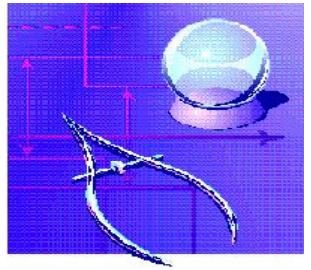
A Peek into the Security Features in DB2 9 for z/OS

Gene Fuh, Ph.D.
Distinguished Engineer and Senior Manager
Query Technology, DB2 for z/OS
IBM Silicon Valley Laboratory

DB2 for z/OS Technical Forum
Taipei, October 5-6, 2009

Disclaimers & Trademarks*



Information in this presentation about IBM's future plans reflect current thinking and is subject to change at IBM's business discretion. You should not rely on such information to make business plans. Any discussion of OEM products is based upon information which has been publicly available and is subject to change. The opinions expressed are those of the presenter at the time, not necessarily the current opinion and certainly not that of the company.

The following terms are trademarks or registered trademarks of the IBM Corporation in the United States and/or other countries: AIX, AS/400, DATABASE 2, DB2*, Enterprise Storage Server, ESCON*, IBM, iSeries, Lotus, NOTES, OS/400, pSeries, RISC, WebSphere, xSeries, z/Architecture, z/OS, zSeries, System p, System i, System z

The following terms are trademarks or registered trademarks of the Microsoft Corporation in the United States and/or other countries: MICROSOFT, WINDOWS, ODBC

For more copyright & trademark information see ibm.com/legal/copytrade.phtml

Agenda

1. **Security objectives**

2. **Data Security – data encryption**

3. **Access Control – authorization**

4. **Trend and Direction**

Explosive Growth of Information Drives Enterprise Infrastructure Challenges

Information Availability

How to deliver continuous and reliable access to information?

Downtime costs can amount up to 16% of revenue in some industries.

Information Security

How to protect and enable secure sharing of information?

84% of security breaches come from internal sources.

Information Retention

How to support our information retention policies?

Average legal discovery request can cost organizations from $150k to $250k.

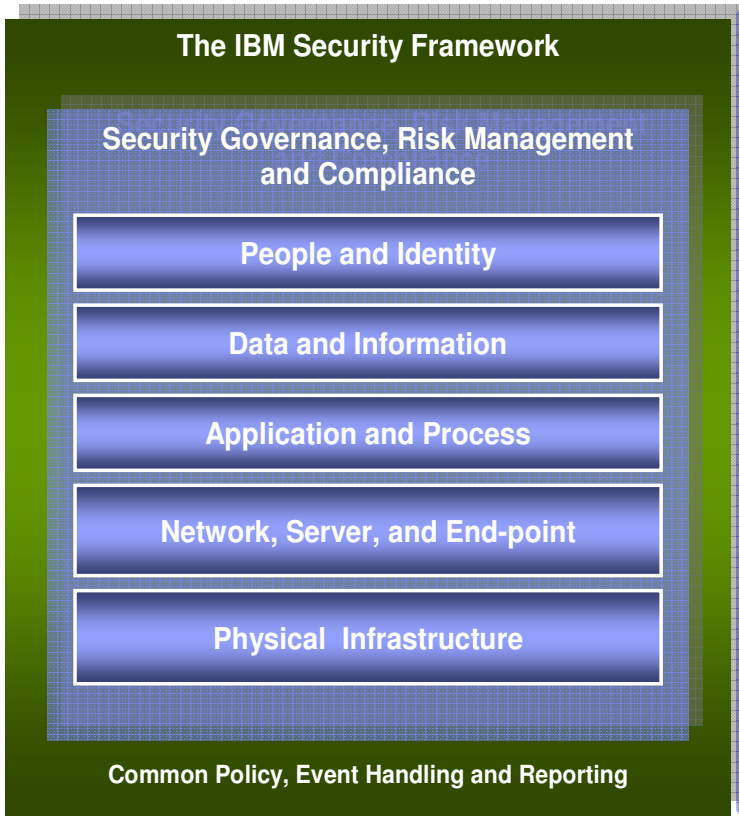
Information Compliance

How to reduce our reputation risks and audit deficiencies?

63% IT executives rate compliance with regulations a top challenge.

Sources: CIO Magazine survey 2007; IBM Tivoli Market needs and profiling study 2005;
The Costs of Enterprise Downtime: NA Vertical Markets 2005" Information Research; IBM Market Intelligence

The IBM Security Framework



- **SECURITY COMPLIANCE**
  - Demonstrable policy enforcement aligned to regulations, standards, laws, agreements (PCI, FISMA, etc..)
- **IDENTITY & ACCESS**
  - Enable secure collaboration with internal and external users with controlled and secure access to information, applications and assets
- **INFORMATION SECURITY**
  - Protect and secure your information assets
- **APPLICATION SECURITY**
  - Continuously manage, monitor and audit application security
- **INFRASTRUCTURE SECURITY**
  - Comprehensive threat and vulnerability management across networks, servers and end-points

System z as the information security hub

- Security-rich holistic design to help protect system from mal-ware, viruses, and insider threats
- Centralized security management for the enterprise
  - Identity and authorization management
  - Certificate Authority in System z
  - Encryption key management
- Encryption solutions to help secure data from theft or compromise
  - Cryptographic acceleration and centralized key management
  - Tamper-resistant secure-key processing
    - FIPS 140-2 Level 4
  - Internet security features
- Collaboration with Tivoli's enterprise security management solutions
  - Identity and access management
  - Monitoring, audit and compliance via zSecure Suite
- *Network Product Guide Magazine recently issued its annual awards for the best business technology products and services that readers trust. The IBM System z10 took home two awards:*
  - *Best in Server Solutions*
  - *Best in Cryptography*

Today's Mainframe:

