

# An Introduction to Distributed Processing with DB2 for z/OS

Hugh Smith ([smithhj@us.ibm.com](mailto:smithhj@us.ibm.com))

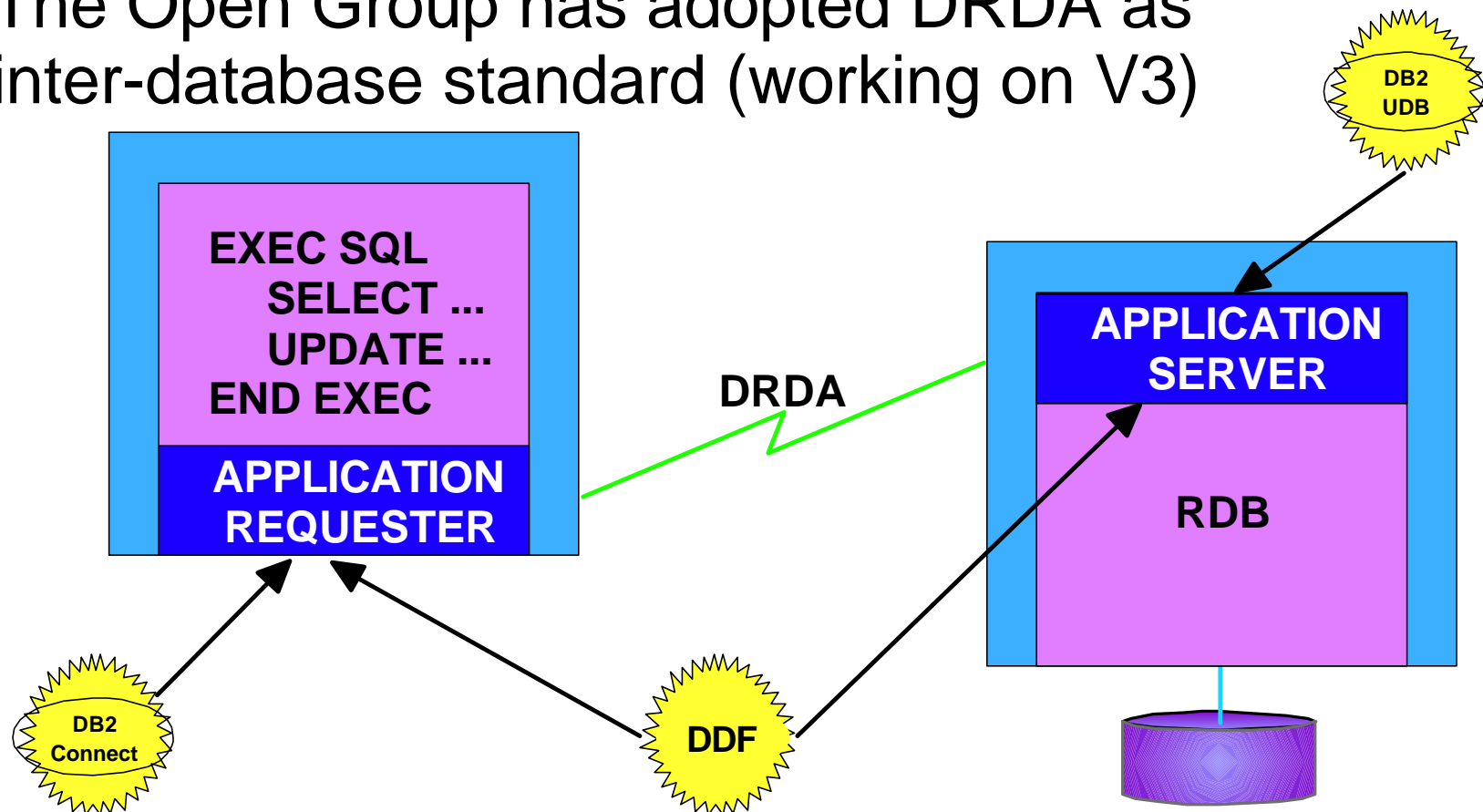
08/16/04 -- 1316

# Topics

- Overview of Distributed Relational Database Architecture (DRDA)
- DDF 101
  - ▶ Overview of DDF
  - ▶ Installation Parameters and JCL
  - ▶ DB2 Commands
  - ▶ Basic Communications Database (CDB)
  - ▶ How Clients Connect
  - ▶ Controlling Access

# Distributed Relational Database Architecture (DRDA)

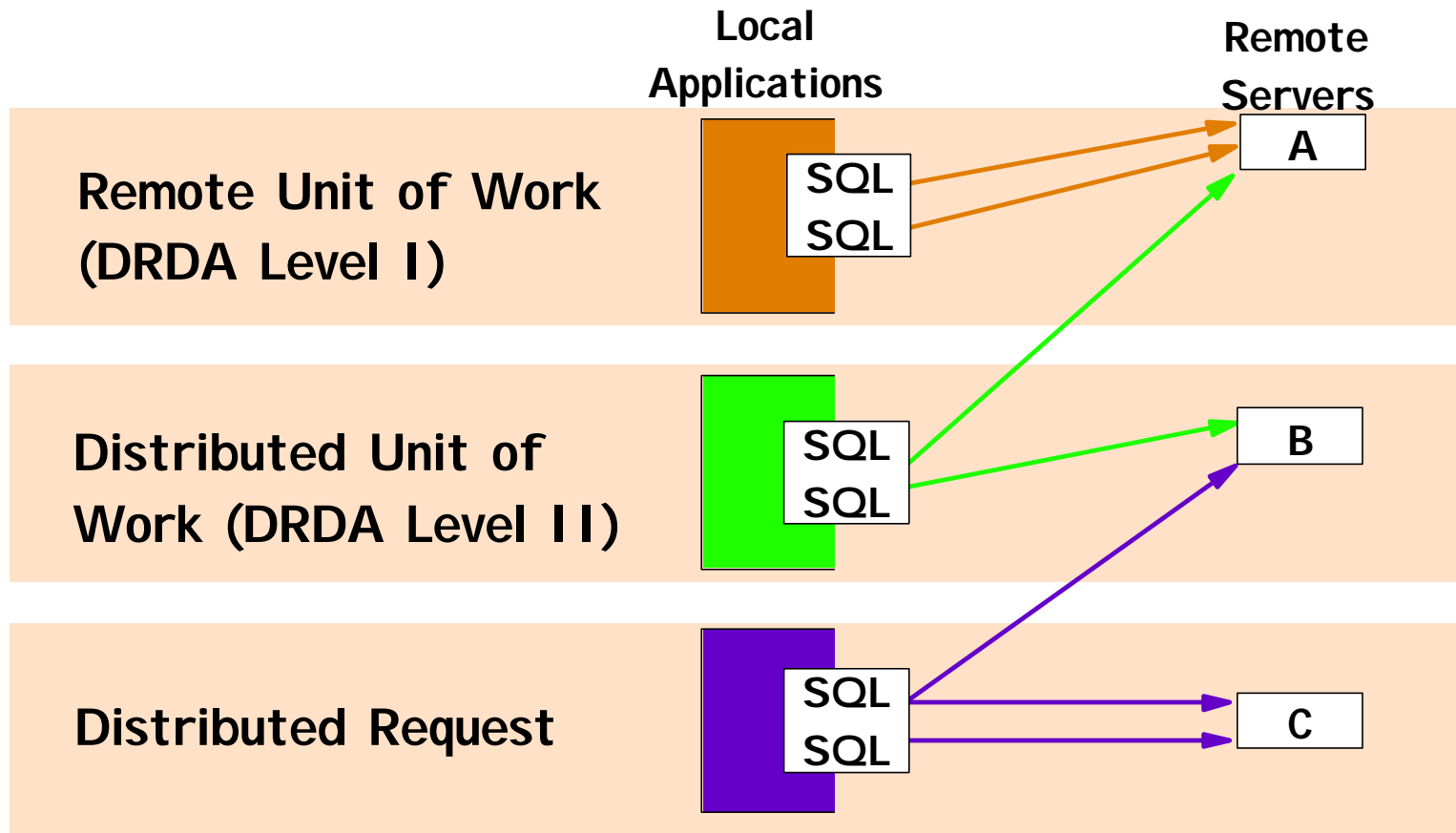
- Open Client/Server Database Architecture
  - ▶ Provides Any-to-Any support
  - ▶ The Open Group has adopted DRDA as inter-database standard (working on V3)



