A14

IMS Java Enhancements

(including WebSphere, DB2, and CICS)

Kyle Charlet

Miami Beach, FL October 22-25, 2001

RETURN TO INDEX

Overview

- **IMS Java Refresher**
- **J2EE Connector Architecture (JCA)**
- **Additional Environments**
  - IMS TM (JMP, JBP)
  - WebSphere
  - CICS
  - DB2
- **Driver Requirements**
  - JDBC 2.1
  - SQL Keywords
- **Tooling**

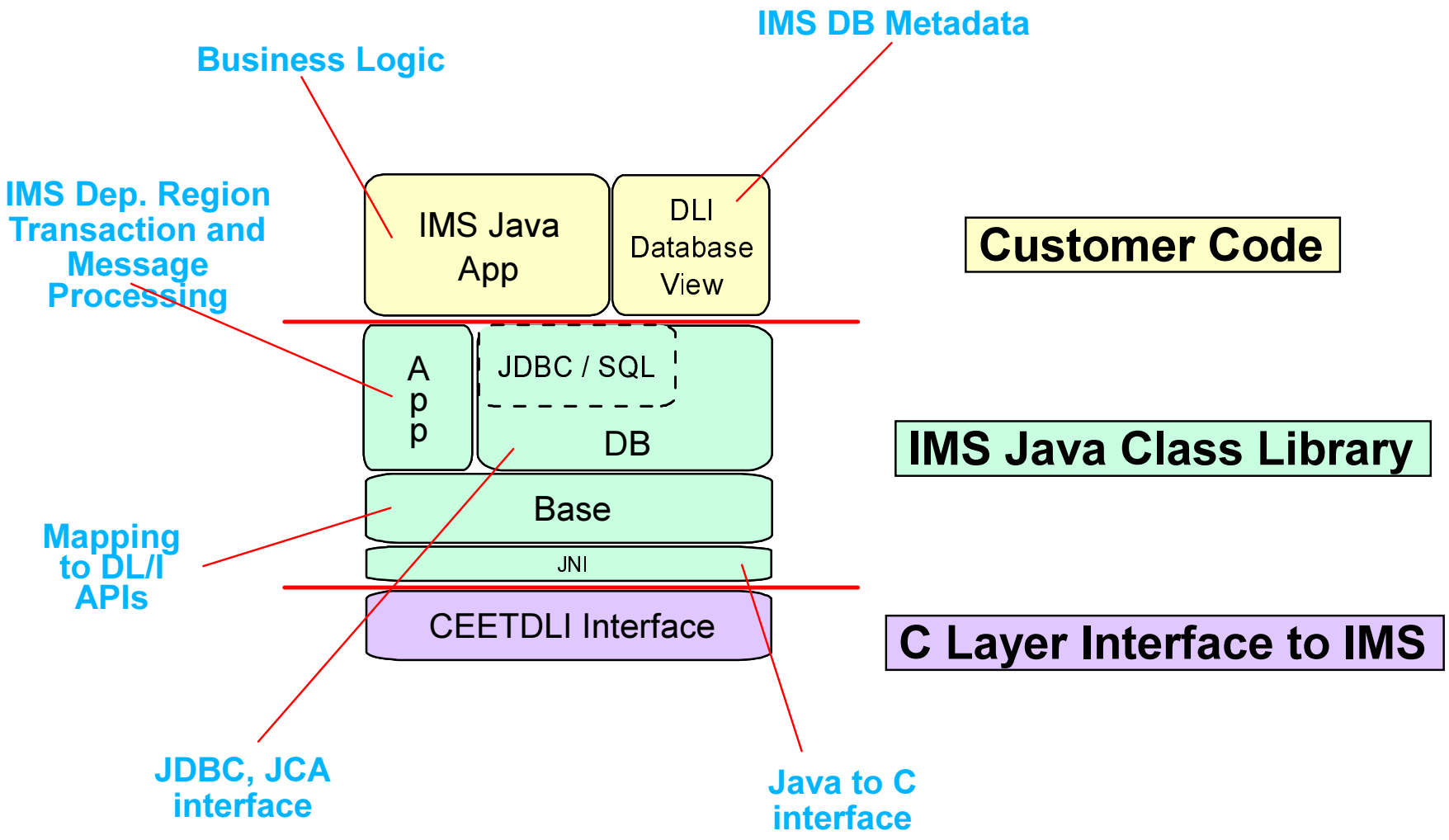
What is IMS Java?

- **A new feature in IMS v7**
- **A set of classes that...**
  - Offers Java support to access IMS Databases from various environments (CICS, DB2, WebSphere)
  - Enables SQL access through the JDBC interface
- **Java Virtual Machine (JVM) support in dependent regions**
  - JDK 1.3 support
  - JDBC 2.1 support
  - Just-In-Time (JIT) compilation
  - Resetable JVM
- **High Performance Java (HPJ) compiled**
  - Runs as a Language Environment run unit on IMS v7
  - Not supported on IMS v8 (including prior v7 HPJ applications)
  - Frozen at JDK 1.1.8
  - Frozen at JDBC 1.0

Why IMS Java?