The power of industry-leading security, the simplicity of centralized management

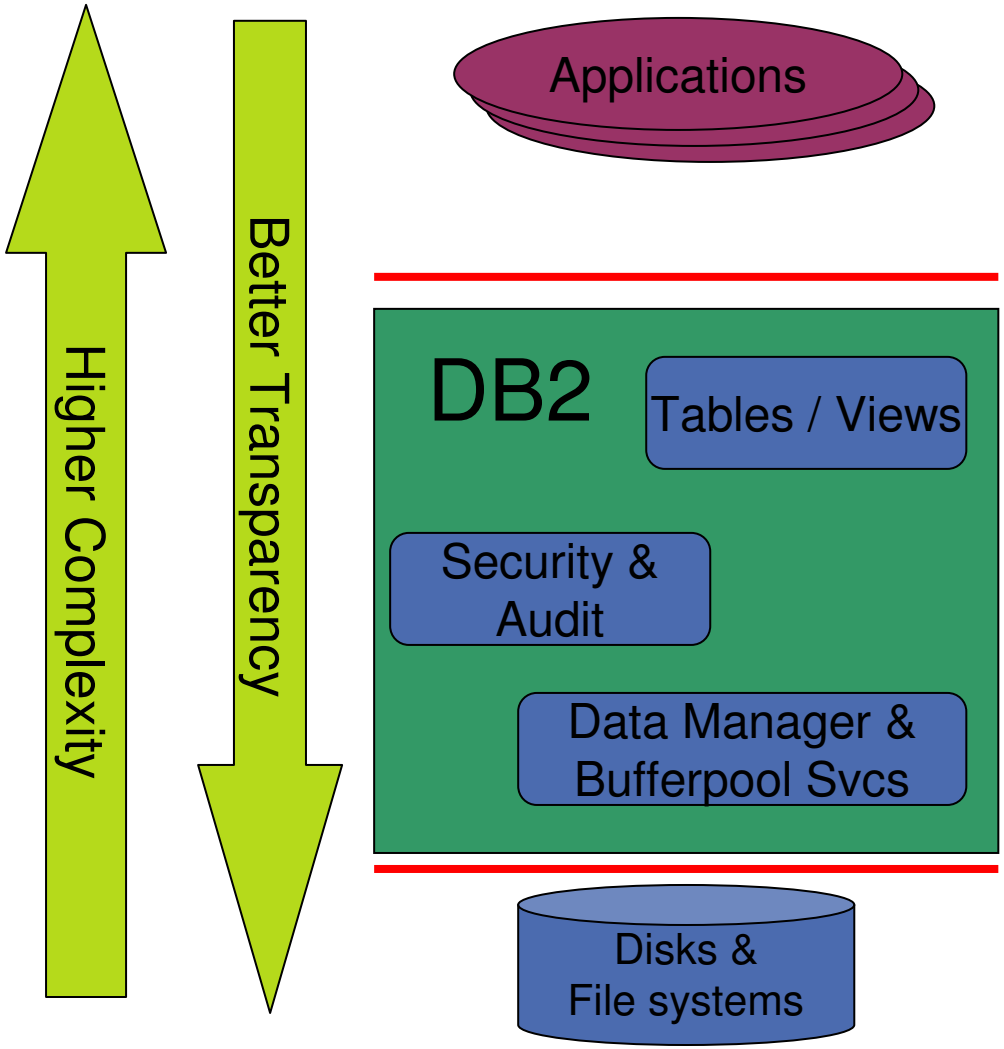
Agenda

1. **Security objectives**
2. **Data Security – data encryption**
3. **Access Control – authorization**
4. **Trend and Direction**

Options for Data Encryption

Outside of DB2 (ICSF, IBM Encryption for z/OS)	General, flexible, no relational range comparisons FOR BIT DATA
On the wire (DRDA V8, SSL V9, IPSec)	General, flexible
User-defined function or stored procedure	General, flexible, invocation needed, no relational range comparisons, manage keys
SQL functions (DB2 V8)	General, flexible, invocation needed, no relational range comparisons, manage keys
DB2 FIELDPROC	No relational range comparisons, FIELDPROC restrictions, FOR BIT DATA
DB2 EDITPROC (IBM tool)	indexes are not encrypted, EDITPROC restrictions
Disk (z/OS, DS8000)	Protect retirement, repair, loss of disk

Data Encryption at Different Levels



Outside of DB2

Encryption on the wire

ENCRYPT/DECRYPT BIFs UDFs

FIELDPROC

EDITPROC

File system and disk subsystem solutions

Disk Encryption - IBM System Storage DS8000

- Drive-Level Encryption
  - Encryption of “data at rest”
  - Continuous, real-time encryption of individual drives
  - Expected to:
    - Have no performance impact
    - Require no application changes
  - Uses Tivoli Key Lifecycle Manager
    - Key management via ICSF and RACF®
    - Audit via SMF

IBM Tivoli Key Lifecycle Manager

Simplified key management across distributed and mainframe

- Client Value
  - Reduces encryption management costs related to set up, use and expiration of keys
  - Enables organizations to comply with disclosure laws and regulations
  - Ensures against loss of information due to key mismanagement
  - Transparently detects encryption-capable media to assign necessary authorization keys
  - Runs on most existing server platforms to leverage resident server's existing access control/high availability/disaster recovery configurations

Software > Tivoli > Products > IBM Tivoli Key Lifecycle Manager >

Tivoli Key Lifecycle Manager

Overview

Simplify, centralize and strengthen encryption key management

IBM Tivoli® Key Lifecycle Manager helps IT organizations better manage the encryption key lifecycle by enabling them to centralize and strengthen key management processes.

Learn more
- System requirements
- Product library
- Data sheet
- Technical article

Trials and Demos
- Demo

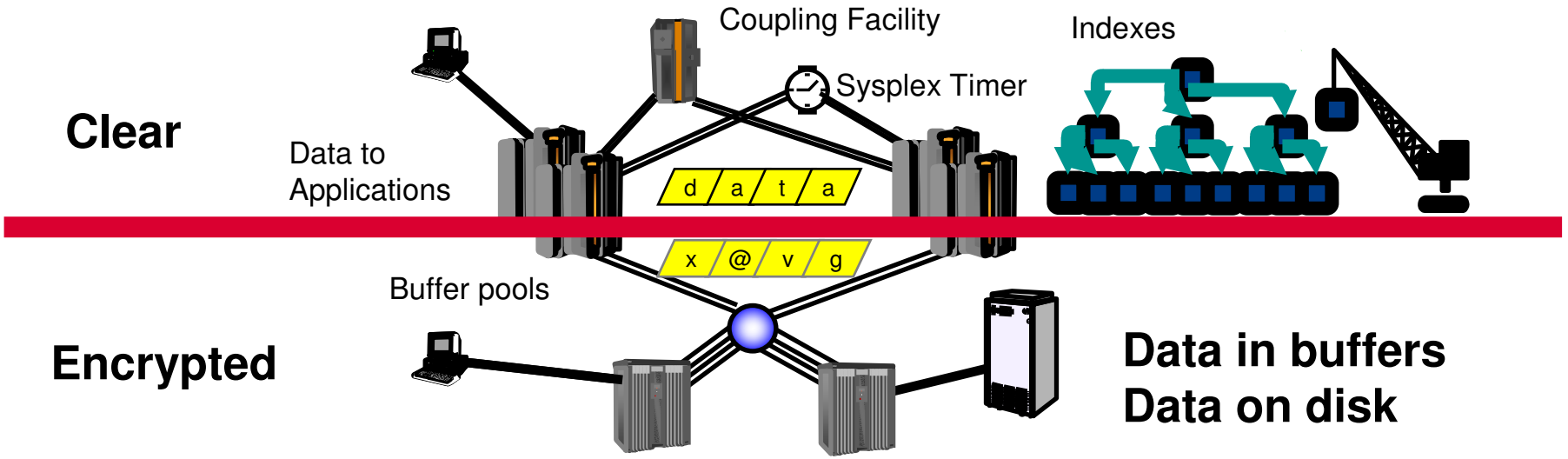
- Centralize and automate the encryption key management process
- Enhance data security while dramatically reducing the number of encryption keys to be managed
- Simplify encryption key management with an intuitive user interface for configuration and management
- Help minimize the risk of loss or breach of sensitive information
- Help facilitate compliance management of regulatory standards such as Sarbanes-Oxley and the Health Insurance Portability and Accountability Act (HIPAA)
- Extend key management capabilities to both IBM and non-IBM products
- Leverage open standards to help enable flexibility and facilitate vendor interoperability

Its predecessor EKM is proven key management system with 2000 customers worldwide!

