



# IMS Application Programming Techniques

Kenny Blackman  
[kblackm@us.ibm.com](mailto:kblackm@us.ibm.com)  
IBM

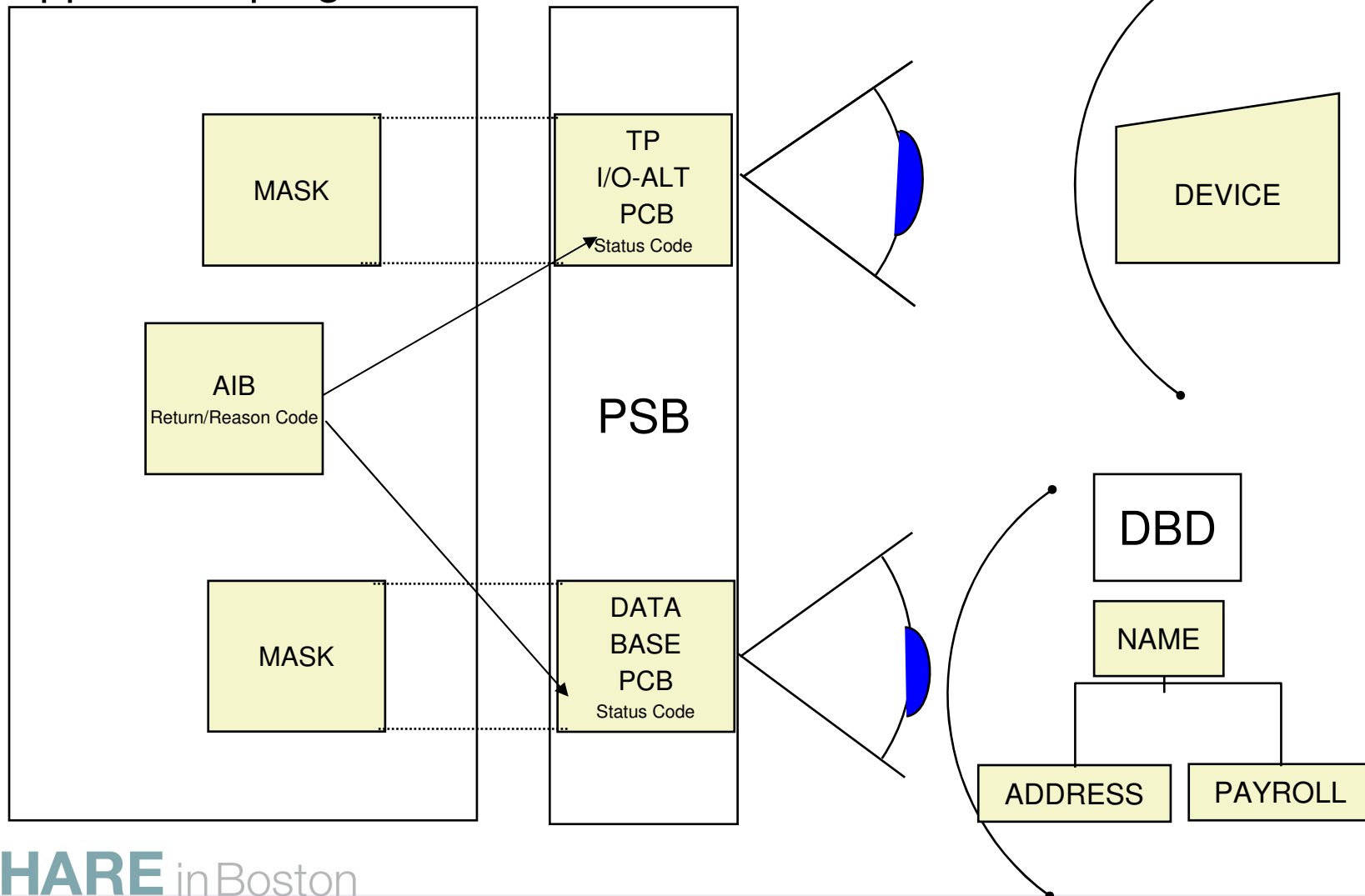
8/02 /2010  
Session



**SHARE** in Boston

# Device and Data Independence

Application program



# Application Interface Block (AIB)

- An application program can refer to a PCB by a given NAME, not an address (PCBNAME is 8 bytes).
  - For the I/O-PCB, the name is 'IOPCBbbb'
  - For DB-PCB, the name is specified in the PSBGEN:
    - PCBNAME=... parameter on PCB macro
    - LIST=Y|N - Display PCBNAME in PSB listing?

Most DL/I calls can be issued in two ways:

Using a PCB (standard technique):

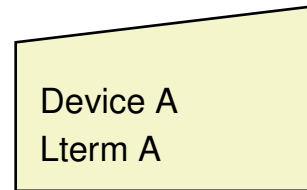
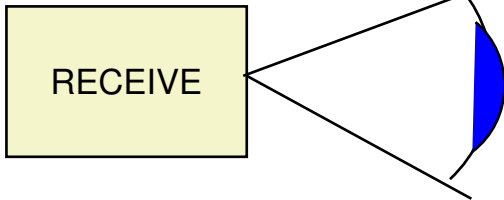
```
CALL xxxTDLI ( <count>,FUNC,PCB,I/O AREA,... )  
CEETDLI
```

Using an AIB (new technique):

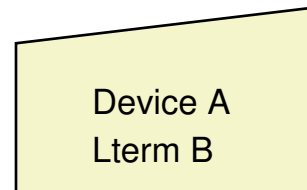
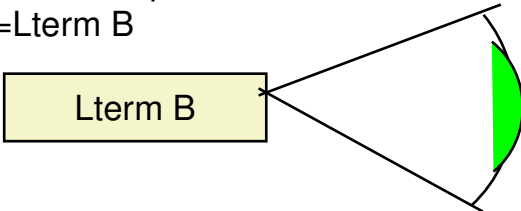
```
CALL AIBTDLI ( <count>,FUNC,AIB,I/O AREA,... )  
AERTDLI  
CEETDLI
```

# Alternate PCBs

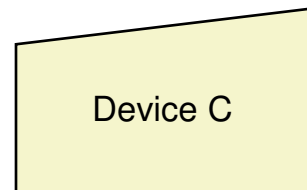
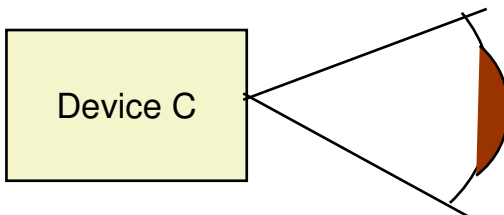
I/O PCB



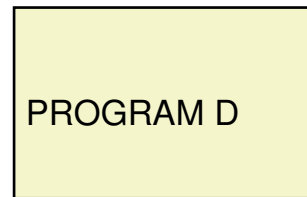
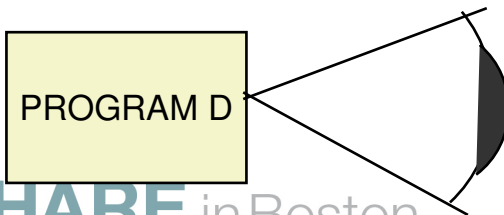
ALTERNATE Response PCB  
LTERM=Lterm B



ALTERNATE Express PCB



Modifiable ALTERNATE PCB



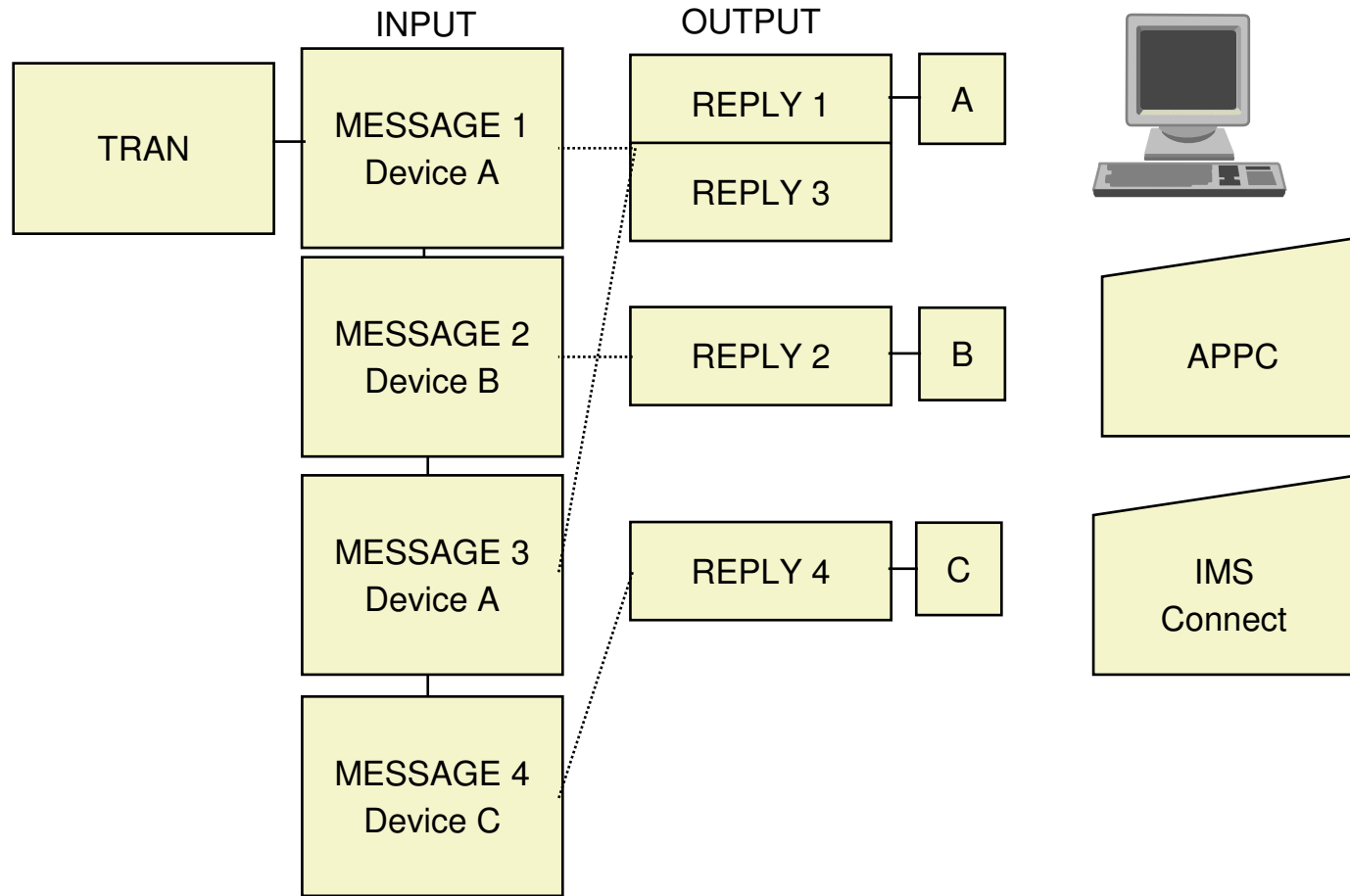
# IMS Application Program Runtime Environments