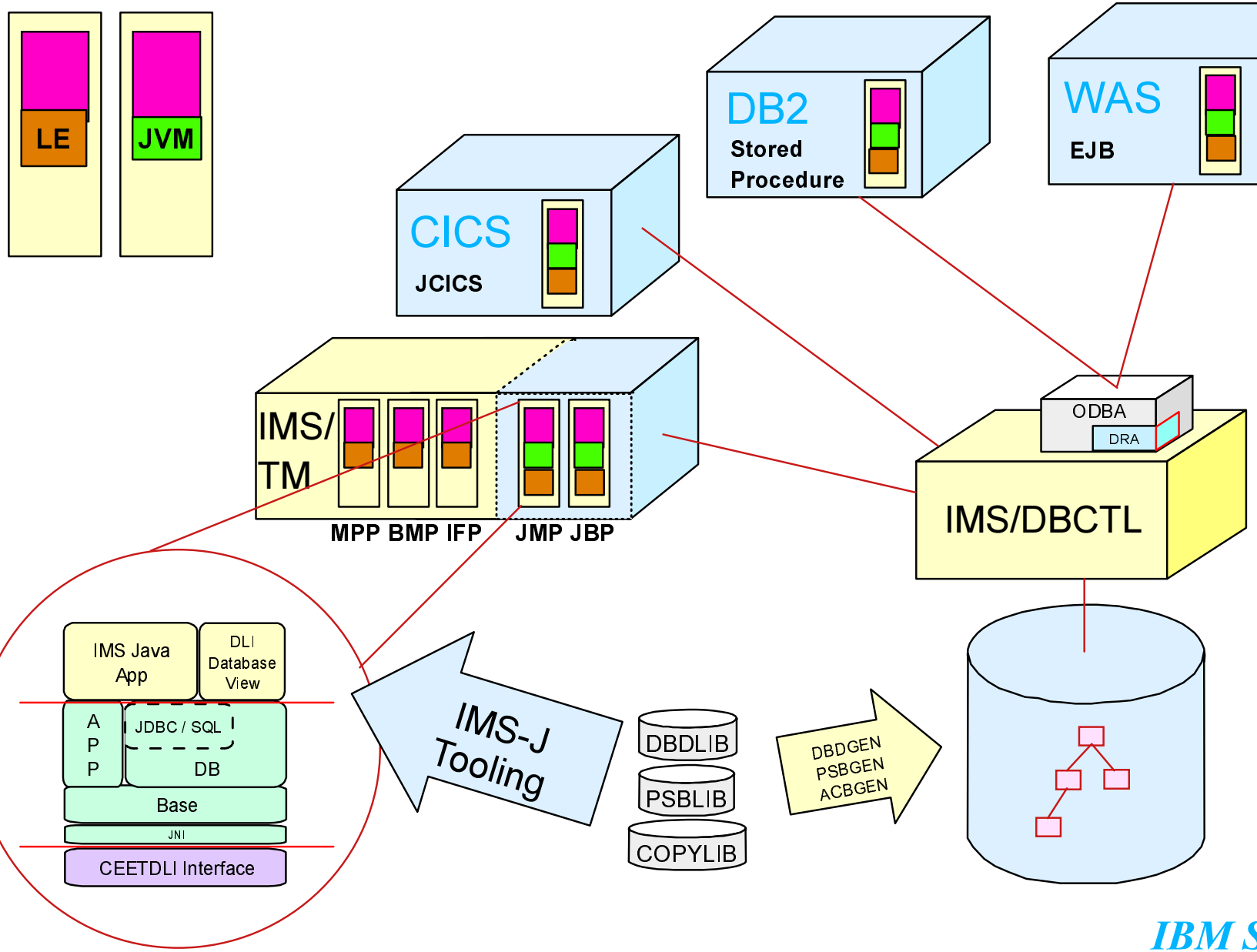
- **Colleges teach Java, very few still teach COBOL**
- **Colleges teach relational DBs with SQL access, very few teach hierarchical with SSA access**
- **JDBC is an industry standard**
  - Minimizes specific backend DB knowledge of IMS
- **Customer requests for Java support**

