DB2 Data Management Software

IBM

e business software

Chapter 4: Controlling Data Access

IBM DB2 Universal Database V8.1

Database Administration Certification Preparation Course

Maintained by Clara Liu

IBM Software Group

Objectives

- In this section, we will cover:
  - Methods of Authentication
  - Hierarchy of Authorities
  - Levels of Privileges
  - Users and Groups

DB2 Data Management Software

IBM

e business software

Chapter4: Controlling Data Access

Authentication

Authorities

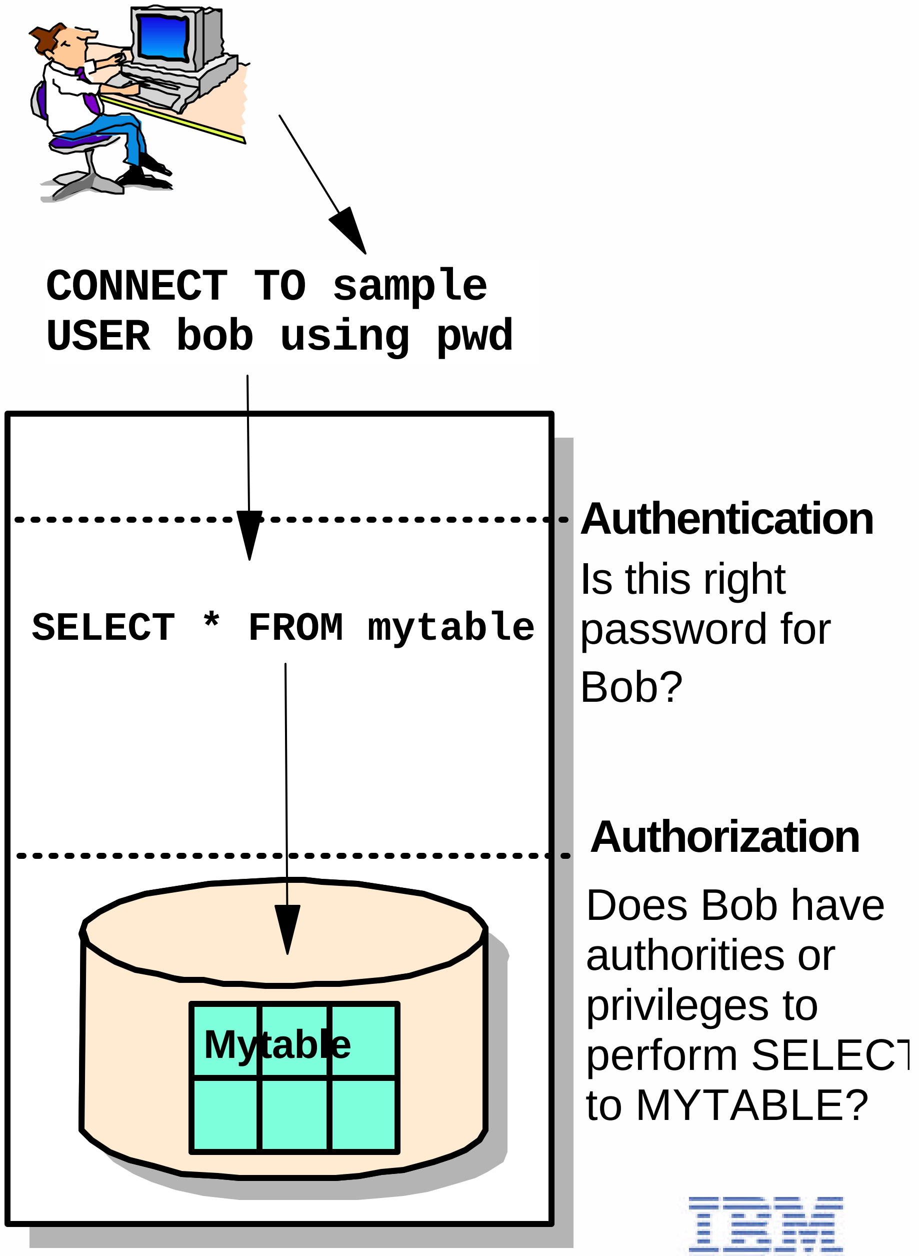
Privileges

Users and Groups

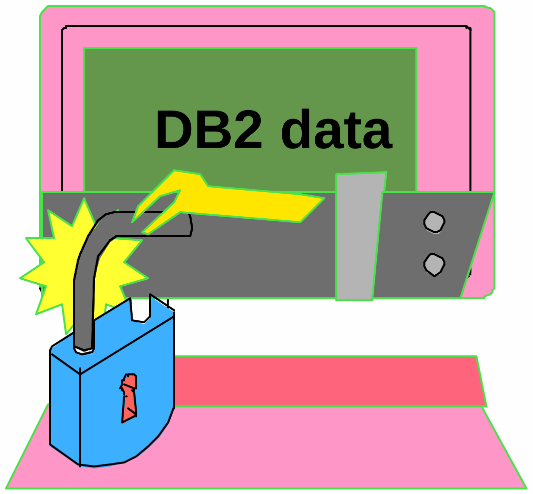
IBM Software Group

DB2 Security Overview

- DB2 uses a combination of:
  - External security service
  - Internal access control information
- Authentication
  - Identify the user
  - Check entered user name and password
  - Done by security facility outside of DB2 (Part of the OS, Kerberos and so forth)
- Authorization
  - Users can access only DB2 objects for which they have the appropriate authorization - the required authorities or privileges
  - Check if authenticated user may perform requested operation
  - Done by DB2 facilities
  - Information stored in DB2 catalogs and DBM configuration file


CONNECT TO sample
USER bob using pwd
SELECT * FROM mytable
Mytable
Authentication
Is this right password for Bob?
Authorization
Does Bob have authorities or privileges to perform SELECT to MYTABLE?

