

May 6-10, 2007

San Jose Convention Center

San Jose, California, USA

## Session G13

# DB2 for z/OS and Websphere Integration Update



IDUG® 2007

North America

Maryela Weihrauch  
*IBM, Silicon Valley Lab*

Wednesday, May 10, 2007 • 10:40 p.m. – 11:40 p.m.

Platform: DB2 for z/OS



GoFurther

# Agenda

- Introduction to WAS and J2EE
- WebSphere Data Persistence
- Configuring DB2 DataSources within WebSphere
- Transaction Management
- Security and Auditing, DB2 z/OS V9 Trusted Context
- Views and 'INSTEAD OF' triggers
- Optimistic locking

# What is WebSphere?

- J2EE Execution environment for executing Java applications
- Built with abstraction and standards as central theme to remove reliance on vendors and difficulty of porting
- Build all the common application behaviors into the runtime environment:
  - Transaction Support
  - Data Persistence
  - Networking Support (location transparency)
  - Security, Error Handling
- Beans in J2EE
  - Servlets and JSPs
  - ✓ Does not have transaction support, data persistence, location transparency,
  - ✓ Generally used for presentation of information
  - Session beans
  - ✓ Contains mostly business logic
  - Entity beans
  - ✓ Used for data access

# EJB Persistence

- EJB 2.0 Container Managed Persistence (CMP)
  - requires a J2EE environment
  - generates SQL for database access, mostly single table SQL.
  - EJBQL to represent more complex SQL
  - too complicated to use, many limitations
- EJB 3.0 Java Persistence Architecture (JPA)
  - only requires a J2SE environment, Hibernate a major contributor
  - uses Java 1.5 annotation or XML for O-R mapping
  - generates SQL for database access, also joins
  - EJBQL is more flexible and native SQL can be included via named queries
  - much easier to use, more flexible and simpler runtime environment

## EJB 3.0 JPA

- BEA and IBM are main contributor to OpenJPA project, a Open Source JPA implementation
- IBM's JPA implementation is based on OpenJPA and optimized for accessing DB2 and IDS
  - planned to ship as FeaturePac for WebSphere 6.1 later in 2007
  - exploits WebSphere and DB2 features like
    - ✓ optimistic locking
    - ✓ changing connection properties at runtime, e.g. schema
    - ✓ native XML data type of DB2 V9
  - will support SQLJ and static SQL as integrated feature (generated SQL or named queries)

# JPA Sample

```
@Entity  
@Table (name="CUSTOMER")  
public class Customer {  
    private Long id;  
@Column (name="CUST_NAME")  
    private String name;  
    private Address address;  
    private Set<PhoneNumber> phones = new HashSet();  
// No-arg constructor  
    public Customer() {}  
// annotate either field or methods.  
@Id  
    public Long getId() {...}  
    public String getName() {...}  
    public Address getAddress() {...}  
@OneToMany  
    public Set<PhoneNumber> getPhones() {...}  
// Business method to add a phone number  
    public void addPhone(PhoneNumber phone) {  
        this.getPhones().add(phone);  
        phone.setCustomer(this); }  
}
```

