Session C01
IMS Connect Experience at

Gary Hartley – gary.hartley@bankofamerica.com
Rich Coords – rich.coords@bankofamerica.com

Las Vegas, NV **September 15 – September 18, 2003**

RETURN TO INDEX

Topics To Be Covered

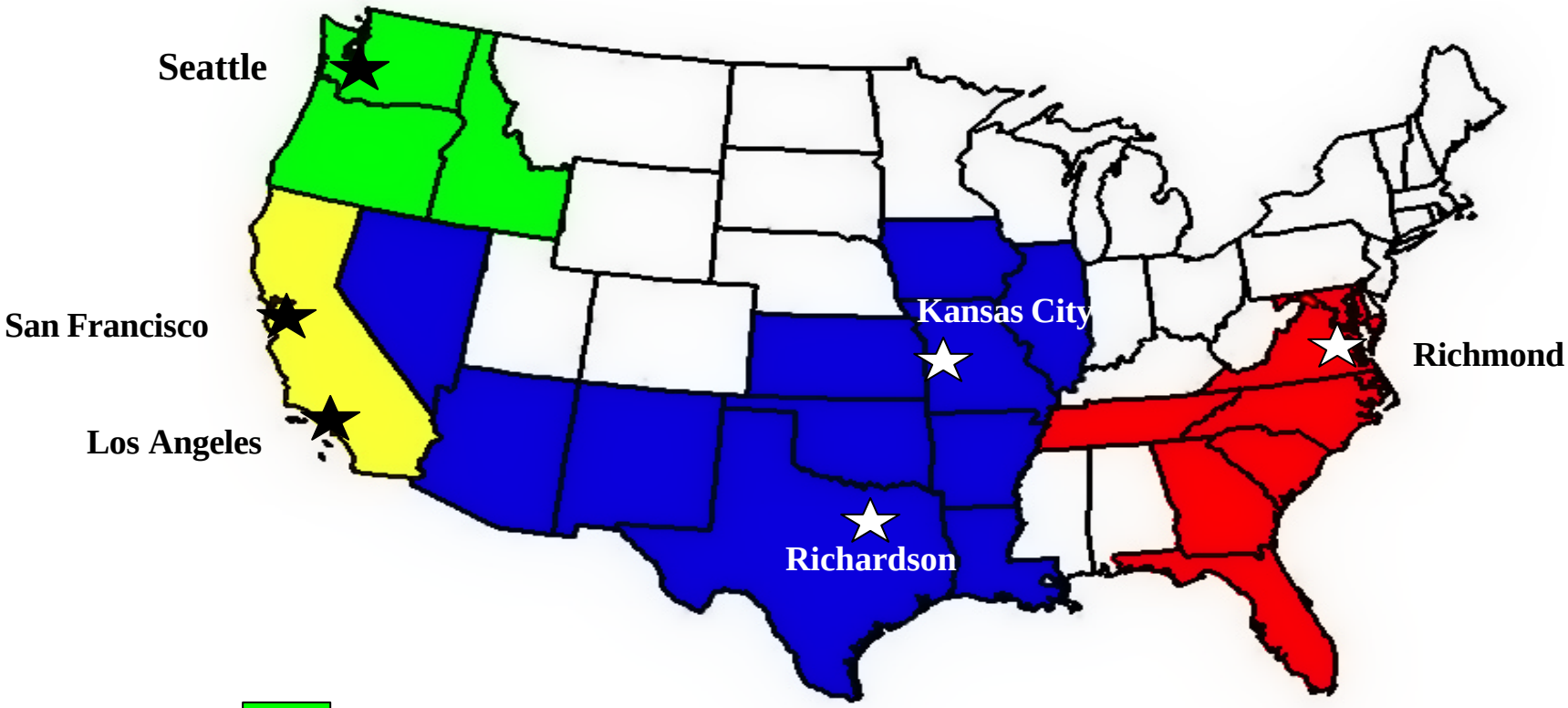
- Introduction
- Our Environment
- How It All Started (OTMA)
- OTMA Timeline
- Major Obstacle / Solution
- Problems Encountered / Solutions
- IMS Connect Timeline
- IMS Connect Requirements
- Basic IMS Connect Configurations
- Our IMS Connect Solution
- Required Exits