External Database Security



AUTHENTICATION

– verifying that the user is really the person he/she claims to be

- Authentication is handled by a security facility outside of DB2
- The security facility can be part of the operating system or a separate product (Kerberos)
- There are no security facilities on the Windows 9x or Windows ME
- The security facility requires two items to authenticate a user:
  - User ID
  - Password

Authentication Types

- Authentication type determines HOW and WHERE the user is verified
- Authentication types supported in DB2 UDB
  - SERVER (default)
  - SERVER_ENCRYPT
  - KERBEROS
  - KRB_SERVER_ENCRYPT
  - CLIENT
- Two places to specify authentication type: Server and Client
- At Server:
  - One authentication type per instance, applies to all databases under the instance
  - Defined at the instance level in the database manager configuration file
  - Can be updated: UPDATE DBM CFG USING AUTHENTICATION [...]
- At Client:
  - Defined in the CATALOG DATABASE command, therefore authentication type applies to the specified database only
  - Example:

    CATALOG DATABASE sample AT NODE db2server **AUTHENTICATION SERVER**

Authentication Type - Server

AUTHENTICATION=SERVER

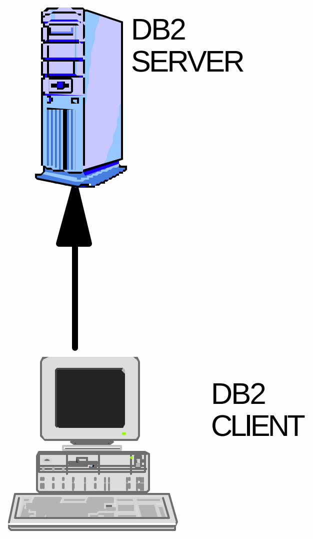
- Authentication occurs at the server
- Userid and password are sent to the server for validation
- Central management of users
- User ID and password flow over network
  - Can be encrypted with AUTHENTICATION = SERVER_ENCRYPT
  - Both user ID and password are encrypted
- User required to reenter the user name and password for connecting to a remote DB2 server

4 Compared to the valid username and password at the server

3 "usersrv" and "pwdsrv" are sent to the server

2
```
CONNECT TO
sample
USER usersrv
USING pwdsrv
```

1 Logon to client
Username: USRCLT
Password: PWDCLT



Authentication Type - KERBEROS

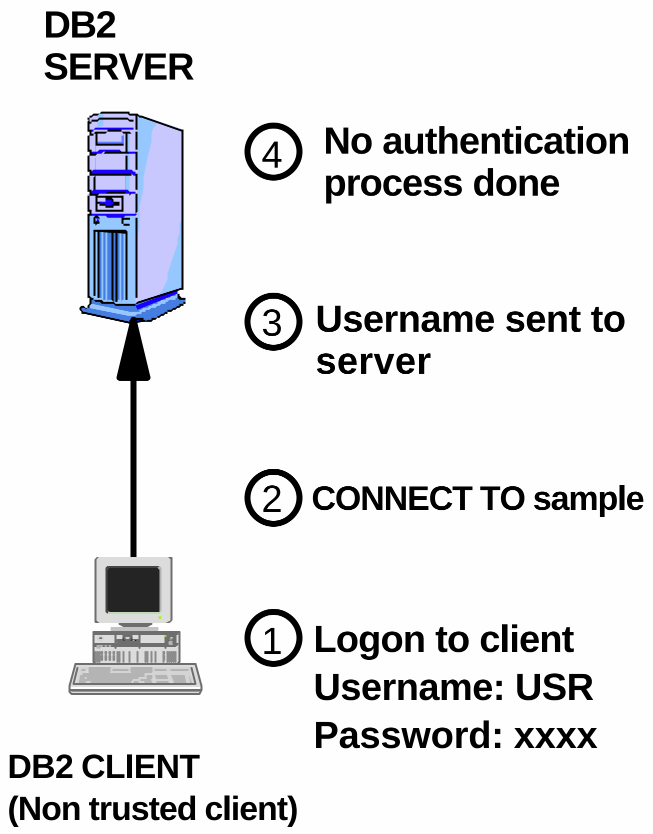
- Use conventional cryptography to create a shared secret key
- This key becomes a user's credential and is used to verify the identity of users when local or network services are requested
- Eliminates the need to pass the user name and password across the network as clear text
- Enables the use of a single sign-on to a remote DB2 server
- AUTHENTICATION=KRB_SERVER_ENCRYPT
  - Authentication SERVER_ENCRYPT and KERBEROS can be used by clients accessing the same DB2 server instance
  - KERBEROS and KRB_SERVER_ENCRYPT only support clients and servers running Windows 2000 or XP or .NET platforms
- If authentication type specified at client and server is different:

Client Specification	Server Specification	Client/Server Resolution
KERBEROS	KRB_SERVER_ENCRYPT	KERBEROS
Any other setting	KRB_SERVER_ENCRYPT	SERVER_ENCRYPT

Authentication Type - CLIENT

AUTHENTICATION=CLIENT

- Authentication occurs at the client
- Password is NOT sent to the server for validation unless you have CLIENT authentication with SERVER validation
- Enables single point logon
- Be careful in insecure environments
  - Windows 9x, Windows 3.1, Mac do not have a reliable security facility
  - They can connect to server as an administrator without any authentication unless TRUST_ALLCLNTS=NO is set on server
- If the remote instance has CLIENT authentication, two other parameters determine the final authentication type:
  - TRUST_ALLCLNTS
  - TRUST_CLNTAUTH


DB2 SERVER
4 No authentication process done
3 Username sent to server
2 CONNECT TO sample
1 Logon to client Username: USR Password: xxxx
DB2 CLIENT (Non trusted client)

TRUST_ALLCLNTS

- Decide whether to trust all clients
- TRUST_ALLCLNTS = YES
  - Trust all clients including trusted, non-trusted, and host clients
  - Authentication will take place at client (except one case)
- TRUST_ALLCLNTS = NO
  - All untrusted clients will be authenticated at the server
  - Must provide user ID and password
- TRUST_ALLCLNTS = DRDAONLY
  - Only hosts clients are allowed to authenticate at client

TRUST_CLNTAUTH

- Specify where authentication will take place when a user ID and password are supplied with a CONNECT statement or ATTACH command
- Active when AUTHENTICATION=CLIENT only
  - If AUTHENTICATION=SERVER, userid/password must be sent to DB2 server on connect
- Active when userid and password provided for connection
- TRUST_CLNTAUTH=CLIENT
  - Authentication done at CLIENT
  - UserID and password not required in CONNECT and ATTACH
- TRUST_CLNTAUTH=SERVER
  - Authentication done at SERVER if a user ID and password are provided with a CONNECT or ATTACH