# IBM DB2 Driver for JDBC and SQLJ

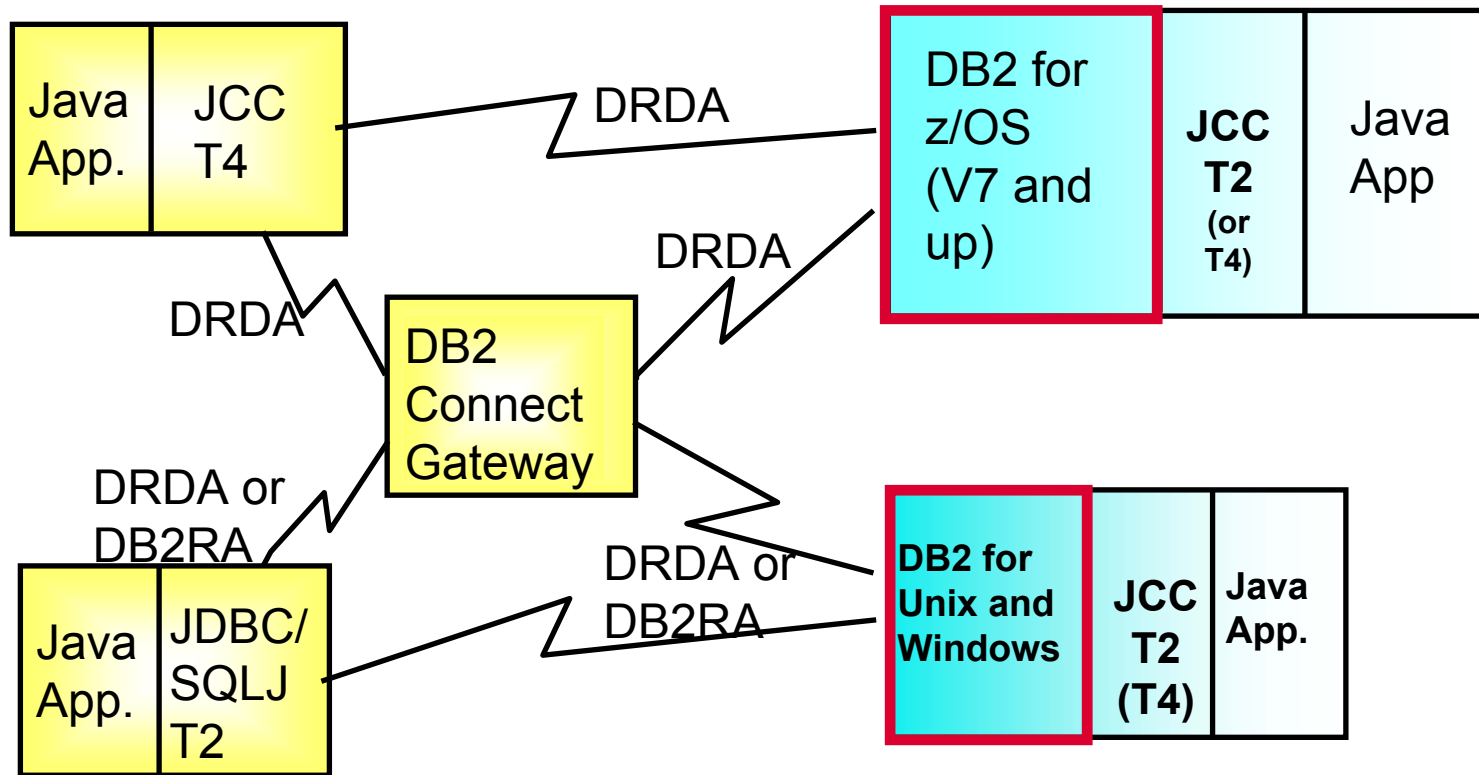
## ➤ Legacy JDBC drivers

- DB2 for LUW CLI based legacy driver (support until DB2 LUW V9, not the default starting with Viper II)
- JDBC / SQLJ Type 2 Driver for z/OS (only supported until DB2 z/OS V8 )
- Start planning the migration to the Universal JDBC Driver!

## ➤ Universal JDBC Driver (all platforms, common software, db2jcc.jar)

- Should be used for all new development
- Type 2 & Type 4
- JDBC 3.0 standard level

# Configurations





## Configuring WebSphere JDBC providers

- Tight integration between WAS Datasource & JDBC Driver
  
- WebSphere admin panels contain:
  - DB2 Universal JDBC driver
  - DB2 Universal JDBC driver (XA)
  
- For WebSphere on z/OS
  - Non-XA provides 2PC with T2 only (RRS provides 2PC)
  - XA version provides 2PC for Type 4.
  - Type 4 XA requires DB2T4XAIndoubtUtil to execute by SYSADM against target Database for DB2 UDB for z/OS V7 (not required for DB2 z/OS V8 NFM)