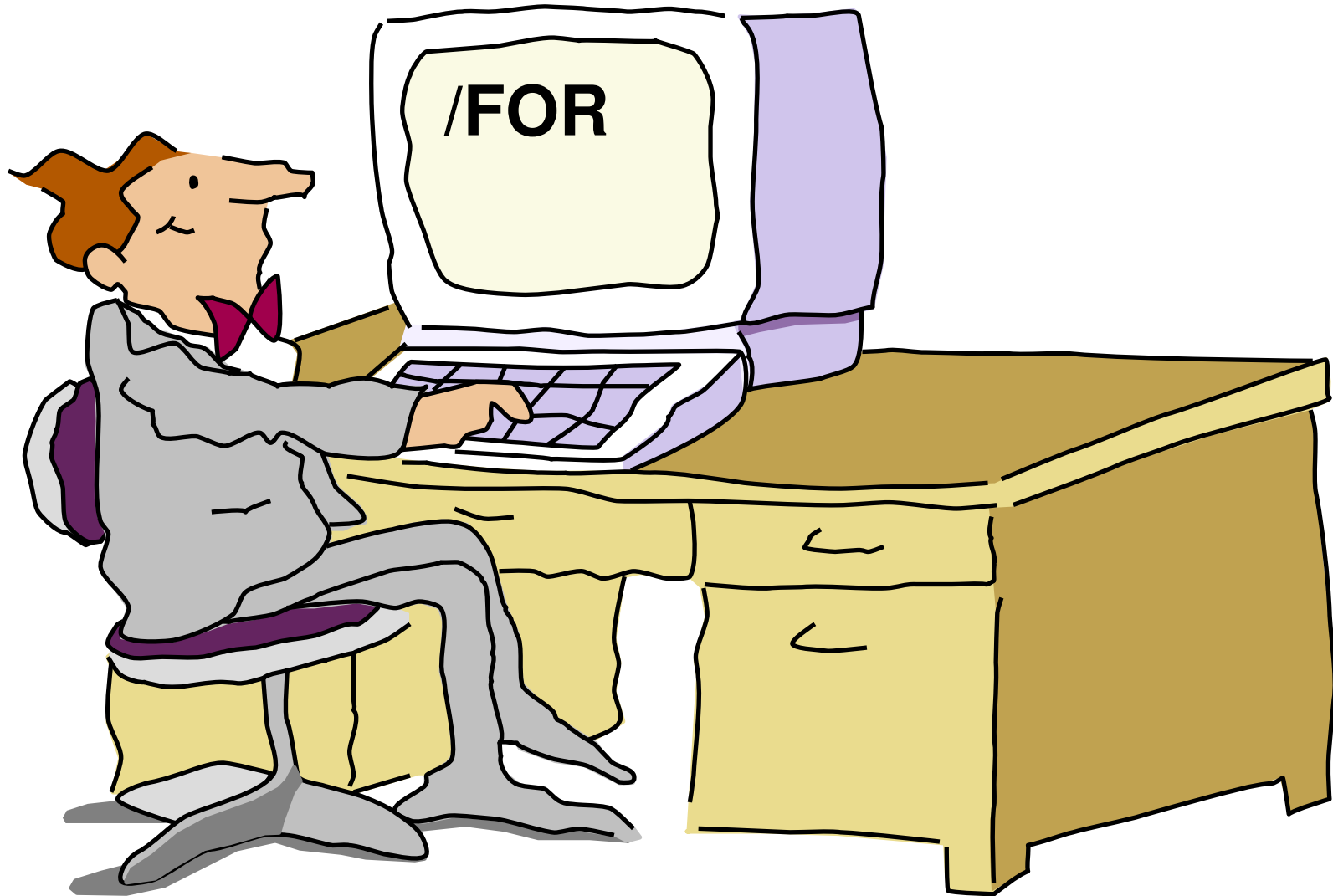
## Application Programs

		SUPPORTED BY IMS TM/DB CTL		STAND ALONE	
		MESSAGE REGION <i>(MPP, IFP, JMP)</i>	BATCH MESSAGE Driven PROCESSING (BMP) BATCH Non-MESSAGE Driven PROCESSING (BMP, JBP)	DB BATCH REGION <i>(DLI)</i> TM BATCH REGION <i>(DB2)</i>	
IMS TM CONTROL REGION <i>(CTL)</i>					
	FUNCTIONS				
	•QUEUING	•SCHEDULED BY	IMS	USER	USER
	•SCHEDULING	•ONLINE DB'S	YES	YES	SOME PROGRAMS
	•LOGGING	•OS/VS FILES	NO	YES	ARE INTER- CHANGEABLE
	•I/O - DATA BASE - TERMINAL	•MSG Q	YES	YES	NO
	•I/O PCB	YES	YES	OPTIONAL	