Java Class Library



IBM Software

IMS Java - The Big Picture

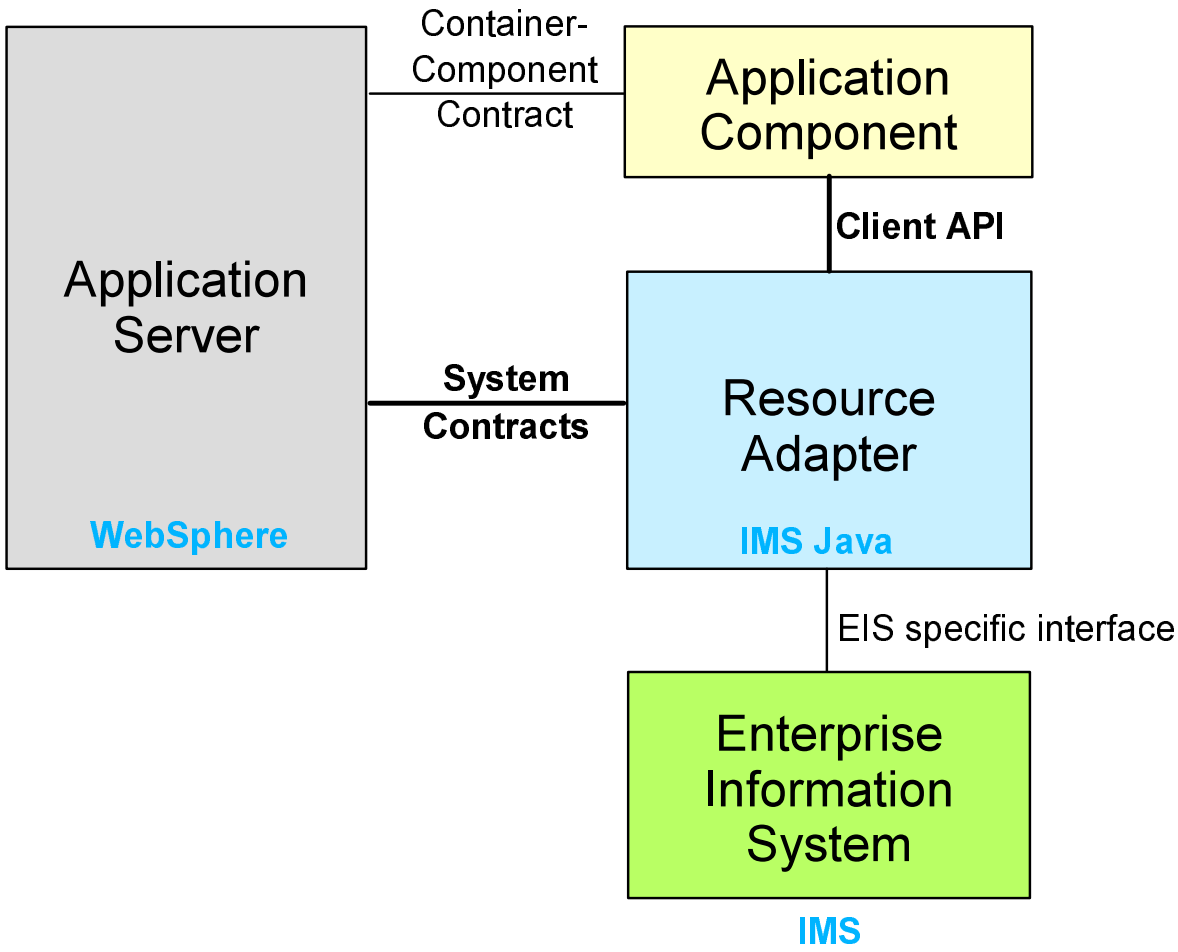


IBM Software

J2EE Connection Architecture

- **Proposed Java standard architecture for deploying a resource adapter in a J2EE compliant application server**
- **Defines contracts between...**
  - resource adapter and the application component
  - resource adapter and the application server

J2EE Connection Architecture



JCA Deployment

- **We provide a J2EE Resource Adapter ARchive (RAR) file containing:**
  - IMS Java runtime library (imsjava.jar)
  - Deployment descriptor (ra.xml)
  - Native code (libJavTDLI.so)
- **WebSphere provides tooling to deploy RAR file into server**
  - DataSource name (bound to JNDI namespace with this name)
    - Map a common ***logical*** name to a particular object and store the object persistently
    - Allow the retrieval, given the ***logical*** name
  - DLIDatabaseView subclass
  - DRA name

JCA Deployment (continued)

- **ManagedConnectionFactory**
  - Contains properties to create a particular connection
  - Deployed in JNDI namespace (in DataSource)

	Property Name	Type	Description
JDBC required	databaseName	String	Class name of DLIDatabaseView
	description	String	Description of this DataSource
IMS Java specific	databaseView	DLIDatabaseView	DLIDatabaseView Metadata
	draName	String	DRA name to connect through

***It is suggested that the draName is always included in a DLIDataSource, although draName is not required if running from a CICS application nor from an IMS dependent region.**

Managed Server Deployment

- **Managed Environment (WebSphere)**
  - DataSource created by server during deployment
  - Application makes a request for the DataSource and acquires a Connection from it

```
IMSJdbcManagedConnectionFactory mcf = new IMSJdbcManagedConnectionFactory();
mcf.setDRAName("DRA1");
mcf.setDatabaseViewName("myPackage.AppDatabaseView");
DataSource dataSource = (DataSource)mcf.createConnectionFactory();
Connection connection = dataSource.getConnection();

InitialContext ctx = new InitialContext();
ctx.bind("myDLIDatabase");
```

The server creates and binds a DataSource through JNDI.

```
Context ctx = new InitialContext();

DataSource dds = (DataSource)ctx.lookup("myDLIDatabase");
Connection con = dds.getConnection();
```

The application looks up the DataSource in the namespace and gets a database connection

IBM Software

Non-Managed Server Deployment

- **Non-Managed Environment (IMS, DB2, CICS)**
  - An application can construct a DataSource prior to its use
    - Alternatively, a DataSource can be put in the namespace
  - Application acquires a Connection from the DataSource

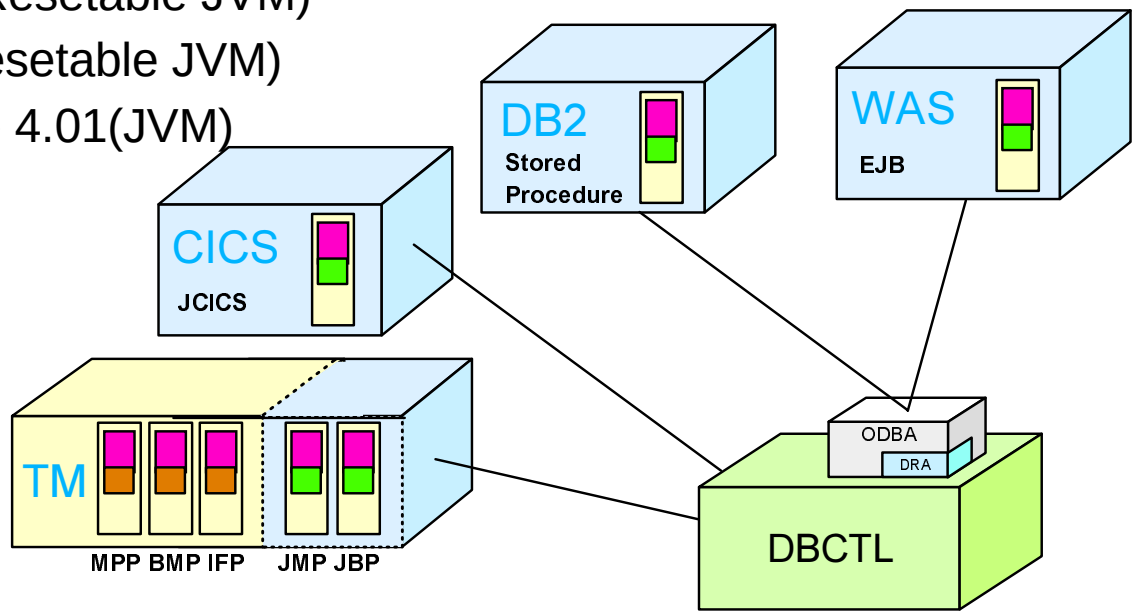
```
IMSJdbcManagedConnectionFactory mcf = new IMSJdbcManagedConnectionFactory();
mcf.setDRAName("DRA1");
mcf.setDatabaseViewName("myPackage.AppDatabaseView");
DataSource dataSource = (DataSource)mcf.createConnectionFactory();
Connection connection = dataSource.getConnection();
```

