



IBM Software Group

IMS31 APPC/OTMA Shared Message Queue Enablement

Richard Schneider
IMS developer



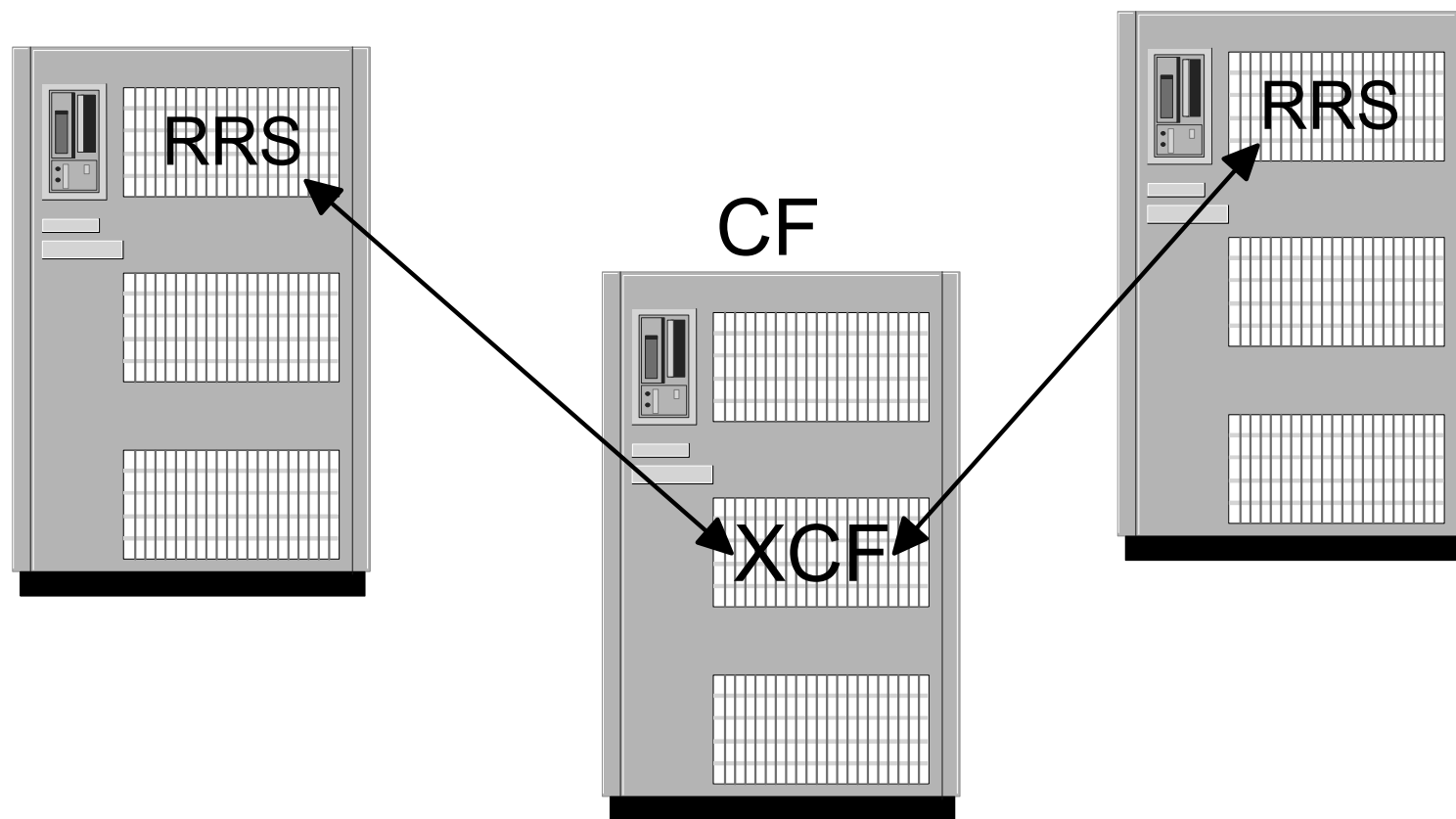
ON DEMAND BUSINESS™

History

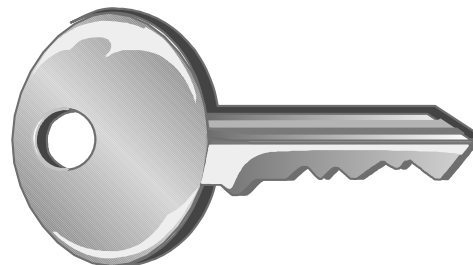
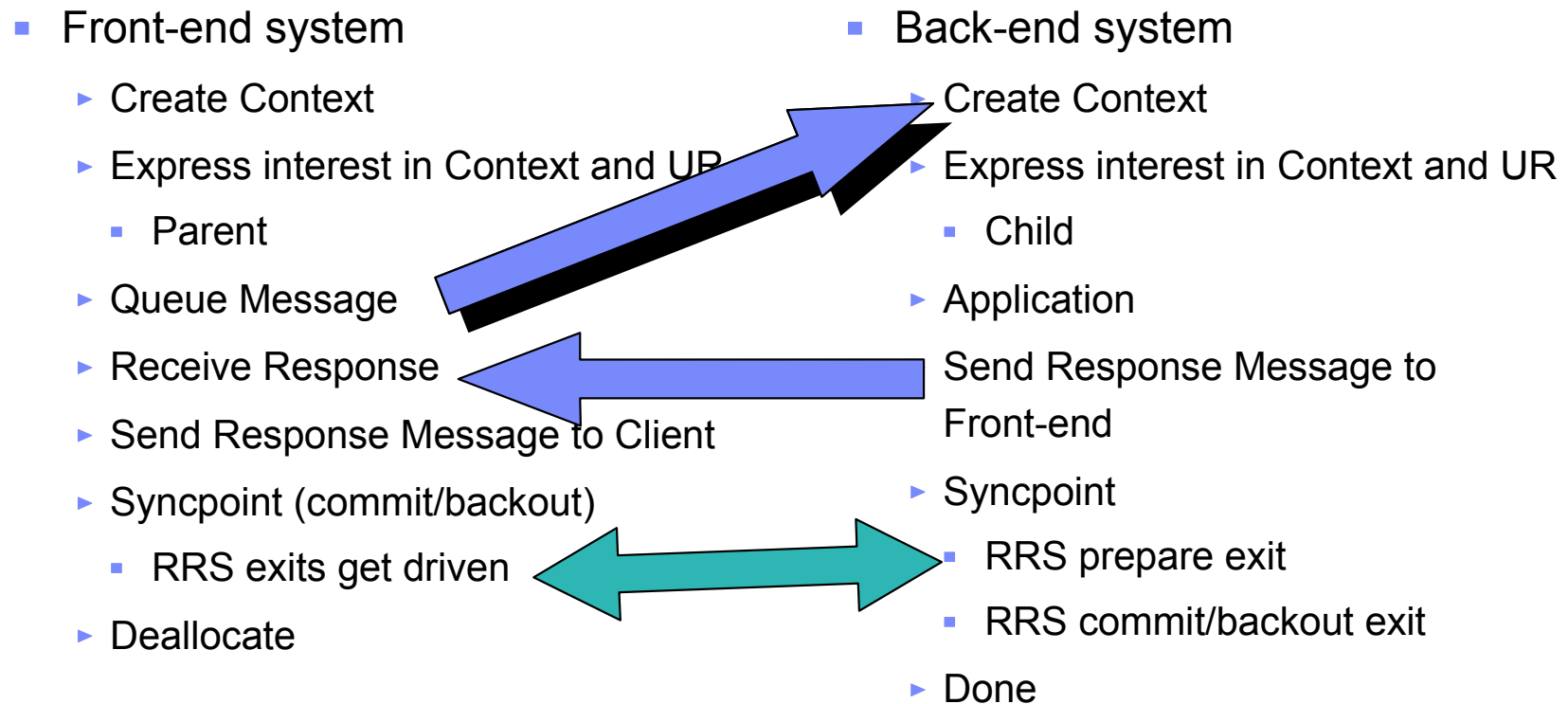
- ✦ v4.1 APPC
- ✦ V5.1 OTMA
- ✦ V6.1 Shared Queues and distributed syncpoint
- ✦ V7.1 Asynchronous SMQ support
- ✦ V8.1 Synchronous SMQ support SPE



RRS MultiSystem Cascaded Transactions