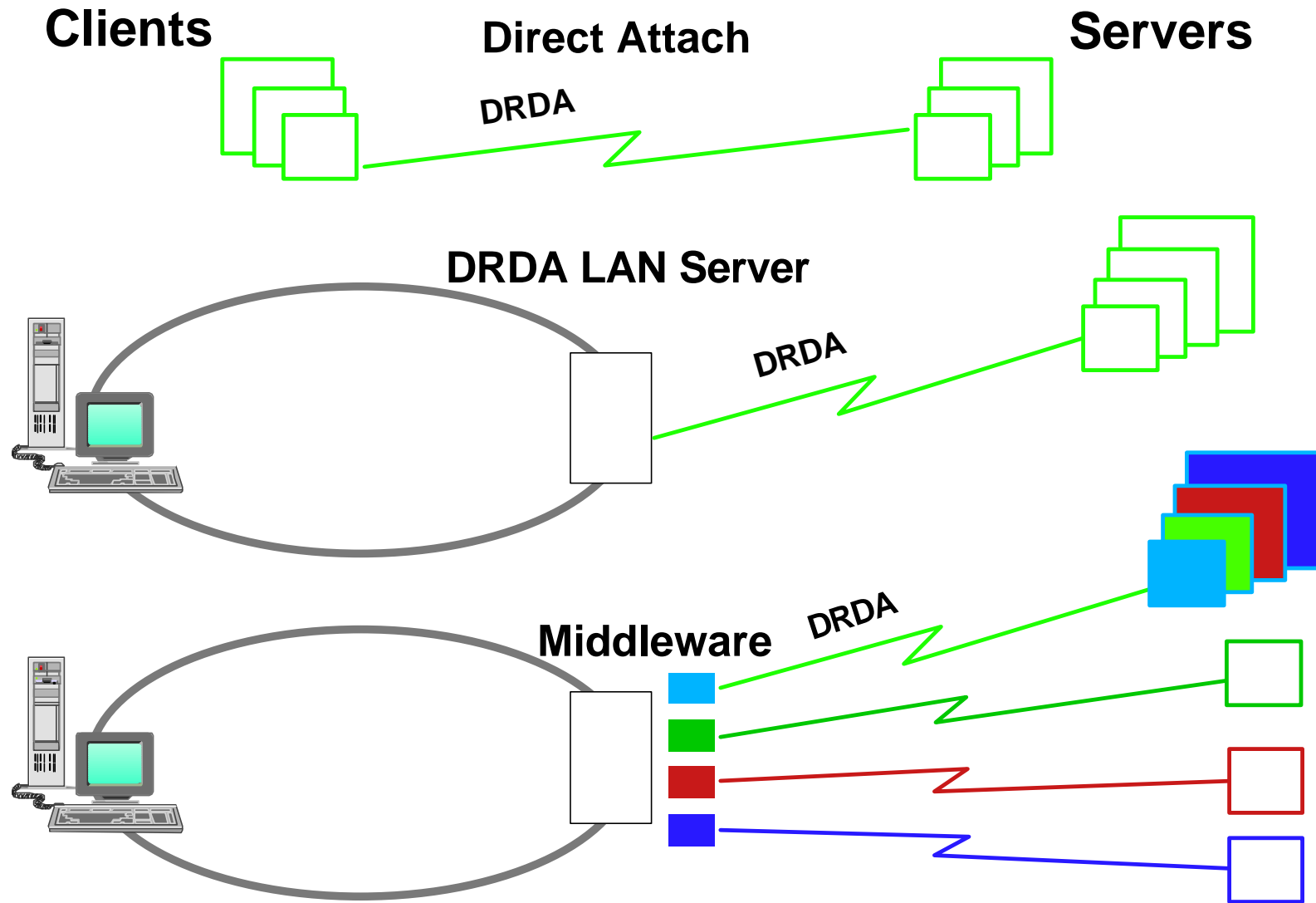
# DRDA Characteristics

- Any SQL dialect, prebound and dynamic
- Automatic data transforms
  - ▶ "receiver makes right"
- Unit of work support
  - ▶ Coordinated commit, recovery
  - ▶ Single or Multi-site
- Supports stored procedures
- Superior performance, availability
- Security
  - ▶ No subversion of existing security mechanisms
  - ▶ Integrates with DCE/Windows Kerberos security networks
- Systems management