IMS JCA Implementation

- **Supports Connection and PreparedStatement pooling**
  - Connection pool is a cache of database connections maintained in memory
    - Does not impact application code
  - When a connection is closed it is released back to the pool and can be reused
  - We don't implement javax.sql.ConnectionPoolDataSource
    - Connection pooling always done in our implementation of javax.sql.DataSource
  - PreparedStatements are pooled per ManagedConnectionFactory
    - keyed set based on SQL query

Running Environments

- **Currently Supported Environments**
  - IMS TM 7.1 (HPJ)
- **New Environments**
  - IMS TM 7.1 (Resetable JVM)
  - IMS TM 8.1 (Resetable JVM)
  - CICS 2.1 (Resetable JVM)
  - DB2 V7 (Resetable JVM)
  - WebSphere 4.01(JVM)



IBM Software

Java Virtual Machine

- **A resetable JVM**
  - Persistent
    - Saves over 2,000,000 instructions loading JVM
- **Use is no different for user**
  - Applications do not change
- **Two new IMS dependent region types**
  - JMP (analogous to MPP)
  - JBP (analogous to non-message driven BMP)
  - No analogous IFP region
    - PCB access to DEDB and MSDB databases in JMPs and JBPs

Java Dependent Regions

- **JMP region type (Java Message Processing region)**
  - For message-driven Java applications
  - New IMSJMP JOB that EXECs the new DFSJMP procedure
  - DFSJMP procedure added to IMS.PROCLIB
    - Similar to the DFSMPR procedure for MPPs
    - Couple of new parameters
    - Several DFSMPR parameters not supported

- **JBP region type (Java Batch Processing region)**
  - For non-message driven Java applications
  - New IMSJBP JOB that EXECs the new DFSJBP procedure
  - DFSJBP procedure added to IMS.PROCLIB
    - Similar to the IMSBATCH procedure for BMPs
    - Couple of new parameters
    - Several IMSBATCH parameters not supported

Java Dependent Regions - DFSJMP Procedure

- **DFSJMP Procedure**
  - Starts a JMP region
  - Only a message driven IMS Java application can execute in this region
  - APPLFE=, DBLDL=, PRLD=, VSFX=, VFREE= are not supported
  - New parameter JVMOPMAS=
    - Specifies name of IMS.PROCLIB member that contains the JVM options for the master JVM
    - Required. If not present, region abends with ABENDU0101
  - New parameter JVMOPWKR=
    - Specifies name of IMS.PROCLIB member that contains the JVM options for the worker JVM
    - Required. If not present, region abends with ABENDU0101
  - New parameter ENVIRON=
    - Specifies name of IMS.PROCLIB member that contains the LIBPATH= environment variable specification
    - Required. If not present, region abends with ABENDU0101

Java Dependent Regions - DFSJBP Procedure

- **DFSJBP Procedure**
  - Starts a JBP region
  - Only a non-message driven IMS Java application can execute in this region
  - IN=, PRLD= are not supported
  - New parameter JVMOPMAS=
    - Specifies name of IMS.PROCLIB member that contains the JVM options for the master JVM
    - Required. If not present, region abends with ABENDU0101
  - New parameter ENVIRON=
    - Specifies name of IMS.PROCLIB member that contains the LIBPATH= environment variable specification
    - Required. If not present, region abends with ABENDU0101
  - The name specified on the MBR= parm represents either the actual name of the Java application or a symbolic for the actual name of the Java application (see DFSJVMAP PROCLIB member)

Java Dependent Regions - APPLCTN Macro

- **APPLCTN macro changes**
  - Added support for LANG=JAVA if GPSB= specified
  - Following error message issued if LANG=JAVA and FPATH=YES:

    G220 LANG=JAVA INVALID WHEN FPATH=YES.
  - In addition to being the name of the PSB, the name specified on the (G)PSB= parm also represents either the actual name of the Java application for message driven or a symbolic for the actual name of the Java application (see DFSJVMAP PROCLIB member)

Java Dependent Regions - PSBGEN Macro

- **PSBGEN macro changes**
  - Added support for LANG=JAVA
  - Must specify LANG=JAVA for any PSBs associated with a JMP type IMS Java application

Java Dependent Regions - IMSGEN Macro

- **IMSGEN macro changes**
  - Added SCEERUN= parameter
    - Specifies the name of the C Runtime Library
    - STEPLIB concatenation for DFSJMP and DFSJBP
    - Max of 44 alphanumeric characters for the name
    - Default is SYS1.SCEERUN

