SWG BetaWorks

DB2 9 for z/OS

Technical Education Series

“Security”

Important Disclaimer

THE INFORMATION CONTAINED IN THIS PRESENTATION IS PROVIDED FOR INFORMATIONAL PURPOSES ONLY.

WHILE EFFORTS WERE MADE TO VERIFY THE COMPLETENESS AND ACCURACY OF THE INFORMATION CONTAINED IN THIS PRESENTATION, IT IS PROVIDED “AS IS”, WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED.

IN ADDITION, THIS INFORMATION IS BASED ON IBM’S CURRENT PRODUCT PLANS AND STRATEGY, WHICH ARE SUBJECT TO CHANGE BY IBM WITHOUT NOTICE.

IBM SHALL NOT BE RESPONSIBLE FOR ANY DAMAGES ARISING OUT OF THE USE OF, OR OTHERWISE RELATED TO, THIS PRESENTATION OR ANY OTHER DOCUMENTATION.

NOTHING CONTAINED IN THIS PRESENTATION IS INTENDED TO, OR SHALL HAVE THE EFFECT OF:

- CREATING ANY WARRANTY OR REPRESENTATION FROM IBM (OR ITS AFFILIATES OR ITS OR THEIR SUPPLIERS AND/OR LICENSORS); OR
- ALTERING THE TERMS AND CONDITIONS OF THE APPLICABLE LICENSE AGREEMENT GOVERNING THE USE OF IBM SOFTWARE.

Agenda

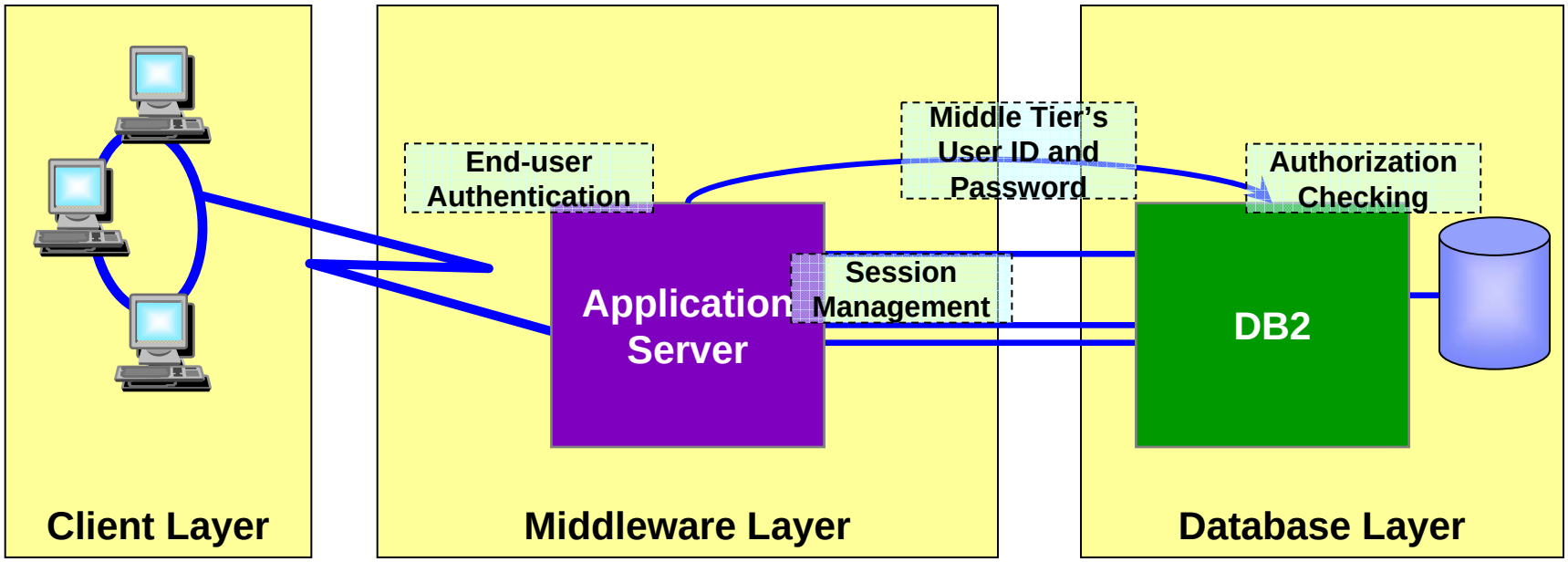
- **Trusted Contexts, Roles and SSL**
  - Current Three-tier Authorization
  - Trusted Contexts – A Quick Overview
  - Trusted Context Attributes
  - Defining A Trusted Context
  - Establishing A Trusted Connection
  - Authorization ID Switching
  - Performing Actions On Behalf Of Other Users
  - Roles and Context-specific Privileges
  - DB2 Support for Roles And Trusted Contexts
  - Trusted Contexts And Object Ownership
  - Authorization ID Checking
  - SSL Support
  - Migration Consideration

SWG BetaWorks

Trusted Contexts, Roles, and SSL Support

DB2 9 for z/OS Technical Education Series

Current Authentication in a Three-Tier Architecture



- **A three-tiered application model with DB2 as the database server:**
  - The middle layer authenticates users running client applications.
  - It also manages interactions with the database server.
  - The middle layer's user ID and password are used for database authentication.
  - The privileges of the associated authorization id are checked when accessing the database, including all access on behalf of all end-users.

Three-tier Authentication – The Issues

- **Problems with the current implementation:**
  - Loss of end-user identity.
  - Loss of control over end-user access of the database.
  - Diminished accountability.
  - The middleware server's AUTHID needs the privileges to perform *all* requests from *all* end-users.
  - If the middleware server's security is compromised, so is that of the database server.
- **Problems with establishing a new connection using the end user's ID and password:**
  - Performance overhead:
    - Creating a new connection to the database server;
    - Re-authenticating the end-user at the database server
  - Not possible for servers without access to end-user credentials.