Our Environment

- All Systems are IMS V7.1 and Non-Shared Queues
  - Production
    - 17 Control regions
      - 2 XRF environments
    - 12 IMS Connect address spaces
  - Development
    - 25 Control regions
    - 4 IMS Connect address spaces
  - SVLS
    - 6 Control regions
      - 1 XRF environment
    - 3 IMS Connect address spaces
  - System Programmer
    - 12 Control regions
      - 4 under VM
      - 1 XRF environment
    - 6 IMS Connect address spaces
      - 4 under VM

Our Environment (continued)

- Financial Authorization systems
- ATM / Point of Sales
- Various Teller systems
- Banking Centers
- Online / Web Banking
- Wire Transfers
- Loan Application Scoring System
- Voice Guided Service - IMS Connect



Seattle, Washington Data Center - Production

San Francisco – Los Angeles, California Data Center - Production

Richardson, Texas Data Center – Production

Richmond, Virginia Data Center - Production

Kansas City, Missouri Data Center – Development, Contingency & SVLS

How It All Started

- Moratorium on SNA based application development in 1999

- This triggered our R&D efforts for the MQ-IMS Bridge and IMS Connect projects
  - Learn product functions and limits
  - Set standards
  - Install and configure products
  - Modify existing exits and usermods, if required
  - Create new exits and usermods, if required
  - Develop IVP transactions
  - Provide training
  - Create User's Guide for applications and systems programmers
  - Provide turnover for ongoing support

OTMA Timeline

- May 2000
  - Began R&D project for OTMA and MQ-IMS Bridge
- July 2000
  - Installed into development environment
- February 2001
  - Installed into California production environment
  - TPF (Transaction Processing Facility) conversion to MQS

Major Obstacle

- Systems that key on LTERM field of I/O PCB
  - Filled in by IMS logon process
  - OTMA clients use TPIPE name by default
- Development Transaction Router
  - Provides multi-level application development environment
    - The backbone of the development systems
  - MFS screen will force you to pick a level based on LTERM or Userid if not in static table
  - 3270 based system (Sign-on required)

Solution

- MQ OTMA header (MQIIH) field **LtermOverride**
  - This field is optional
  - Places the field into the IO PCB LTERM
  - If not defined, TPIPE name is used
- **IRM_LTERM** field does the same function for non IMS Connector for Java clients
  - User message exit must move to OMHDRLM field in the OTMA State Data Header when building the OTMA header record
- IMS Connector for Java users can set the "**LTERM NAME**" in the **IMSinteractionSpec** bean

OTMA Problems Encountered

- California system converting from Transaction Processing Facility (TPF) to MQ-IMS Bridge
  - Application received status code “AM”
  - OTMA does not support updates to MSDB
- Temporary Solution
  - Return all MSDB update transactions to TPF control
- Permanent Solution
  - Massive project to convert all MSDBs to DEDBs

OTMA Problems Encountered (cont)

- Application converted to MQ-IMS Bridge
  - Application not sending response to device when using ALTPCB
  - Inserts to non-transaction ALTPCB will always return messages to originating source destination without the OTMA Routing exit installed (DFSYPRX0)
- Temporary Solution
  - Revert back to posting the ALTPCB messages with end-of-day batch processing.
- Permanent Solution
  - Code a DFSYPRX0 exit to handle these conditions
  - Exit is still in testing

IMS Connect Timeline

- November 2000
  - Began R&D project with IMS TCP/IP OTMA Connection (ITOC) under IMS V6.1
- February 2001 - Web Teller initiative
  - IMS Connect V1.1 installed into development with IMS V7.1
- September 2002
  - Converted all development systems to IMS Connect V1.2
- January 2003
  - Installed IMS Connect V1.2 into production
  - Many false starts
- June 2003
  - Installed IMS Connect V2.1 into development

IMS Connect Requirements

- Security
- Automated
  - Startup and Recovery
- Alternate Availability
  - Rolling Sysplex Outage Support
  - Reduce FCI counts – Failed Customer Interactions
- Distribution of Workload
- Scalability
- Simplicity of User Interface

Richmond, Virginia Data Center

- Production system used in next slides
  - 3 IMS systems – sysplexed and data-shared over 3 LPARs
  - 3 IMS Connect address spaces over 3 LPARs
  - .12 average response time
  - 590 peak transactions per second
  - 2,124,000 peak transactions per hour
  - 627 MPP regions
  - 258 FP regions
  - 656 databases
  - 515 Fast Path areas
  - 30 MSC links
  - 117 ISC links