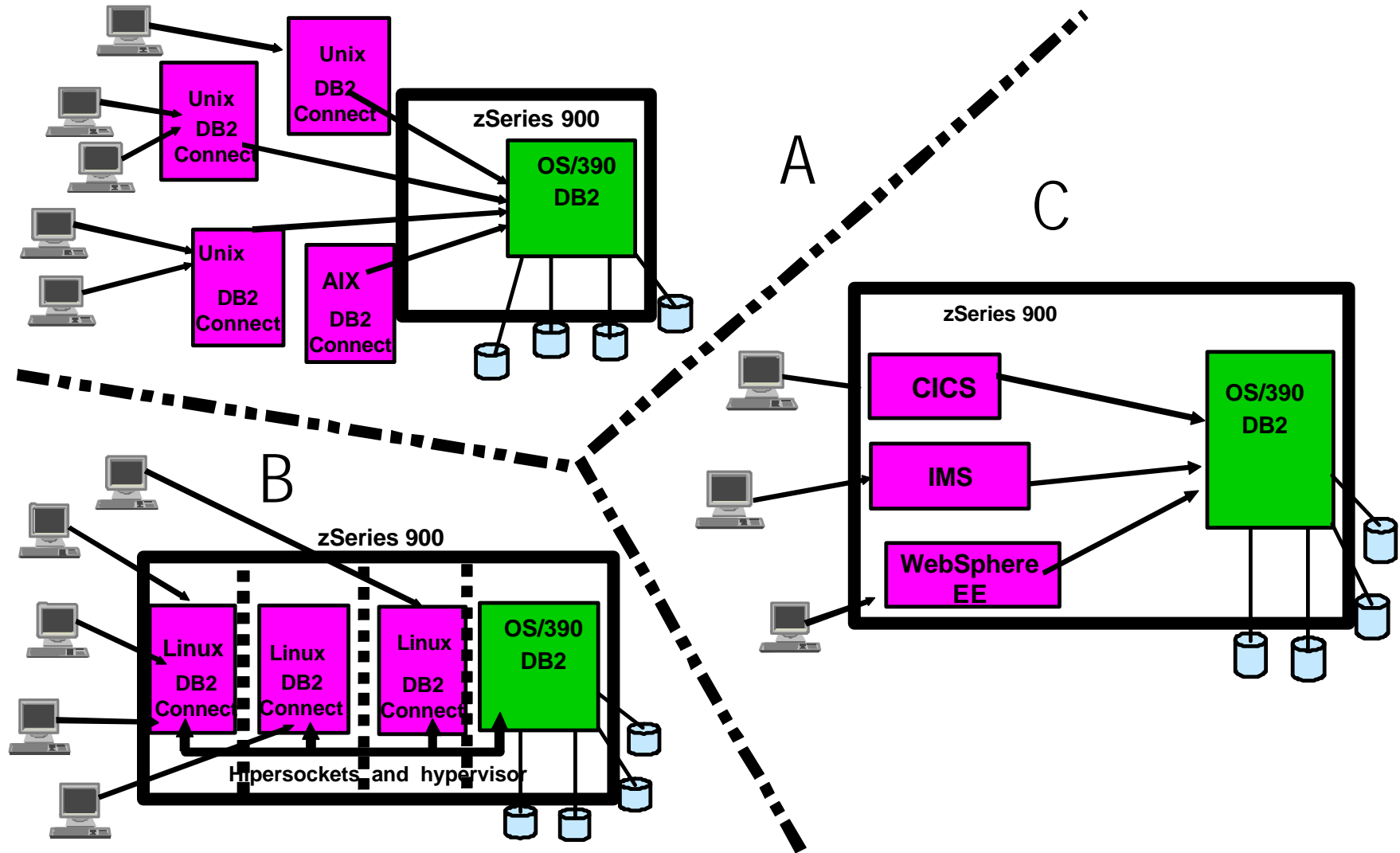
# Remote Relational Data Access Types



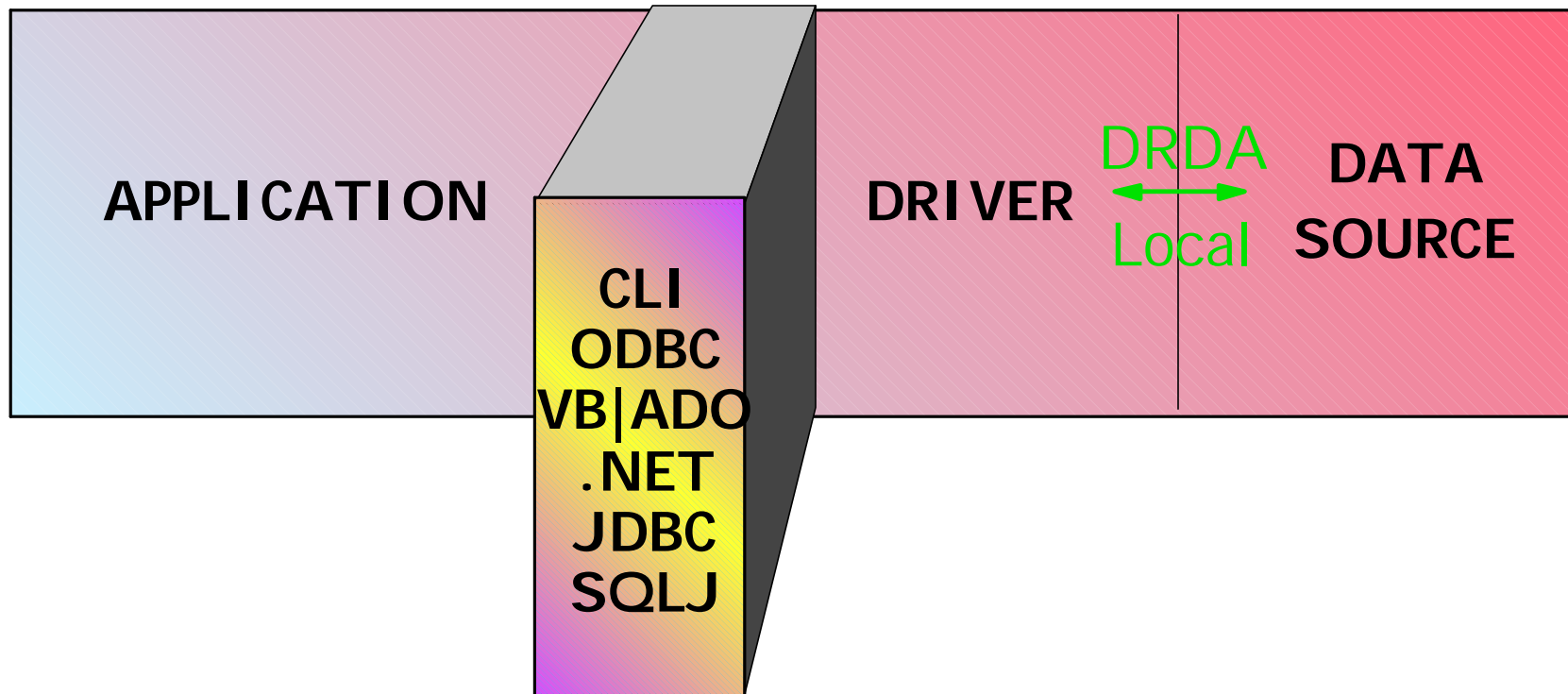
# DRDA Configuration Models



# Customer eBusiness Solutions



# Application Programming Interface to DRDA Relationship



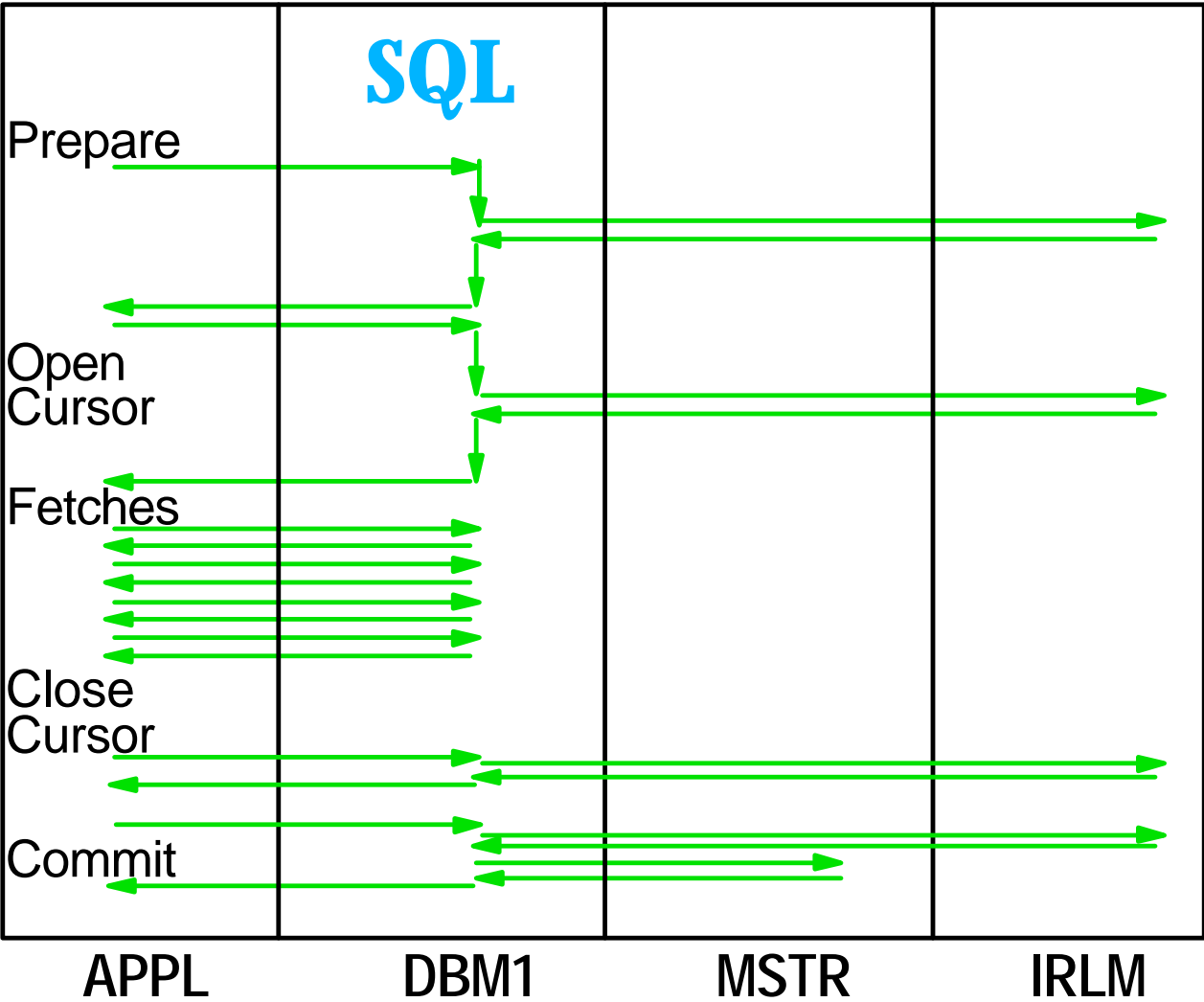
- DB2 distributed client provides driver for ODBC|CLI|VB|ADO|.NET applications
- DB2 distributed client and mainframe provide new Universal Driver for SQLJ and JDBC