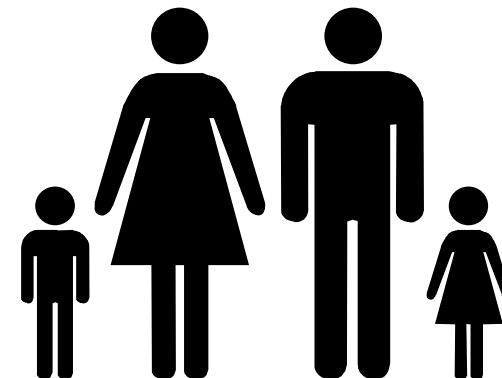
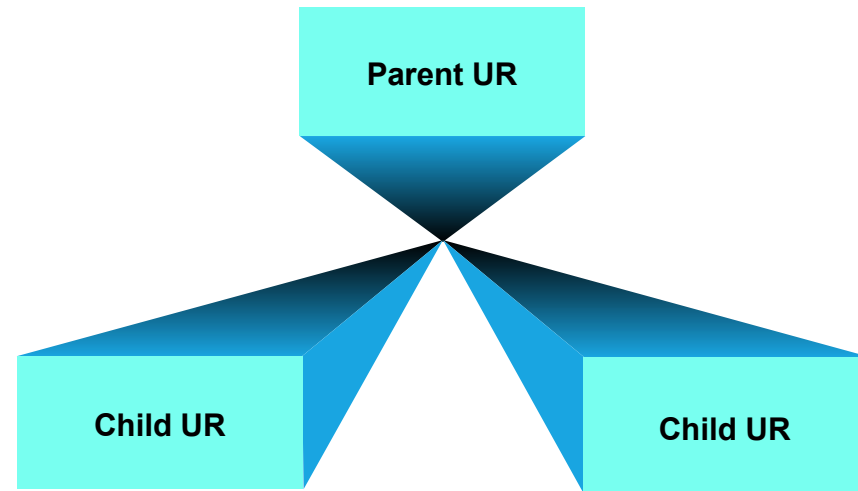


RRS MultiSystem Cascaded Transactions



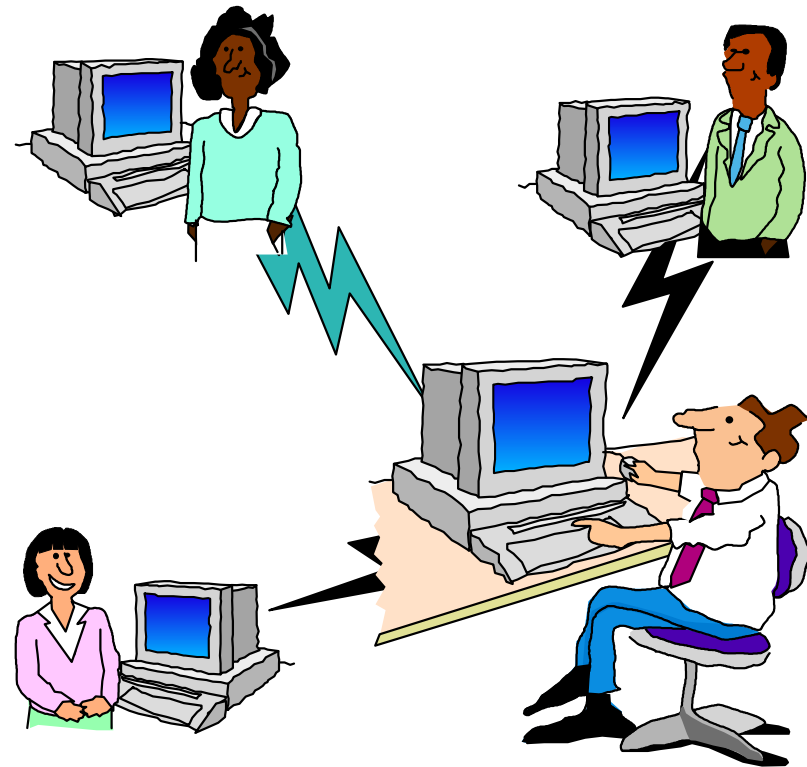
RRS MultiSystem Cascaded Transactions

- Front-end and Back-end creates context
- Front-end express interest as parent
 - ▶ Is the only one allowed to issue commit
- Back-end express interest as child
 - ▶ Could issue backout but not commit
- Two phase commit between back-end and front-end