Trusted Contexts – An Introduction

- A ***TRUSTED CONTEXT*** establishes a trusted relationship between DB2 and an external entity such as a middleware server. For example:
  - WebSphere Application Server
  - Lotus Domino
  - SAP NetWeaver
  - PeopleSoft V7

- A set of *trust attributes* is evaluated to determine if a specific context is to be trusted.
- A trusted context allows the external entity to use a database connection under a different user ID without the database server authenticating that ID.
- It also allows an AUTHID to acquire database privileges associated with that trusted context, and not available outside it, via a ***ROLE***.

Trusted Context Attributes

- A trusted context is a database entity based upon a system authorization ID and connection trust attributes.
- The system AUTHID is the primary AUTHID used to establish the trusted connection.
- Remote connection trust attributes:
  - SYSTEM AUTHID
  - ADDRESS
  - SERVAUTH
  - ENCRYPTION
- Local connection trust attributes:
  - SYSTEM AUTHID
  - JOBNAME

Local And Remote Trusted Context Attributes

- Remote connection trust attributes:
  - SYSTEM AUTHID – the system user ID provided by e.g. a middleware server.
  - ADDRESS – IP address or domain name (restricted to TCP/IP only).
  - SERVAUTH – a resource in the RACF SERVAUTH class.
  - ENCRYPTION – minimum level of encryption for the connection.
- Local connection trust attributes:
  - SYSTEM AUTHID is typically derived from:
    - Started task (RRSAF) – JOB statement USER or RACF USER
    - TSO – TSO logon ID
    - BATCH – JOB statement USER
  - JOBNAME is derived from:
    - Started task (RRSAF) – JOB or started class name
    - TSO – TSO logon ID
    - BATCH – JOB name

SERVAUTH Profiles

- Profiles in the SERVAUTH class represent IP addresses

TCP/IP Profile definitions:

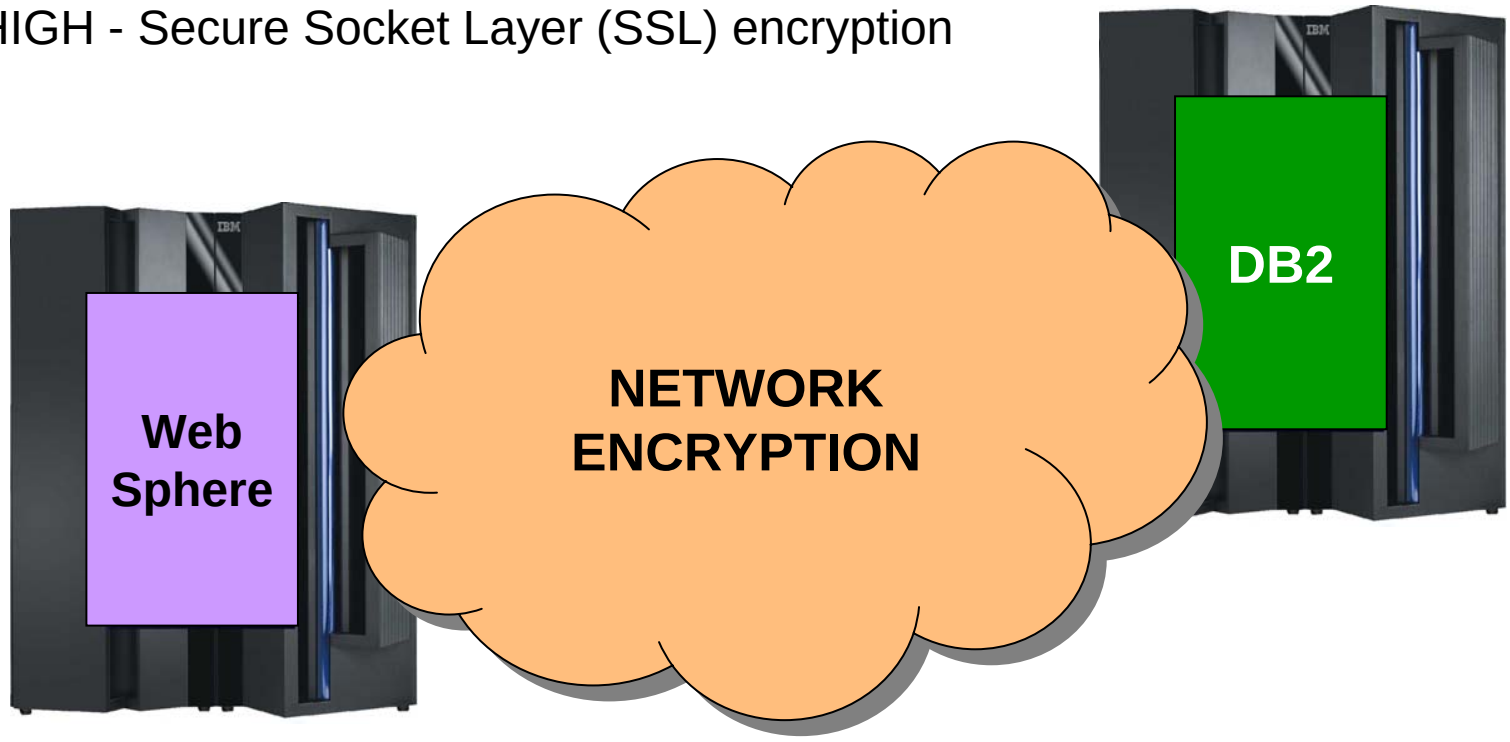
```
NETACCESS INBOUND OUTBOUND
9.67.40.0 255.255.248.0 ZONEB
9.67.0.0  255.255.0.0   ZONEA
Default                 WORLD
ENDNETACCESS
```

RACF SERVAUTH Resources:

```
EZB.NETACCESS.sysname.tcpname.ZONEA
EZB.NETACCESS.sysname.tcpname.ZONEB
EZB.NETACCESS.sysname.tcpname.WORLD
```

The ENCRYPTION Attribute

- Minimum encryption level of the data stream for the connection. Supported values are:
  - NONE - No encryption. The default.
  - LOW - DRDA data stream encryption
  - HIGH - Secure Socket Layer (SSL) encryption

