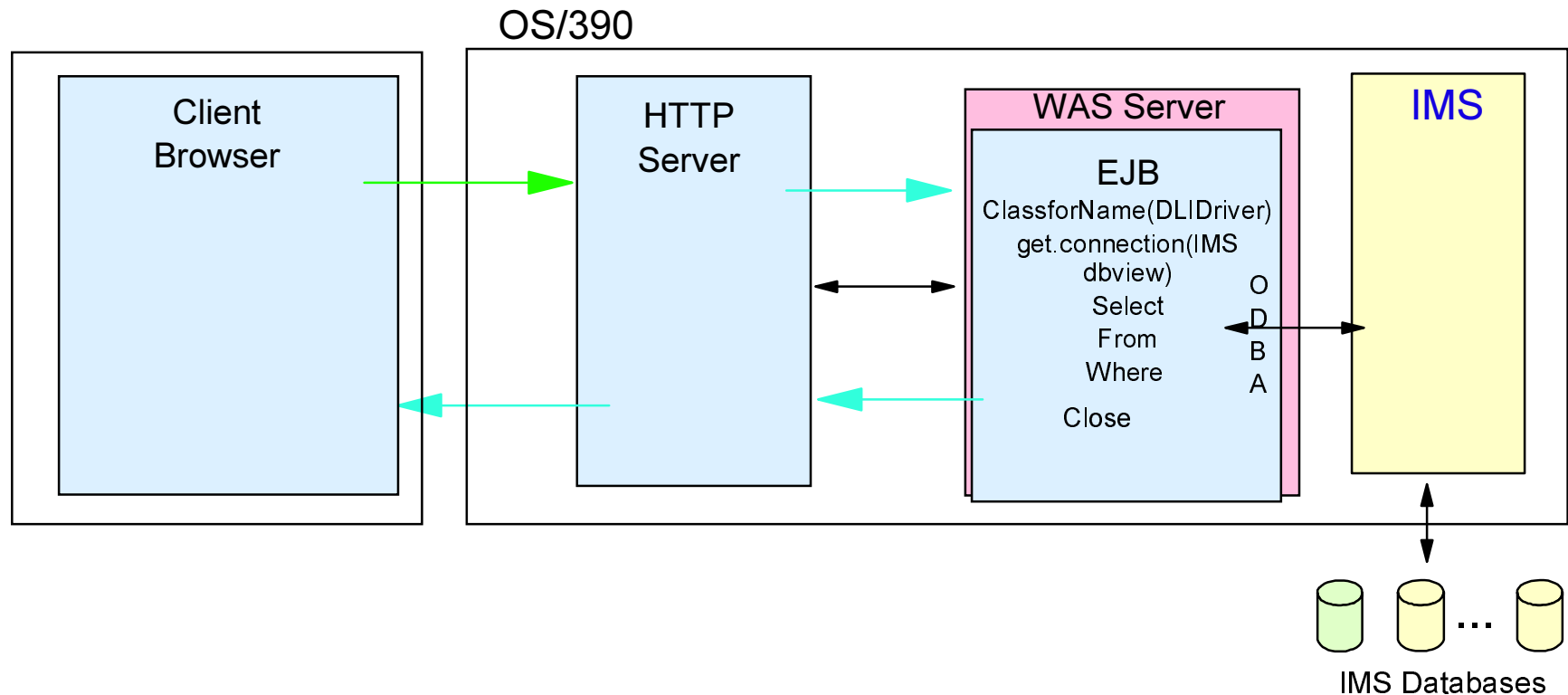
- **CICS Java Application example**
  - IMS Java Classes can be used to access IMS DB

# Using IMS V7 Java Classes JDBC interface



- **DB2 Java stored procedure example**
  - IMS Java Classes can be used to access IMS DB

# OS/390 WebSphere Application Server Using IMS V7 Java Classes JDBC Interface



**Requirement: To provide OS/390 WebSphere Application Server JDBC access to IMS DB**

# What is IMS Java? -- Summary

- **Function that enables the use of Java for IMS application programming**

- Requires only basic IMS knowledge
- Supports IMS application services
- Provides a foundation for the visual tools

- **IMS Java classes**

- Support conversational and non-conversational transactions
- Support MFS options
- Run in IMS dependent regions
- Provide JDBC access to IMS DB
- Support JDBC/SQLJ access to DB2
- HPJ complied support
  - JVM support later