Security Limitations

- Two security options with base IMS Connect
  - Security checking off - **RACF=N**
  - Security checking on - **RACF=Y**
  - Security checking on, validates the userid and password. If they are valid, access is allowed to IMS Connect

- We developed our own security exit
  - Validates the userid and password
  - Verifies the user is allowed access to development or production IMS Connect address spaces

- Current enhancements to base IMS Connect security
  - RACF PassTicket - PTF **UQ73278** V1.2
  - Secured Socket Layer (SSL) - Available with V2.1

IMSLSECX – User Security Exit

- Security requirements
  - Provide Userid/Password verification
  - Provide sign on type authorization
    - IMSPCONN for production
    - IMSTCONN for development
  - Minimize number of RACF calls
    - Provide **UTOKEN** support
    - Provide caching support
      - All RACF caching under VLF

- Implementation
  - Set **RACF=N** to turn off IMS Connect security checking.
  - Replaced IMS Connect supplied message exits with our own
    - HWSIMSO0 - IMS Connect TCP/IP user message exit
    - HWSJAVA0 - IMS Connector for Java user message exit
  - All sample user message exits were modified to call our security exit

IMSLSECX – User Security Exit (cont)

```
RACROUTE REQUEST=VERIFY,                X
ENVIR=CREATE,                           X
USERID=USERIDL1,                        X
PASSWRD=PASSWDL1,                       X
ACEE=ACEEANCH,                          X
APPL=ACLASSN1,                          X
TOKNOUT=UTOKAREA,                       X
INSTLN=0,                               X
PASSCHK=YES,                            X
STAT=ASIS,                              X
SMC=NO,                                 X
WORKA=RACWORK,                          X
RELEASE=1.9.2,                          X
MF=(E,RACVRFYL)
```

IMSLSECX – User Security Exit (cont)