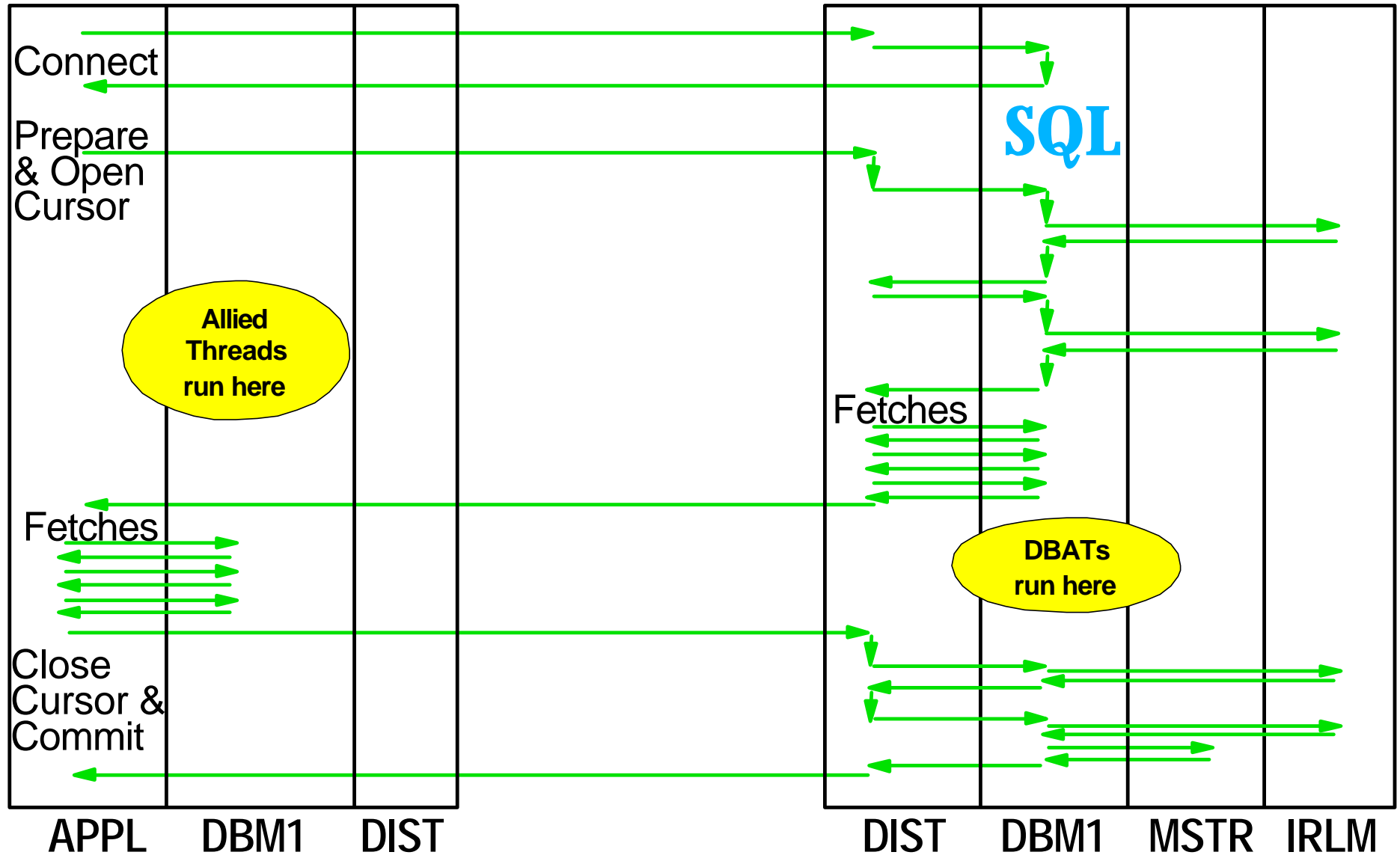
# DDF 101

- Distributed Data Facility originally delivered in V2R2
- Implements full DRDA AR and DRDA AS
- Code runs in separate address space (ssidDIST)
  - ▶ accesses into database manager (ssidDBM1) via Cross Memory Services
  - ▶ trusted address space (runs under same key as database server - less CPU)
  - ▶ started via DDF ZPARM value (AUTO - started with DB2, or COMMAND - started via START DDF command)
  - ▶ ssidDIST should have same priority or goal as other DB2 address spaces but less than IRLM

# DB2 Address Spaces with Application



# DB2 Address Spaces with Application and DDF



# DDF 101 (*continued*)

- Accesses are called Database Access Threads (DBATs)
  - ▶ run as enclaves (preemptible SRBs) which originate in ssidDIST
  - ▶ enclaves are separately managed by System Resource Manager (SRM) than address space
  - ▶ enclaves can have goals/priorities different than that of ssidDIST and each other (SUBSYS=DDF classification rules)
  - ▶ ssidDIST started task must have higher priority or goal than enclaves in it so new work requests will be started in a timely manner
  - ▶ minimum 200KB of ssidDBM1 storage for DBAT (connection is 7.5KB in ssidDIST address space)

# DDF 101 (*continued*)

- Communication to/from DDF via SNA/APPC (Application Program to Program Communication) or TCP/IP

- ▶ DDF must open its VTAM ACB at startup even if only using TCPIP

- ▶ Sample VTAM APPL statement:

```
DSN*      APPL  AUTH=(ACQ), PARSESS=YES,  
          HAVAIL=YES, EAS=1600,  
          APPC=YES, DSESLIM=1024,  
          DMI NWNL=512, DMI NWR=512, AUTOSES=1,  
          SECACPT=ALREADYV, SRBEXIT=YES,  
          SYNCLVL=SYNCPT,  
          MODETAB=DB2MODET, VPACING=63
```

X  
X  
X  
X  
X  
X

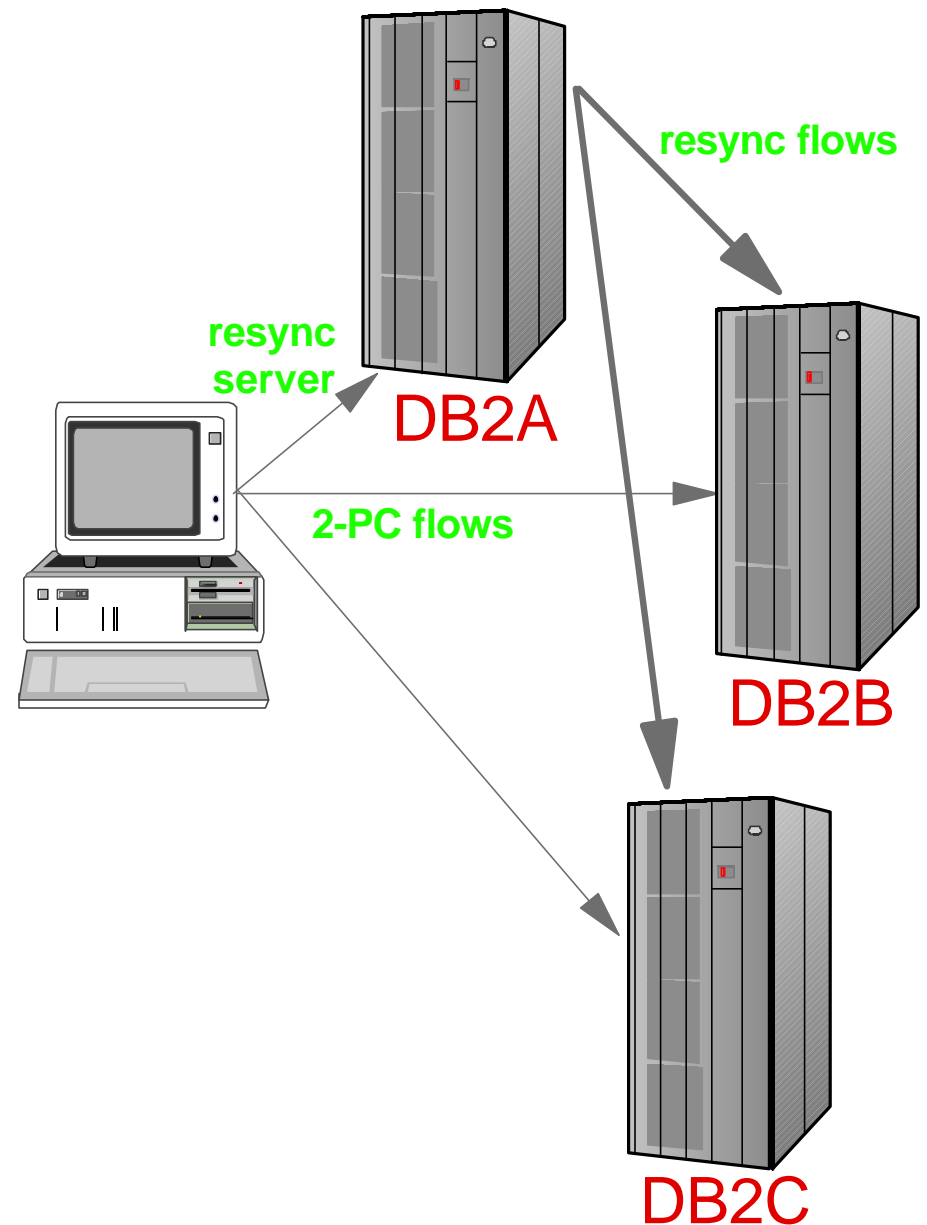
TIP: if you make every DB2 have an LUNAME beginning with DSN and the remainder being the subsystem id, then with the above VTAM APPL in any system, DB2 DDF will always be able to open its VTAM ACB!

# DDF 101 (*continued*)

- For TCP/IP:
  - ▶ UNIX System Services (USS) required for DB2's use of Asynchronous socket I/O
  - ▶ DDF address space must have a STARTED userid which has been defined to USS as a superuser (UID of 0 - no longer required as per APAR PQ87444)
  - ▶ V6 and V7 now supports Dynamic Virtual IP Addressing (see APAR PQ46659 - primarily for data sharing)
- DB2 can participate in or coordinate a distributed unit-of-work via DRDA or APPC
  - ▶ for DRDA clients over TCP/IP, DB2 can be asked to become coordinator (no client log)

# 2-phase Commit -- No Client Log

- Requester picks server to perform 2-phase logging
- Sends resync server name to all partners
- Normal 2-pc protocol is used until errors occur.
- Resync server takes over for error resolution



# DDF 101 (*continued*)

## ■ ssidDIST JCL when using TCP/IP

```
//D81BDIST PROC RGN=OK, LIB=' DSN810B. SDSNEXIT'
```

```
//STEPAFF EXEC PGM=BPXTCAFF, PARM=TCPIP NAM
```

You need the above line if you have multiple TCP/IP stacks on your LPAR. TCPIP NAM is the procedure name of the TCP/IP stack to which you want affinity.

```
//D81BDIST EXEC PGM=DSNYASCP, REGION=&RGN
```

```
//STEPLIB DD DISP=SHR, DSN=&LIB
```

```
// DD DISP=SHR, DSN=DSN810. SDSNLOAD
```

```
//* DD DISP=SHR, DSN=CEE. SCEERUN
```