# WebSphere Connection Pool

The screenshot shows the WebSphere Administrative Console interface in Microsoft Internet Explorer. The browser address bar displays the URL: `http://carlasr31:9090/admin/preferenceAction.do?checkbox1=on&node1=System%2Fworkspace%23auto-refre`. The page title is "WebSphere Application Server Administrative Console Version 5". The breadcrumb navigation path is: `Resource Adapters > t1ResourceAdapter > J2C Connection Factories > t1J2C3 >`. The main heading is "Connection Pools". Below the heading is a descriptive paragraph: "Connection pool properties that can be modified to change the behavior of the J2C connection pool manager. Default values are provided for non-production use. Reviewing and possible modification of these configuration values is recommended." The "Configuration" section contains a table of properties:

General Properties	
Scope	cells:carlasr31Network:nodes:carlasr31
Connection Timeout	180 seconds
Max Connections	10 connections
Min Connections	1 connections
Reap Time	180 seconds
Unused Timeout	1800 seconds
Aged Timeout	0 seconds
Purge Policy	EntirePool

At the bottom of the configuration area are buttons for "Apply", "OK", "Reset", and "Cancel". The status bar at the bottom of the console shows "WebSphere Status", navigation links "< Previous" and "Next >", the date and time "February 13, 2003 2:28:17 PM EST", and the network status "Local intranet".

# WebSphere Connection Pool Properties

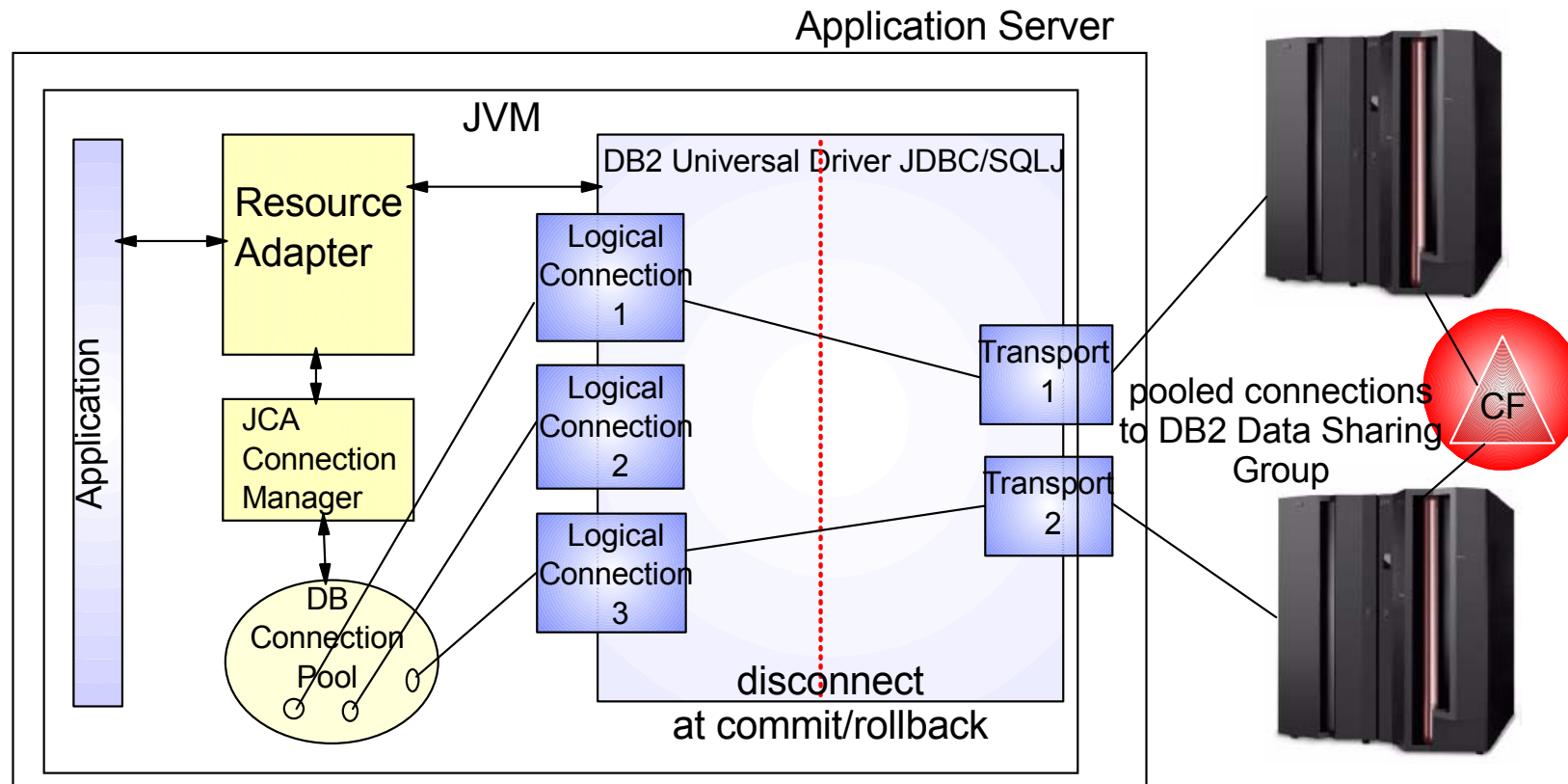
- Connection Timeout
  - How long to attempt connection creation before timeout
- Max Connections
  - max connections from JVM instance
- Min Connections
  - lazy minimum number of connections in pool
- Reap Time
  - How often cleanup of pool is scheduled in seconds
- Unused Timeout
  - How long to let a connection sit in the pool unused
- Aged Timeout
  - How long to let a connection live before recycling
- Purge Policy
  - After StaleConnection, does the entire pool get purged or only individual connection
- Statement Cache Size (different window)
  - Number of statements to keep in prepared Statement Cache (default 10)