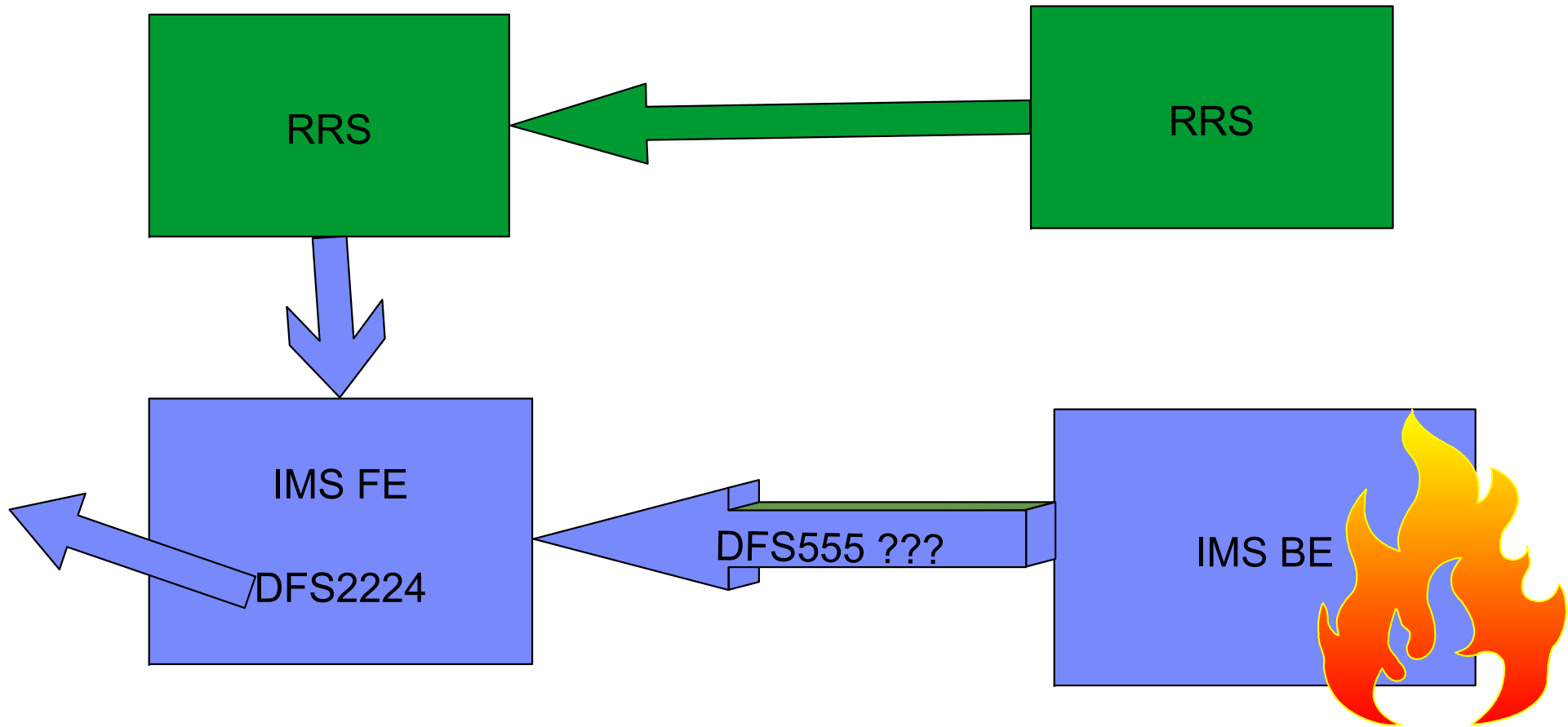


Subordinated Failure Exit

- New function in RRS
- At IMS restart time we tell RRS that we want to use Subordinated Failure Exit
- The exit gets driven at the front-end system whenever a child process ends without committing.
- IMS will issue DFS2224 message to the client.



Subordinated failure exit

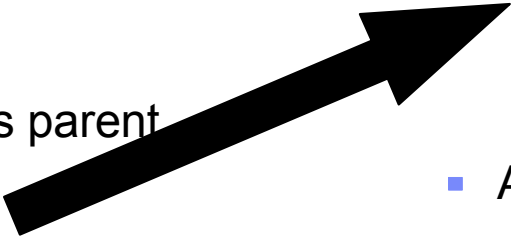


IMS processing *** Front-end not equal Back-end

- | | | |
|---|--|--|
| <ul style="list-style-type: none"> ■ Front-end IMS control region ■ Receives Message ■ Creates Context ■ Express interest as parent ■ Queues Message ■ Receives response message ■ Sends response message to child ■ Syncpoint - Commit <ul style="list-style-type: none"> ▶ prepare exit gets driven and returns forget. | | <ul style="list-style-type: none"> ■ Back-end IMS dependent region ■ GU service <ul style="list-style-type: none"> ▶ Creates Security ▶ Creates Context ▶ Express interest in CNTX and UR as child ■ Application ■ Syncpoint <ul style="list-style-type: none"> ▶ Sends response message to front-end ▶ Set side info ▶ Wait for Front-end <ul style="list-style-type: none"> ■ prepare exit gets driven ■ commit or back-out exit gets driven ▶ Commit or backout |
| | | |