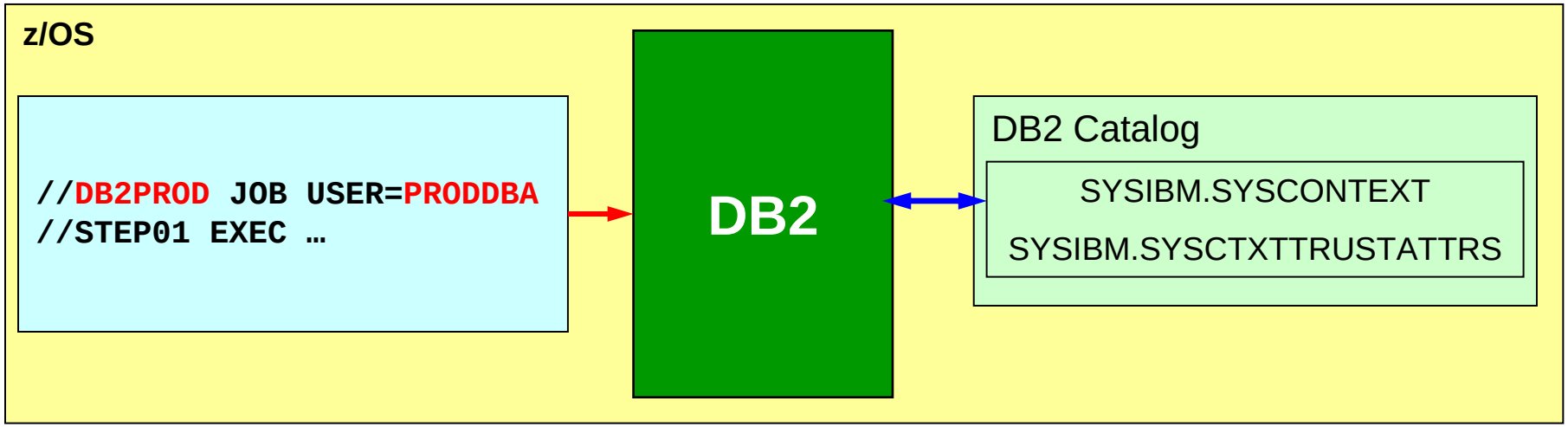


Defining A Trusted Context

- New DDL statements to add, alter or drop trusted contexts.
- New catalog tables SYSIBM.SYSCONTEXT, and SYSIBM.SYSCTXTTRUSTATTRS.
- Each SYSTEM AUTHID can only be associated with a single trusted context.

```
CREATE TRUSTED CONTEXT CTX1
BASED UPON CONNECTION USING SYSTEM AUTHID WASADM1
ATTRIBUTES (ADDRESS '9.67.40.204', ADDRESS '9.67.40.219',
SERVAUTH 'EZB.NETACCESS.ZOSV1R5.TCPIP.ZONEA')
ENABLE;

CREATE TRUSTED CONTEXT CTX2
BASED UPON CONNECTION USING SYSTEM AUTHID WASADM2
ATTRIBUTES (JOBNAME 'WASPROD')
ENABLE;
```

Establishing A Trusted Connection (Local DB2 Server)



- With DB2 as a server, upon receipt of a local connection request DB2:
  - Performs standard authorization checking and invokes the connection exit.
  - Tries to match the primary AUTHID with a trusted context SYSTEM AUTHID.
  - Checks that the JOB name and the JOBNAME attribute match.
  - If the SYSTEM AUTHID has a SECURITY LABEL, then validates it with RACF (MLS).
- If validation is successful the connection is established as trusted. Otherwise, it is established as a normal "untrusted" connection.

Establishing A Trusted Connection (Remote DB2 Server)

- With **DB2 as a server**, upon receipt of a remote connection request DB2:
  - Performs standard authorization checking and invokes the connection exit.
  - Tries to match the primary AUTHID with a trusted context SYSTEM AUTHID.
  - Attempts the following:
    - If a SERVAUTH attribute is defined for the trusted context and a RACF SERVAUTH profile name for the TCP/IP resource exists, matches the two.
    - If there is no SERVAUTH, or the SERVAUTH and the trusted context names don't match, matches the remote client address with the trusted context ADDRESS attribute.
    - Checks that the encryption level used matches the ENCRYPTION attribute.
  - If the SYSTEM AUTHID has a SECURITY LABEL, then validates it with RACF (MLS).
- If validation is successful the connection is established as trusted. Otherwise, the connection is established as a normal "untrusted" connection.

Establishing A Trusted Connection (DB2 Requester)