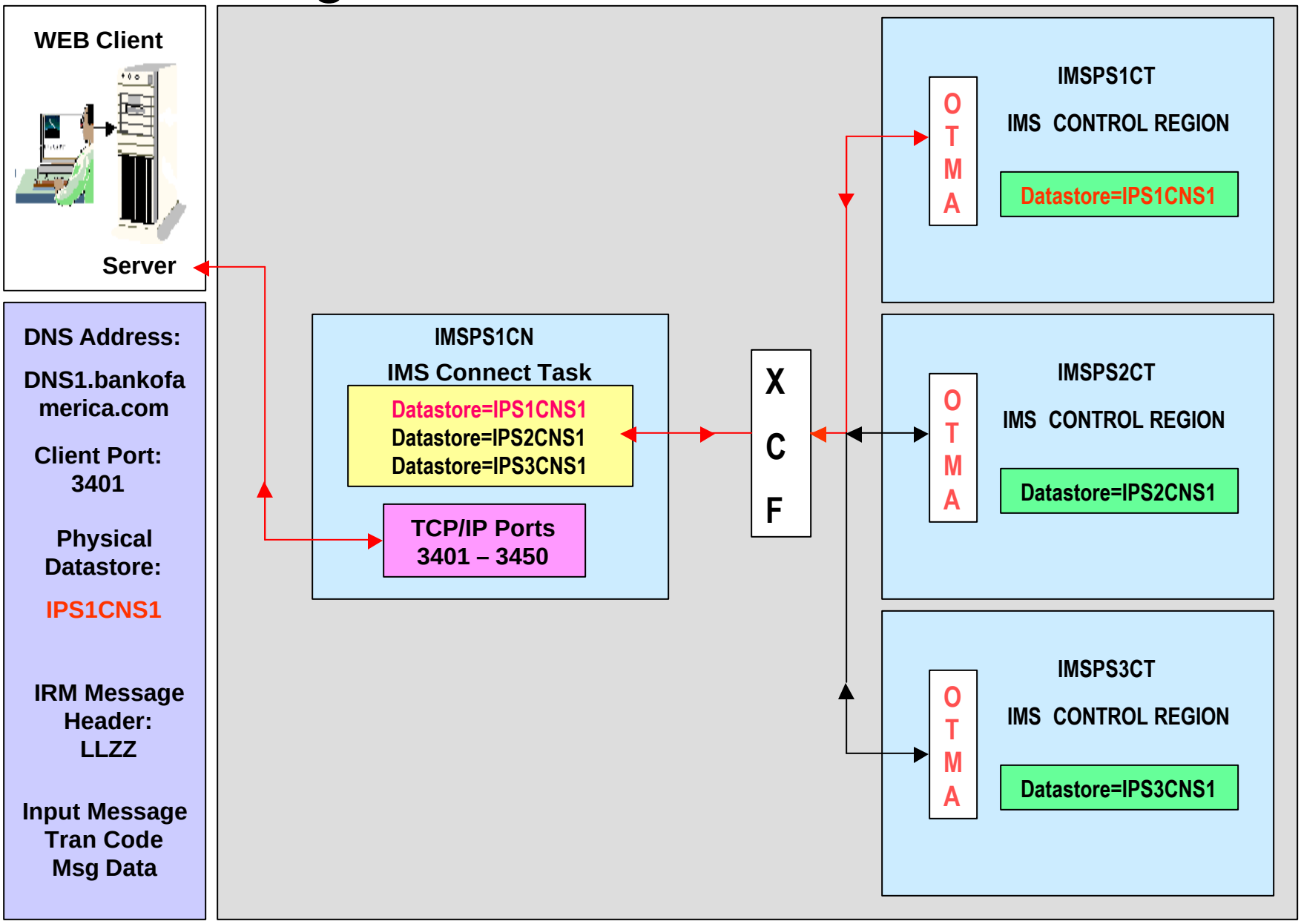
- The call must use the execute form of the macro since the exit is required to be assembled and link edited as re-entrant (**RENT**).
- The **ENVIR=CREATE** keyword creates the ACEE control block. This call must be followed by an **ENVIR=DELETE** when RACF returns with **RC(0)** or the address space will eventually abend on storage.
- The **APPL=** keyword will be set to **IMSTCONN** for development **or IMSPCONN** for production**.**
- The **TOKNOUT=** keyword builds the UTOKEN from the ACEE control block. The UTOKEN will be passed by the message exits to IMS Connect in the **OMSECPRF** field of the OTMA Security Data Header . No additional I/O to the RACF database is required by OTMA or IMS TM for userid verification.

Automation Solution
SA390

- IMS Connect address space starts
  - During IPL process and after TCP/IP has started
  - After a CLOSEHWS command
  - After IMS Connect failure
    - Will attempt 3 times within 5 minute period

- Datastore opens
  - IMS Connect will perform if IMS is active
  - After **/START OTMA** command from IMS
    - **DFS2360I** XCF GROUP JOINED SUCCESSFULLY. **IPS1**
  - xxxOPENDS **IPS1**CNS1 / **IPS1**CNS2 / **IPS1**CNS3
  - Not required with APAR PQ68764