You need the above line uncommented if the Language Environment (LE) runtime library is not available via LNKLST in your LPAR. It is highly recommended that LE runtime library be available via LNKLST.

```
//* DD DISP=SHR, DSN=CSF. SCSFMDDO
```

The above line is only a comment and illustrates a possible V8 dependency on accessing the IBM Cryptographic Services Facility from DDF. The ICSF library should be in LNKLST.

```
//SYSTCPD DD DISP=SHR, DSN=TCPIP NAM TCPIP. DATA
```

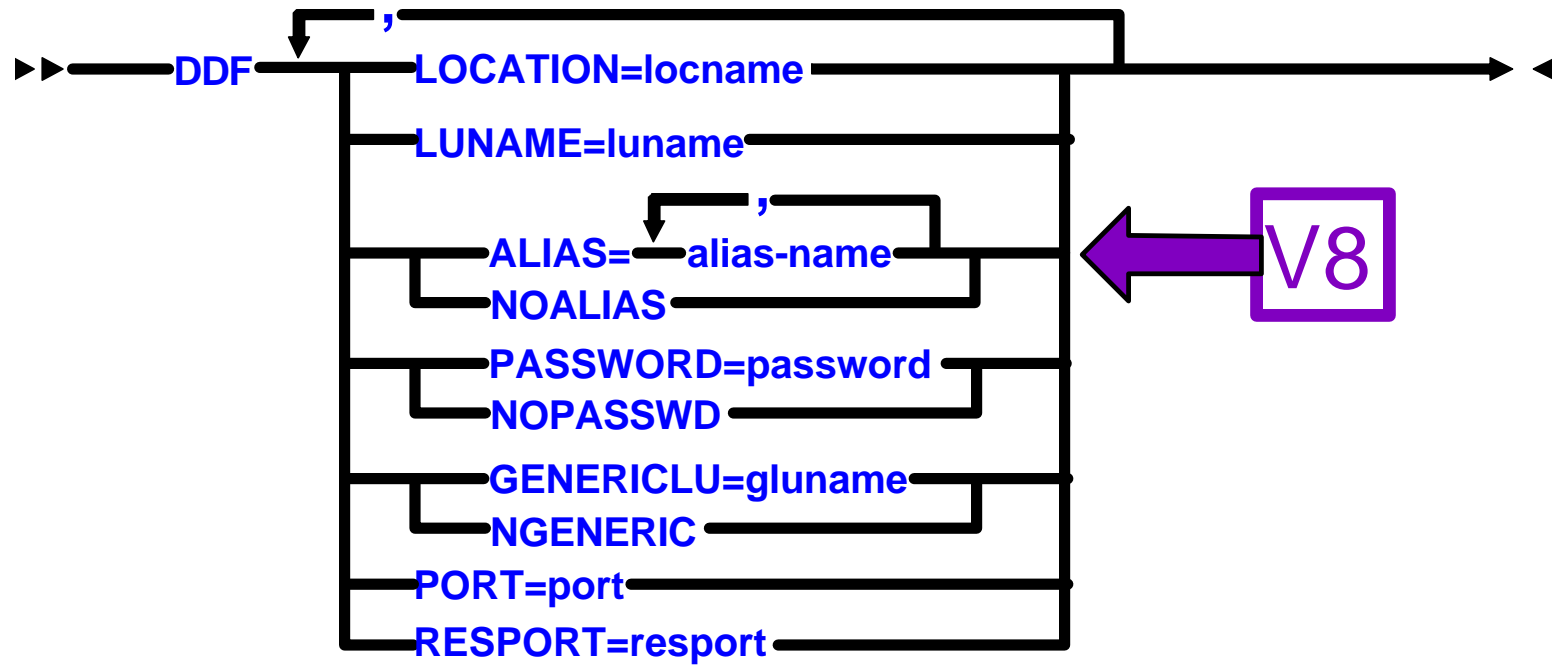
You need the above line if the default search order does not yield the correct TCP/IP resolver configuration dataset.



# DDF 101 (*continued*)

- If DB2 just a server:
  - ▶ DSNTIJSJG adds a default "blank" entry in SYSIBM.LUNAMES
  - ▶ no PLANS must be bound (DISTSERV is default plan)
  - ▶ packages must be bound at server for any remote application access
- If DB2 just a requester or both:
  - ▶ Communications DataBase (CDB) entries needed
  - ▶ PLANS still required for requesting applications
- SQL processing performed via packages
  - ▶ requesting PLANS must have PKLIST entries for remote location packages
  - ▶ TIP: specify "\*" for location name of package(s) in PKLIST (allows package access to any other system)

# Change Log Inventory (DSNJU003)



## ■ Examples:

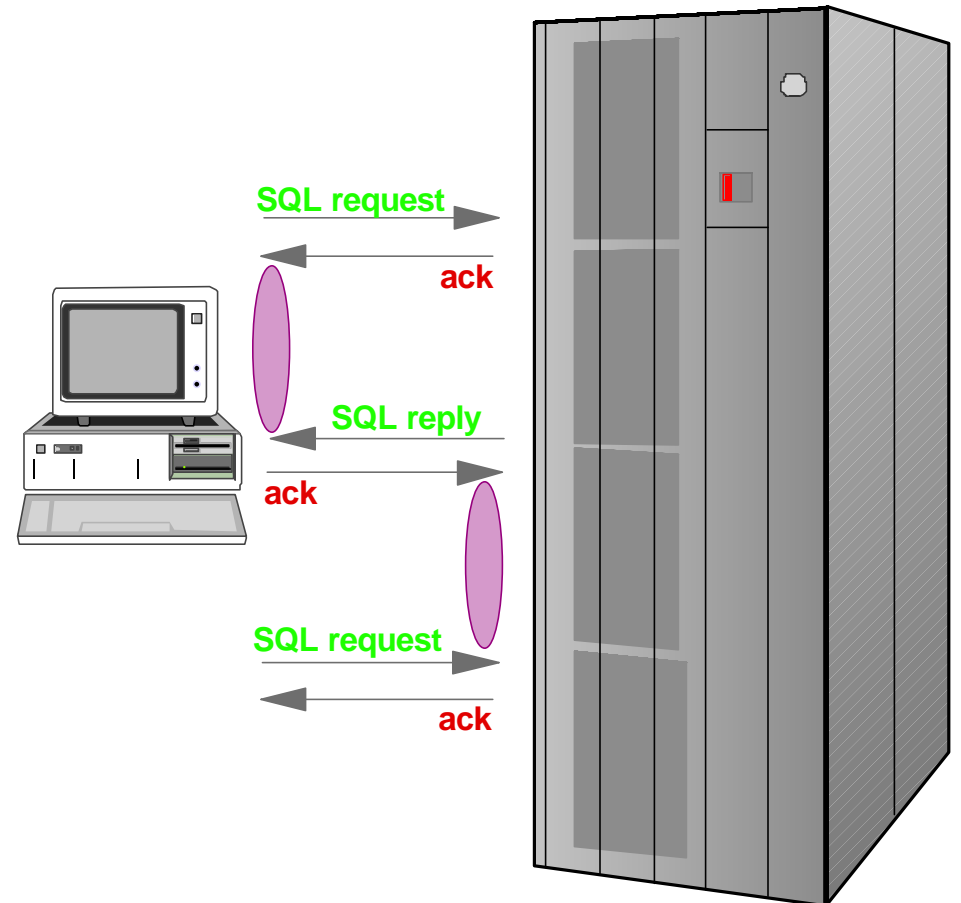
- ▶ DDF LOCATION=DSND81P,LUNAME=DSND81P,  
PORT=446,RESPORT=5020
- ▶ DDF PORT=0,RESPORT=0 (disables TCP/IP)

# DB2 Installation Parameters for DDF

<b>MAXDBAT</b>	Maximum number of concurrent DBATs (<=1999) or connections if CMTSTAT=ACTIVE
<b>CMTSTAT</b>	ACTIVE or <u>INACTIVE</u> governs whether DBATs/connections remain active across commits - accounting record cut when DBAT/connection goes inactive
<b>CONDBAT</b>	Maximum number of concurrent connections (<=150000)
<b>IDTHTOIN</b>	Idle thread timeout interval has applicability with CMTSTAT set to INACTIVE. If inflight DBAT has not received message for interval, then abort it. If DBAT has been pooled, then it is not idle and timer does not apply. If CMTSTAT set to ACTIVE, then timer applies between messages (>= <u>120</u> , 0 disables)
<b>TCPALVE R</b>	NO (leave default unless you "trust" every requester to perform authentication of userid - applies to TCP/IP only)
<b>TCPKPAL V</b>	Time value in seconds ( <u>120</u> ), ENABLE (use stack default of 2 hours), or DISABLE (no keep alive at all!!!)
<b>POOLINAC</b>	Time duration in seconds ( <u>120</u> ) that a DBAT remains in pool waiting to process a new unit-of-work request