## WAS/DB2 Active Thread - Tuning Considerations

- WAS connections in connection pool that keep an active thread in DB2 are target of the "idle thread timeout"
  - Type 2 on z/OS (RRSAF)
  - Type 4 connection that holds on to resources, e.g. WITH HOLD cursor, KEEP DYNAMIC YES.
- Set WAS "connection unused time" to a smaller value than DB2 "idle thread timeout" to avoid stale connection conditions.
- Consider setting min connections to 0 (zero) and connection unused time to not higher than 10 - 15 min to free up unused resources in DB2 in a controlled way and to reduce the exposure of long living threads

## Sysplex Workload Balancing and Connection Concentration

- Universal Driver Type 4 supports sysplex distribution and transparent failover at transaction boundary similar to DB2 Connect V7 and up



# DB2 z/OS Global Dynamic Statement Cache

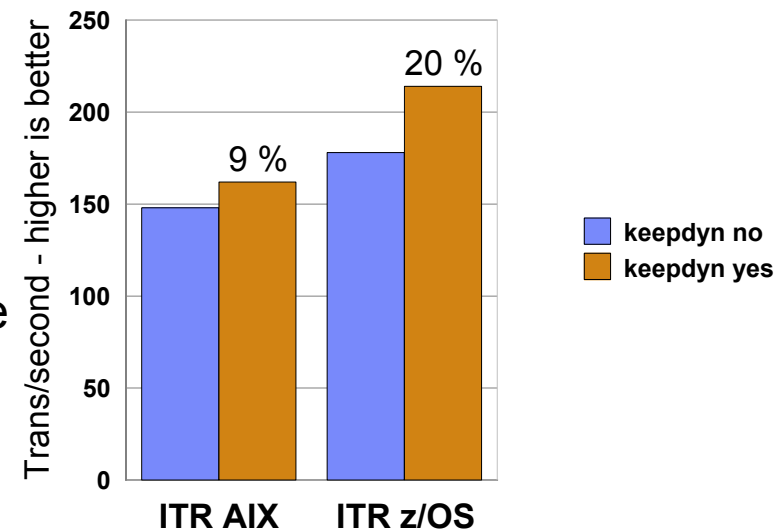
- Introduced in DB2 z/OS V5
- Allows applications to reuse and share prepared SQL statements in DB2
- Conditions for reuse of SQL statement from dynamic statement cache
  - SQL is dynamically prepared SELECT, UPDATE, DELETE or INSERT
  - The statement text is identical - character for character (literals problematic)
  - The authorization ID is the same
- ZPARM value CACHEDYN = YES turns on global cache
  - Statement text and executable of the prepared statements are kept in the EDM pool for reuse across all threads
- REOPT(ALWAYS) disables use of cache for that plan/packages