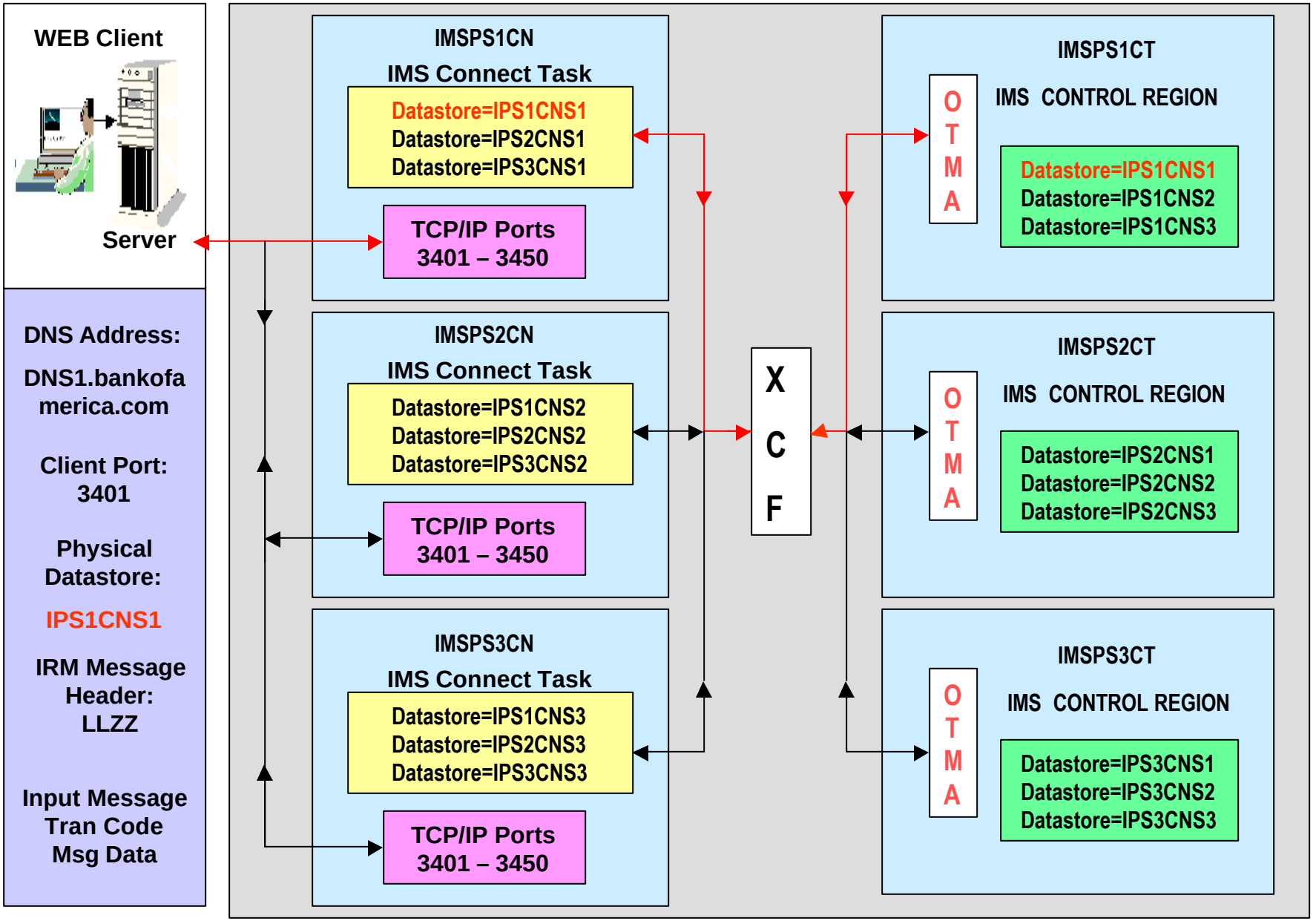
Single IMS Connect Architecture



Single IMS Connect Architecture Limitations

- Alternate Availability
  - Single point of failure on IMS Connect address space
- Distribution of Workload
  - Server could balance the work load but would need to change the datastore name for each connect request
  - Server will not know what IMS systems are available
- Scalability
  - Server group would need to be informed of additional IMS systems or IMS Connect address spaces added to the sysplex