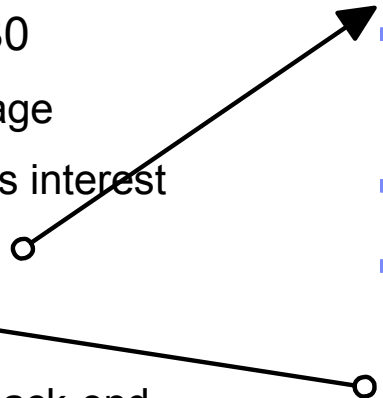
IMS processing *** Front-end = Back-end

- Front-end IMS control region
 - Receives Message
 - Creates Context
 - Express interest as parent
 - Queues Message
- 
- Front-end IMS dependent region
 - GU service
 - ▶ Creates Security
 - ▶ no RRS calls
 - Application
 - Syncpoint
 - ▶ Sends response message to client
 - ▶ RRS commit to clean-up RRS control blocks

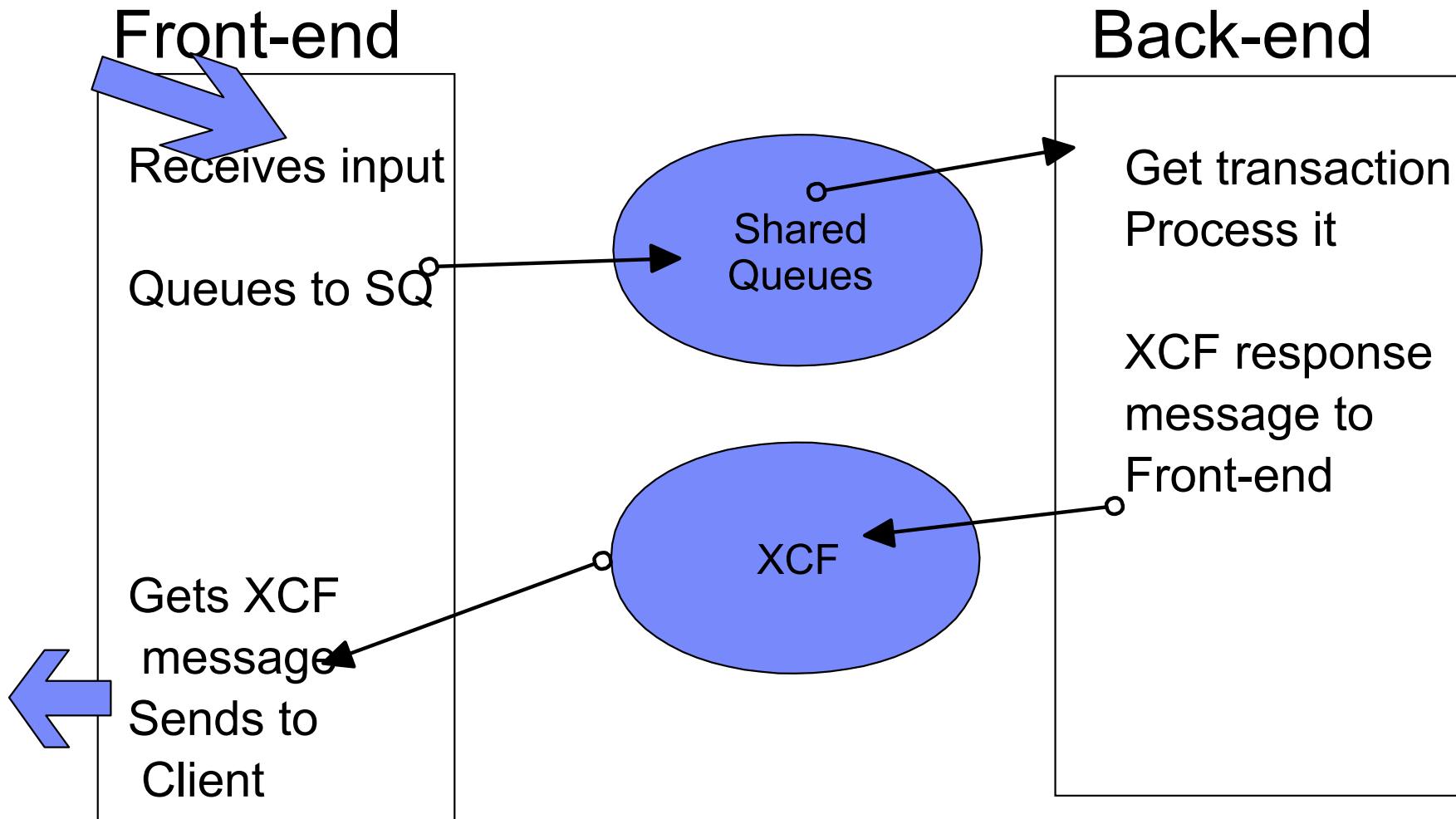


Messages < 62K and non-conversational

- DFSRLM10/DFSYTIB0
 - ▶ Receives input message
 - ▶ create context/express interest
 - ▶ queues transaction
- DFSAOSW0
 - ▶ Gets message from Back-end
 - ▶ Validates (Y)TIB
 - ▶ Posts DFSRLM10/DFSYTIB0
- DFSRLM10/DFSYTIB0
 - ▶ Sends message to client
 - ▶ Issues commit/backou
 - ▶ Frees (Y)TIB
- DFSTMAS0
 - ▶ Creates context/express interest
- Application starts
- Syncpoint
 - ▶ DFSSLUM0/DFSYSLM0
 - Sends complete message to Front-end using XCF
 - ▶ DFSTMS00
 - Waits for commit/backout exit to post
- Syncpoint ends