## WebSphere PreparedStatement Cache

- WebSphere manages a cache of previously created preparedStatement objects on a connection
- When a new prepared statement is requested on a connection, the cached preparedStatement object is returned if available on that connection
- Creating a new preparedStatement object is costly in Java besides the cost to prepare the SQL to DB2
- ✓ **Both, DB2 Global Dynamic Statement cache and WebSphere preparedStatement object cache should be used for best performance**

# KEEPDYNAMIC

- Is only effective when used in conjunction with PreparedStatement Cache
- Configure KEEP DYNAMIC for JDBC driver:
  - Rebind JDBC driver into second collection with KEEP DYNAMIC=YES
  - Set JDBC property jdbcCollection to new collection
  - Set JDBC property KeepDynamic=YES
- SQL statement & access path stored in z/OS local thread storage
- No prepare in client, across network, or optimization within DB2.
- Beware of impact to DBM1 Virtual Storage
- Distr. connection do not turn inactive, no transaction pooling





## Commitment issues... in EJB 2.0

- Enterprise java Beans should never issue JDBC commit
  - CMPs - transactions commit via deployment rules
  - Session Beans (CMT) – transaction commit via deployment rules
  - Session Beans (BMT) – transaction commit via User Transaction Interface (UTI)
  
- Servlets / JSPs –
  - If other resources involved, or EJBs referenced, UTI should be used to maintain transaction integrity
  - JDBC commit can be issued with no 2PC support.

## Transaction Support in EJB 3.0

- Controlled through JTA Entity Manager
  - Container-managed entity managers can only be JTA entity managers. JTA entity managers are only specified for use in J2EE containers
  - container-managed entity manager uses a container-managed persistence context to define a scope of a transaction.
- or Resource-local Entity Manager
  - entity manager whose transactions are controlled by the application through the EntityTransaction API.
  - Application-managed entity managers may be either JTA entity managers or resource-local entity managers.

## Resource-local Entity Manager API - Sample

```
// Create an Entity Manager EntityManager
em = emf.createEntityManager();
// get a Transaction
EntityTransaction tx = em.getTransaction();

// create a POJO instance of the Customer
// create a POJO instance of the Order

// Make the Customer and Order instances persistent
// and insert them into the database
tx.begin();
em.persist(customer);
em.persist(order);
tx.commit();
```

## DB2 Isolation Levels

- Isolation levels determine the types of locks acquired and how long they are held
- DB2 Isolation levels are RR, RS, CS, and UR
- WebSphere Isolation levels are Serializable, Repeatable Read, Read Committed, Read Uncommitted
- Data Sharing overhead can be significant for applications with RS

## Session Bean and Servlet Isolation Levels

- Prior to V5.1 Isolation level cannot be set within WebSphere via `con.setTransactionIsolation()` API
- Session and servlet isolation is determined via Resource, each resource has default level.
- DS Lookup should be `java:comp/env/jdbc/ds`
- Most applications can use CS with qualified update / delete / insert statements

The screenshot shows the 'References' view in the WebSphere IDE. The resource is named 'jdbc/T4DBD8' and is of type 'javax.sql.DataSource'. The authentication is set to 'Container' and the sharing scope is 'Shareable'. The JNDI name is 'jdbc/T4DBD8'. The isolation level is set to 'TRANSACTION\_READ\_COMMITTED' and the connection policy is 'TRANSACTION\_READ\_COMMITTED'.