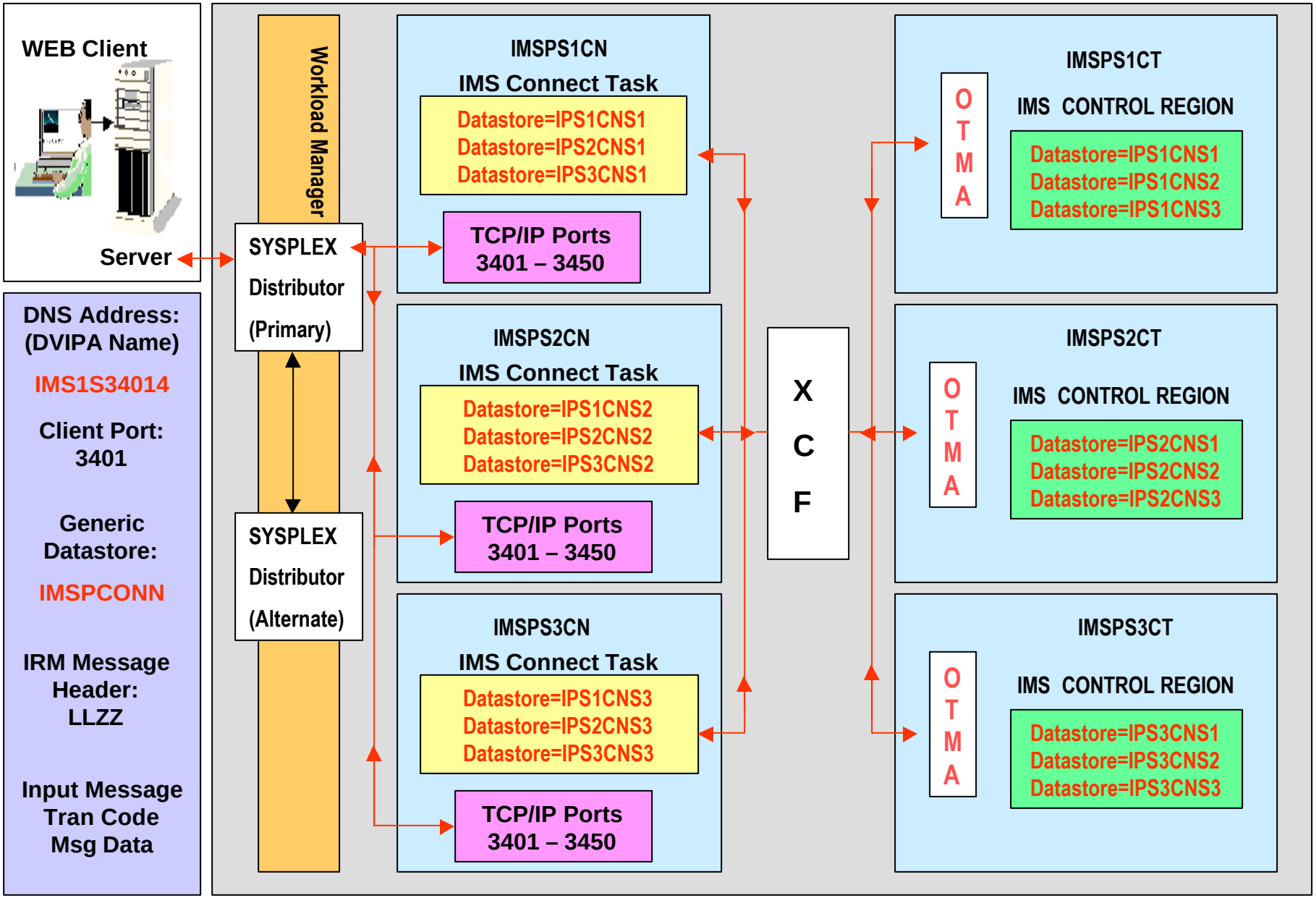
Multiple IMS Connect Architecture



Multiple IMS Connects Architecture Limitations

- Distribution of Workload
  - Server could balance the work load but would need to change the datastore name and IMS Connect DNS name for each connect request
  - Server will not know what IMS systems or IMS address spaces are available
- Scalability
  - Server group would need to be informed of additional IMS systems or IMS Connect address spaces added to the sysplex