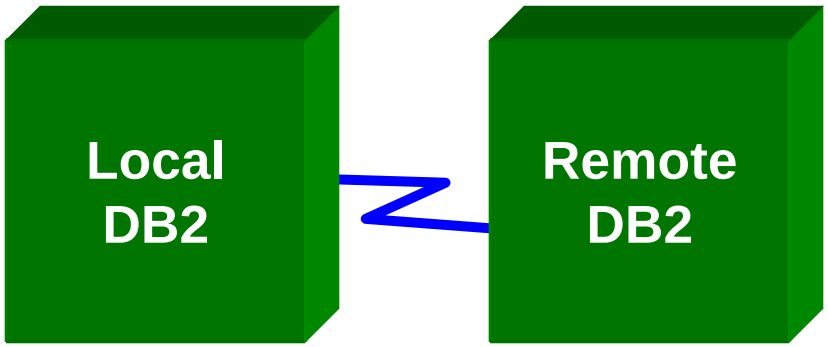
SYSIBM.LOCATIONS

LOCATION	LINKNAME	...	TRUSTED
DB2LOC	DB2LINK		Y

Establishing a trusted connection, **as a requester**, to a remote DB2 subsystem

SYSIBM.IPNAMES

LINKNAME	SECURITY_OUT	USERNAMES	...
DB2LINK	'E', 'P' or 'R'	S	



SYSIBM.USERNAMES

TYPE	AUTHID	LINKNAME	...
S	FRED	DB2LINK	

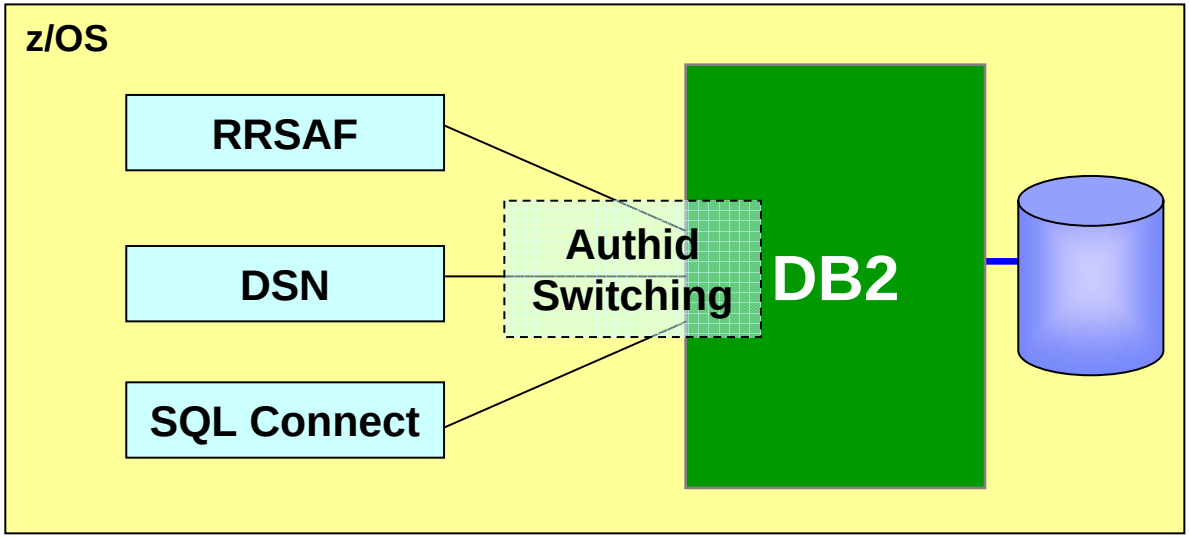
Authid Switching

- An established trusted connection can be used with a different user id.
- To allow this, the specific user must be added to the trusted context.
  - Can be PUBLIC.
- WITH/WITHOUT AUTHENTICATION specifies whether authentication is required when switching to a different AUTHID.
- Switching only occurs on a transaction boundary.
- New catalog table SYSIBM.SYSCONTEXTAUTHIDS stores the AUTHIDs that can be used in a trusted connection.

```
CREATE TRUSTED CONTEXT CTX1
BASED UPON CONNECTION USING SYSTEM AUTHID WASADM1
DEFAULT ROLE CTXROLE
ATTRIBUTES (ADDRESS '9.67.40.219')
ENABLE
WITH USE FOR JOE ROLE JROLE;
```

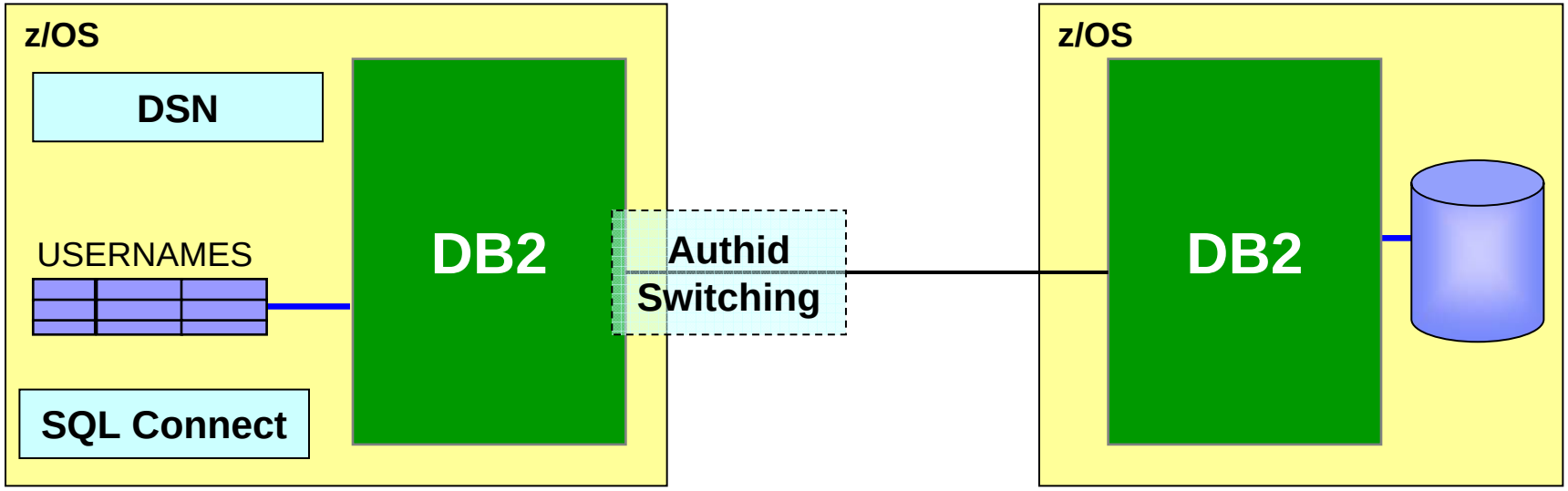
Authid Switching – Local Processing

- Allowing a trusted connection to be used by a different user at a local DB2:
  - RRSAF: the SIGNON function in CALL DSNRLI.
  - The DSN Command processor: the new ASUSER option.
  - SQL CONNECT, via the USER and USING clauses (only locally).
- In all cases, if the primary AUTHID does not have access to the trusted context, then the connection request fails and returns to an unconnected state.