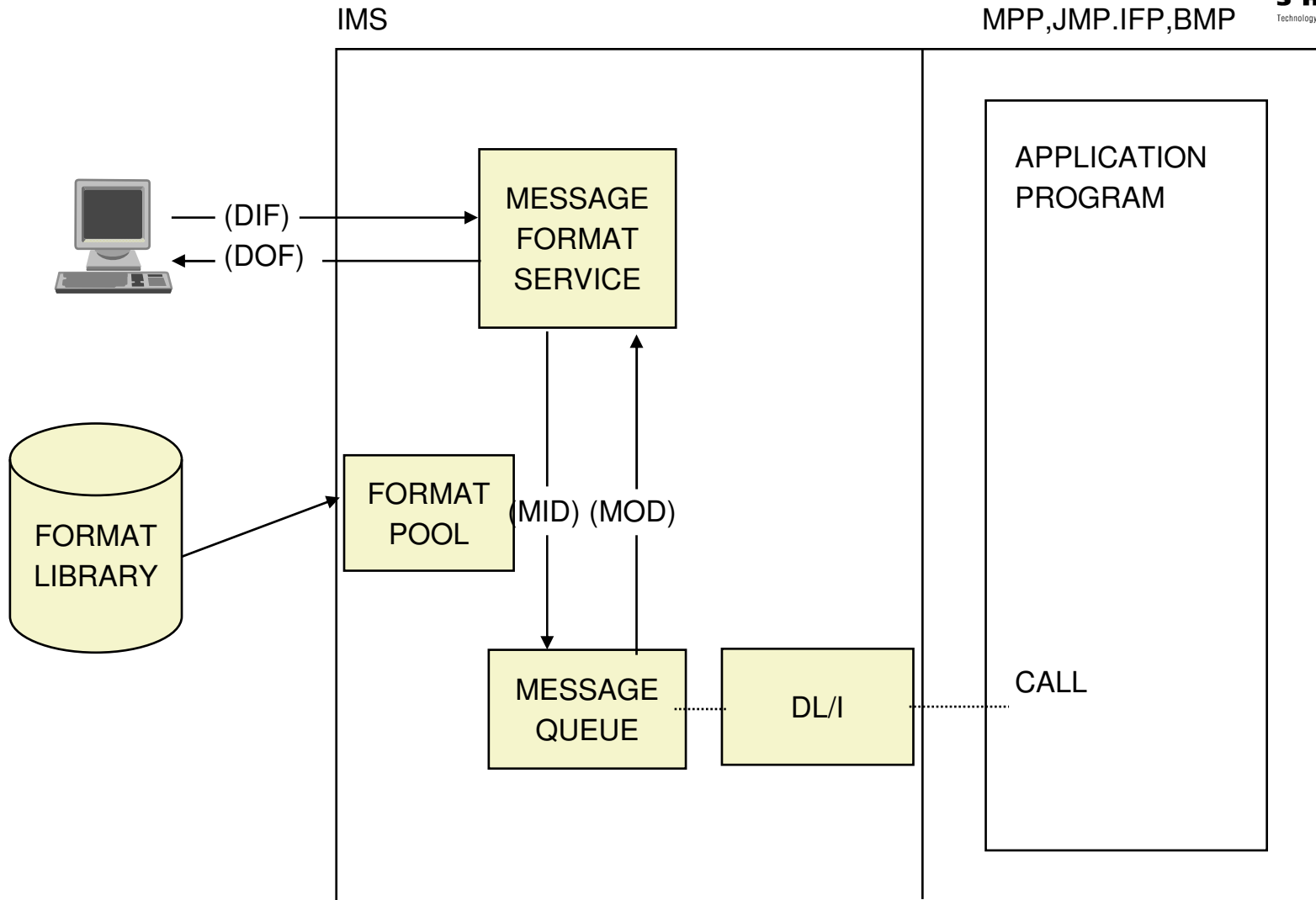
# Message Queuing



# Message Format Service



# MESSAGE FORMAT SERVICE



DIF - DEVICE INPUT FORMAT

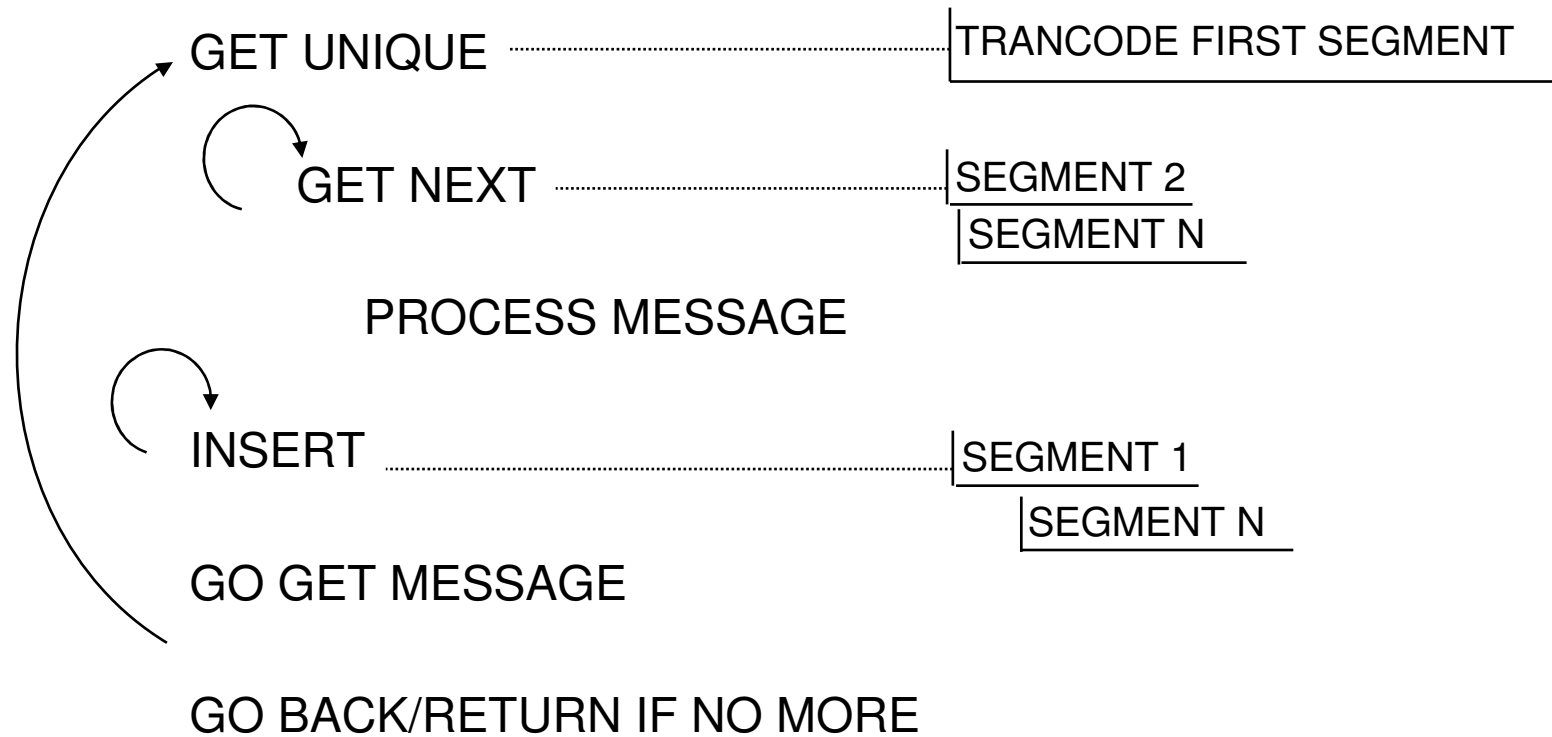
DOF - DEVICE OUTPUT FORMAT

MID - MESSAGE INPUT DESCRIPTOR

MOD - MESSAGE OUTPUT DESCRIPTOR



# Call Sequence

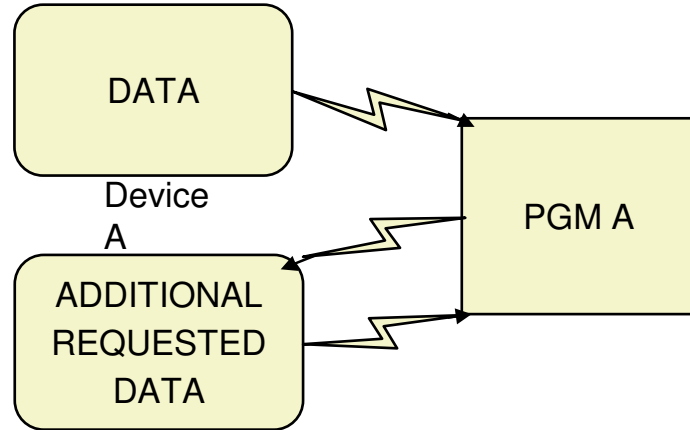


# Problem W/O Conversational

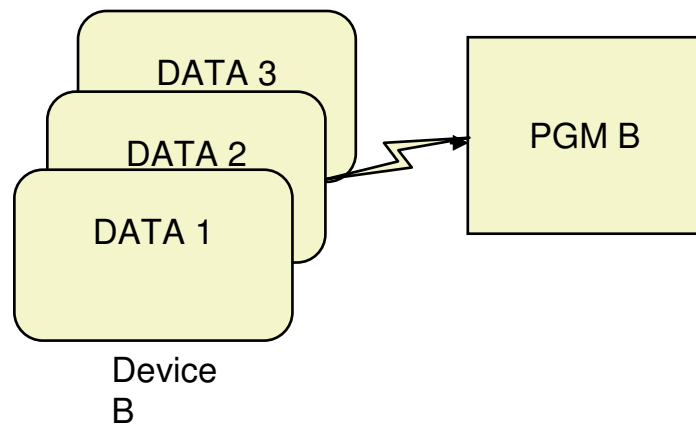


## Processing requirement

### Interactive program

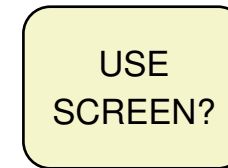


### Multi-screen input



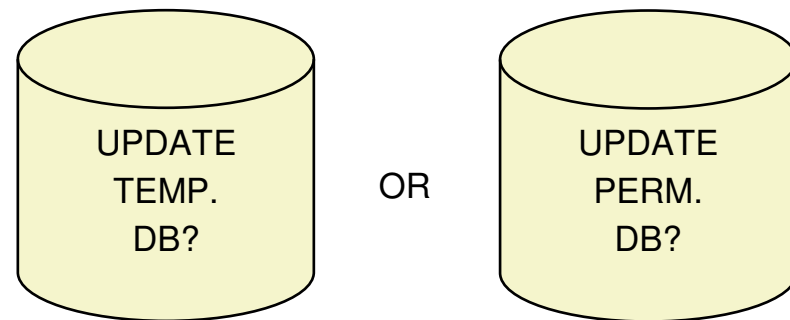
## Need temporary storage

### Solution 1



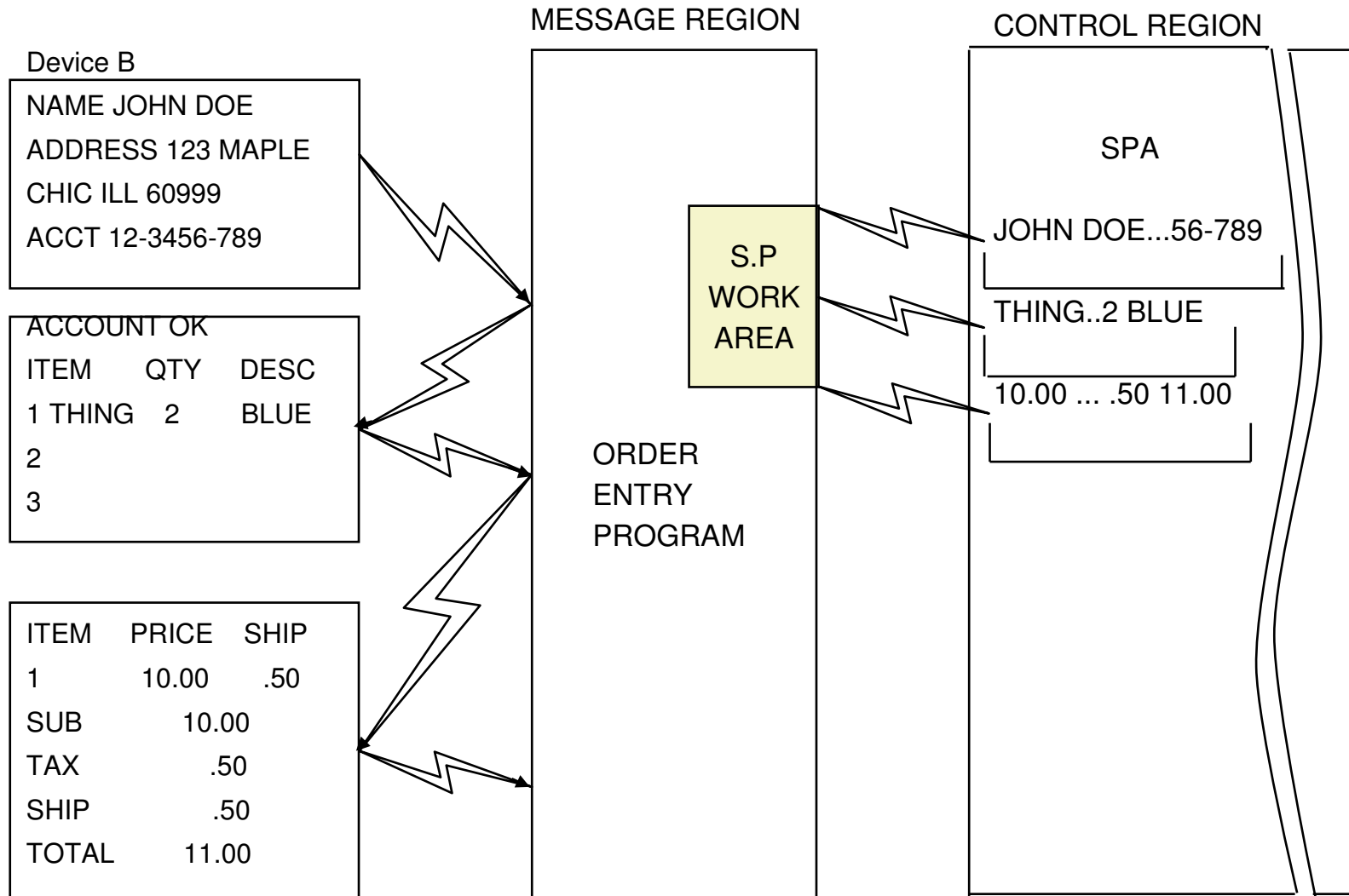
1. Data lost if screen cleared
2. More data sent over comm lines

### Solution 2

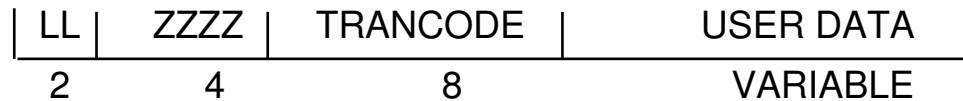
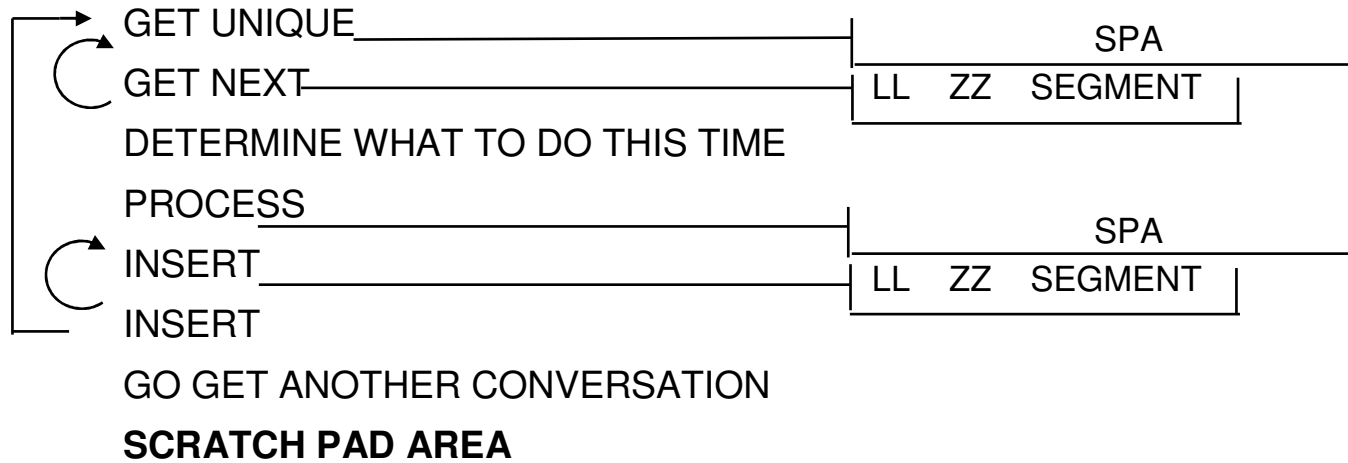


1. Complicates program logic
2. Multiple db retrievals and updates
3. Input error, pgm abend
  - Scratch temp db record
  - Backout perm db record

# Conversational Solution - Scratch Pad Area



# Conversational Call Flow



## FIRST SEGMENT OF A CONVERSATIONAL TRANSACTION



Note Application Program cannot modify the first 6 bytes of the SPA (LL and ZZZZ). IMS TM uses these fields to identify the SPA

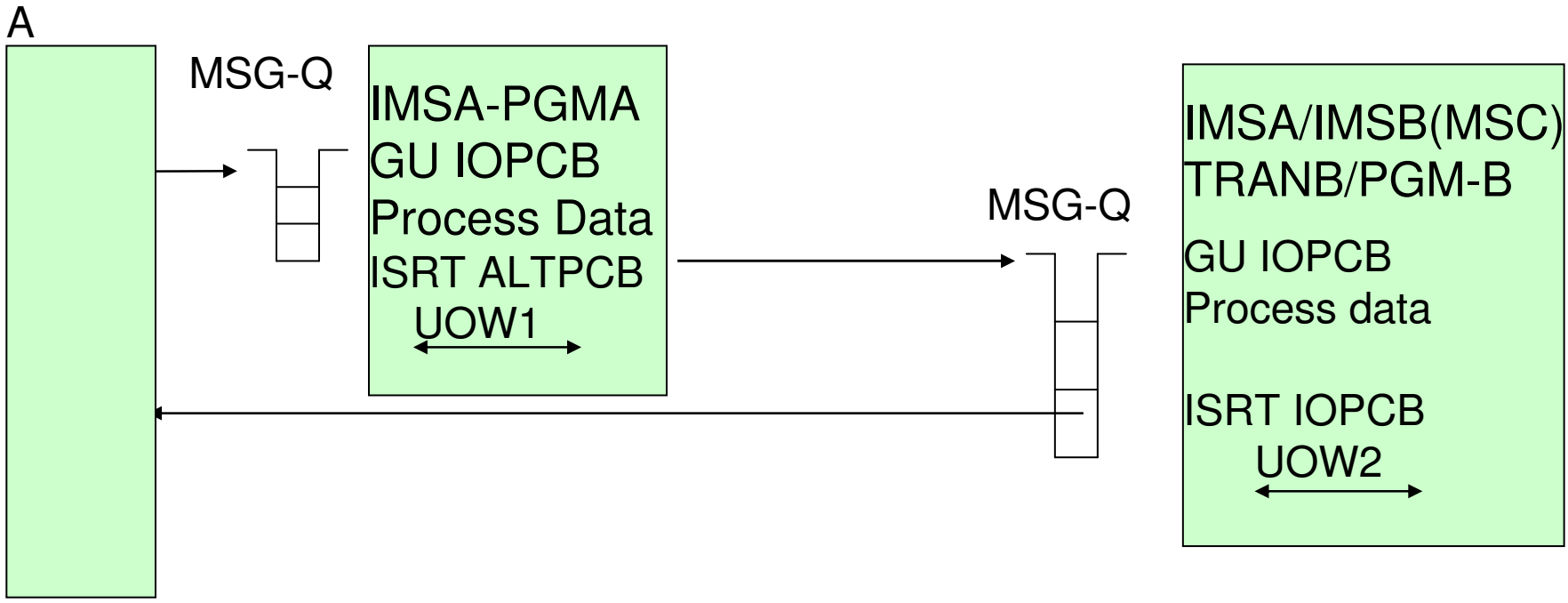
# Programming Tips

- All programs should be written to go back and get another message (GU I/O PCB) ('QC' status if none exist)
- Data base requests should be handled with the fewest number of calls
- Message processing programs should not do lengthy data base scans or updates
- Never issue file open/close operations - beware of COBOL "displays" and PL/I "put datas"
- Issue all message segment gets/inserts as consecutive calls if possible
- The PSB should be completely accurate in procopt's selected for each segment type
- Large, single segment replies take fewer calls and less overhead (check system standard for maximum size)
- Do not issue GN calls to the i/o pcb for single segment messages