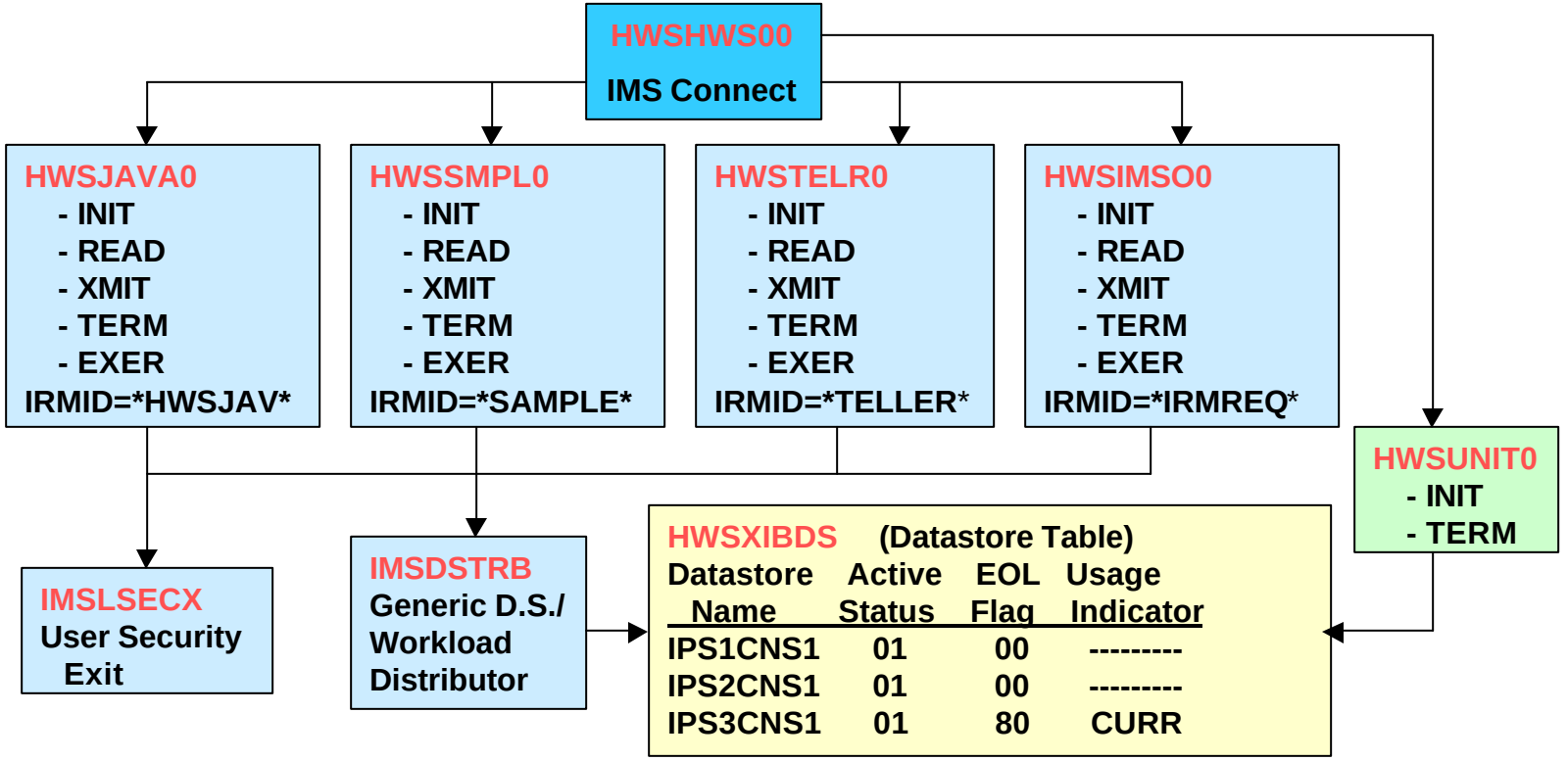
Sysplex Distributor Architecture



Sysplex Distributor Solution

- Alternate availability
  - Alternate Sysplex Distributor inherits TCP/IP stack when primary fails
  - Sysplex Distributor only passes messages to available IMS Connect systems based on Workload Manager decision
  - IMS Connect will pass messages to any active IMS systems
- Distribution of workload
  - IMS Connect exit evenly distributes messages to all IMS systems
- Scalability
  - IMS Connect exit distributes to IMS systems based on configuration member definitions
- Simplicity of User Interface
  - Server only needs to know about generic datastores (**IMSTCONN**) for development or (**IMSPCONN**) for production
  - Server only needs to know the generic DNS name for location of IMS Connect address spaces

User Exit Routines




```
Configuration Member 'SYS1.IMSVS.PROCLIB(HWSCFG01)
HWS (ID=IMSPS1CN,RACF=N)
TCPIP (HOSTNAME=TCPIP,RACFID=,PORTID=(XX01,XX02,XX03,XX04,
XX05,XX06,XX07,XX08,XX09,XX10,XX11,XX12,XX49,XX50),
MAXSOC=2000,
EXIT=(HWSJAVA0,HWSSMPL0,HWSTELR0,HWSIMSO0))
DATASTORE (ID=IPS1CNS1,GROUP=OTMAP1S,MEMBER=IPS1CNM1,TMEMBER=IPS1)
DATASTORE (ID=IPS2CNS1,GROUP=OTMAP1S,MEMBER=IPS2CNM1,TMEMBER=IPS2)
DATASTORE (ID=IPS3CNS1,GROUP=OTMAP1S,MEMBER=IPS3CNM1,TMEMBER=IPS3)
```