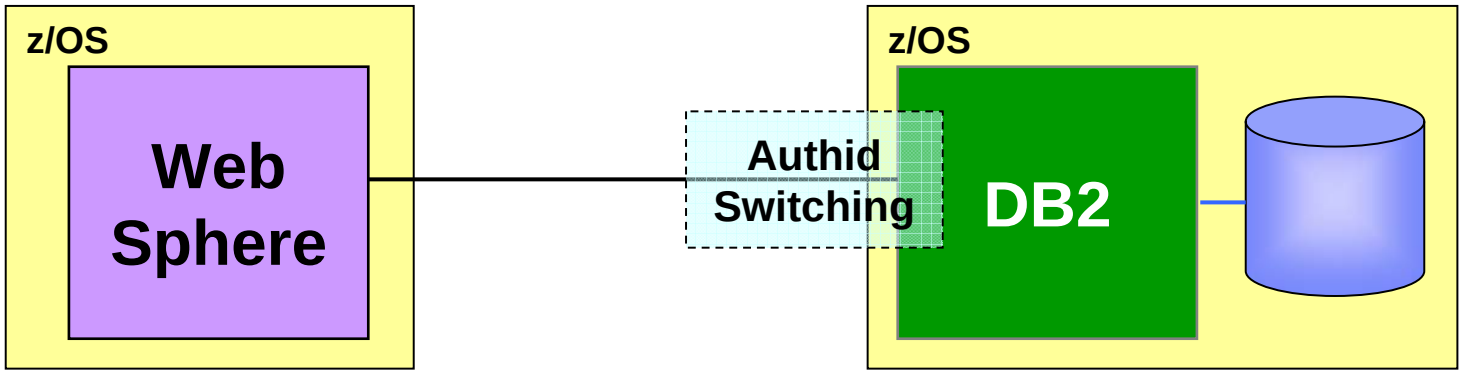
Authid Switching, Remote Processing (DB2 Requester)

- As a requester, DB2 automatically switches the user on a trusted connection to the primary AUTHID when:
  - The SYSTEM AUTHID differs from the primary AUTHID associated with the application user.
  - The SYSTEM AUTHID differs from the AUTHID in the SQL CONNECT statement with the USER and USING clauses.
  - Outbound translation is required for the primary AUTHID and SYSTEM AUTHID row is defined in the SYSIBM.USERNAMES table.



Authid Switching, Remote Processing (DB2 Server)

- When DB2, as a server, receives a request to switch users it:
  - Calls the connection exit, which associates AUTHID set and an SQL ID with the remote request, replacing the previous ones.
  - Determines if the primary AUTHID is allowed to use the trusted connection: if WITH AUTHENTICATION is specified, an authentication token is required.
  - Performs SECURITY LABEL verification for the new user ID.
  - Initializes the connection, creating a 'clean' environment, e.g. open cursors are closed, temporary table information is dropped.
  - If the primary AUTHID is not allowed to use the trusted connection or SECURITY LABEL verification fails, then the connection state is *unconnected*.



Performing Actions On Behalf Of Other Users

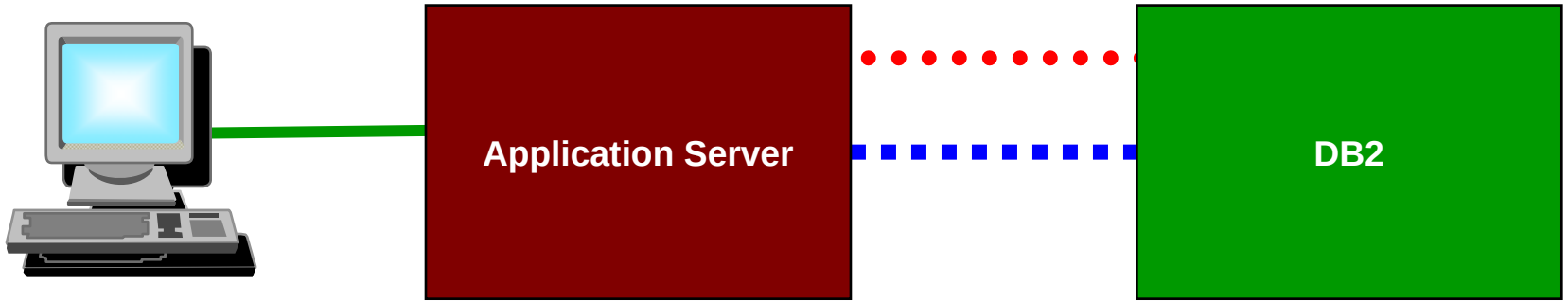

```
  CREATE TRUSTED CONTEXT CTX1
  BASED UPON CONNECTION
  USING SYSTEM AUTHID PRODDBA
  ATTRIBUTES (JOBNAME 'PRODDBA1')
  ENABLE
  WITH USE FOR PRODOWNR;

//PRODDBA1 JOB  USER='PRODDBA'
//IKJEFT1B EXEC PGM=IKJEFT1B
//SYSTSPRT DD   SYSOUT=*
//SYSPRINT DD   SYSOUT=*
//SYSTSIN  DD   *
  DSN SYSTEM(DB1P) ASUSER(PRODOWNR)
  END
//SYSIN    DD   *
  CREATE VIEW PRODVIEW AS SELECT ... ;
  COMMIT ;
  GRANT SELECT ON PRODVIEW TO PUBLIC;
  COMMIT;
//
```

Roles and Context-specific Privileges

- Roles provide the flexibility to grant privileges to an AUTHID only when the user is connected via a trusted context.
- They greatly simplify management of authorization.
- An individual role can be defined for any AUTHID using the trusted connection, in which case the user inherits the privileges granted to the individual role.
- Where there is no individual role, any AUTHID using a trusted context inherits the privileges of the trusted context's default role, if defined.

```
CREATE TRUSTED CONTEXT CTX1
BASED UPON CONNECTION USING SYSTEM AUTHID WASADM1
DEFAULT ROLE CTXROLE
ATTRIBUTES (ADDRESS '9.67.40.219')
ENABLE
WITH USE FOR JOE ROLE JROLE;
```