Simple, Secure and Cost-effective Key Storage, Key Serving and Key Management

DB2 encryption tool using EDITPROC

- Data encryption on disk, data at rest
  - Data on channel, in buffer pools are encrypted
  - Data to applications & indexes are not encrypted
- Existing authorization controls are unaffected



Agenda

1. **Security objectives**

2. **Data Security – data encryption**

3. **Access Control – authorization**

4. **Trend and Direction**

Views

SQL - Data Definition Language

```
CREATE VIEW SW_CUSTOMER AS
    SELECT CUST_NBR, CUST_NAME,
    CUST_CREDIT
        FROM  CUSTOMER
    WHERE CUST_REGION='SW'
```

- Only customers in SW
- Only customer number, name & credit

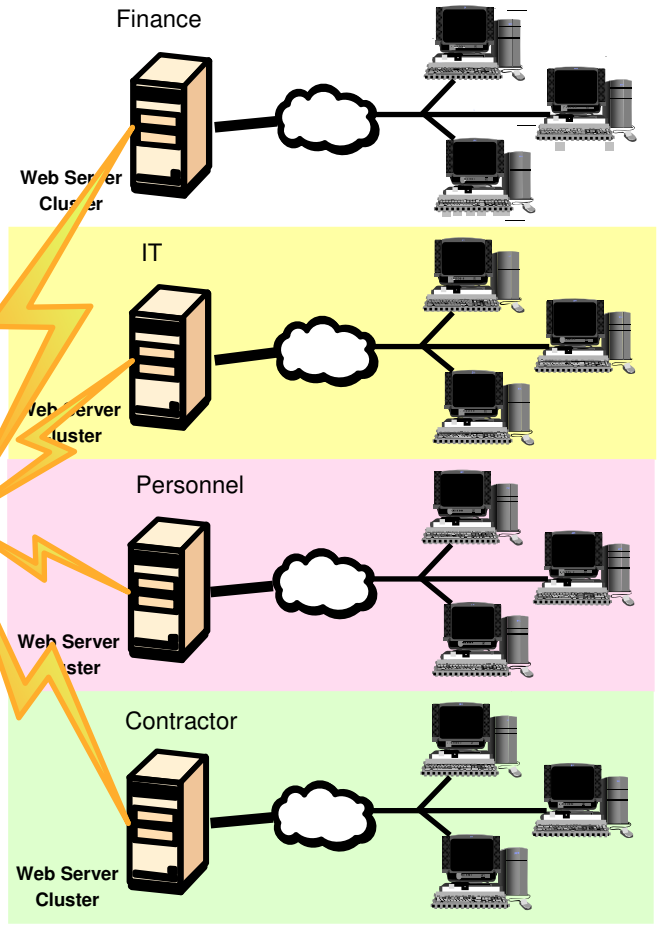
Views can:

- Protect data: rows and/or columns
- Simplify access to data
- Join or union to add or remove information

Multilevel Security and DB2

- Labeled security allows sharing of resources with mixed levels of security in a single image
- Integrated access control, consistent for files, communications, print, database
- Control SQL and utility access

SECURITY LABEL	Col 1	Col 2	Col 3
Personnel	234	USA	50%
Finance	198	France	23%
Personnel	2	UK	9%
Finance	234	USA	11%
Personnel	22	Germany	9%
IT	87	USA	14%
Contractor	23	UK	20%
Personnel	34	Germany	43%
Finance	981	USA	12%
IT	223	USA	10%
Contractor	45	Canada	29%

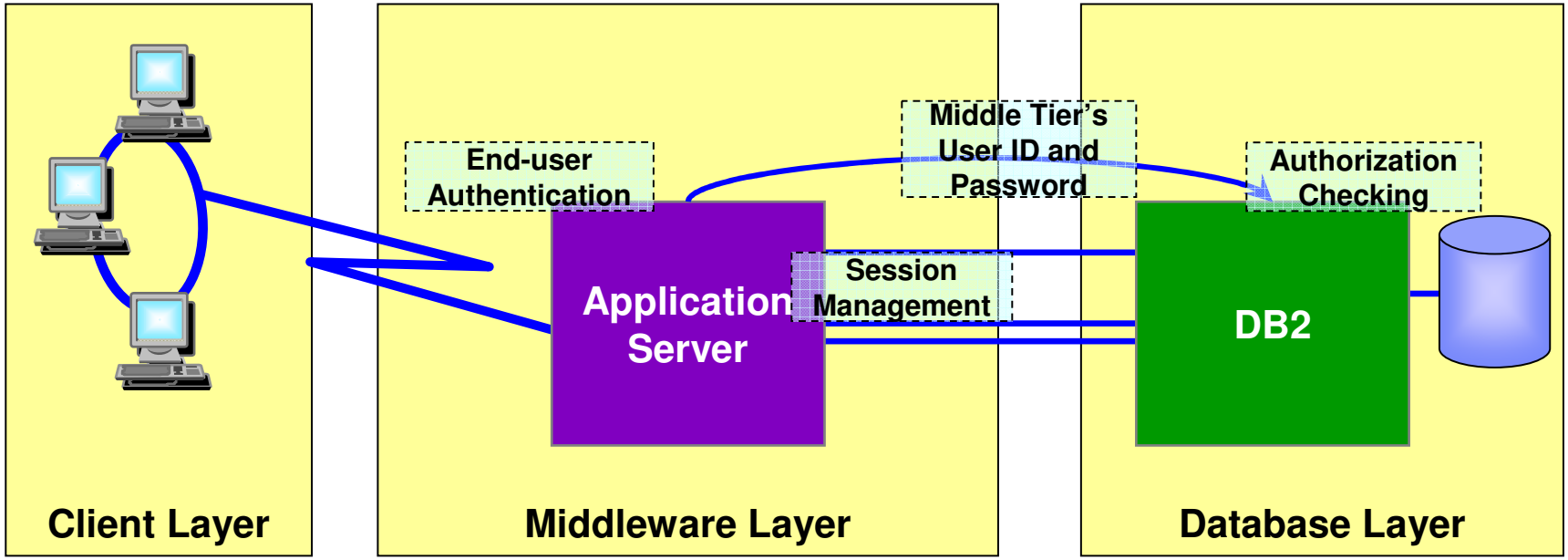


Multilevel Security on z, z/OS, RACF

Trusted Contexts, Roles and SSL

- Current Three-tier Authorization
- Trusted Contexts – A Quick Overview
- Defining A Trusted Context
- Establishing A Trusted Connection
- Authorization ID Switching
- Roles and Context-specific Privileges
- Trusted Contexts And Object Ownership
- Authorization ID Checking