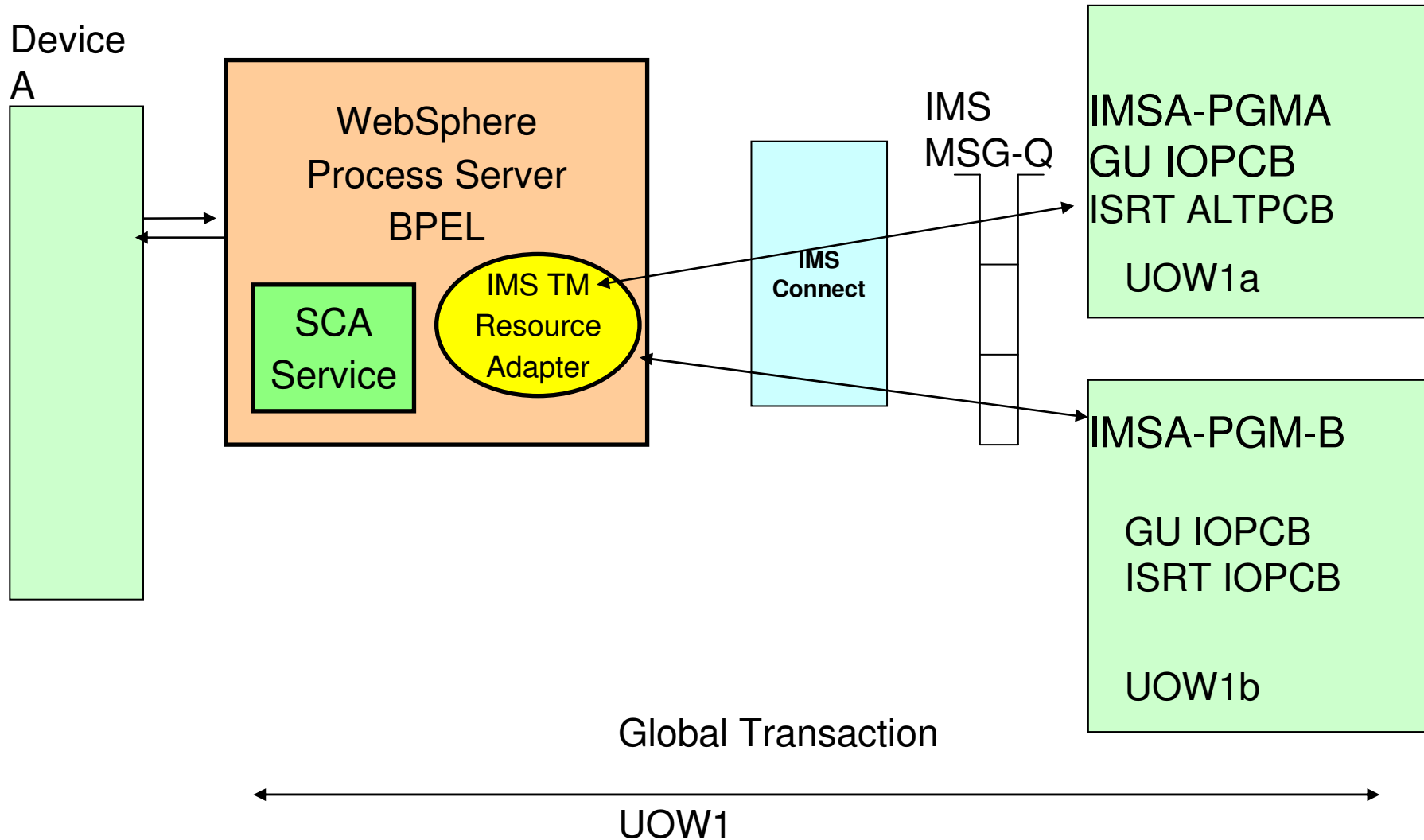
# IMS Managed Service Flow Program-to-Program Switch



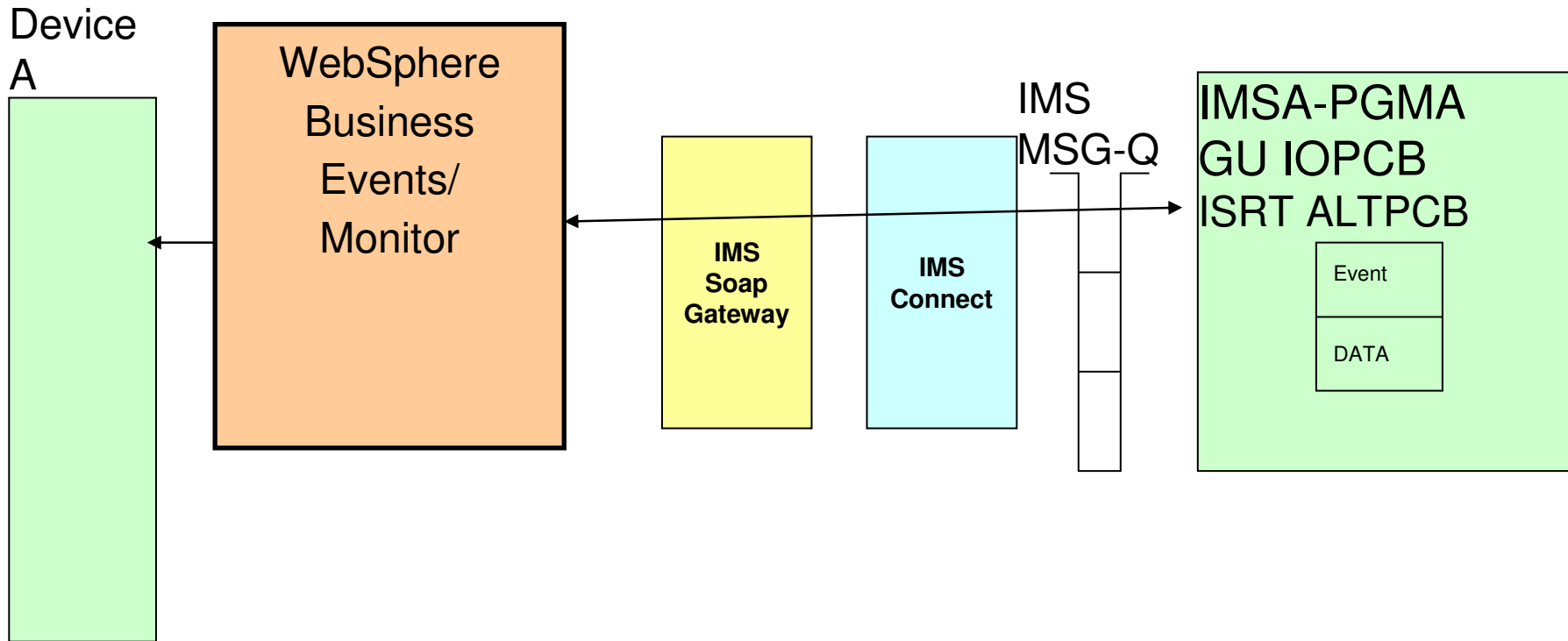
Device



# WebSphere Process Server Managed Service Flow



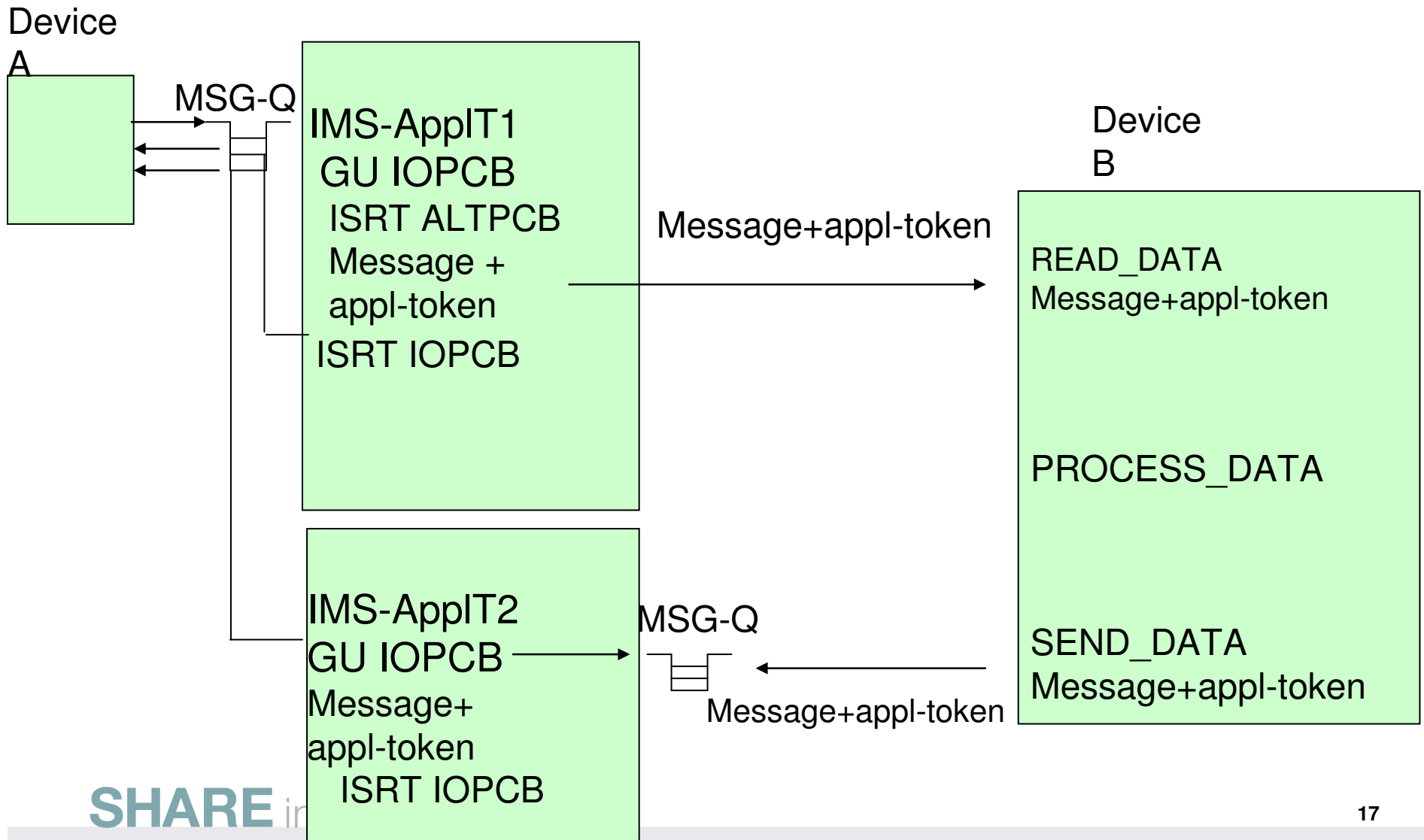
# IMS Enterprise Suite Soap Gateway Business Events



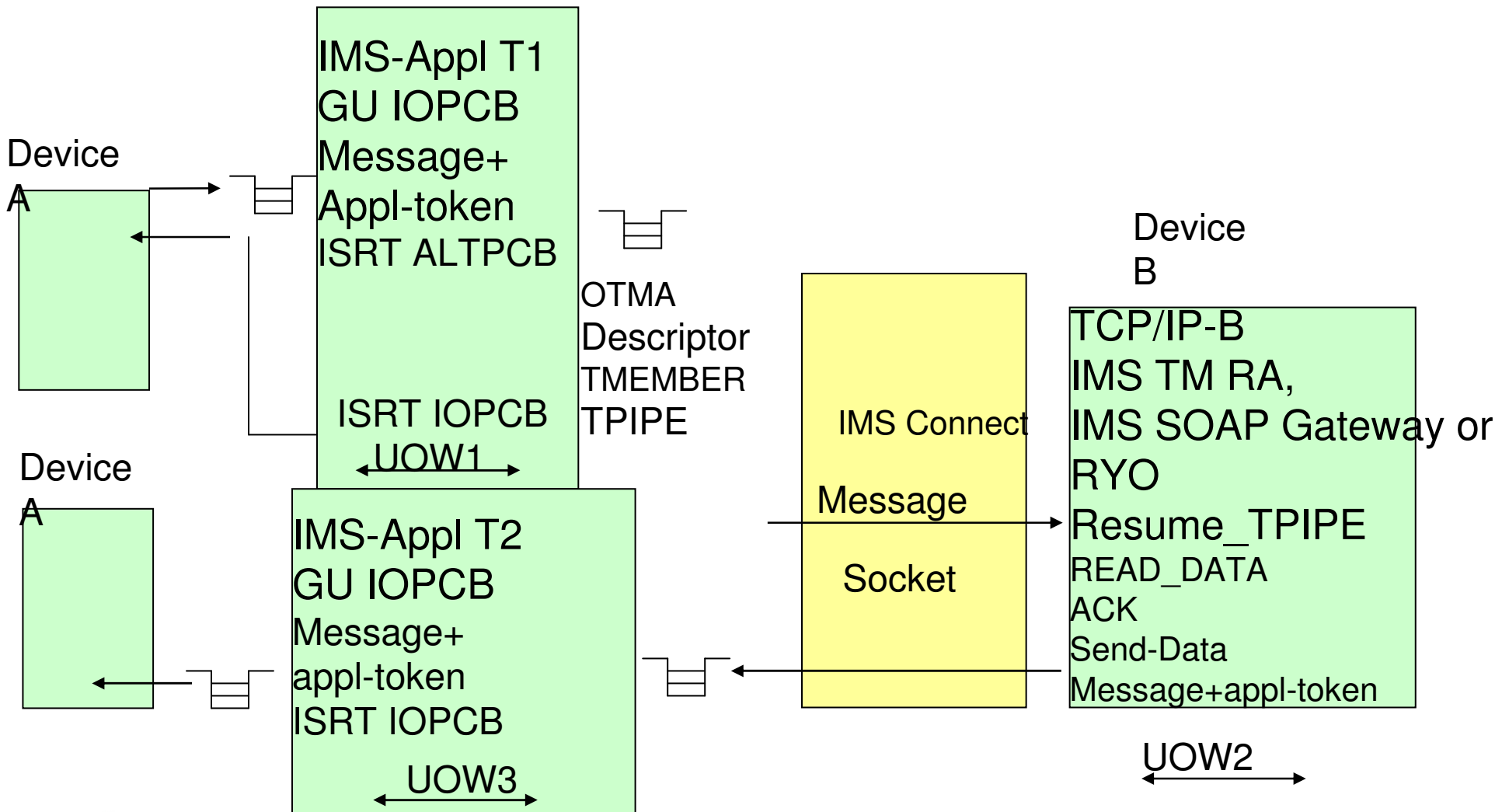
**WebSphere Business Events**  
**WebSphere Business Monitor**