Connections, SQL Processes And Authids



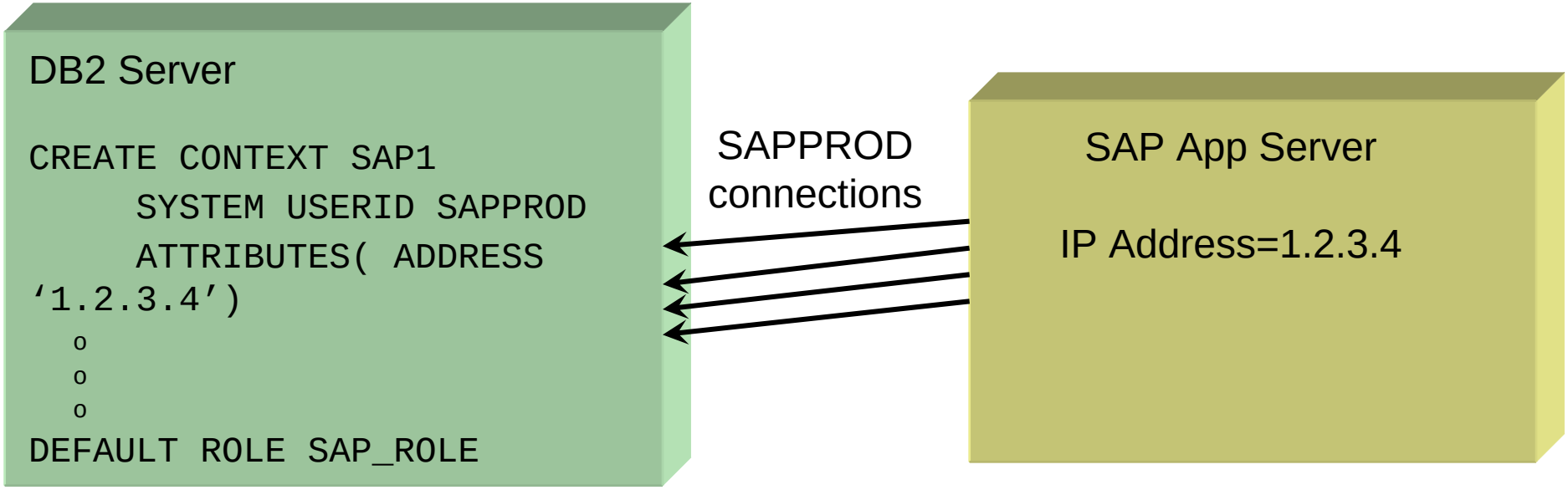
User ID flows to Application Server

Application Server establishes trusted connection to DB2 using primary AUTHID derived from SYSTEM AUTHID

SQL issued on behalf of end user runs with privileges of:
Role assigned to primary AUTHID, if any;
If none, then default role for trusted context, if any;
CURRENT SQLID;
Primary AUTHID and secondary AUTHIDs, if any

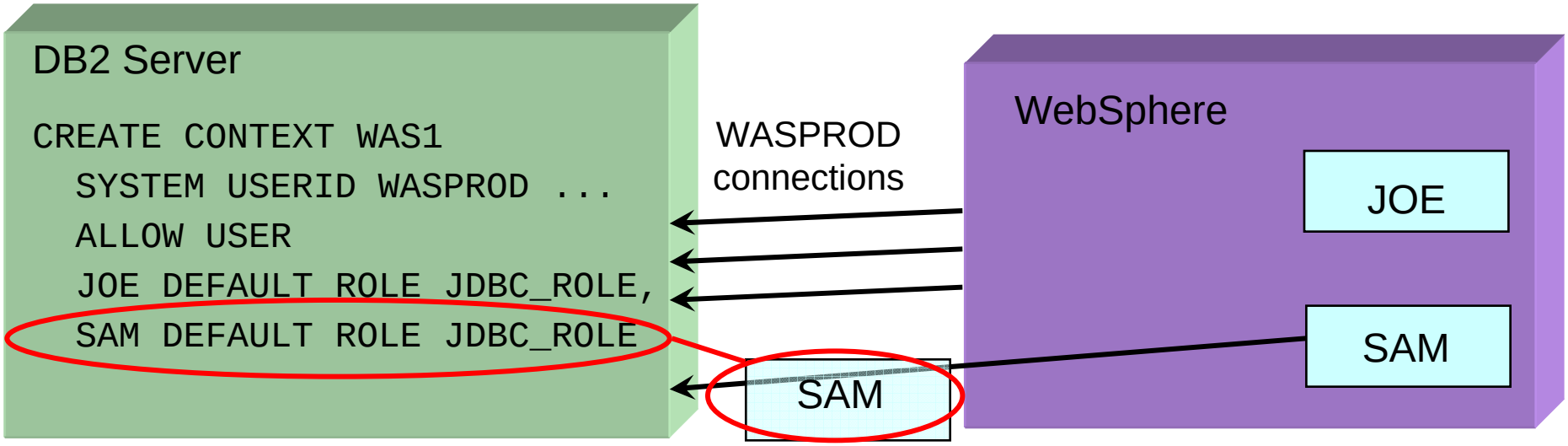
Example 1: Securing An Application Server

- Most existing application servers connect to DB2 using userid/password pairs:
  - Significant exposure if someone steals the userid/password!!!
- Trusted Context and ROLEs can be used to limit exposure:
  - GRANTs to SAP_ROLE can be restricted so that they are only valid when used by a valid SAP app server IP address.
- No change required to the code in the application server.