Current Authentication in a Three-Tier Architecture

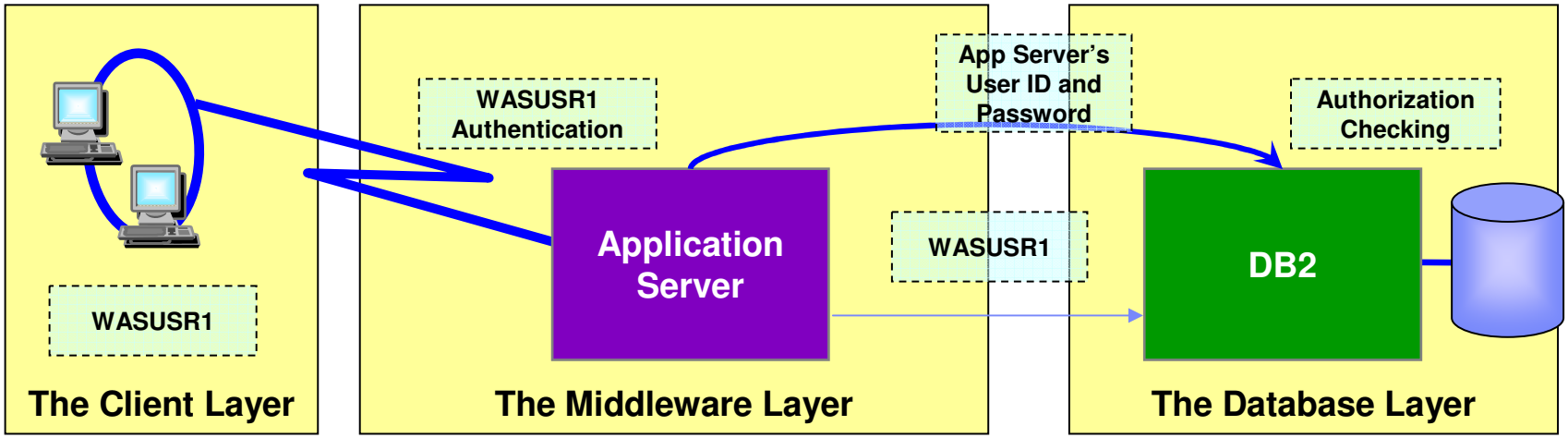


- **A three-tiered application model with DB2 as the database server:**
  - The middle layer authenticates users running client applications.
  - It also manages interactions with the database server.
  - The middle layer's user ID and password are used for database authentication.
  - The privileges 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 Authentication in a Three-tier Architecture



- **The application server's user ID and password are used to establish the trusted connection.**
- **The user is switched in the trusted connection and client user ID is propagated to the server**
- **The client authorization ID's privileges are checked for database access and used for auditing**

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:

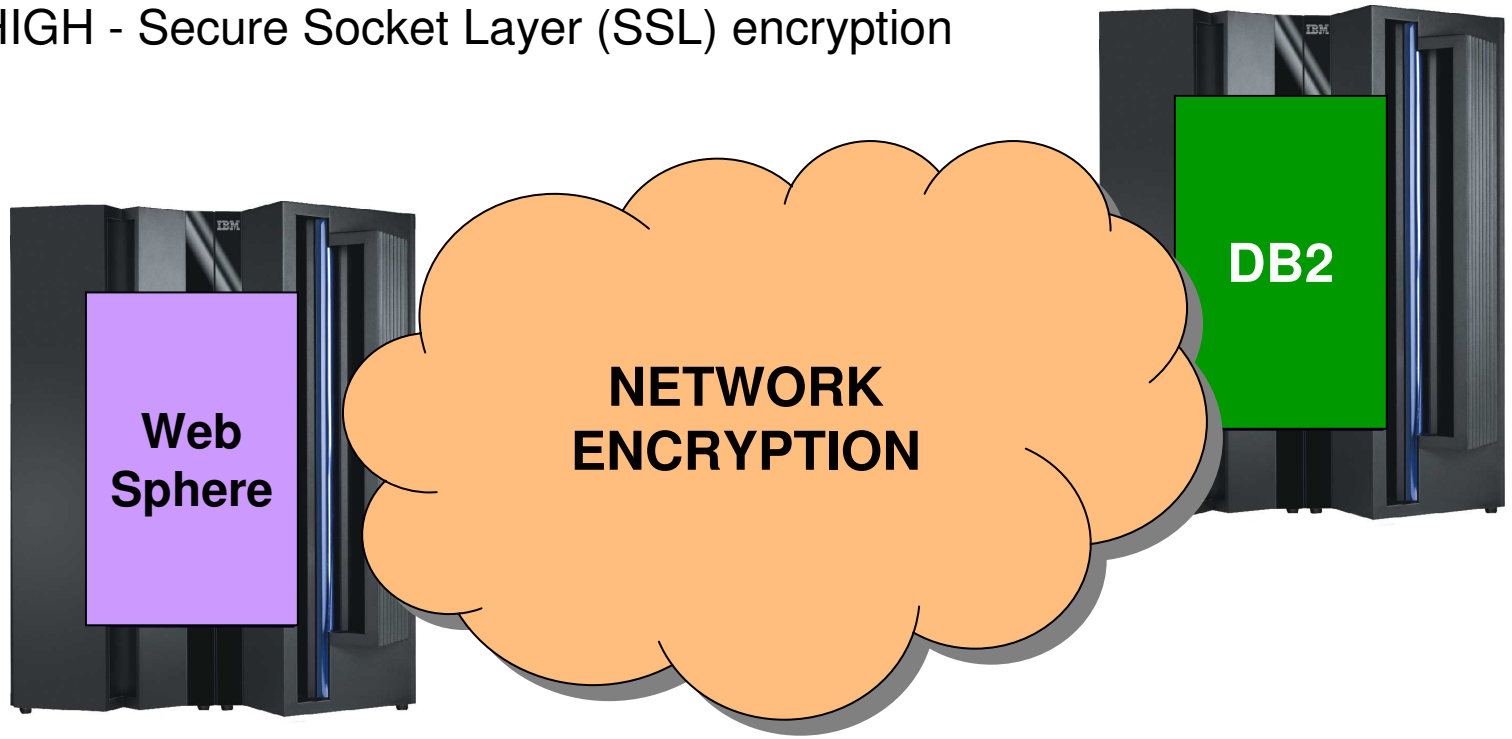
```
NETACCESS INBOUND OUTBOUND
9.67.40.0 255.255.248.0 ZONEB
9.67.0.0  255.255.0.0   ZONEA
Default                 WORLD
ENDNETACCESS
```