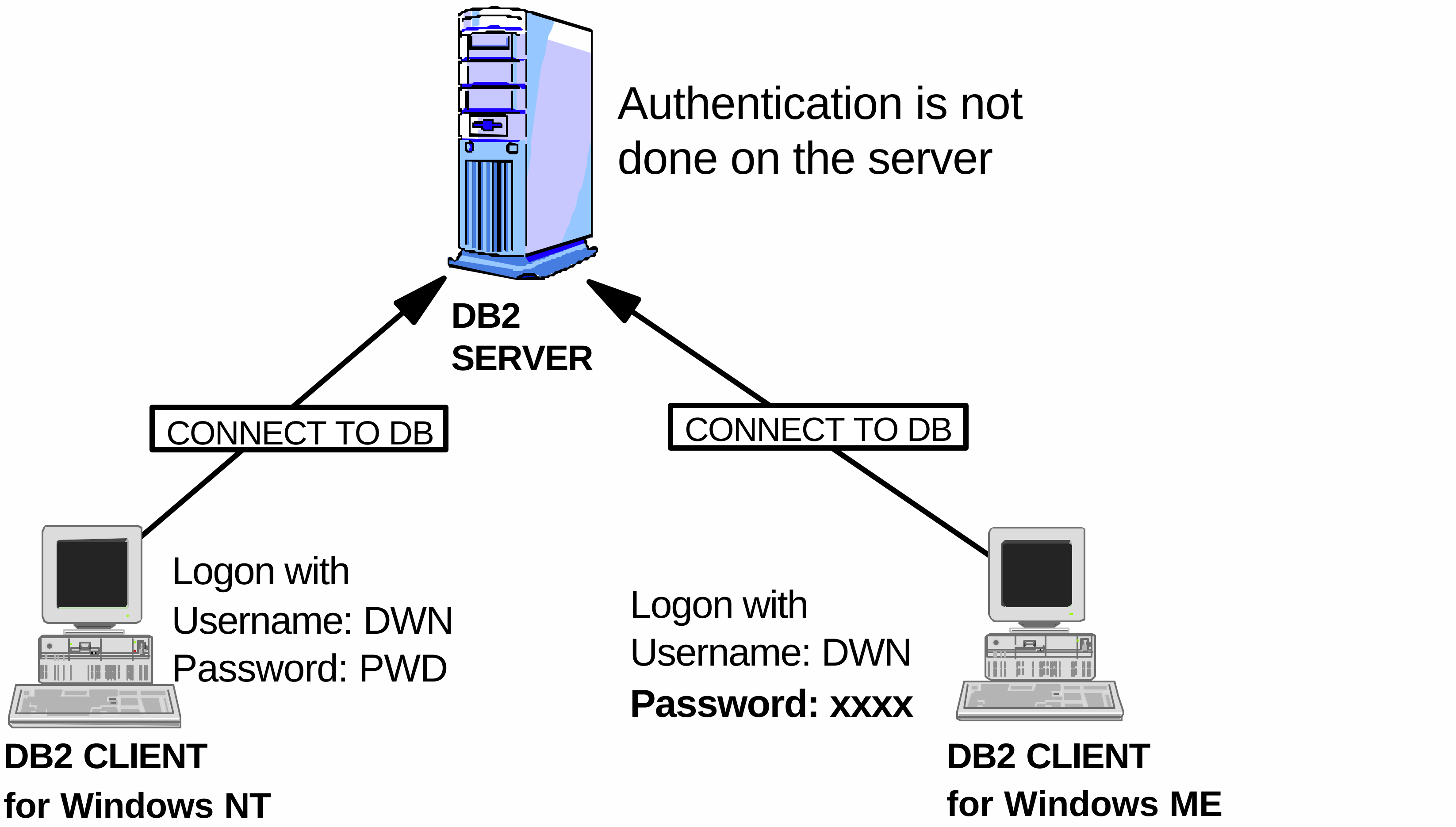
TRUST_CLNTAUTH

- Specify where trusted client is authenticated
  - Untrusted clients always validated at DB2 server if TRUST_ALLCLNTS=NO (regardless of the setting of TRUST_CLNTAUTH)
- Useful if you need to control where authentication takes place based on whether CONNECT sends userid/password or not
  - Set TRUST_CLNTAUTH=SERVER to reduce RPC to domain controller

TRUST_ALLCLNTS	TRUST_CLNTAUTH	Trusted Client Authentication no password	Trusted Client Authentication with password	Untrusted Client Authentication
YES (default)	CLIENT (default)	CLIENT	CLIENT	N/A
YES (default)	SERVER	CLIENT	SERVER	N/A
NO	CLIENT (default)	CLIENT	CLIENT	SERVER
NO	SERVER	CLIENT	SERVER	SERVER