IMS TM Java Applications

- **IMSMessageQueue to receive and send input and output messages**
- **IMSTransaction to commit or rollback resources**
  - U118 abend if you do not commit/rollback
- **Supports conversational and non-conversational transactions**
- **Supports MFS options**

WebSphere 4.01 Support

- **Runs in JVM**
  - JDK 1.3
  - JDBC 2.1
- **Applications run as Enterprise Java Beans (EJBs)**
- **J2EE Connection Architecture (JCA)**
- **WebSphere does not support XA resource adapters**
  - Supports RRS compliant adapters
  - 2-phase commit via RRS
- **Access through ODBA/DRA**
- **Connection pooling provided by WebSphere ConnectionManager**

CICS TS Support for Java

- Allows a CICS program to be written in Java
- Allows CICS Java programs to interoperate with CICS programs written in any other CICS supported languages
- Allows a CICS application written in Java to access CICS services
- Provides a Java version of the CICS API known as the JCICS class library
- Supports Java development tools such as VisualAge for Java

CICS Java Access to IMS Data

- **Run in JVM**
  - CICS TS 2.1
  - JCICS
  - JDK 1.3
  - JDBC 2.1
- **Can only connect to (allocate) one PSB at a time**
  - Only one Connection active at a time in an application
- **Synchpoint done at deallocate PSB**

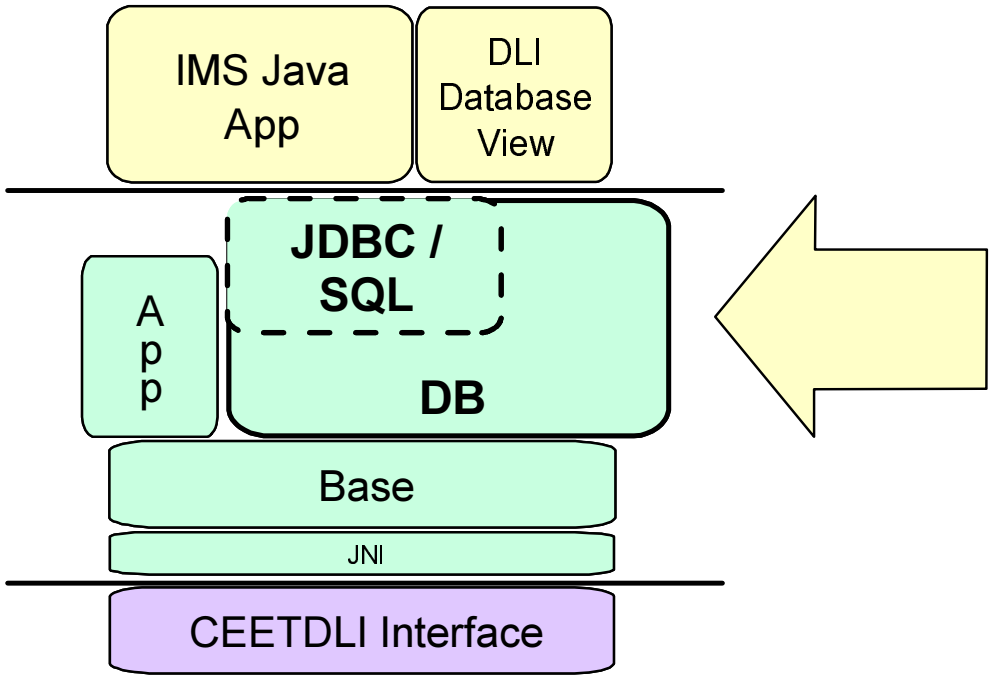
DB2 Stored Procedures

- User-written structured query language (SQL) programs that are stored at the DB2 server and can be invoked by a client application
- Can contain most statements that an application program usually contains
- Can execute SQL statements at the server as well as application logic for a specific function