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



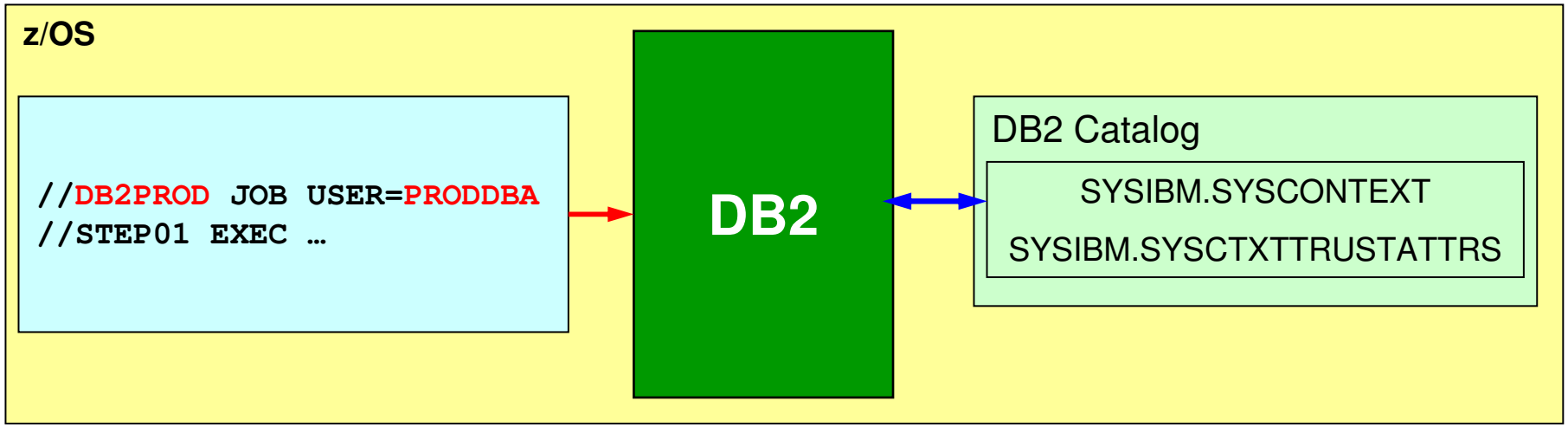
Defining A Trusted Context

- 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)