# Dealing with TCP/IP Network Failures

- TCP/IP communication failures can cause "hung" users
- To minimize impact take one or both of these actions:
  - ▶ Specify a small value for TCPKPALV (seconds)
  - ▶ Utilize the IDLE THREAD TIMEOUT in DB2



# DB2 Installation Parameters for DDF

## *(continued)*

<b>EXTRASRV</b>	Maximum number of cursor blocks sent from server when requester indicates that multiple blocks on a single DRDA OPNQRY or CNTQRY can be processed ( $\leq 100$ ). Processing triggered at server when requester sends its value and SQL is a cursor SELECT with OPTIMIZE for N ROWS (N larger than number of rows which can fit in a 32KB block)
<b>EXTRAREQ</b>	Maximum number of cursor blocks that a requester can receive in reply to a single DRDA OPNQRY or CNTQRY request ( $\leq 100$ ). Actual value used in a connection will be the lower of two values. As of DB2 Connect V7.2 fixpak 4, it sends unlimited value.
<b>DBPROTCL</b>	<u>DRDA</u> or PRIVATE - implies default package DBPROTOCOL
<b>EXTSEC</b>	<u>YES</u> for extended security codes during connection processing
<b>RESYNC</b>	Interval to next RESYNC of Indoubts retry
<b>MAXTYPE1</b>	Applies only when using PRIVATE PROCOL - maximum number of inactive DBATs (aka type-1 inactive DBATs). If at requester thread originally was a DRDA DBAT, then value applies to DBAT when commit is processed.

# What Causes a Network Exchange?

- Most DML operations require a network exchange between client and server
  - ▶ FETCHs do not because FOR FETCH ONLY cursors can be blocked (up to 32KB blocks)
  - ▶ Updatable cursors (FOR UPDATE OF) cause row at a time processing since position must be maintained
  - ▶ Packages with CURRENTDATA YES will cause row at a time processing unless FOR FETCH ONLY specified
  - ▶ Packages with CURRENTDATA NO will have cursors blocked except "FOR UPDATE OF" cursors
- Packages with DEFER(PREPARE) will cause chained PREPARE and EXECUTE/OPEN exchange

# V8 Enhanced Network Support

- DDF Server/Requester can now support up to 2MB cursor blocks
  - ▶ normal 32KB cursor blocks with up to 100 sent from server if using OPTIMIZE FOR n ROWS
  - ▶ larger message blocks used when supporting array fetch/insert/update operations
  - ▶ V8 distributed client will support up to 64KB cursor blocks unless using dynamic scrollable cursors
- Array insert supported from V8 distributed client
  - ▶ must have fixpak 4
  - ▶ only supported via ODBC/CLI/JDBC clients

# DB2 Commands

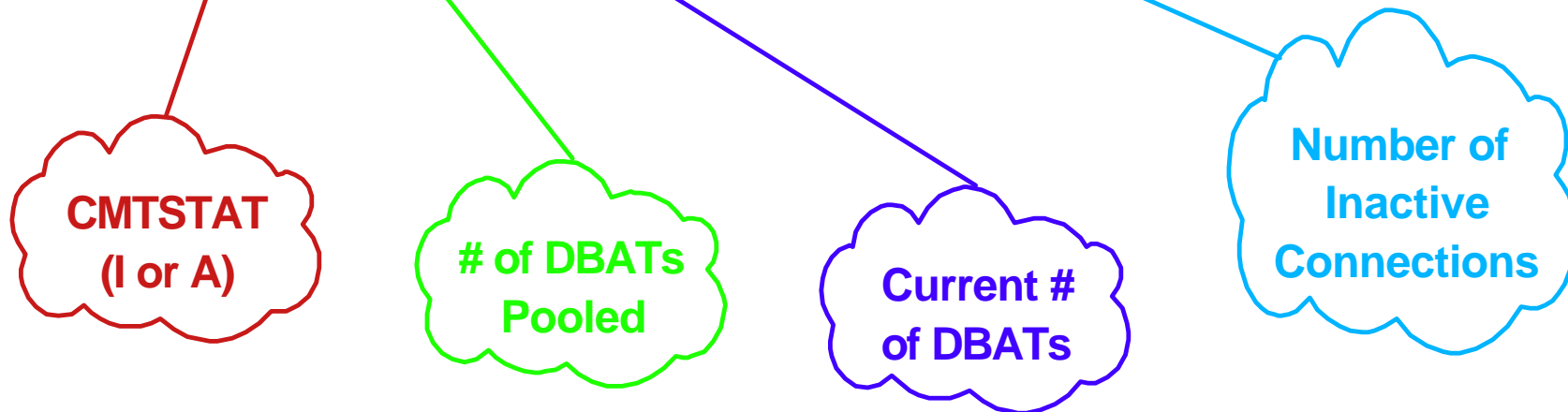
- -DISPLAY DDF DETAIL
- -DISPLAY LOCATION(...)
  - ▶ can be "\*", IP address, LUNAME, and location name
- -DISPLAY THREAD LOCATION(...)
  - ▶ limits active thread display to DBATs
- Tracing can be limited to a location
  - ▶ -START TRACE LOCATION(...)
  - ▶ -STOP TRACE LOCATION(...)
- -RESET INDOUBT IPADDR(12.34.56.78:446) or  
-RESET INDOUBT LUNAME(MYNET.MYLU)



# -DISPLAY DDF DETAIL

```

DSNL080I  - PM01 DSNLTDDF DISPLAY DDF REPORT FOLLOWS:
DSNL081I  STATUS=STARTD
DSNL082I  LOCATION                LUNAME                GENERI CLU
DSNL083I  APPLPM01                USIBMSY. APPLPM01    - NONE
DSNL084I  IPADDR                  TCPPO RT RESPO RT
DSNL085I  9. 30. 129. 247         8030                8031
DSNL086I  SQL      DOMAI N=PM01
DSNL086I  RESYNC  DOMAI N=PM01
DSNL090I  DT=I  CONDBAT= 150000  MDBAT= 1000
DSNL092I  ADBAT=      4  QUEDBAT=      0  INADBAT=      0  CONQUED=
0
DSNL093I  DSCIBAT=      4  INACONN=      4
DSNL099I  DSNLTDDF DISPLAY DDF REPORT COMPLETE
    
```



# -DISPLAY LOCATION(...)

DSNL200I - DISPLAY LOCATION REPORT FOLLOWS -

LOCATI ON	PRDI D	LI NKNAME	REQUESTERS	SERVERS	CONV
9. 30. 27. 21		9. 30. 27. 21	0	1	1
DB2A	DSN06010	124. 38. 54. 16	3	0	3
DB2A	DSN06010	124. 38. 54. 17	5	0	5

DISPLAY LOCATION REPORT COMPLETE

# -DISPLAY THREAD LOCATION(...)

DSNV401I - D6AD DISPLAY THREAD REPORT FOLLOWS -

DSNV402I - D6AD ACTIVE THREADS -

NAME	ST A	REQ ID	AUTHID	PLAN	ASID	TOKEN
SERVER	SP	317	db2spb.exe	SMITHHJ	DISTSERV	0095 515

V437- WORKSTATION=GEIDI PRM, USERID=smi thhj ,  
APPLICATION NAME=db2spb.exe

V436- PGM=DEBUG. , SEC=0, STMT=0

V429 CALLING PROCEDURE=DEBUG . STCKLVLD2 ,  
PROC=D6ADSPAS, ASID=00E9, WLM\_ENV=

V445- COBC42D6. L206. 094023173351=515 ACCESSING DATA FOR

9. 15. 95. 91

V447- - LOCATION SESSID A ST TIME

V448- - 9. 15. 95. 91 8024: 1618 W S2 0120410344766

DISCONN	DA *	0	NONE	NONE	DISTSERV	006A 331
---------	------	---	------	------	----------	----------

V471- USIBMSTM. STM4T01A. B802A34B1A1C=331

DISPLAY ACTIVE REPORT COMPLETE

# Communications Database

## *SYSIBM.LOCATIONS*

LOCATION	LINKNAME	IBMREQD	PORT	TPN	DBALIAS

## *SYSIBM.IPNAMES*

LINKNAME	SECURITY_OUT	USERNAMES	IBM_REQD	IPADDR

## *SYSIBM.IPLIST*

LINKNAME	IPADDR	IBM_REQD

V8

## *SYSIBM.USERNAMES*

TYPE	AUTHID	LINKNAME	NEWAUTHID	PASSWORD	
IBM_REQD					

Entries needed when DB2 requests access to other DB2s

# **SYSIBM.LOCATIONS**

- **LOCATION** -- the name of the remote DBMS
- **LINKNAME** -- identifies row in **SYSIBM.IPNAMES** or **SYSIBM.LUNAMES**
  - For DRDA, **SYSIBM.IPNAMES** is searched first
  - For PRIVATE, **SYSIBM.LUNAMES** is searched
- **PORT** -- DRDA TCP/IP port or service name
- **TPN** -- LU6.2 transaction program name for DRDA SNA connections
  - ▶ (defaults to '07F6C4C2'x) can also use DB2DRDA
- **DBALIAS (V8)** -- used in connection request to remote server (should only be used when accessing DB2 UDB on Linux/Unix/Windows)