DB2 Stored Procedure Support

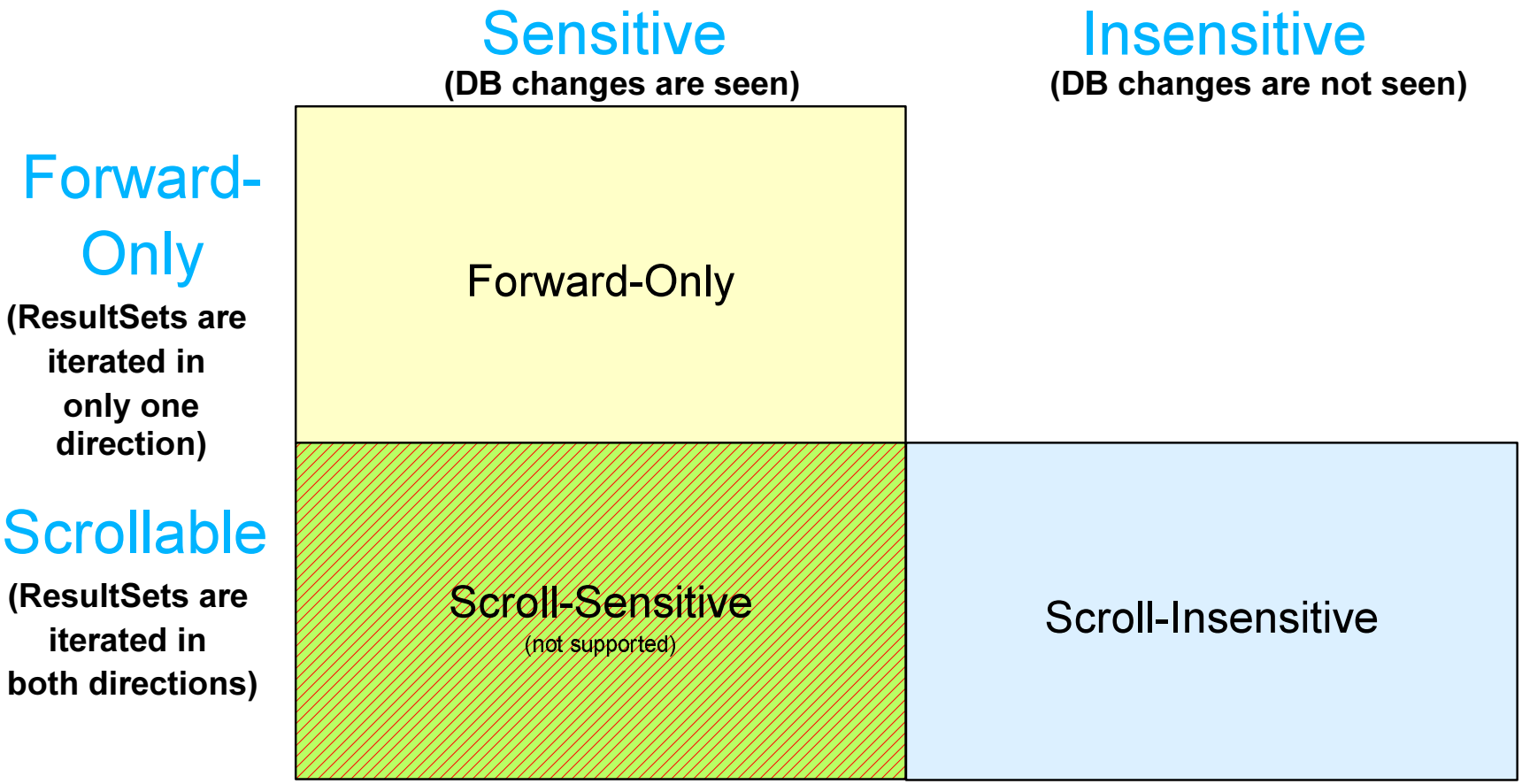
- **Run in JVM**
  - DB2 Version 7
  - APAR PQ46673 (resetable JVM)
- **Stored Procedures that access IMS Databases**
- **Return DB2 Result Sets**
  - populate temporary table
- **DB2 handles synchpoint (not stored procedure)**
- **DRA table and name required**

JDBC 2.1 Enhancements



- **DataSource**
- **ResultSet features**
- **Batch Updates**
- **New SQL Keywords**

IMS Java ResultSet Types



*IMS has no means to traverse a query backwards (iterate last to first through a result set)

IMS Java ResultSet Types

- **Forward-Only (currently supported) (default)**
  - Each next() call hits the DB
  - TYPE_FORWARD_ONLY
  - Calls:
    - ResultSet.next()
- **Scroll-Insensitive**
  - executeQuery hits DB, and caches all results
  - TYPE_SCROLL_INSENSITIVE
  - Calls:
    - ResultSet.next()
    - ResultSet.previous()
    - ResultSet.absolute(int)
    - ResultSet.relative(int)

IMS Java ResultSet Concurrency

- **Read-Only (default)**
  - CONCUR_READ_ONLY
  - Does not allow updates using the ResultSet interface
- **Updatable**
  - CONCUR_UPDATABLE
  - Allows updates using the ResultSet interface

*Concurrency is hard-coded into the PCB (PROCOPT) and cannot be modified

```
Statement stmt = con.createStatement(
                                ResultSet.TYPE_SCROLL_INSENSITIVE,
                                ResultSet.CONCUR_READ_ONLY)
```

IMS Java Batch Updates

- **Allows multiple update operations to be submitted for processing at once**
- **Similar to DB2 Stored Procedure**
- **Has little performance gain since not running over the wire**

```
Statement stmt = con.createStatement();

stmt.addBatch("INSERT INTO employees VALUES (1000, 'Eugene Dong')");
stmt.addBatch("INSERT INTO departments VALUES (260, 'Lingerie')");

// submit a batch of update commands for execution
int[] updateCounts = stmt.executeBatch();
```

New SQL keywords support

- **Field Renaming**
  - AS

```
SELECT EMPNO AS 'Employee Number'
FROM Employees
```

Display all the values in EMPNO in a column labeled Employee Number.

- **Aggregates**
  - AVG, COUNT, MAX, MIN, SUM, and GROUP BY

```
SELECT AVG(age), department
FROM Employees
GROUP BY department
```

Display the average age per department.

New SQL keywords support (cont...)

- **Ordering**
  - ORDER BY, ASC, DESC

```
SELECT firstName, lastName, department
FROM Employees
ORDER BY lastName ASC, firstName
```

Order by lastName in ascending order, followed by firstName in ascending order in the case of a tie.

- **Scalar Functions**
  - ABSOLUTE, +, -, *, /, UPPER, LOWER

```
SELECT UPPER(lastName) AS lastCaps
FROM Employees
```

List employees' last names in all CAPS, into a field named "lastCaps"

```
UPDATE Employees
SET Salary = Salary / 2
WHERE lastName = 'Holtz'
```

Halve employee Holtz's salary

New SQL keywords support (cont...)