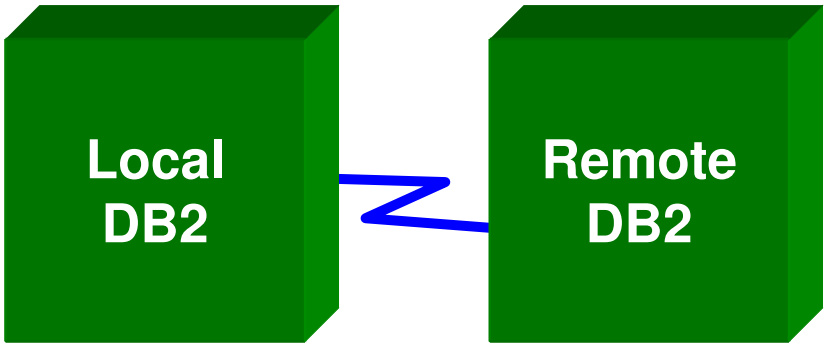
SYSIBM.LOCATIONS

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	

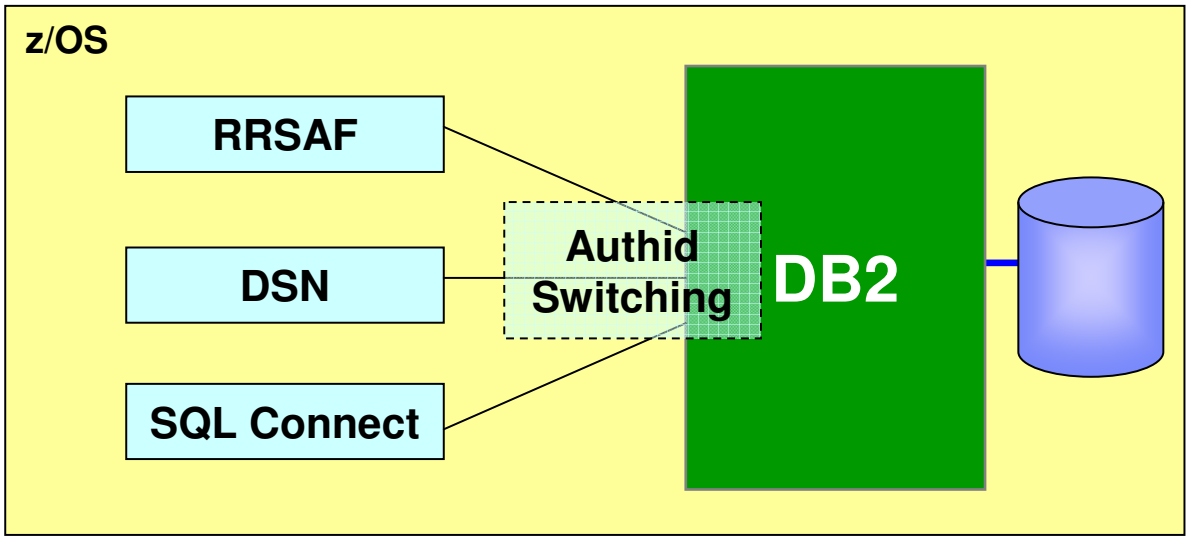
Authid Switching

- 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;
```

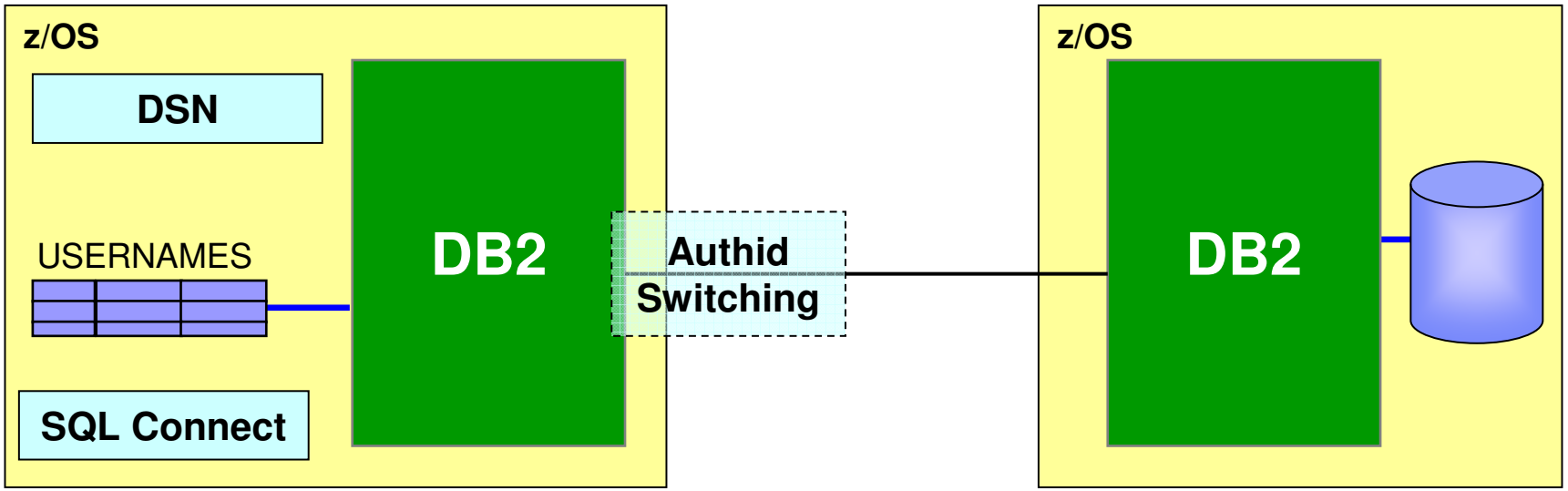
Authid Switching – Local Processing

- 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.



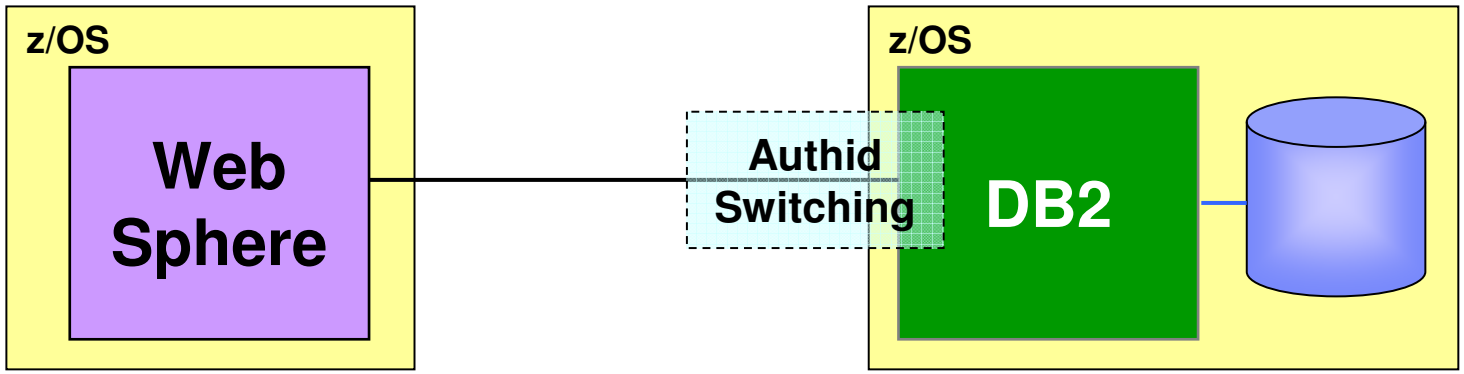
Authid Switching, Remote Processing (DB2 Requester)

- 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*.

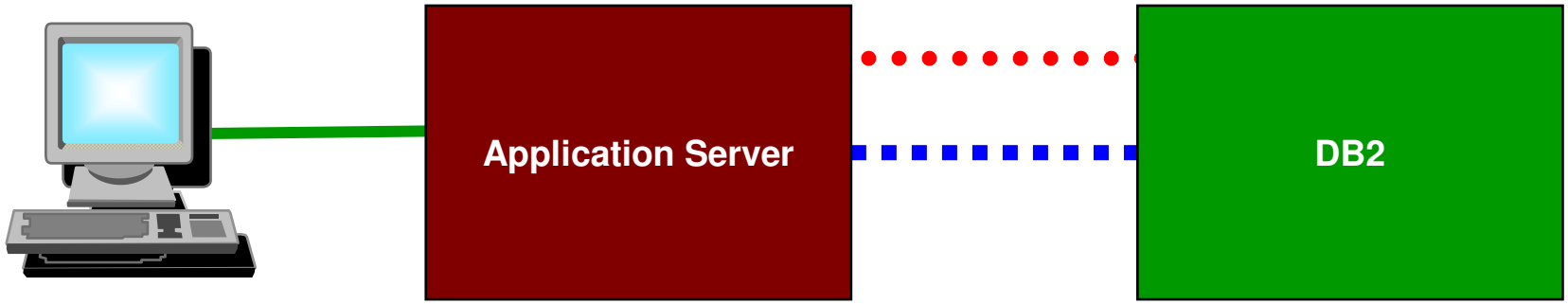


Roles and Context-specific Privileges

- 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



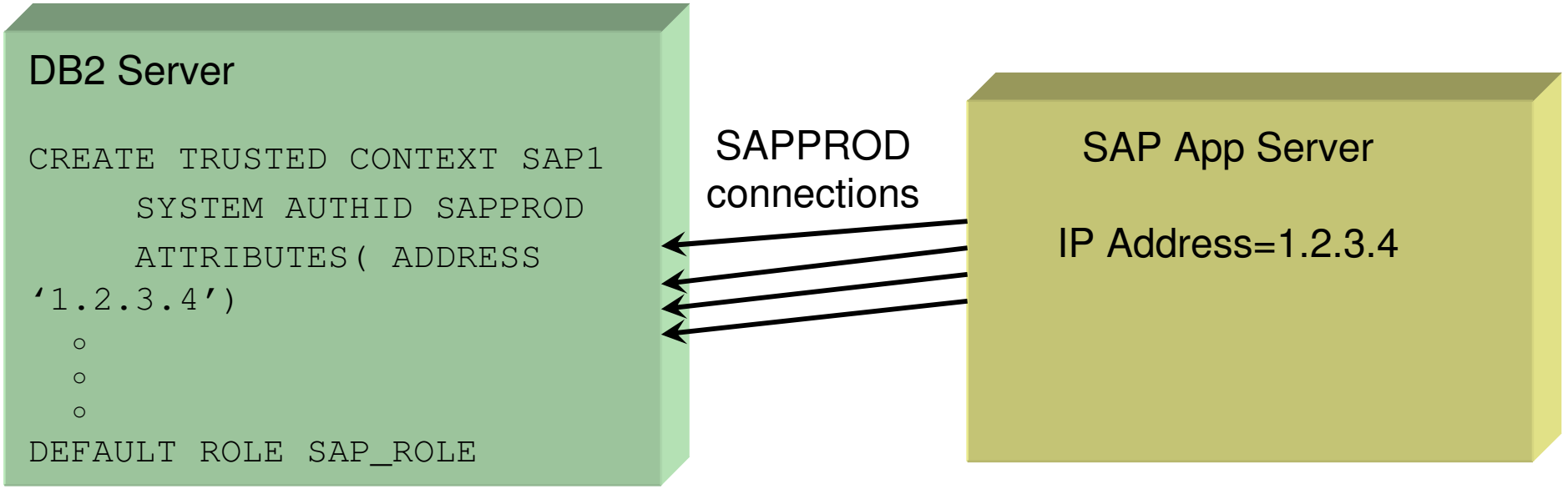
User ID flows to Application Server

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

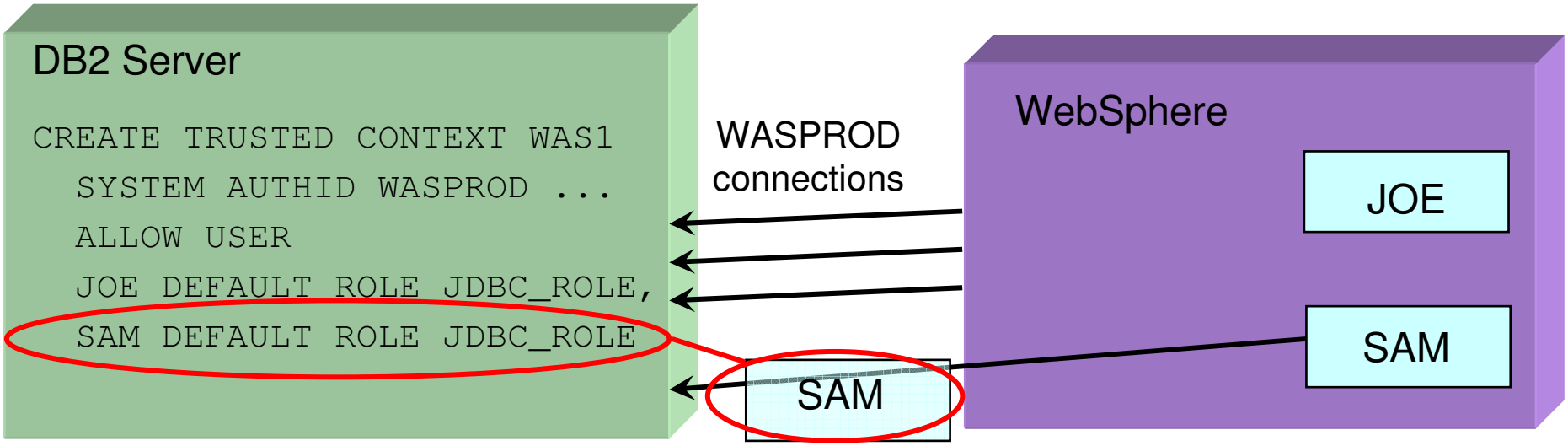
Example: Securing An Application Server

- 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.



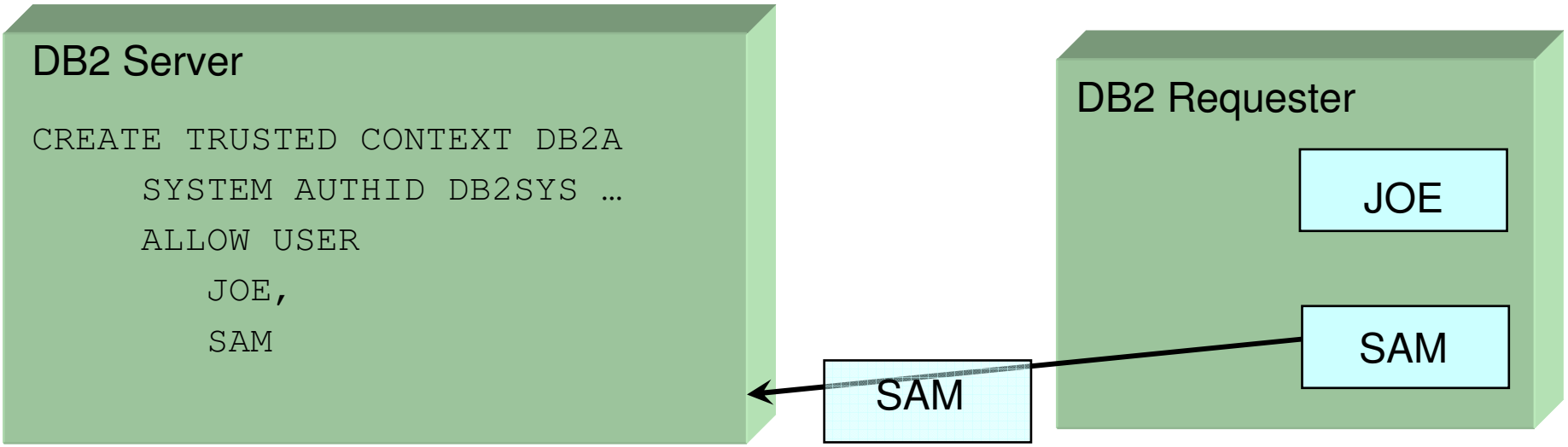
Example: Dynamic SQL Auditing

- 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!

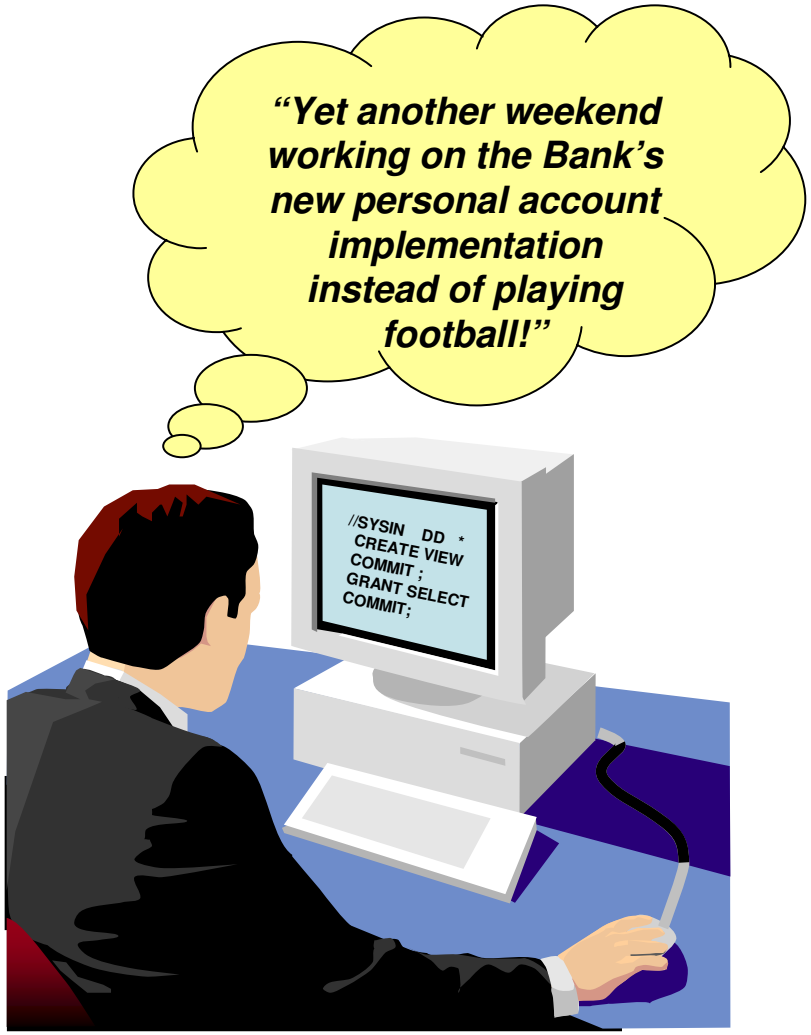


Example: Roles, Trusted Context and Already-Verified DRDA

- 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: 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:
     - Add the auth id to the trusted context (ALTER statement)
     - DBA performs the change activity
     - Drop the auth id from the trusted context (ALTER statement)
  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;
```