## DB2 Authentication and Authorization with WebSphere

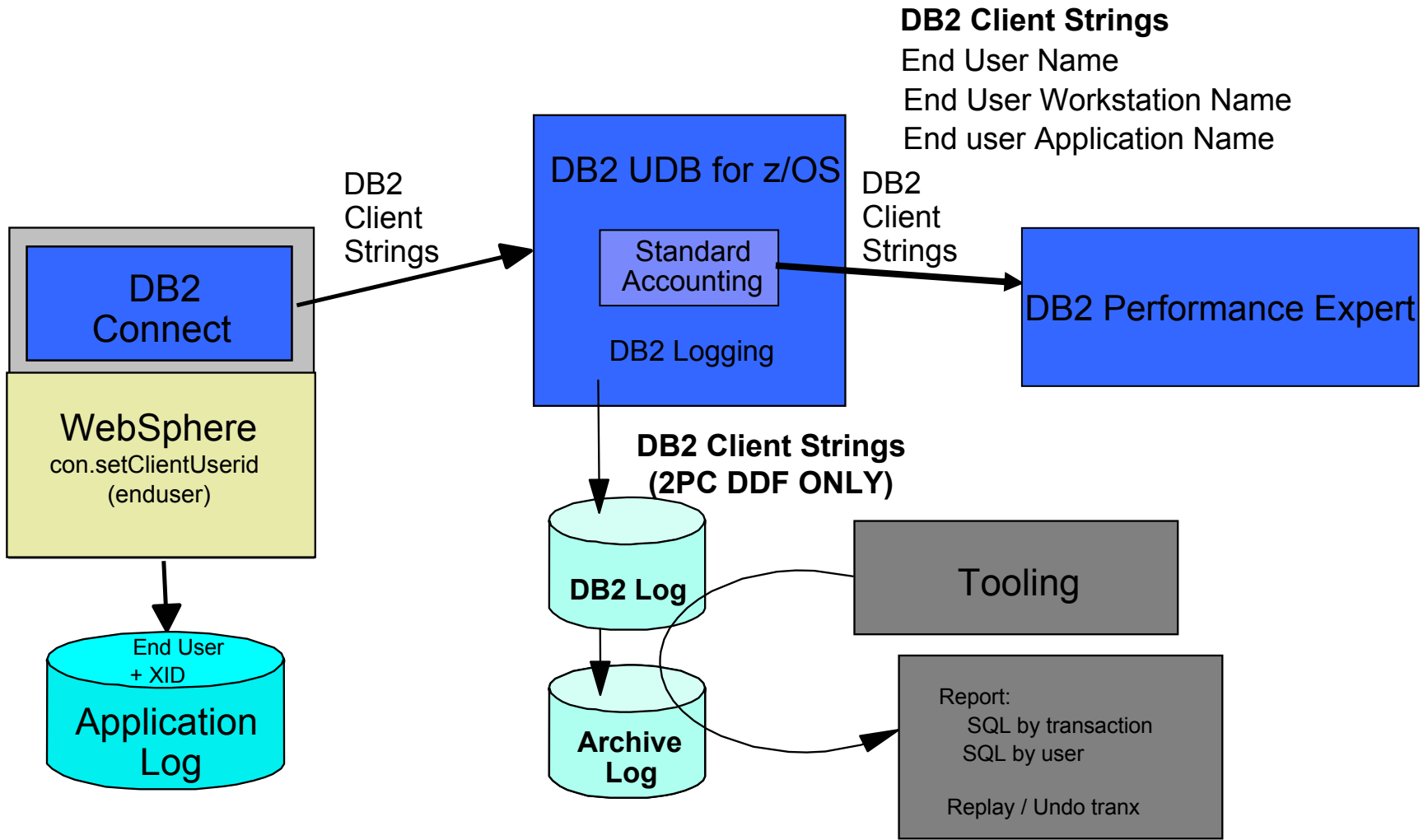
- Every End User has a RACF userid
  - Authenticate to z/OS LDAP (RACF exit)
  - ✓ Either code userid / password in application
  - ✓ Trust authentication to client with userid, potential security exposure
  
- End Users share a RACF ID
  - Authentication performed by WebSphere
  - Authorization implemented at group level. Each group represented by a valid RACF UserID
  - DataSource defined with JAAS ID (SAF)
  - Need a mechanism to support auditing....