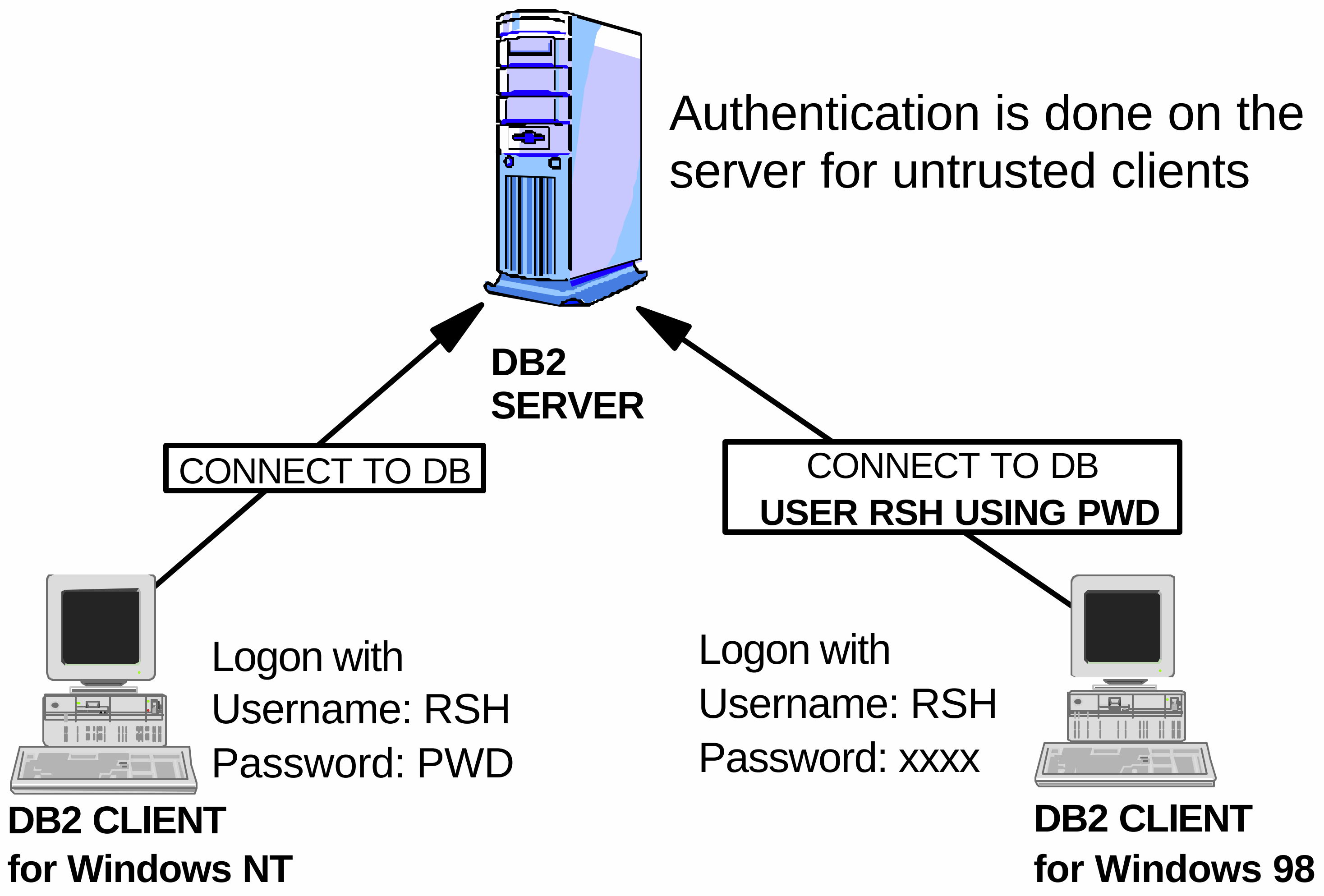
Client Authentication Scenario

TRUST_ALLCLNTS=YES
TRUST_CLNTAUTH=CLIENT


Authentication is not done on the server
DB2 SERVER
CONNECT TO DB
CONNECT TO DB
Logon with Username: DWN Password: PWD
Logon with Username: DWN Password: xxxx
DB2 CLIENT for Windows NT
DB2 CLIENT for Windows ME

Client Authentication Scenario

TRUST_ALLCLNTS=NO
TRUST_CLNTAUTH=CLIENT


Authentication is done on the server for untrusted clients
DB2 SERVER
CONNECT TO DB
CONNECT TO DB USER RSH USING PWD
Logon with Username: RSH Password: PWD
Logon with Username: RSH Password: xxxx
DB2 CLIENT for Windows NT
DB2 CLIENT for Windows 98