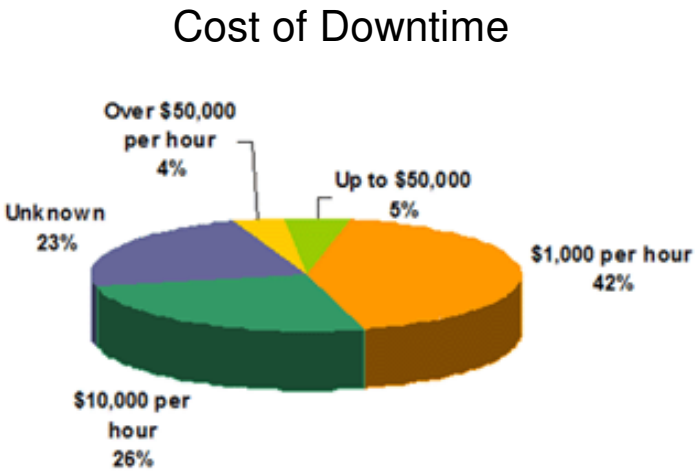
- GRANT and REVOKE are extended

```
GRANT SELECT ON T1 TO ROLE CTXROLE;

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

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

Improved Audit in DB2 9: Trace (IFCID) changes

- **Correlation Header - Trusted context name, role name, original application user, security token**
- **IFCID 062 -Statement and object types for trusted contexts and roles**
- **New ID type is added to distinguish between auth ID and role:**
  - IFCID 140 – Auth ID checked type
  - IFCID 141 – Grantor / Revoker type
  - IFCID 142 – Table owner type
- **IFCID 169 – Identifier type 'S' traces system auth ID translation**
- **New Audit Trace Class 10**
  - IFCID 269 – Establish trusted connection and Switch user
  - IFCID 270 – CREATE and ALTER TRUSTED CONTEXT statements
- **IFCID 314 – Role name**

Improved Audit in DB2 9: Trace INCLUDE Filtering

- **-START TRACE new filtering capabilities that INCLUDE based on keywords:**
  - USERID – client user ID
  - WRKSTN – client workstation name
  - APPNAME – client application name
  - PKGLOC – package LOCATION name
  - PKGCOL – package COLLECTION name
  - PKGPROG – PACKAGE name
  - CONNID – connection ID
  - CORRID – correlation ID
  - ROLE – end user's database Role
- **Positional and terminating wildcards can be used**