# WebSphere V6 Client String API

```
import com.ibm.websphere.rsadapter.WSConnection;
try {
    InitialContext ctx = new InitialContext();
    //Perform a naming service lookup to get the DataSource object.
    DataSource ds = (javax.sql.DataSource)ctx.lookup("java:comp/jdbc/myDS");
} catch (Exception e) {System.out.println("got an exception during lookup: " + e);}

WSConnection conn = (WSConnection) ds.getConnection();
Properties props = new properties();
props.setProperty(WSConnection.CLIENT_ID, "user123");
props.setProperty(WSConnection.CLIENT_LOCATION, "127.0.0.1");
props.setProperty(WSConnection.CLIENT_ACCOUNTING_INFO, "accounting");
props.setProperty(WSConnection.CLIENT_APPLICATION_NAME, "appname");
props.setProperty(WSConnection.CLIENT_OTHER_INFO, "cool stuff");
conn.setClientInformation(props);
conn.close()
```

# End to End transaction Auditing



### DB2 Client Strings

- End User Name
- End User Workstation Name
- End user Application Name



## DB2 z/OS V9 Trusted Context

- Identifies “trusted” DDF, RRS Attach, or DSN application servers
- Allows selected DB2 authids on connections without passwords
  - reduces complexity of password management
  - reduces need for an all-inclusive “system authid” in app servers
- more visibility/auditability of which user is current running
- enables mixed security capabilities from a single app server

### DB2 Server

```
CREATE CONTEXT WAS1
  SYSTEM USERID WASPROD
  ADDRESS MY.WAS.SERVER
  ALLOW USER
  JOE WITHOUT AUTHENTICATION,
  SAM WITHOUT AUTHENTICATION;
```