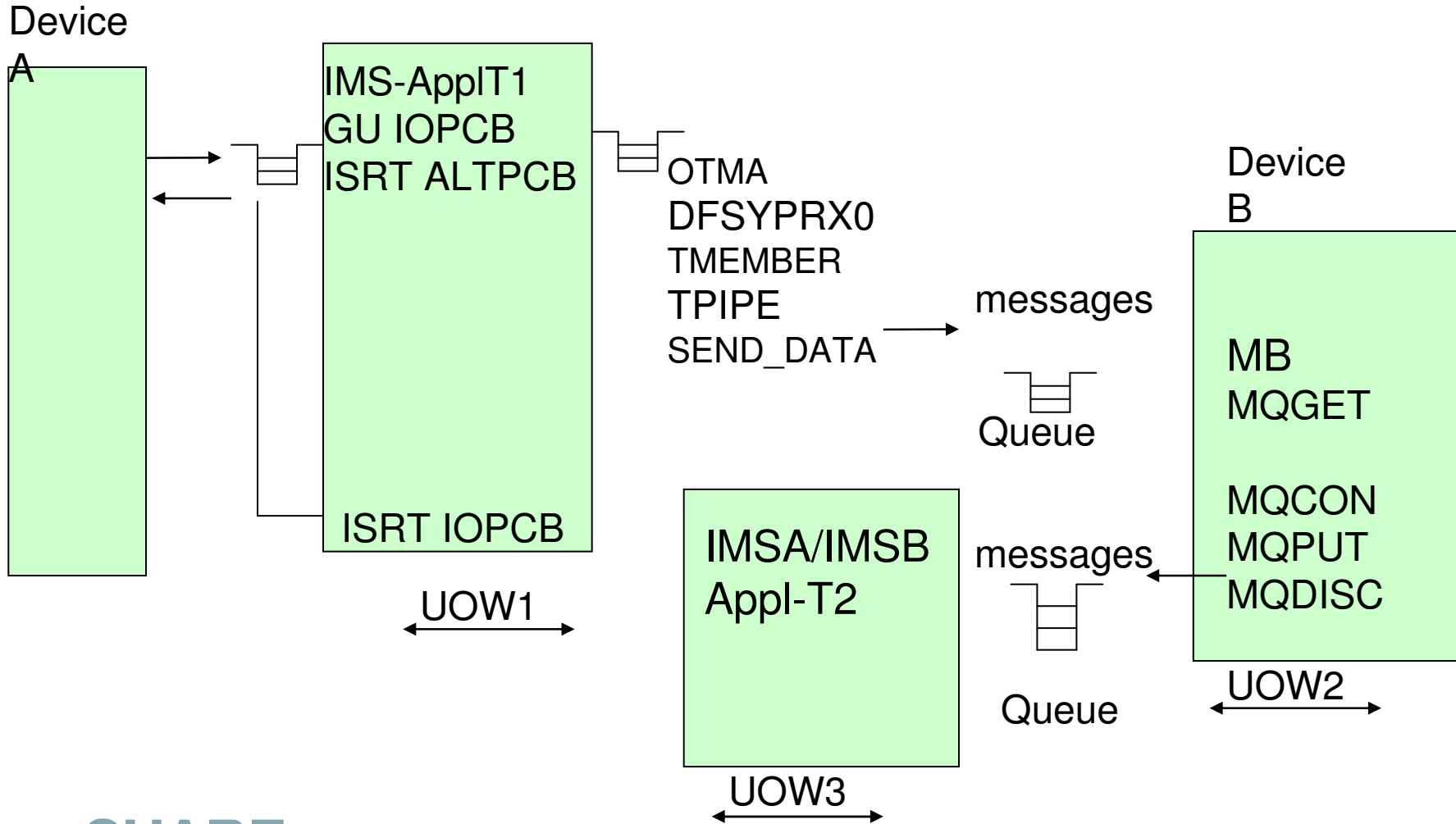
# Asynchronous Model



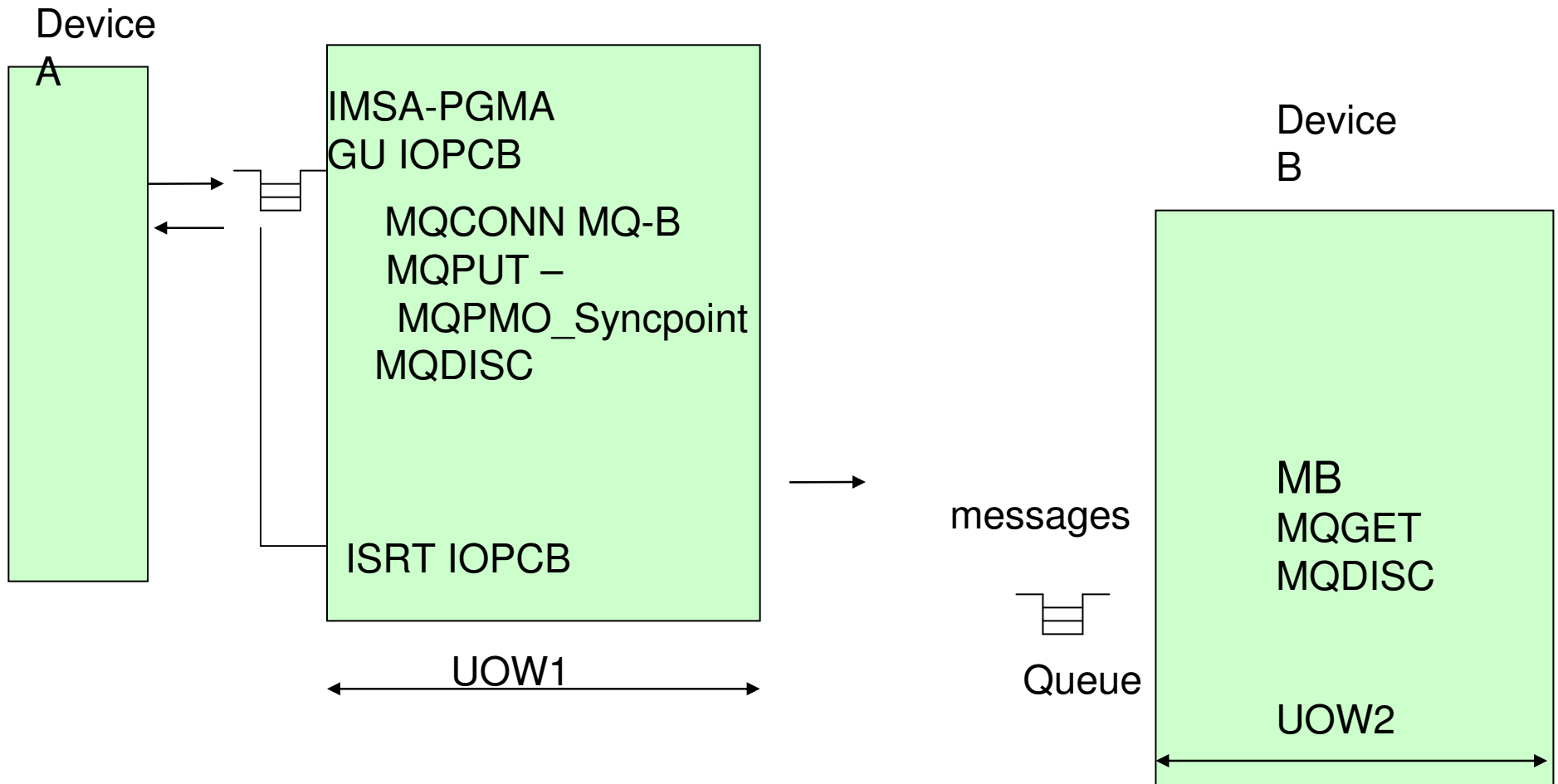
# IMS Asynchronous via OTMA IMS Connect



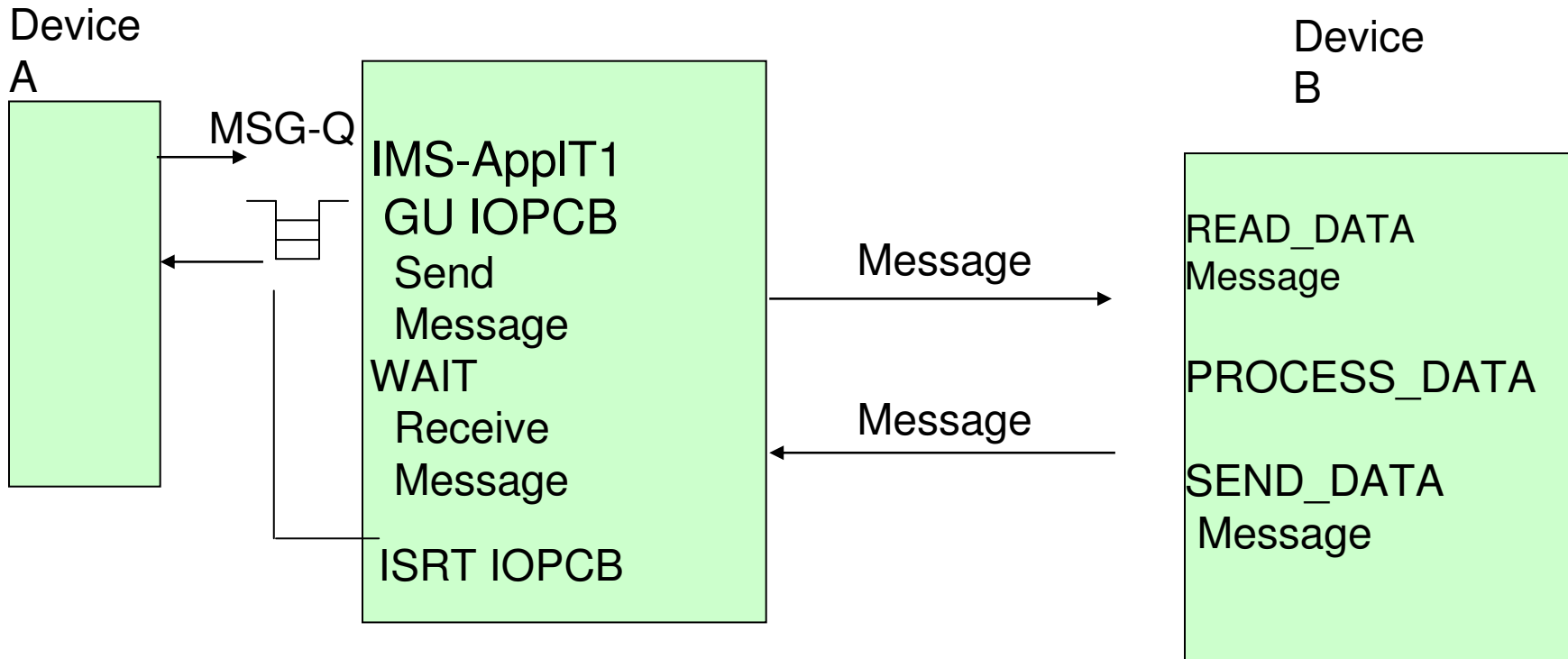
# WebSphere MQ Asynchronous via OTMA IMS Bridge