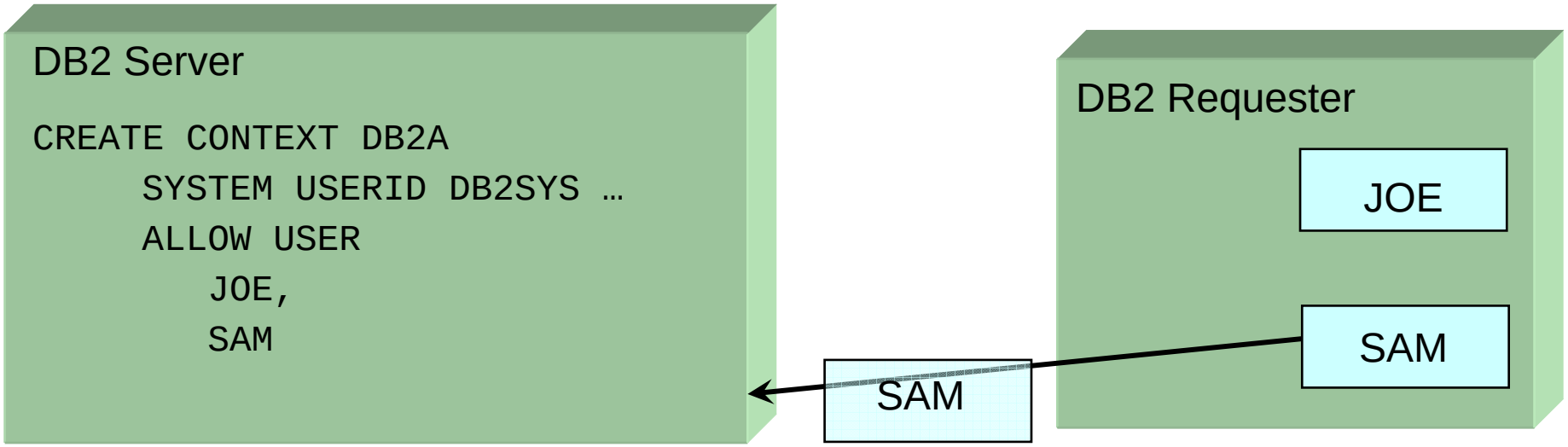
Example 2: Dynamic SQL Auditing

- Better auditing controls:
  - GRANT dynamic SQL privileges to a ROLE
  - End user identity can be delegated directly to DB2 without granting dynamic SQL privileges directly to the end user
  - End user passwords can be optional.
  - No added complexity for administration of GRANTs, while retaining the ability to audit the end user's identity!



Example 3: Roles, Trusted Context and Already-Verified DRDA

- Can be used to establish already-verified TCP/IP connections:
  - Improves ability to replace SNA connections with TCP/IP
  - Communication Database is used to identify trusted connections and specify “system userid” for the Trusted Context
  - End user identity is automatically propagated from one DB2 system to the other.



Example 4: Auditing DBA Activities

- Many sites need to be able to audit DBA access to sensitive customer data. DB2 9 for z/OS can help by enabling an auditable DBA process:

1. Grant DBA privileges to a ROLE
2. Start audit trace for that ROLE
3. When a DBA needs to perform a system change:
   - Use Trusted Context to assign DBA ROLE to person
   - DBA is given request and performs activity
   - Revoke Trusted Context
4. Have another person review the audit trace

DB2 Support for Roles

- New DDL statements.
- New Catalog tables SYSIBM.SYSROLES and SYSIBM.OBJROLEDEP.

```
CREATE ROLE CTXROLE;

CREATE TRUSTED CONTEXT CTX1
BASED UPON CONNECTION USING SYSTEM AUTHID WASADM1
DEFAULT ROLE CTXROLE
ATTRIBUTES (ADDRESS '9.67.40.219')
ENABLE;
```

- GRANT and REVOKE are extended

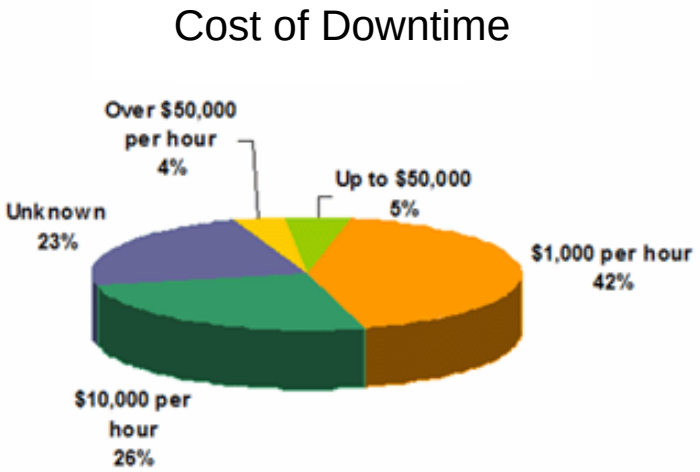
```
GRANT SELECT ON T1 TO ROLE CTXROLE;

GRANT BIND ON PLAN DSN9PLN TO ROLE CTXROLE;
```

New And Modified SQL Statements

- ALTER TRUSTED CONTEXT
  - Can alter, add or drop attributes e.g. SYSTEM AUTHID.
  - Can add, replace or drop USE FOR options for an authorization name.
- COMMENT ON
- CREATE ROLE
- CREATED TRUSTED CONTEXT
- DROP
- GET DIAGNOSTICS
  - A new value for DB2_AUTHENTICATION_TYPE, ‘T’, indicates trusted context authentication.
- GRANT
- REVOKE
  - The ROLE keyword is required when granting to or revoking from roles.

Trusted Contexts And Object Ownership

- Outside trusted contexts and roles, object ownership is tied to a user.
- When a user creates an object, they become its owner.
- To remove the privileges of that user on the object, it has to be dropped; all grants associated with it are revoked.
- The object then has to be recreated and the privileges re-granted.
- If the object owner is a role, removing privileges from the end-user will not require the object to be dropped and recreated.



Trusted Contexts And Object Ownership (cont.)

- Role ownership allows tighter security controls: e.g. DBAs only exercise privileges when performing approved activities via a trusted context and role.
- When a trusted context has a default role, the role becomes the owner of created objects, if ROLE AS OBJECT OWNER is specified.
- When a role is defined as the object owner, then it must have all the privileges necessary to create the object.
- If ROLE AS OBJECT OWNER is not specified, there is no change in determining object ownership.
- If a role owns a created object, then the user inheriting the privileges of the role through a trusted context requires a GRANT to access it outside the trusted context.

```
CREATE ROLE CTXROLE;

CREATE TRUSTED CONTEXT CTX1 …
DEFAULT ROLE CTXROLE WITH ROLE AS OBJECT OWNER … ;
```

Plan And Package Ownership:

- Determining ownership when BIND or REBIND are issued in a trusted context, and WITH ROLE AS OBJECT OWNER is specified:
  - If the OWNER BIND option is not specified, the role associated with the binder becomes the owner.
  - If the OWNER BIND option is specified, the ROLE specified for OWNER becomes the owner (the OWNER specified must be a ROLE).
    - The binder needs to be granted BINDAGENT from that ROLE.
    - The binder also receives BINDAGENT, if the ROLE associated with the binder has BINDAGENT.

- If WITHOUT ROLE AS OBJECT OWNER is specified (or defaulted) for the trusted context, then the current rules for BIND and REBIND ownership apply.
  - If a role is associated in a trusted context, then the role privileges are included in the binder's privilege set to determine if the binder is allowed to perform the bind.