```
-STA TRACE … ROLE(DBAROLE, USRROLE)
-STA TRACE(ACCTG) CLASS(1,2,3) AUTHID(A_M*)
```

Improved Audit in DB2 9: Trace EXCLUDE Filtering

- **-START TRACE new filtering capabilities that EXCLUDE based on keywords:**
  - XPLAN - PLAN name
  - XAUTH - authorization ID
  - XLOC - LOCATION name
  - XUSERID - client user ID
  - XWRKSTN - client workstation name
  - XAPPNAME - client application name
  - XPKGLOC - package LOCATION name
  - XPKGCOL - package COLLECTION name
  - XPKGPROG - PACKAGE name
  - XCONNID - connection ID
  - XCORRID - correlation ID
  - XROLE - end user's database ROLE
- **Positional and terminating wildcards can be used.**
  - Wild card logic cannot be used to exclude all threads

```
-STA TRACE … XROLE(DBAROLE, USRROLE)
-STA TRACE … XPLAN(A*, B*)
```

Agenda

1. **Security objectives**

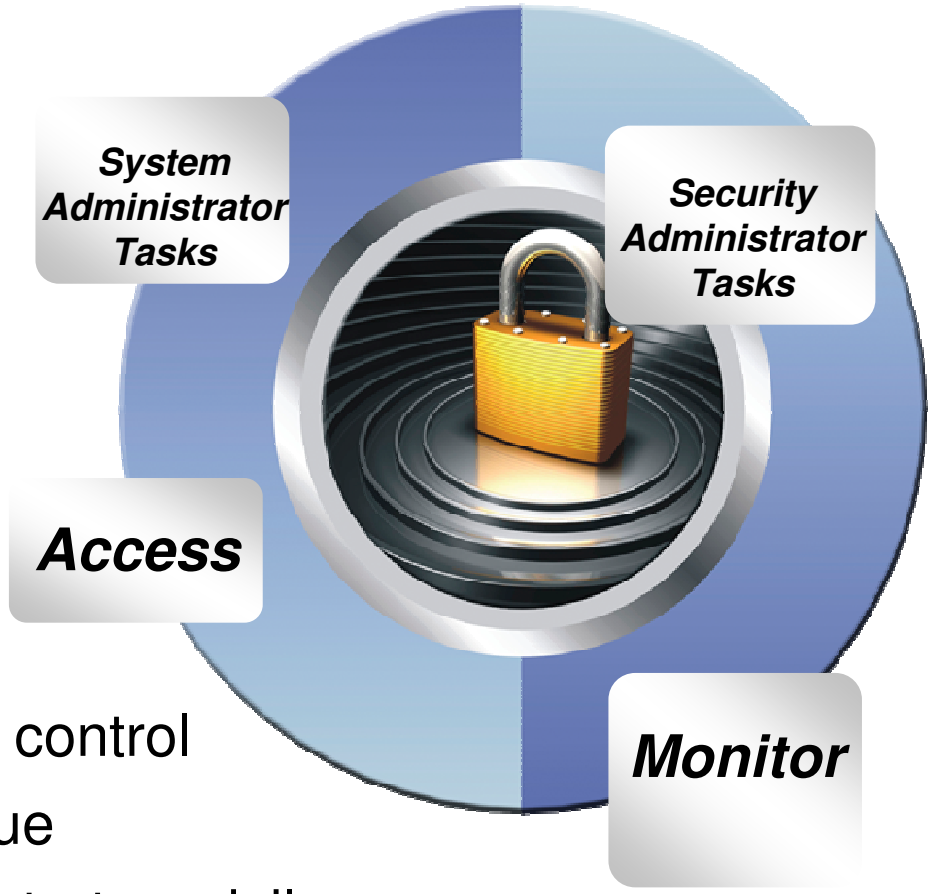
2. **Data Security – data encryption**

3. **Access Control – authorization**

4. **Trend and Direction**

Business Security & Compliance Needs

- Data retention
- Protect sensitive data from privileged users
- Separate authority to perform security related tasks
- Allow EXPLAIN without execute privilege or ability to access data
- Audit privileged users use



DB2 X

- Row and column access control
  - Allow masking of value
- Finer granularity administrator privileges

For Additional Information on IBM's Security Solutions

Learn more about how IBM can provide a holistic, business-driven security approach

www.ibm.com/security

Learn more about specific solutions with IBM security

- IBM System z Security Building Blocks
- IBM Data Encryption Solutions
- IBM Security Management Solutions
- IBM Facility Security Solutions
- IBM Security Services

Next Steps & More Information

- **Are you ready for DB2 9 for z/OS ?**
  Contact your local IBM representative or email
  WW DB2 for z/OS Market Manager Surekha21@uk.ibm.com
- **Need More Information**
  DB2 for z/OS Landing page
- **DB2 for z/OS Whitepaper**
  DB2 9 for z/OS Data On Demand
- **IBM Redbooks**
  Latest Redbooks
- **Join "The World of DB2 for z/OS !"**
- **Twitter DB2: http://twitter.com/IBMDB2**


Thank
You