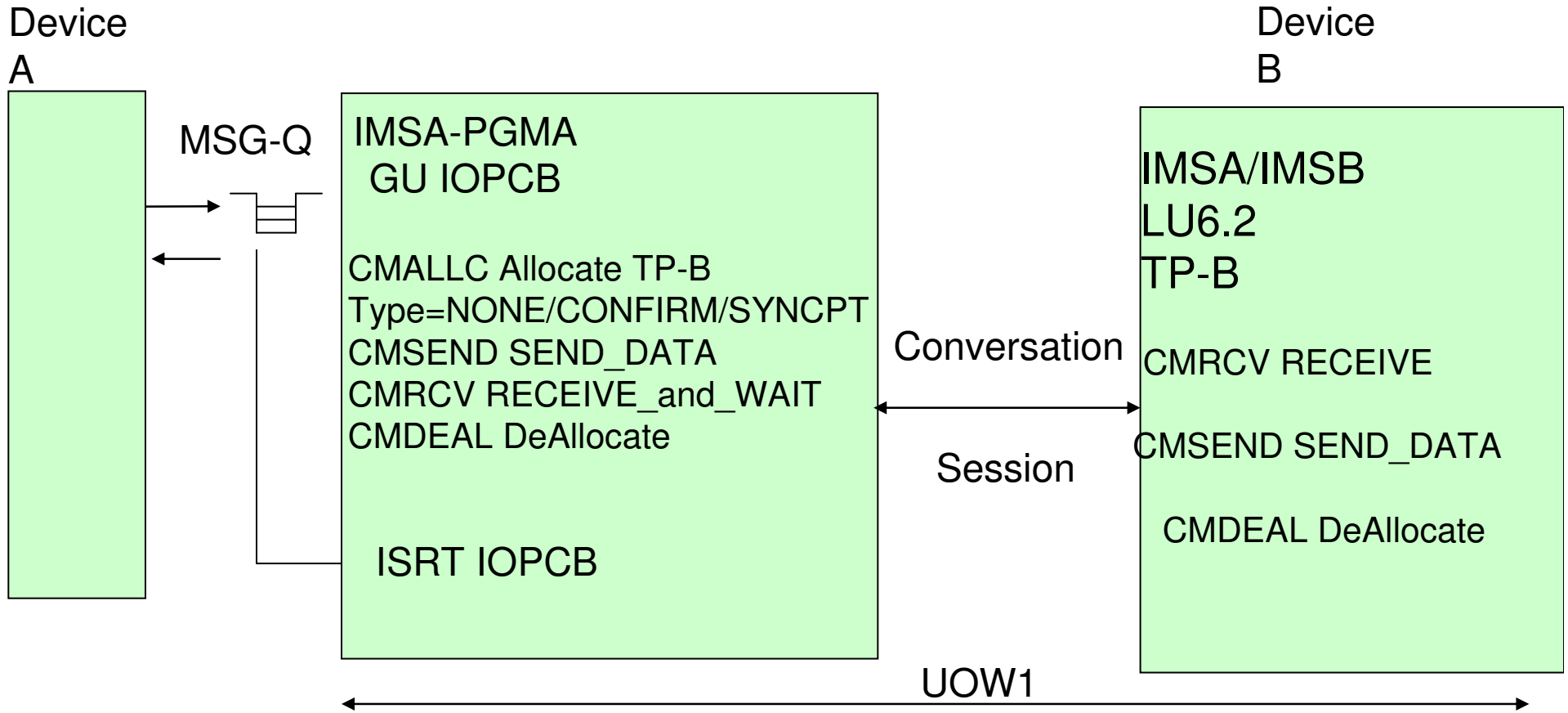
# WebSphere MQ via ESAF Asynchronous



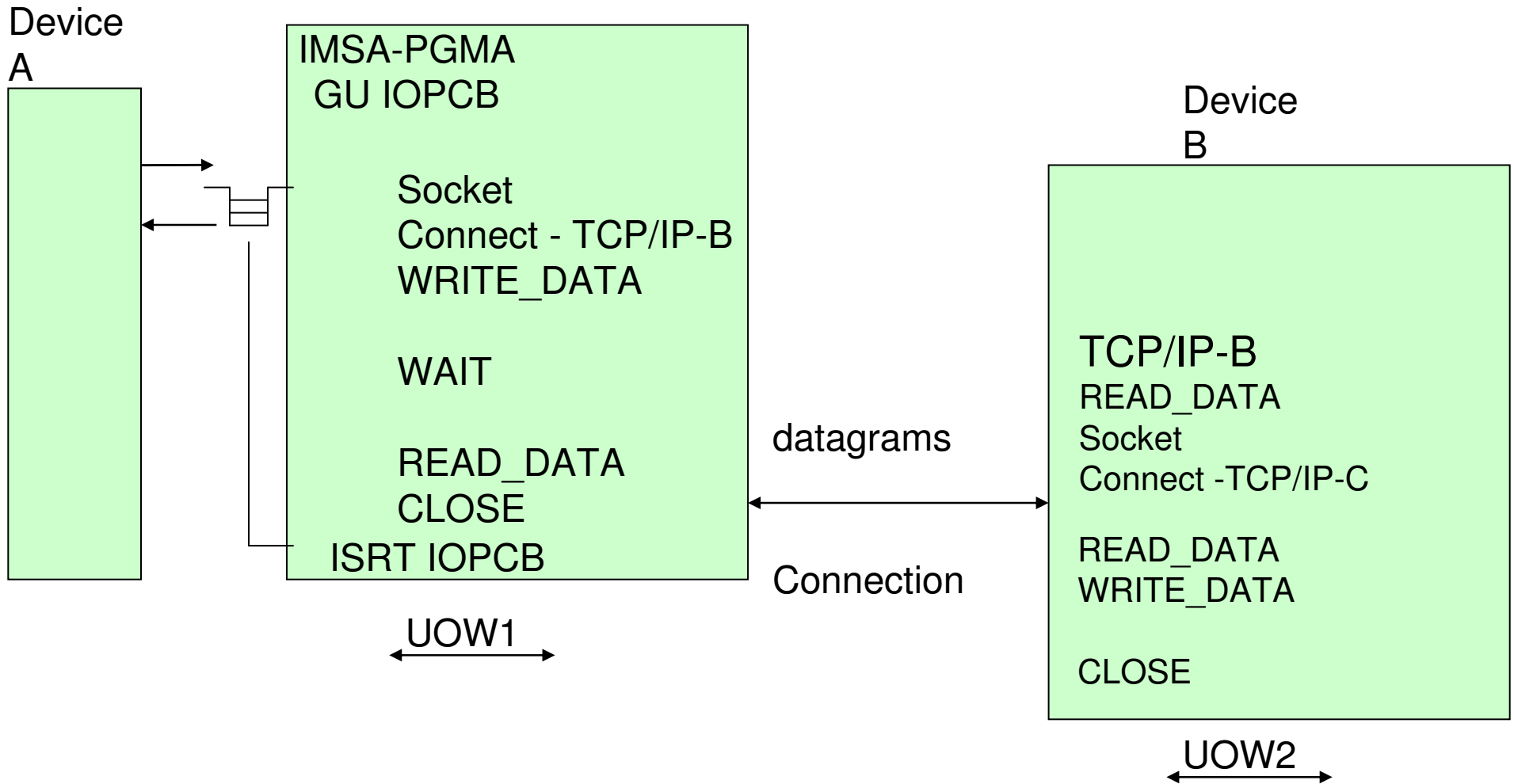
# Synchronous Model



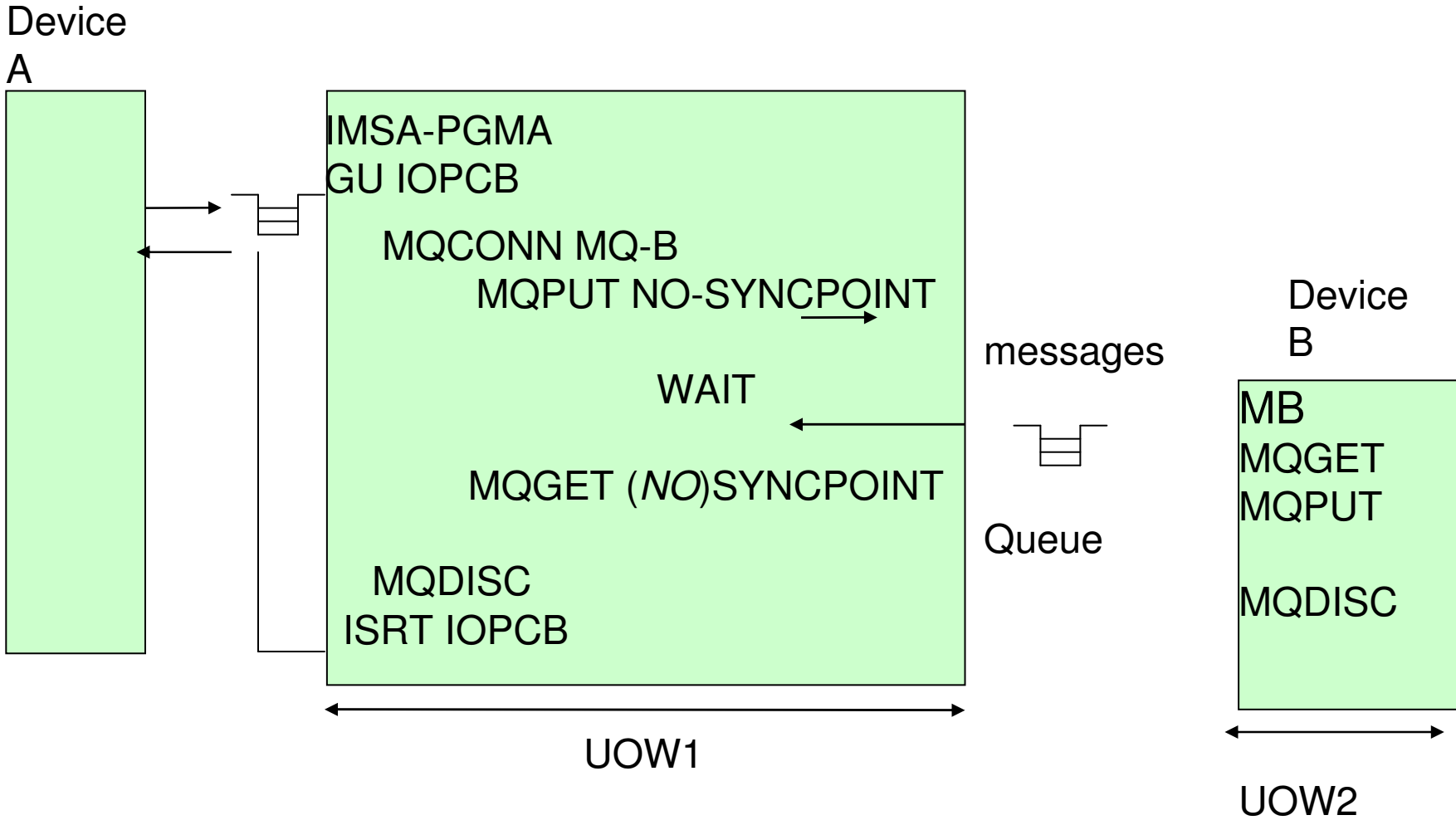
# CPI-C/APPCC Synchronous



# User Written