SAM or JOE w/o  
password



MY.WAS.SERVER

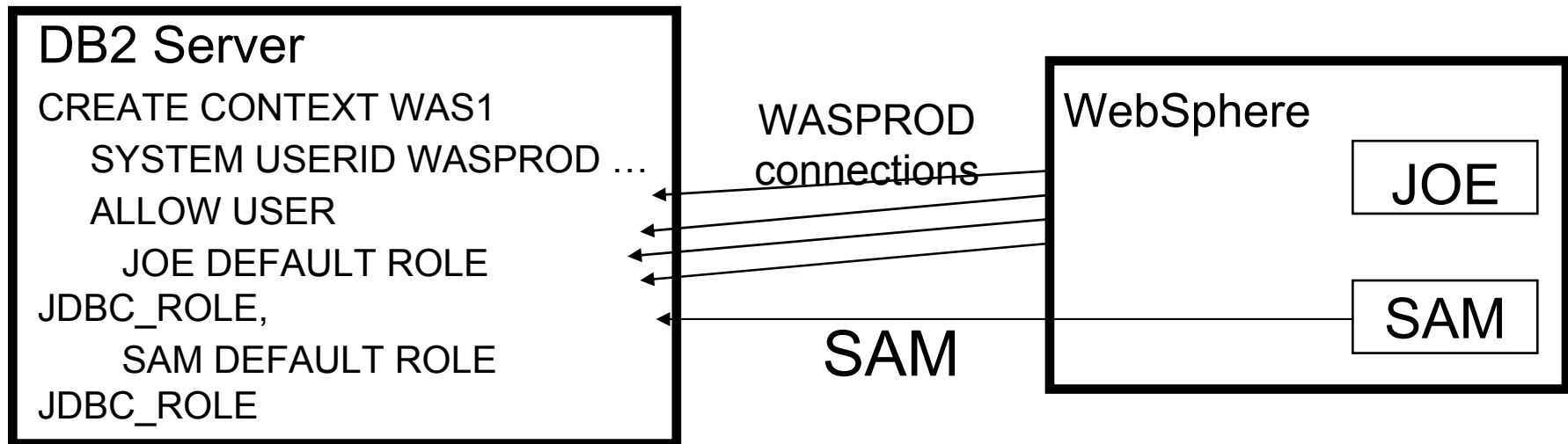
## Database Role

- ROLE is a “virtual authid”
  - Assigned via TRUSTED CONTEXT
  - Provides additional privileges only when in a trusted environment using existing primary AUTHID.
  - Can optionally be the OWNER of DB2 objects

```
CREATE ROLE PROD_DBA;  
GRANT DBADM ... TO PROD_DBA;  
  
CREATE TRUSTED CONTEXT DBA1 ...  
    DEFAULT ROLE PROD_DBA OWNER(ROLE);
```

## ROLEs and Trusted Context for 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!!!



## Views and V9 'INSTEAD OF' Trigger

- INSTEAD OF triggers are triggers that process instead of the update, delete or insert statement that activated the trigger.
- Unlike other forms of triggers that are defined only on tables, an INSTEAD OF trigger can only be defined on views.
- Allow complex SQL usage in generated Applications
  - DBA define view and INSTEAD OF trigger as logical layer
  - Applications operate on views as 'simple' objects, e.g. for O-R mapping

## Sample View

```
CREATE VIEW V_1 ( EMPNO, FIRSTNAME,  
                MIDINIT, LASTNAME,  
                WORKDEPT, DEPTNAME,  
                MGRNO, PHONENO, SALARY,  
                DEPTCHANGE, EMPCHANGE,  
                DEPTRID, EMPRID,  
                DEPTTOKEN, EMPTOKEN )  
AS SELECT EMPNO, RTRIM(FIRSTNAME),  
          RTRIM(MIDINIT), RTRIM(LASTNAME),  
          WORKDEPT, RTRIM(DEPTNAME),  
          MGRNO, PHONENO, SALARY,  
          DEPTCHANGE, EMPCHANGE,  
          RID(DEPT), RID(EMP),  
          ROW CHANGE TOKEN FOR DEPT,  
          ROW CHANGE TOKEN FOR EMP  
FROM DEPT, EMP  
WHERE WORKDEPT = DEPTNO ;
```

## INSTEAD OF Trigger Sample

```
CREATE TRIGGER TR1_UPDATE INSTEAD OF UPDATE ON V_1
  REFERENCING NEW AS N_ROW OLD AS O_ROW
  FOR EACH ROW MODE DB2SQL
  BEGIN ATOMIC
    UPDATE DEPT D
    SET D.DEPTNAME = N_ROW.DEPTNAME
    WHERE D.DEPTNO = O_ROW.WORKDEPT
      AND D.DEPTCHANGE = O_ROW.DEPTCHANGE
      AND ROW CHANGE TOKEN FOR D = O_ROW.DEPTTOKEN;

    UPDATE EMP E
    SET E.SALARY = O_ROW.SALARY * 1.20,
      E.LASTNAME = N_ROW.LASTNAME
    WHERE E.WORKDEPT = O_ROW.WORKDEPT
      AND E.EMPCHANGE = O_ROW.EMPCHANGE
      AND ROW CHANGE TOKEN FOR E = O_ROW.EMPTOKEN;

  END?
```

## Optimistic Locking

- Built-in timestamp for each row or page
  - Automatically updated by DB2  
GENERATED ALWAYS/BYDEFAULT FOR EACH ROW ON  
UPDATE AS ROW CHANGE TIMESTAMP
  - Allows simple timestamp predicate to validate that  
row has not changed since last access
- Eliminates need for complex predicates on  
WebSphere CMP/JPA updates, improves  
performance
- A new built-in function which returns the Record  
identifier (RID) of a row.

Session G13:

WebSphere and DB2 z/OS Integration Update

# Maryela Weihrauch

IBM Silicon Valley Lab.

[weihrau@us.ibm.com](mailto:weihrau@us.ibm.com)