HWSUNIT0 – User Initialization Exit

- Called by IMS Connect during start-up and termination
- Initializes the datastore table area (DSECT HWSXIBDS), pointed to by Register 1
- Datastore table is loaded from the configuration file and it's entries are modified by IMS Connect whenever a datastore is opened or closed
- Verify the last datastore entry in the list is set to X'80' - (XIBDS_FLAG)
- Display error message and terminate if datastore table does not contain an entry with X'80' indicator set
- Move eye catcher 'CURR' to the table entry containing X'80', which will be used by the generic datastore exit as a starting point
- Display a message indicating initialization complete and return control to IMS Connect

IMSDSTRB - Generic Datastore Exit

- Called by each of the User Message exit routines when the pre-defined generic datastore name of "**IMSPCONN**" is selected by client application.

- Uses the datastore table area DSECT '**HWSXIBDS**' to determine which datastores are available for routing.
  - When a datastore status field is set to X'**01**', the datastore entry is eligible for use.
  - The entry pointer is set to '**CURR**' and the datastore name from that entry is used to overlay the generic name and routing occurs using the physical datastore name.
  - When a datastore status field is set to X'**00**', the datastore entry is unavailable and is bypassed. Status checking continues until the next active entry in the datastore table is found.

- The next transaction through will be assigned to the physical datastore extracted from the next active datastore entry in the table.

IMSDSTRB - Generic Datastore Exit (cont)

- The process to determine which datastore to select will continue until the entry containing X'**80**' in the **XIBDS_FLAG** field is found. If active, routing occurs and the datastore table pointer will be re-positioned back to the first active physical entry in the table and the "round-robin" transaction delivery continues.

- If the IMS Connect system changes the status flag of a datastore entry to inactive due to a datastore being stopped, the datastore will be bypassed until the entry is marked as active again by IMS Connect.

- A datastore will be set to '**NOT ACTIVE**' if the IMS system to which it was connected had a **/STOP OTMA** command issued against it or the IMS System is no longer available.

IMSDSTRB - Generic Datastore Exit (cont)

- System automation will issue an **open datastore** command using the IMS Connect outstanding reply and the datastore status is set to '**ACTIVE**' by IMS Connect whenever the message "**DFS2360I hh:mm:ss XCF GROUP JOINED SUCCESSFULLY. IMSID**" is generated by the IMS TM.

- If all datastore entries are marked as '**INACTIVE'**, message "**NOACTIMS**" will be passed back to the calling User Message exit routine and error message "**NFNDDST**" will be passed back to the client application. The following IMS Connect error message will be displayed on the System Console."**HWSS0742W Message failed, ORIGIN=3401_CLIENTNM to DESTID=NOACTIMS; R=4, S=NFNDDST, M=SRE4**"

HWSSMPL0 – User Message Exit

- **INIT Subroutine** – Control is passed to this routine after the exit has been successful loaded. You can modify the HWSSMPL0 source in SHWSSRC or create your own user message exit with a unique MSGID by coding an INIT subroutine in your exit that specifies the MSGID you want associated with your exit.
  The following are reserved:

  SAMPLE - used by HWSSMPL0
  IRMREQ - used by HWSIMSO0
  HWSJAV - used by HWSJAVA0

  User written example:

  TELLER - for exit HWSTELR0
- Modify EBCDIC token EIRMID with MSGID.
- Modify ASCII token AIRMID with MSGID.
- Add the exit name to the EXIT parameter of the TCP/IP statement in your configuration member.
- Assemble and Link Edit as re-entrant (RENT).
- Copy the exit to SHWSRESL.

HWSSMPL0 – User Message Exit (cont)

- **READ Subroutine** – Control is passed to this routine after a complete request message is received from a TCP/IP client. Data conversion from ASCII to EBCDIC is performed. The translate table that can be modified as needed. The User Security exit (**IMSLSECX**) is called from this routine and the UTOKEN is moved to the OTMA header. The Generic Datastore user exit (**IMSDSTRB**) is also called from this routine.
- **XMIT Subroutine** – Control is passed to this routine after a complete response message has been received from the datastore. If needed, data translation from EBCDIC to ASCII is performed using a translate table that can be modified as required.
- **TERM Subroutine** – Control is passed to this routine when IMS Connect is shutting down.

Questions ?