Plan And Package Ownership Considerations:

- Plan and Package ownership considerations:
  - For a package to be bound remotely with a ROLE as the owner of the package at the remote DB2, then the trusted context at the remote DB2 must be specified as WITH ROLE AS OBJECT OWNER.
  - If OWNER is specified for a remote BIND across a trusted connection, OWNER could be a role or an AUTHID. Outbound AUTHID translation is not performed for the OWNER.
  - If the plan owner is a role and the application uses a package bound at a remote DB2 server, then the plan owner privilege to execute the package is not considered at the remote DB2 server.
    - The package owner/the process runner (as determined by DYNAMICRULES) at the DB2 server must have the EXECUTE privilege on the package at the remote server.

Ownership of Other Objects

- If CREATE is issued by static SQL, for the ROLE to become the owner of the objects created by executing the plan or package, then the bind of that plan or package must have been performed in a trusted connection where WITH ROLE AS OBJECT OWNER is specified.
  - Otherwise, normal object ownership rules apply.

- If CREATE is issued by dynamic SQL in trusted context where WITH ROLE AS OBJECT OWNER is specified, then the role becomes the owner of the objects.
  - A limitation is that it is not possible to specify the owner of an object created in a trusted context. If specified, SET CURRENT SQLID is ignored.
  - Otherwise, normal object ownership rules apply.

Authorization ID Checking

- Authorization IDs and static SQL:
  - The authorization ID used for the authorization checking of embedded SQL statements is that of the owner of the plan or package.
  - If the application is bound in a trusted context where WITH ROLE AS OBJECT OWNER is specified, the AUTHID used for authorization checking is the role that owns the plan or package.
    - Otherwise it is the AUTHID of the user that owns the plan/package.

- Authorization IDs and dynamic SQL: how role privileges are considered for authorization checking is dependent on the DYNAMICRULES in effect:
  - RUN
  - BIND
  - DEFINERUN and DEFINEBIND
  - INVOKERUN and INVOKEBIND

SWG BetaWorks

Trusted Contexts, Roles, and SSL Support

SSL Support

SSL Support in DB2

- DB2 now provides Secure Socket Layer (SSL) support.
  - z/OS Communications Server implementation of AT-TLS (Application Transparent Transport Layer Security) is a prerequisite.
- A DB2 server can listen on a secure port for inbound SSL connections.
  - Define a secure port to DB2, either during DB2 installation (panel DSNTIP5), or via DSNJU003.
  - A new DSNJU003 SECPORT parameter is provided – this must be different from the values specified for PORT and RESPORT.
  - If SECPORT is disabled, the client can still use the DRDA PORT, and use SSL on it, but DB2 won't validate the use of SSL.
- Data sharing considerations:
  - Each DB2 member that wants SSL support must specify a secure port
  - The secure port for each DB2 member of the group should be the same, just as the DRDA PORT for each member should also be the same.

SSL Support in DB2 (cont.)

- To specify that a secure connection is required for a connection to a remote DB2 server, set new SYSIBM.LOCATIONS column SECURE to ‘Y’.

- The value in the PORT column should be that of the remote server’s SECPORT.
  - If the value for PORT column is blank, but SECURE specifies 'Y', the default reserved secure DRDA port (448) is used.

- Using LOCATION ALIAS to allow some DB2 applications to benefit from SSL protection, and others to be satisfied with unprotected connections:
  - At the requester, define a row in SYSIBM.LOCATIONS specifying the location name to be used for non-secure communications, with the server’s DRDA port.
  - Define a second row with a different location name to be used for secure communications, with the server’s SECPORT, and SECURE specified as 'Y'.
  - At the remote DB2 server, define a location alias to provide a name for DB2 requesters needing to access the server using SSL.

SWG BetaWorks

Trusted Contexts, Roles, and SSL Support

Migration Considerations

-DISPLAY DDF And LOCATION Report Changes

```
DSNL080I - DISPLAY DDF REPORT FOLLOWS:
DSNL081I STATUS=STARTD
DSNL082I LOCATION          LUNAME              GENERICLU
DSNL083I XYZ               NETXYZ.LUXYZ        -NONE
DSNL084I TCPPORT=446 SECPORT=448 RESPORT=5001
DSNL085I IPADDR=::100.10.10.20
DSNL086I SQL DOMAIN=xyzhost.ibm.com
DSNL086I RESYNC DOMAIN=xyzhost.ibm.com
DSNL087I ALIAS              PORT
DSNL088I XYZ_S              448
DSNL099I DSNLTDDF DISPLAY DDF REPORT COMPLETE
```

```
DSNL200I - DISPLAY LOCATION REPORT FOLLOWS
LOCATION                                PRDID    REQSTR SERVER CONVS
9.30.115.135                            DSN09010      9      0      9
 TRUSTED = Y
1080:0:0:0:8:800:200C:417A              DSN09010      7      0      7
STL912B                                 DSN09010      0     23     23
 9.30.115.135
 TRUSTED = Y
DISPLAY LOCATION REPORT COMPLETE
```

-DISPLAY THREAD Report and Other Changes

```
DSNV401I - DISPLAY THREAD REPORT FOLLOWS -
DSNV402I - ACTIVE THREADS - 133
NAME     ST A   REQ ID           AUTHID     PLAN     ASID TOKEN
BATCH    T  *    10 JOB01        ADMF001    APPL01   0027 12
 V485-TRUSTED CONTEXT=DOMINOCONTEXT, SYSTEM AUTHID=SYSADM,
      ROLE=USRROLE
DISPLAY ACTIVE REPORT COMPLETE
DSN9022I - DSNVDT '-DIS THD' NORMAL COMPLETION
```

Other changes:

- Trusted context name, role name, original application user, and security token fields are added to the IFCID correlation header.
- IFCIDs which include extra information about trusted contexts and roles:
  - IFCIDs 62, 140, 141, 142, 169 and 314.
- New IFCIDs (added to audit trace class 10):
  - IFCIDs 269 and 270
- Changes to the Access Control Authorization Exit

"Thank You for listening"

If you have any questions on this DB2 9 for z/OS session, then please send them to the BetaWorks team at:

Ian_Cook@uk.ibm.com
FLETCHPL@uk.ibm.com