# SYSIBM.IPNAMES

- LINKNAME key of SYSIBM.LOCATIONS row
- SECURITY\_OUT controls outbound security
  - 'A' -- send already verified (userid only)
  - 'P' -- send userid and password
  - 'R' -- send RACF passticket
- USERNAMES defines userid translation reqmts
  - blank -- no translation
  - 'O' -- outbound translation
  - **NO INBOUND TRANSLATION FOR TCP/IP USERS!**
- IPADDR -- partner's domain name or IP address

# **SYSIBM.IPLIST (V8)**

- LINKNAME key of SYSIBM.LOCATIONS row and corresponding IPNAMES row
- IPADDR -- a data sharing group's member specific IP address
  - ▶ multiple rows in IPLIST can exist for each member of the data sharing group that you want the list to access
  - ▶ each IPADDR must be different than the others
  - ▶ IPADDR in IPNAMES row must be blank

# Example CDB Definition

## SYSIBM.LOCATIONS

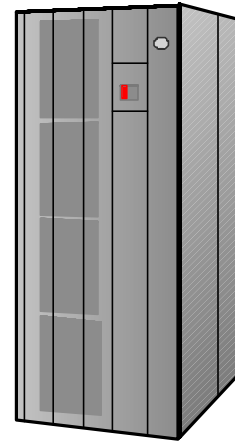
LOCATION	LINKNAME	PORT
'DB2A'	'LINK1'	'446'

## SYSIBM.IPNAMES

LINKNAME	SECURITY_OUT	IPADDR	USERNAMES
'LINK1'	'P'	'12.36.114.5'	'O'

## SYSIBM.USERNAMES

TYPE	LINKNAME	NEWAUTHID	PASSWORD
'O'	'LINK1'	'DB2AUSR'	'DB2APW'



LOCATION/IP Address:

DB2A

12.36.114.5

Ports:

DB2A

– 446 for SQL

– 5020 for resync



# Establishing a Connection from DB2 UDB for z/OS Requester

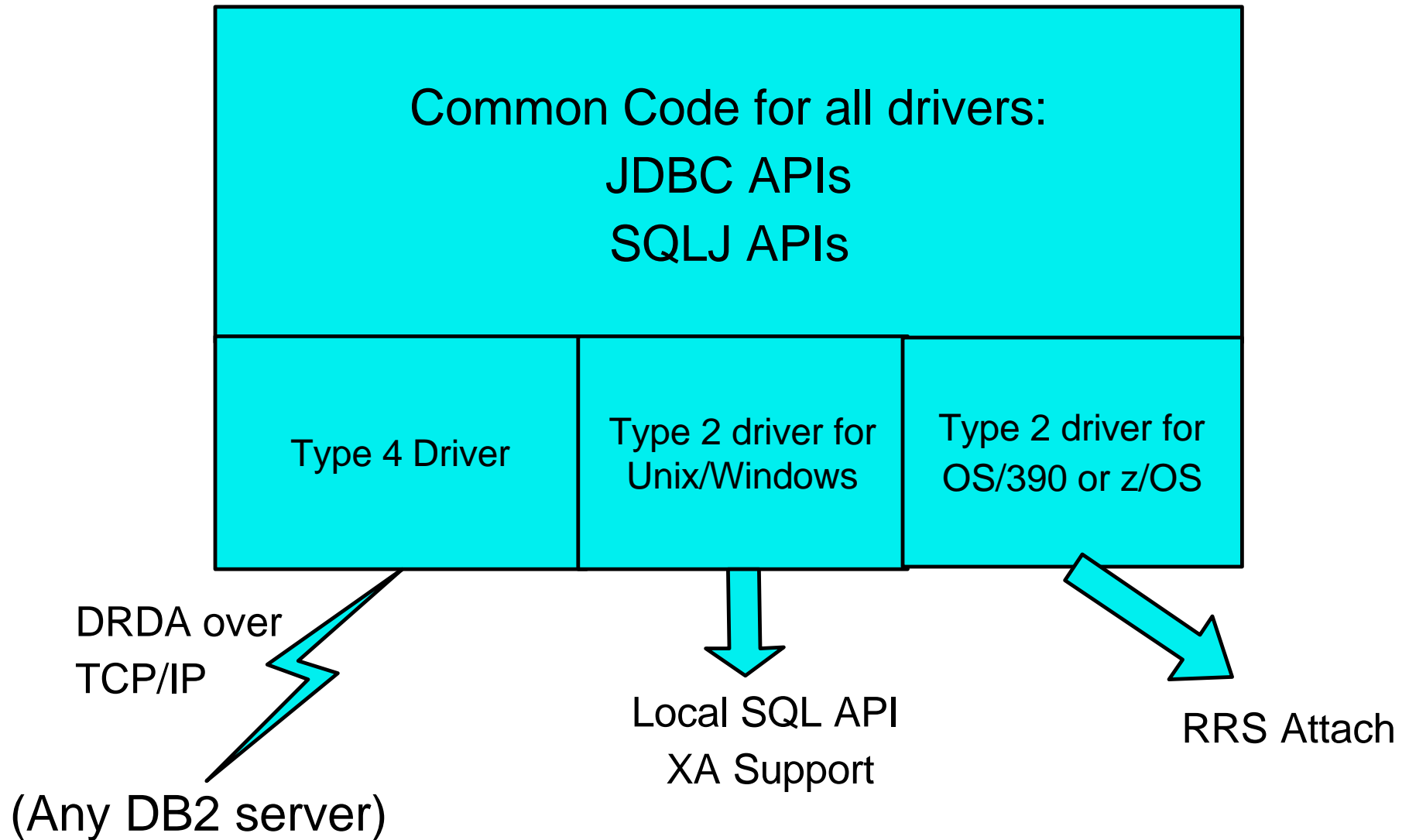
- Application connects to server:
  - ▶ EXEC SQL CONNECT TO DB2A
  - ▶ EXEC SQL CONNECT TO DB2A USER USER1 USING PW1
  - ▶ implicitly via CURRENTSERVER parameter of BIND PLAN
- Objects can be referenced via 3-part name or ALIAS (no connect to server required but implied):
  - ▶ SELECT COUNT(\*) FROM DB2A.SYSIBM.SYSTABLES
  - ▶ or:  
CREATE ALIAS DB2ACAT.SYSTABLES  
FOR DB2A.SYSIBM.SYSTABLES  
  
SELECT COUNT(\*) FROM DB2ACAT.SYSTABLES
  - ▶ **Note:** dynamic SQL/DRDA requires an additional two-part name ALIAS at server in the above example

# DB2 Connect Client Configuration

## Example ...

- DB2 Connect definition statements
  - ▶ db2 catalog tcpip node **db2a** remote 12.36.114.5 server 446
  - ▶ db2 catalog db **db2a** as **db2a** at node **db2a**  
**authentication server**
  - ▶ db2 catalog dcs db **db2a** as db2a
- Client configuration assistant (GUI)
- If V8 client AND db directory entry does not specify authentication, **server\_encrypt** will be used
- Now connect to DB2
  - ▶ db2 connect to **db2a** user xyz using pw

# DB2 Java Client Architecture



# Universal Driver for SQLJ and JDBC T4

## Example

- Access to DB2 specified in URL passed into GetConnection method

- Example URLs:

- ▶ `jdbc:db2://stplex4a.svl.ibm.com:446/MYDB2ZOS`

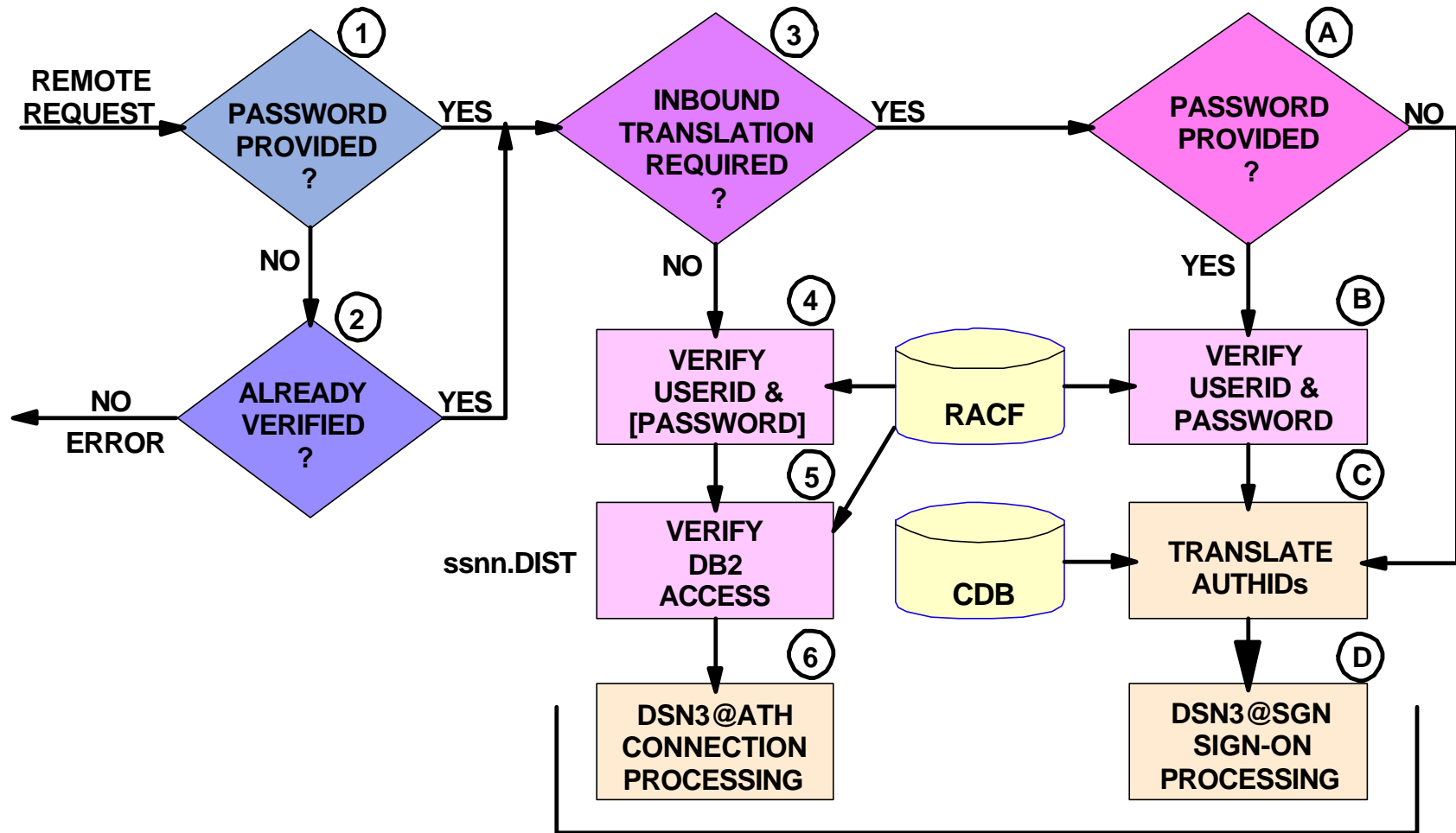
or if connection requested through DB2 Connect V8 EE server listening on port 50000 at domain name, ender.svl.ibm.com, and dbalias is MYDB2ZOS

- ▶ `jdbc:db2://ender.svl.ibm.com:50000/MYDB2ZOS`

# V8 TCP/IP SECURITY\_OUT Enhancements

- New values for column
  - ▶ 'D' - like 'A' but userid/data sent encrypted
  - ▶ 'E' - like 'P' but userid/pw/data sent encrypted
  - ▶ Must be V8 NFM and ICSF accessible or connect attempt will fail
  - ▶ 'D' and 'E' can only be used to access V8 server with access to ICSF
- Existing value changed
  - ▶ 'P' - send userid/pw encrypted
  - ▶ Must be V8 NFM and ICSF accessible or userid/pw will default to being sent in the clear
  - ▶ If accessing pre-V8 server, or ICSF not accessible, then BSAFE routines used for decryption

# Controlling Remote Access - Inbound

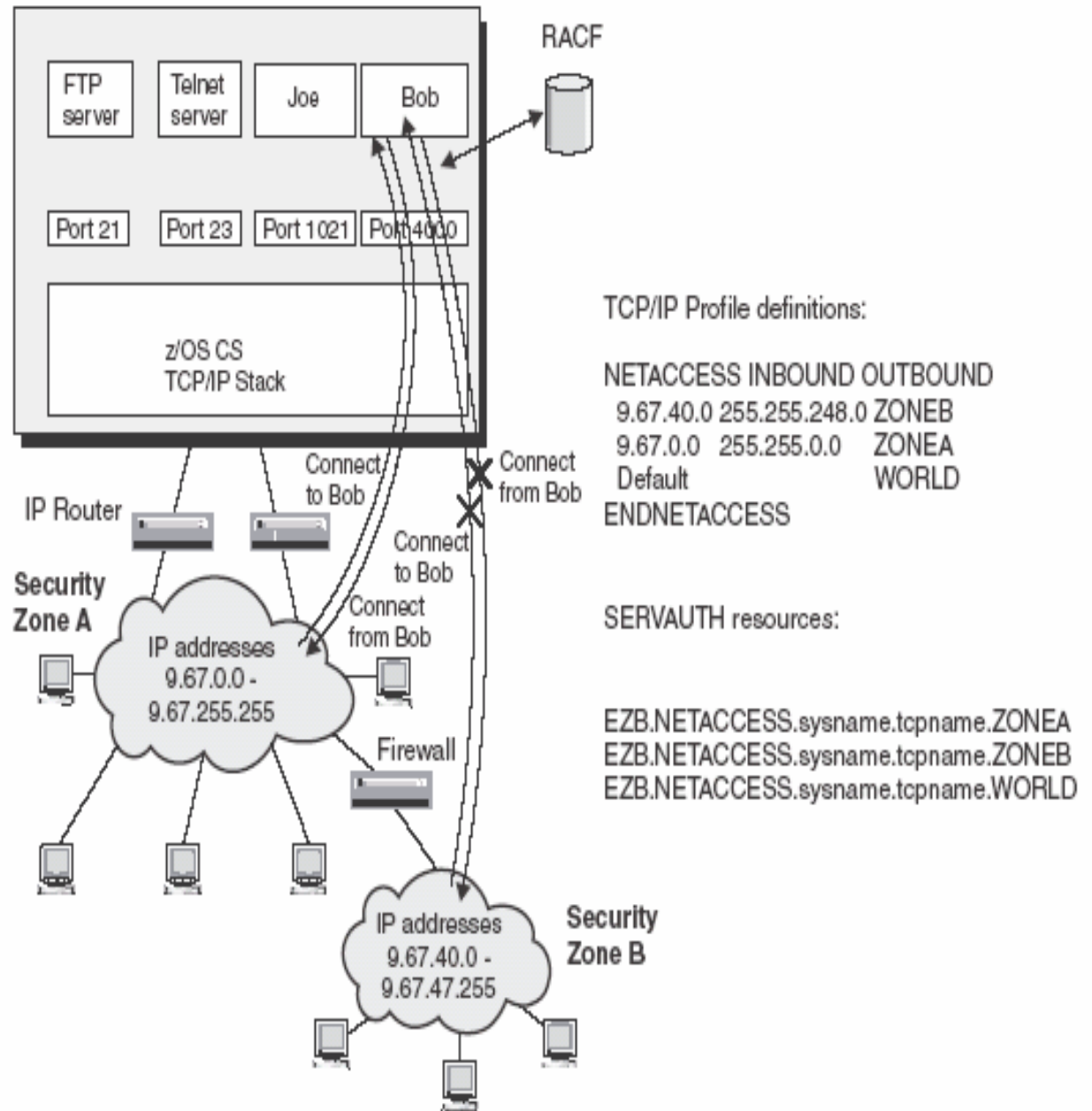


ssnn.DIST

- ESTABLISH SECONDARY AUTHIDs
- CHANGE PRIMARY AUTHID
- ACCEPT/REJECT CONNECTION

# V8 TCP/IP Port-of-Entry Support

- z/OS V1R5 required
- TCP/IP NETACCESS defined
- SERVAUTH class active and profiles defined
  - ▶ don't forget DDF address space userid needs to be able to read profiles
- After userid authenticated/verified, and SERVAUTH profile passed in from TCP/IP, userid is authenticated to use SERVAUTH profile



# Summary

- Basic discussion of DDF
- Advanced discussion in Session II
  - ▶ Path of a Distributed Task
  - ▶ What is inactive, pooled, etc.?
  - ▶ Statistics and Accounting in a Distributed World
  - ▶ Controlling DDF Work
  - ▶ Data Sharing and Distributed
  - ▶ DB2 and Dynamic Virtual IP Addressing
  - ▶ DB2 Connect and Sysplex
  - ▶ Subsetting a Data Sharing group