DB2 Data Management Software

IBM

e business software

Chapter 4: Controlling Data Access

Authentication

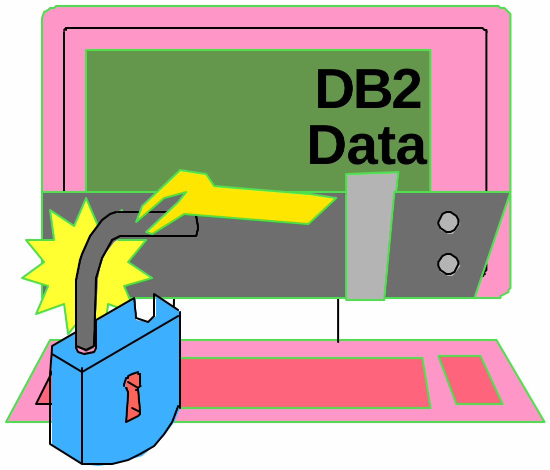
Authorities

Privileges

Users and Groups

IBM Software Group

Database Internal Security



- Inside the database manager

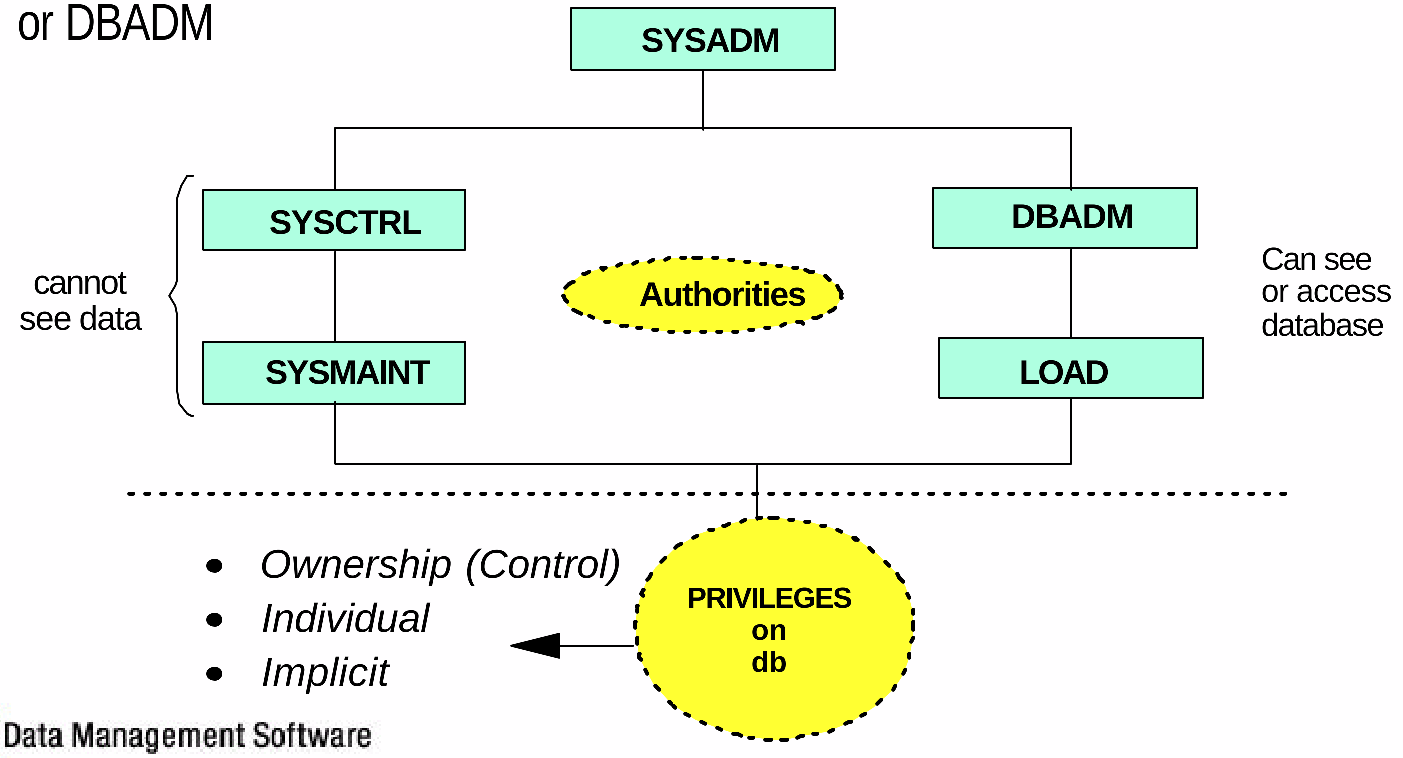
  ACCESS CONTROL

  - ability to create or access database objects

- Authorization is the process whereby DB2 obtains information about an authenticated DB2 user that indicates the database operations a user may perform and what data objects may be accessed
- Privileges enable users to create or access database resources
- Authority levels provide a method of grouping privileges and higher-level database manager maintenance and utility operations
- Privileges and Authorities control access to the database manager and its database objects

DB2 Access Control Authority

- Five authorities in DB2 UDB
  - SYSADM held the most authorities and privileges for the DB2 instance
  - System authority (SYSCTRL or SYSMAINT) gives full privileges for managing the system, but does not allow access to the data
  - DBADM authority gives privileges to perform administrative task on the database and has full data access to the database
  - Load authority gives privileges for running the LOAD utility without the need for SYSADM or DBADM



Set System Authorities in DBM Configuration

- System authorities are not established by GRANT statement
- User groups defined in operating system or security facility are assigned to the system authorities in database manager configuration
- No default values
- Maximum length for group name is 8

Database Manager Configuration

```
SYSADM group name                  (SYSADM_GROUP) = ADM1
SYSCTRL group name                (SYSCTRL_GROUP) = CTRL1
SYSMAINT group name              (SYSMAINT_GROUP) = MAINT1
```

```
db2 update dbm cfg using sysadm_group adm1
db2 update dbm cfg using sysctrl_group ctrl1
db2 update dbm cfg using sysmaint_group maint1
```

Grant/Revoke Database Authorities

- Database level authorities, DBADM and LOAD, are granted to a user or a group of user by the GRANT statement
- Revoke them with the REVOKE statement
- When DBADM authority is granted, BINDADD, CONNECT, CREATETAB, CREATE_NOT_FENCED privileges, and IMPLICIT_SCHEMA authority are granted as well
- Users with LOAD authority also requires INSERT privilege to load data in a table
- Example:
  - GRANT DBADM ON DATABASE TO USER john;
  - GRANT LOAD ON DATABASE TO GROUP maintgrp;
  - REVOKE LOAD ON DATABASE FROM maintgrp;

Function	SYSADM	SYSCTRL	SYSMAINT	DBADM
UPDATE DBM CFG	YES			
GRANT/REVOKE DBADM	YES			
ESTABLISH/CHANGE SYSCTRL	YES			
ESTABLISH/CHANGE SYSMAINT	YES			
FORCE USERS	YES	YES		
CREATE/DROP DATABASE	YES	YES		
RESTORE TO NEW DATABASE	YES	YES		
UPDATE DB CFG	YES	YES	YES	
BACKUP DATABASE/TABLE SPACE	YES	YES	YES	
RESTORE TO EXISTING DATABASE	YES	YES	YES	
PERFORM ROLL FORWARD RECOVERY	YES	YES	YES	
START/STOP INSTANCE	YES	YES	YES	
RESTORE TABLE SPACE	YES	YES	YES	
RUN TRACE	YES	YES	YES	
OBTAIN MONITOR SNAPSHOTS	YES	YES	YES	
QUERY TABLE SPACE STATE	YES	YES	YES	YES*
PRUNE LOG HISTORY FILES	YES	YES	YES	YES
QUIESCE TABLE SPACE	YES	YES	YES	YES*
LOAD TABLES	YES			YES*
SET/UNSET CHECK PENDING STA-TUS	YES			YES
CREATE / DROP EVENT MONITORS	YES			YES

Database Privileges

- **CONNECT** allows a user to access the database
- **BINDADD** allows a user to create new packages in the database
- **CREATETAB** allows a user to create new tables in the database
- **CREATE_NOT_FENCED** allows a user to create a user-defined function (UDF) or stored procedure that is "not fenced".
- **IMPLICIT_SCHEMA** allows theuser to create objects in a schema that does not already exist, SYSIBM becomes the owner of the implicitly created schema and PUBLIC is given the privilege to create objects in this schema
- **QUIESCE_CONNECT** allows a user to access the database while it is quiesced
- **CREATE_EXTERNAL_ROUTINE** allows a user to create a procedure for use by applications and other users of the database.
- Only users with SYSADM or DBADM authority can grant and revoke these privileges to and from other users

Schema Privileges and Table Space Privileges

Schema Privileges

- **CREATEIN** allows the user to create objects within the schema
- **ALTERIN** allows the user to alter objects within the schema
- **DROPIN** allows the user to drop objects from within the schema

- To specify the schema owner other than the authorization ID used to execute the CREATE SCHEMA command:
  - CONNECT TO sample USER dbowner ;
  - CREATE SCHEMA dev AUTHORIZATION devusers ;

Table Space Privilege

- **USE OF TABLESPACE** privilege allows users to create tables within the specified table space
- USE privilege cannot be used with SYSCATSPACE or any system temporary table spaces

Table and View Privileges

- **CONTROL** provides the user with all privileges for a table or view, as well as having the ability to extend those privileges to others (except CONTROL)
- **ALTER** allows the user to add columns to a table, to add or change comments on a table and its columns, to add a primary key or unique constraint and to create or drop a table check constraint
- **DELETE** allows the user to delete rows from a table or view
- **INDEX** allows the user to create an index on a table
- **INSERT** allows the user to insert an entry into a table or view
- **REFERENCES** allows the user to create and drop a foreign key, specifying the table as the parent in a relationship
- **SELECT** allows the user to retrieve rows from a table or view, to create a view on a table, and to run the EXPORT utility
- **UPDATE** allows the user to change an entry in a table, a view, or one or more specific columns in a table or view. The user may have this privilege only on specific columns
  - Example: GRANT UPDATE (col1, col2) ON TABLE employee TO user1 ;
- The **ALL PRIVILEGES** option grants all the appropriate privileges, except CONTROL, on the base table, view, or nickname named in the ON clause
  - Example: GRANT ALL PRIVILEGES ON TABLE employee TO user1;