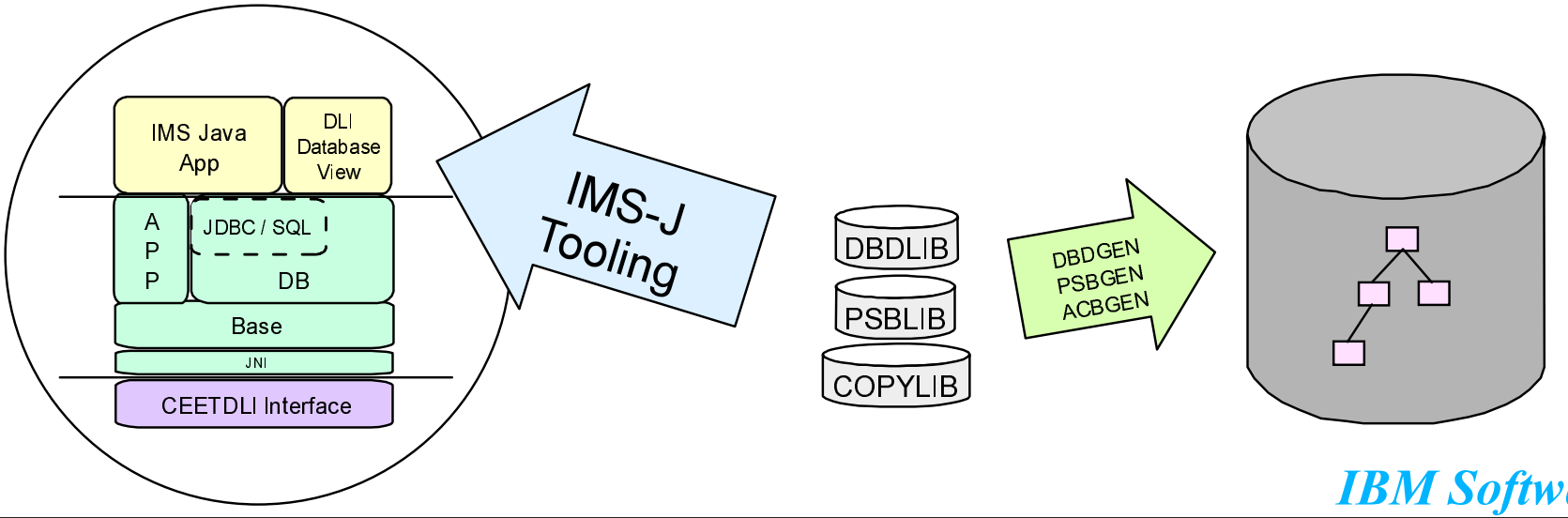
- **Unions**
  - UNION

```
SELECT firstName, lastName
FROM IBMEmployees
UNION
SELECT firstName, lastName
FROM SUNEmployees
```

List all employees working for IBM and SUN.