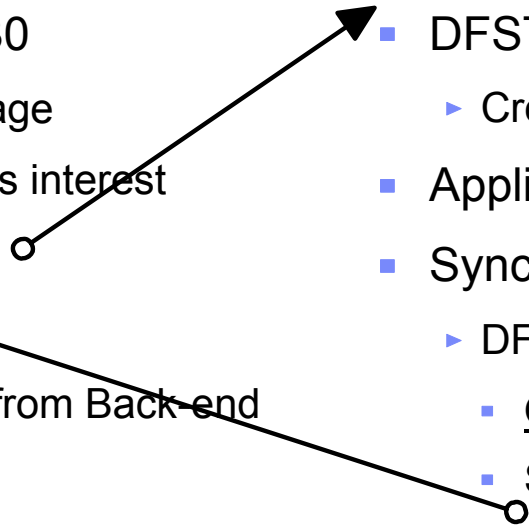


Messages < 62K and non-conversational

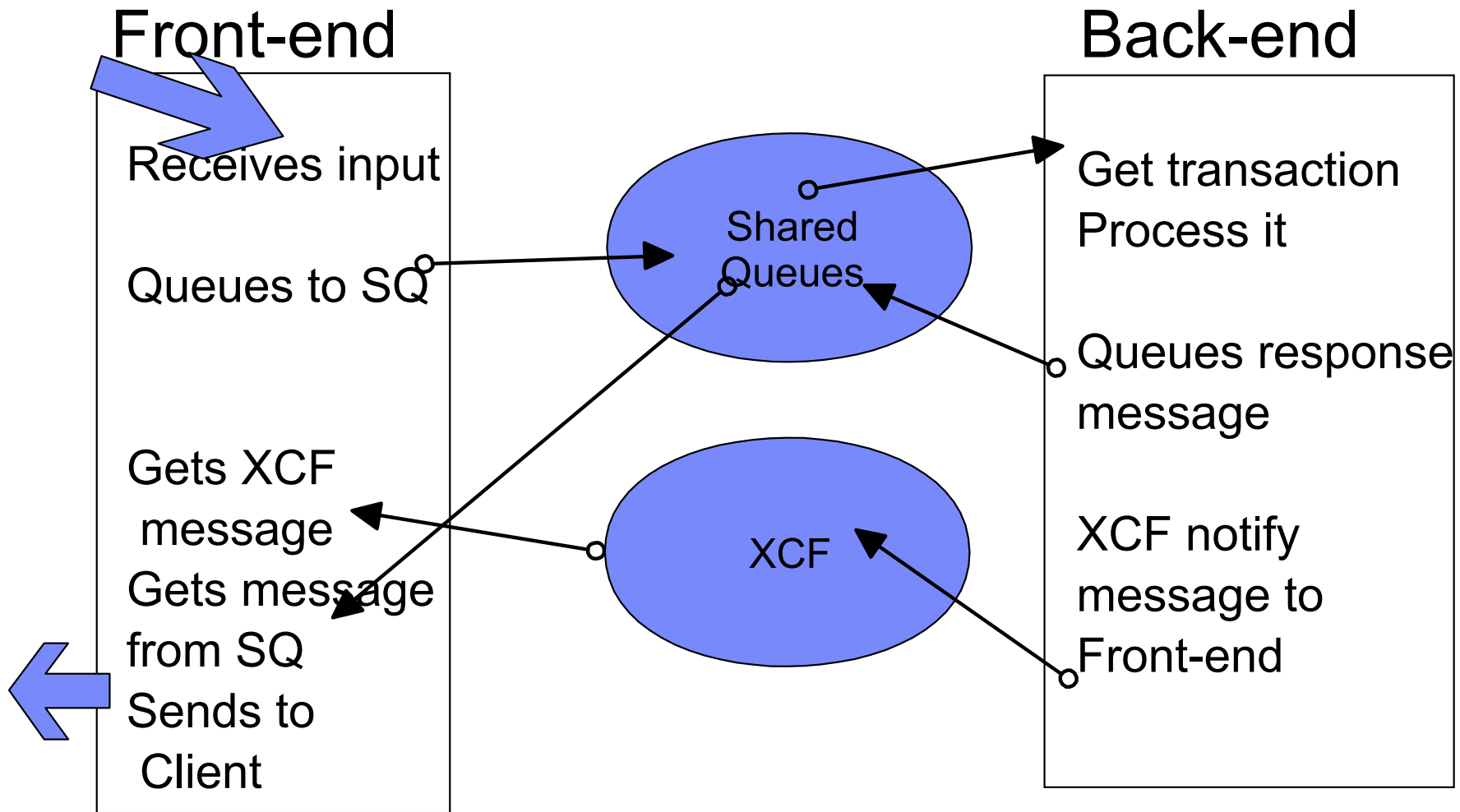


Messages > 62K or conversational

- DFSRLM10/DFSYTIB0
 - ▶ Receives input message
 - ▶ create context/express interest
 - ▶ queues transaction
- DFSAOSW0
 - ▶ Gets notify message from Back-end
 - ▶ Validates (Y)TIB
 - ▶ Posts DFSRLM10/DFSYTIB0
- DFSRLM10/DFSYTIB0
 - ▶ Gets message from SQ
 - ▶ Sends message to client
 - ▶ Issues commit/backou
 - ▶ Frees (Y)TIB
- DFSTMAS0
 - ▶ Creates context/express interest
- Application starts
- Syncpoint
 - ▶ DFSSLUM0/DFSYSLM0
 - Queues response message to SQ
 - Sends notify message to Front-end using XCF
 - ▶ DFSTMS00
 - Waits for commit/backout exit to post
- Syncpoint ends