Package Privileges and Index Privilege

Package Privileges

- **CONTROL** provides the user with the ability to rebind, drop, or execute a package as well as the ability to extend those privileges to others (except CONTROL)
- **BIND** allows the user to rebind an existing package
- **EXECUTE** allows the user to execute a package

Index Privilege

- **CONTROL** allows user to drop the index

Routine Privilege and Sequence Privilege

Routine Privilege

- **EXECUTE** allows user to invoke a routine, create a function that is sourced from that routine (applies to functions only), and to reference the routine in any DDL statement such as CREATE VIEW, CREATE TRIGGER; or, when defining a constraint
- Example:
  - GRANT EXECUTE ON FUNCTION calc_salary(empno) TO jones ;
  - GRANT EXECUTE ON SPECIFIC FUNCTION calc_salary
    TO jones WITH GRANT OPTION ;

Sequence Privileges

- **USAGE** privilege allows user to use NEXTVAL and PREVVAL expressions for the sequence
- Example:
  - GRANT USAGE ON SEQUENCE org_seq TO PUBLIC ;

GRANT .... WITH GRANT OPTION

- Granting a privilege with the WITH GRANT OPTION allows the authorization ID to grant the specified privilege to others (conditions apply, refer to example below)
- Example:

  GRANT UPDATE ON TABLE calendar TO frank WITH GRANT OPTION

  - This statement allows frank to grant UPDATE ON TABLE calendar to others

  GRANT CONTROL ON TABLE calendar TO frank WITH GRANT OPTION

  - This statement will complete with a warning (SQLSTATE 01516) that CONTROL was not given the WITH GRANT OPTION
  - Frank now has the ability to grant any privilege on CALENDAR including INSERT and SELECT as required, however he cannot grant CONTROL on CALENDAR to other users unless he has SYSADM or DBADM authority

- The WITH GRANT OPTION is only available to GRANT statements of package, routine, schema, table, view, and table space

Implicit Privileges

- Grant DBADM
  - Implicitly granted BINDADD, CONNECT, CREATETAB, CREATE_NOT_FENCED, and IMPLICIT_SCHEMA privileges
- Create database
  - Grant DBADM authority with the implicit privileges to the creator
  - Grant CONNECT, CREATETAB, BINDADD, and IMPLICIT_SCHEMA privileges to PUBLIC
  - Grant SELECT on system catalog tables to PUBLIC
  - Grant USE privilege for table space USERSPACE1 to PUBLIC
  - Grant BIND and EXECUTE privileges on each successfully bound utility to PUBLIC
  - Grant EXECUTE WITH GRANT privilege on all functions in the SYSFUN schema to PUBLIC
- Create object (table, index, package)
  - Grant CONTROL privilege of object to creator

Implicit Privileges Scenarios

- **Scenario 1.**
  - ivo is placed in SYSADM group.
  - ivo creates database DB1
  - ivo is removed from SYSADM group.
  - What privileges does ivo retain?
- **Scenario 2.**
  - db2 connect to eddb
  - db2 grant dbadm on database to user mel
  - db2 revoke dbadm on database from user mel
  - What privileges does mel retain?

Implicit Privileges Scenarios

- **Scenario 1.**
  - ivo is placed in SYSADM group.
  - ivo creates database DB1
  - ivo is removed from SYSADM group.
  - What privileges does ivo retain?
    - **Answer: DBADM on DB1**

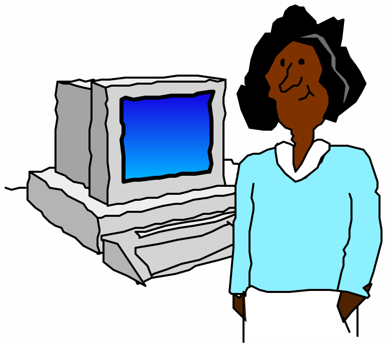
- **Scenario 2.**
  - db2 connect to eddb
  - db2 grant dbadm on database to user mel
  - db2 revoke dbadm on database from user mel
  - What privileges does mel retain?
    - **Answer: CONNECT, CREATETAB, BINDADD, IMPLICIT_SCHEMA, CREATE_NOT_FENCED**

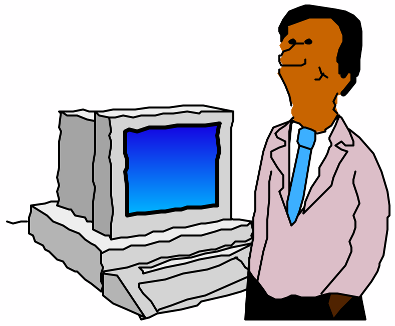
Privileges Required for Development of DB2 Applications

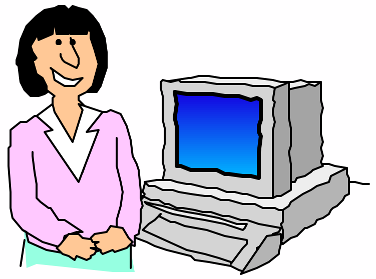
- All actions below require CONNECT on database

Action	Privileges Required
Precompile an application to a bind file	CONNECT on database
Create a new package	BINDADD on database Privileges for each static SQL statement (or PUBLIC)
Modify existing package	BIND on package Privileges for each static SQL statement (or PUBLIC)
Recreate existing package	BIND on package
Execute a package	EXECUTE on package
DROP a package	CONTROL on package or be the creator of the package

Authority & Privilege Scenario







Bob - End user who executes a program *app1* and use a table to track personal addresses

Clara - Application developer who will develop a program *app1* that contains dynamic SQLs

Steve - Needs to load tables

Susan - Wants to be able to create a database to store personal information

NEEDS

EXECUTE on package
CONTROL on table bob.personal

BINDADD on database
access to req'd objects
SELECT, INSERT, UPDATE DELETE on various tables

DBADM on the database or LOADauthority

SYSADM for the instance

All users require **CONNECT** authority on the database

Static SQL Requires explicit privileges granted to user or PUBLIC privileges

GROUP1

mel
patti
doug

prog1.sqc

...
Update T1
...
Select C1 from T1
...
Insert into T1

```
db2 connect to eddb

db2 grant update on table t1 to group1

db2 grant select on table t1 to public

db2 grant insert on table t1 to mel

db2 grant bindadd on database to group1
```