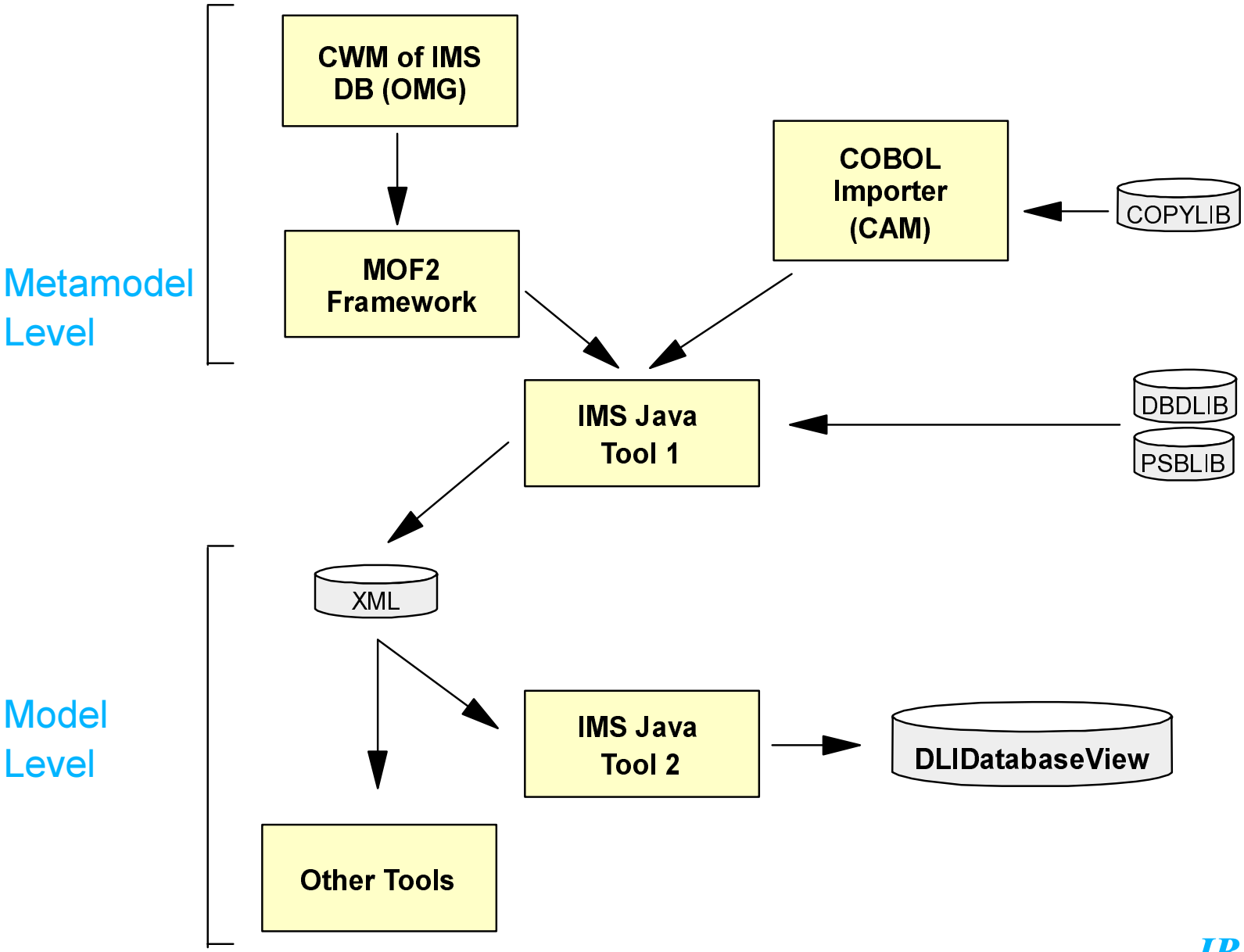
IMS Java Tooling

- **Problems:**
  - Confusion generating IMS Java Metadata (DLIDatabaseView)
  - Time consuming
  - Too prone to simple mistakes
- **Solution:**
  - Tooling support to:
    - Parse DBD, PSB, and COBOL CopyLibs
    - Produce XML to act as a standard form of IMS Metadata
    - Generate the IMS Java Metadata (DLIDatabaseView) from the XML



IBM Software

IMS Java Tooling

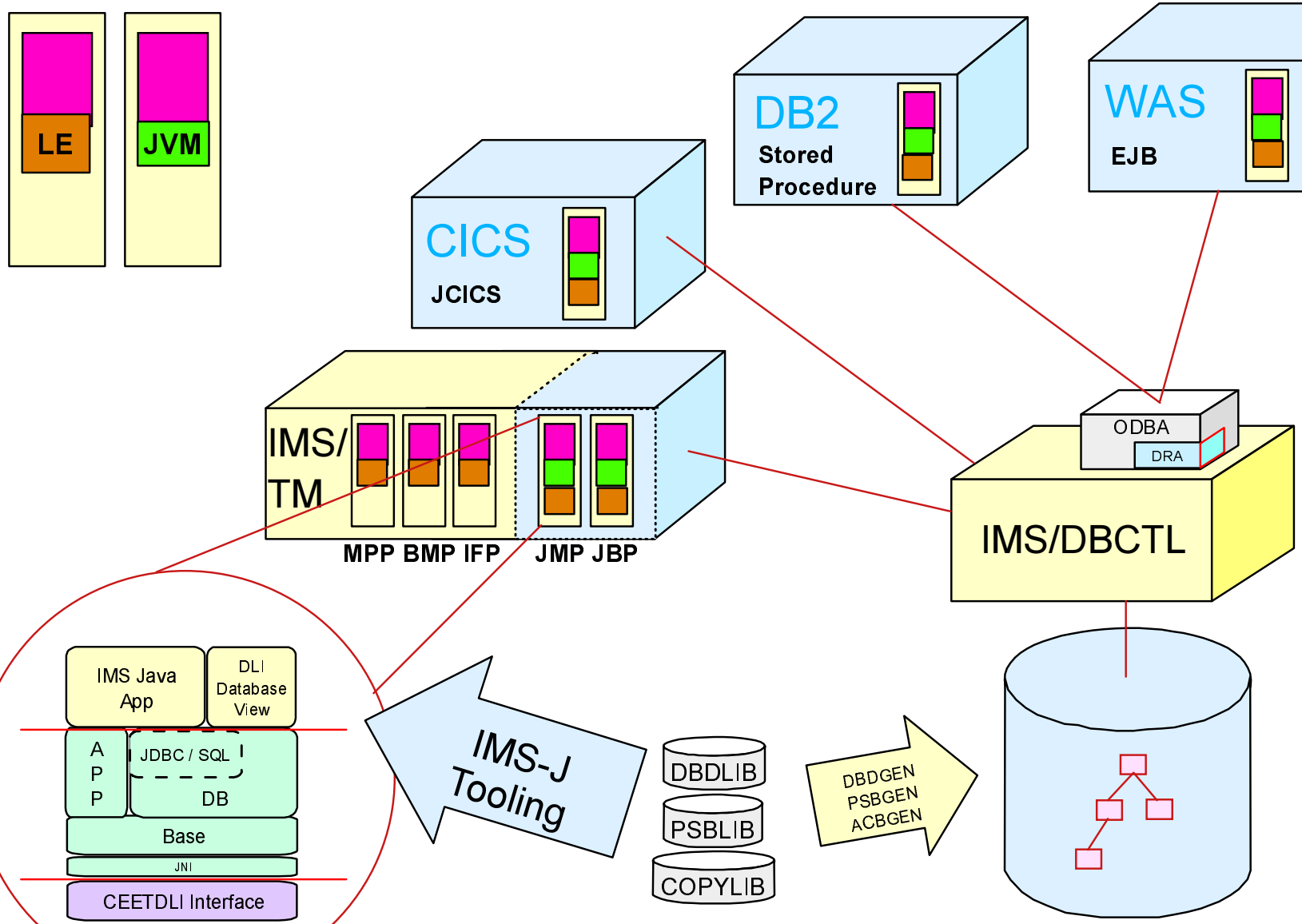


IBM Software

Tooling - Architectural Features

- **Conform to relevant standards:**
  - Common Warehouse Model (CWM) -- OMG
  - XML Metadata Interchange standard -- OMG
- **Develop two versions**
  - Version I: Stand-alone, command-line
  - Version II: WSA "pluggable", GUI

IMS Java - The Big Picture



IBM Software

RETURN TO INDEX