TCP/IP Synchronous

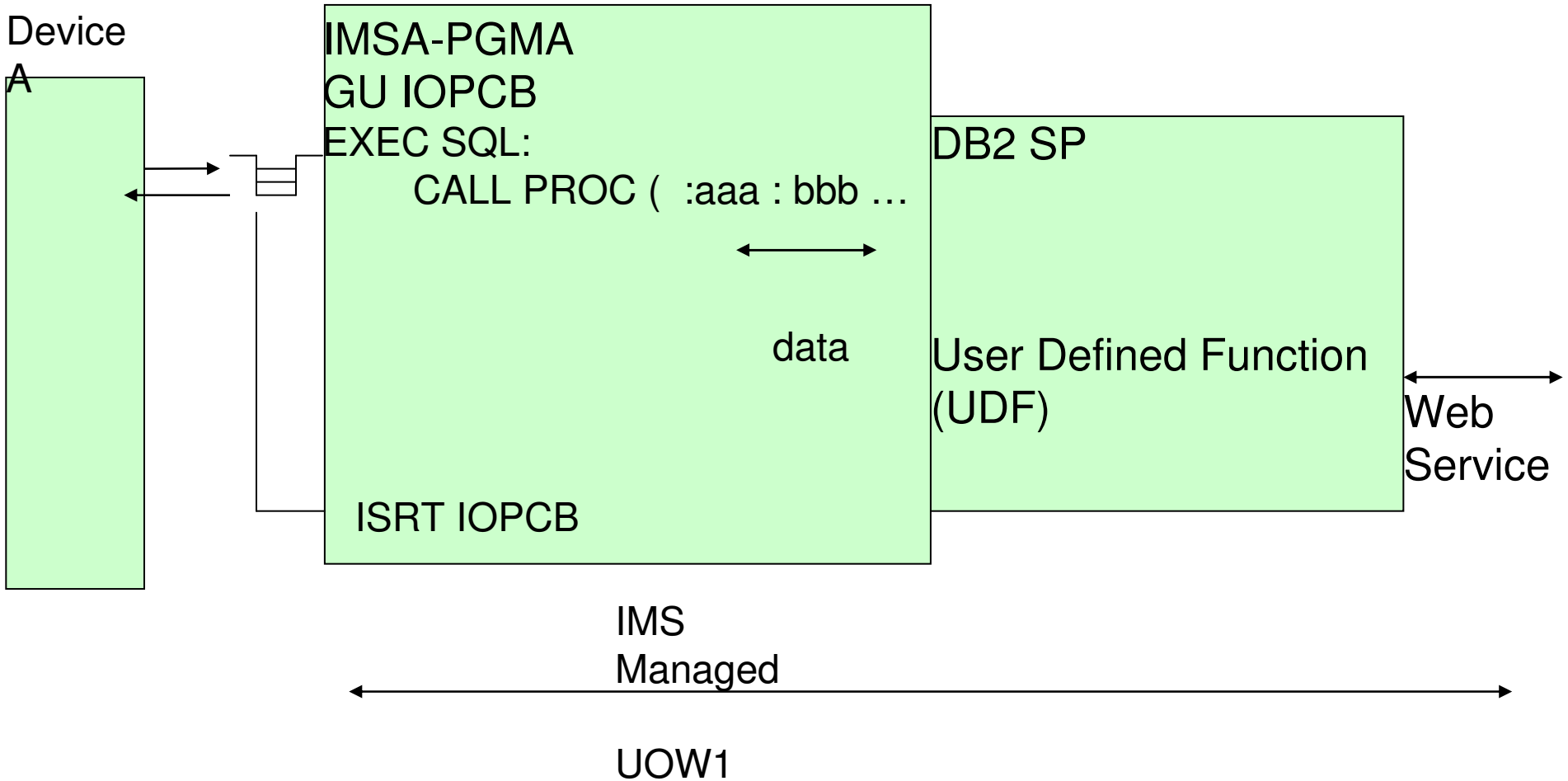


# WebSphere MQ via ESAF Synchronous

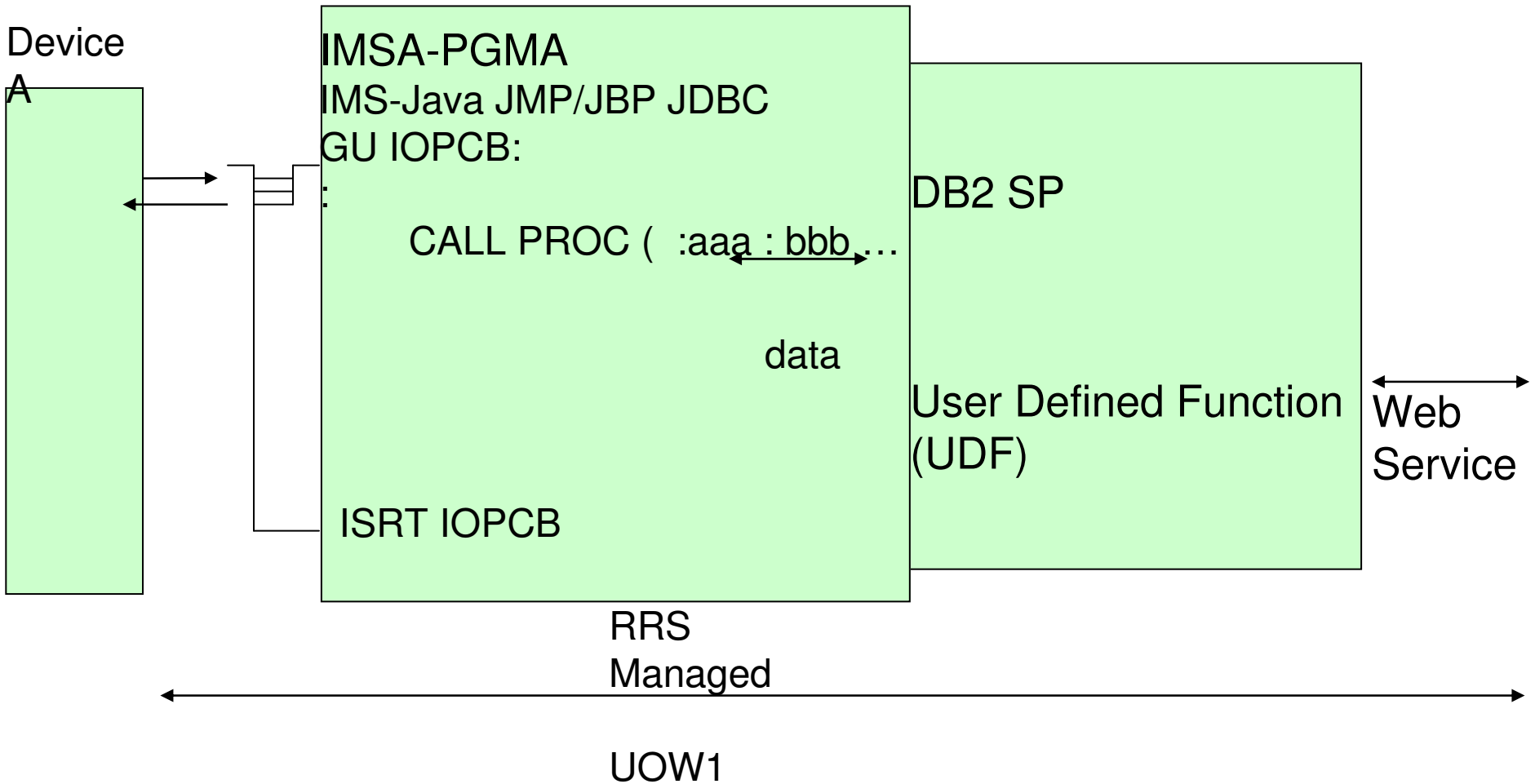




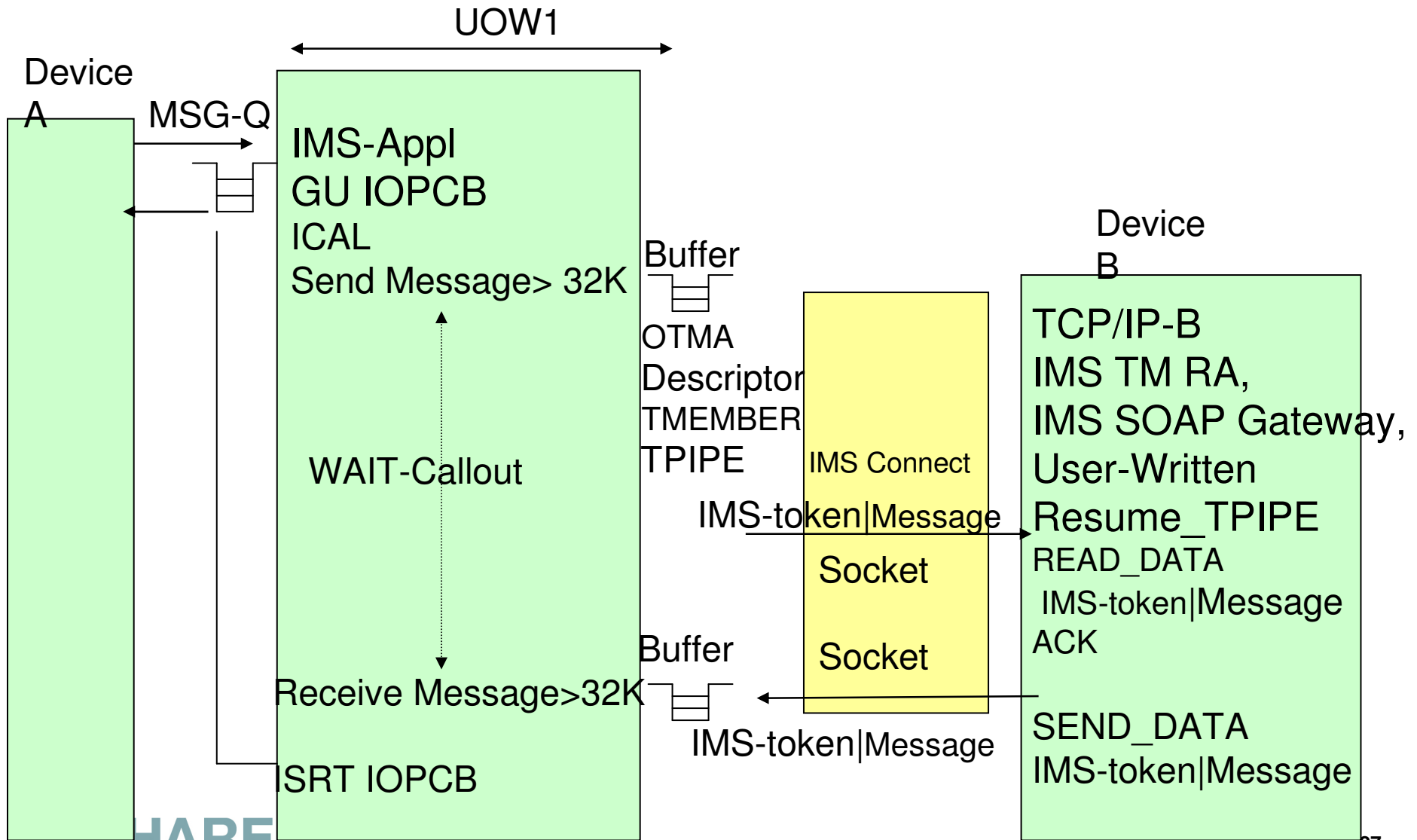
# DB2 Stored Procedure via ESAF Synchronous