Mel attempts to bind

```
db2 connect to eddb
db2 bind prog1.bnd
```

Bind fails

- **no update**

Privileges Required for Static and Dynamic SQL programs

	Developer	User
Static SQLs	BINDADD + Explicit data access privileges or PUBLIC	EXECUTE
Dynamic SQLs	BINDADD + data access privileges	EXECUTE + data access privileges

System Catalog Views with Privileges Information

- Most of the information on authorizations is maintained in the following catalog views:

SYSCAT.DBAUTH	Database privileges
SYSCAT.TABAUTH	Table and view privileges
SYSCAT.COLAUTH	Column privileges
SYSCAT.PACKAGEAUTH	Package privileges
SYSCAT.INDEXAUTH	Index privileges
SYSCAT.SCHEMAAUTH	Schema privileges
SYSCAT.PASSTHRUAUTH	Federated server privileges
SYSCAT.ROUTINEAUTH	Routine privileges (functions, methods, and stored procedures)

DB2 Data Management Software

IBM

e business software

Chapter 4: Controlling Data Access

Authentication

Authorities

Privileges

Users and Groups

IBM Software Group

Group and User Support

- Privileges can be granted to groups.
- Groups defined through operating system security facility.
- UNIX permits groups and users to have the same name.

	Permitted on		*Does the System Know About?*	
SQL	**UNIX**	**Windows NT / 2000**	**User - cal**	**User - group**
1	✓	✓	✓	
2	✓	✓		✓
3	✓	***N/A***	✓	✓

Group - cal

User - cal

1
```
GRANT SELECT ON TABLE EMPLOYEE TO CAL
```
- or -
```
GRANT SELECT ON TABLE EMPLOYEE TO USER CAL
```

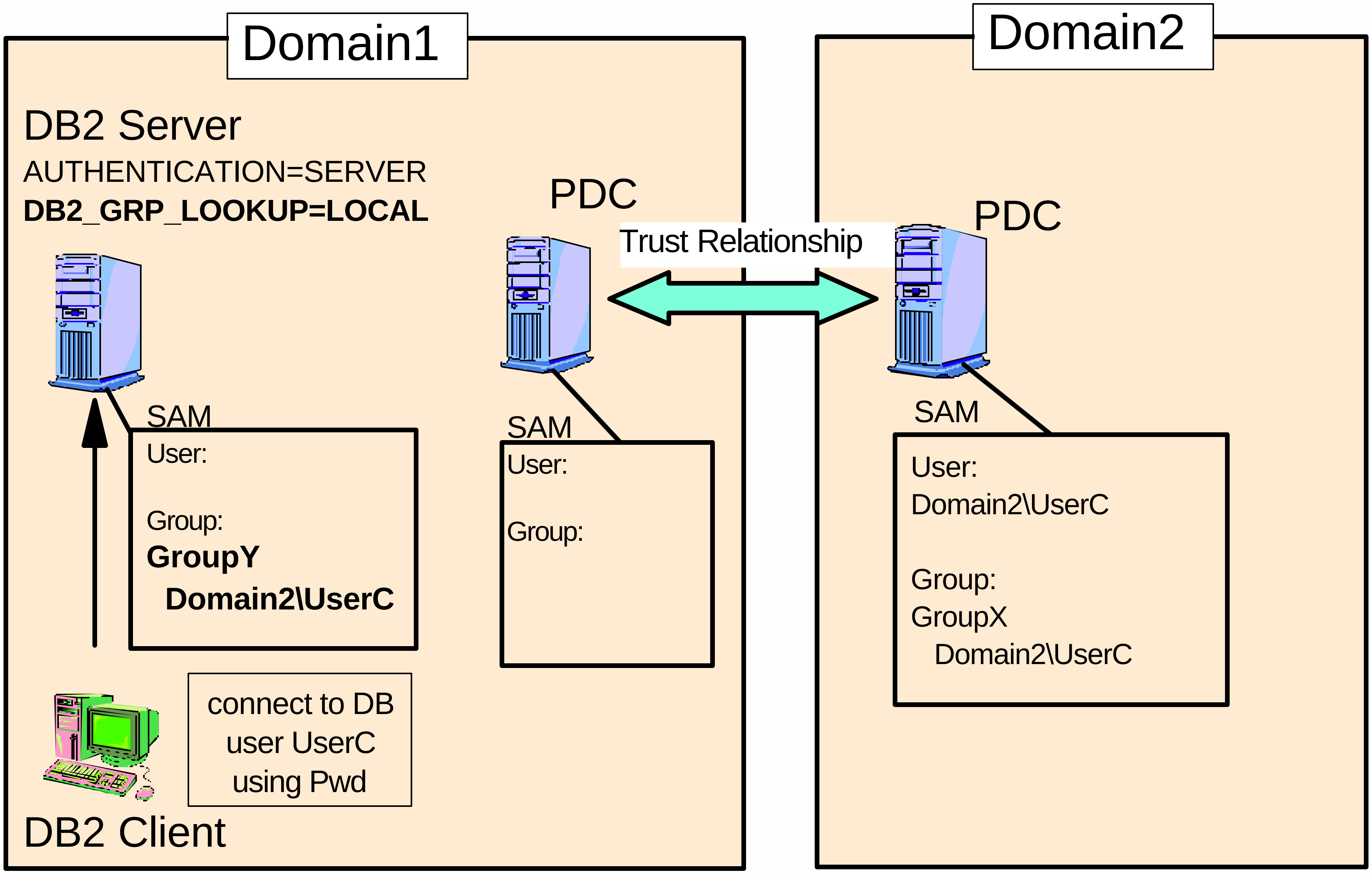
2
```
GRANT SELECT ON TABLE EMPLOYEE TO CAL
```
- or -
```
GRANT SELECT ON TABLE EMPLOYEE TO GROUP CAL
```

3 X
```
GRANT SELECT ON TABLE EMPLOYEE TO CAL
            SQLCODE -569
```

DB2_GRP_LOOKUP registry variable

- **Windows NT only.**
- **Addresses Windows NT's unique handling of users and groups.**
- **DB2_GRP_LOOKUP=(LOCAL,DOMAIN, NULL)**
  - Option allows you to specify how DB2 should proceed when attempting to enumerate groups that user belongs to
  - If not set (the default), then DB2 enumerates groups where the user id is defined.
- **DB2_GRP_LOOKUP=LOCAL**
  - Group enumeration done at DB2 server machine, not where the userid is defined
  - DBA does not need to be Windows NT domain administrator

DB2_GRP_LOOKUP=LOCAL



GroupY is the group for UserC

Notes:

In this example, although the user account UserC is found in the SAM of Domain2 (trusted domain), DB2 UDB enumerates GroupY from the local SAM as DB2_GRP_LOOKUP=LOCAL is set. The PDCs (Primary Domain COntrollers) of Domain1 and Domain2 are not searched for group information.