Messages > 62K or conversational



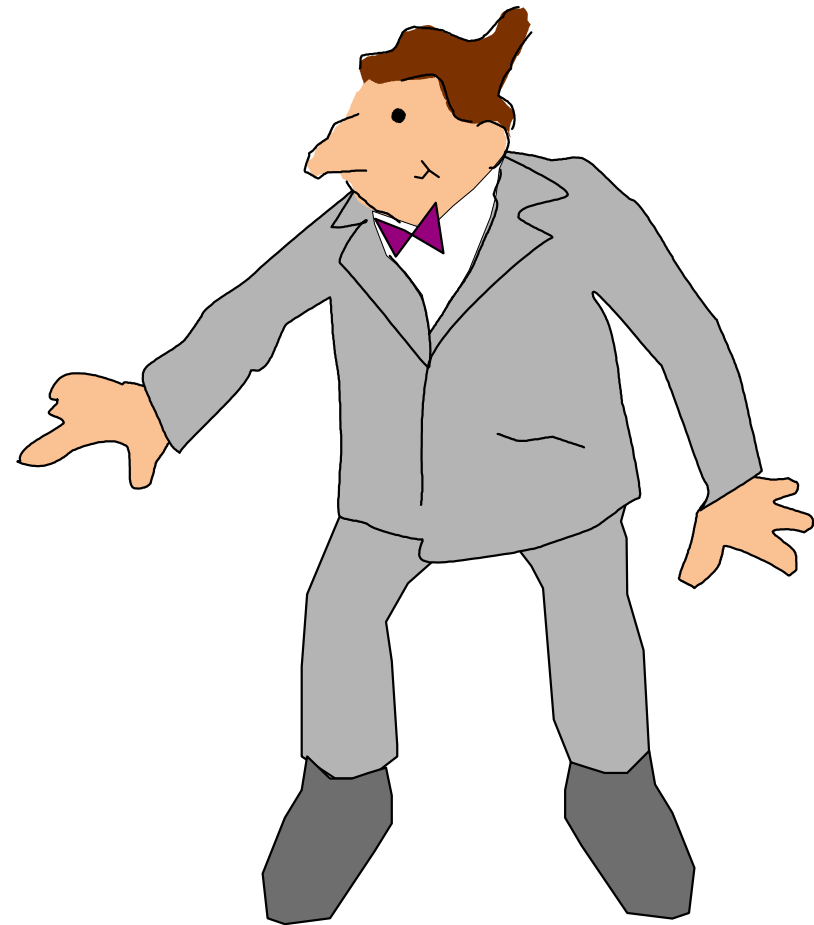
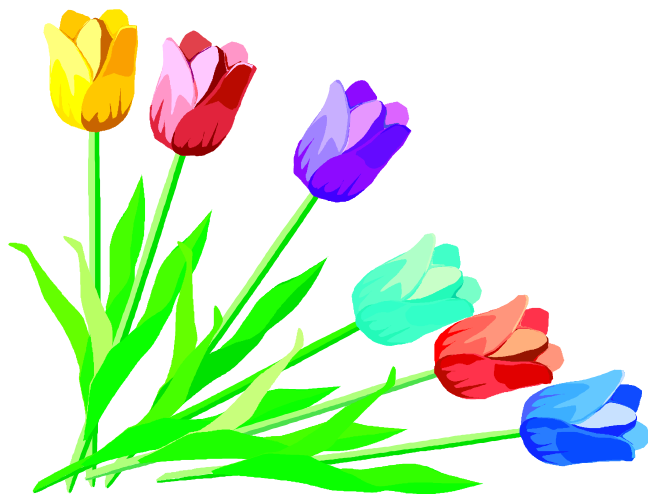
Supported transactions

- Conversational transactions
- Fast Path
- MSC
 - ▶ Stays on front-end
- Full Function
- Program to Program Switch
 - ▶ Must stay on the same system
-



Not supported

- CPI-C
 - ▶ Explicit APPC
 - ▶ Forced to the Front-end
 - ▶ Same as V7



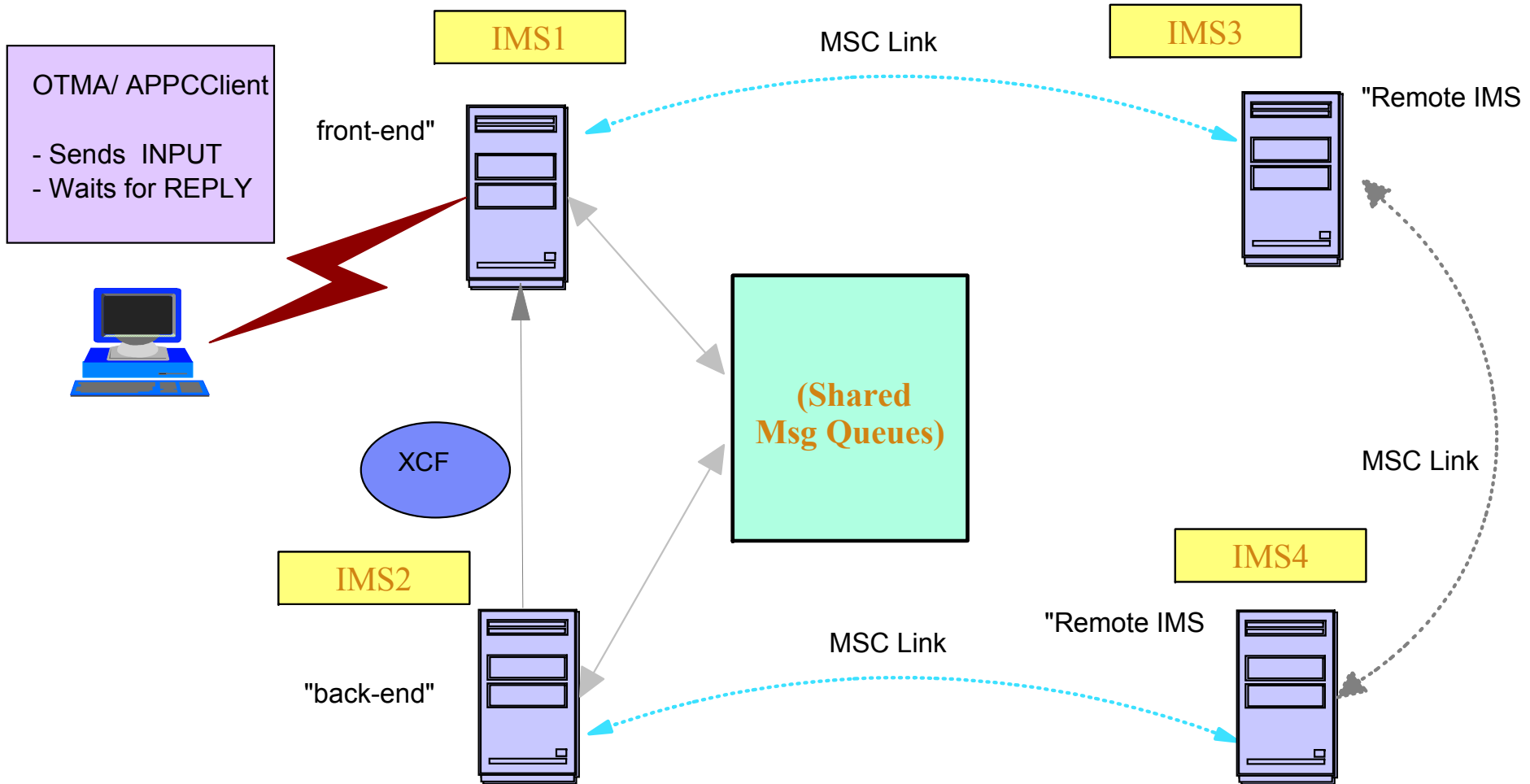
Fast Path considerations

- Shared EMH (different structure than SQ).
- If the transaction gets queued to the local EMH, no RRS calls are made.
- Algorithm favors local queue.
 - ▶ After 5 messages on local EMH, next message goes to global