# IMS Java and DB2 Attach Facility

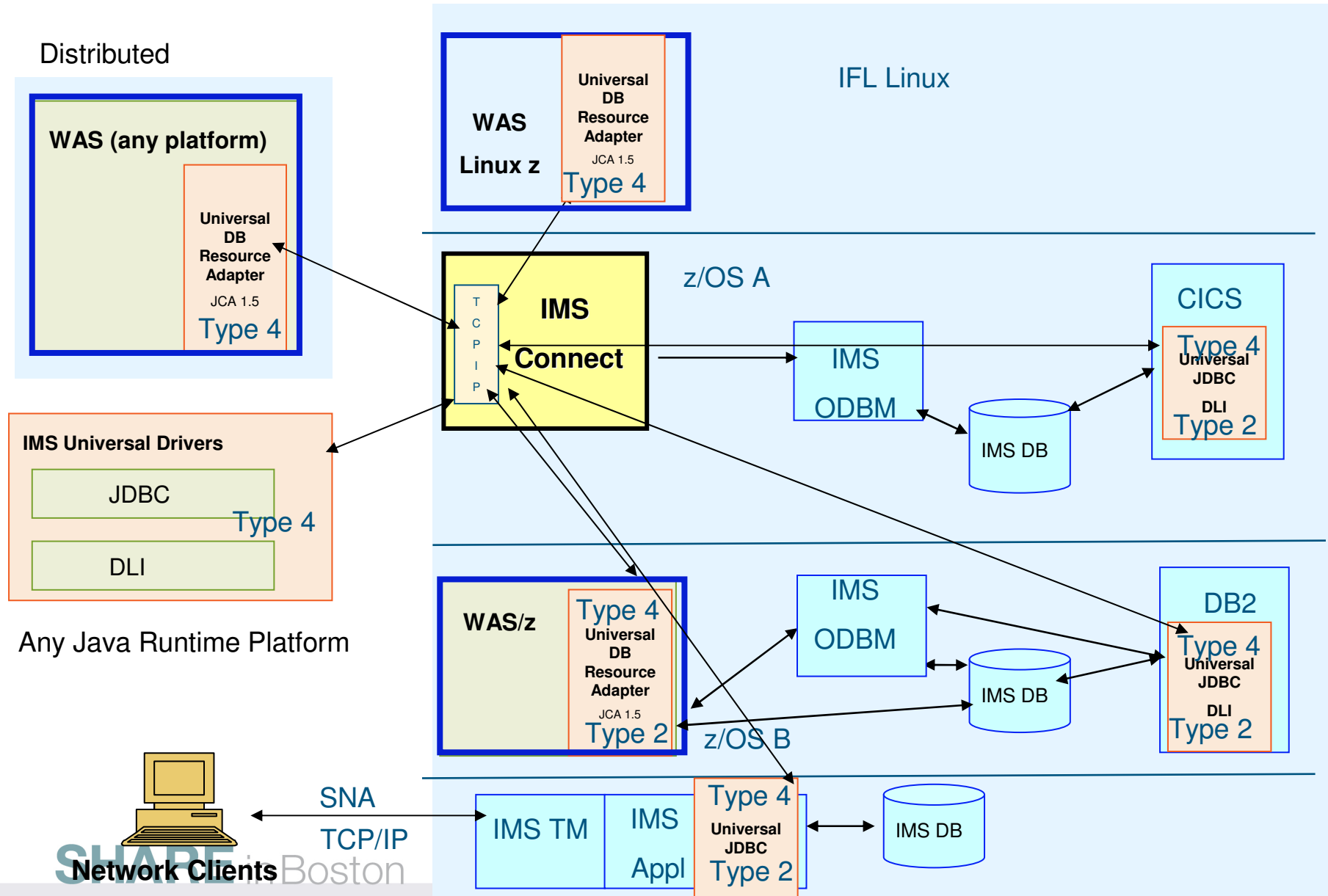
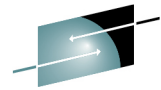


# IMS Synchronous ICAL



# Universal Drivers

System z

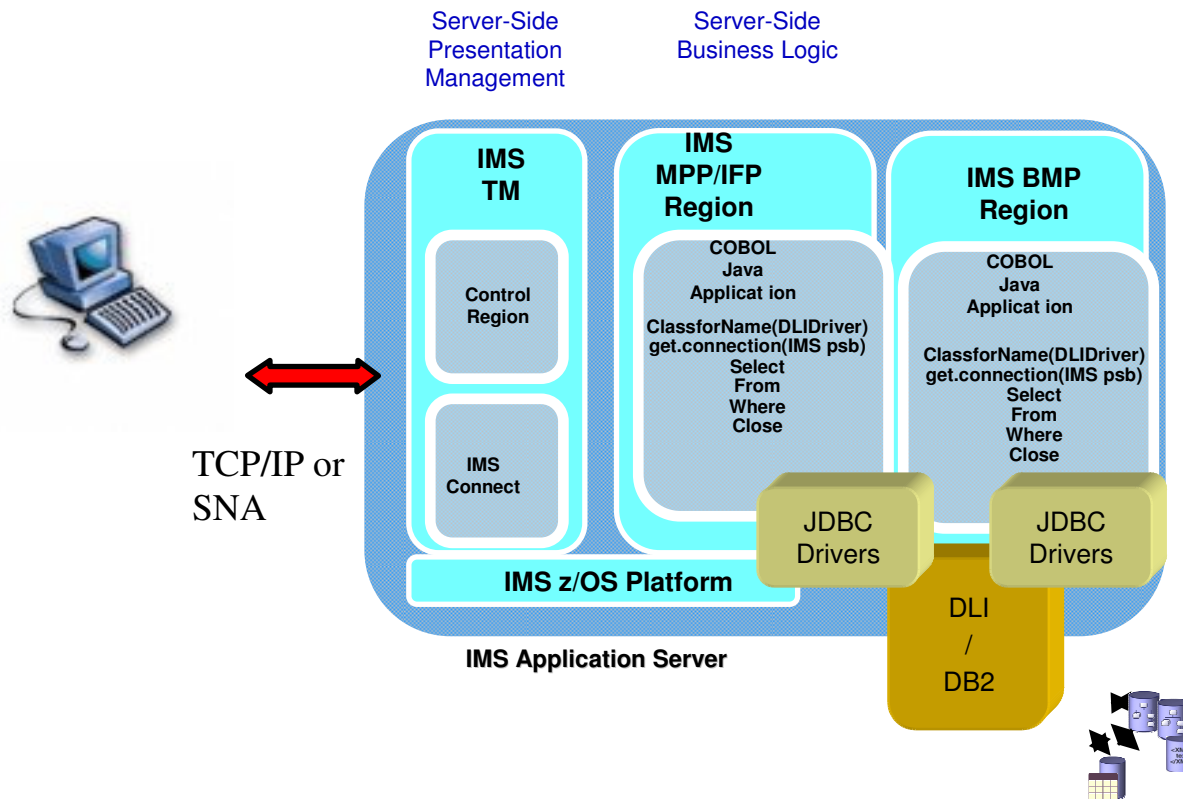


# IMS TM MPP - JVM

IMS 10 Apar PK82214

LE Apar PK99010

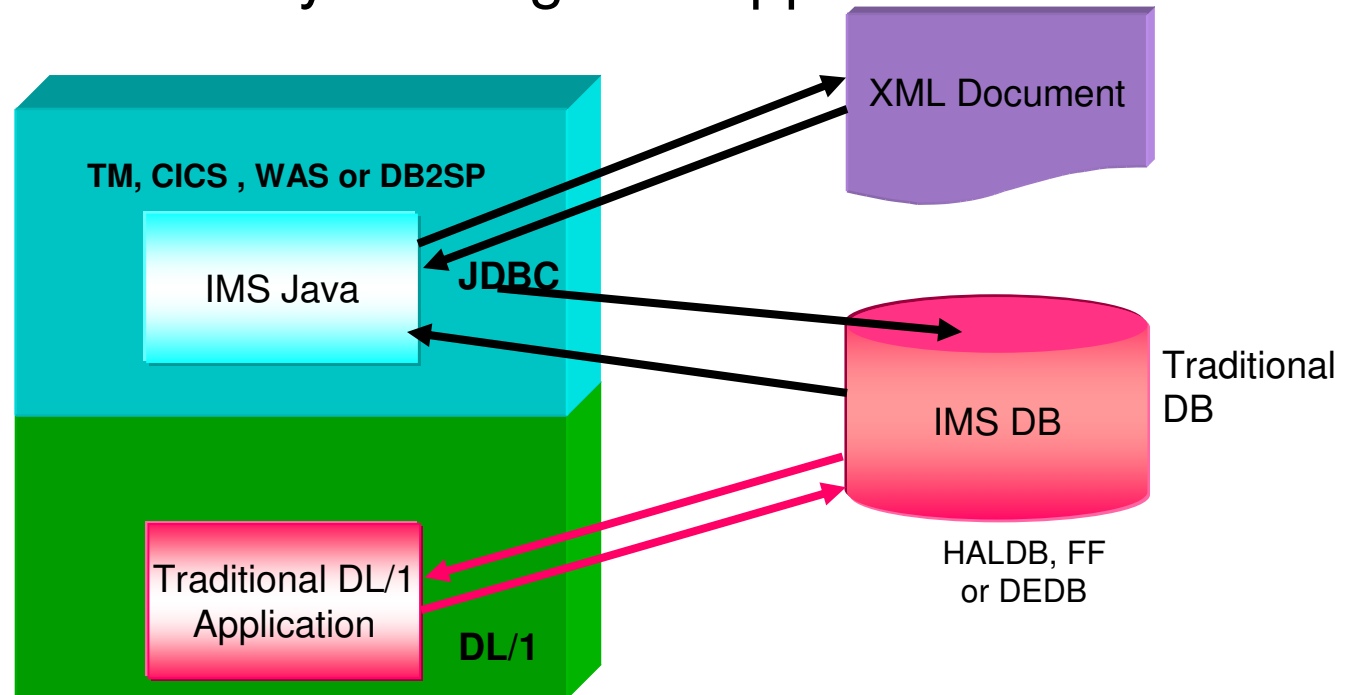
DB2 Apar PK93123 – use ESAF no RRS



# XML DB Highlights - Decomposed data

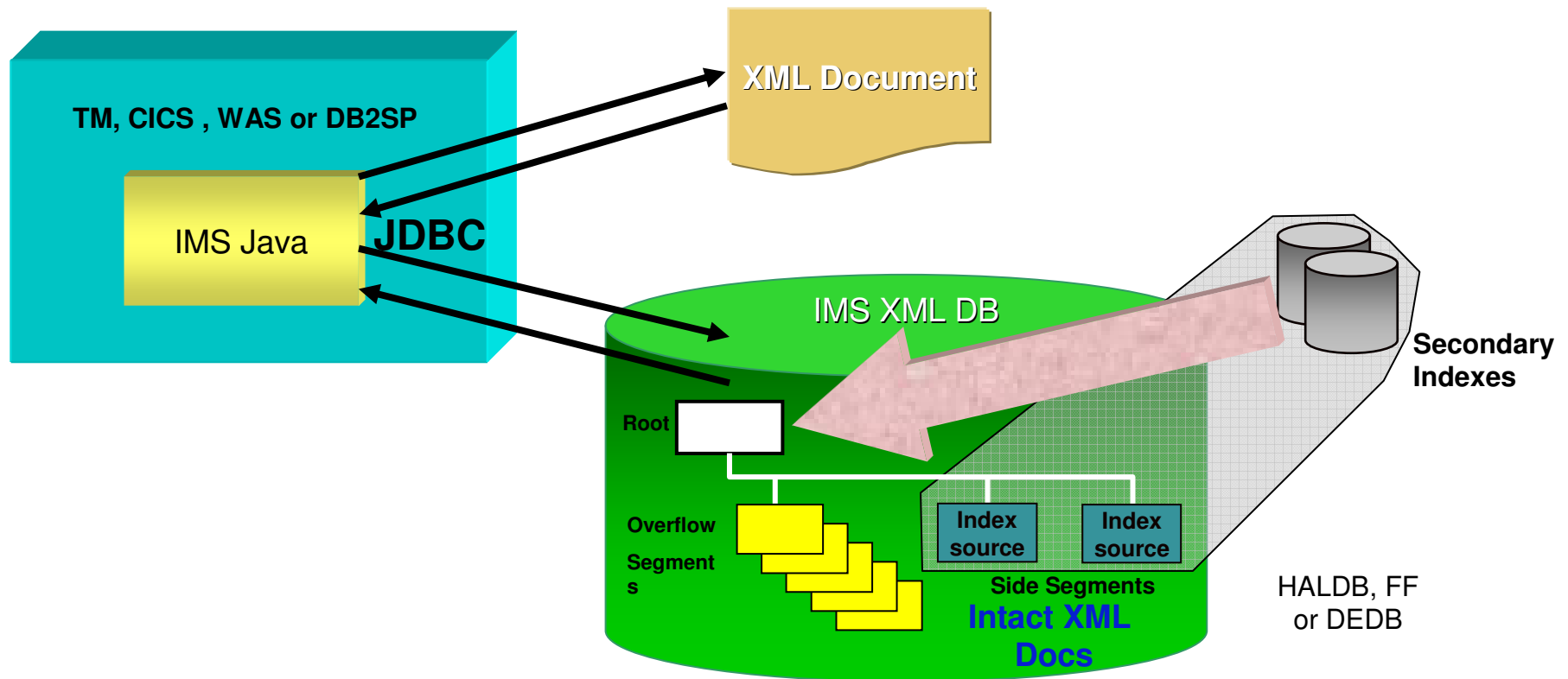


- Retrieve - Compose XML document from **any** existing traditional database
- Insert - Decompose XML docs back into same DB
- Same data can be read by existing IMS applications



# XML DB Highlights - Intact Data

- Insert/Retrieve/Delete new XML documents **INTACT** in new IMS databases
- **Intact data** is not expected to be understood by other IMS applications
  - XML Documents span IMS segments
  - Stored in Unicode



# IMS Application Programming techniques



- Summary
  - IMS Application Programming provides simple device/data independent model
    - For online processing
    - For Batch Processing