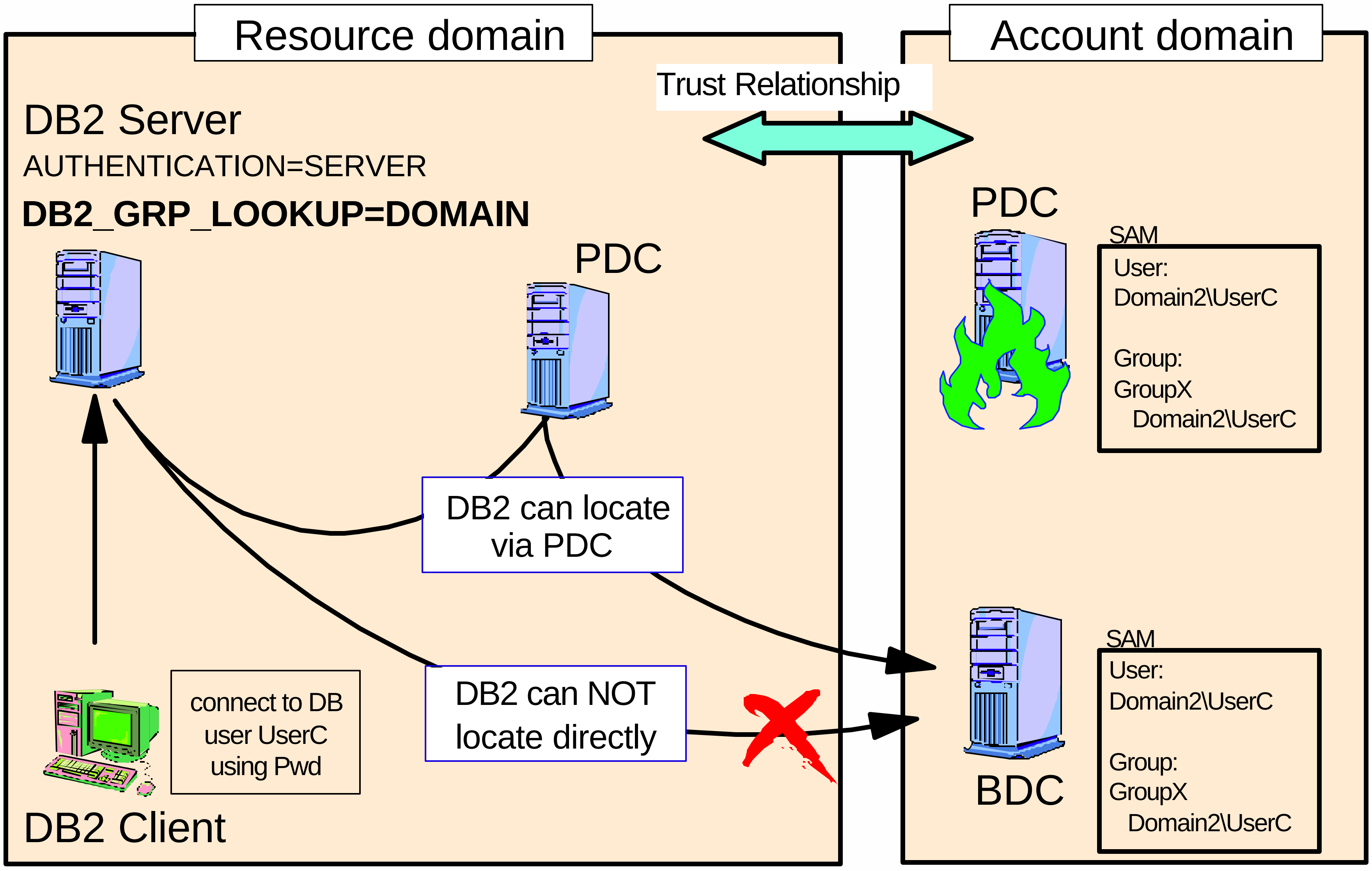
Let's assume that the DBA (assuming he/she is a Windows NT administrator of the DB2 server machine) wants UserC to behave as a SYSADM user. SYSADM_GROUP DBM CFG parameter has to be set to GroupY to accomplish this. The DBA can control membership of GroupY because he/she is a local administrator. It means he/she can control membership of the SYSADM group even if he/she is not a Windows NT domain administrator.

NOTE: If you specify **db2set DB2_GRP_LOOKUP=LOCAL,** then you must create local machine groups containing the qualified domain account names Domain2\UserC.

DB2_GRP_LOOKUP=DOMAIN

- DB2_GRP_LOOKUP=DOMAIN
- Specify to use a domain controller in current domain to locate a controller for the domain in which user account exists
- This value is significant when:
  1. DB2 server is not a domain controller
  2. DB2 server is in a domain other than domain where user accounts reside
  3. Accounts domain has backup domain controllers
- Why?
  - BDC of trusted domain cannot be looked up directly
    - unless DB2 server is a domain controller
  - Current domain controller is used to run the API to look up BDC of trusted domain if DB2_GRP_LOOKUP=DOMAIN is set

DB2_GRP_LOOKUP=DOMAIN



GroupX is the group for UserC

Notes:

This parameter applies to both client and server configurations. Setting this parameter will tell DB2 to use a domain controller in the current domain to locate a controller for the account domain. Setting DB2_GRP_LOOKUP=DOMAIN solves this problem.

In order to enumerate groups (and to find out whether you are an DB2 administrator, i.e. in the SYSADM group), DB2 uses an NT API to find a domain controller for the domain in which the account is defined (User accounts domain). It uses an API that tries to find the Primary Domain Controller (PDC) of the user account domain and, failing that, a backup domain controller. If the machine that is running this API is also a domain controller, this will always work. If your machine is not a domain controller, then this methodology will fail when the PDC of the account domain is down. When DB2_GRP_LOOKUP=DOMAIN, DB2 will find a domain controller in the current domain (Resource domain) with which to run the API to determine the domain controller for the account domain. This will not fail when the PDC of the account domain is down.