MSC Considerations

APPC Synchronous (Allocate - Send - Receive)
 OTMA Commit-then-Send (Commit mode 1)



APPC/OTMA synchronous Transactions can be routed to remote MSC systems for processing from either the Front-end (FE) or Back-end (BE) IMS. This includes remote MSC transactions sent directly from the client, or RMT program-to-program switches from local transactions. MSC routes responses from the remote IMSs back to the FE IMS where APPC/OTMA support sends the output to the client.

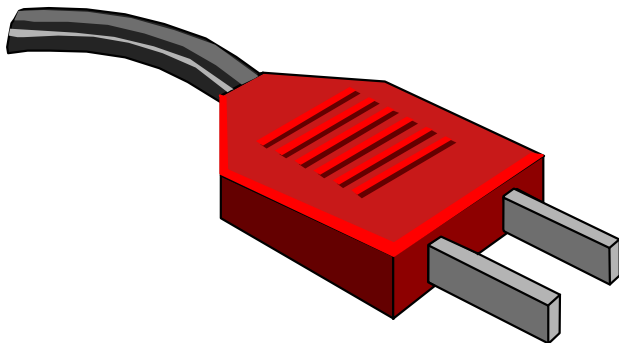
DFSCMUX0 user exit

■ Front-end

- ▶ Called after send failure
- ▶ Can route message
- ▶ If it does, Front-end will issue commit

■ Back-end

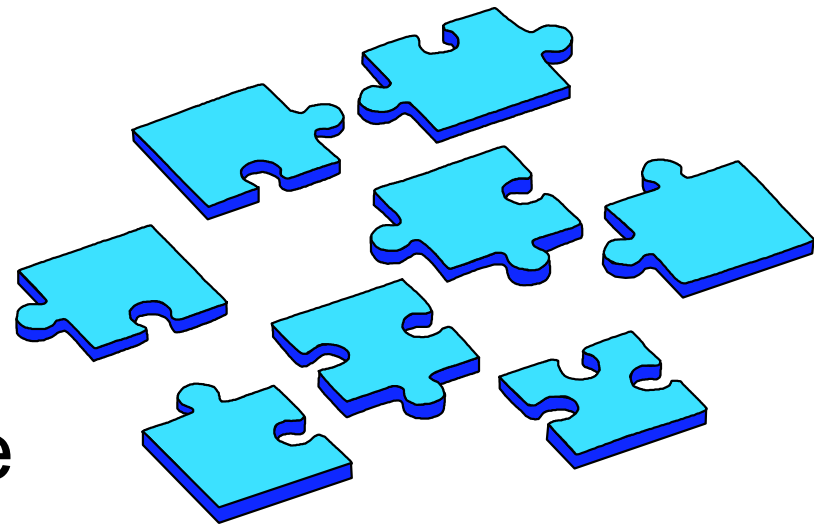
- ▶ Not Called when XCF
Send to Front-end
fails



Multiple RRS TCBs for commit/backout

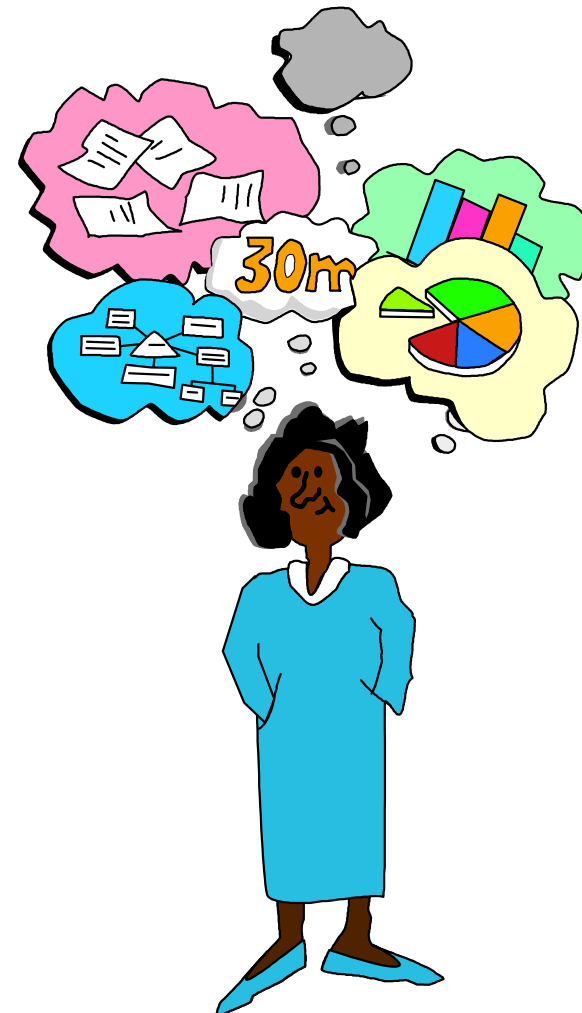
- attaches one at IMS initialization
- attaches ten more when function becomes active
- more if needed
- maximum 40
- one reserved for "fast" RRS services

Needed because RRS commit/backout will wait the TCB until completed and for ESTAE protection.



Requirements

- All IMS system within an IMSPLEX on V8 or V9
 - ▶ MINVERS in RECON 81
- RRS active on all MVS images
 - ▶ RRS=Y
- z/OS 1.2 or higher
- Shared Queues
- AOS=Y/F in DFSDCxxx

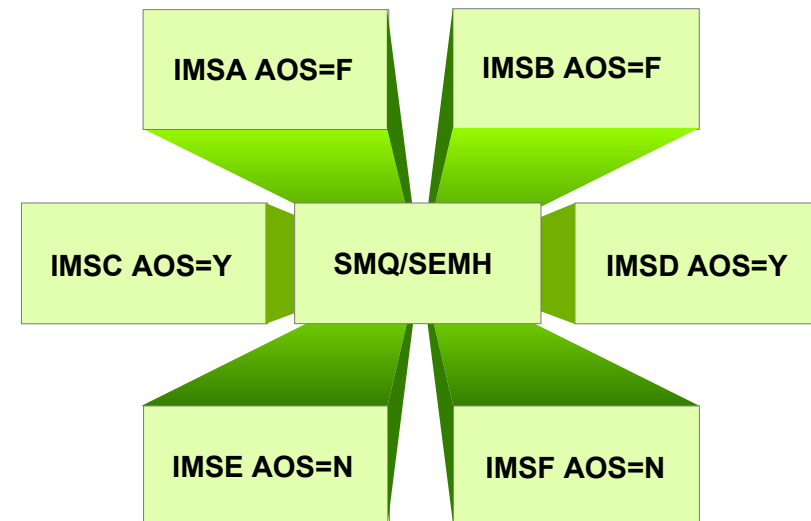


How to get it active

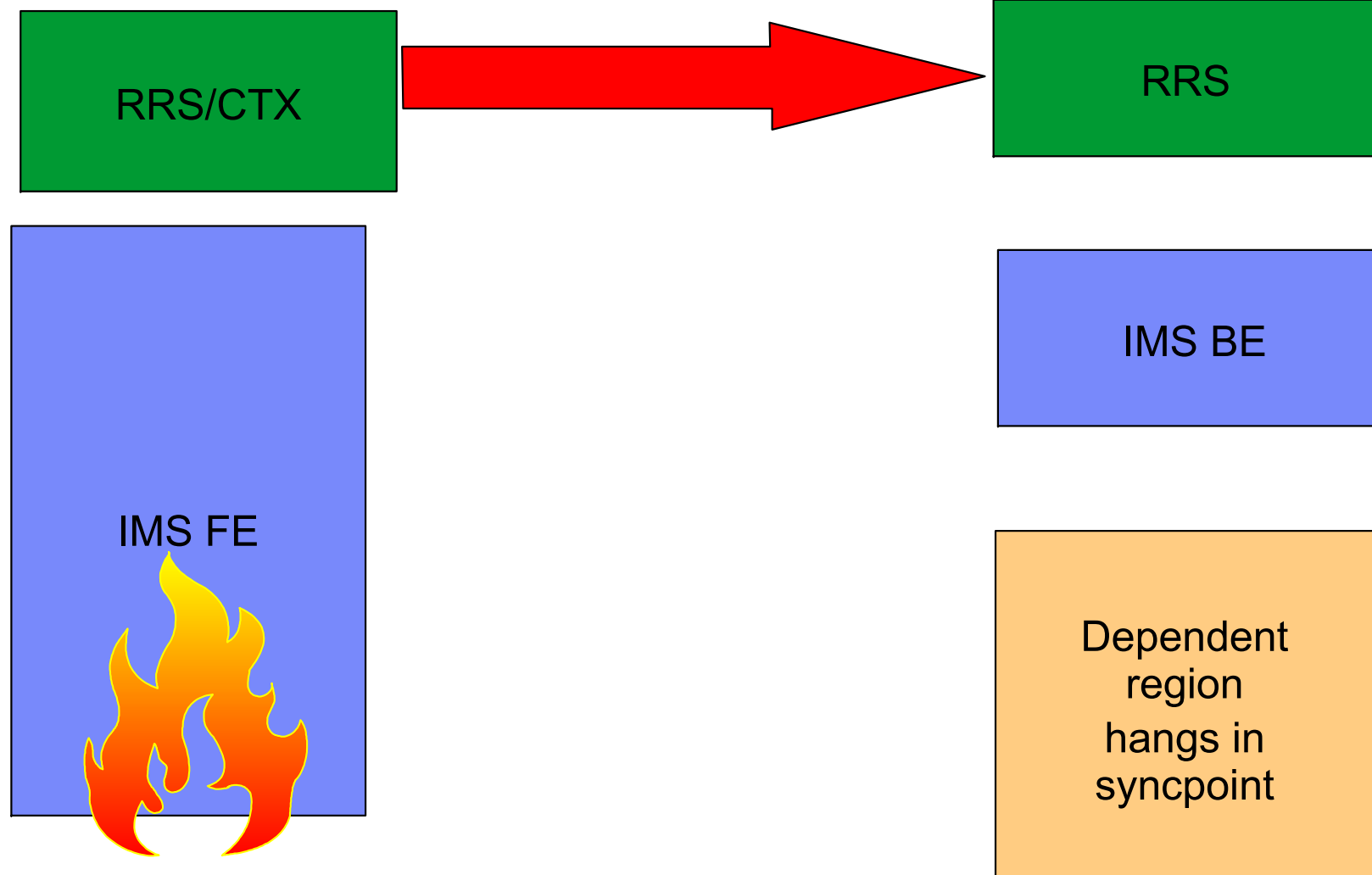


AOS parameter in DFSDCxxx

- AOS=N deactivates APPC/OTMA SMQ Enablement (default).
- AOS=Y activates APPC/OTMA SMQ Enablement if IMSPLEX is ready.
- AOS=F activates APPC/OTMA SMQ Enablement for the single IMS member. If one of the members in the IMSPLEX cannot process a transaction from another member it will abend it with U711.



Back-end dependent region waits for RRS to drive back-out exit when Front-end IMS abends.



PQ94029/PK00139

- Front-end (abend)
 - ▶ ESTAE sends XCF msg to members
 - ▶ Creates dump
 - ▶ Terminates address space
 - Now CTX will inform RRS about all the active contexts IMS-FE had.
 - RRS will inform the back-end RRS
 - Back-end RRS drives back-out exit
 - will abend application on back-end with U711
 - Back-end
 - ▶ DFSAOSW0 gets XCF abend notification message
 - ▶ Queues AWE to itself to scan LCREs
 - ▶ For every LCRE found (dependent region)
 - Check if dependent region is in RRS-PC wait for the abending FE
 - If yes, POST the ECB to wake up dependent region
 - Application will abend with U711
- 